

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 البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière télécommunications  
Spécialité réseaux et télécommunications

Présenté par

**RAKOTOZAFY Njariniaina Emmanoella**

**DE SOUSA Dércio Afonso**

# CODAGE ET DECODAGE DES CODES POLAIRES PAR LA TECHNIQUE DE LISTE DANS UN CANAL GAUSSIEN

Proposé par :

- MAHDI Bersali

Membre des jury :

- ZINE Leila

- HOCINE Ait Saadi

Année Universitaire 2019-2020

**Résumé:** Dans ce travail, nous avons étudié les codes polaires qui sont, avec les LDPC, une des techniques de codage canal prévue par la 5 G. Les technique de codage polaire ainsi que celles décodage polaire par les algorithmes SCD et SCLD ont été développées ainsi avec des simulations montrant les performances du BER par rapport aux SNR.

**Mots clefs:** polarisation, arbre binaire, LLR, minsum, SCD, SCLD, BER, SNR

**Abstract:** In this work, we have studied the polar codes which are, together with the LDPCs, one of the channel coding techniques foreseen in 5G. The polar coding techniques as well as those of polar decoding by the SCD and SCLD algorithms were developed as well as simulations showing the performance of BER vs SNR.

**Keywords:** polarization, binary tree, LLR, minsum, SCD, SCLD, BER, SNR.

**الخلاصة:** في هذا العمل ، درسنا الرموز القطبية التي تعد ، مع LDPCs ، إحدى تقنيات تشفير القنوات المتوقعة في 5G. تم تطوير تقنيات التشفير القطبي وكذلك تقنيات فك التشفير القطبي بواسطة خوارزميات SCD و SCLD بالإضافة إلى عمليات المحاكاة التي توضح أداء BER مقابل SNR.

**الكلمات الرئيسية:** الاستقطاب ، الشجرة الثنائية ، LLR ، minsum ، SCD ، SCLD ، BER ، SNR

## **Remerciements**

C'est avec humilité et gratitude que nous tenons à remercier particulièrement notre directeur de mémoire Mr MAHDI Bersali, pour tous ses efforts et sa grande disponibilité de nous avoir encadré, aidé, orienté et conseillé durant la préparation de notre travail. Et, nous sommes très reconnaissants aussi envers toute l'équipe pédagogique de l'université de Saad-Dahleb et les corps professoral du département de génie électrique, responsables de notre formation pour avoir assuré la partie théorique que pratique de celle-ci et pour leurs accompagnements durant notre parcours universitaire. Et que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici nos sincères reconnaissances.

# Table des matières

Introduction Générale .....	1
Etat de l'art sur les performances des codes polaires .....	3
Chapitre 1 : Le principe d'une Communication Numérique .....	9
1.1 Introduction .....	9
1.2 Eléments d'une chaine de communication.....	9
1.3 Canal de transmission.....	12
1.3.1 Familles des canaux de transmission .....	13
1.3.1.1 Canal binaire symétrique (CBS) .....	13
1.3.1.2 Canal à effacement (CAE) .....	14
1.3.1.3 Canal à bruit blanc gaussien (BBAG).....	14
1.3.2 Support physique de transmission .....	16
1.3.2.1 Propagation libre .....	16
1.3.3 Modélisation d'un canal de transmission.....	16
1.3.4 Bande passante pour la 5G.....	19
1.3.5 Capacité de canal pour la 5G.....	20
1.3.6 Modèles des canaux .....	21
1.3.6.1 Canal Gaussien .....	21
1.3.6.2 Canal de Rayleigh .....	22
1.3.7 Caractéristiques des canaux de propagation .....	22
1.3.8 Modulation .....	27
1.3.8.1 Transmission en bande de base .....	29
1.3.8.2 Types de modulations numériques .....	29
1.4 Conclusion .....	33
Chapitre 2 : Les codes polaires .....	34
2.1 Introduction .....	34
2.2 Construction des codes polaires .....	35
2.2.1 Polarisation du canal .....	38
2.2.2 Conception des codes polaires .....	43
2.3 Processus de codage .....	44

2.4	Décodage par Annulation Successive .....	48
2.4.1	Décodage SC par graphe de factorisation .....	49
2.4.2	Décodage SC par arbre binaire .....	52
2.5	Spécificité des codes polaires pour le canal AWGN et le canal de Rayleigh ...	54
2.5.1	Canal AWGN .....	55
2.5.2	Canal de Rayleigh .....	56
2.6	Conclusion .....	57
Chapitre 3 Simulation des codes polaires .....		58
3.1	Introduction .....	58
3.2	Design SNR et séquence de rentabilité (reliability sequence) .....	58
3.3	Séquence de fiabilité (reliability sequence) .....	58
3.4	Encodage des codes polaires .....	64
3.5	Décodage de codes polaires .....	66
3.5.1	Décodage par SCD (successive cancellation decoder) .....	66
3.5.2	Décodage par SCLD (Successive Cancellation List Decoder) .....	74
3.6	Conclusion .....	80
Conclusion Générale .....		81
Références Bibliographiques .....		82

## Liste des Figures

Figure 1.1: Chaîne de transmission numérique .....	10
Figure 1.2: Canal CBS .....	13
Figure 1.3: Canal CAE .....	14
Figure 1.4: Canal BBAG .....	15
Figure 1.5: Spectre électromagnétique .....	16
Figure 1.6: Canal idéal .....	17
Figure 1.7: Modélisation d'un canal de transmission .....	17
Figure 1.8: Modèle de canal avec bruit additif .....	18
Figure 1.9: Modèle de canal de transmission à bande passante limitée .....	19
Figure 1.10: Schématisation des canaux de Rayleigh et de Rice .....	22
Figure 1.11: Scénario typique de propagation multi trajet .....	23
Figure 1.12: Schématisation de l'effet Doppler .....	24
Figure 1.13: Schématisation d'une réponse impulsionnelle dans le cas d'un canal variable du temps.....	25
Figure 1.14: Réponses temporelle (a) et fréquentielle (b) du modèle de canal COST 207 de la norme GSM-900 .....	26
Figure 1.15: Mapping à 2 bits pour la modulation QPSK .....	27
Figure 1.16: Modulation et Démodulation numérique .....	28
Figure 1.17: Exemples des modulations numériques .....	30
Figure 1.18: Rotation de constellation .....	32
Figure 2.1: Représentation en graphe de factorisation de l'encodage de codes polaires.....	37
Figure 2.2: Encodage systématique .....	38

Figure 2.3: Polarisation du canal. Adaptée d'Arikan (2009) .....	39
Figure 2.4: Codage canal et décodage canal d'un Code Polaire de taille $N=2$ .....	40
Figure 2.5: Répartition du nombre de bits suivant la probabilité d'effacement sur un canal à effacement pour différentes tailles de code, 64 et 1024 .....	42
Figure 2.6: Matrice génératrice et graphe de codeur de Code Polaire CP(2,2) .....	46
Figure 2.7: Matrice génératrice et graphe de codeur de Code Polaire CP(4,4) .....	46
Figure 2.8: Factor graph d'un codeur de Code Polaire CP(8,4) non systématique ....	48
Figure 2.9: Fonctions élémentaires et séquençement du décodage SC d'un noyau $N = 2$ .....	51
Figure 2.10: Arbre de décodage SC .....	53
Figure 2.11: Ordonnancement du décodage SC d'un code polaire de taille $N=4$ .....	54
Figure 3.1: Chaîne de transmission basique utilisant les codes polaire. F est une sous matrice de la 4 <sup>ème</sup> puissance de Kronecker .....	59
Figure 3.2: Position des bits gelés (f) selon la séquence de fiabilité, pour un code polaire CP(8,4) .....	60
Figure 3.3: Séquence de fiabilité (dans les colonnes paires) pour un code de taille $N=1024$ .....	62
Figure 3.4: Canal BAWGN .....	66
Figure 3.5: CP (2,2) a) arbre binaire pour un CP (2,2), b) principe du codeur .....	67
Figure 3.6: CP (4,4) a) arbre binaire, b) principe du codeur .....	69
Figure 3.7: CP (4,4) étapes 1 et 2 de décodage .....	69
Figure 3.8: Ordonnancement des profondeurs et nœuds dans un code CP (10,16).....	72
Figure 3.9: Simulation d'un code polaire CP(1024,500) avec un décodage SCD dans un canal AWGN sous BPSK comparé à un schéma sans codage .....	73

Figure 3.10: Schéma bloc d'un décodeur SCLD .....	75
Figure 3.11: Décision au niveau d'une feuille d'un arbre binaire .....	76
Figure 3.12: Organigramme d'un décodeur SCLD de taille $L = 4$ .....	77
Figure 3.13: Organigramme d'un décodeur SCLD de taille $L = 4$ à un nœud particulier .....	79
Figure 3.14: BER vs SNR dans le cas de décodage SCD, SCDL-4, SCDL-8 et SCDL- 11 pour un CP(500,1024) dans un canal AWGN sous modulation BPSK .....	80



## Liste des Abréviations

**3GPP:** 3rd Generation Partnership Project

**4-QAM:** 4-level quadrature amplitude modulation

**5G:** Cinquième Génération

**Ad hoc:** Advanced Developers Hands-On Conference

**ARQ:** Automatic Repeat request

**ASIC:** Application-Specific Integrated Circuit

**AWGN:** Addictive White Gaussian Noise

**CBS:** Canal binaire symétrique

**CAE:** Canal à effacement

**BAWGN:** Binary Addictive White Gaussian Noise

**BBAG:** Bruit Blanc Addictif gaussienne

**BCH:** Bose–Chaudhuri–Hocquenghem

**BEC:** Binary Erasure Channel

**BER:** Bit Error Ratio

**BP:** Belief Propagation

**Bps:** Bits par seconds

**BPSK:** Binary Phase Shift Keying

**BRD:** Blu-Ray Disk

**BSC:** Binary Symmetric Channel

**BSM:** Binary Symmetric Memoryless

**CD:** Compact Disc

**CP:** Code Polaire

**CRC:** Cyclic Redundancy Check

**COST 207:** Cooperation in the field Of Scientific and Technical research

**dB:** Décibel

**DM:** Decision Metric

**DVD:** Digital Versatile Disc

**FG:** Factor Graph

**FPGA:** Field Programmable Gate Arrays

**GHz:** Giga Hertz

**GSM:** Global System for Mobile communications

**GMSK:** Gaussian Minimum Shift Keying

**KHz:** kilohertz

**Km:** kilomètre

**LDPC:** Low Parity Density Check

**LLR:** Logarithmical Likelihood Ratio

**LOS:** Line Of Sight

**LR:** Likelihood Ratio

**M-PAM:** Multi-level Pulse Amplitude Modulation

**M-PSK:** multi-level pulse shift keying modulation

**M-QAM :** multi-level quadrature amplitude modulation

**MHz:** MegaHertz

**ML:** Maximum Likelihood

**ms:** millisecond

**NLOS:** *non-line of sight*

**nJ:** nanoJoule

**Np:** Neper

**OSI:** Open Systems Interconnection

**PM:** Path Metric

**PS:** Partial Sums

**QPSK:** Quadrature Phase Shift Keying

**RISC:** Reduced Instruction Set Computer

**RSB:** Rapport Signal Sur Bruit

**SC:** Successive Cancellation

**SCAN:** Successive Cancellation

**SCD:** Successive Cancellation Decoder

**SCF:** Successive Cancellation Flip

**SCL:** Successive Cancellation List

**SCLD:** Successive Cancellation List Decoder

**SCS:** Successive Cancellation Stack

**SNR:** Signal-to-Noise Ratio

**SSC:** Simplified Successive Cancellation

**TS:** Technical Specification

**Wi-Fi:** Wireless Fidelity

**WiMax:** Worldwide Interoperability for Microwave Access

# Introduction générale

La technologie ne cesse pas d'évoluer de générations en générations. L'intention d'améliorer la qualité de la communication et le fonctionnement de la transmission des données, et d'offrir de nouveaux services et applications à très hauts débits (objets intelligents...etc.) aussi que d'assurer la fiabilité des réseaux et la sécurité du système sur les canaux sans fil, constituent la principale obsession de nombreux chercheurs depuis des années.

A notre époque actuelle, le secteur des télécommunications connaît une évolution importante marquée par l'arrivée d'une nouvelle génération « réseaux 5G » qui semble être la solution pour répondre à tous ces besoins. Ces réseaux 5G commenceraient à être déployés après 2020 et ils se caractérisent par 3 paramètres pour l'amélioration des performances afin d'obtenir une haute capacité (en Km). Premièrement, les débits de données doivent être cent fois meilleures que les générations précédentes. Deuxièmement, ils nécessitent un temps de latence faible qui devrait être de 0,5 ms dans la liaison radio. Et troisièmement, les connexions doivent être cent fois plus que celles des générations précédentes.

Le problème est qu'en général, le canal sans fil est un canal à trajets multiples ou bruité qui varie continuellement dans le temps. Ce qui nécessite l'utilisation d'un code permettant corriger l'erreur dans les données reçues causée par le bruit et les interférences lors du codage canal, sans prendre le moindre risque de perdre beaucoup d'énergies pendant la transmission des informations. Les codes polaires sont conçus pour répondre à ces exigences dans l'intérêt des chercheurs et des industriels. Ce sont des codes classés comme correcteurs d'erreurs récemment inventés.

Ce mémoire s'intéresse particulièrement à la simulation des codes polaires dans un canal AWGN des réseaux 5G en utilisant la technique de liste.

Pour ce faire, nous avons réalisé des études théoriques et pratiques répartissant dans trois chapitres successifs.

Le premier chapitre constitue une étude générale sur le principe de communication numérique. Le second chapitre consiste à parler des codes polaires : leur définition et leur processus d'encodage et de décodage.

Et enfin, le troisième chapitre fait l'objet de notre contribution ayant trait aux études de la simulation des codes polaires dans un canal AWGN, enchaînées des résultats qui constitue la fin du chapitre. Mais avant d'entamer ces chapitres, nous allons présenter l'état de l'art sur les performances des codes polaires.

# Etat de l'art sur les performances des codes polaires

---

- **Codes polaires pour le codage source et canal** fait par Satish Babu Korada en 2009

Dans cette thèse, ils montrent que les codes polaires conviennent non seulement pour le codage de canal comme Arikan l'a démontré, mais permettent aussi d'obtenir des performances optimales pour plusieurs autres problèmes importants de la théorie de l'information. Le premier problème est la compression source avec perte. Pour y remédier, des codes polaires qui approchent asymptotiquement la distorsion de débit de Shannon pour une grande classe de sources ont été construits, en concevant des codes polaires selon le «canal de test», qui apparaît naturellement dans la formulation de Shannon de la fonction débit-distorsion. Au niveau de l'opération d'encodage, l'algorithme d'annulation successive d'Arikan a été combiné avec un nouvel ingrédient crucial appelé «arrondi aléatoire».

Leur travail a permis également la construction des codes polaires afin d'obtenir des performances optimales pour les problèmes Wyner-Ziv et Gelfand-Pinsker. Ces deux problèmes peuvent être résolus à l'aide de codes «imbriqués» et les codes polaires sont naturellement adaptés à cet effet. En outre, il s'agit de montrer que les codes polaires atteignent la capacité des canaux asymétriques, des scénarios multi-terminaux tels que des canaux d'accès multiples et des canaux de diffusion dégradés. En effet, c'est leurs constructions qui sont les premiers schémas «pratiques» connus se rapprochant de la performance optimale pour chacun de ces problèmes.

Selon Arikan, les codes polaires atteignent une probabilité d'erreur de bloc décroissant de manière exponentielle dans la racine carrée de la longueur de bloc. Mais dans leur travail, d'autres constructions de code de type polaire avec de meilleurs taux de

désintégration ont été explorés c.-à-d. que des désintégrations proches de l'exponentielle peuvent être obtenues pour le codage de canal et de source. De même qu'ils ont réalisé des nouvelles constructions, ces dernières restent toujours l'imitation de la construction récursive d'Arikan et, par conséquent, héritent de la même complexité d'encodage et de décodage.

Dans une autre partie, ils ont proposé aussi des algorithmes basés sur le passage de messages dans le but d'améliorer les performances de longueur finie des codes polaires.

➤ **Exploration architecturale pour le décodage de codes polaires** fait par Guillaume Berhault en 2015

Le sujet de cette thèse porte sur l'exploration architecturale de décodeurs de Codes Polaires implémentant des algorithmes de décodage particuliers. Ainsi, deux algorithmes de décodage ont été utilisés: un premier algorithme de décodage à décisions dures et un autre algorithme de décodage à décisions souples.

Le premier algorithme de décodage repose sur l'algorithme par annulation successive (SC) comme proposé originellement. Selon les recherches menées, le fait d'installer une architecture de mise à jour des sommes partielles à base de registres à décalages permet de réduire la complexité de l'unité de calcul des sommes partielles. Et ils ont proposé aussi une nouvelle méthodologie permettant de revoir la conception d'une architecture de décodeur déjà existante de manière relativement simple afin de réduire le besoin en mémoire. Pour caractériser ces contributions, des synthèses en technologie ASIC et sur cibles FPGA ont été effectués.

Le second algorithme de décodage, à décisions souples, traité dans ce mémoire, est l'algorithme SCAN qui a des meilleures performances que celles de l'algorithme BP. Si l'algorithme BP proposé dans l'état de l'art nécessite une cinquantaine d'itérations pour obtenir des performances de décodages au niveau de l'algorithme SC, l'algorithme SCAN à son tour en demande seulement 2 itérations. Et aussi, il est caractérisé par plus faible empreinte mémoire qui le rend encore plus pratique et permet l'implémentation de décodeurs plus grands. Cette comparaison a été faite après la première implémentation de cet algorithme sur cibles FPGA.

Toutes ces contributions proposées dans cette thèse ont permis d'apporter une réduction de la complexité matérielle du calcul des sommes partielles ainsi que du besoin général du décodeur en éléments de mémorisation. L'utilisation du décodeur SCAN permet alors d'ouvrir le champ d'applications des Codes Polaires dans la chaîne de communication avec d'autres blocs nécessitant des entrées souples.

- **Aspects de la mise en œuvre matérielle des décodeurs polaires et des décodeurs LDPC ultra-rapides** fait par Alexios Konstantinos BALATSOUKAS STIMMING en 2016

Dans cette thèse, leurs études se basent sur certains sujets qui forment une interaction entre les concepts théoriques de l'information abstract et l'implémentation matérielle.

Ainsi, un examen sur l'implémentation matérielle des décodeurs pour codes polaires a été fait. C.-à-d. grâce à une transformation algorithmique de haut niveau, il y a des améliorations significatives et des opportunités d'optimisation dans l'implémentation matérielle d'un décodeur de liste d'annulation successive pour les codes polaires.

Pour remédier à la complexité globale du décodeur dans les codes polaires avec une longueur de bloc relativement courte (par exemple,  $N \leq 256$ ) et de grandes tailles de liste (par exemple,  $L \geq 16$ ) pour une faible latence et / ou les applications à faible puissance et à faible débit, le tri métrique a été utilisé au niveau du décodage SCL. Par contre, d'après la comparaison faite par rapport au code LDPC et le Turbocode, les décodeurs SCL ne peuvent pas encore correspondre aux nombres à haut débit indiqués pour les décodeurs SC et BP en raison du fait que le décodage SSC rapide n'est pas totalement appliqué au décodage SCL.

Il semble évident que le décodeur SC lui-même et le décodeur SCL basé sur LLR sont tous les deux bénéfiques de toute amélioration architecturale et algorithmique. Cependant, la famille des décodeurs rapides SSC n'en est pas applicable.



➤ **Décodage de codes polaires sur des architectures programmables** fait par Mathieu Léonardon en 2018

Les travaux présentés dans cette thèse portent sur l'implémentation logicielle des algorithmes de décodage de codes polaires et la conception d'architectures programmables spécialisées pour leur exécution. Leur but c'est d'améliorer la flexibilité du réseau en utilisant des algorithmes sous forme d'une logicielle qui permet de mieux répartir l'effort de calcul sur l'ensemble des nœuds et d'améliorer la coopération entre cellules.

L'instabilité du canal de communication est l'une des caractéristiques principales d'une chaîne de communication mobile. Afin de remédier à ce problème, des techniques de modulations et de codages adaptatifs sont utilisées dans les normes de communication, auquel les décodeurs doivent être génériques.

Ce qui leur a poussé d'implémenter une logicielle de décodeurs génériques et flexibles avec des algorithmes de décodage "à Liste" sur des processeurs à usage général.

Par conséquent, cette première contribution a permis le pouvoir de correction, débit et latence de décodage par la paramétrisation fine des algorithmes, dont les débits arrivent même à atteindre et dépasser les performances de l'état de l'art.

Par la suite, une nouvelle architecture programmable performante spécialisée dans le décodage de codes polaires a été proposé dans la deuxième contribution. C'est une famille des processeurs à jeu d'instructions de type RISC de faible consommation, dédiés à l'application. Beaucoup de bénéfices ont été obtenus après la simulation, à savoir que cette architecture atteint des débits et des latences proches des implémentations logicielles de l'état de l'art sur des processeurs à usage général et que la consommation énergétique est réduite d'un ordre de grandeur.

La troisième contribution de ces travaux est également une architecture de processeur à jeu d'instructions dédié à l'application. Sa différence du précédent est qu'elle utilise une méthodologie de conception alternative c.-à-d. au lieu d'être basée sur une architecture de type RISC, l'architecture du processeur proposé est classé parmi les architectures déclenchées par le transport qui permet d'améliorer très significativement l'efficacité du processeur. En effet, les débits mesurés sont alors supérieurs à ceux obtenus sur les processeurs à usage général et la consommation énergétique est réduite à environ 0.1 nJ par bit décodé pour un code polaire avec

l'algorithme de décodage par annulation successive. Bref, la réduction est de deux ordres de grandeur en comparaison de la consommation mesurée sur des processeurs à usage général.

- **Contribution à la construction et au décodage des codes Polaires** fait par Ludovic Chandesris en 2019

Dans cette thèse, l'utilisation des techniques de poinçonnage et de raccourcissement a été faite pour explorer la question de la flexibilité des codes en termes de longueurs et elle développe aussi une méthode pour optimiser le choix des motifs de sorte à minimiser le taux d'erreur pour le décodeur par annulation successive (SC, pour successive Cancellation, en anglais).

Afin de corriger plusieurs erreurs, un nouvel algorithme de décodage est proposé, appelé décodeur dynamique à inversion, grâce à la mise à jour d'une liste d'inversions binaires, ordonnées par probabilités de succès décroissantes.

En ce qui concerne la question de la réduction de la complexité et de la latence du décodeur, un décodeur à inversion simplifié a été proposé et envisagée pour que les nœuds de rendement 1 du code n'aient plus besoin d'être décodés et pour avoir une dégradation très légère des performances. Il est possible que des similitudes entre le décodeur à inversion et le décodeur séquentiel pour les codes polaires existent, et notamment que la métrique proposée peut également être bénéfique pour le décodage séquentiel, en offrant un gain significatif de performance pour une même quantité de mémoire.

Dans le but de comparer la modulation multi-niveaux et la modulation à entrelacement de bits lorsque des codes polaires sont utilisés, la question de la combinaison d'un code polaire binaire avec une modulation d'ordre supérieur est considérée. L'approche multi-niveaux offre une excellente flexibilité et la possibilité de facilement résoudre le problème du choix des rendements sur les différents niveaux binaires pour le décodage par annulation successive.

Pour optimiser le choix des rendements lorsqu'un décodeur par liste est utilisé, une méthode de construction est également proposée, en s'efforçant d'équilibrer les taux d'erreurs estimés grâce à la borne de l'union, estimée à partir du nombre de mots de

code de poids faible obtenu par la formule introduite dans le chapitre 1. La mise en évidence de la modulation multi-level assure le processus successif des décodeurs polaires par un décodeur opérant continument sur les différents niveaux binaires, plutôt que d'utiliser des décodeurs séparés et garantit dans les deux cas un gain significatif en termes de performances, au prix d'une augmentation modeste de la complexité

Sur la question des codes polaires non-binaires dont un noyau non binaire de dimension  $l = 2$  est considéré, une méthode permettant un processus de décodage tout à fait similaire à celui des codes binaires est proposée afin d'optimiser chaque bloc de permutation du graphe du code non-binaire spécifiquement, aussi bien pour une permutation sur le groupe symétrique que dans le groupe général linéaire ou le corps de Galois. Les simulations montrent qu'un gain substantiel peut être obtenu en optimisant les coefficients dans le corps de Galois par rapport à un choix aléatoire, mais aussi que la différence entre l'optimisation dans le corps de Galois ou dans le groupe général linéaire reste très limitée.

# Chapitre 1 : Le principe d'une Communication Numérique

---

## 1.1 Introduction

Une communication numérique désigne l'ensemble des actions visant à diffuser des messages par le biais d'un media numérique tels l'internet , les medias sociaux, les applications mobiles et satellites ...etc.

Dans une communication numérique, on cherche à transmettre de l'information provenant d'une source vers un destinataire à travers un canal de transmission où l'on doit prendre en compte toutes sortes de caractéristiques et de contraintes que représente ce dernier (perturbations, traitement, pertes...etc.).

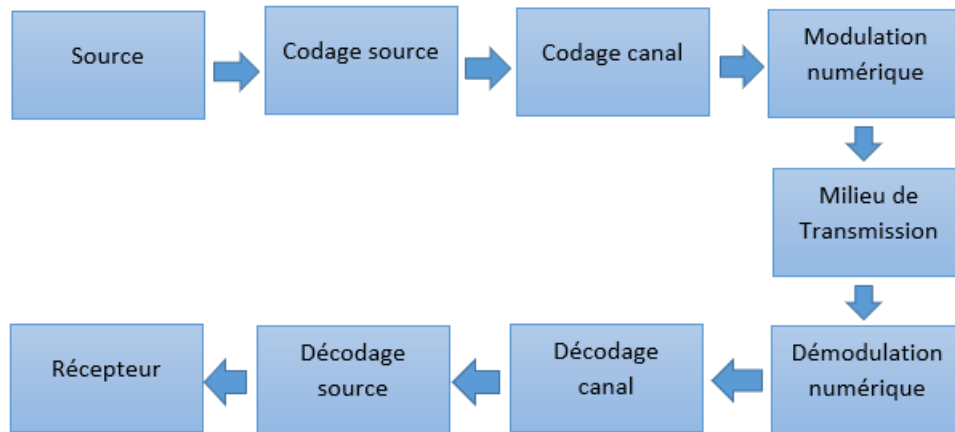
## 1.2 Eléments d'une chaine de communication

Il existe plusieurs modèles de communications dont la communication dite point à point qui est la base de tous les autres modèles tels:

- une communication de type multi-utilisateurs qui est une communication entre plusieurs sources - un destinataire,
- une communication de type diffusion (broadcast) qui est une communication entre une seule source et plusieurs destinataires,
- un réseau ad hoc (advanced developers hands-on conference) où plusieurs sources communiquent avec plusieurs destinataires. [1]

Une chaine de communication représente les principales étapes d'une transmission de données numériques depuis une source vers un destinataire à travers un canal de transmission dont le but c'est de transmettre, le plus vite possible (avec un débit

maximal), le maximum d'information et d'éviter les erreurs de transmission. Ce dernier est le support via lequel transite l'information.



**Figure 1.1:** Chaîne de transmission numérique

La figure ci-dessus (figure 1.1) représente une chaîne de transmission numérique, et est constituée:

- **d'une source** qui fournit le signal à émettre (voix, vidéo, image, texte...etc). Celui-ci peut être analogique ou discret. Dans notre cas nous nous intéresserons aux communications numériques où le signal est de type binaire. La source fournit un message sous forme d'une suite de 0 et 1 logique représentant deux niveaux de tensions.
- **d'un codeur de source** dont le rôle est de représenter le message sur un minimum de bits (compression de données). Ceci permettra de transmettre le maximum d'informations dans le minimum de temps.
- **d'un codeur de canal** qui a pour rôle de structurer le message pour permettre au récepteur de pallier aux erreurs de transmission. Cette modification du message génère un mot de code qui est une séquence de bits plus longue que celle du message. Cela peut se faire par simple adjonction de bits redondants mais aussi par des méthodes d'encodage plus élaborées.

- **d'un modulateur** qui a pour rôle de transformer en un signal analogique adapté au canal de transmission le signal binaire issu du codeur source. La forme la plus basique est d'associer respectivement aux niveaux logiques 0 et 1 deux sinusoïdes différentes en amplitude, en fréquence ou en phase ( $0 \rightarrow S_0(t)$  et  $1 \rightarrow S_1(t)$ ). Il existe également la modulation de type M-aire qui, plutôt que de transmettre le code bit par bit, elle permet de transmettre des paquets de  $b$  bits. Nous aurons donc  $M = 2^b$  paquets possibles ce qui nécessite  $M$  formes d'ondes  $S_i(t)$  différentes, avec  $i = 1 : M$ .
- **d'un démodulateur** ayant pour rôle inverse de celui du modulateur, il reçoit les ondes (éventuellement perturbées et les transforme en paquets de bits pour reconstituer le code sur lequel peut se trouver des erreurs de transmission.
- **d'un décodeur de canal** qui, grâce au codeur de canal à l'émission, il peut soit détecter une erreur de transmission et dans ce cas éventuellement réinterroger l'émetteur mais certains codages permettent, dans une certaine mesure, de détecter et de corriger les erreurs de transmission. En général les codes correcteurs d'erreur ne peuvent pas tout corriger car cela trop de bits redondants, ils se contentent de corriger les erreurs les plus probables.
- **d'un décodeur de source** qui a pour rôle de reconstituer l'information qui a été transmise. C'est l'opération de décompression. [2]

A souligner que le codage-décodage de canal est les parties de la chaîne qui nous intéressent tout particulièrement dans ce mémoire. C'est en effet ces parties de la chaîne de transmission qui mettent en place à l'émission, une structure permettant d'ajouter de la redondance (bits supplémentaires) au message à transmettre (voix, vidéo ou image numérisée, texte...etc.) et une autre à la réception ayant pour but de corriger les erreurs à partir de la redondance. L'ensemble s'appelle code correcteur d'erreur.

La transmission est modélisée par un canal à bruit additif gaussien seul ou en conjonction avec un canal multi trajet de type Rayleigh ou Rice et ce selon les phénomènes que l'on souhaite mettre en évidence.

### 1.3 Canal de transmission

Un canal de transmission est la liaison câblée ou non permettant la transmission d'une certaine quantité d'information depuis une source vers un destinataire.

En fonction du type de liaison (point à point ou diffusion), des contraintes économiques et matérielles, s'effectue le choix du media de transmission (câble coaxial ou, paires torsadées, fibres optiques, radio pour les communications Hertziennes, ...etc.). Et c'est au niveau de ce canal que s'ajoutent les perturbations et déformations (bruits, diaphonie, distorsions...etc.) de l'information. [3]

Dans les communications numériques, le canal de transmission est composé par : les entrées  $x_i$ , les sorties  $y_i$  et la probabilité de transition entre l'entrée et la sortie  $W(y/x)$ . Deux types de sortie du canal de transmission sont possibles : souple ou dure. La sortie souple donne une probabilité de la valeur du bit tandis que la sortie dure donne directement une valeur  $\{0,1\}$  car il fait une prise de décision. Les valeurs de la sortie dure sont plus simples à calculer mais elles ne sont pas optimales dans un décodage, et les valeurs de la sortie souple sont plus difficiles à calculer mais elles peuvent offrir un meilleur décodage.

Les canaux de transmission ont quatre propriétés fondamentales. Un canal de transmission doit être :

- **Discret** : Il s'occupe de garantir la transmission des nombres entiers, plus les binaires  $\{0,1\}$ .
- **Continu** : Son alphabet d'entrée et de sortie doit appartenir à l'ensemble des valeurs réelles.
- **Stationnaire** : Il doit avoir une probabilité de transition qui est indépendante du temps.

- **Sans effet mémoire** (Memoryless) : L'entrée d'un symbole est la seule responsable pour la sortie de ce même symbole, pas un autre symbole précédant ou suivant. [4]

### 1.3.1 Familles des canaux de transmission

Il existe trois grandes familles importantes des canaux de transmission. Ce sont le canal binaire symétrique (CBS), le canal à effacement (CAE) et le canal à bruit gaussienne (BBAG).

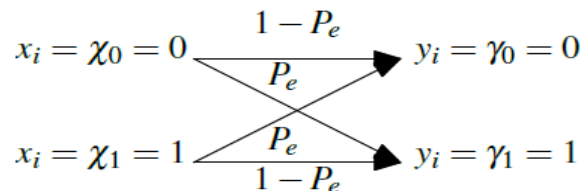
#### 1.3.1.1 Canal binaire symétrique (CBS)

Il est à entrée binaire  $x_i = \{0,1\}$  avec un ensemble de sortie  $y_i = \{0,1\}$  et ils ont des dimensions finies. Ces probabilités de transition du canal permettent d'avoir la sortie  $y_i$  en sachant l'entrée  $x_i$ , sans effet mémoire, et il est indépendant de la variation du temps.

$$P_r(y_i = y_0/x_i = x_1) = P_r(y_i = y_1/x_i = x_0) = P_e \quad (1.1)$$

$$P_r(y_i = y_0/x_i = x_0) = P_r(y_i = y_1/x_i = x_1) = 1 - P_e \quad (1.2)$$

Où  $P_e$  correspond à la probabilité d'erreur. [4]



**Figure 1.2:** Canal CBS [4]

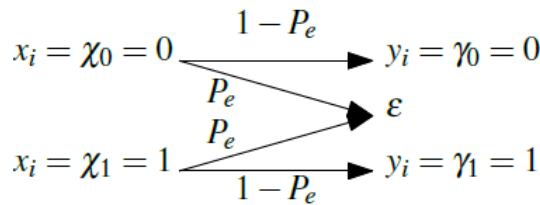


### 1.3.1.2 Canal à effacement (CAE)

Cette famille de canaux de transmission est similaire au CBS, la grande différence est basée sur le fait qu'elle a 3 sorties possibles  $\{0, \varepsilon, 1\}$ , où  $\varepsilon$  correspond à une perte totale d'information (Effacement). Les probabilités de transition du canal se définissent comme suit :

$$P_r(\text{effacement}) = P_e \quad (1.3)$$

$$P_r((y_i = y_0/x_i = x_0) = P_r(y_i = y_1/x_i = x_1) = 1 - P_e \quad (1.4)$$



**Figure 1.3:** Canal CAE [4]

### 1.3.1.3 Canal à bruit blanc gaussien (BBAG)

Ce canal a la particularité d'avoir une sortie  $y_i$  continue à partir d'une entrée  $x_i$  discrète. Il est un canal continu lorsque les deux canaux précédant sont discrets. Mais le reste de ses propriétés est similaire aux autres canaux précédents. Actuellement, ce type de canal est très utilisé dans le domaine des communications numériques.

Le BBAG ou AWGN (Addictive White Gaussian Noise) est un bruit qui s'ajoute automatiquement au signal d'entrée du canal et il possède la même densité de probabilité pour toutes les fréquences. Au lieu d'avoir tout simplement la probabilité de transition du canal, l'utilisation de sa densité de probabilité est important :

$$P_r(x_i/y_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - y_i)^2}{2\sigma^2}\right) \quad (1.5)$$

Où  $\sigma$  est la variance du signal

Cette variance est une fonction du RSB (SNR), donnée par :

$$RSB = \frac{E_b}{N_0} = \frac{1}{2\sigma^2} \quad (1.6)$$

Où :  $E_b$  correspond à l'énergie moyenne du bit

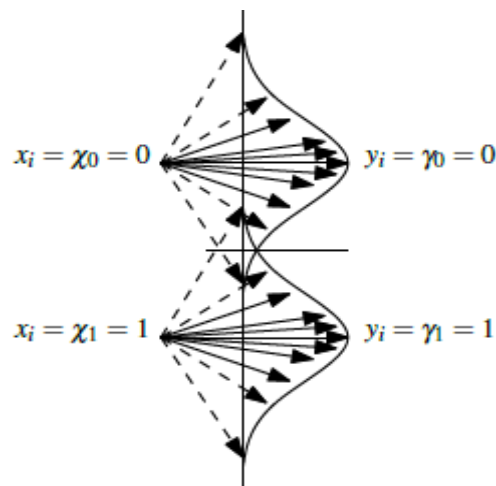
$N_0$  correspond à la densité spectrale du bit.

La fonction RSB est la valeur de la probabilité d'erreur de ce canal pour une modulation BPSK. [4]

Pour un autre type modulation, la probabilité d'erreur binaire est définie comme suit:

$$P_{eb} = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right) \quad (1.7)$$

Pour mieux comprendre, prenons en considération la figure suivante :



**Figure 1.4:** Canal BBAG [4]

### 1.3.2 Support physique de transmission

Il existe deux types de transmission: on parle d'une propagation libre d'un signal quand le signal peut se propager librement suivant toutes les directions et celle est dite guidée dans un support quand le signal est contraint de se déplacer dans un espace limité. [3]

#### 1.3.2.1 Propagation libre

La transmission hertzienne est une transmission libre entre une antenne émettant une onde électromagnétique modulée par le signal utile et une antenne réceptrice. Elle se fait sans support c'est à dire dans l'atmosphère.

Selon les fréquences  $f$  utilisées, et les longueurs d'ondes associées  $\lambda$ , différentes bandes de fréquences relatives à différentes applications (industrielles, médicales et scientifiques) sont définies tels les systèmes cellulaires pour la téléphonie mobile, les faisceaux hertziens pour le radio et télédiffusion entre autre, les réseaux WiFi (*wireless fidelity*) et WiMax (*worldwide interoperability for microwave access*). [3]

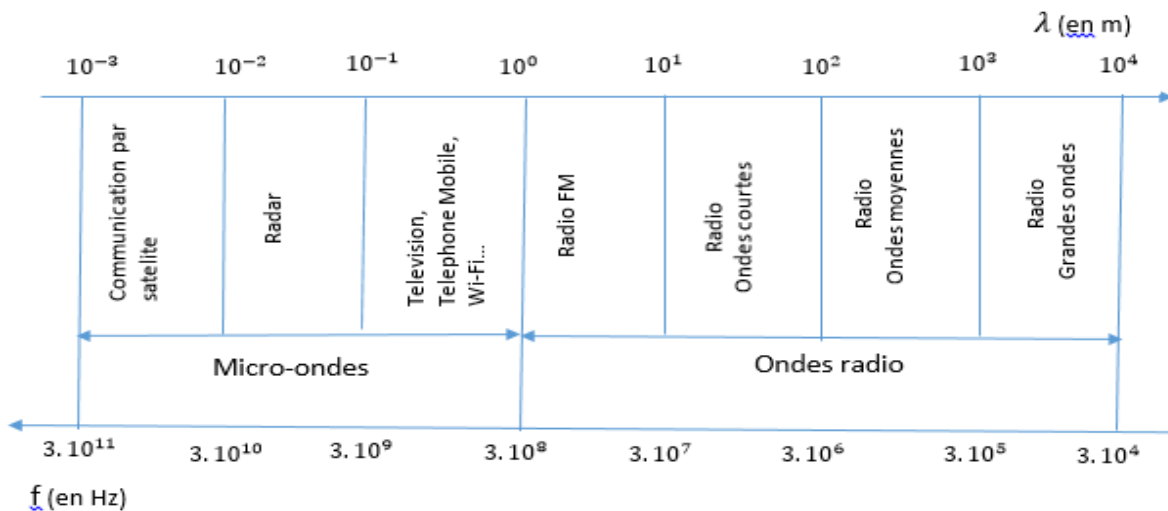


Figure 1.5: Spectre électromagnétique [3]

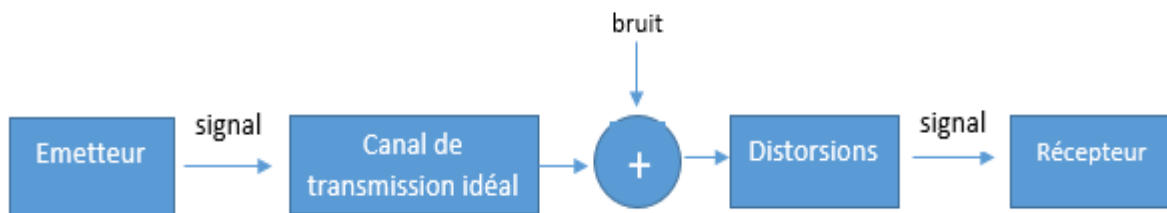
### 1.3.3 Modélisation d'un canal de transmission:

Un canal de transmission idéal (voir figure 1.6) où il n'existe aucun bruit et où la source et la destination sont parfaitement adaptées n'existe pas. [3]



**Figure 1.6 :** Canal idéal [3]

Dans un canal de transmission réel (voir figure 1-7), il y a présence de bruit (dû au mouvement d'électrons dans les semi-conducteurs et aux multi trajets) non négligeable qui a pour effet de déformer le flux binaire et ce en inversant, atténuant, effaçant et déphasant certains bits. L'impact de ce bruit dans une transmission numérique peut être minimisé par différentes techniques de détection et correction d'erreurs. [3]



**Figure 1.7 :** Modélisation d'un canal de transmission [3]

Un signal peut trouver plusieurs perturbations lors de son passage dans un canal de transmission, on peut les représenter par un bruit additif et des distorsions.

La réponse impulsionnelle d'un canal de transmission linéaire est notée  $h(t)$ .

$$s(t) = e(t) * h(t) = \int_{-\infty}^{+\infty} e(\tau) \cdot h(t - \tau) d\tau \quad (1.8)$$

Où l'opérateur (\*) désigne une opération de convolution

Si on procède à une transformée de Fourier de la réponse Impulsionnelle  $h(t)$  on obtient la réponse fréquentielle appelée aussi fonction de transfert du canal de transmission:

$$H(\omega) = e^{[-\alpha(\omega)-j\beta(\omega)]} \quad (1.9)$$

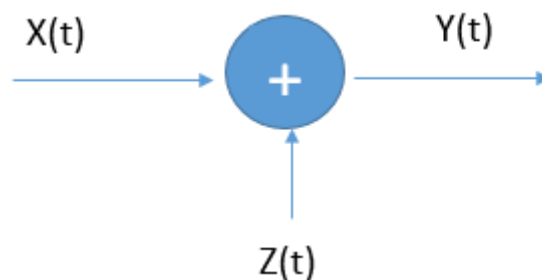
Où :  $\alpha$  est l'atténuation exprimée généralement en Neper (Np).

$\beta$  : est le déphasage en radian.

Une distorsion d'amplitude a lieu lorsque l'atténuation( $\alpha$ ) dépend de temps ( $\alpha(t)$ ) et, celle de la phase lorsque le déphasage ( $\beta$ ) n'est pas proportionnelle au temps.

Les signaux logiques sont très sensibles aux distorsions de phase. Une distorsion de phase pure ne se fait pas sentir par l'oreille humaine dans le cas des signaux sonores, et elle ne cause pas des changements dans la densité spectrale de puissance du signal [3].

La figure 1.8 nous montre à travers d'un schéma simplifié un canal de transmission avec le bruit additif :



**Figure 1.8** : Modèle de canal avec bruit additif [5]

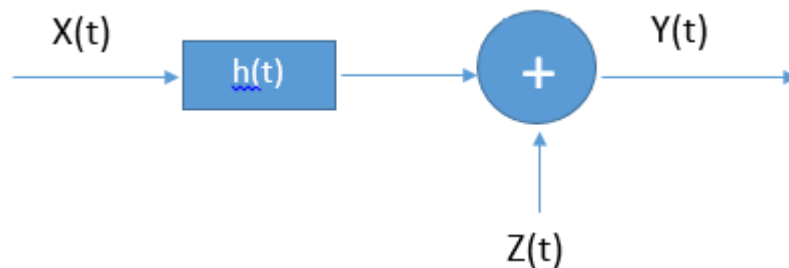
Où :  $X(t)$  représente le signal d'entre

$Y(t)$  représente le signal de sortie

$Z(t)$  représente le bruit additif gaussien

$$Y(t) = X(t) + Z(t) \quad (1.10)$$

Dans plusieurs cas, le modèle de canal de transmission ayant une bande limitée adopte le comportement d'un filtre avec une réponse impulsionnelle  $h(t)$ . Pour bien montrer cela, une petite mise à jour de la figure 1.8 est donnée ci-dessous (figure 1.9):



**Figure 1.9 :** Modèle de canal de transmission à bande passante limitée [5]

$$Y(t) = X(t) * h(t) + Z(t) \quad (1.11)$$

où

$$Y(t) = \int_{-\infty}^{+\infty} X(t - \tau)h(\tau)d\tau + Z(t) \quad (1.12)$$

#### 1.3.4 Bande passante du canal:

Chaque canal de transmission a une bande passante limitée c'est-à-dire:

- Certains signaux se propagent correctement sur le support.
- Certains ne se propagent pas du tout.

La bande passante c'est la bande fréquence des signaux dont la puissance à la sortie, après la transmission sur le support est supérieure à un seuil donné (3 dB). [3]

Par exemple la puissance d'un signal sinusoïdale est au pire divisé par 2.

$$\frac{P_{entrée}}{P_{sortie}} = 2 \Rightarrow 10 \log_{10} \left( \frac{P_{entrée}}{P_{sortie}} \right) = 3dB \quad (1.13)$$

Pour la 5G, la fréquence se répartit en trois bandes:

- La bande des **3,5 GHz (3,4-3,8 GHz)** offrant une largeur de bande suffisante (310 MHz) pour empêcher l'effet de saturation et avoir une bonne longueur d'ondes.
- La bande des **700 MHz (694-790 MHz)** appartenant aux fréquences basses qui ont la plus grande longueur d'ondes et permettant une meilleure couverture même pour les zones rurales puisqu'elles ont une grande portée. La bonne pénétration à l'intérieur des bâtiments en est un autre avantage pourtant elles n'autorisent pas des débits très élevés.
- La bande des **26 GHz (24,25-27,5 GHz)** permettant non seulement d'avoir un très large spectre mais aussi d'avoir un débit comparable à ceux de la fibre. Cette bande est encore simplement l'objet d'une expérimentation. [6]

### 1.3.5 Capacité de canal pour la 5G

La capacité d'un canal est définie par le fait que la quantité d'information qu'un canal peut transporter est limitée.

En cas de la présence de bruit sur le canal, la transmission n'est pas parfaite. Ceci convient alors de déterminer la proportion des symboles qui risquent d'être entachés d'erreurs. Pour chaque milieu de transmission, on peut chiffrer la capacité théorique du canal  $C$  en bits par seconde (bps), c'est-à-dire le nombre maximum de bits par seconde qu'il est possible de transmettre avec une erreur aussi faible que possible, exprimée à partir de la bande passante du canal  $B$  en Hz et du rapport signal- au bruit  $S/N$  à l'intérieur du canal, les deux exprimés en watt(W). [3]

Cette relation est dite celle de Shannon-Hartley et est formulée comme suit

$$C = B \log_2 (1 + S/N) \quad (1.14)$$

### 1.3.6 Modèles des canaux

De nombreux modèles permettent les modélisations des canaux. Pour ce code que nous avons implémenté, on a choisi de se baser sur le canal gaussien et le canal de Rayleigh.

#### 1.3.6.1 Canal Gaussien:

Un canal gaussien est affecté par un bruit additif blanc gaussien. Ainsi, ce modèle de canal convient à stimuler le bruit de fond du canal étudié. Et ce bruit est basé sur les deux suppositions suivantes:

- tous les bits ont même probabilité, quelle que soit leur position.
- les erreurs des différentes positions sont indépendantes.

Le bruit est dit gaussien du fait qu'il a une densité de probabilité gaussienne.

À première vue, ce modèle paraît simple, mais en pratique on rencontre plusieurs difficultés. En effet, on peut avoir des probabilités variables suivant les positions et, de plus, les erreurs ont tendance à être liées entre elles, et non indépendantes (par exemple quand la cause de l'erreur est commune et intervient de façon globale sur le canal). Cependant, comme au niveau de la modélisation il n'est pas possible d'avoir plus d'informations sur le bruit, la modélisation considérée paraît raisonnable. C'est d'ailleurs une excellente modélisation du bruit thermique, perturbation universelle s'il en est. [3]

Dans le cas d'un canal gaussien blanc AWGN (*additive white gaussian noise*), on considère le signal reçu  $y(t)$  suivant :

$$Y(t) = X(t) + N(t) \tag{1.15}$$

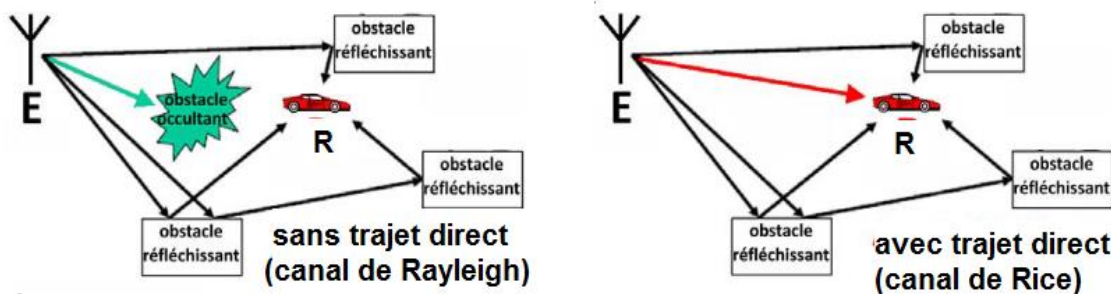
Où  $X(t)$  est le signal envoyé et  $N(t)$  le bruit (purement additif). [3]



### 1.3.6.2 Canal de Rayleigh

Un autre modèle très pratique consiste en un canal de transmission affecté par les évanouissements de Rayleigh. Pour ce modèle de canal, deux types de propagations sont possible:

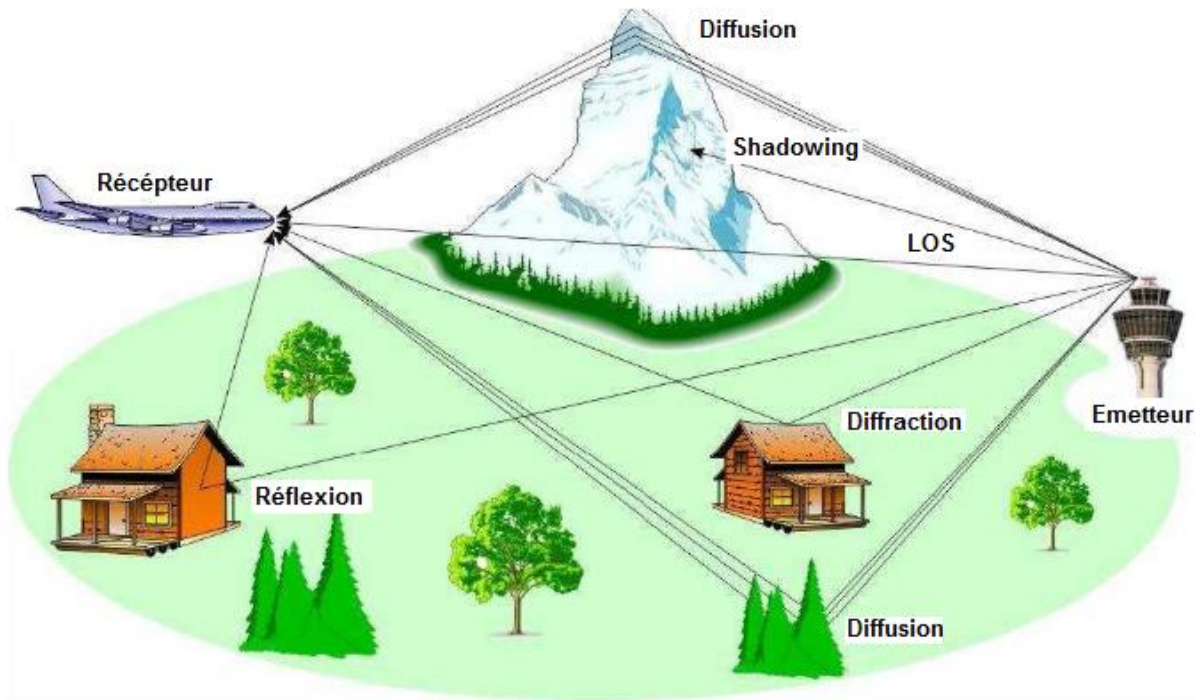
- La propagation NLOS (*non-line of sight*) pour laquelle il n'y a pas de ligne de vue.
- Et la propagation LOS (*line of sight*) qui est marqué par l'existence d'une ligne de vue, appelé aussi canal de Rice. [8]



**Figure 1.10** : Schématisation des canaux de Rayleigh et de Rice [7]

### 1.3.7 Caractéristiques des canaux de propagation:

- **canal multi trajet** : Lors d'une transmission sans fil dans un espace non dégagé, on a le phénomène de multi trajets (multi-path) où des copies du signal émis arrivent en réception affectées par divers phénomènes physiques aléatoires telles : la réflexion (surfaces lisses), la réfraction (milieux translucides), la dispersion (milieux rugueux) et la diffraction (angles, pointes). [9]



**Figure 1.11** : Scénario typique de propagation multi trajet [10]

Le signal émis peut s'écrire :

$$s(t) = \mathcal{R}[u(t)e^{j2\pi f_c t}] \quad (1.16)$$

Avec  $u(t)$  le signal en bande de base et  $f_c$  la fréquence porteuse et  $\mathcal{R}$  désigne la partie réelle.

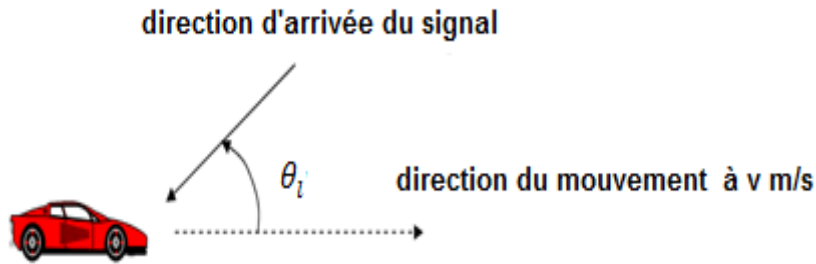
Le signal reçu est une somme de plusieurs signaux. Chacun de ces signaux est atténué, retardé et déphasé. Cette sommation peut être constructive ou destructrice.

$$r(t) = \mathcal{R}\left[\sum_{l=0}^{L(t)-1} \alpha_l(t)u(t - \tau_l(t))e^{j(2\pi f_c(t - \tau_l(t)) + \Phi_{Dl})}\right] \quad (1.17)$$

Avec  $L(t)$  le nombre de trajets.

Chaque trajet a une amplitude  $\alpha_l(t)$ , un retard  $\tau_l(t)$  et une phase égale à  $2\pi f_c(t - \tau_l(t))$ .

$\Phi_{Dl}$  est le déphasage dû à l'effet Doppler. [9]



**Figure 1.12** : Schématisation de l'effet Doppler [9]

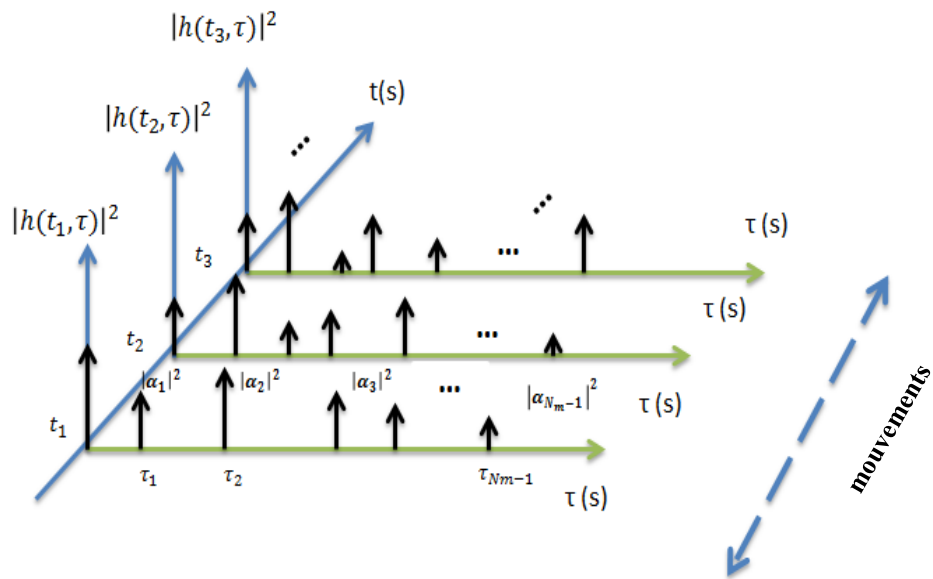
La réponse impulsionnelle du canal est donnée par:

$$h(t, \tau) = \sum_{l=0}^{L(t)-1} \alpha_l(t) e^{-j\Phi_l(t)} \delta(t - \tau_l(t)) \quad (1.18)$$

$$\Phi_l(t) = 2\pi f_c \tau_l(t) - \Phi_{Dl}(t) \quad (1.19)$$

$$\Phi_{Dl} = 2\pi f_{Dl} = 2\pi \frac{v}{\lambda} \cos(\theta_l) \quad (1.20)$$

La réponse impulsionnelle de l'équation 1.18 peut-être schématisée de la manière suivante (figure 1-13). Quand le mobile se déplace aux instants  $t_1, t_2, t_3, \dots$  etc. d'un endroit à un autre, le nombre de trajets représenté en flèches verticales sur l'axe des retards  $\tau$ , ainsi que leurs retards et déphasages respectifs change d'un instant à un autre.

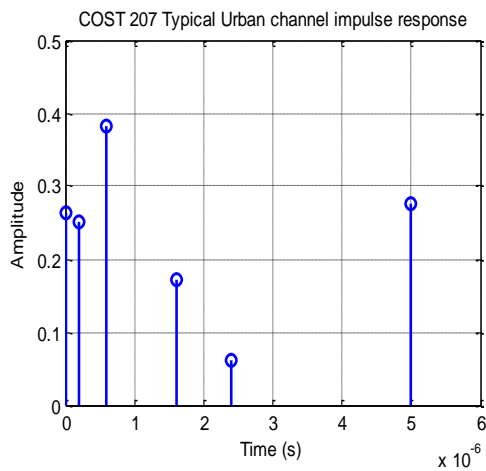


**Figure 1.13** : Schématisation d'une réponse impulsionnelle dans le cas d'un canal variable du temps [9]

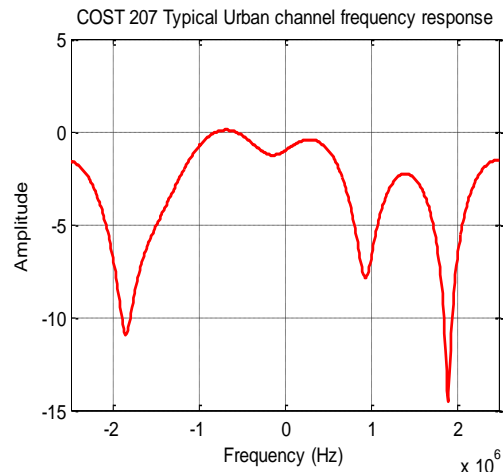
La transformée de Fourier de cette réponse impulsionnelle est donnée par :

$$H(f, t) = \int_{-\infty}^{+\infty} h(t, \tau) e^{-j2\pi f\tau} d\tau \quad (1.21)$$

Ci-dessous (figure 1-14) est représentée, à un instant donné, les réponses impulsionnelle temporelle et fréquentielle du modèle réel COST 207 (*cooperation in the field of scientific and technical research*) concerne la norme GSM (*global system for mobile communications*) à 900 MHz. [9]



(a)



(b)

**Figure 1.14** : Réponses temporelle (a) et fréquentielle (b) du modèle de canal COST 207 de la norme GSM-900 [9]

La présence de réflecteurs dans le chemin de propagation entre un émetteur et un récepteur crée de multiples chemins qu'un signal émis peut traverser. En conséquence, le récepteur voit la superposition de multiples copies du signal émis, chacune parcourant un chemin différent. Chaque copie de signal connaîtra des différences d'atténuation, de retard et de déphasage lors du déplacement de la source au récepteur. Cela peut entraîner des interférences constructives ou destructives, amplifiant ou atténuant la puissance du signal vu au niveau du récepteur. Une forte interférence destructive est souvent appelée un évanouissement profond (*deep fading*) et peut entraîner une défaillance temporaire de la communication en raison d'une forte baisse du rapport signal sur bruit du canal.

- Principales sources de bruit dans un canal multi trajet sont :
- **Affaiblissement de parcours** (*path loss*) : c'est la diminution de la puissance du signal due à l'éloignement. qui modélise la diminution de l'amplitude du signal avec l'éloignement.
- **Effet de masque** (*shadowing*) : c'est un phénomène plus local, aléatoire dû aux atténuations successives. C'est la variation de la puissance due à de gros obstacles.

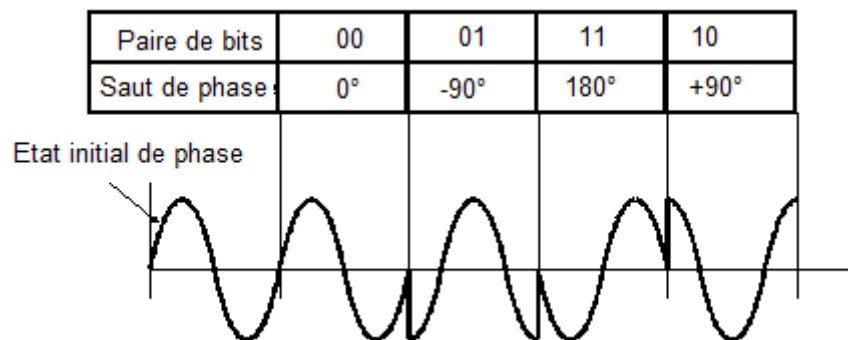
- **Evanouissement** (*fast fading ou small scale*): ce sont des variations rapides de l'amplitude du signal (addition constructive ou destructive des ondes). Il se produit dès que le mobile se déplace d'une faible distance (*small scale*). Il y a nécessité d'une étude statistique. Ces variations sont le plus souvent destructrices pour le signal (problème majeur en télécom). [9]

### 1.3.8 Modulation

Comme il a été déjà mentionné, le message doit être adapté au milieu de transmission, cette procédure est appelée modulation.

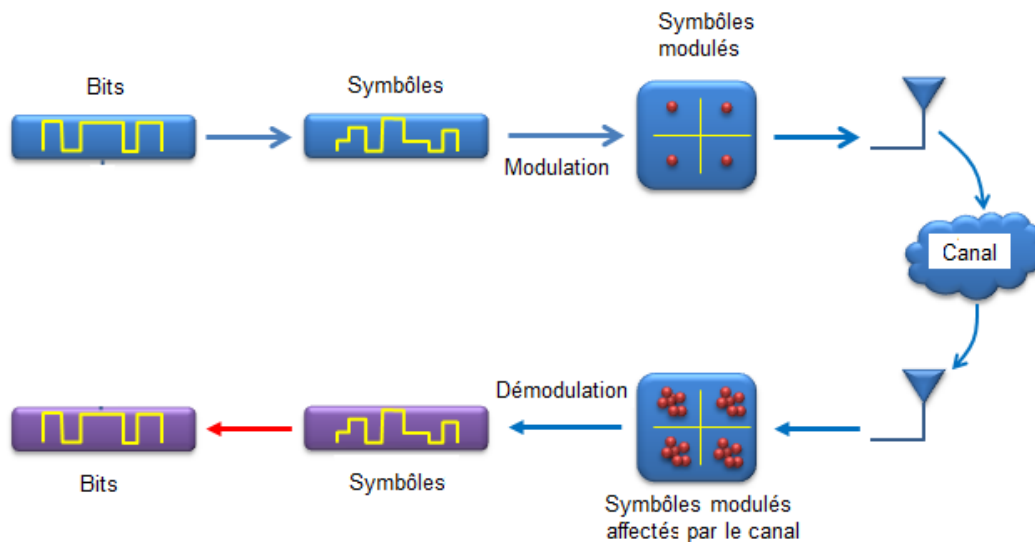
Dans notre cas on s'intéresse tout simplement aux modulations numériques, où le message est binaire, qui se basent sur l'association des bits récupérés dans un flux binaire à des symboles. On appelle cette association mappage binaire (*mapping*). [4]

Dans un mapping plusieurs bits peuvent être associé à un symbole (2, 3, 4 ...etc.), l'ensemble de ces valeurs est appelé constellation. Un exemple de mapping à 2 bits pour la modulation QPSK (*quadrature phase shift keying*) est illustré en figure 1.15.



**Figure 1.15** : Mapping à 2 bits pour la modulation QPSK [11]

A la sortie du canal on peut récupérer une estimation du symbole en utilisant la fonction inverse ou démodulation numérique. [4]



**Figure 1.16** : Modulation et Démodulation numérique [12]

L'adaptation entre le signal à la bande transposée est faite à base des techniques de modulations. On a 3 techniques possibles :

- Modulation d'amplitude ;
- Modulation de fréquence ;
- Modulation de phase. [1]

Plus souvent, on utilise la modulation d'amplitude et la modulation de phase, soit séparément ou ensemble en établissant une dépendance entre eux. Le signal à être émis est sous la forme complexe et est aussi divisé en deux parties : la partie réel qui correspond au composant I (In-phase) et la partie imaginaire que correspond au composant Q (Quadrature).

Quand chaque symbole émis prendre  $M$  valeurs possibles, on peut dire que cette modulation est appelée M-aire, où  $M = 2^k$  et  $k$  le nombre de bits d'un vecteur que forme le symbole.

Les symboles  $a_k$  (complexes) sont formés par le modulateur en fonction des bits à son entrée, après le codage canal. Au niveau de la réception on a l'estimation des

valeurs des bits correspondant aux symboles  $a_k$  fait par le démodulateur, en ayant une estimation des  $a_k$  à son entrée.

On a deux points à avoir en compte pendant la conception d'un modulateur :

- type de modulation ou la constellation dans le plan complexe
- étiquettes des points de la constellation (mapping).

Une limitation en puissance ou en bande passante peut déterminer quel type de modulation peut être fait dans un système. On place le démodulateur derrière ou il remplace le module de prise de décision. Pour démoduler on doit:

- définir les régions de décision sur la plane complexe
- accorder au vecteur des bits à la valeur correspondant. [1]

### **1.3.8.1 Transmission en bande de base**

Cette transmission consiste à transmettre directement le signal sur le support de transmission sans transposition de fréquence, en utilisant, par exemple, un modem à bande de base pour les signaux numériques qui permet de l'application directe dans le support.

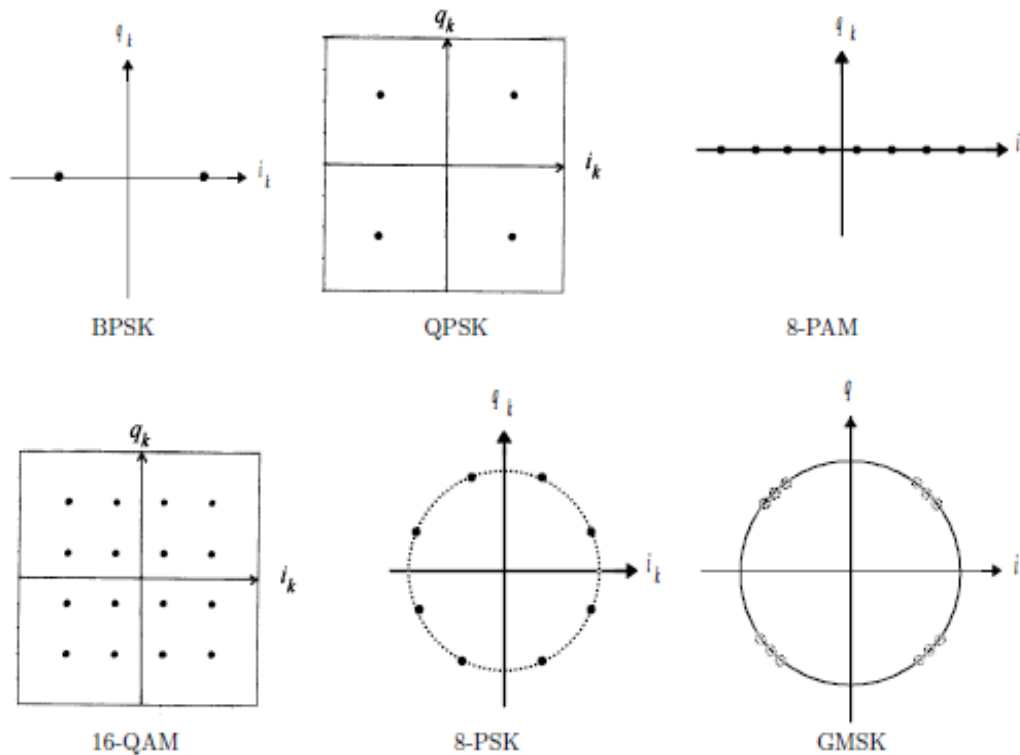
L'application de ce type de transmission présente ces inconvénients :

- Il est sensible aux parasites (bruits en  $1/f$ ) ;
- Dans un câble coaxial ou fibre optique ce type de transmission a des coûts élevés ;
- C'est impossible de partager le canal avec plusieurs sources au même temps ;
- Pour les signaux qui ont une basse fréquence c'est impossible de le transmettre à l'air. (exemple : le son dont les fréquences vont de 20 à 20 kHz soit des longueurs d'onde de 15 à 15000 km !). [3]

### **1.3.8.2 Types de modulations numériques**

On a plusieurs types de modulations numériques et maintenant on va parler un peu sur quelques-unes qui sont plus souvent utilisées dans cette opération.





**Figure 1.17 :** Exemples des modulations numériques. [1]

- **BPSK** (*binary phase shift keying*)

C'est la modulation la plus simple. Elle est dite bipolaire car elle peut prendre deux valeurs ( $M = 2$ ) et sa constellation nous montre que chacun d'eux prend un pôle.

On a  $a_k = \{\pm 1\}$  et on a un seul bit dans le vecteur ( $k = 1$ ). Chaque symbole est complètement réel, et

$$s(t) = \sum_{k \in \mathbb{Z}} a_k g(t - kT_s) \cos(2\pi f_0 t) \quad (1.22)$$

- **QPSK** ou **4-QAM** (*quadrature phase shift keying* ou *4-level quadrature amplitude modulation*)

Ici, on a  $M = 4$  et les symboles  $a_k$  prennent les valeurs  $\{1 + j, 1 - j, -1 + j, -1 - j\}$ . Chaque symbole est généré à partir d'un vecteur de 2 bits ( $k = 2$ ) et

$$s(t) = \sum_{k \in \mathbb{Z}} i_k g(t - kT_s) \cos(2\pi f_0 t) - \sum_{k \in \mathbb{Z}} q_k g(t - kT_s) \sin(2\pi f_0 t) \quad (1.23)$$

Où  $i_k = \text{Re}\{a_k\}$ , qui représente la partie réel du symbole  $a_k$  et  $q_k = \text{Im}\{a_k\}$  qui représente la partie imaginaire du symbole  $a_k$ .

Les symboles  $a_k$  sont réels et peuvent prendre les valeurs  $\{\pm 1, \pm 2, \dots \pm M/2\}$ . On doit utiliser cette modulation quand  $M$  est petit car s'il est grand il va dépenser trop de puissance émise. Le BPSK est le M-PAM avec  $M$  égale à deux et avec des symboles réels.

- **M –QAM** (*multi-level quadrature amplitude modulation*)

C'est l'une des modulations la plus utilisée dans les systèmes avec des limitations en bande passante. On doit faire attention sur la valeur de  $M$ , plus souvent on utilise  $M = 4; 16; 64; 256$ , c.à.d, le vecteur du signal peut avoir au maximum huit bits ( $k = 8$ ), car pour les constellations carrées, les voies I et Q peuvent être indépendantes. Le symbole  $a_k = n + jm$ , où  $n, m \in \mathbb{Z}, n, m \leq \sqrt{M}$ .

- **M-PSK** (*multi-level pulse shift keying modulation*)

Cette modulation est très utilisée dans les systèmes avec des limitations en puissance émise car, normalement, la puissance émise par symbole est constante et égale à 1. La BPSK et la QPSK en sont des cas particuliers. Le symbole  $a_k = n + jm$  où l'amplitude de  $a_k$  est égale à 1, ( $\|a_k\| = \sqrt{n^2 + m^2} = 1$ ). Ces sont situés sur un cercle unitaire autour du zéro. C'est rare de trouver plus que 8-PSK car la probabilité d'erreur par symbole augmente selon l'augmentation des nombres des points.

- **GMSK** (*gaussian minimum shift keying*)

C'est une modulation utilisée dans le système GSM (*global system for mobile communications*). Elle présente une amplitude constante et phase linéaire. On a

$$s(t) = \cos(\pi f_0 t + 2\pi \int_{-\infty}^t \Delta f(\theta) d\theta) \quad (1.24)$$

Avec :

$$\Delta f(t) = \frac{1}{T_s} \sum_k a_k h(t - kT_s) \quad (1.25)$$

Sachant que le filtre  $h(t)$  est le filtre passe-bas gaussien alors :

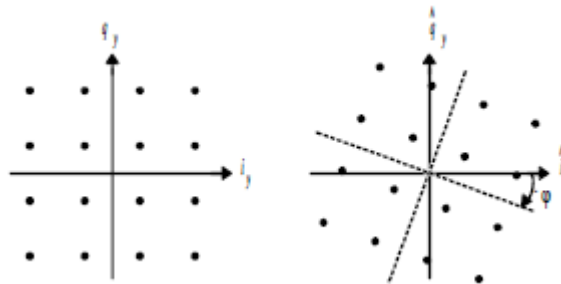
$$h(t) = b \sqrt{\frac{2\pi}{\ln 2}} e^{-\frac{2\pi^2 b^2 t^2}{\ln 2}} \quad (1.26)$$

Où

$$h(f) = e^{-\frac{f^2}{2b^2} \ln 2} \quad (1.27)$$

Où  $b$  est la fréquence de coupure à -3dB du filtre. [16]

Considérons une rotation de la constellation comme on peut voir dans la figure suivant.



**Figure 1.18** : Rotation de constellation. [1]

On peut avoir un mélange des composants I et Q dans la détection car une erreur de phase non-nulle entraîne une rotation de l'angle. On peut dire que pour la reconstruction de ces symboles, on doit utiliser une erreur de phase nulle (démodulation cohérente), mais ceci est seulement possible dans le cas idéal.

On peut observer que chaque constellation à un angle et on ne sait pas lequel est celui de la constellation initiale ou de la constellation tournée. On peut essayer d'éviter des erreurs en utilisant le codage différentiel dans laquelle les symboles  $a_k$  correspondent seulement à la différence entre deux vecteurs voisins. Cette opération est optimale vis à vis de la probabilité d'erreur si on utilise un filtre adapté à la réception.

On fait l'étiquetage car il nous permet de minimiser la probabilité d'erreur par bit pour une probabilité d'erreur par symbole donné et ce en choisissant les étiquettes des points de constellation de telle sorte qu'une erreur de démodulation d'un symbole introduit le minimum d'erreurs dans l'estimation des valeurs de bits.

Par exemple : L'utilisation du code Gray pour l'étiquetage dans un mapping d'une modulation QPSK.

Nous avons quatre (04) points :  $1 + j$ ,  $1 - j$ ,  $-1 - j$  et  $-1 + j$  de la constellation auxquels nous devons faire la correspondance avec quatre (04) vecteurs des étiquettes 00, 01, 10 et 11. La distribution de ces étiquettes se fait en imposant que la différence dans les étiquettes de deux symboles voisins est d'un seul bit.

Par exemple:  $1 + j \rightarrow 00$ ;  $1 - j \rightarrow 01$ ;  $-1 - j \rightarrow 11$ ; et  $-1 + j \rightarrow 10$ . [1]

#### **1.4 Conclusion**

Ce chapitre nous avons essayé de rassembler tout l'essentiel d'une communication numérique. Des choix peuvent être possibles en ce qui concerne les types des canaux eux-mêmes, leurs caractéristiques et ses leurs modèles de propagation ainsi que les types de modulations, qui jouent tous un rôle important pour la bonne marche de la communication. Une fois cette étude faite, nous pouvons aborder celle qui nous intéresse dans ce mémoire et qui concerne le codage décodage canal à l'aide des codes polaires.

# Chapitre 2 : Les codes polaires

---

## 2.1 Introduction

Les codes polaires font partie de l'ensemble des codes en bloc. L'un des types de code le plus important de ce même ensemble est le code correcteur d'erreurs. Les codes correcteurs en bloc les plus connues faisant partie de cet ensemble sont :

- **les codes BCH** (Bose–Chaudhuri–Hocquenghem): sont cycliques et se basent sur un polynôme défini par ces racines (zéros). Sa qualité de décodage est directement proportionnelle à la distance de Hamming. Ils sont appliqués dans la communication satellitaire.
- **les codes de Reed-Solomon** : ce sont des codes correcteurs d'erreur et ont des polynômes générateurs comme base. Leur point fort est leur grande capacité de détection et de correction de plusieurs erreurs de symboles. Très utilisés dans le CD (*compact disk*), DVD (*digital versatile disc* ou *digital video disc*), BRD (*blu-ray disk*)...etc.
- **les codes Produits** : est basé sur l'union de deux codes, un pour les lignes et l'autre pour les colonnes, les deux ont une grande capacité de correction. Il peut être aussi des symboles non binaires.
- **les codes LDPC** (*low parity density check*) : ils exigent une forte complexité calculatoire et à cause de laquelle elle reste inutilisable jusqu'à quelques dizaines d'année après sa création. Son décodage est fait par l'algorithme Propagation de Croyances BP (*belief-propagation*).
- **Les codes polaires** : Dans notre travail, nous allons nous focaliser sur les codes polaires qui sont assez récents (E. Arıkan, 2009) et théoriquement, la capacité du

canal de transmission peut atteindre la limite de Shannon pour un code à taille infinie, et ce dernier peut être démontré par la réalisation de certains calculs. Ces codes sont aussi importants à cause de leurs preuves mathématiques, ils sont les seuls possédant cette caractéristique. [4]

Cette limite est atteinte lorsque le canal de transmission de type BSM (*binary symmetric memoryless*) ou canal binaire, symétrique, sans mémoire et le récepteur utilisé au niveau de son décodeur utilise l'algorithme SCD (*successive cancellation decoding*) ou décodage par annulations successives.

Les autres caractéristiques qui laissent les codes polaires encore plus intéressants sont leur faible complexité de codage et de décodage à l'aide du SCD.

La polarisation du canal binaire sert comme base des codes polaires. Elle consiste de diviser les canaux binaires, après un long processus en arborescence, en deux groupes de canaux individuels, l'un considéré comme étant parfait (sans bruit) et l'autre très bruité (imparfait). Pour se rapprocher de la limite de Shannon, il faut augmenter la longueur du mot de code, notée  $N$ .  $N = 2^n$  avec  $n$  est un entier naturel pour quelques canaux sans bruit et utiliser ces mêmes canaux pour transmettre les informations importantes. Les canaux bruités sont utilisés pour envoyer des bits connus par l'émetteur et le récepteur, ces bits sont généralement des 0 binaires appelé bits gelés (*frozen bits*).

La complexité d'un code polaire est  $O(n \log_2 n)$ , incluant les opérations d'encodage et de décodage, où  $n$  représente la longueur du code. Ce type de code devient très intéressant lorsque  $n$  est grand ( $n > 2^{20}$ ), mais si  $n$  est faible la performance de ce code devient plus faible que celle des autres codes comme les codes LDPC. Enfin, il est possible d'améliorer les performances d'un code polaire en l'associant avec des autres techniques existantes, comme par exemple le décodage par l'algorithme d'annulation successive par liste (*successive cancellation list decoding*). [13]

## 2.2 Construction des codes polaires

Ce type de code fait partie de l'ensemble des codes en bloc linéaires, et présente  $K$  bits d'informations à l'intérieur du mot de code de taille  $N = 2^n$ .

Un code polaire utilise, dans la partie codage en émission, une matrice génératrice qui est une sous matrice de la  $n^{\text{ème}}$  puissance de Kronecker de :

$$k = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (2.1)$$

$$F = k^{\otimes n} = \begin{bmatrix} k^{\otimes n-1} & 0_{n-1} \\ k^{\otimes n-1} & k^{\otimes n-1} \end{bmatrix} \quad (2.2)$$

Et est formé par de  $k$  lignes.

En supposant, en réception, un décodage de type SCD, ces  $k$  lignes choisies permettent de polariser le canal, en termes de probabilités d'erreur des bits du message. [4]

Soient les ensembles  $B^n = \{0,1\}^n$  et une matrice  $G$  particulière qui appartient à l'ensemble  $B^K * B^N$ . L'encodage d'un code en bloc linéaire est une application linéaire injective de  $B^K$  vers  $B^N$ , où un élément  $b \in B^K$  fait la correspondance avec un élément  $x \in B^N$  de telle façon que :

$$x = b.G. \quad (2.3)$$

L'association de deux matrices  $E \in (B^K * B^N)$  et  $F^{\otimes n} \in (B^N)^2$ , permet d'avoir la matrice génératrice d'un code polaire qui peut être écrit :

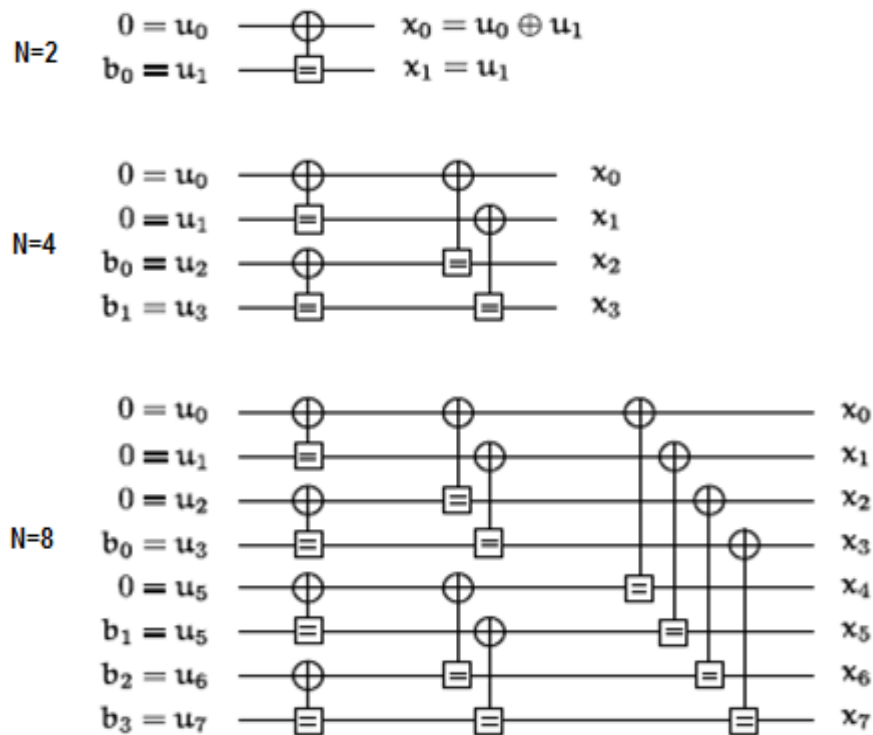
$$G = EF^{\otimes n}. \quad (2.4)$$

Les bits gelés dont la valeur est 0, sont des produits de la multiplication de la matrice  $E$  avec la séquence d'information  $b$ . Le résultat d'opération  $E \times b$  est formé par :

- $k$  bits d'information  $b_i \in b$
- $(N - k)$  bits gelés.

Le pouvoir de correction du code est déterminé par la position de tous ces bits en prenant en compte la polarisation du canal.

Pour représenter graphiquement l'encodage de ces codes polaires, les graphes de factorisation FG (*factor graph*) sont utilisés, dont les opérations effectués dans ce processus sont basés par ou-exclusif, avec le symbole  $\oplus$ . [14]



**Figure 2.1:** Représentation à l'aide de FG de l'encodage de codes polaires. [14]

Un code polaire systématique est un code polaire dont le mot de code contient les bits de la séquence d'information et ce après avoir fait quelques modifications sur le processus d'encodage. Le paramètre qui doit être modifié est la matrice génératrice pour devenir :

$$G_{\text{sys}} = EF^{\otimes n}EF^{\otimes n} \quad (2.5)$$

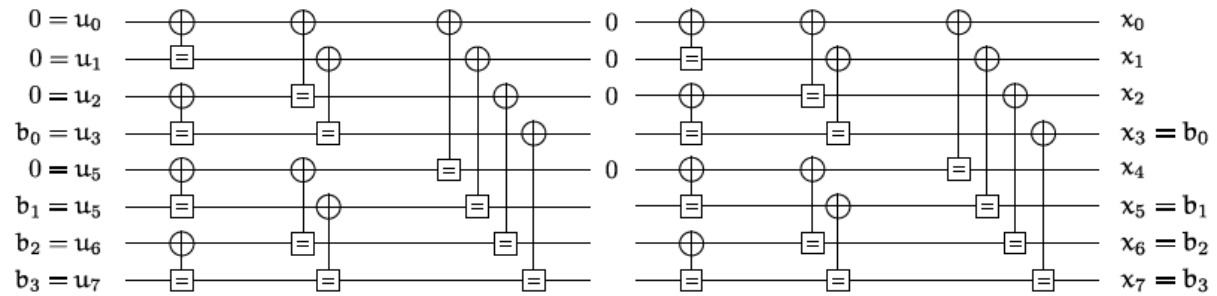
De cette façon après son usage, les bits d'information  $b$  se trouvent dans le mot de code  $x$ , car si :



$$x = bG_{\text{sys}} \quad (2.6)$$

Alors :

$$b = E^{-1}x. \quad (2.7)$$



**Figure 2.2** : Encodage systématique. [14]

### 2.2.1 Polarisation du canal

La caractéristique essentielle des codes polaires est la polarisation du canal. Soit  $N$  canaux de transmission dont chacun d'eux transmet un bit, dans le parcours entre l'encodeur polaire et le décodeur polaire.

La polarisation est un phénomène de division des  $N$  canaux en deux groupes, dans lequel, un groupe de canaux considéré comme parfait (très fiable) si la probabilité d'erreur de chaque canal est très petite et dans le cas contraire, l'autre groupe de canaux est fortement bruité avec une probabilité d'erreur important. [14]

Soit  $W(W: X \rightarrow Y)$  un canal BSM discret. Selon Arikan (2009), la capacité symétrique  $I(W)$  de ce canal est égale à la capacité de Shannon. Cette capacité symétrique  $I(W)$  donne la vitesse la plus élevée sur le canal dans une transmission de données fiable en utilisant des entrées avec une probabilité égale. Elle est donnée par :

$$I(W) = \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y | x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \quad (2.8)$$

L'ensemble de canaux polarisés  $W_N^{(i)}$ , où  $0 \leq i \leq N - 1$ , est formé par la conversion de  $n$  copies indépendantes du canal  $W$  original.

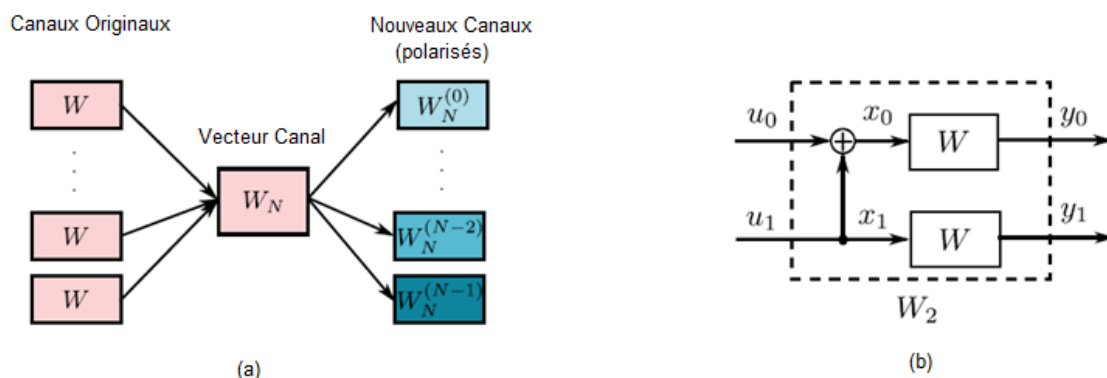
Comme  $N$  tend vers l'infini, chacun des canaux qui font partie de cet ensemble devient soit bruités ou sans bruit et la capacité symétrique de ces canaux  $I(W_N^{(i)})$  pour  $0 \leq i \leq N - 1$  tend vers 0 ou 1.

Ces canaux sans bruit auront la vitesse  $I(W)$  pendant sa transmission et les canaux bruités auront la vitesse  $1 - I(W)$ .

C'est seulement les canaux dits parfaits qui peuvent atteindre la capacité symétrique du canal. Ils sont utiles pour envoyer les bits d'information et les canaux bruités reçoivent des valeurs déjà fixes et connues de l'émetteur et le récepteur (*frozen bits*). [15]

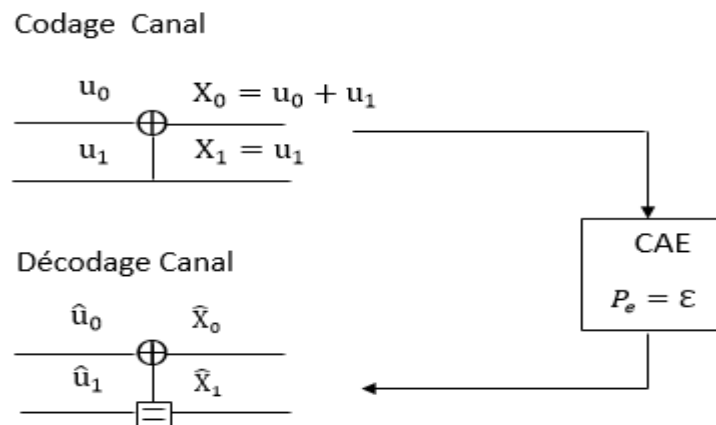
La combinaison et la division des canaux sont deux parties importantes dans le processus de la polarisation de canaux. Les  $n$  canaux, du même type que  $W$  s'associent pour former un seul vecteur canal  $W_N$ , c'est la combinaison. Mais ils sont séparés selon sa capacité, c'est la division.

Pour mieux comprendre, prenons en considération la figure suivante :



**Figure 2.3** : Polarisation du canal. Adaptée d'Arıkan (2009) : (a) polarisation et scission des canaux, (b) opération transformation de polarisation de base [15]

Ce phénomène de polarisation peut être facilement compréhensible dans le cas d'un canal binaire à effacement BEC (*binary erasure channel*) avec un code de taille  $N = 2$ . Ici, la probabilité d'effacement est notée  $\varepsilon$  et chacun des bits de données du message voit un canal équivalent plus ou moins fiable que le canal de transmission. Considérons l'exemple pris dans [4] où CAE désigne canal à effacement,  $P_e$  la probabilité d'effacement et  $0 < \varepsilon < 1$  :



**Figure 2.4** : Codage canal et décodage canal d'un code polaire de taille  $N = 2$ . [4]

Soient  $\hat{X}_0$  et  $\hat{X}_1$  sont les probabilités à la réception:

$$P_r(\hat{X}_0 \neq X_0) = P_r(\hat{X}_1 \neq X_1) = \varepsilon \quad (2.9)$$

Où  $\varepsilon$  représente la probabilité d'effacement du canal.

Il est possible d'avoir la valeur de  $\hat{u}_0$  seulement si  $\hat{X}_0$  et  $\hat{X}_1$  sont reçus correctement. Il vient alors :

$$P_r(\hat{u}_0 = u_0) = P_r(\hat{X}_0 = X_0 \cap \hat{X}_1 = X_1) \quad (2.10)$$

Comme  $\hat{X}_0$  et  $\hat{X}_1$  sont des variables indépendantes à cause du type de canal utilisé, donc :

$$P_r(\hat{u}_0 = u_0) = P_r(\hat{X}_0 = X_0) \times P_r(\hat{X}_1 = X_1) = (1 - \varepsilon) \times (1 - \varepsilon) = (1 - \varepsilon)^2 \quad (2.11)$$

Alors :

$$P_r(u_0) = 1 - (1 - \varepsilon)^2 = 1 - (1 - \varepsilon - \varepsilon + \varepsilon^2) = 2\varepsilon - \varepsilon^2 \quad (2.12)$$

Il est important de noter que la probabilité d'effacement de  $u_0$  est supérieure à celle du canal, d'après ce raisonnement :

$$1 > \varepsilon \Leftrightarrow \varepsilon > \varepsilon^2 \Leftrightarrow -\varepsilon < -\varepsilon^2 \Leftrightarrow \varepsilon < 2\varepsilon - \varepsilon^2 \Leftrightarrow P_e < P_e(\hat{u}_0) \quad (2.13)$$

Ici le signal  $u_0$  a une forte possibilité d'être dégradé pendant sa transmission, puisque la transmission n'est pas très fiable.

Si  $\hat{u}_0$  est la valeur originelle  $u_0$ , quel est la probabilité d'avoir  $\hat{u}_1 = u_1$ ?

En sachant que  $u_1$  est présent en  $X_0$  et  $X_1$ , nous avons deux possibilités:

- Si le  $\hat{X}_0$  est bien reçu, alors:  $\hat{u}_0 + \hat{X}_0 = u_1$
- Si le  $\hat{X}_1$  est bien reçu, alors:  $u_1 = X_1 \Rightarrow \hat{u}_1 = \hat{X}_1$

D'après ces deux cas, pour avoir le bon  $\hat{u}_1$  ( $\hat{u}_1 = u_1$ ), il faut que le  $\hat{X}_1$  ou le  $\hat{X}_0$  soit bien reçu. Nous pouvons écrire :

$$P_r(\hat{u}_1 = u_1) = P_r(\hat{X}_0) + P_r(\hat{X}_1) - P_r(\hat{X}_0 \cap \hat{X}_1) = (1 - \varepsilon) + (1 - \varepsilon) - (1 - \varepsilon)^2 \quad (2.14)$$

Comme les variables sont indépendantes, alors :

$$P_r(\hat{u}_1 = u_1) = 1 - \varepsilon^2 \quad (2.15)$$

En déduisant que :

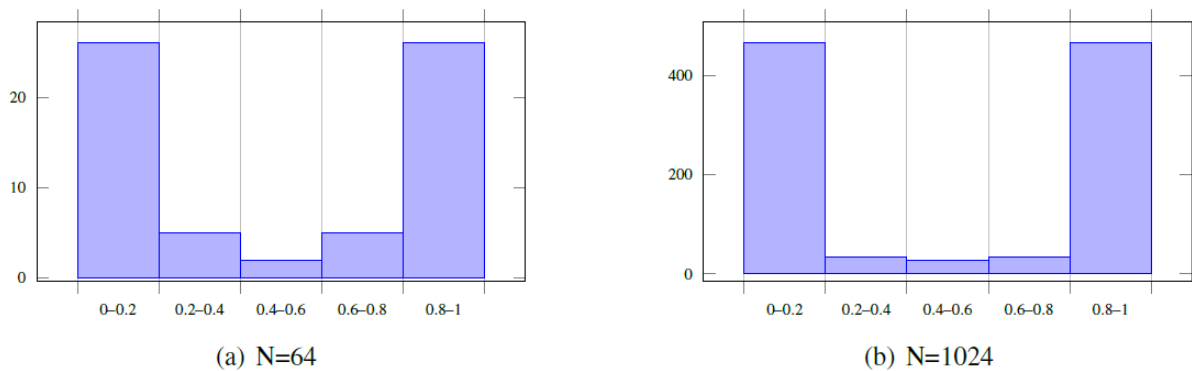
$$P_e(\hat{u}_1) = 1 - P_r(\hat{u}_1 = u_1) = 1 - (1 - \varepsilon^2) = \varepsilon^2 \quad (2.16)$$

Ce qui nous permet de dire que cette probabilité d'effacement est inférieure à celle du canal car :

$$0 < \varepsilon \leq 1 \Leftrightarrow \varepsilon^2 < \varepsilon \quad (2.17)$$

Vue la formation de deux cas présents d'après le passage dans ce canal, dont l'une a la probabilité d'effacement très élevée et l'autre a une probabilité d'effacement faible par rapport à celle du canal. C'est la polarisation du canal.

Visualisons ce phénomène dans la figure suivante :



**Figure 2.5** : Répartition du nombre de bits suivant la probabilité d'effacement sur un canal à effacement pour différentes tailles de code, 64 et 1024. [4]

La figure ci-dessus (figure 2.5) montre les histogrammes qui représentent la distribution des  $P_e(u_1)$  pour les bits de deux codes à taille  $N = 64$  et  $N = 1024$ .

Pour  $N = 64$ , le nombre de bits qui sont polarisés (vers le deux sens) est plus faible que celui du  $N = 1024$ . Ce qui confirme que le code polaire fonctionne mieux pour une taille de code qui tend vers l'infinie. Pour rendre le canal fiable, les indices de bits avec une  $P_e$  élevée sont prises comme les indices des bits gelés. Pour un canal de type BEC, la conception d'un code polaire de rendement  $R$  peut être fait en gelant  $(1 - R)N$  bits. Il est important de noter que  $P_e(u_1)$  dépend des caractéristiques du canal. Cela

signifie, qu'un code polaire conçue pour une valeur de  $\epsilon$ , ne sera plus optimal si les caractéristiques du canal changent.

Arikan a fait des efforts de proposer la construction des codes polaires qui peuvent être utilisées dans les canaux à effacement et les canaux binaires symétriques.

D'autres travaux tels que ceux de Vangala et al. (2015) et Korada e Sasogeu (2009) ont cherché à généraliser le principe de polarisation aux canaux binaires symétriques.

L'un des cas qui est un peu plus compliqué est celle du canal gaussienne, expliqué par le fait d'utiliser les densités de probabilités au lieu des probabilités. Ce qui demande une complexité de calcul très élevé. Le travail de Tal & Vardy (2013) a essayé de proposer une méthode d'estimation rapide de  $P_e(u_i)$  pour ce type de canal.  
[4]

## 2.2.2 Conception des codes polaires

Pour construire un code polaire, il faut faire attention au type de canal binaire à utiliser ainsi qu'aux conditions qu'il présente. Selon Arikan (2009), les canaux choisis doivent être silencieux, avec  $K$  les canaux binaires avec la plus faible borne Bhattcharayya  $Z_N^{(i)}$  qui est dépendante de la probabilité d'erreur de décision  $P_e^{(i)}$ .

Le paramètre Bhattcharayya se base sur la probabilité d'erreur de décision de vraisemblance maximale ML (*maximum likelihood*). Ce paramètre est la limite supérieure de cette probabilité et il prend aussi en considération une mesure des performances d'erreur du canal. Ce paramètre prend des valeurs dans l'intervalle entre 0 et 1 et est calculé récursivement. Ce type de construction présente une complexité temporelle de  $O(N)$ .

Ce paramètre a comme formule :

$$Z(W) = \sum_{y \in Y} \sqrt{W(y|0)W(y|1)} \quad (2.18)$$

Quelques problèmes se présentent à cause de la récursivité de cette construction puisque l'alphabet de sortie de ces canaux binaires s'accroît de façon exponentielle

avec la longueur du bloc de telle manière qu'elle rende difficile la tâche de faire une estimation exacte des canaux binaires en pratique. Beaucoup d'autres travaux ont été réalisés dans le but d'améliorer la construction de code polaire par l'approximation des canaux binaires.

Selon Arikan (2009), pour améliorer et rendre l'estimation des canaux possible aux autres types de canaux, pas seulement le BEC et le BSC (*binary symmetric channel*), une estimation Monte-Carlo de ces canaux binaires a été proposé. En contrepartie, sa complexité devient plus grande  $O(MN \log_2 N)$ , où  $M$  est le nombre d'itérations de Monte-Carlo qui essaye de délimiter la précision de construction (Vangala et al 2015). Dans la construction Bhattcharayya, la détermination des canaux binaires silencieux est basée sur la probabilité d'erreur de décision, mais dans l'estimation Monte-Carlo, le même travail est fait mais en prenant en considération le taux d'erreur sur les bits.

Une autre construction avec un type d'estimation basée sur les approximations gaussiennes peut se faire, proposée par Trifonov (2012). La construction de ces codes donne presque la même performance que celle de Monte-Carlo proposé par Arikan (2009) mais avec une complexité de  $O(N)$  (Vangalla et al (2015)).

La construction des codes polaires Tal & Vardy (2013) est considérée comme la meilleure. Elle se base sur la probabilité d'erreur de chaque canal binaire en effectuant des opérations de qualification pour donner les bornes inférieures et supérieures de ce même probabilité. Elle a pour complexité  $O(N_\mu^2 \log_2 M)$ , où  $\mu$  représente le nombre des symboles de quantifications et après celle de Monte-Carlo, elle a la plus grande complexité. [15]

## 2.3 Processus de codage

Le code polaire est un type de code en bloc, ce qui fait qu'il a une matrice génératrice  $G$  avec la dimension  $(K \times N)$  permettant l'union entre les mots de codes et les codes polaires. Elle peut être obtenue à partir de la matrice  $F$  (équation (2.2)), en faisant une suppression de  $N - K$  lignes de cette même matrice.

Le processus de codage équivalent consiste ensuite à multiplier un vecteur de taille  $K$  par cette matrice  $G$ .

Un processus de codage alternatif consiste à construire un vecteur, noté U, contenant les K bits d'information et N – K bits gelés fixés à 0. Cela c'est le processus de codage alternatif. Dans ce cas, le mot de code x sera donné par la formule :

$$X = U \times k^{\otimes n}. \quad (2.19)$$

Pour y arriver, il faut que le vecteur U soit construit de telle façon que les bits d'informations se trouvent dans les indices les plus favorables (fiables) en correspondance aux K lignes de  $k^{\otimes n}$  sélectionnées précédemment.

Prenons en considération l'exemple illustré en [4] qui montre un code polaire CP (N,K) (code polaire (N,K)) avec N=8 et K=4 et un message U = [0,0,0, u<sub>3</sub>, 0, u<sub>5</sub>, u<sub>6</sub>, u<sub>7</sub>] avec la position des bits gelés arbitraire à l'aide d'une multiplication matricielle. Le mot de code est obtenu par le produit U × F :

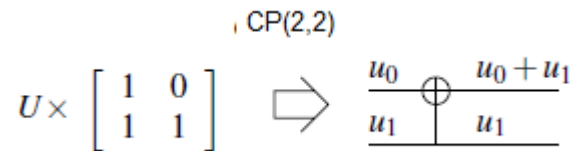
$$U \times F = [0,0,0, u_3, 0, u_5, u_6, u_7] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.20)$$

$$U \times F = \begin{bmatrix} u_3 + u_5 + u_6 + u_7 \\ u_3 + u_5 + 0 + u_7 \\ u_3 + 0 + u_6 + u_7 \\ u_3 + 0 + 0 + u_7 \\ 0 + u_5 + u_6 + u_7 \\ 0 + u_5 + 0 + u_7 \\ 0 + 0 + u_6 + u_7 \\ 0 + 0 + 0 + u_7 \end{bmatrix}^T \quad (2.21)$$

L'un des caractéristiques de la construction de la matrice génératrice est la récursivité qui est la capacité de se faire appel à elle-même. La construction de son graphe peut être aussi récursive. [4]



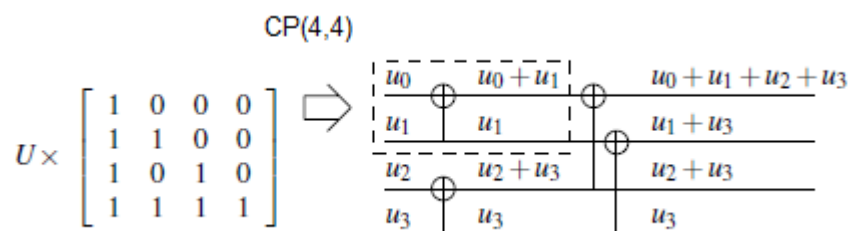
Voici un exemple [4] où il y a deux matrices génératrices avec leurs graphes respectifs. Dans la figure suivante, le code polaire est de taille  $N = 2$  avec un message  $U = [u_0, u_1]$ ,  $CP(2,2)$ , et ses représentations matricielles et graphiques.



**Figure 2.6 :** Matrice génératrice et graphe de codeur de Code Polaire CP(2,2). [4]

C'est l'une des représentations la plus simple de codage d'un code polaire. La présence d'un nœud qui fait l'union entre la donnée  $u_0$  et le  $u_1$  pour former le  $u_0 + u_1$  est remarqué. L'opérateur (+) représente un ou exclusif.

La figure 2.7 suivante montre le cas d'un code polaire de taille  $N = 4$  avec le message  $U = [u_0, u_1, u_2, u_3]$  :  $CP(4,4)$ .



**Figure 2.7 :** Matrice génératrice et graphe de codeur de Code Polaire CP(4,4). [4]

Les nœuds de variable représentés par ( $\perp$ ) permettent de faire passer et conduire les valeurs binaires, tandis que les nœuds de parité représentés par ( $\oplus$ ) se comportent comme des XOR et associent les valeurs binaires avec une autre valeur.

Un code de taille  $2N$  utilise deux fois la structure de celle de taille  $N$  et ensuite il additionne les  $N$  premières lignes et fait la transmission directe des  $N$  derniers lignes, comme il peut être mis en évidence dans la formule suivante :

$$U_{n+1} = [U_n, U'_n] \tag{2.22}$$

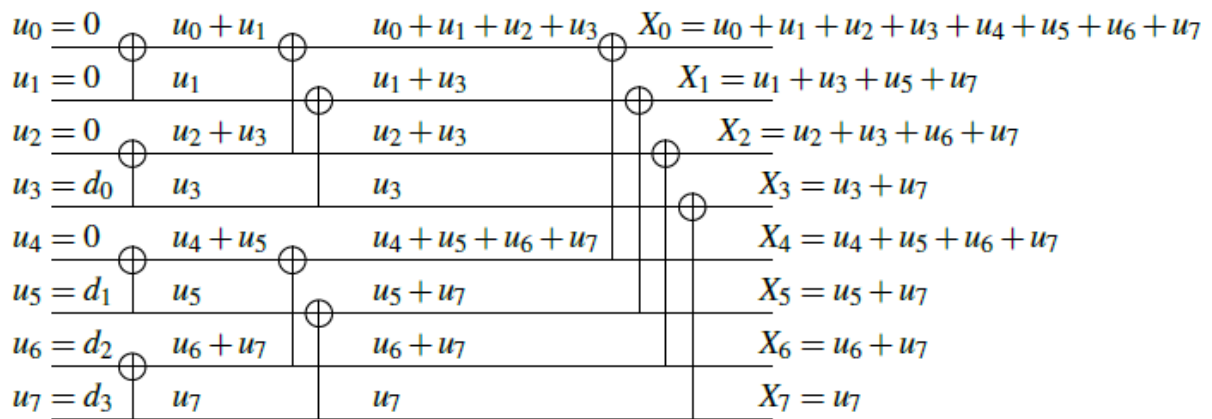
Où  $U_n$  représente les  $N$  premiers bits de  $U_{n+1}$  et  $U'_n$  représente les  $N$  derniers bits de  $U_{n+1}$ .

$$\begin{aligned}
 U_{n+1} \times F_{n+1} &= [U_n, U'_n] \times F_{n+1} \\
 &= [U_n, U'_n] \times \begin{bmatrix} F_n & 0_n \\ F_n & F_n \end{bmatrix} \\
 &= [U_n \times F_n + U'_n \times F_n, U'_n \times F_n] \\
 &= \left[ [U_n, U'_n] \times \begin{bmatrix} F_n \\ F_n \end{bmatrix}, U'_n \times F_n \right] \\
 &= \left[ U_{n+1} \times \begin{bmatrix} F_n \\ F_n \end{bmatrix}, U'_n \times F_n \right] \tag{2.23}
 \end{aligned}$$

Le  $U'_n \times F_n$  représente les  $N$  derniers bits de  $U_{n+1}$  qui sont directement transmis, et le  $U_{n+1} \times \begin{bmatrix} F_n \\ F_n \end{bmatrix}$  représente l'association entre les premiers et les dernières bits de  $U_{n+1}$ , comme il doit être démontré. [4]

Les processus de codage et décodage font l'usage de FG(s). Dans un code polaire, le FG avec une taille  $N = 2^n$ , en général, est formé par  $n$  étages de  $\frac{N}{2}$  nœuds avec la parité de degré 3 et  $\frac{N}{2}$  nœuds de variables de degré 3. Le degré d'un nœud peut être mesuré par la quantité de liaisons qu'il a avec des autres nœuds. Dans le processus de codage, il y a la propagation du vecteur d'entrée  $U$  sur le côté gauche, avec le but de créer le mot de code  $X$  sur le côté droit.

La figure 2-8 ci-dessous illustre un code polaire de rendement  $R = 0.5$ ,  $N = 2K$  car  $R = \frac{K}{N}$ , dont le vecteur d'entrée  $U$  est divisé sur la moitié en formant deux parties, une partie est formée par les bits gelés et l'autre par les bits d'information  $d_i$ .



**Figure 2.8** : FG d'un codeur de Code Polaire CP (8;4) non systématique. [4]

Le résultat est comme celui de la figure 2.7, les N dernières lignes ont été transmises directement et les N premières lignes sont sommées.

## 2.4 Décodage par Annulation Successive

Les codes polaires ont été créés de telle façon que leur décodage soit fait par le processus SCD en réception et avec un concept de sortie dure (*hard decision*), en général, dans lequel les données à la sortie sont composées par bits.

Le décodage des codes polaires utilise la notion du logarithmique du rapport de vraisemblance LLR (*logarithmical likelihood ratio*) qui permet de fournir les estimations en sortie du décodeur sous forme de rapport logarithmique de vraisemblance et est donnée par l'équation suivante :

$$L_i = \log \left( \frac{\Pr(y_i/x_i=0)}{\Pr(y_i/x_i=1)} \right) \quad (2.24)$$

Où  $\Pr(y_i/x_i = "b")$  désigne la probabilité conditionnelle d'avoir  $y_i$  sachant que  $x_i$  prenne la valeur binaire b.

Ce rapport cherche à déterminer la valeur d'entrée  $x_i$  plus probable pour chacune des données de sortie du canal  $y_i$ . Il permet d'obtenir le degré de confiance ou fiabilité de cette information en observant la valeur absolue. Le signe correspond à la décision dure :

- si  $L_i \geq 0$  alors le decodé est bit égal à 0
- si  $L_i < 0$  alors le decodé est bit égal à 1

En réalité, le LLR est tout simplement le logarithme népérien de LR (Likelihood Ratio) qui définit le rapport de vraisemblance. Ces deux notions sont généralement utilisées lorsque le démodulateur/décodeur est à sortie souple (*soft decision*), c.-à-d., les données de la sortie se présentent sous forme des estimations. Les LLR sont typiquement utilisés dans l'algorithme de décodage par propagation de croyance (*belief propagation*).

En visant l'amélioration des performances de correction, plusieurs travaux ont été proposés et en grande partie, ce sont tout simplement des petites variations du SCD, telle le SCL, l'algorithme d'annulation successive par pile SCS (*successive cancellation stack*) et celui de l'annulation successive par flip SCF (*successive cancellation flip*). Mais il existe un autre algorithme qui est très intéressant car il propose d'associer l'algorithme SC à l'algorithme BP. C'est l'algorithme à d'annulation souple SCAN (*soft cancellation*).

Dans notre travail, nous allons nous limiter au SCD. Ce décodage peut être représenté sous forme de FG et aussi sous forme des arbres binaires. [14]

### **2.4.1 Décodage SC par graphe de factorisation**

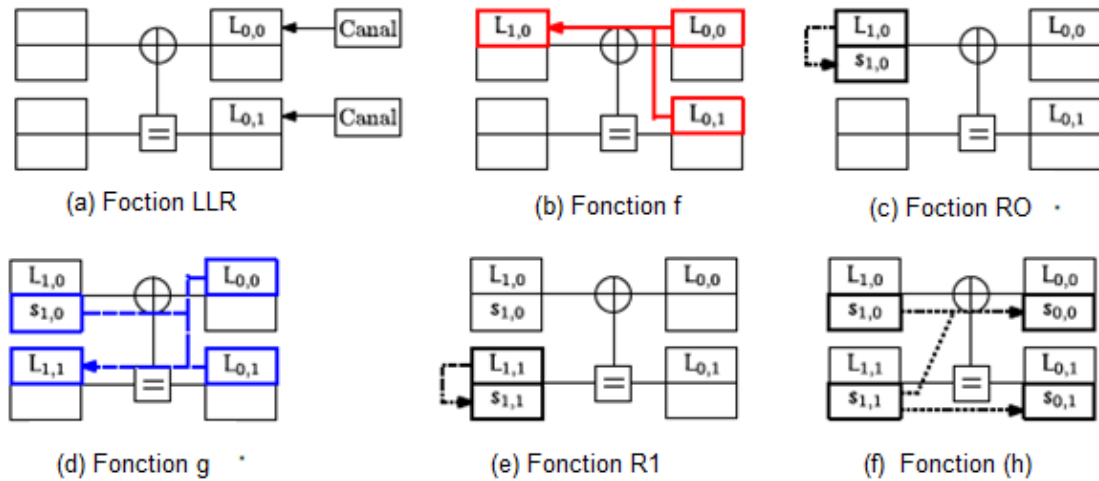
Lors de l'encodage, le FG peut être réalisé en faisant des associations entre les bits pour former un mot de code et en utilisant l'opérateur XOR ( $\oplus$ ) à partir de la gauche vers la droite.

Pour le cas du décodage, son FG doit prendre le chemin inverse. Les informations du canal viennent de la droite vers la gauche. Elles représentent les estimations des bits du mot de code. La figure 2.9 suivante montre ce processus de décodage et ses étapes pour  $N=2$  (taille 2).

Dans cette figure, le canal utilisé est un modèle de canal composite qui fait l'association de la valeur d'entrée  $x = \{0,1\}$  à une valeur réelle  $\hat{x} = \{-1,1\}$  à l'aide de la modulation BPSK (*binary phase shift keying*). Il prend comme base le canal AWGN à entrée binaire qui est un modèle utilisé pour caractériser les performances des codes correcteurs d'erreurs.

La sortie de ce canal est des LLR ( $s$ ) qui représentent les données d'entrée de notre processus de décodage pour être traités dans l'algorithme. Les LLR( $s$ ) sont contenus dans le vecteur  $L$  de taille  $N$ .

Le vecteur  $\hat{b}$  donne la sortie de cet algorithme de décodage avec la taille  $K$ . Un aspect intéressant de cet algorithme de décodage est qu'il permet de trouver deux formats de données, les LLR et les sommes partielles PS (*partial sums*). Les LLR( $s$ ) représentent les valeurs approximées (estimation) du bit dans une position  $i, j$  ( $L_{i,j}$ ) lorsque les PS représentent la prise de décision, soit 0 ou 1, comme valeur de ce même bit du factor graph. Donc pour avoir un bon décodage pour  $N = 2$ , il faut avoir 4 variables pour les LLR ( $L_{00}, L_{01}, L_{10}, L_{11}$ ) et 4 variables pour les PS ( $s_{00}, s_{01}, s_{10}, s_{11}$ ). Ce qui donne au totale huit (08) variables, d'après la fonction  $h$ .



**Figure 2.9** : Fonctions élémentaires et séquençage du décodage SC d'un noyau  $N = 2$ . [14]

La figure 2.9 ci-dessus montre toutes ces variables et les étapes qu'elles font apparaître. La récupération des LLR du canal  $L$  ( $L_{0,0}, L_{1,0}$ ) se déroule dans l'étape (a), qui sont les données d'entrée dans ce décodage. Les flèches présentes dans chaque étape indiquent les opérations intermédiaires qui sont utilisées pour calculer le LLR et les PS. Ces opérations sont appelées  $f$ ,  $R_0$ ,  $g$ ,  $R_1$  et  $h$  et elles sont calculées par les équations suivantes :

$$f(L_a, L_b) = \text{sign}(L_a \cdot L_b) \cdot \min(|L_a|, |L_b|) \quad (2.25)$$

$$g(L_a, L_b, \hat{s}_a) = (1 - 2\hat{s}_a)L_a + L_b \quad (2.26)$$

$$h(\hat{s}_a, \hat{s}_b) = (\hat{s}_a \oplus \hat{s}_b, \hat{s}_b) \quad (2.27)$$

$$R_0(L_a) = 0 \quad (2.28)$$

$$R_1(L_a) = \begin{cases} 0 & \text{si } L_a \geq 0 \\ 1 & \text{si } L_a < 0 \end{cases} \quad (2.29)$$

Le calcul de la variable  $L_{1,0}$  est fait dans l'étape (b) à partir de l'utilisation de la fonction  $f$ . Dans le nœud supérieur, la fonction  $R_0$  de l'étape (c) est utilisé seulement dans le cas où de la présence d'un bit gelé (égale à 0) en  $u_0$ . Au niveau de l'étape (d), la

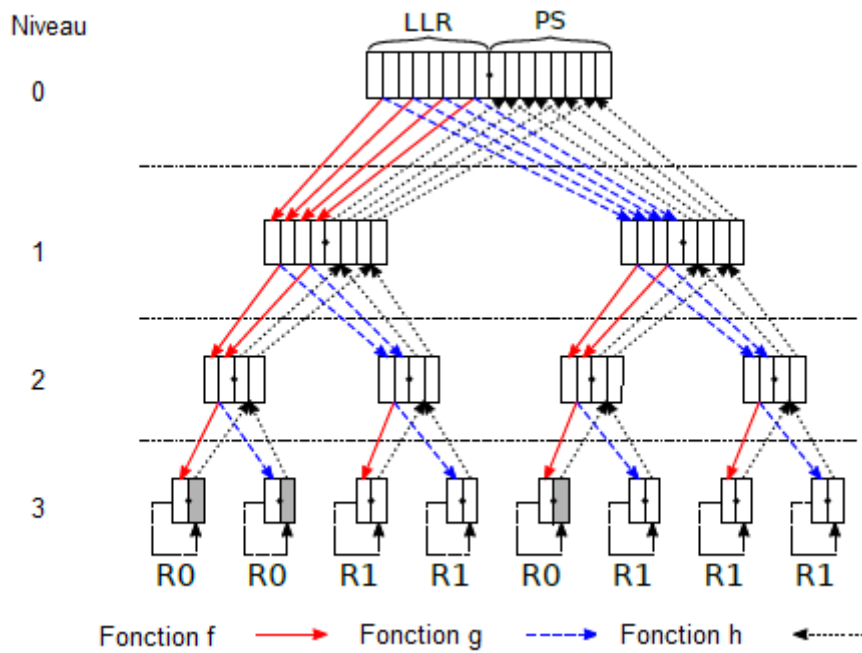
fonction  $g$  fait le calcul de la variable  $L_{11}$ . A l'aide de la fonction  $R1$ , le  $s_{11}$  est calculé à partir de  $L_{11}$  dans l'étape (e).

Cette opération est correspondante à un bit d'information. C'est la notion de seuillage (*thresholding*) qui est une opération consistant à donner une valeur de 0 à tous les bits qui ont une valeur inférieure à une certaine valeur déjà établie et attribuer la valeur de 1 à tous les bits qui ont une valeur supérieure à celle déjà établie. Donc, pour obtenir la somme partielle ( $s_{11}$ ), il faut faire le seuillage du LLR à partir de l'opération  $R1$ . Puisque le seuillage ne fait que des approximations, ce qui entraîne la perte de l'information de fiabilité après son utilisation.

Les étapes précédentes se limitent seulement à faire des calculs pour voir le changement des informations initiales. Après le décodage de la somme partielle  $s_{11}$ , la fonction  $h$  est appliquée afin de propager les sommes partielles, lors de l'étape (f).  
[14]

### **2.4.2 Décodage SC par arbre binaire**

Ce type de représentation du décodage SC se diffère de la précédente car elle a une représentation algorithmique le plus simple, c.-à-d., plus facile à comprendre. Elle est aussi divisée en niveaux mais ces données s'affichent sous le format d'un arbre. Il est possible d'avoir  $\log_2 N + 1$  niveaux. Pour mieux comprendre son fonctionnement, observons cet arbre à quatre (04) niveaux selon la figure 2.10 suivante.



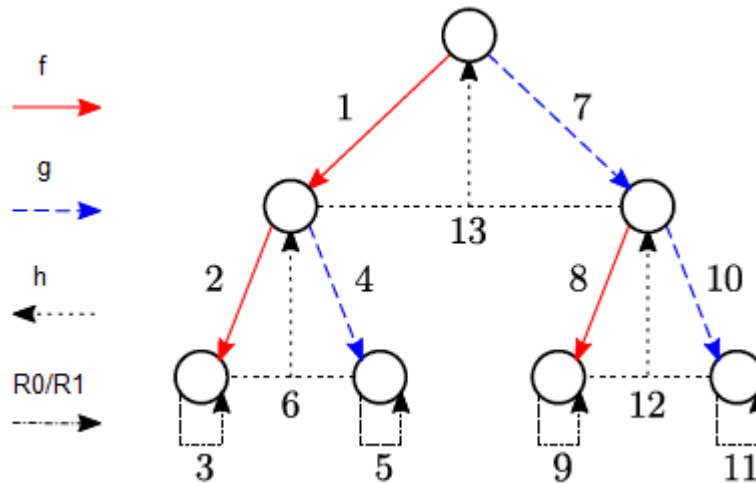
**Figure 2.10** : Arbre de décodage SC. [14]

Dans chaque niveau de cet arbre, il y a au moins un (01) 1 nœud (*node*). Le nœud principale, qui se localise au niveau 0, est appelée racine (*root*). Le reste des niveaux de cet arbre sont des ramifications de la racine et il contient un N nombre des LLR et des sommes partielles au niveau de chaque nœud. A la fin de l'arbre, les derniers nœuds, au niveau 3 dans la figure, sont appelés feuilles (*leaves*). Ils peuvent être manipulés à partir des calculs des fonctions R0 et R1.

Les nœuds sont formés dans chaque niveau  $d$  par  $2^{n-d}$  LLR et  $2^{n-d}$  PS, sachant que  $n = \log_2 N$ . Ce qui donne à la fin  $2^d$  nœuds par niveau. L'application de la fonction  $h$  permet de diffuser les sommes partielles. Cette diffusion est totalement libérée jusqu'à ce qu'il trouve un niveau dépendant de l'indice de feuille qui vient d'être traité.

Une séquence décodée  $\hat{u}$  est possible à avoir pour un code polaire non systématique. Cette séquence représente les feuilles de cet arbre de décodage c-à-d, elles sont les sommes partielles des nœuds dans le dernier niveau. Voici la représentation plus simple de l'arbre de décodage :





**Figure 2.11** : Ordonnancement du décodage SC d'un code polaire de taille  $N = 4$ . [14]

En arrivant à cette étape, il n'est pas intéressant de recalculer les sommes partielles de la racine, c.-à-d. les opérations 12 et 13 sont inutiles.

Les sommes partielles de la racine peuvent correspondre au mot de code décodé pour le cas d'un code systématique. Cet arbre offre plusieurs chemins. Pour avoir un bon décodage il faut suivre l'ordre et comme il a été indiqué dans la figure précédente, avec plusieurs flèches numérotées. Chacune d'elles représente une fonction et est numérotée selon son ordre d'exécution. [14]

## 2.5 Spécificité des codes polaires pour le canal AWGN et le canal de Rayleigh

La synthèse des codes polaires pour ces deux canaux se différencie selon certaines caractéristiques et quelques méthodes de son implémentation aux niveaux de chaque canal tels que la façon dont les codes polaires sont construits.

### 2.5.1 Canal AWGN

Dans ce canal, la construction de code polaire se base sur le fait de sélectionner  $K$  meilleur parmi  $N$  canaux binaires polaires possibles de n'importe quel algorithme dans le rapport signal / bruit de design SNR (*signal to noise ratio*) en termes de taux d'erreur des bits BER (*bit error rate*). Mais la construction optimale du code polaire est difficile et par conséquent, de nombreuses constructions de codes polaires sous-optimales ont été proposées à différentes complexités de calcul. Les codes polaires sont également non-universels, ce qui signifie que le code change considérablement avec le design SNR.

Pour un canal discret de type BSM la polarisation des canaux est une technique par laquelle on fabrique  $N$  canaux polarisés canaux-bit (*bit channel*) sur  $N$  copies indépendantes et identiques des canaux BSM. Les canaux sont polarisés sans aucune perte de capacité, dans le sens où ils sont soit extrêmement bruyants ou silencieux comme  $N \rightarrow \infty$ .

Ensuite, un taux de transmission proche de la capacité peut être réalisé en choisissant simplement pour transmettre uniquement sur les bons canaux binaires. Cependant, à tout longueur de bloc finie  $N$  et taux  $R = K/N$ , un algorithme de classement pour les canaux binaires en fonction de leur taux d'erreur binaire (BER) devient nécessaire pour sélectionner  $K$  bons canaux sur  $N$ , avec  $K$  est le nombre de bits d'information dans chaque mot de code de longueur  $N$ . La construction du code polaire est essentielle pour obtenir les meilleures performances à des longueurs de bloc finies. C'est qu'elle a une définition explicite en théorie mais difficile en pratique car l'estimation précise du bit-canaux est insoluble.

Une caractéristique importante des codes polaires est leur non-universalité grâce auquel il est possible de construire un code polaire à un design SNR et l'utiliser pour une gamme de SNR possibles. Autrement dit, différents codes polaires sont générés en fonction de la valeur spécifiée du SNR, connu sous le nom de design SNR. Un changement du SNR de fonctionnement est possible en pratique mais un changement de code selon SNR n'est pas souhaitable.

Par contre, le choix du design SNR est critique pour la performance à tous les SNR d'intérêt et aucune étude n'a permis d'identifier le meilleur design SNR pour toute construction du code polaire.

Un problème majeur avec les codes polaires est aussi leur performance inférieure du BER à des longueurs de bloc finies, par rapport à l'état des codes LDPC et Turbo de la même longueur de bloc.

De plus, nous trouvons qu'une telle performance dépend de nombreux paramètres tels que rapport, longueur du bloc et l'algorithme utilisé pour la construction. Cette observation fait une recherche exhaustive du design SNR inévitable.

Le fait que le canal AWGN soit dépendant SNR au niveau du récepteur présente un inconvénient, car le code polaire doit être reconstruit chaque fois que le SNR change afin de garantir performances optimales.

Heureusement, la dépendance n'est pas extrêmement stricte dans le sens où un code construit pour un SNR spécifique peut maintenir une bonne performance sur une gamme de SNR voisins. [16]

## **2.5.2 Canal de Rayleigh**

Afin de construire un code polaire pour un canal, il faut étudier les performances des sous-canaux binaires induites par la transformation polarisante. Typiquement, l'intéressant c'est la probabilité d'erreur sur les bits pour ces sous-canaux. Ces informations sont nécessaires non seulement pour un concepteur de code, mais aussi pour les algorithmes de décodage séquentiel qui ont une complexité moyenne très faible et fournissent des performances de vraisemblance.

Pour tout canal de type BSM, les probabilités d'erreur de sous-canal peuvent être calculées avec la complexité  $O(n\mu^2 \log \mu)$  en utilisant une implémentation d'évolution de densité. Cependant, cet algorithme est assez difficile à mettre en œuvre et nécessite un nombre élevé de  $\mu$  de niveaux de quantification afin d'obtenir une précision raisonnable.

Dû à la sous-optimalité de l'algorithme SCD, et à la mauvaise distance minimale des codes polaires, les performances pratiques des codes polaires de longueur moyennes semblent d'être bien pire par rapport au turbo code et le code LDPC.

Le SCD ne parvient pas à fournir un décodage vraisemblance maximale. Cependant, sa liste et ses extensions ont démontré que les performances fournies sont presque optimales pour une liste de taille suffisamment grande. Cette performance semble pourtant assez médiocre en raison de la très faible distance minimale des codes polaires. Pour résoudre ce problème, il est suggéré de définir certains symboles d'entrée gelés  $u_i$  de la transformation polarisante non à 0, mais à certaines combinaisons linéaires d'autres symboles  $u_i$ . Aussi que les combinaisons linéaires doivent être construites d'une telle manière que le code polaire obtenu à une distance minimale soit suffisamment élevé. [17]

## 2.6 Conclusion

En conclusion, les codes polaires sont les premiers codes pouvant atteindre la capacité du canal pour les canaux sans mémoires, vu leur faible complexité d'encodage et de décodage. Pour concevoir un bon codage, il suffit de choisir les meilleurs canaux pour envoyer les bits d'information : soit un canal AWGN soit un canal de Rayleigh, et utiliser un algorithme permettant de construire les codes polaires de façon plus simple parmi tant d'autres, à savoir l'algorithme basé sur le paramètre de Bhattacharyya.

Malgré leur perfection, ils représentent aussi des faiblesses surtout au niveau de leurs performances, ce qui nous mène à faire des études et tests pratiques dans le chapitre suivant.

# Chapitre 3 Simulation des codes polaires

---

## 3.1 Introduction

Dans ce dernier chapitre consacré à la concrétisation de la théorie par des simulations, nous allons programmer, en Matlab, les deux principales techniques de décodage étudiés en théorie en l'occurrence les techniques SCD et SCLD dans un canal AWGN. Les principes de décodage seront basés sur l'arbre binaire et leurs principes de fonctionnement seront expliqués. En fin des résultats ainsi que des courbes BER vs SNR (préliminaires) montrant les performances des codes polaires par rapport à d'autres codes et les performances du SCLD par rapport au SCD y seront présentées.

## 3.2 Design SNR et séquence de rentabilité (*reliability sequence*)

Comme déjà vu au chapitre II (§ 2.4.1), une caractéristique importante des codes polaires est le design-SNR (design Signal-to-Noise-Ratio) qui est une valeur spécifique du SNR pour laquelle le code change à chaque fois que cette dernière change.

## 3.3 Séquence de fiabilité (*reliability sequence*)

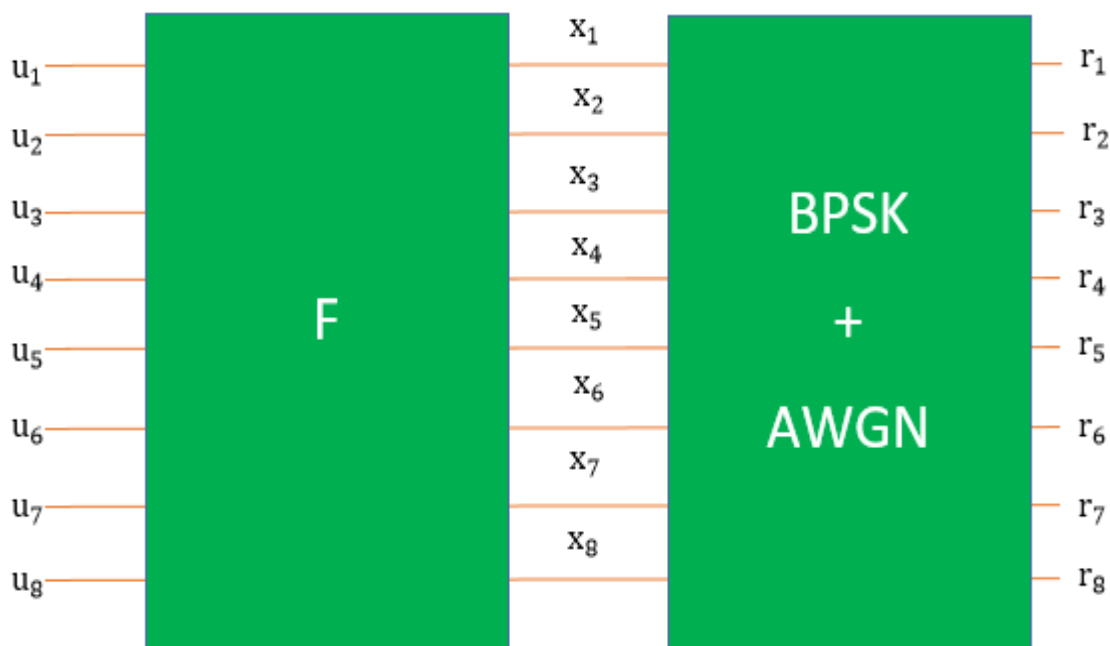
Dans une transmission typique utilisant les codes polaires et après l'opération de polarisation, les bits sont modulés en BPSK (*binary phase shift keying*) et sont transmis à travers un canal AWGN (*additive white gaussian noise*) (voir figure 3.1). Certains canaux bit deviennent très fiable (sans bruit) et d'autres très entachés de bruit comme il a été vu au § 2.1.1.

Question : Comment identifier les bons canaux des mauvais canaux?

La réponse se trouve dans la séquence de fiabilité donnée dans la norme 3GPP TS 38.212 [17] en page 19.

Une séquence de fiabilité est un ordonnancement des canaux bit du pire au meilleur parmi les  $N$  bits à coder.

Cette séquence de fiabilité selon la norme 5G, pour  $N = 1024$  est donnée dans la figure 3.3.



**Figure 3.1** : Chaîne de transmission basique utilisant les codes polaire. F est une sous matrice de la 4<sup>ème</sup> puissance de Kronecker [16]

Plusieurs méthodes sont citées dans la littérature pour trouver ces séquences, combien réalistes sont-elles...etc. [16]

Pour le travail de ce mémoire, nous ne nous sommes pas intéressés à ces méthodes mais seulement à l'affichage et la façon d'utiliser cette séquence.

Il s'avère d'après le tableau de la figure 3.3 que par exemple pour :

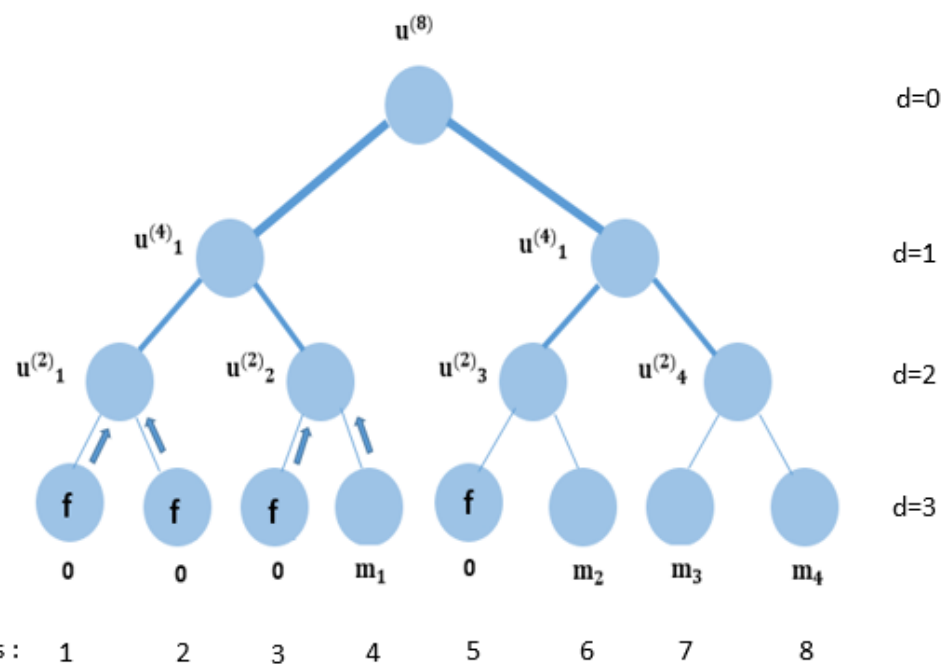
$$N = 8 : r_{s_8} = 1 \ 2 \ 3 \ 5 \ 4 \ 6 \ 7 \ 8$$

Si canal bit de position 1 est le moins faible suivi de le 2<sup>ème</sup>, du 3<sup>ème</sup>, du 5<sup>ème</sup>, du 4<sup>ème</sup>, du 6<sup>ème</sup>, du 7<sup>ème</sup> et du 8<sup>ème</sup>.

Pour un code polaire CP(N = 8, K = 4) par exemple :

- les (N - K) positions des bits gelés sont 1 2 3 5
- les K positions des bits de messages sont 4 6 7 8

Ceci peut être représenté à l'aide de l'arbre binaire (voir figure 3.2)



**Figure 3.2** : Position des bits gelés (f) selon la séquence de fiabilité, pour un code polaire CP(8,4). [16]

Avec : d le niveau de profondeur (*depth*) de l'arbre binaire pour  $i = 0$  à 3 et  $d = 3$  est le niveau des feuilles (*leafs*) de l'arbre.  $m_i$  sont les bits du message et les 0 sont les bits gelés (*frozen*).

Nous rappelons qu'à chaque niveau de profondeur nous avons les équations suivantes :

$$u^{(2)}_1 = [0 \ 0] \tag{3.1}$$

$$u^{(2)}_2 = [m_1 \ m_1] \tag{3.2}$$

$$u^{(2)}_3 = [m_2 \ m_2] \tag{3.3}$$

$$u^{(2)}_4 = [m_3 + m_4 \ m_4] \tag{3.4}$$

$$u^{(4)}_1 = [u^{(2)}_1 + u^{(2)}_2 \quad u^{(2)}_2] \tag{3.5}$$

$$u^{(4)}_2 = [u^{(2)}_3 + u^{(2)}_4 \quad u^{(2)}_4] \tag{3.6}$$

$$u^{(8)} = [u^{(4)}_1 + u^{(4)}_2 \quad u^{(4)}_2] \tag{3.7}$$



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

83	76	211	114	339	182	467	405	595	668	723	470	851	917	979	986
84	137	212	277	340	643	468	303	596	790	724	483	852	727	980	943
85	82	213	156	341	562	469	569	597	460	725	415	853	493	981	891
86	56	214	87	342	286	470	244	598	249	726	485	854	873	982	998
87	27	215	197	343	585	471	595	599	682	727	905	855	701	983	766
88	97	216	116	344	299	472	189	600	573	728	795	856	931	984	511
89	39	217	170	345	354	473	566	601	411	729	473	857	756	985	988
90	259	218	61	346	211	474	676	602	803	730	634	858	860	986	1001
91	84	219	531	347	401	475	361	603	789	731	744	859	499	987	951
92	138	220	525	348	185	476	706	604	709	732	852	860	731	988	1002
93	145	221	642	349	396	477	589	605	365	733	960	861	823	989	893
94	261	222	281	350	344	478	215	606	440	734	865	862	922	990	975
95	29	223	278	351	586	479	786	607	628	735	693	863	874	991	894
96	43	224	526	352	645	480	647	608	689	736	797	864	918	992	1009
97	98	225	177	353	593	481	348	609	374	737	906	865	502	993	955
98	515	226	293	354	535	482	419	610	423	738	715	866	933	994	1004
99	88	227	388	355	240	483	406	611	466	739	807	867	743	995	1010
100	140	228	91	356	206	484	464	612	793	740	474	868	760	996	957
101	30	229	584	357	95	485	680	613	250	741	636	869	881	997	983
102	146	230	769	358	327	486	801	614	371	742	694	870	494	998	958
103	71	231	198	359	564	487	362	615	481	743	254	871	702	999	987
104	262	232	172	360	800	488	590	616	574	744	717	872	921	1000	1012
105	265	233	120	361	402	489	409	617	413	745	575	873	501	1001	999
106	161	234	201	362	356	490	570	618	603	746	913	874	876	1002	1016
107	576	235	336	363	307	491	788	619	366	747	798	875	847	1003	767
108	45	236	62	364	301	492	597	620	468	748	811	876	992	1004	989
109	100	237	282	365	417	493	572	621	655	749	379	877	447	1005	1003
110	640	238	143	366	213	494	219	622	900	750	697	878	733	1006	990
111	51	239	103	367	568	495	311	623	805	751	431	879	827	1007	1005
112	148	240	178	368	832	496	708	624	615	752	607	880	934	1008	959
113	46	241	294	369	588	497	598	625	684	753	489	881	882	1009	1011
114	75	242	93	370	186	498	601	626	710	754	866	882	937	1010	1013
115	266	243	644	371	646	499	651	627	429	755	723	883	963	1011	895
116	273	244	202	372	404	500	421	628	794	756	486	884	747	1012	1006
117	517	245	592	373	227	501	792	629	252	757	908	885	505	1013	1014
118	104	246	323	374	896	502	802	630	373	758	718	886	855	1014	1017
119	162	247	392	375	594	503	611	631	605	759	813	887	924	1015	1018
120	53	248	297	376	418	504	602	632	848	760	476	888	734	1016	991
121	193	249	770	377	302	505	410	633	690	761	856	889	829	1017	1020
122	152	250	107	378	649	506	231	634	713	762	839	890	965	1018	1007
123	77	251	180	379	771	507	688	635	632	763	725	891	938	1019	1015
124	164	252	151	380	360	508	653	636	482	764	698	892	884	1020	1019
125	768	253	209	381	539	509	248	637	806	765	914	893	506	1021	1021
126	268	254	284	382	111	510	369	638	427	766	752	894	749	1022	1022
127	274	255	648	383	331	511	190	639	904	767	868	895	945	1023	1023

**Figure 3.3** : Séquence de fiabilité (dans les colonnes paires) pour un code de taille  $N = 1024$  [17].

Cette séquence nous permet de retrouver donc les indices  $(N - K)$  des positions des canaux les moins fiables où seront placés les bits gelés et les  $K$  indices de canaux restantes seront réservés les  $K$  bits d'information.

Autre exemple CP(16,10) :

$r_{s_{16}}$ : 1 2 3 5 9 4 6 10 7 11 13 8 12 15 16

Indice des bits gelés sont : 1 2 3 5 9 4

Indices des bits de message sont : 6 10 7 11 13 8 12 15 16.

### 3.4 Encodage des codes polaires

L'encodage des codes polaire est obtenu par l'équation (2.19) du chapitre II. Cette équation peut être traduite par un arbre binaire on prend celui de la figure 2.11, on a sens de codage du bas vers le haut, c.-à-d., de  $d = 3$  à  $d = 0$  et à chaque niveau  $d$  nous avons des équations (voir les équations).

Les points essentiels du programme en matlab sont :

- Saisir la sequence de fiabilité fixe par la norme 3GPP TS 38.212, valable pour tout  $N \leq 1024$
- Fixer la valeur  $N(16, 64, \dots, 512, 1024)$
- Former un vecteur de  $N$  bits comme suit :
  - Trouver les  $N - K$  bits « mauvais » canaux-bit a entré de la séquence de fiabilité
  - Fixer les entrées  $u_i$  pour le  $N - K$  canaux-bit à 0.
  - Dans les  $K$  entrés  $u_i$  restantes introduire le message  $m$  à coder
  - Initialisation du vecteur mot de code à zéro
  - Initialiser le nombre de bit à combiner à 1, sachant que se nombre double lorsqu'on progresse de feuille de l'arbre vers la racine ( $d = 0$ )
  - Imbriquer deux boucles for-end.

- La boucle extérieure permet d'incrémenter le nombre de K bits à combiner (multiplier par 2 à chaque fois)
- La boucle interne permet de combiner les bits.

Résultats de la simulation :

On peut vérifier tous ces résultats en utilisant, entre autre, la matrice de codage appropriée pour  $N = 16$ . Résultats de la simulation à partir du *workspace* de malab :

```
>> Taille du code est :
```

```
N = 16
```

```
>> Taille du message à coder K est :
```

```
K = 8
```

```
>> Séquence de fiabilité pour N est :
```

```
Q1 = 1    2    3    5    9    4    6    10    7    11    13
      8    12   14   15   16
```

```
>> Indices des positions des bits gelés
```

```
F = 1    2    3    5    9    4    6    10
```

```
>> Message de K bits a encoder
```

```
msg = 1    1    0    0    1    1    0    0
```

```
>> Assignement des bits du message à leurs positions
```

```
u =  0    0    0    0    0    0    1    0    0    0    1
    1    0    1    0    0
```

```
>> Message en sortie de l'encodeur
```

```
u = 0    0    1    1    0    1    1    0    1    0    0
    1    1    1    0    0
```

### 3.5 Décodage de codes polaires

Dans ce travail, nous allons utiliser deux types de décodages le SCD et le SCLD.

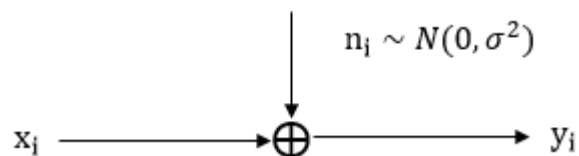
#### 3.5.1 Décodage par SCD (*successive cancellation decoder*)

En se basant sur la figure 3.1, nous avons en entrée des bits  $[u_1, u_2, \dots, u_N]$ , certains de ces bits sont gelés et d'autres sont les messages à coder.

Les bits à la sortie du codeur sont modulés en BPSK (*binary phase shift keying*), où le 0 est codé en +1 et le 1 en -1.

Ce canal est du type AWGN (*additive white gaussian noise*). C'est le plus important des canaux BMS (*binary memoryless channel*) qui modélise une grande partie des canaux de communications.

Le canal B-AWGN (*binary AWGN*) est représenté en figure 3.4



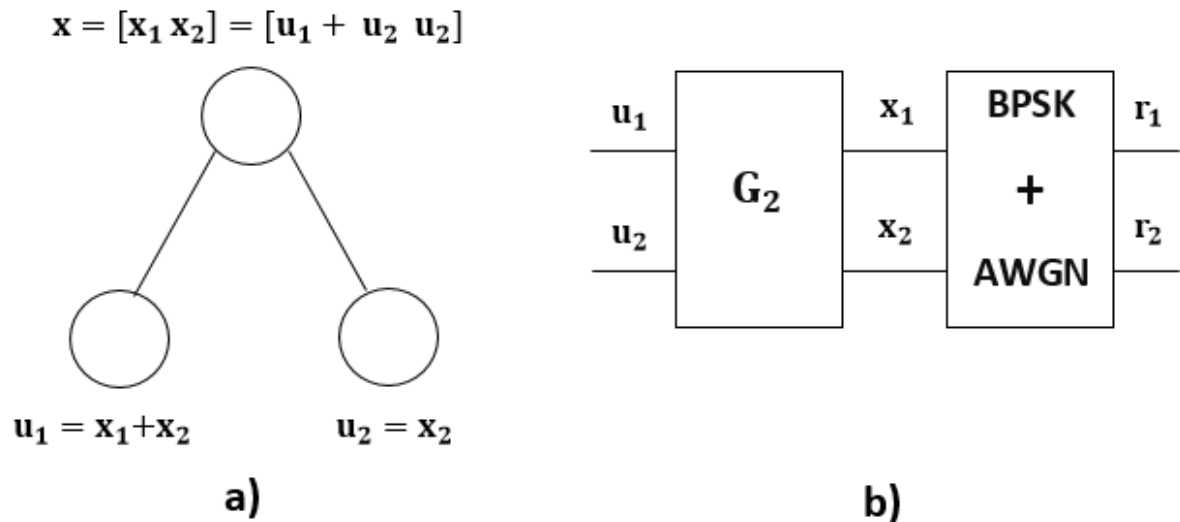
**Figure 3.4** : canal BAWGN

Le B-AWGN a une moyenne nulle et une variance  $\sigma^2$  (énergie du bit).  $x_i$  est le bit d'entrée et  $y_i$  est la sortie du canal ( $y_i \in \mathbb{R}$ ).

$$y_i = x_i + m_i \quad (3.8)$$

Le canal BAWGN est caractérisé par le rapport signal au bruit SNR.

Dans cette partie de simulation, nous allons utiliser le décodage du type *minsum* à l'aide de l'arbre binaire. Le principe de cette technique pour  $N = 2$  est le suivant (fig 3.5):



**Figure 3.5 :** CP (2,2) a) arbre binaire pour un CP (2,2), b) principe du codeur. [18]

Sur la figure 3.5 b), la sortie  $r_1$  est la croyance (*belief*) pour  $x_1$  et  $r_2$  celle pour  $x_2$ .

Question : comment à partir de ces données peut-on décoder  $U_1$  et  $U_2$ ?

Sur le figure 3.5 a) on remarque que l'on peut aisément écrire  $U_1$  en fonction de  $x_1$  et  $x_2$  et  $U_2$  en fonction de  $x_2$ .

Ayant la croyance pour  $x_1$  et celle pour  $x_2$ , quelle est la croyance de  $x_1 + x_2$  ?

Avec + : l'opérateur XOR.

La méthode minsum stipule que :

$$L(U_1) = f(r_1, r_2) = \text{sign}(r_1) \text{sig}(r_2) \min(|r_1|, |r_2|) \quad (3.9)$$

$$\hat{U}_1 = 0 \text{ si } L(U_1) \geq 0 \quad (3.10)$$

$$\hat{U}_1 = 1 \text{ si } L(U_1) < 0 \quad (3.11)$$

Ces deux décisions sur  $U_1$  et  $U_2$  sont appelées décisions durs (*hard decision*).

$L(U_1)$  est la croyance pour  $U_1$ . Une fois  $\hat{U}_1$  calculée, on peut l'utiliser pour décoder  $U_2$  de la manière suivante :

$$L(U_2) = g(r_1, r_2, \hat{U}_1) = r_2 + (1 - 2b)r_1 \quad (3.12)$$

Avec  $b$  est la valeur binaire de l'estimation de  $\hat{U}_1$ .

$$\text{Si } b = 0, L(U_2) = r_1 + r_2 \quad (3.13)$$

$$\text{Si } b = 1, L(U_2) = r_2 - r_1 \quad (3.14)$$

Une décision dure est appliquée à  $L(U_2)$ :

$$\text{Si } L(U_2) \geq 0 \text{ alors } \hat{U}_2 = 0 \quad (3.15)$$

$$\text{Si } L(U_2) < 0 \text{ alors } \hat{U}_2 = 1 \quad (3.16)$$

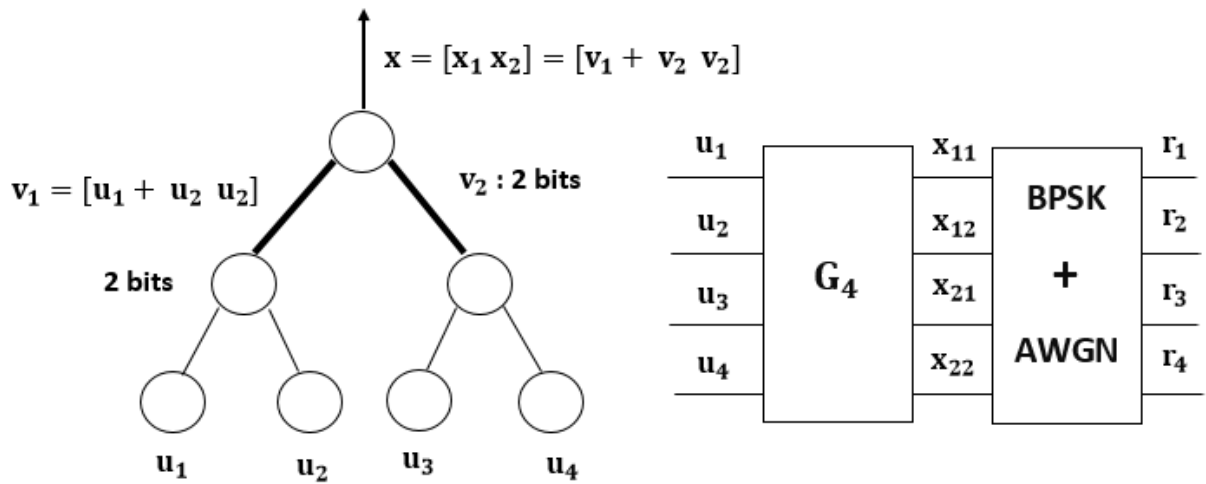
Une fois que les deux citant aient donné une décision dure chacun alors le nœud racine peut décider comme suit :

$$\hat{X} = [\hat{U}_1 + \hat{U}_2 \hat{U}_2] \quad (3.17)$$

Nous pouvons élargir ce principe à des codes plus larges, exemple CP(4,4) (voir figure 3.6.)

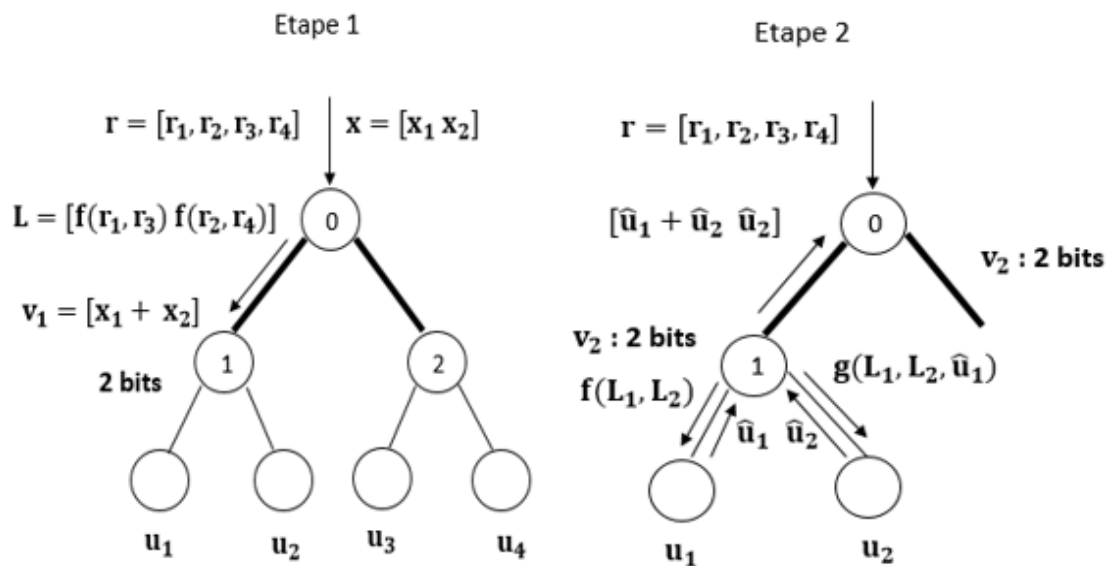
L'arbre binaire est plus large dans ce cas avec deux niveaux de profondeur. D'après la figure 3.6b) :

$r_1$  est la croyance pour  $x_{11}$ ,  $r_2$  celle pour  $x_{12}$ ,  $r_3$  celle pour  $x_{21}$  et,  $r_4$  est la croyance pour  $x_{22}$ .



**Figure 3.6 :** CP (4,4) a) arbre binaire, b) principe du codeur [18]

Donc  $r_1$  et  $r_2$ , ensemble, sont les croyances de  $X_1$  et  $r_3$  et  $r_4$ , ensemble, sont les croyances de  $X_2$ . Il y a un groupement par deux à chaque fois que l'on descend en profondeur.



**Figure 3.7 :** CP (4,4) étapes 1 et 2 de décodage [18]

On peut facilement vérifier que (voir figure 3.7)

$$r_1 = x_1 + x_2 \tag{3.18}$$



Ayant donc les croyances pour  $x_1$  et les croyances pour  $x_2$ , quelle est la croyance  $x_1 + x_2$  ?

On utilise la méthode minsum une nouvelle fois de la manière suivante :

$$h = [f(r_1, r_3)f(r_2, r_4)] = [L_1, L_2] \quad (3.19)$$

$$V = x_1 + x_2 = [x_{11} \ x_{12}] + [x_{21} \ x_{22}] \quad (3.20)$$

$$V = [x_{11} + x_{21} \ x_{12} + x_{22}] \quad (3.21)$$

Avec + : l'opérateur XOR

Donc  $f(L_1)$  n'est autre que la croyance pour  $x_{11}+x_{21}$  et  $f(L_2)$  est la croyance pour  $x_{12}+x_{22}$ .

Enfin pour le vecteur  $V$ , il y a deux croyances.

Nous nous retrouvons dans la situation  $N = 2$  (de la figure 3.5). Ça sera l'étape 2 dans figure 3.7.

$[L_1, L_2]$  arrivent au niveau du noeud 1 (ou enfant de gauche). Le reste du calcul est simple. On utilise la fonction  $f(L_1, L_2)$  de la technique minsum pour calculer  $\hat{U}_1$  ensuite on utilise la fonction  $g(L_1, L_2, \hat{U}_1)$  de la technique minsum pour estimer  $\hat{U}_2$ .

Au niveau de nœud 0 (ou racine) nous avons une estimation de  $[\hat{U}_1 + \hat{U}_2 \ \hat{U}_2]$ .

A présent, le nœud racine envoie ces informations au nœud 2 (ou enfant de droite) pour l'estimation des bits  $U_3$  et  $U_4$ . Sa fonction  $g$  est utilisé mais de la manière suivante :

$$L = [g(r_1, r_2, \hat{U}_1 + \hat{U}_2) \ g(r_2, r_4, \hat{U}_2)] = [L_3 \ L_4] \quad (3.22)$$

On se retrouve encore une fois dans le cas  $N = 2$  (de la figure 3.5).  $f(L_3, L_4)$  sera utilisé pour calculer  $\hat{U}_3$  (estimé de  $U_3$ ) et  $\hat{U}_3$  avec  $g(L_3, L_4)$  sera utilisé pour calculer  $\hat{U}_4$ .

Une fois  $\hat{U}_3$  et  $\hat{U}_4$  obtenus, ils seront combinés (dans le sens du codage) pour obtenir le vecteur :

$$V_2 = [\hat{U}_3 + \hat{U}_4 \quad \hat{U}_4] \quad (3.23)$$

Donc au niveau du nœud (racine 0) on obtient les vecteurs  $\hat{V}_1$  et  $\hat{V}_2$  qui vont être combinés pour obtenir le mot de code.

$$\hat{X} = [\hat{V}_1 + \hat{V}_2 \quad \hat{V}_2] = [\hat{U}_1 + \hat{U}_2 + \hat{U}_3 + \hat{U}_4 \quad \hat{U}_2 + \hat{U}_4 \quad \hat{U}_3 + \hat{U}_4 \quad \hat{U}_4] \quad (3.24)$$

Dans le cas général  $CP(N, K)$  et pour pouvoir écrire le programme en matlab, pour le SCD (*successive cancellation decoding*) nous devons:

- a) Avoir l'arbre binaire
- b) Connaitre le numéro du niveau de profondeur : il varie de 0 à  $n$  avec  $n = \log_2(N)$
- c) Connaitre le numéro de nœud à chaque niveau de profondeur : il varie de 0 à  $2^{d-1}$ .
- d) Savoir, pour chaque nœud, qu'est l'enfant de droite, l'enfant de gauche et les parents.

Remarque la racine ( $d = 0$ ) n'a pas de parents et les feuille ( $d = n$ ) n'ont pas d'enfants.

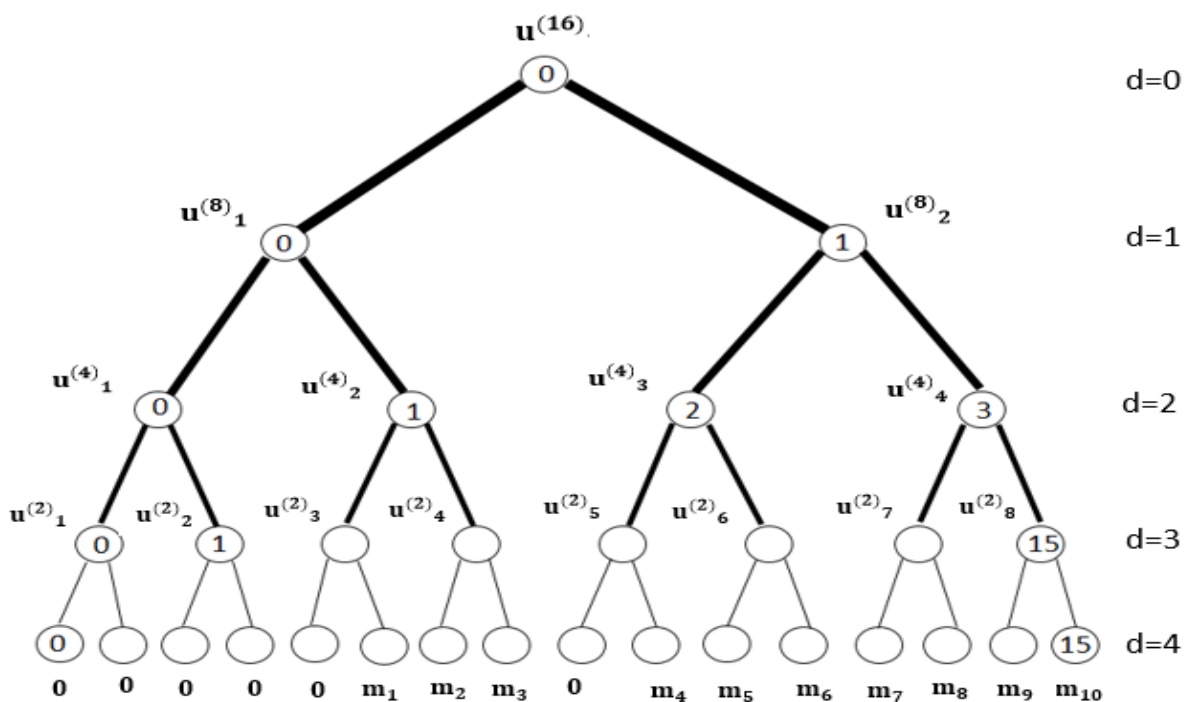
- e) Savoir, quels sont les messages à stocker : au fait le décodeur procède de haut en bas et se dirige toujours vers la gauche et revient en arrière pour aller vers la droite.

- Dans son décodage du haut vers le bas, un nœud doit mémoriser des croyances  $L = [L_1 L_2 \dots]$ .
- Dans son décodage du bas vers le haut, un nœud mémorise les estimés ou leur somme.

Prenons l'exemple (figure 3.9) du nœud 0 à  $d = 1$ . Ce dernier reçoit huit (08) croyances qu'il mémorise et traite pour envoyer quatre (04) au nœud 0 à  $d = 2$ .

Le nœud (0,1) doit garder en mémoire les huit (08) croyances car, l'enfant de gauche lui retournera une décision, cette dernière, avec les huit (08) croyances, sera utilisée pour le calcul de la fonction  $g$  de l'enfant de droite du nœud (0,1). Une fois que les enfants de gauche et de droite aient renvoyés leurs décisions respectives, le nœud en question peut effacer sa mémoire et sera relaxé.

Donc à chaque niveau de profondeur, des messages sont stockés pour un certain temps.

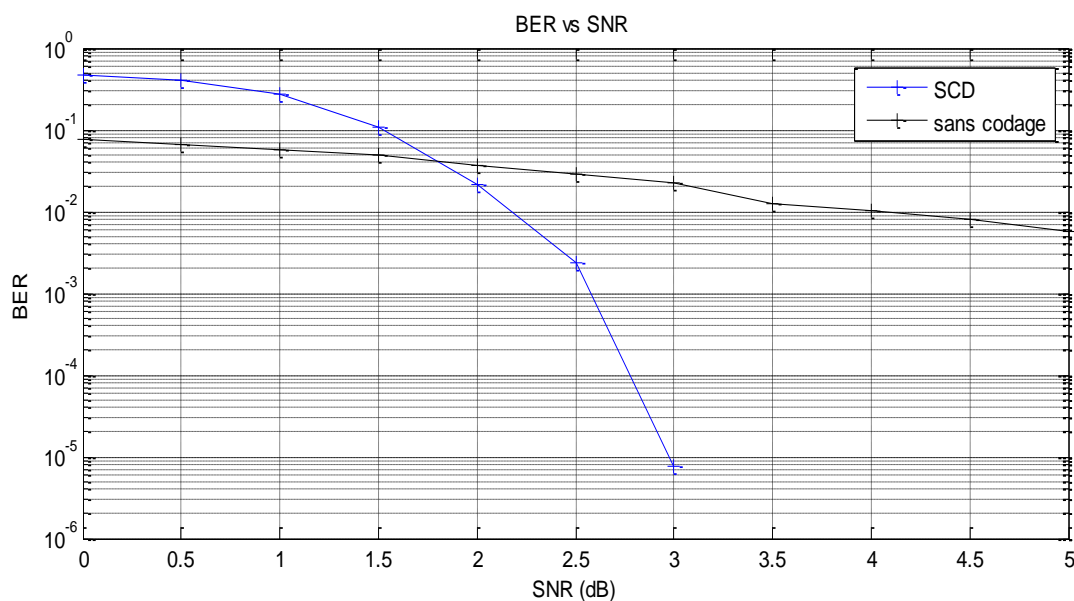


**Figure 3.8** : Ordonnancement des profondeurs et nœuds dans un code CP (10,16) [19]

f) Connaître et sauvegarder l'état du nœud :

- En attente d'être activé (0)
- A fini l'étape L (1)
- A fini l'étape R (2)
- A fini l'étape 4 (3)

Avec toutes ces mesures prises nous pouvons écrire notre programme. Le meilleur résultat que nous pouvons présenter est les performances de cette technique du point de vue BER (*bit error rate*) en fonction du SNR (*signal to noise ratio*) et comparer les résultats avec d'autres techniques de codage.



**Figure 3.9:** Simulation d'un code polaire CP(1024,500) avec un décodage SCD dans un canal AWGN sous BPSK comparé à un schéma sans codage.

### 3.5.2 Décodage par SCLD (*Sucessive Cancellation List Decoder*)

La technique SCD (*sucessive cancellation decoder*), en la comparant avec d'autres codes de même taille reste pauvre en performances et est approximativement à 1 dB ou moins de plus [20]. Des efforts ont été déployés ces dernières années pour améliorer les performances. Le décodage par liste est parmi les meilleurs. A L'opposé du SCD, le SCLD délivre une liste de mot de codes possible.

Question : en présence de plusieurs mots de code (*codeword*) délivrés par le SCLD, quel est celui qu'il faut choisir ?

La technique liste existe depuis les années 40 et 50 et est adaptée au techniques de décodage séquentiel tels les codes polaires mais difficilement adaptée aux LDPC (*low density parity check*) dont le décodage se fait de manière parallèle.

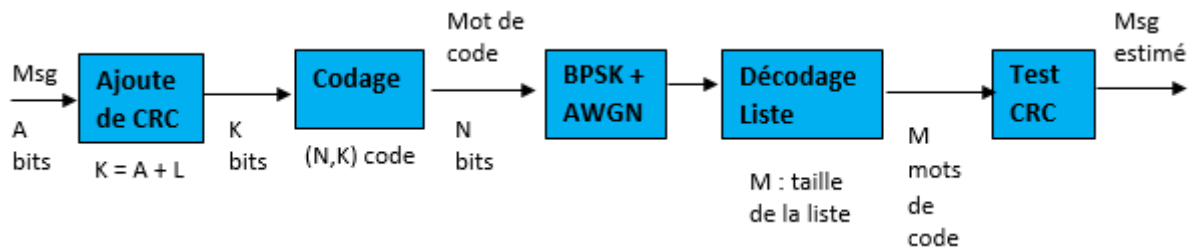
La réponse se trouve dans l'utilisation de la CRC (*cyclic redundancy check*) qui usuellement est utilisée par la couche liaison de données (*data link layer*) du système OSI (*open systems interconnection*). Si une erreur est trouvée, les couches supérieures demanderont une retransmission des mots erronés via la technique ARQ (*automatique repeat request*) entre autre.

Dans le SCLD, la CRC est utilisée dans la couche physique et aura pour rôle de décider lequel des M mots de code de la liste sera le bon et sélectionné. Donc, un mot de code, parmi M, est valide s'il passe par la CRC. Enfin, des situations ou aucun mot de code ne passe est très rare. Si tous passent, des métriques de décision (*decision metric*) seront appliquées pour choisir le meilleur candidat.

Le prix à payer est que la SCLD ajoute de la complexité dans le traitement de décodage puisque, comme nous allons le voir, plusieurs décodeurs seront exécutés.

Enfin, on s'attend que les décodeurs de la 5G (*fifth generation technology standard for cellular networks*) soient du type SCLD.

Le schéma bloc d'un décodeur SCLD est présenté en figure 3.11



**Figure 3.10** : Schéma bloc d'un décodeur SCLD [20]

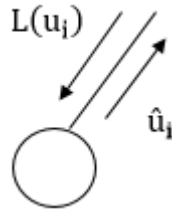
- la taille du message en entrée est  $A$ .
- La CRC ajoute de la redondance pour arriver à message de taille  $K = A + L$ .
- l'encodeur  $CP(N, K)$  fournit un mot de code de  $N$  bits.
- la combinaison BPSK/AWGN fournit également un mot de  $N$  valeurs réelles.
- le SCLD fournit  $M$  mots de codes.
- ces mots de codes passeront par le test CRC pour obtenir, précisons-le, une estimation du message à laquelle est attachée une estimation de la CRC.
- un seul mot de code candidat passera par la CRC.

**Principe de fonctionnement du SCLD:** dans le SCD, la décision est unique :

- si le bit est gelé, le choix se fait pour 0 ( $\hat{u}_i = 0$ ).
- si le bit est celui du message,  $\hat{u}_i = 0$  ou 1.

Au fait, ce choix peut être erroné et ce n'importe quelle position et non seulement au premier bit décodé.

Dans le SCLD, dans de telles situations, ce dernier prend deux (02) décisions 0 ou 1 et assigne des métriques pour chacune d'elle :



**Figure 3.11:** Décision au niveau d'une feuille d'un arbre binaire [20]

Il existe deux types de métrique : la métrique de décision (*decision metric*) et la métrique de chemin (*path metric*) :

- **Métrique de décision :**

Si  $L(u_i) \geq 0$  :

$\hat{u}_i = 0$  a une métrique de décision  $DM_i = 0$

$\hat{u}_i = 1$  a une métrique de décision  $DM_i = |L(u_i)|$ , opposée à la décision (croyance) ; c'est une pénalité égale à la croyance pour le bit à estimer  $\hat{u}_i$ .

- si  $L(u_i) < 0$  :

$\hat{u}_i = 1$  a une métrique de décision  $DM_i = 0$

$\hat{u}_i = 0$  a une métrique de décision  $DM_i = |L(u_i)|$ , opposée à la décision (croyance); c'est une pénalité égale à la croyance pour le bit à estimer  $\hat{u}_i$ .

Nous remarquons qu'il y a deux chemins (*paths*) pour chaque décision. Chaque chemin correspond à un mot de code candidat.

- si le bit est gelé (*frozen*) et que sa croyance est négative ( $L(u_i) < 0$ ), dans ce cas, il faut lui assigner une métrique  $DM_i = |L(u_i)|$ . Au fait, les bits gelés sont placés, rappelons-le, dans les canaux bits les moins fiables d'où possibilité d'erreurs.





Si le bit est gelé aucune décision (scission) n'est prise. Il suffit de faire une mise à jour de la  $L(u_i)$  si  $L(u_i) < 0$ , c-a-d, changer la valeur de la métrique de chemin en lui ajoutant  $|L(u_i)|$ .

Si le bit est non gelé, une scission (*splitting*) a lieu et deux décisions sont prise 0 ou 1 pondérées respectivement par des métriques de décisions  $DM1(0)$  et  $DM1(1)$ . Les métriques de trajets sont :

$$PM1(0) = DM1(0) + PM = DM1(0) + 0 = DM1(0) \quad (3.25)$$

et

$$PM1(1) = DM1(1) + PM = DM1(1) + 0 = DM1(1) \quad (3.26)$$

Toutes les informations sauvegardées concernant le décodeur 1 (voir § 3.4.1 points i, ii ...v) doivent être copiées et transférées vers le décodeur 2.

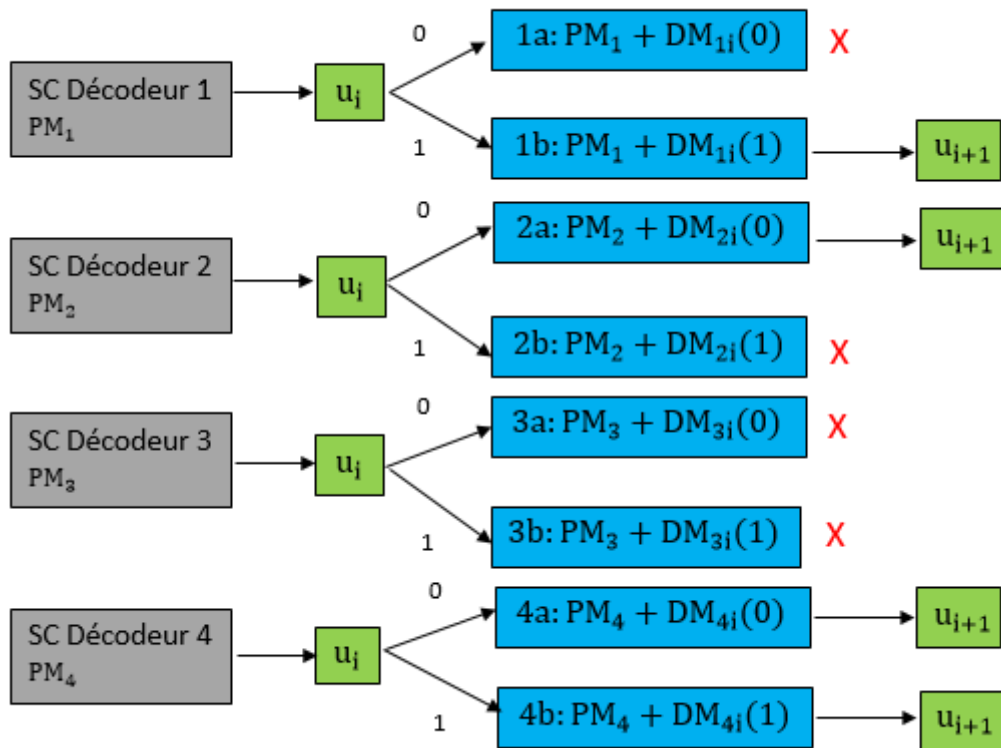
Même chose pour l'estimation de  $u_2$ , deux nouvelles scissions on lieux générant deux (02) nouveaux décodeurs SCD3 et SCD4 avec quatre (04) nouvelles métriques de trajets :  $DM1(0) + DM12(0)$ ,  $DM1(0) + DM12(1)$ ,  $DM1(1) + DM22(0)$  et  $DM1(1) + DM22(1)$ .

Encore une fois des copies auront lieu de SCD1 vers SCD2 et de SCD3 vers SCD4.

Qu'en est-il pour l'estimation de  $u_3$ ? Logiquement huit (08) nouveaux chemin seront créés mais que la liste est fixée à  $L = 4$  alors les quatre (04) chemins les moins fiables ayant donc les plus grandes pénalités ou métriques sont rejetés. Ainsi fonctionne le SCLD.

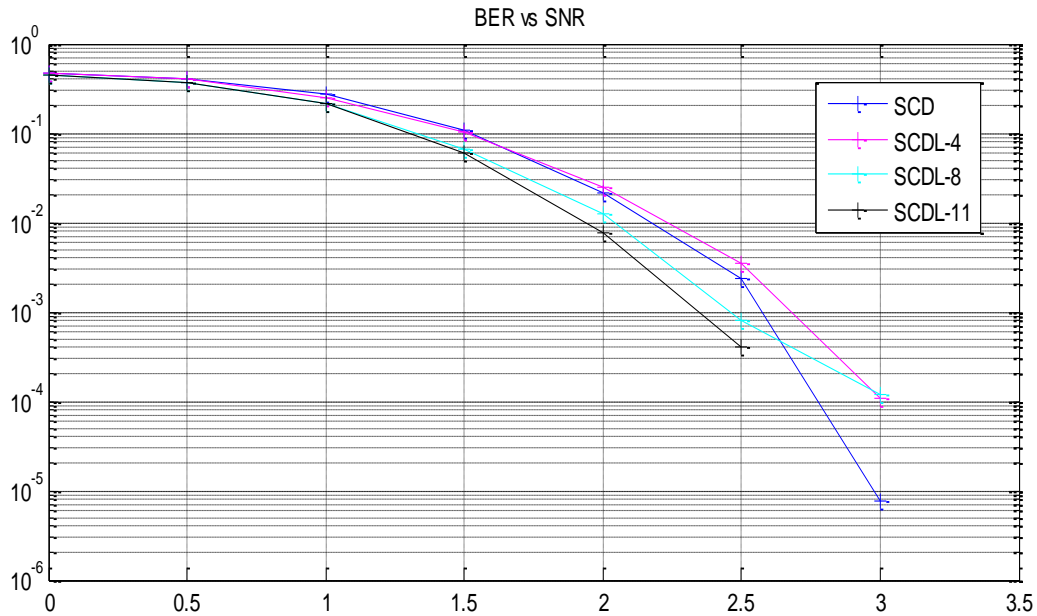
Donc pour un nœud particulier  $i$ , nous avons quatre (04) SCD(s) tel montré dans la figure 3.13, quatre (04) métriques de chemin  $PM1$ ,  $PM2$ ,  $PM3$  et  $PM4$ . Ensuite la décision sur  $u_i$  (suppose non gelé) : 0 ou 1. Nous obtenons ainsi huit (08) métriques de chemin dont quatre seront éliminées ( $L = 4$ ) Enfin des copies seront effectués de

chacun des SCD initiaux vers les SCD subsistants. Le décodage continuera ainsi avec les quatre (04) décodeurs survivants.



**Figure 3.13** : Organigramme d'un décodeur SCLD de taille  $L = 4$  à un nœud particulier [20]

Dans la figure 3.14 sont représentés les courbes du BER vs SNR dans le cas du SCD et SCLD-4, SCLD-8 et SCLD-11 et ce dans un canal AWGN avec une modulation BPSK. On remarque que plus  $L$  augmente et moins de puissance est injecté pour le signal à émettre par rapport au cas du SCD seul. Par exemple, pour un BER de  $10^{-3}$ , le SNR diminue de pratiquement 0.35 dB, c'est-à-dire que pour obtenir un BER de  $10^{-3}$  le SDL-11 en SNR de  $-0.35$  dB par rapport au SCD.



**Figure 3.14 :** BER vs SNR dans le cas de décodage SCD, SCDL-4, SCDL-8 et SCDL-11 pour un CP(500,1024) dans un canal AWGN sous modulation BPSK.

### 3.6 Conclusion

En conclusion à ce chapitre, nous pouvons exprimer notre satisfaction quant au bon déroulement des programme des simulations, néanmoins, nous aurions aimé faire plus de simulation en utilisant des modulations multiniveaux tel la QAM (*quadrature amplitude modulation*) dans une combinaison canal de Rayleigh et AWGN pour ainsi approcher de cas réels et de comparer les performances des codes polaires par rapport à celles des LDPC dans les mêmes conditions de simulations.

## Conclusion Générale

Pour avoir une bonne marche de la communication numérique, il faut bien savoir choisir les types de canaux, les modèles de propagations ainsi que les types de modulations à utiliser. Comme nous avons vu dans notre travail, le fait d'avoir utilisé le canal AWGN nous a permis de travailler sur la même densité de spectre pour toutes les fréquences, et que quel que soit les positions des bits, leur probabilités restent les mêmes et que les erreurs sont indépendantes les uns de autres. Nous avons choisi d'utiliser la modulation BPSK parce qu'elle est la plus simple et qui garantit la précision de la non perte d'informations.

Après avoir utilisé les codes polaires CP (1024,500), vu leur faible complexité d'encodage et de décodage, l'arbre binaire a pu être réalisé lors de l'encodage. Les deux décodages utilisés : SCD et SCLD nous ont permis tous à avoir des résultats. Ce qui nous a autorisés de confirmer ce qui est déjà dit dans l'état de l'art, que pour un code de petite ou moyenne taille  $N < 8192$ , l'utilisation du décodage SCDL favorise les meilleures performances par rapport au décodage SDC qui entraîne un effet de performances de correction médiocres. Mais pour  $N > 8192$ , les meilleures performances sont garanties en utilisant le SCD.

Enfin, malgré que nous ayons rencontré beaucoup de problèmes, notre simulation a bien déroulé comme nous l'avions souhaité. Les codes polaires sont vraiment les codes qu'il faut utiliser pour les réseaux 5G afin de répondre à toutes leurs exigences et réaliser des meilleures technologies constituées par les objets intelligents.

Nous envisageons de faire une amélioration dans l'avenir proche en utilisant d'autre types de modulations et un autre type de canal tel le canal de Rayleigh...

## Références Bibliographiques

- [1] **Andriyanova, I.** (2012). Introduction aux Communications Numériques Master M1 ISIM. Université de Cergy-Pontoise. France
- [2] **Rhouma, R.** (2015). Théorie de l'information Chap : Codage canal et codes de bloc linéaires. École Supérieure d'Économie Électronique (ESEN). Tunisie
- [3] **Hanafi, A., Benyagoub M.** (2015). Simulation de la chaîne de communication IEEE 802.11a. Thèse de Master. Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent. Algérie
- [4] **Berhault, G.** (2015). Exploration architecturale pour le décodage de codes polaires. Electronique. Université de Bordeaux. France.
- [5] **Azni, M.** (2018). Communications Numériques 1 (UEF 21). Université Abderahmane MIRA BEJAIA. Algérie
- [6] **Bondlet, M.** (2020). Tout savoir sur les fréquences de la 5G. France. Disponible sur : <https://blog.ariase.com/mobile/dossiers/5g-frequences>
- [7] **Radio-club F6KRK, F5NB** (2019). Comment ça marche ? Les transmissions radio-numériques 7 - Le canal de propagation. France. Disponible sur : [https://blog.f6krk.org/wpcontent/uploads/2019/02/Trans\\_num7.pdf](https://blog.f6krk.org/wpcontent/uploads/2019/02/Trans_num7.pdf)
- [8] **Coulon, M.** (2007). Canal de Propagation. Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique et d'Hydraulique et Télécommunication (ENSEEIH). France. Disponible sur : [http://coulon.perso.enseeiht.fr/transp\\_Canal.pdf](http://coulon.perso.enseeiht.fr/transp_Canal.pdf)
- [9] **Bersali, M.** (2012). Technique d'accès multiple en Uplink dans la norme 3GPP-LTE. Exposé. Laboratoire OAE/IEMN, UVHC de Valenciennes. France
- [10] **Baouni, A.** . Estimation du Canal et Elimination des Interférences ICI dans les Communications Mobiles avec Modulation OFDM. Thèse. Université Mentouri Constantine. Constantine. Disponible sur : <https://fr.scribd.com/doc/242265896/Baouni-A-pdf>.
- [11] **Weiss, J.** (1998). Modulation QPSK. Disponible sur : <http://www.rennes.supelec.fr/ren/perso/jweiss/tv/nicam/sld014.htm>

- [12] **Sharetechnote**. Matlab-Communication System. Disponible sur : [https://www.sharetechnote.com/image/Matlab\\_Communication\\_QamProcess\\_07.png](https://www.sharetechnote.com/image/Matlab_Communication_QamProcess_07.png)
- [13] **Leroux, C., Tal, I., Vardy, A., Gross, W. J.** (2011). Architectures matérielles pour le décodage des codes polaires. GRETSI, Bordeaux. France.
- [14] **Léonardon, Mathieu** (2018). Décodage de codes polaires sur des architectures programmables. Electronique. Université de Bordeaux; Ecole polytechnique. France, Canada.
- [15] **Homero, A., Erazo, R.** (2017). Détection précoce à l'aide des codes CRC et des codes polaires pour les communications à faible latence. Ecole de Technologie Supérieure Université de Québec. Canada
- [16] **Thangaraj, A.** (2019). LDPC and Polar Codes in 5G Standard: Channel Polarization, Definition of (N, K) Polar Code and Encoding. Indian Institute of Technology Madras. India. Disponible sur: <https://www.youtube.com/watch?v=1uYEq4ueOok>
- [17] **3GPP TS 38.212 V15.1.1** (2018-04). *Technical Specification*; 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; Multiplexing and channel coding (Release 15)
- [18] **Thangaraj, A.** (2019). Successive Cancellation (SC) Decoder for Polar Codes: illustration of its Building Blocks with N=2,4. Indian Institute of Technology Madras. India. Disponible sur: <https://www.youtube.com/watch?v=wK2KI2LtdQI>
- [19] **Thangaraj, A.** (2019). MATLAB Implementation of Success Cancellation Decoder Part 1. Indian Institute of Technology Madras. India. Disponible sur: [https://www.youtube.com/watch?v=D\\_gadv-V-MQ](https://www.youtube.com/watch?v=D_gadv-V-MQ)
- [20] **Thangaraj, A.** (2019). Successive Cancellation List Decoding. Indian Institute of Technology Madras. India. Disponible sur: <https://www.youtube.com/watch?v=WbC5Ux5Pjp8>