REPUBLIQE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

Faculté de Technologie

Mémoire de fin d'études de Master

Filière : Télécommunication

Spécialité : Systèmes des télécommunications

Thème :

Etude et simulation de l'effet des paramètres d'entrelacement au sein des codes turbo dans un canal de transmission sans fil

Réalisé par :

RABAH BELHAOUARI Riham
CHAOUCH Roweida

Dirigé par :

Mr. BAILICHE Mohamed Amine.
Mr.KHELFI Tahar.

Promotion :2023/2024

Remerciement

Nous tenons tout d’abord à remercier ALLAH le tout puissant et miséricordieux, qui nous a donné la force et la patience d’accomplir ce Modeste travail.
En second lieu, nous tenons à remercier tout le corps enseignant qui a contribué à notre formation, plus particulièrement notre encadreur Mr.BAILICHE Mohamed Amine et notre co-encadreur KHELFI Tahar pour leurs aides et les précieux conseils qu’ils ont bien voulu nous donner.
Nos remerciements s’étendent également aux membres du jury pour l’intérêt qu’ils ont porté à notre mémoire en acceptant d’examiner notre travail.

Dédicace

Je dédie ce modeste travail

A mes très chers parents, pour leurs patience, leurs soutien et Leurs confiance ; mon père CHAOUCH Hamid qui s'est sacrifié afin que rien n'entrave le déroulement de mes études,

ma mère Faiza qui m'encourager, et me soutenir dans les moments difficiles ;

qui a sa su m'entourer de tout son amour et son affectation pour que je ne puisse manquer de rien.

A mes très chères sœurs Hanane Amira Maroua et Anfel et mon frères Hichem. A toute ma famille.

A mon binôme Mrs. RABEH BELHAOUARI Riham et toute sa famille.

A tous mes enseignants sans exception.

A mes collègues de l'université de Médéa.

A tous les étudiants de l'université de Médéa on particulier les étudiants de télécommunication.

A tous ceux qui me sont chères, en particulier :

YOUCEF,ISMAIL ,ABDELLEH,RANIA,AMEL ,YASMINE

Roweida

Dédicace

Je dédie ce modeste travail

A mes très chers parents, pour leurs patience, leurs soutien et Leurs confiance ; mon père RABEH BELHAOUARI Rabah qui s'est sacrifié afin que rien n'entrave le déroulement de mes études,

ma mère Nadjia qui m'encourager, et me soutenir dans les moments difficiles ;

qui a sa su m'entourer de tout son amour et son affectation pour que je ne puisse manquer de rien.

A mes très chères sœurs Meriem Ritadj et Nansi et mes frères Mourad et Fares. A toute ma famille.

A mon binôme Mrs. CHAOUCH Roweida et toute sa famille.

A tous mes enseignants sans exception.

A mes collègues de l'université de Médéa.

A tous les étudiants de l'université de Médéa on particulier les étudiants de télécommunication.

A tous ceux qui me sont chères, en particulier :

YOUCEF,ISMAIL ,ABDELLEH,RANIA,AMEL ,YASMINE

Riham

ملخص :

الهدف من عملنا هو إجراء دراسة مقارنة لتأثير الانتشار والتشتت على أداء رموز التوربو. وينصب الاهتمام على تحسين تصحيح الأخطاء في أنظمة الاتصالات الرقمية.

يتم إجراء التشفير التوربيني باستخدام سلسلة متوازية من اثنين من أجهزة التشفير التلافيفية العودية والمنهجية من خلال التشذير. تعد خوارزمية MAP المستخدمة لفك رموز Turbo معقدة للغاية. يتم استخدام تبسيط MAP الذي يحمل اسم Log-MAP في عملنا. كنا مهتمين بفك التشفير التكراري.

يستخدم جهاز فك التشفير Turbo خوارزمية Log-MAP بأطوال إطارات مختلفة (64,128)، ومعدل تشفير قدره 3/1 ويستخدم تشذير QPP.

الكلمات المفتاحية: التشذير، الكود التوربيني، الرموز التلافيفية، الكود المنهجي العودي، وحدة فك التشفير التكراري، MAP، BER، BBAG، BPSK، SNR

Résumé

Le but de notre travail est de faire une étude comparative sur l'influence de l'étalement et de la dispersion sur la performance des Turbo Codes. L'interet est d'améliorer la correction des erreurs dans les systèmes de communication numériques.

Le codage turbo se fait à l'aide d'une concaténation parallèle de deux codeurs convolutionnels récursifs et systématiques à travers un entrelaceur. L'algorithme MAP qui est utilisé pour le décodage des Codes Turbo est très compliqué. Une simplification du MAP qui porte le nom du Log-MAP est utilisée dans notre travail. On s'est intéresséspar le décodage itératif.

Le décodeur Turbo utilise l'algorithme Log-MAP avec différentes longueurs de trame(64,128), un taux de codage de 1/3 et utilise l'entrelaceur QPP.

Mot Clés : L'entrelacement, code turbo, codes convolutifs, code systématique récursif, décodeur itératif, MAP, BER, BBAG, BPSK, SNR.

Summary:

The aim of our work is to carry out a comparative study on the influence of spreading and dispersion on the performance of Turbo Codes. The interest is to improve error correction in digital communication systems.

Turbo coding is done using a parallel concatenation of two recursive and systematic convolutional encoders through an interleaver. The MAP algorithm which is used for decoding Turbo Codes is very complicated. A simplification of the MAP which bears the name Log-MAP is used in our work. We were interested in iterative decoding.

The Turbo decoder uses the Log-MAP algorithm with different frame lengths (64,128), coding rate of 1/3 and uses the QPP interleaver.

Keywords: Interleaving, turbo code, convolutional codes, recursive systematic code, iterative decoder, MAP, BER, AWGN, BPSK, SNR.

Table des matières

Résumé.....I

Table des matiéres.....II

Liste des figures…..III

Liste des Tableaux.....IV

Liste Des Abréviations et symboles.....V

Introduction générale.....1

Chapitre I Généralités dur le système de communication numérique

I.1 Introduction.....3

I.2 Généralité:.....3

I.2.1 La chaine de transmission.....3

I.2.2 Le codage Canal.....4

I.2.3 La théorème de Shannon sur le codage de canal.....5

I.2.4 Canal de transmission.....6

I.2.5 Les types des canaux.....7

I.2.5.1 Le Canal à bruit blanc additif Gaussien BBAG.....7

I.2.5.2 Le canal de Rayleigh.....9

I.2.6. La modulation.....9

I.7 Conclusion.....13

Chapitre II TURBO CODE

II.1 Introduction:.....14

II.2 Généralités sur les codes convolutionnels:.....14

II.2.1 Le principe de base.....16

II.2.2 Type de codeur.....17

II.2.2.1 code convolutionnel systématique:.....17

II.2.2.2 Code convolutionnel récursif:.....18

II.2.2.3 Code convolutionnel récursif et systématique.....18

II.3 Représentation graphique des codes convolutionnels.....19

II.3.2 Diagramme de treillis.....20

II.4. Décodage convolutif.....21

II.4.1 Hard décision (Algorithme Viterbi):.....21

II.4.2 Soft décision (algorithme MAP).....22

II.4.3 Versions simplifiées de l'algorithme Map.....24

II.5 Turbo code 25
II.6 Décodage itératif 27
II.6.1 Principe de décodage itératif 27
II.6.2 Décodeur turbo 28
II.7 Conclusion: 29

Chapitre III Les entrelaceurs

III.1 Introduction 30
III.2 Définition de l'entrelaceur 30
III.3 Principe des entrelaceurs 31
III.4 Méthode de l'entrelacement 31
III.4.1 Entrelacement par bloc 32
III.4.2 Entrelaceur déterministe 32
III.4.3 entrelaceur étalé (S-Random interleaver) 33
III.4.4 Entrelaceur aléatoire (random interleaver) 34
III.5 Programme de Paiement de la Qualité QPP 34
III.5.1 LTE Normalisation 36
III.6 UMTS: Universal Mobile Telecommunications System 38
III.6.1 Entrelaceur interne du turbo code 38
III.6.2 Génération de l'entrelaceur mère 38
III.7 Paramètre de l'entrelacement 40
III.7.1 Le délai 40
III.7.2 Mémoire: 41
III.7.3 Le facteur de dispersion (dispersion interleaver) 41
III.7.4 L'étalement de l'entrelaceur (spreading interleaver) 42
III.8 Advantage de l'entrelaceur 43
III.9 Conclusion 43

Chapitre IV Programmation et discutions des resultats

IV. 1 Introduction *45*
IV.2 Schéma de bloc *45*
IV.3 Schéma de Turbo codeur 46
IV.4 Schéma de code convolutif 46
IV.5 *Schéma du turbo décodeur* 47
IV.6 Méthodologie de simulation *47*

IV.7 *Résultats et discussion* 48
IV.8 *Conclusion* 65
Conclusion générale 66
References bibilioographiques

LISTE DES FIGURES :

Figure1.1 : Principe d'une chaine de communication numérique 4

Figure 1.2 : Schéma simplifié du codeur et décodeur..... 5

Figure 1.3 : Canal à bruit additif 8

Figure 1.4 : Modèle de canal de Rayleigh..... 9

Figure1.5 : Forme d'un signal modulée BPSK..... 10

Figure 1.6: Comparaison du BER de la modulation avec différents ordre de modulation ... 12

Figure1.7: Constellation Diagramme du signal BPSK..... 12

Figure 1.8 :BPSK Modulator 13

Figure 2.1 : Principe d'un codeur convolutionnel 16

Figure 2.2 : Codeur convolutionnel de taux de codage R=1/2..... 17

Figure 2.3 : Codeur convolutionnel récursif systématique 19

Figure 2.3.1 : Diagramme d'état..... 20

Figure 2.3.2 : Diagramme de treillis..... 21

Figure 2.3.3 : Algorithme Viterbi 21

Figure 2.4: Algorithme MAP 22

Figure 2.5 : Schéma d'un turbo code..... 26

Figure 2.6 : Shéma de décodage itératif..... 27

Figure 2.6.1 : Shéma d'un décodeur turbo..... 28

Figure 3.1 : Entrelaceur 31

Figure 3.2 : Entrelaceur bloc de taille 3×3..... 32

Figure 3.3: Principe d'un entrelaceur déterministe..... 33

Figure3.4 : Mode d'opération pour réordonner des symboles..... 33

Figure 4.1: Schéma de bloc..... 45

Figure 4.2: Schéma turbo code..... 46

Figure 4.3: Schéma de code convolutif 46

Figure 4.4: Schéma turbo décodage 47

Figure 4.5: BER en fonction de Eb/No(dB) S=4..... 49

Figure 4.6:BER en fonction de Eb/No(dB) S=6 51
Figure 4.7:BER en fonction de Eb/No(dB) S=4 S=6 S=10 53
Figure 4.8: BER en fonction de Eb/No(dB) S=4 S=10 55
Figure 4.9: BER en fonction de Eb/No(dB) S=4 57
Figure 4.10: BER en fonction de Eb/No(dB) S=8 59
Figure 4.11: BER en fonction de Eb/No(dB) S=12 61
Figure 4.12: BER en fonction de Eb/No(dB) S=4 S=8 63
Figure 4.13 : BER en fonction de Eb/No(dB) S=4 S=14 65

LISTE DEX TABLEAUX

Tableau 1.1 : Limite théorique de l'efficacité spectrale de modulation….....11

Tableau 3. 1 : Paramètre de l'entrelaceur interne de turbo code… 37

Tableau 4.1 : Les valeurs de BER pour S=4 49

Tableau 4.2 : Les valeurs de BER pour S=6 50

Tableau 4.3 : Les valeurs de BER pour déférent étalement (spreading).....52

Tableau 4.4 : Les valeurs de BER pour déférent étalement (dispersion).....54

Tableau 4.5 : Les valeurs de BER pour S=4 dans trame 128 56

Tableau 4.6 : Les valeurs de BER pour S=8 dans trame 128 58

Tableau 4.7 : Les valeurs de BER pour S=12 dans trame 128… 60

Tableau 4.8:Les valeurs de BER pour déférent étalement trame 128 62

Tableau 4.9 : Les valeurs de BER pour déférent étalement dispersion trame 128 64

LISTE DES ABRIVIATIONS ET SYMBOLES:

LISTE DES SYMBOLES :

σ^2 : Variance d'AWGN

n **:** Bruit gaussien suit une distribution gaussienne

H : l'entropie

L : la longueur moyen de codage

C : Capacité de canal de transmission

B : La bande passante du canal

K : Longueur de contrainte d'un code convolutionnel

SNR : Le rapport signal sur bruit

E_b**:** Étant l'énergie reçue par symbole binaire d'information transmis

N_0 **:** La densité spectrale de puissance

E_b/N_0 **:** Rapport signal sur bruit

a_K:Enveloppe du signal Rayleigh

Xk **:** Signal émis

Rn : signal sortie

Los : la propagation multi-trajets

S_0**(t) :** Signal BPSK lorsqu'on transmet un 0

S_1**(t) :** Signal BPSK lorsqu'on transmet un 1

T_b(t) **:** Durée de transmission d'un bit

fp**(t):** Fréquence porteuse

E_b**(t) :** Energie par bit du signal transmis

K : Longueur de contrainte d'un code convolutionnel

a_K **:**Métrique d'état en avant de l'état m à l'instant k

Q_K :Métrique d'état en arrière de l'état m à l'instant k

R : Rate de codage

D_K **:** Séquence de symboles

I : Entrelaceur

I^{-}:Désentrelaceur

k_k^s **:** Symbole systématique à la sortie de l'encodeur turbo

k_k^{1p} **:** Premier symbole de parité à la sortie de l'encodeur turbo

k_k^{2p} **:**Deuxième symbole de parité à la sortie de l'encodeur turbo

D_{min} **:** Distance minimale

r_k^s **:** Symbole d'information systématique reçu à l'instant k

r_k^{1p} **:** Symbole de parité 1 reçu à l'instant k

r_k^{2p} **:** Symbole de parité 2 reçu à l'instant k

L_a **:** Information a priori à l'instant k

Lex_1 **:** Information extrinsèque de décodeur 1

Lex_2 **:** Information extrinsèque de décodeur 2

LISTE DES ABRIVIATIONS:

APP: A posteriori Brobability.

AWGN: Additive White Gaussian Noise.

BBAG: Bruit Blanc Additif Gaussien.

BER : Bit Error Rate.

BPSK :. Modulation par Décalage de Phase Binaire.

CNS : Convolutif Non Systématique.

CRS : Convolutif Récursif Systématique.

CP: Concaténation Parallèle.

DEC 1 : Décodeur 1

DEC 2: Décodeur 2

FER : Fram Error Rate.

LLR : Logarithme du rapport de vraisemblance

Los : Line de vue

MAP: Maximum A posteriori.

NRNSC : Non Récursif Non Systématique Code.

PGCD : Plus Grand Commun Diviseur

Qpp : Programme de Paiement de la Qualité

SNR : Rapport Signal sur Bruit.

SOVA: Soft Output Viterbi Algorithme.

TEB : Taux d'Erreur Binaire.

Introduction Générale

Introduction générale

Les codes de correction d'erreurs sont utilisés dans les systèmes de communication numérique pour rectifier les erreurs pouvant survenir lors de la transmission d'informations. Parmi ces codes, les turbo codes, se distinguent par leur capacité à atteindre un faible taux d'erreur tout en consommant peu d'énergie.

Lors de l'émission, les turbo codes encodent le message en parallèle au moins deux fois. Sur l'une de ces voies, le message est codé dans sa forme originale, tandis que sur les autres il est préalablement modifié par un entrelaceur avant d'être encodé. L'entrelaceur, un composant clé des encodeurs turbo, réorganise la position des symboles d'information sur chaque voie de codage. En d'autres termes, il effectue une permutation des symboles d'information. Cette permutation peut avoir pour effet de protéger les parties consécutives du message contre les rafales d'erreurs, selon les propriétés spécifiques de l'entrelaceur, telles que l'étalement et la dispersion, que nous expliquerons ultérieurement.

Les entrelaceurs aléatoires et S-aléatoires peuvent fonctionner efficacement, mais ils présentent des limitations que les entrelaceurs construits de manière algébrique pourraient ne pas avoir. Tout d'abord, les entrelaceurs aléatoires et S-aléatoires nécessitent une mémoire pour leur stockage, tandis que les permutations fournies par les entrelaceurs algébriques peuvent être générées au moment de l'encodage. De plus, la structure algébrique des entrelaceurs permet une analyse préalable de leurs propriétés, contrairement aux entrelaceurs aléatoires et S-aléatoires dont les propriétés nécessitent souvent des simulations pour être étudiées. Actuellement, les entrelaceurs aléatoires et S-aléatoires sont largement utilisés, faute de constructions algébriques offrant un rendement supérieur.

Les nouvelles applications de communication numérique, telles que le multimédia, exigent des codes correcteurs d'erreurs très performants capables de maintenir des taux d'erreur bas, même dans des conditions de rapport signal sur bruit (SNR) faible à modéré. Les turbo codes, malgré leur complexité raisonnable, peuvent atteindre des taux d'erreur très faibles lorsqu'ils sont associés à un entrelaceur bien conçu. L'utilisation d'entrelaceurs soigneusement élaborés permet d'atteindre des taux d'erreur très bas, particulièrement avec

des entrelaceurs de taille moyenne à grande, où les turbo codes montrent leur potentiel maximal en termes de distance minimale élevée (dmin).

Ce mémoire est divisé en quatre chapitres, dans le plan de travail sera comme suit :

Le premier chapitre est consacré à la présentation des systèmes de communication numérique en général, nous avons situé quelques notions.

Dans le deuxième chapitre, un des paramètres de turbo codes est présenté.

En effet sans l'algorithme de décodage, ces derniers ne sont plus d'excellents correcteurs d erreur.il existe de nombreux algorithme de décodage et un des meilleurs pour les codes maximum a posteriori qui est la base de l'algorithme MAP. En second lieu, nousavons présentés la version de l'algorithme MAP que nous avons adoptée dans notre travail.

Le troisième chapitre, nous abordons également la technique d'entrelacement utilisés en codage turbo. Les paramètres des entrelaceurs ont été étudiés. Nous avons ainsi vu la notion de délai, celle de mémoire, mais surtout le facteur d'étalement et dispersion qui sera une notion primordiale dans notre étude des entrelaceurs pour les Codes Turbo.

Le quatrième chapitre présente la programmation et discussions des résultats de simulation et analyse les performances des turbo codes selon différents paramètres étalement et dispersion

Et en finalement on termine par conclusion générale

Chapitre I Généralité sur le système de communication numérique

I.1 Introduction

Les systèmes de communication numérique sont des infrastructures essentielles dans notre ère de l'information, permettant le transfert efficace et précis de données à travers divers canaux. Dans cette introduction générale, nous allons explorer les principes fondamentaux et les composants clés des systèmes de communication numérique.

Un système de communication numérique est un système qui utilise des signaux numériques pour représenter, transmettre et traiter l'information. Contrairement aux systèmes analogiques, qui utilisent des signaux continus, les systèmes numériques représentent l'information sous forme de séquences de bits binaires (0 et 1).

Les systèmes de communication numérique sont omniprésents dans de nombreuses applications, notamment les réseaux informatiques, les télécommunications, la télévision numérique, la radio numérique, les systèmes de navigation par satellite, les systèmes de contrôle industriel, etc.

En somme, les systèmes de communication numérique jouent un rôle central dans notre société moderne, facilitant l'échange rapide et fiable d'informations à travers le monde.

En résumé, le but principal des systèmes de communication numérique est de fournir une plateforme fiable, efficace et sécurisée pour la transmission et l'échange d'informations à travers divers canaux et médiums, facilitant ainsi la connectivité mondiale et la diffusion de connaissances.

I.2 Généralités

I.2.1 La chaine de transmission

Dans un système de communication, on veut transmettre l'information provenant d'un émetteur vers un récepteur, à travers un canal de transmission Figure 1.1. Ce dernier possède un certain nombre de caractéristiques, notamment sa capacité, sa nature physique, et le bruit qui falsifie l'information. La figure1.1illustre le principe de base d'une transmission numérique de messages avec codage de source et codage de canal. Comme on l'a déjà indiqué, la redondance est d'abord éliminée par le codage de source, puis ajoutée de manière contrôlée par le codage de canal. La redondance éventuellement présente dans les données sources est inutile pour le codage de canal, puisque les caractéristiques de cette redondance ne

sont pas exactement contrôlables. Les données de base peuvent ne pas être redondantes du tout. [1]

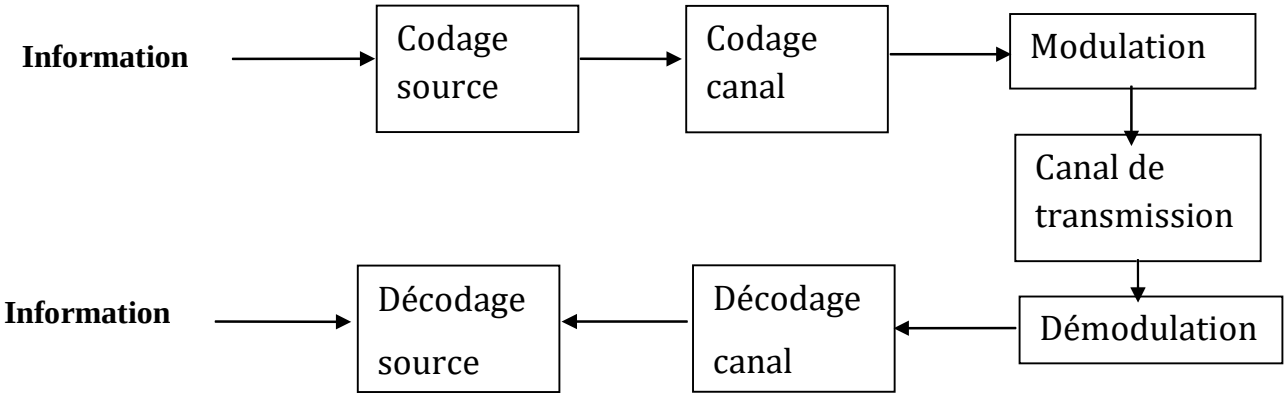


Figure 1. 1 : Principe d'une chaine de communication numérique [1]

I.2.2 Le codage Canal

Le codage de canal est une étape essentielle dans la transmission de données, particulièrement dans les systèmes de communication. Son rôle principal est d'assurer la fiabilité de la transmission en détectant et dans certains cas, en corrigeant les erreurs qui peuvent survenir pendant la transmission des données à travers le canal de communication.

En théorie de l'information, le théorème du codage de canal aussi appelé deuxième théorème de Shannon montre qu'il est possible de transmettre des données numériques sur un canal bruité avec un taux d'erreur arbitrairement faible si le débit est inférieur à une certaine limite propre au canal. [1]

Le "codage canal" dans le contexte des turbos codes fait référence à une technique de correction d'erreurs utilisée dans les communications numériques. Les turbos codes sont une classe de codes correcteurs d'erreurs qui ont une performance exceptionnelle, les rendant très efficaces dans la correction des erreurs de transmission dans les systèmes de communication

Dans un turbo code, le codage canal consiste à encoder les données brutes à transmettre en une séquence de symboles, en utilisant des algorithmes spécifiques. Cette

séquence est ensuite envoyée à travers un canal de communication, qui peut introduire des erreurs, telles que des perturbations, des interférences, ou du bruit.

Une fois que les données sont reçues, le processus de décodage commence. Le décodage est effectué en utilisant des algorithmes itératifs qui exploitent la redondance introduite par le codage canal pour détecter et corriger les erreurs. Dans le cas des turbo codes, deux (ou plus) codes convolutifs sont utilisés en parallèle et en série, et les informations sont échangées entre eux lors du processus itératif de décodage, ce qui améliore considérablement les performances de correction d'erreur.

En résumé, le codage de canal joue un rôle crucial pour assurer une transmission fiable des données à travers des canaux sujets à des perturbations, contribuant ainsi à garantir l'intégrité des informations échangées dans les systèmes de communication.

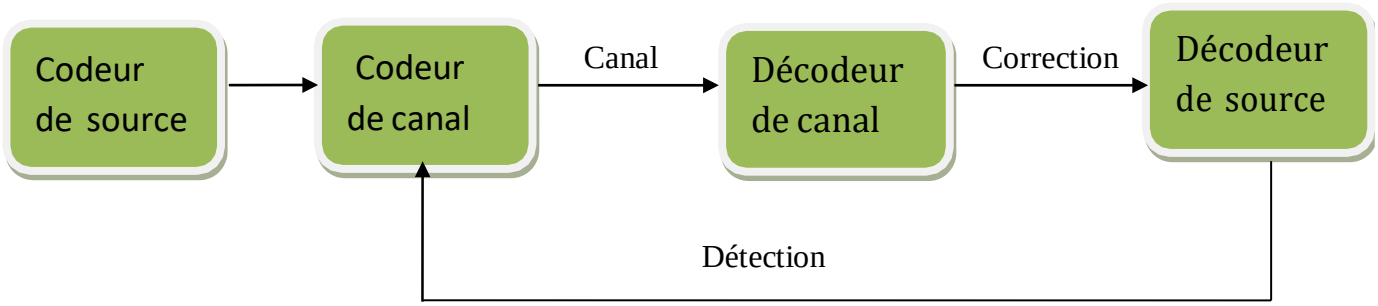


Figure1.2 : Schéma simplifié du codeur et décodeur [3]

Un bon code doit avoir :

- Une bonne capacité de détection et correction d'erreurs.
- Un bon rendement (taux) c'est-à-dire un grand nombre de bits d'information par rapport aux bits codés.
- Une procédure de décodage (et de codage) suffisamment simple et rapide.

I.2.3 Le théorème de Shannon sur le codage de canal

Ou théorème de Shannon, énonce les limites fondamentales de la communication de données sur un canal sujet au bruit. Formulé par Claude Shannon en 1948 dans son article révolutionnaire intitulé "A Mathematical Theory of Communication", ce théorème est l'un des résultats les plus importants dans le domaine de la théorie de l'information.

Limite du deuxième théorème de Shannon

La limite fondamentale énoncée par le deuxième théorème de Shannon est que pour une source d'information donnée avec une entropie H (mesurée en bits par symbole), la longueur moyenne du codage ne peut pas être réduite en dessous de *H*. En d'autres termes, il est impossible de compresser efficacement une source d'information au-delà de son entropie sans perdre d'information.

Définition

Le deuxième théorème de Shannon stipule que dans un système de communication, la longueur moyenne *L* du code de source nécessaire pour représenter les symboles d'une source d'information est liée à l'entropie de cette source. Plus précisément, si une source d'information a une entropie H, alors il existe un code de longueur moyenne par symbole *L* tel que $L \geq H$

Loi associée

La loi fondamentale associée au deuxième théorème de Shannon est que la longueur moyenne du codage *L* est liée à l'entropie H de la source d'information par l'inégalité $L \geq H$

I.2.4 Canal de transmission

Le canal de transmission fait référence au moyen par lequel les informations sont envoyées d'un point à un autre.

Dans le contexte des turbos codes, le "canal de transmission" fait référence à la voie par laquelle les données sont envoyées à travers un canal de communication. Les turbo codes sont une technique de codage de canal utilisée dans les communications numériques pour corriger les erreurs qui peuvent survenir lors de la transmission de données à travers des canaux de communication imparfaits, tels que les canaux sans fil ou les canaux avec du bruit.

Le processus de codage en turbo consiste généralement en l'utilisation de deux (ou plus) codes convolutifs en parallèle, avec un schéma de permutation entre les deux pour créer une redondance supplémentaire et améliorer la capacité de correction d'erreurs. Le canal de transmission dans le contexte des turbos codes se réfère donc à la voie à travers laquelle les données encodées avec ce schéma sont envoyées. Cela peut être une transmission sans fil, une transmission sur un câble, ou tout autre canal de communication utilisé dans les systèmes de communication numérique.

La capacité du canal de transmission, souvent appelée capacité de Shannon, est une mesure fondamentale dans la théorie de l'information. Elle représente le taux maximal auquel des informations peuvent être transmises de manière fiable à travers un canal de communication, en présence de bruit. La capacité est déterminée par les propriétés du canal lui-même, telles que sa bande passante, sa puissance et le niveau de bruit

La capacité C d'un canal de transmission est donnée par la formule de Shannon :

$$C=B\times log_2(1+\text{SNR}) \qquad (1.1)$$

- C: la capacité du canal en bits par seconde (bps)

- B : la bande passante du canal en hertz (Hz),

- SNR : le rapport signal sur bruit (Signal-to-Noise Ratio) (Rapport Signal sur Bruit) du canal, exprimé comme une valeur sans unité

$$\text{SNR}=\frac{E_b}{N_0}\ (\text{dB}) \qquad (1.2)$$

- E_b *:* étant l'énergie reçue par symbole binaire d'information transmis

- N_0 *:* la densité spectrale de puissance du bruit contenue dans la bande

I.2.5 Les types des canaux

I.2.5.1 Le Canal à bruit blanc additif Gaussien BBAG

Le modèle du bruit additif blanc et gaussien est le plus utilisé en théorie. Il fournit un modèle presque parfait pour certains systèmes de communication. Le canal à bruit blanc additif gaussien (BBAG) modélise une transmission en bande de base sur un canal idéal et d'un bruit gaussien .Ce modèle implique que le bruit du canal est une variable aléatoire ***n*** qui s'additionne au signal modulé. Dans ce cas, la variable ***n*** est gaussien, de moyenne nulle et de variance ***σ2*** *(*figure 1.3) [5]

$$\boldsymbol{\sigma 2 = \frac{N_0}{2}} \qquad \textbf{(1. 3)}$$

Si on considère la variable modulée Xk, il est possible de donner la sortie du canal comme : □□ = □□ + □ **(1.4)**

Le modèle de bruit additif blanc Gaussien est très pratique, pour les calculs théoriques. Il reflète également très bien la réalité dans les systèmes de communications.

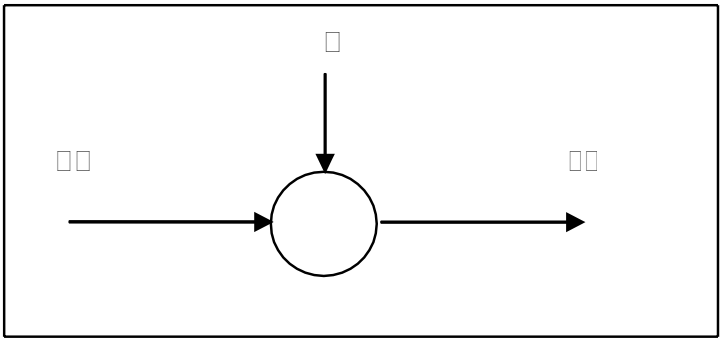


Figure 1.3 : canal à bruit additif [5].

□□ : Signal émis

□: Bruit gaussien suit une distribution gaussienne, caractérisée par une moyenne μ et une variance σ^2.

rn : signal sortie

Mathématiquement, nous pouvons écrire la densité de probabilité du bruit additif blanc gaussien comme:

$$\mathbf{P(x)} = \frac{\mathbf{1}}{\sqrt{\sigma \mathbf{2\pi}}} \boldsymbol{exp} \frac{\boldsymbol{-n}^{\mathbf{2}}}{\mathbf{2}\boldsymbol{N_0}} \qquad \mathbf{(1.5)}$$

$\boldsymbol{\sigma 2} = \frac{\boldsymbol{N_0}}{\mathbf{2}}$**:** dont N_0 est la densité spectrale de puissance mono latérale constante

– l'adjectif ;"additif" dans AWGN signifie que l'impact du bruit sur le signal transmis peut être modélisé comme une variable aléatoire n qui s'additionne au signal modulé.

– Le terme "blanc" indique que la densité spectrale de puissance du bruit est considérée unilatérale (c.à.d. de la fréquence 0 a inf) et constante. [2]

En utilisant le bruit BBAG dans les simulations de turbo codes, on peut analyser comment les performances de décodage varient en fonction du rapport signal sur bruit SNR $\frac{E_b}{N_0}$ et d'autres paramètres de conception du système. Cela aide les concepteurs à optimiser les performances des systèmes de communication dans des environnements réels où le bruit est omniprésent. [1]

Les turbos codes sont des codes correcteurs d'erreurs qui exploitent la redondance pour corriger les erreurs de transmission. Leur performance est souvent évaluée en simulant le processus de transmission avec l'ajout de bruit AWGN. [2]

I.2.5.2 Le canal de Rayleigh

Le canal de Rayleigh est un modèle communément utilisé en théorie de la communication pour représenter les canaux de transmission sans fil (mobile) où les signaux subissent une atténuation multiplicative due à la propagation Line de vue (los). Contrairement au canal BBAG, le canal Rayleigh ne présente pas de bruit additif gaussien. Au lieu de cela, il modélise les variations aléatoires de l'amplitude et de la phase des signaux transmis dus à la superposition de plusieurs trajets de propagation avec des chemins différents et des phases aléatoires. Ce modèle est souvent utilisé pour les communications sans fil (mobile), en particulier dans les environnements urbains ou à l'intérieur des bâtiments, où les signaux se réfléchissent, se diffractent et se dispersent sur divers trajets entre l'émetteur et le récepteur.
Le modèle d'un canal de Rayleigh est schématisé à la figure 1.4, où rn est la version bruitée du signal 2 à l'entrée du canal. Ce modèle est caractérisé par deux paramètres. L'un de ces deux paramètres est une variable aléatoire de distribution gaussienne, et l'autre variable est l'enveloppe du signal ***ak.*** ***[4]***

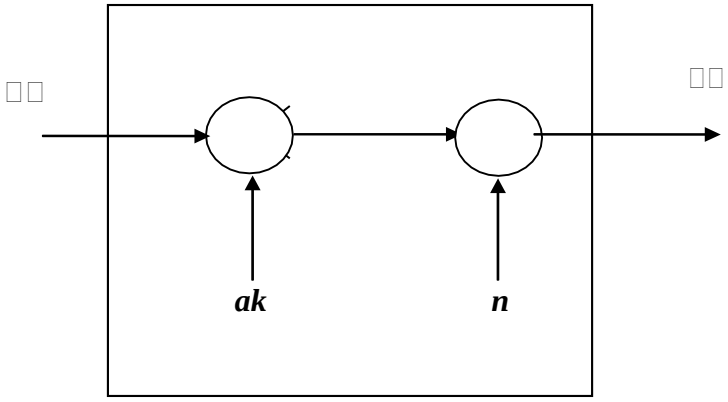


Figure 1.4 : Modèle de canal de Rayleigh [4]

I.2.6. La modulation

La modulation a pour objectif d'adapter le signal à émettre au canal de transmission. Elle utilise une onde porteuse qui sert à transposer les données par modification d'une ou de

plusieurs caractéristique de onde, amplitude (M-ASK), phase (BPSK, QPSK,…etc.), fréquence (M-FSK), amplitude et phase (QAM).

Modulation BPSK « Modulation par Décalage de Phase Binaire »

Dans notre travail, nous avons considéré le cas où la modulation est BPSK et le canal est un canal à bruit gaussien additif BBAG.

La modulation BPSK (Modulation par Décalage de Phase Binaire) est parmi les modulations numériques les plus utilisées. D'après les simulations qu'on a fait, en comparant la modulation BPSK avec autre type de modulation (Figure (1.5)) et la modulation BPSK avec le même type de modulation PSK (Décalage de phase) mais avec des ordres de modulation plus grands (Figure (1.)), en a remarquer qu'elle est la plus performante au niveau de la probabilité d'erreur.

En BPSK, les phases opposées de la porteuse (0 et pi) sont transmises toutes les T secondes (si on considère que la durée d'un bit est T) et sont basées sur les symboles d'information +l où -1. BPSK peut être considéré comme un type de modulation d'amplitude ou la porteuse est multipliée par -1 ou +1 suivant le bit de parité comme le montre la Figure (1.6) où les données + 1 et - 1 sont représentées respectivement par cos (ωct) et cos($\omega ct + \pi$). Nous pouvons représenter ceci sous forme de constellation dans un plan x /y.

BPSK se limite à deux signaux sur l'axe des x tel qu'il est montré à la Figure (1.6) où Eb est l'énergie d'un bit. Contrairement à la structure du modulateur BPSK, montrée à la Figure (1.7), le démodulateur est beaucoup plus complexe. En effet le récepteur BPSK doit fournir une phase de référence cohérente afin de démoduler le signal reçu. [2]

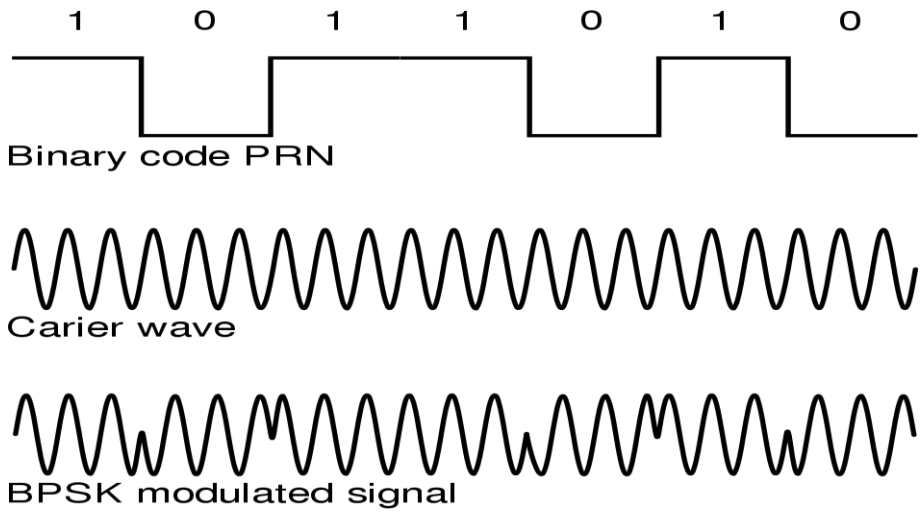


Figure1.5 : Forme d'un signal modulée BPSK

Table1.1 : Limite théorique de l'efficacité spectrale de modulation

Modulation	Limite théorique de l'efficacité spectrale (bit /second /Hz)
MSK	1
BPSK	1
QPSK	2
8 PSK	3
16 QAM	4
32 QAM	5
64 QAM	6
256 QAM	8

Mathématiquement on peut exprimer un signal BPSK de la façon suivante :

$$S_0(\mathrm{t}) = -(\sqrt{2\frac{E_b}{T_b}})\cos 2\pi fpt \qquad 0 \leq t \leq Tb \qquad (1.6)$$

$$S_1(\mathrm{t}) = +(\sqrt{2\frac{E_b}{T_b}})\cos 2\pi fpt \qquad 0 \leq t \leq Tb \qquad (1.7)$$

$\boldsymbol{S_0(t)}$ **:** Signal BPSK lorsqu'on transmet un 0

$\boldsymbol{S_1(t)}$ **:** Signal BPSK lorsqu'on transmet un 1

$\boldsymbol{T_b}(\mathbf{t})$ **:** Durée de transmission d'un bit

$\boldsymbol{fp}$ **(t):** Fréquence porteuse

$\boldsymbol{E_b(t)}$: Energie par bit du signal transmis

Le problème du démodulateur BPSK est de récupérer la bonne phase pour démoduler. Ceci constitue la principale difficulté du récepteur BPSK

Taux d'erreur binaire :

Alors que la qualité d'un signal analogique est dégradée par toute distorsion ou atténuation non linéaire créé par le canal de transmission, la qualité d'un signal numérique ne sera réduite que si les effets négatifs du canal conduisent le récepteur à confondre plusieurs symboles ou bits dans le cas d'un signal binaire. Afin de quantifier la dégradation subie par un signal numérique ou de spécifier la qualité que doit atteindre une transmission

$$\text{BER (bit error rate)} = \frac{\text{Nombre de bits erronées}}{\text{Nombre de bits transmis}} \qquad (1.8)$$

Comme nous l'avons déjà dit, le BER est relié au rapport signal sur bruit par bit E_b/N_0 et aux caractéristiques de la technique de modulation employée. Plus la contrainte sur la qualité de service d'une transmission est élevée, plus le BER est faible.

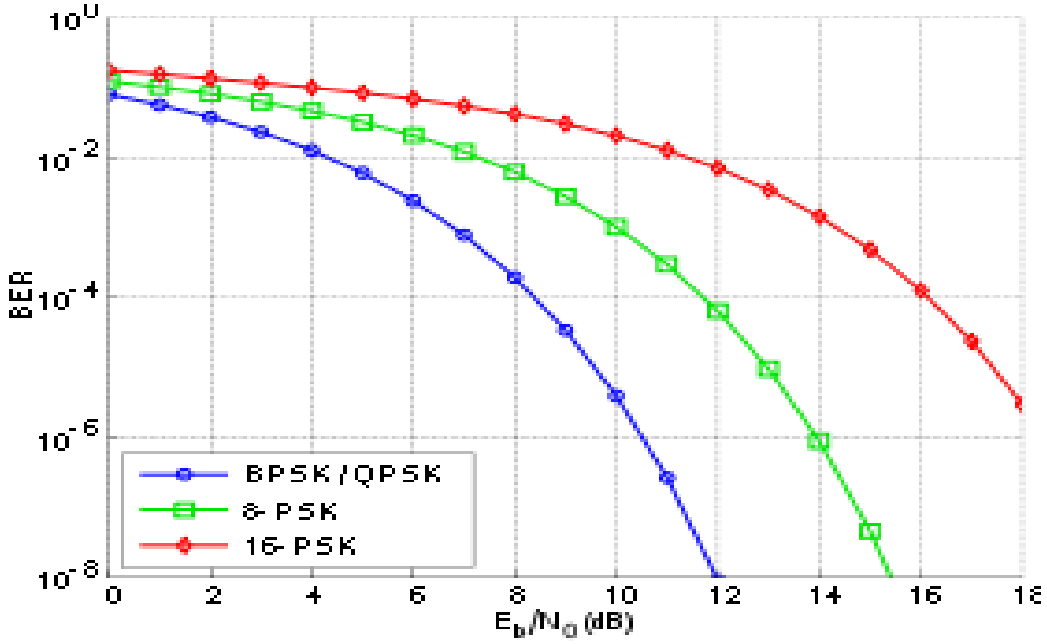


Figure 1.6 : Comparaison du BER de la modulation avec différents ordre de modulation

Ceci se représente sous forme de constellation a deux points. Chaque bit entrant dans le modulateur **BPSK** se retrouvera en l'un de ces deux points de cette constellation. Ceci est dors une modulation par phase.

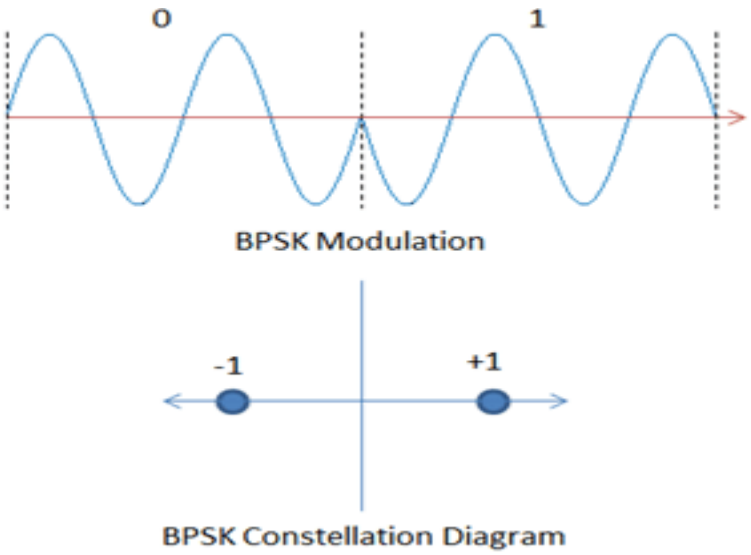


Figure1.7: Constellation Diagramme du signal BPSK

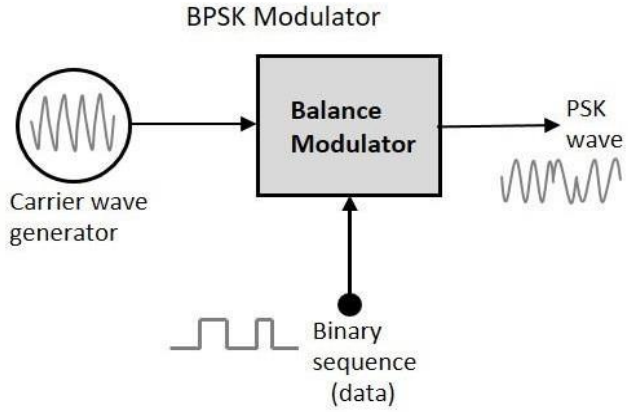


Figure 1.8 : BPSK Modulator

I.7Conclusion

Dans ce chapitre, nous avons introduit, dans un premier temps, le schéma fondamental d'une communication numérique, en expliquant brièvement chacune de ses parties. Ensuite, nous avons présenté le codage de canal, en introduisant le théorème fondamental de codage de canal.

Nous nous sommes, par la suite intéressés au codage canal qui permet d'introduire de la redondance dans le message afin de lutter contre les erreurs.

Enfin, nous avons mentionné le type de la modulation a travers les différentes simulations présentées dans ce chapitre, nous avons remarqué que la modulation BPSK présente de bonnes performances de point de vu BER, en justifiant avec ça le choix de la modulation BPSK dans le reste de notre travaille le type de canal que nous avons utilisés dans notre étude. Il sera question dans le prochain chapitre d'étudier les codes convolutifs et les algorithmes de décodages, et puis nous allons présenter le principe de fonctionnement du turbo codes et leur décodage itératif.

Chapitre II
TURBO CODE

II.1 Introduction

Les turbos codes sont une classe de codes correcteurs d'erreurs avancés largement utilisés dans les systèmes de communication numérique pour améliorer la fiabilité de la transmission des données. Ils ont été inventés par Claude Berrou (17), Alain Glavieux et Punya Thitimajshima en 1993. Les turbo codes sont célèbres pour leur performance proche de la limite de Shannon, ce qui signifie qu'ils s'approchent de la capacité maximale théorique de transmission d'informations à travers un canal de communication donné.

Le codeur turbo-code est construit à l'aide d'une concaténation parallèle de deux codes convolutifs systématiques récursifs, et le décodeur associé, à l'aide d'une règle de décodage par rétroaction. (17)

Dans le domaine des codes correcteurs d'erreurs, plusieurs recherches ont été entreprises au fil des ans pour développer des algorithmes de décodage efficaces. En 1966, Chang et Hancock ont introduit un algorithme appelé MAP (Maximum a Posteriori), initialement conçu pour minimiser la probabilité d'erreur par symbole. Par la suite, des chercheurs ont adapté cet algorithme aux codes correcteurs d'erreurs, ouvrant ainsi la voie à de nouvelles avancées dans ce domaine. Le choix de l'algorithme de décodage revêt une importance majeure, et pour les turbo codes, deux principales approches ont émergé. La première est le SOVA (Soft Output Viterbi Algorithme), une variante de l'algorithme de Viterbi. La seconde approche, que nous aborderons dans la première partie de ce chapitre, est l'algorithme MAP, qui représente une autre méthode puissante pour décoder les turbo codes.

Dans un premier lieu, nous allons introduire le principe du maximum a posteriori qui est la base de l'algorithme MAP. Ensuite nous allons présenter l'algorithme dans le cadre des codes convolutionnels. Dans un deuxième lieu, nous allons présenter le principe de décodage itératif.

II.2 Généralités sur les codes convolutionnels

Les codes convolutionnels sont une classe de codes correcteurs d'erreurs utilisés dans les systèmes de communication numérique pour améliorer la fiabilité de la transmission des données sur des canaux sujets à des erreurs. Voici quelques généralités sur les codes convolutionnels :

1. **Structure et fonctionnement :** Les codes convolutionnels sont généralement définis par des diagrammes d'états et des polynômes générateurs. Ils fonctionnent en appliquant des opérations de convolution sur les bits d'entrée pour générer des séquences de bits de sortie, en utilisant des registres à décalage et des opérations de combinaison linéaire.
2. **Distance minimale :** La distance minimale d'un code convolutionnel, qui mesure la capacité du code à détecter et corriger les erreurs, dépend des paramètres du code tels que la longueur des registres à décalage et les polynômes générateurs.
3. **Performance :** Les performances des codes convolutionnels sont souvent évaluées en termes de distance minimale, de taux d'erreur binaire (BER), de taux d'erreur trame (FER) et d'autres métriques de performance dans des conditions de canal spécifiques.
4. **Démodulation et décodage :** Pour décoder les données reçues codées avec des codes convolutionnels, des algorithmes de décodage tels que l'algorithme de Viterbi sont utilisés et l'algorithme MAP et SOVA. Ces algorithmes sont conçus pour estimer la séquence de bits d'entrée la plus probable à partir de la séquence de bits de sortie, en tenant compte des erreurs introduites par le canal de transmission.
5. **Application :** Les codes convolutionnels sont largement utilisés dans les systèmes de communication numérique, tels que les réseaux sans fil, les systèmes de téléphonie mobile, les systèmes de télévision numérique, les communications par satellite, les réseaux de capteurs sans fil, et bien d'autres.

II.2.1 Le principe de base

Le principe d'un codeur convolutionnel est illustré par le schéma de la figure :

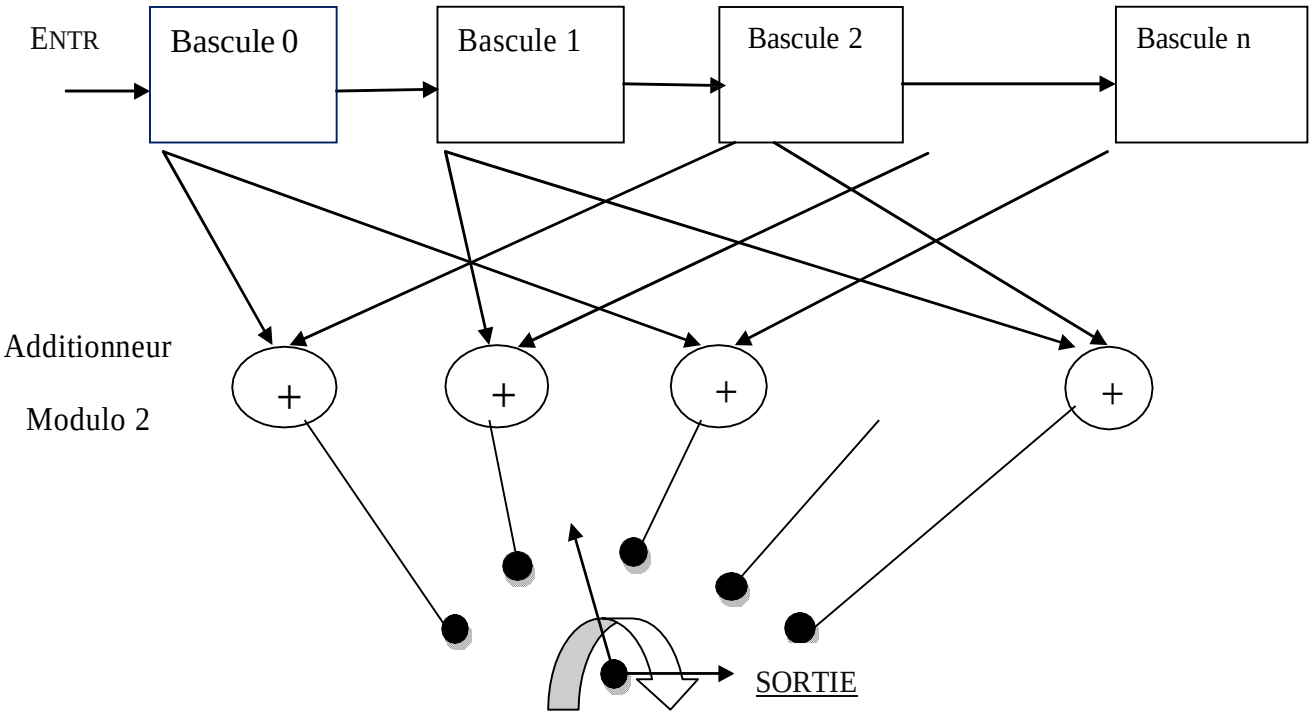


Figure 2.1 : Principe d'un codeur convolutionnel

Un codeur convolutionnel constituée d'un registre à décalage de m+1 étages de additionneurs modulo qui sont reliés à certaines des cellules de registre et d'un commutateur qui sélectionne séquentiellement les sorties des additionneurs.

La valeur K est appelée la longueur de contrainte du codeur. Le mot de code dépend ainsi non seulement du message d'information à son entrée, mais aussi des états de l'encodeur (les bits de message qui ont déjà été codés) et qui sont stockés dans les registres à décalage.

Au début du codage, toutes les cellules de l'encodeur sont initialisées à zéro. Les bits d'information sont introduits dans le registre un par un et la sortie de l'encodeur est égale au produit de convolution entre le symbole d'information présent à l'entrée du codeur convolutionnel et les (K - 1) symboles qui précédent ce dernier.

Les trois principaux paramètres qui caractérisent un codeur convolutionnel sont : la longueur de contrainte, le taux de codage, et le polynôme générateur.

La longueur de la contrainte d'un code convolutif, est définie comme le nombre de changement produits en sortie par un seul bit du message.

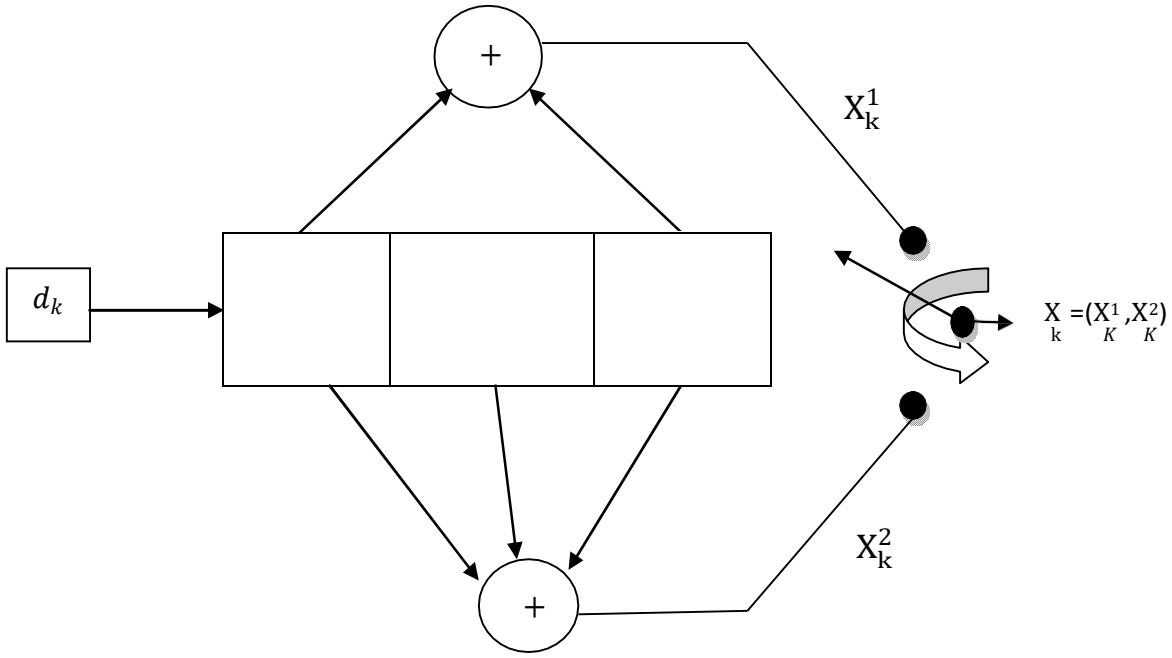


Figure 2.2 : Codeur convolutionnel de taux de codage R=1/2

Le taux de codage R : en général, il s'agit d'une fraction dont le numérateur correspond au nombre d'entrées et le dénominateur correspond au nombre de sorties.

- Les polynômes générateurs G1, …, Gn: ceux-ci caractérisent le code et sont les opérations binaires qui permettent d'obtenir les symboles codés de sortie.

II.2.2 Type de codeur

Il existe plusieurs types de codeurs convolutionnels: codeur convolutionnel systématique récursif, codeur convolutionnel systématique non récursif, codeur convolutionnel non systématique récursif et codeur convolutionnel non systématique récursif.

II.2.2.1 code convolutionnel systématique

C'est un type de code utilisé en communication ou une partie des données d'entrée, ce qui simplifie le processus de décodage en conservant un partie des informations d'origine. Autrement dit, un codeur systématique de taux de codage R = 1/2, laisse passer directement un bit d'information de l'entrée à la sortie comme symbole systématique. Les autres symboles à la sortie de l'encodeur sont appelés symboles de parités. Pour des faibles rapports signal sur

bruit, ces codes offrent des performances de correction d'erreurs meilleures que les codes non-systématiques [18].

II.2.2.2 Code convolutionnel récursif

Un code convolutionnel récursif est un type de code correcteur d'erreurs utilisé dans les systèmes de communication numérique. Contrairement aux codes convolutionnels non récursifs, qui n'ont pas de rétroaction de l'état interne, les codes convolutionnels récursifs utilisent des rétroactions pour générer leurs sorties. Cette rétroaction permet aux codes convolutionnels récursifs d'avoir des propriétés de mémoire, ce qui signifie qu'ils peuvent prendre en compte les entrées précédentes lors de la génération de leurs sorties. Cela les rend particulièrement efficaces pour la correction d'erreurs dans des canaux de communication sujets à des distorsions ou des bruits.

II.2.2.3 Code convolutionnel récursif et systématique

Un code convolutionnel récursif et systématique combine donc les avantages de la récursivité, qui améliore la capacité de correction d'erreurs du code, avec la systématicité, qui simplifie le processus de décodage. Ces codes sont souvent utilisés dans les systèmes de communication numérique où une correction d'erreurs efficace et une complexité de décodage réduite sont des priorités, tels que les communications sans fil, les transmissions par satellite, et les systèmes de stockage à haute densité.

La sortie d'un codeur convolutif est calculée en appliquant une fenêtre glissante sur la totalité du message.

Cette fenêtre glissante est constituée de K groupes de k bits chacun : elle est donc de taille K. K est appelée **longeur de contrainte** (constraint length). À chaque cycle d'horloge, la fenêtre avance d'un groupe de bits, et donc k nouveaux bits rentrent dans la fenêtre tandis que les k plus anciens bits en ressortent. Des opérations sur les bits présents la fenêtre (toujours en respectant les règles du corps de Galois) permettent alors d'obtenir les bits de sortie. On note n le nombre de bits de sortie, ils sont ajoutés au code déjà produit.

Les bits utilisés par les additions sont représentés par des polynômes générateurs ; il y a donc autant de polynômes que de bits de sortie. Par exemple, le polynôme générateur $g=15_8$ a pour représentation binaire 001101_2: l'addition qu'il représente s'effectue donc entre les bits 1, 3 et 4[19].

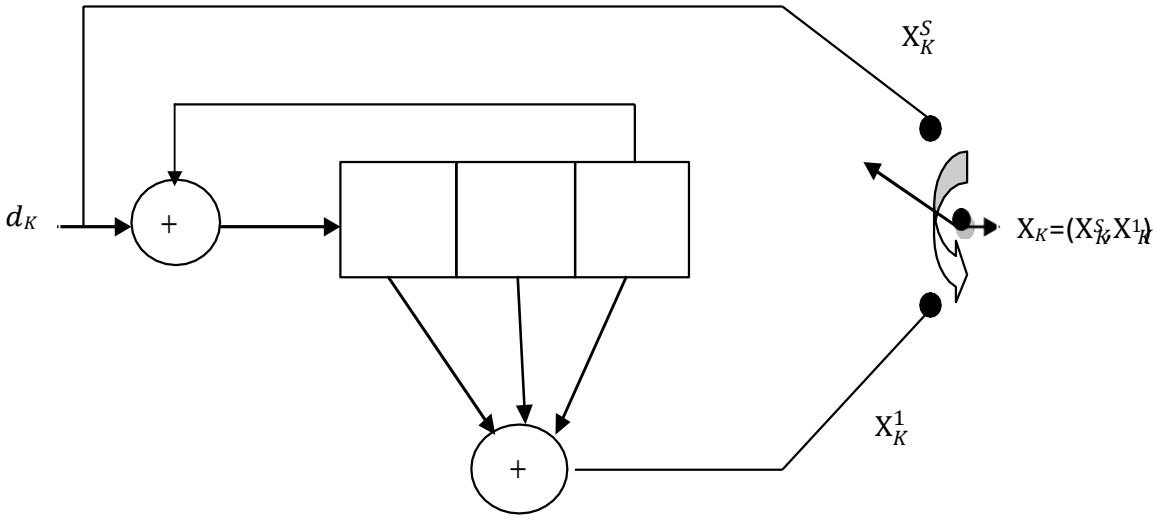


Figure 2.3 :Codeur convolutionnel récursif systématique

Remarques :

- Le rendement d'un code convolutif est, comme pour les codes en bloc k/n.
- Traditionnellement, le registre est initialisé à zéro.
- Contrairement aux codes en bloc, les n bits de sortie ne dépendent pas seulement des k bits d'entrée mais également des K−1 blocs précédents : il y a donc un effet de mémoire.
- Les codes convolutifs sont appelés ainsi car chacune des n sortie du codeur correspond au produit de convolution de l'entrée avec le polynôme générateur correspondant (toujours dans le corps de Galois !).

II.3 Représentation graphique des codes convolutionnels

Les codes convolutionnels peuvent être représentés de différentes manières graphiques pour aider à leur compréhension et à leur analyse. Deux des représentations graphiques les plus couramment utilisées sont le diagramme d'états et le diagramme de treillis. Voici une explication de chacun :

II.3.1 Diagramme d'états

Un diagramme d'états représente les différents états internes du code convolutionnel ainsi que les transitions entre ces états en fonction des bits d'entrée. Chaque état est représenté par un cercle et chaque transition est représentée par une flèche qui indique le nouvel état atteint lorsque certaines conditions de transition sont remplies. Les états et transitions sont déterminés par la structure du registre à décalage et les polynômes générateurs du code. Le

diagramme d'états est souvent utilisé pour décrire la structure et le comportement dynamique du code. On va prend un exemple [5,7] en binaire étant donné le code (101111) qui appeler (In put)

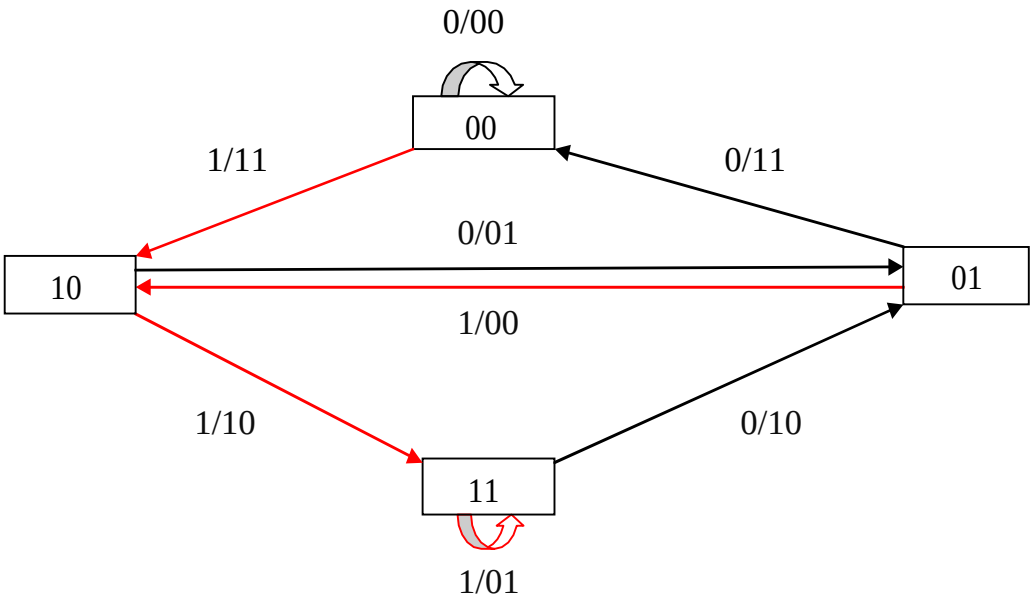


Figure 2.3.1- Diagramme d'état

Entrée à 0 ⟶

Entrée à 1 ⟶

II.3.2 Diagramme de treillis

Un diagramme de treillis est une représentation graphique des chemins possibles à travers le code convolutionnel pour une séquence donnée de bits d'entrée. Chaque nœud du treillis représente un état du code à un instant donné, et les transitions entre les nœuds représentent les différentes combinaisons de bits d'entrée et de sortie. Les chemins à travers le treillis représentent les différentes séquences de bits de sortie générées par le code pour une séquence d'entrée donnée. Le diagramme de treillis est souvent utilisé pour visualiser les opérations de décodage, en particulier lors de l'utilisation d'algorithmes tels que l'algorithme de Viterbi.

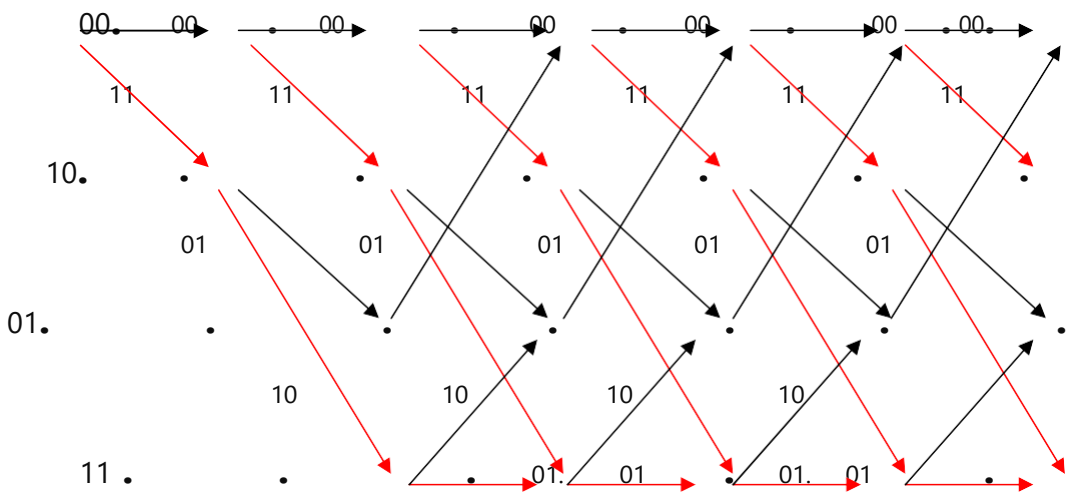


Figure 2.3.2 Diagramme de treillis

II.4. Décodage convolutif

II.4.1 Hard décision (Algorithme Viterbi)

L'algorithme de Viterbi est nommé d'après Andrew Viterbi, qui l'a proposé en 1967 comme algorithme de décodage pour les codes convolutifs sur des laissions de communication numériques bruyantes. Il a cependant une histoire d'invention multiple, avec au moins sept découverte indépendantes, dont celle de Viterbi, Needleman et Wunsch et Wagner et Fischer [21].

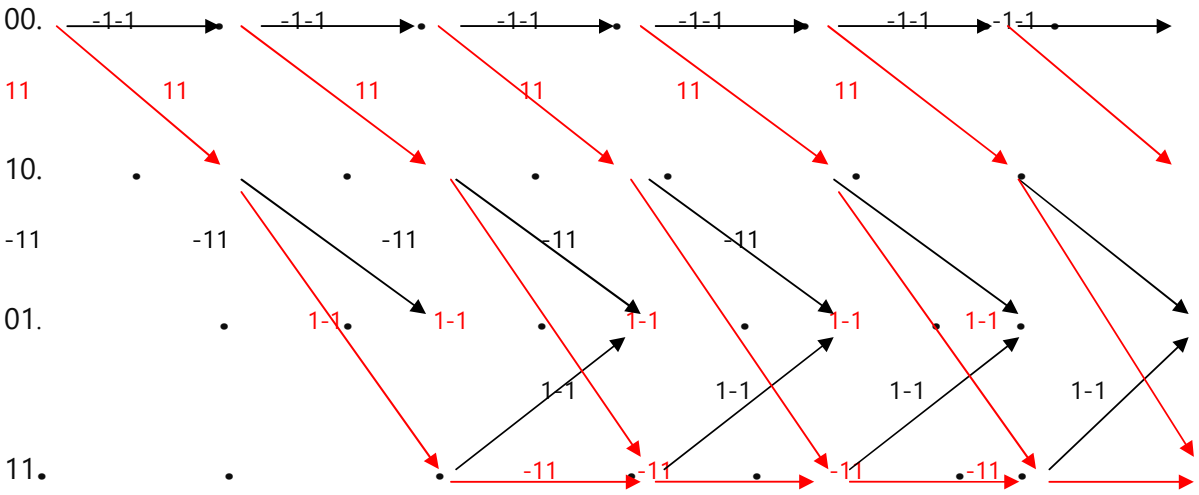


Figure 2.3.3 : Algorithme Viterbi

II.4.2 Soft décision (algorithme MAP)

Dans le domaine des codes correcteurs d'erreurs, plusieurs recherches ont été entreprises au fil des ans pour développer des algorithmes de décodage efficaces. En 1966, Chang et Hancock ont introduit un algorithme appelé MAP (Maximum a Posteriori), initialement conçu pour minimiser la probabilité d'erreur par symbole. En effet, deux algorithmes de décodage ont été proposés pour les codes turbo. Le premier est le SOVA (Soft Output Viterbi Algorithm) qui est une variante de l'algorithme de Viterbi. Le deuxième est celui que nous allons traiter dans la première partie de ce chapitre et qui porte le nom de l'algorithme MAP.[11]

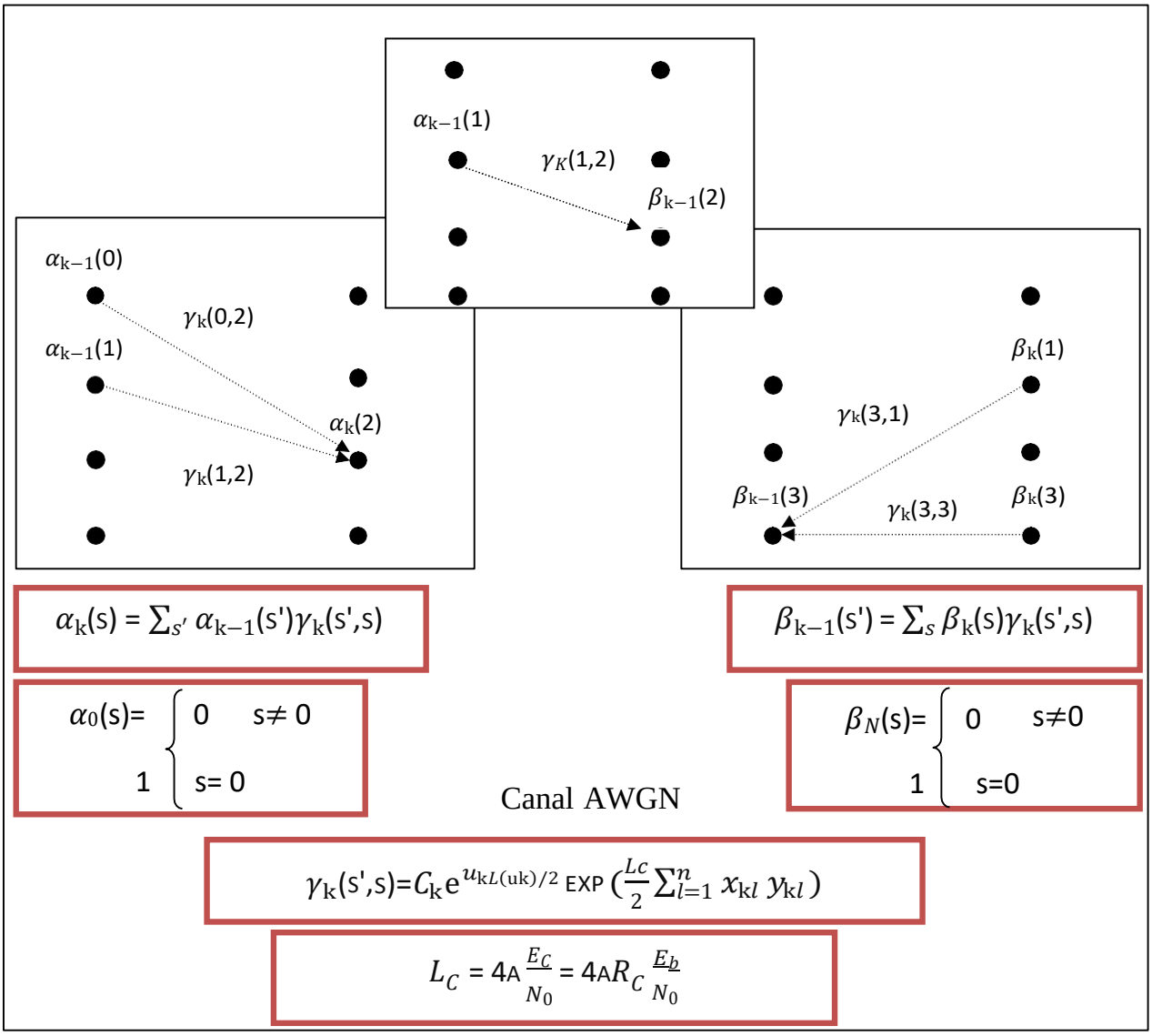


Figure 2.4: Algorithme Map

EXEMPLE :

$\alpha_k(2) = \alpha_{k-1}(0)\ \gamma_k(0,2) + \alpha_{k-1}(1)\ \gamma_k(1,2)$ (2.1)

$\beta_{k-1} = \beta_k(0)\ \gamma_k(1,0) + \beta_K(2)\ \gamma_k(1,2)$ (2.2)

VEUILLEZ NOTER QUE :

DANS LES DEUX CAS, NOUS AVONS BESOIN DE LA MEME QUANTITE, γ K (S',S) . ELLE DEVRA ETRE CALCULEE EN PREMIER.

DANS LE CAS α_k(S), LA SOMMATION PORTE SUR TOUS LES ETATS ANTERIEURS s_{k-1}=S' LIES A L'ETAT S PAR DES BRANCHES CONVERGENTES.

BRANCHES CONVERGENTES, TANDIS QUE DANS LE CAS β_{k-1}(S') LA SOMMATION PORTE SUR TOUS LES ETATS SUIVANTS s_k = S ATTEINTS PAR LES BRANCHES ISSUES DE S'.

BRANCHES ISSUES DE S'. CES SOMMATIONS NE CONTIENNENT QUE DEUX ELEMENTS AVEC DES CODES BINAIRES.

LA PROBABILITE α EST CALCULEE AU FUR ET A MESURE QUE LA SEQUENCE Y EST REÇUE. EN D'AUTRES TERMES, LORSQUE NOUS CALCULANT α, NOUS AVANÇONS DU DEBUT A LA FIN DE LA SEQUENCE DU DEBUT A LA FIN DE TREILLIS.

LA PROBABILITE β NE PEUT ETRE CALCULEE QU'APRES AVOIR REÇU TOUTE LA SEQUENCE Y. EN D'AUTRES TERMES, LORS DU CALCULANTβ, NOUS REVENONS EN ARRIERE DE LA FIN AU DEBUT DU TREILLIS.

NOUS VERRONS QUE α ET β SONT ASSOCIES AUX ETATS DU CODEUR ET QUE γ EST ASSOCIE AUX BRANCHES OU TRANSITIONS ENTRE LES ETATS.

LES VALEURS INITIALES α_0 (S) ET β_N(S) SIGNIFIENT QUE LE TREILLIS EST TERMINE, C'EST-A-DIRE QU'IL COMMENCE ET SE TERMINE DANS L'ETAT ZERO.

DANS L'ETAT ZERO .IL SERA DONC NECESSAIRE D'AJOUTER QUELQUES BITS DE QUEUE AU MESSAGE AFIN QU LE CHEMIN DU TREILLIS SOIT FORCE DE REVENIR A L'ETAT ZERO.

TREILLIS SOIT FORCE DE REVENIR A L'ETAT INITIAL.

CALCULATOINγ :

DANS LE CAS PARTICULIER D'UN CANAL AWGN, L'EXPRESSION PRECEDENTE DEVIENT :

$$\gamma_k(s', s) = C_k e^{u_{kL}(uk)/2} \text{EXP} \left(\frac{Lc}{2}\sum_{l=1}^{n} x_{kl}\, y_{kl}\right) \qquad (2.3)$$

OU C_k EST UNE QUANTITE QUI S'ANNULERA LORSQUE NOUS CALCULONS LE LLR CONDITIONNEL L(UK/Y), CAR IL APPARAIT A LA FOIS DANS LE NUMERATEUR ET LE DENOMINATEUR DE L'EQUATION, LA MESURE OU LA VALEUR DE LA FIABILITE DU CANAL , EST EGALE A:

$$L_C = 4A\frac{E_C}{N_0} = 4R_C\frac{E_b}{N_0} \qquad (2.4)$$

DANS L'ANNEXE, ON MONTRE QUE LE LLR A POSTERIORI (UK/Y) EST DONNE PAR L'EXPRESSION:

$$L(U_K/Y) = LN\frac{\sum_{R1}\alpha_{K-1}(s')\gamma_k(s,s')\beta_k(s)}{\sum_{R0}\alpha_{K-1}(s')\gamma_k(s,s')\beta_k(s)} \qquad (2.5)$$

II.4.3 Versions simplifiées de l'algorithme Map

Trois versions simplifiées de l'algorithme MAP (Maximum a Posteriori Probability) sont souvent utilisées dans la pratique : l'algorithme SOVA (Soft Output Viterbi Algorithme) en 1989 [22] et l'algorithme Log-MAP en 1990-1994 [11]et l'algorithme MAX-Log-MAP en 1995 [8] , Ces deux algorithmes remplacent les multiplications par des additions. Trois nouvelles variables A B et Γ.[8]

$\alpha \Longrightarrow A$

$\beta \Longrightarrow B$

$\gamma\Gamma \Longrightarrow$

$\Gamma_k(s',s) = \ln \gamma_k(s',s)$

$= \ln C_k + \frac{U_k L(U_k)}{2} + \frac{L_C}{2}\sum_{l=1}^{n} X_{kl}\, Y_{kl}$

$A_k(s) = \ln \alpha_k(s)$

$= \max^*[A_{k-1}(s') + \Gamma_k(s',s)]$

$A_0(s) = \begin{cases} 0 & s = 0 \\ -\infty & s \neq 0 \end{cases}$

$B_{k-1}(s') = \ln \beta_{k-1}(s')$

$= \max^*[\beta_k(s) + \Gamma_k(s',s)]$

$B_N(s) = \begin{cases} 0 & s = 0 \\ -\infty & s \neq 0 \end{cases}$

ET ona:

$$\max^*(a,b) = \begin{cases} \max(a,b)+\ln(1+e^{-|a-b|}) & \text{log-MAP} \\ \max(a,b) & \text{max-log-MAP} \end{cases}$$

$$L(U_k|y) = max_{R1}^*[A_{k-1}(s')+\Gamma_k(s',s)+B_K(s)]-max_{R0}^*[A_{k-1}(s')+\Gamma_k(s',s)+B_K(s)] \quad (2.6)$$

II.5 Turbo code

Des turbo codes comme tout code correcteur d'erreur, est d'introduire une redondance dans le message afin de le rendre moins sensible aux bruits et perturbations subies lors de la transmission. Le codage consiste à utiliser deux codeurs simples, dont les entrées ont été entrelacées ; ainsi, chaque codeur voit une série d'informations différentes à son entrée [20]

La concaténation de deux codeurs récursifs et systématiques vise à produire deux encodages de la même information. Il s'agit de deux codeurs avec des structures identiques, utilisant les mêmes polynômes générateurs.

L'encodeur turbo est constitué de deux ou plusieurs codeurs convolutionnels systématiques récursifs (CSR) concaténés en parallèle et séparés par un ou plusieurs entrelaceur. La concaténation parallèle provient du fait que les codeurs utilisent la même séquence d'information mais dans un ordre temporel. Une nouvelle forme de concaténation, dite concaténation parallèle (CP), a été introduite au début des années 1990 pour élaborer les turbo codes.

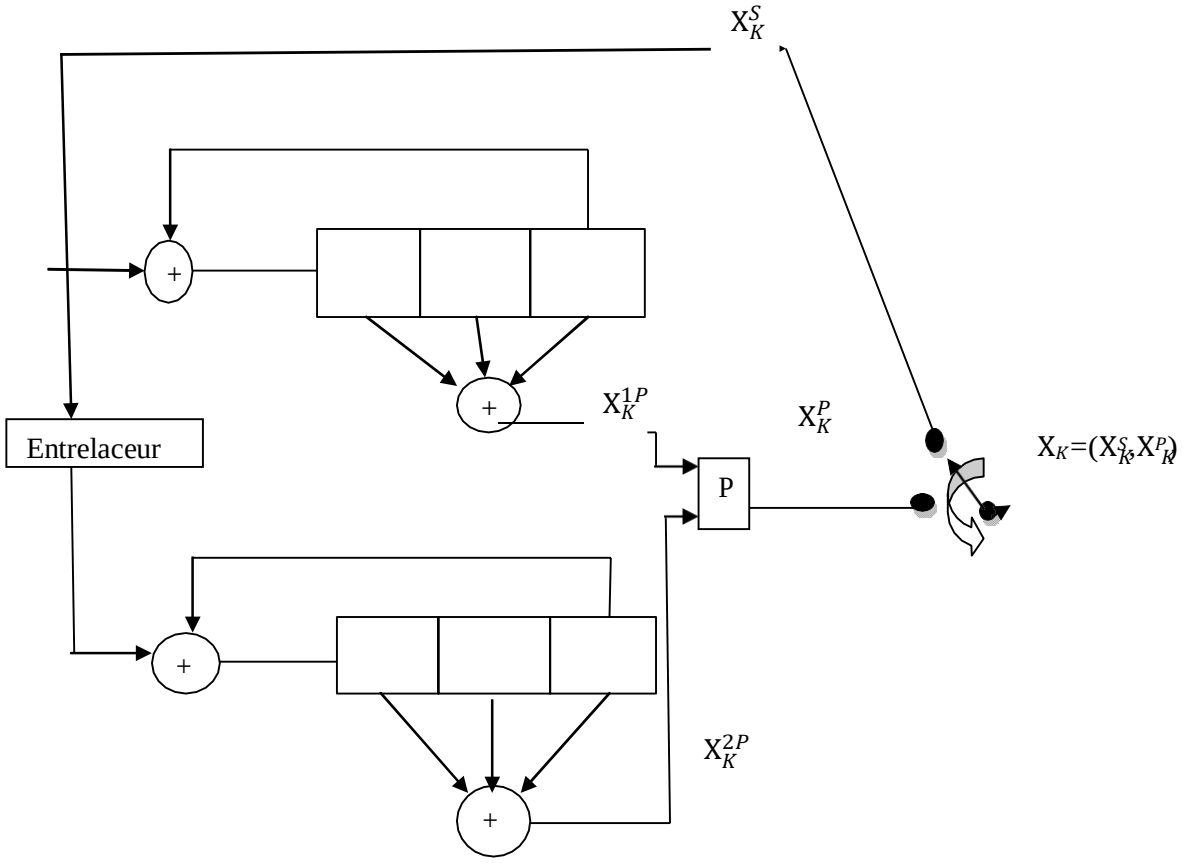


Figure 2.5 : Schéma d'un turbo code

Lorsqu'une séquence de symboles D_K arrive au codeur, elle passe par deux étapes parallèles.

La première correspond au premier codeur de l'étage supérieur. Cette étape est simplement lecodage convolutionnel de cette séquence. Elle produit alors une séquence de symboles de ParitéX_K^{1P}. La séquence d'entrée passe en parallèle par le codeur inférieur après avoir été entrelacée. Ce deuxième codeur produit une séquence de symboles de paritéX_K^{2P}. Une fois ces deux symboles de parité générés, ils peuvent être perforés ou non afin de produire la séquence de parité$X_K^P = (X_K^{1P}, X_K^{2P})$ qui sera multiplexé avec les symboles d'informationX_K^S.

II.6 Décodage itératif

Le décodage d'un turbo-code binaire utilise la technique de décodage itératif, Le décodage itératif est une méthode de décodage utilisée dans les turbo codes et d'autres codes correcteurs d'erreurs avancés. Contrairement aux méthodes de décodage classiques qui effectuent une seule passe sur les données pour estimer les bits originaux, le décodage itératif itère entre les différentes étapes du processus de décodage pour améliorer progressivement la précision de la récupération des données.

II.6.1 Principe de décodage itératif

Le principe de décodage itératif est très simple. Il s'agit de répéter plusieurs fois l'échange d'information pondérée sur les bits d'information entre les deux décodeurs jusqu'à l'obtention des performances de décodage désirées. En fait, le décodage itératif se fait en plusieurs itérations. Chaque itération de décodage implique deux décodeurs.[22]

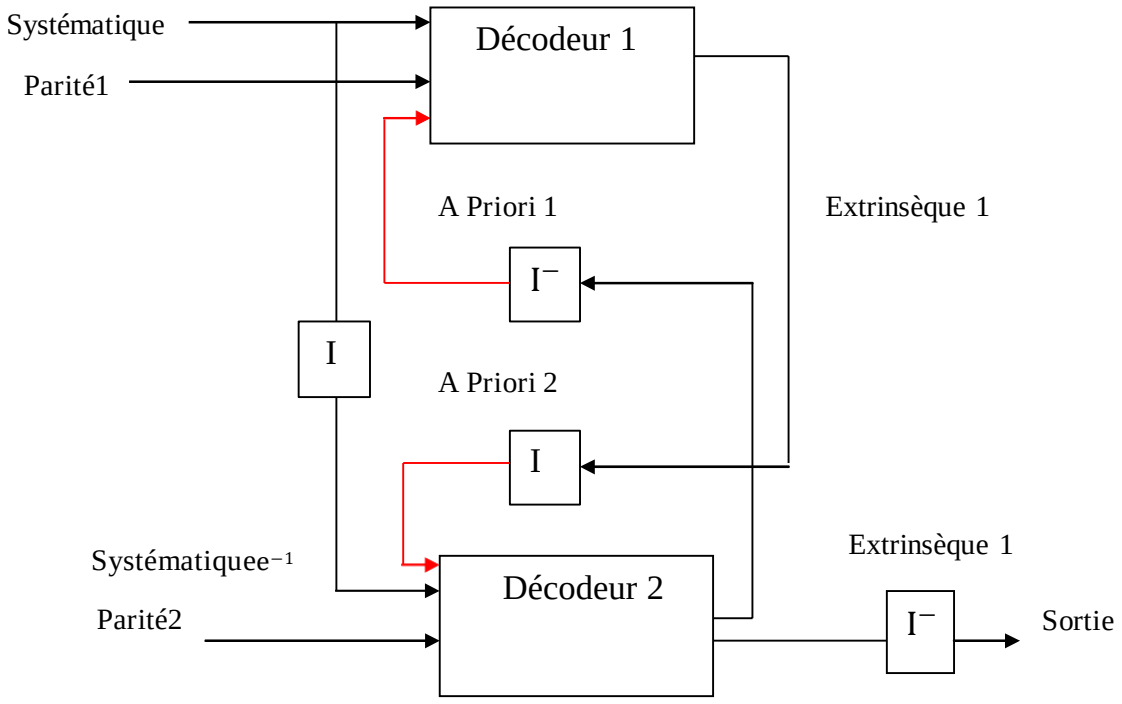


Figure 2.6 : Schéma de décodage itératif

La technique de décodage concernée se fait par itérations de deux étapes.

Pour une première itération et après avoir décodé les symboles associés au premier codeur, le décodeur 1 fait passer une information vers le décodeur 2. Ce dernier dispose de cetteinformation ainsi que des symboles bruités issus du deuxième codeur pour pouvoir

corrigercertaines erreurs qui n'ont pas été détectés par le décodeur 1. A la deuxième itération, si le décodeur 1 peut avoir accès à plus d'informations alors sûrement il fournira des performances meilleures que celles obtenues durant la première étape. Autrement dit, le décodeur 1 reçoit en plus des mêmes symboles bruités issus du premier codeur, une information provenant du décodeur 2. Ainsi, d'une itération a une autre, le processus d'échange d'information se répète jusqu'à une saturation où les deux décodeurs ne peuvent pas améliorer la performance d'erreur. La séquence d'information supplémentaire échangée qui se présente à l'entrée de l'un des deux décodeurs s'appelle séquence d'information à priori. Elle doit être indépendante de toutes les entrées de l'autre décodeur. En plus, elle doit contenir des symboles erronés qui sont réordonnés et non regroupés ensemble afin d'augmenter l'efficacité de décodage. D'où l'intérêt de la présence de l'entrelaceur ou de celle du désentrelaceur entre les deux décodeurs.

II.6.2 Décodeur turbo

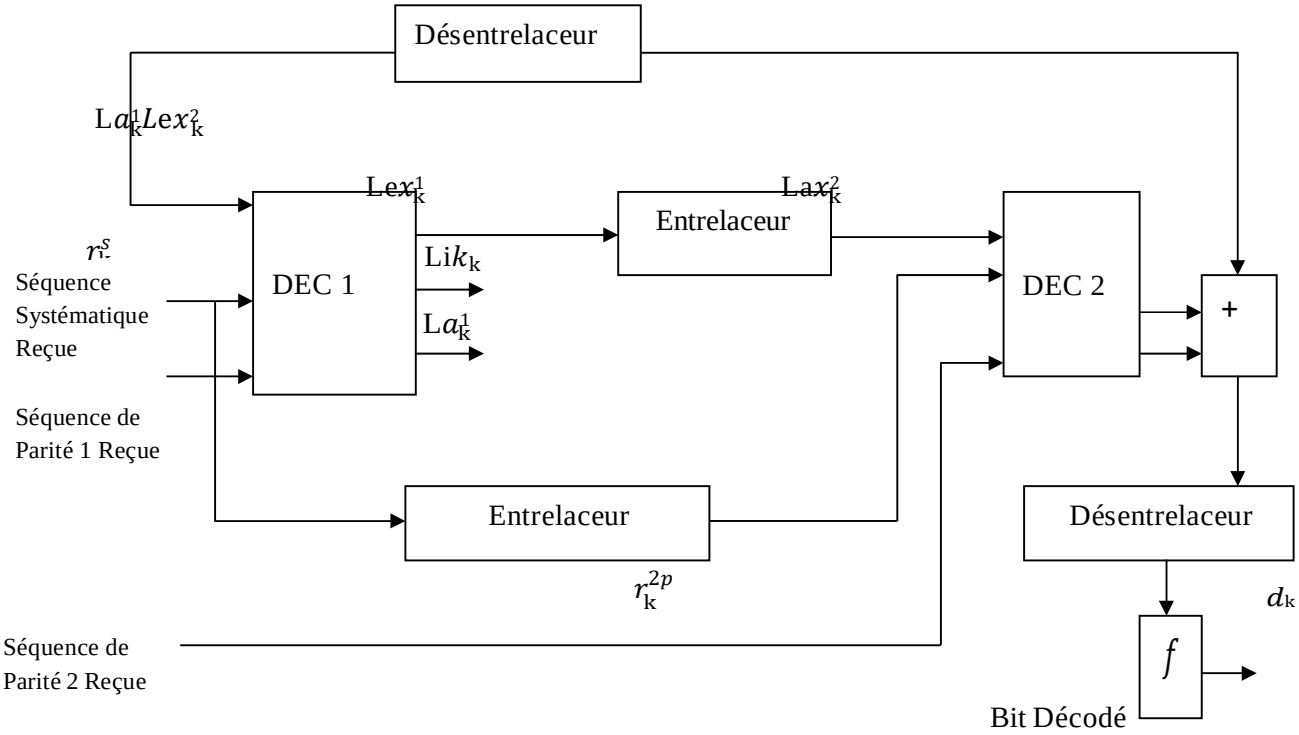


Figure 2.6.1 : Schéma d'un décodeur turbo

Après décodage des symboles par le premier décodeur MAP, la sortie à l'instant k est égale à:

$$LLR^1_K(\mathrm{i}) = Lik_\mathrm{k} + La^1_\mathrm{k}(\mathrm{i}) + Lex^1_\mathrm{k}(\mathrm{i}) \qquad (2.7)$$

Cette donnée est ensuite entrelacée pour correspondre aux symboles entrelacés par le deuxième codeur (voir figure 2.6.1) deuxième décodeur MAP reçoit alors l'information suivante : $Lex^1_\mathrm{k}(\mathrm{i})$, représentant l'information extrinsèque entrelacée, l'information systématique entrelacée r^s_k et la parité du deuxième codeurr^{2p}_k, afin de générer l'information extrinsèque $Lex^2_\mathrm{k}(\mathrm{i})$, qui sera délacée et renvoyée via la boucle de retour vers le DEC1. L'objectif de l'entrelacement est de disperser les erreurs en rafale, difficiles à détecter pour les décodeurs MAP. De plus, l'entrelaceur permet de fournir des observables non corrélés aux décodeurs d'une itération à l'autre. Plus la taille de l'entrelaceur est grande, plus les erreurs en bloc sont dispersées

Après avoir obtenu des informations sur le bit à décoder, le deuxième décodeur MAP peut corriger certaines erreurs que le premier décodeur n'a pas pu corriger. La sortie du deuxième décodeur est la suivante :

$$LLR^2_\mathrm{k}(\mathrm{i}) = Lik_{\mathrm{k}'} + La^2_\mathrm{k}(\mathrm{i}) + Lex^2_\mathrm{k}(\mathrm{i}) \qquad (2.8)$$

Après un nombre fixe d'itérations, la décision finale concernant les bits du message est basée sur les valeurs extrinsèques des deux décodeurs ainsi que sur les valeurs du canal. Il est important de noter que seules les valeurs extrinsèques sont mises à jour à chaque itération, tandis que les valeurs du canal correspondant aux bits du message restent constantes.[23]

II.7 Conclusion

Dans la première partie de ce chapitre, nous avons tout d'abord introduit le codage convolutionnel et en particulier aux codeurs récursifs et systématiques qui sont utilisés pour la conception des codes turbo. Après nous avons expliqué le décodage convolutif hard décision viterbi et soft décision MAP est très complique et après la simplification (Sova, log-MAP et max-log-MAP) . Par la suite intéressés au concaténation parallèle introduisant ainsi le codage turbo. Nous avons ainsi décrit la partie codage de notre système. Il nous reste maintenant à parler du décodage turbo (décodage itératif) et leur principe.

Chapitre III
Les Entrelaceurs

III.1 Introduction

Un entrelaceur est un dispositif utilisé dans les communications et le traitement de données pour organiser ou réorganiser sélectivement les éléments de données entrants ou sortants. L'entrelaceur prend une séquence ordonnée de bits et les mélange habilement avant leur envoi à travers les canaux de communication. Cette transformation apporte une robustesse cruciale aux données en les dispersant de manière aléatoire, réduisant ainsi l'impact des erreurs et des perturbations qui peuvent survenir pendant le transit. Avec son rôle essentiel dans la protection des données et l'amélioration des performances des réseaux de communication, l'entrelaceur se positionne au cœur même de la connectivité numérique moderne, assurant que chaque bit voyage avec précision et fiabilité vers sa destination désirée

Dans d'autres contextes, comme en informatique ou en génie logiciel, un entrelaceur peut être un programme ou un mécanisme qui gère l'exécution concurrente de plusieurs tâches ou processus, les entrelaçant de manière à optimiser l'utilisation des ressources du système**.**

III.2Définition de l'entrelaceur

Un entrelaceur est un dispositif ou un algorithme utilisé en télécommunications et en traitement du signal pour réorganiser ou mélanger l'ordre des éléments d'une séquence de données. Son but principal est de réduire les effets des erreurs ou des perturbations qui peuvent survenir lors de la transmission ou du stockage de données. En réarrangeant l'ordre des données, un entrelaceur peut disperser les erreurs sur plusieurs symboles adjacents, ce qui facilite leur correction ou leur détection par les systèmes de correction d'erreurs ou de décodage. En outre, les entrelaceurs sont souvent utilisés pour répartir de manière équitable les données sur différents canaux de transmission, améliorant ainsi l'utilisation efficace de la bande passante disponible.

Le but de l'entrelaceur n'est donc pas de changer l'information, mais d'en modifier l'ordre. La performance d'un code turbo est essentiellement améliorée quand la longueur d'entrelaceur augmente [5]

En somme, l'entrelaceur est souvent un élément crucial pour améliorer l'efficacité, la performance et la gestion des systèmes complexes.

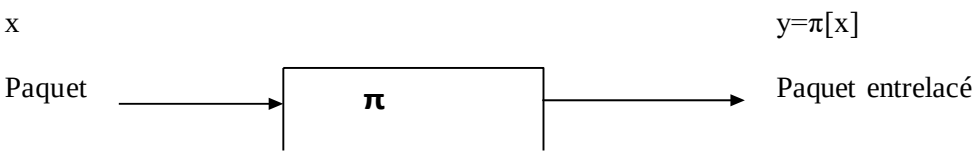


Figure 3.1: Entrelaceur.

Le paquet x est inséré à l'entrée de I'entrelaceur π. La sortie est alors $y = \pi[x]$. L'entrelaceur doit aussi être construit de façon garantir le meilleur passage de l'information extrinsèque d'un décodeur à l'autre. Ceci afin que les performances du décodeur itératif soient meilleures. Parmi les entrelaceurs assurant le meilleur passage de l'information extrinsèque nous pouvons citer les entrelaceurs S-random.

III.3 Principe des entrelaceurs

L'entrelacement est une technique largement employée dans de nombreux systèmes de communication numérique. En effet, un entrelaceur prend une séquence de symboles en entrée et la réorganise totalement dans le temps en sortie, sans aucune répétition. Il agit essentiellement comme un système de permutation des éléments d'une séquence. En combinant deux codeurs à convolution récurrente (RSC) avec un entrelaceur, il est possible de créer un code doté de meilleures propriétés de distance de Hamming. Cet entrelaceur joue un rôle clé dans la concaténation des codes en parallèle ou en série, permettant ainsi l'utilisation de codes avec de courtes longueurs de contrainte K tout en réduisant la complexité des algorithmes de décodage tout en maintenant des performances presque identiques à celles obtenues avec des codeurs classiques à grande mémoire. En ce qui concerne le décodage, l'entrelaceur assume un autre rôle important en décorrélant les entrées des décodeurs, en particulier lorsque l'on utilise un algorithme de décodage itératif basé sur l'échange d'informations entre ces derniers. Autrement dit, dans le cadre d'un décodage itératif utilisant deux décodeurs, si les séquences à leurs entrées sont décorrélées, il y a une forte probabilité que les erreurs indétectables par l'un des décodeurs soient corrigées par l'autre. [5]

III.4 Méthode de l'entrelacement

En télécommunications, en traitement de signal et dans d'autres domaines, l'entrelacement est une technique utilisée pour réorganiser les données de manière à améliorer la robustesse, la fiabilité ou les performances du système. Voici quelques méthodes courantes d'entrelacement:

III.4.1 Entrelacement par bloc

Un entrelaceur en bloc de période P=N*M est caractérisé par un processus où les données sont initialement écrites ligne par ligne dans une matrice, puis lues colonne par colonne.

Pour illustrer son fonctionnement, prenons un exemple : imaginons une séquence de 9 symboles N=3, M=3, P=9 arrivant à l'entrée de l'entrelaceur. Tout d'abord, ces 9 symboles sont disposés dans une matrice (voir Figure 3.2) selon un ordre croissant, où le premier symbole reçu est placé dans la case 0, le deuxième dans la case 1, et ainsi de suite. Ensuite, les symboles sont restitués à la sortie de l'entrelaceur en matrice, suivant l'ordre spécifié : [Insérer l'ordre de sortie ici].

Sortie de l'entrelaceur : 0 3 6 1 4 7 2 5 8

0	1	3
3	4	5
6	7	8

Figure 3.2 : Entrelaceur bloc de taille 3×3 [5].

III.4.2 Entrelaceur déterministe

Dans ce modèle d'entrelaceur, le processus consiste à placer les symboles d'un bloc à entrelacer dans une matrice, ligne par ligne, puis à les extraire en sortie colonne par colonne, conformément à l'illustration présentée dans la Figure(3.3) Plusieurs méthodes existent pour extraire les symboles d'une matrice, telles que de gauche à droite (GD) ou de droite à gauche (DG) pour les colonnes, et de haut en bas (HB) ou de bas en haut (BH) pour les lignes. Ainsi, il existe quatre orientations différentes, comme illustré dans la Figure 4.4 [6].

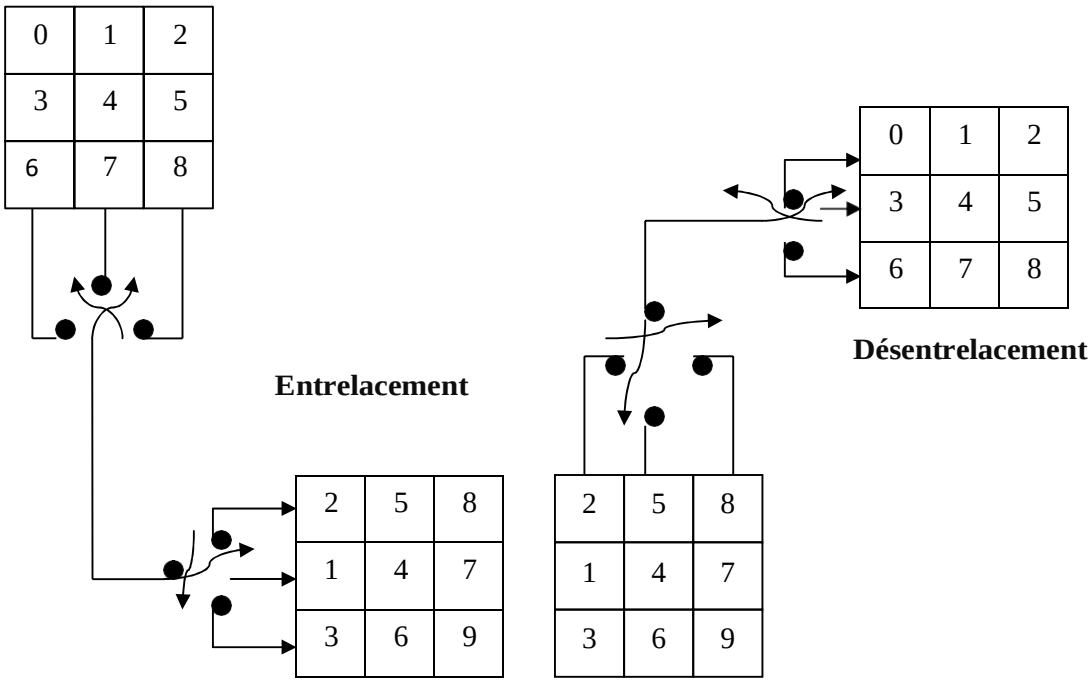


Figure 3.3: Principe d'un entrelaceur déterministe



Figure 3.4:Mode d'opération pour réordonner des symboles [6]

III.4.3 Entrelaceur étalé(S-Random interleaver)

Les entrelaceurs S-random, ou entrelaceurs étalés, sont construits en générant des nombres aléatoires de 1 à L selon une contrainte S où S est la distance d'entrelacement minimale.

Le fonctionnement de l'entrelaceur S-random est le suivant : un entier sélectionné au hasard est comparé aux S1 entiers sélectionnés précédemment. Si les différences absolues entre l'entier sélectionné et l'un des S1 entiers précédemment sélectionnés sont supérieures ou égales à S2, alors l'entier sélectionné au hasard est accepté, sinon il est rejeté. Un entrelaceur S-random est déterminé selon :

$| Ii – Ij | \leq S1$

$| M (Ii) – M (Ij) | \geq S2$

Où Ii désigne l'indice d'origine et M(Ii) est l'indice permuté dans la séquence entrelacée.

Lorsque des codes constitutifs identiques sont utilisés, il est approprié de choisir S1=S2=S, où

$S \leq \sqrt{(N/2)}$ (3.1)

Le principal problème associé à l'entrelaceur S-random conventionnel est son manque de flexibilité, puisque changer le nombre N nécessite une nouvelle recherche d'entrelaceur et que les entrelaceurs générés avec des longueurs différentes doivent être stockés en mémoire séparément. [16]

III.4.4 Entrelaceur aléatoire (random interleaver)

Il s'agit du type d'entrelaceur le plus simple, qui mélange de façon aléatoire les éléments d'un vecteur d'entrée en fonction d'une graine initiale. À chaque utilisation, il peut produire une séquence de sortie différente, car il réorganise aléatoirement les éléments en fonction de cette graine. Pour retrouver l'ordre initial des données, on utilise un processus appelé désentrelacement. Cela nécessite simplement de réutiliser la même graine initiale que celle utilisée pour l'entrelacement. En résumé, le désentrelacement est l'inverse d'entrelacement. [12]

III.5 Programme de Paiement de la Qualité QPP

Les entrelaceurs QPP se caractérisent par [1] : une structure complètement algébrique une mise en œuvre efficace (haute vitesse et faibles besoins en mémoire) et de très bonnes performances. Un entrelaceur QPP de longueur L est défini dans [1], [2], [3], [4] comme suit :

$$\pi(x)=(q_0+q_1x+q_2x_2)\text{mod } L \quad , x=0,1\ldots\ldots,L-1 \qquad (3.2)$$

Où q_1 et q_2 sont choisis de sorte que le polynôme quadratique en 1 soit un polynôme de permutation et où q_0 détermine seulement un décalage des éléments de permutation. L'analyse de la randomisation de ces entrelaceurs en considérant le degré de non-linéarité a été réalisée dans [1] et associe à chaque entrelaceur défini par la permutation π(·), une représentation géométrique appelée code d'entrelacement par la transformation : π(i) → F(i), où F(i) = (i, π(i)) pour i appartenant à I, et où I = {0, 1, …. , L-1} est l'ensemble des indices correspondant aux bits à entrelacer. Le couple (i, π(i)) est appelé point et un ensemble de points dans le code d'entrelacement, qui sont équivalents sous l'action d'un groupe d'isométries, est appelé orbite. Les isométries d'intérêt pour les distances de turbo codes sont les translations d'un point dans le code d'entrelacement, c'est-à-dire la transformation qui décale circulairement un point de k0 unités vers la droite le long de la dimension i et de k1 unités vers le haut le long de la dimension j.

$$A(k_0;\ k_1) : (i;\ \pi(i))\ !\ (i+k_0;\pi(i)\ +k_1);\ k_0;\ k_1 \in I \qquad (3.3)$$

Pour que la translation mène à un point dans le code d'entrelacement, nous avons :

$$(\mathrm{i}+k_0)= (\mathrm{i}) + k_1 \qquad (3.4)$$

keshita définit le degré de non-linéarité ‡ comme le nombre d'orbites distinctes. Ainsi, le degré de non-linéarité est le nombre de solutions distinctes de l'équation (3). Il est démontré dans [1] que le degré de non-linéarité ‡ d'un entrelaceur QPP est donné par :

$$\zeta= {}^{l}/_{gcd}\,(2q_2,\ L) \qquad (3.5)$$

où gcd signifie le plus grand commun diviseur. Généralement, la dispersion est utilisée comme mesure de la randomisation de l'entrelaceur. Elle est définie comme le nombre de vecteurs de déplacement distincts (Δ_S, Δ_y) :

$$\Delta=|\ \{(\Delta_S\ ,\ \Delta_y)\ \in Z2\ |\ \exists \mathrm{i,j}\ \text{tels que}\ \Delta_S\ =\mathrm{j}-\mathrm{i}\ \text{et}\ \Delta_y\ =\pi(\mathrm{j})-\pi(\mathrm{i}\ \ 0\leq \mathrm{i}<\mathrm{j}\leq L-1\}| \qquad (3.6)$$

Dispersion normalisée est la valeur de Δ normalisée à sa valeur maximale, c'est-à-dire

$$\gamma =\frac{2T}{L(l-1)} \qquad (3.7)$$

La dispersion d'un entrelaceur influence les multiplicités des mots de code de faible poids, donc une dispersion élevée est souhaitable [5]. Cependant, les entrelaceurs QPP ont

une faibla dispersion et conduisent toujours à de bonnes performances s'ils sont correctement choisis. [13]

- Les avantages des entrelaceurs polynomiaux incluent des performances spéciales, une structure algébrique complète et une mise en œuvre efficace (haute vitesse et faibles besoins en mémoire).
- Le décodage parallèle peut être activé en utilisant : l'entrelacement/désentrelacement, ce qui entraîne des conflits d'accès à la mémoire lorsque le décodeur MAP tente simultanément de lire/écrire depuis/dans la mémoire.

  L'entrelaceur QPP est basé sur des constructions algébriques via des polynômes de permutation sur des anneaux d'entiers. Il est connu que les polynômes de permutation génèrent des entrelaceurs sans contention [10]. L'entrelaceur QPP peut être représenté par une formule mathématique donnée une longueur de bloc d'information N, la position de sortie d'entrelacement en x est spécifiée par une simple équation quadratique :

$$F(x) = (f_2\, x^2 + f_1\, x) \bmod N \qquad (3.8)$$

Les paramètres f_1 et f_2 sont des entiers et dépendent de la taille du bloc N ($0 \leq x, f_1, f_2 < N$). Pour chaque taille de bloc, un ensemble différent de paramètres f_1 et f_2 est défini. Dans LTE, toutes les tailles de bloc sont des nombres pairs et sont divisibles par 4 et 8. La taille de bloc N est toujours divisible par 16, 32 et 64 lorsque c'est respectivement 512, 1024 et 2048. Selon la définition, le paramètre f1 doit être un nombre impair tandis que f2 doit être un nombre pair. [14]

III.5.1 LTE Normalisation

Les bits en entrée de l'entrelaceur interne du code turbo sont désignés par c_0 , c_1 ,..., c_{k-1} , où K est le nombre de bits en entrée. Les bits en sortie de l'entrelaceur interne du code turbo sont désignés par c_0', c_1',..., c'_{k-1} . La relation entre les bits en entrée et en sortie est la suivante :

Les bits en sortie ci′ sont déterminés par la permutation des bits d'entrée selon l'indice i, où i varie de 0 à K-1. La relation entre l'indice de sortie i et l'indice d'entrée Π(i) satisfait la forme quadratique suivante [10] :

$$\Pi(i) = (f_1 \cdot i + f_2 \cdot i^2) \text{ modulo } K \qquad (3.9)$$

- Les paramètres f_1 et f_2 dépendent de la taille du bloc K et sont résumés dans le Tableau 3.1.

Tableau 3. 1 : Paramètre de l'entrelaceur interne de turbo code. [10]

i	***K***	$f1$	$f2$	***i***	***K***	$f1$	$f2$	***i***	***K***	$f1$	$f2$	***i***	***K***	$f1$	$f2$
1	40	3	10	48	416	25	52	95	1120	67	140	142	3200	111	240
2	48	7	12	49	424	51	106	96	1152	35	72	143	3264	443	204
3	56	19	42	50	432	47	72	97	1184	19	74	144	3328	51	104
4	64	7	16	51	440	91	110	98	1216	39	76	145	3392	51	212
5	72	7	18	52	448	29	168	99	1248	19	78	146	3456	451	192
6	80	11	20	53	456	29	114	100	1280	199	240	147	3520	257	220
7	88	5	22	54	464	247	58	101	1312	21	82	148	3584	57	336
8	96	11	24	55	472	29	118	102	1344	211	252	149	3648	313	228
9	104	7	26	56	480	89	180	103	1376	21	86	150	3712	271	232
10	112	41	84	57	488	91	122	104	1408	43	88	151	3776	179	236
11	120	103	90	58	496	157	62	105	1440	149	60	152	3840	331	120
12	128	15	32	59	504	55	84	106	1472	45	92	153	3904	363	244
13	136	9	34	60	512	31	64	107	1504	49	846	154	3968	375	248
14	144	17	108	61	528	17	66	108	1536	71	48	155	4032	127	168
15	152	9	38	62	544	35	68	109	1568	13	28	156	4096	31	64
16	160	21	120	63	560	227	420	110	1600	17	80	157	4160	33	130
17	168	101	84	64	576	65	96	111	1632	25	102	158	4224	43	264
18	176	21	44	65	592	19	74	112	1664	183	104	159	4288	33	134
19	184	57	46	66	608	37	76	113	1696	55	954	160	4352	477	408
20	192	23	48	67	624	41	234	114	1728	127	96	161	4416	35	138
21	200	13	50	68	640	39	80	115	1760	27	110	162	4480	233	280
22	208	27	52	69	656	185	82	116	1792	29	112	163	4544	357	142
23	216	11	36	70	672	43	252	117	1824	29	114	164	4608	337	480
24	224	27	56	71	688	21	86	118	1856	57	116	165	4672	37	146
25	232	85	58	72	704	155	44	119	1888	45	354	166	4736	71	444
26	240	29	60	73	720	79	120	120	1920	31	120	167	4800	71	120
27	248	33	62	74	736	139	92	121	1952	59	610	168	4864	37	152
28	256	15	32	75	752	23	94	122	1984	185	124	169	4928	39	462
29	264	17	198	76	768	217	48	123	2016	113	420	170	4992	127	234
30	272	33	68	77	784	25	98	124	2048	31	64	171	5056	39	158
31	280	103	210	78	800	17	80	125	2112	17	66	172	5120	39	80
32	288	19	36	79	816	127	102	126	2176	171	136	173	5184	31	96
33	296	19	74	80	832	25	52	127	2240	209	420	174	5248	113	902
34	304	37	76	81	848	239	106	128	2304	253	216	175	5312	41	166
35	312	19	78	82	864	17	48	129	2368	367	444	176	5376	251	336
36	320	21	120	83	880	137	110	130	2432	265	456	177	5440	43	170
37	328	21	82	84	896	215	112	131	2496	181	468	178	5504	21	86
38	336	115	84	85	912	29	114	132	2560	39	80	179	5568	43	174
39	344	193	86	86	928	15	58	133	2624	27	164	180	5632	45	176
40	352	21	44	87	944	147	118	134	2688	127	504	181	5696	45	178

III.6UMTS: Universal Mobile Telecommunications System

III.6.1 Entrelaceur interne du turbo code

L'entrelaceur interne du code Turbo se compose de la génération de l'entrelaceur mère et de l'élagage. Pour une longueur de bloc K donnée arbitrairement, un entrelaceur mère est sélectionné parmi l'ensemble de 134 entrelaceurs mère. Après la génération de l'entrelaceur mère, l-bits sont élagués afin d'ajuster l'entrelaceur mère à la longueur de bloc K. [9]

III.6.2 Génération de l'entrelaceur mère

L'entrelacement se compose de trois étapes. Dans la première étape, la séquence d'entrée est écrite dans la matrice rectangulaire ligne par ligne. La deuxième étape est la permutation intra-ligne. La troisième étape est la permutation inter-lignes. Les permutations en trois étapes sont décrites comme suit, la longueur de bloc d'entrée est supposée être K (de 320 à 5114 bits) [9].

Première étape

(1) Déterminer un numéro de ligne R tel que

R=10 (K = 481 à 530 bits ; Cas-1)

R=20 (K = toute autre longueur de bloc sauf 481 à 530 bits ; Cas-2)

(2) Déterminer un numéro de colonne C tel que

Cas-1 ; C = p = 53

Cas-2 ;

(i) Trouver le plus petit nombre premier p tel que,

0 =< (p+1)-K/R,

(ii) (ii) Si (0 <= p-K/R) alors passer à (iii),

Sinon C = p+1.

(iii) Si (0 <= p-1-K/R) alors C=p-1, sinon C = p.

(3) La séquence d'entrée de l'entrelaceur est écrite dans la matrice rectangulaire RxC ligne par ligne.

Deuxième étape

A. Si C = p

(A-1) Sélectionnez une racine primitive g0 à partir du Tableau 2.

(A-2) Construisez la séquence de base c(i) pour la permutation intra-ligne comme suit :

$$c(i) = [g0 \acute{} c(i-1)] \bmod p \text{ , } i = 1,2,\ldots, (p-2)., \; c(0) = 1. \qquad (3.10)$$

(A-3) Sélectionnez l'ensemble d'entiers premiers minimum {qj } (j=1,2,…R-1) tel que

Pgcd $\{q_j, p-1\} = 1$

$q_j > 6$

$q_j > q_{(j-1)}$ Où pgcd est le plus grand commun diviseur. Et q0 = 1.

(A-4) L'ensemble {qj} est permuté pour former un nouvel ensemble $\{P_j\}$ tel que

$Pp(j) = q_j$, j = 0, 1, …. R-1,

Où P(j) est le motif de permutation inter-lignes défini dans la troisième étape.

(A-5) Effectuez la j-ème (j = 0, 1, 2, …, R-1) permutation intra-ligne comme suit :

$$c_j(i) = c([i \acute{} P_j] \bmod (p-1)), \quad i = 0, 1,2,\ldots, (p-2)., \text{ et } c_j(p-1) = 0, \qquad (3.11)$$

Où c_j (i) est la position du bit d'entrée du i_ème sortie après la permutation de la j-ème ligne.

B. <u>Si C = p+1</u> (3.12)

(B-1) Identique au cas A-1.

(B-2) Identique au cas A-2.

(B-3) Identique au cas A-3.

(B-4) Identique au cas A-4.

(B-5) Effectuez la j-ème (j = 0,1, 2, …, R-1) permutation intra-ligne comme suit :

$$c_j(i) = c([i \acute{} p\, j]) \bmod (p-1)) \text{ , } i = 0,1,2,\ldots, (p-2)., \; c_j(p-1) = 0, \text{ et } c_j(p) = p, \qquad (B\text{-}6)$$

Si (K = C x R) alors échangez $c_{R-1}(p)$ avec $c_{R-1}(0)$.

Où c_j (i) est la position du bit d'entrée de la i-ème sortie après la permutation de la j-ème ligne

C. Si C = p-1 (3.13)

(C-1) Identique au cas A-1.

(C-2) Identique au cas A-2.

(C-3) Identique au cas A-3.

(C-4) Identique au cas A-4.

(C-5) Effectuez la j-ème (j = 0,1, 2, …, R-1) permutation intra-ligne comme suit:

c_j (i) = c([i´ p j]mod(p -1)) -1, i =0,1,2,…, (p-2), (3.14)

Où cj(i) est la position du bit d'entrée de la i-ème sortie après la permutation de la j-ème ligne**.**

Troisième étape

(1) Effectuez la permutation inter-lignes basée sur les motifs P(j) suivants (j=0,1, ..., R-1), où P(j) est la position de ligne originale de la j-ème ligne permutée.

PA : {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} pour R=20

PB : {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} pour R=20

PC : {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} pour R=10

L'utilisation de ces motifs est la suivante : Longueur de bloc K : P(j) 320 à 480 bits : PA 481 à 530 bits : PC 531 à 2280 bits : PA 2281 à 2480 bits : PB 2481 à 3160 bits : PA 3161 à 3210 bits : PB 3211 à 5114 bits : PA

(2) La sortie de l'entrelaceur mère est la séquence lue colonne par colonne à partir de la matrice permutée R × C. [9]

III.7 Paramètre de d'entrelacement

III.7.1 Le délai

Le délai est un élément essentiel dans la conception d'un entrelaceur, surtout dans un monde où la rapidité de transmission des données est cruciale. Comprendre et évaluer précisément ce délai est indispensable pour minimiser la latence introduite par le système. Ce délai englobe le temps nécessaire pour entrelacer et dé-entrelacer une paire de données, ainsi que les opérations sur ces données. On distingue deux types d'entrelaceurs : les entrelaceurs

de permutation et les entrelaceurs multiplexés. Pour les entrelaceurs de permutation, le délai est déterminé par le remplissage de la matrice avant l'entrelacement et la transmission, ce qui dépend de la taille de l'entrelaceur en termes de nombre de bits et de la vitesse de transmission. En revanche, pour les entrelaceurs multiplexés, le délai est directement lié au schéma en blocs, sans nécessiter de remplissage de matrice [7].

III.7.2 Mémoire

Chaque entrelaceur nécessite de la mémoire pour effectuer son processus d'entrelacement. La causalité est un concept crucial dans ce contexte. Si un entrelaceur n'est pas causal, sa mémoire nécessaire est simplement égale à sa longueur. Cependant, la causalité permet de déterminer la mémoire minimale requise. Un entrelaceur est considéré comme causal si sa sortie à un moment donné ne dépend que des entrées précédentes ou actuelles. Ainsi, la mémoire nécessaire correspond au nombre de bits changeant de position. Les concepts de délai et de mémoire sont donc étroitement liés, la mémoire requise étant au moins égale au délai nécessaire [7].

III.7.3 Le facteur de dispersion (dispersion interleaver)

Pour construire des entrelaceurs pouvant offrir de bonnes performances, il est important d'étudier certaines propriétés, telles que le facteur de dispersion. Le facteur de dispersion est une mesure de la régularité d'une permutation. Avoir une bonne propriété de dispersion consiste à éviter les motifs dans la permutation.

La dispersion d'une permutation est donnée par le nombre d'éléments dans l'ensemble :

$$D(\pi)=|\{(j-i, \pi(j)-\pi(i)) \mid 0\leq i<j<k\}|$$

Où ... est une permutation d'un ensemble avec k éléments. La dispersion normalisée est donnée par :

$$\text{Dispersion normalisée} \frac{2|d(h)|}{k|k-1|} \qquad (3.15)$$

Plus la dispersion normalisée se rapproche de 1, meilleure elle devient.

- **Proposition :** Considérons le monôme permutation $\pi(X) = x^{q-2}$ ou est un nombre premier.

  Puis π normalisé la dispersion γ ou $\frac{q-1}{2q} < \gamma < \frac{q+3}{2q}$

Parmi les monômes étudiés, il y avait = x^{q-2}pour un q nombre premier .nous avons constaté que = x^{q-2} a toujours une dispersion meilleure ou égale que les autres monômes. Nous avons trouvé que si et seulement si 3|q-2|, alors la dispersion des autres monômes est égale à celle de = x^{q-2}

Ces monômes sont x^3 et $x^{\frac{2q-1}{3}}$

- **Conjecture 2 :**Soit q un nombre premier. La permutation de Fq donnée par a une dispersion égale à la borne supérieure $\gamma = \frac{q+3}{2q}$et seulement si

  $i \in \{3, \frac{2q-1}{3}, q-2\}$.

  - Pour des exemples, voir le tableau dans la section 5, qui montre la dispersion et la répartition de plusieurs permutations. [15]

III.7.4 L'étalement de l'entrelaceur (spreading interleaver)

La diffusion est une autre propriété importante d'un entrelaceur. C'est une mesure de la distance entre les symboles entrelacés qui étaient initialement proches les uns des autres. On dit qu'un entrelaceur a des facteurs de diffusion (s; k) si :

$|i - j| < (s) \qquad |\pi(i) - \pi(j)| \geq kt$ (3.16)

La diffusion s est la valeur maximale telle que s ≤k Plus s se rapproche de$\sqrt{\frac{k}{2}}$, où T est la longueur du bloc qui est permuté, meilleure est la diffusion.

La prochaine conjecture relie la diffusion et la forme de q.

- **Conjecture 3 :** Soit q un nombre premier. Si 3|q-2|, alors la permutation donnée par les monômes x^3 et $x^{\frac{2q-1}{3}}$a une diffusion supérieure à 1 si et seulement si q est de la forme 30k + 11 ou 30k + 29, pour k appartenant à l'ensemble des entiers. La diffusion de la permutation donnée par x^{q-2} est supérieure à 1 si et seulement si q n'est pas de la forme mentionnée ci-dessus.
  - Pour des exemples, consultez le tableau dans la section 5, qui montre la dispersion et la diffusion de plusieurs permutations. [15]

III.8 Advantage de l'entrelaceur

En théorie des codes correcteurs d'erreurs, plusieurs caractéristiques des mots de code sont définies. Deux des propriétés les plus importantes sont la distance de Hamming et le poids de Hamming. La distance de Hamming entre deux mots de code représente le nombre total de positions où ces deux mots diffèrent. Le poids de Hamming d'un mot de code est le nombre total de positions où il diffère du vecteur nul. La qualité d'un code est souvent déterminée par une propriété appelée la distance minimale, qui mesure la distance la plus courte entre deux mots de code dans l'ensemble du code.

La distance minimale d'un code est définie comme la plus petite distance entre les mots de code distincts, ce qui fournit une indication de la capacité du code à détecter et corriger les erreurs. Un code de distance minimale $\boldsymbol{d}$ peut détecter jusqu'à (d min -1) erreurs et peut corriger un nombre maximum d'erreurs, vérifiant ainsi la relation suivante :

$$d_{min} > 2t+1 \text{ ou } t = \text{floor}\left(\frac{d_{min}-1}{2}\right) \qquad (3.17)$$

En garantissant une distance minimale suffisamment grande, l'entrelaceur contribue à améliorer la robustesse du système de communication, en assurant une transmission plus fiable des données même en présence de perturbations ou de bruit dans le canal de communication. Ainsi, la distance minimale dans l'entrelaceur joue un rôle crucial dans la conception et la performance des systèmes de communication numérique.

III.8 Conclusion

En conclusion, l'entrelaceur joue un rôle fondamental dans le contexte des turbos codes, Les paramètres des entrelaceurs ont été étudiés. Nous avons ainsi vu la notion de délai, celle de mémoire, mais surtout le facteur d'étalement qui sera une notion primordiale dans notre étude des entrelaceurs pour les Codes Turbos.

Chapitre IV
Application et discussion des résultats

IV. 1 Introduction

Ce chapitre présent les résultats obtenus. Commençant par une présentation générale d'un programme de codage et décodage.

La simulation des systèmes des communications est un moyen efficace et rapide pour mettre en relief les performances et les principales difficultés de conception de ces derniers. En se servant d'un outil de simulation comme Matlab/Simulink il est possible de modéliser des chaînes de transmissions, et analyser par exemple les performances en termes de bit erreur rate, on à utilisé le turbo code de deux code convolutif (13,15) et on utilise 3 registre. Après on à étudier l'effet de étalement et dispersion par rapport le turbo décodage on utilisent le programme Matlab et on utilisent les deux trames 64 et 128 qui sont des courts trames (ils ont un peu de délai de traitement parce que qu' na diminue la taille de trame donc on a utilisé dans 4G ,5G et 6G) (4G, 5G et 6G ils nous obligent à utilisée les courtes trames pour diminue le délai de traitement).

IV.2 Schéma de bloc

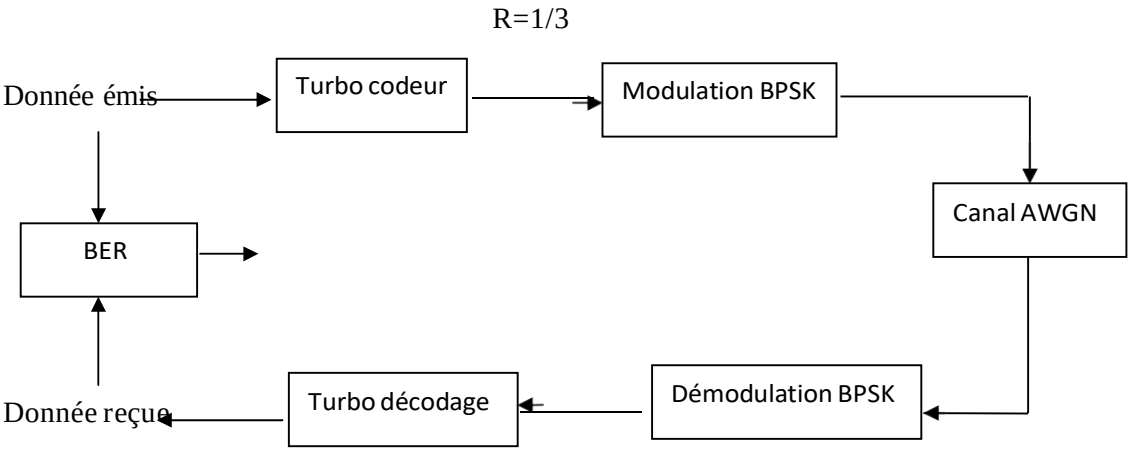


Figure 4.1 : Schéma de bloc d'une chaine de transmission

- Après envoyer donnée émis et recevoir donnée reçue**,** on à comparé donnée émie par rapport donnée reçue pour calculer le BER.
- $BER=\frac{\text{Nombre de bits érronées}}{\text{Nombre de bit transmis}}$

IV.3 Schéma du turbo codeur

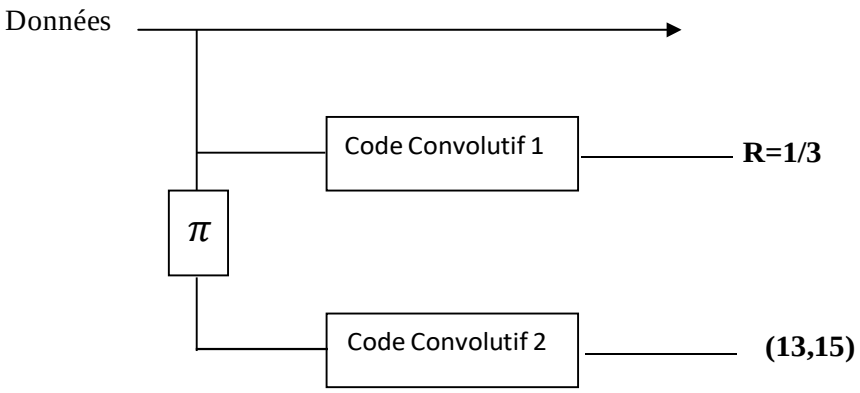


Figure 4.2 : Schéma de turbo code

IV.4 Schéma de code convolutif

On a utilise le turbo code de deux code convolutif (13,15) par 3 registres :

$(13)_2$ = **1101**

$(15)_2$= **1111**

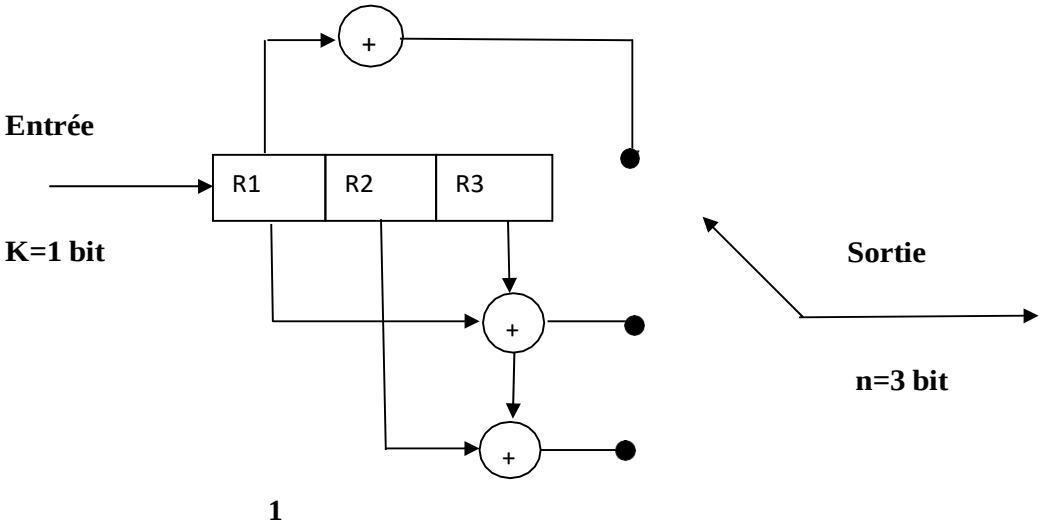


Figure 4.3 : Schéma de code convolutif

IV.5 Schéma du turbo décodeur

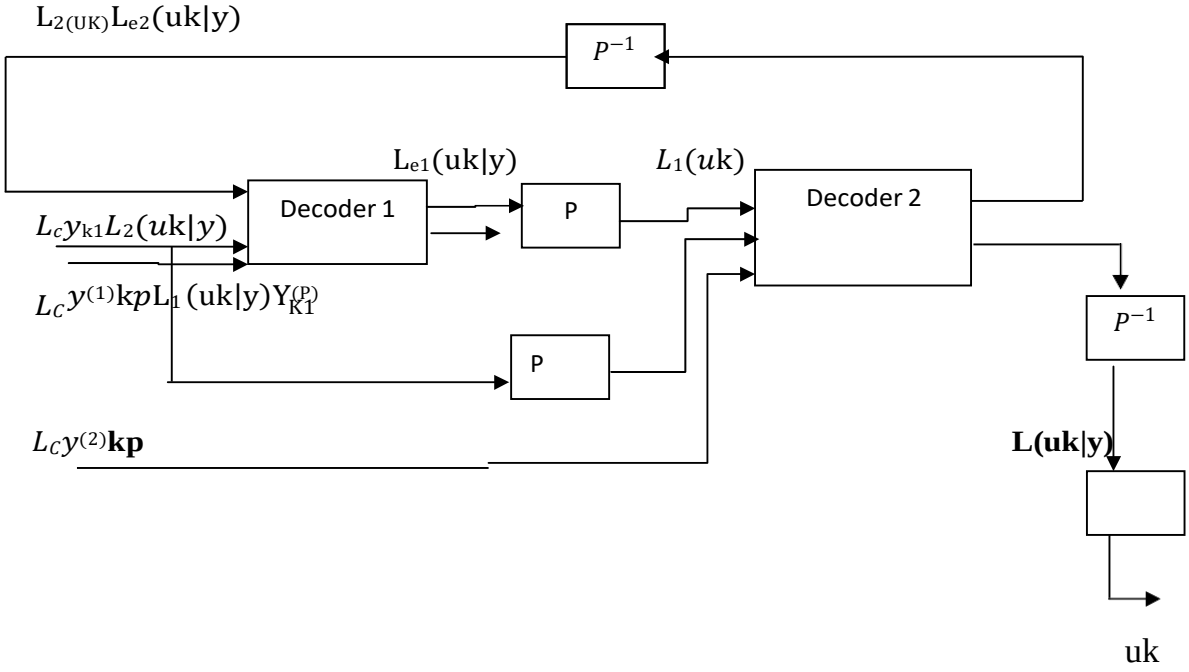


Figure 4.4 : Un Schéma fonctionnel simplifié de turbo décodeur

IV.6Méthodologie de simulation

Les étapes de programme MATLAB :

- Spécifier le nombre d'iteration = 6.
- Spécifier la longueur de trame 64 et 128
- Maximal nombre d'erreurs =110 dans ce programme on a utilisé 110(le nombre d'erreurs maximal auquel nous arrivons a arrêter l'exécution du programme).
- Maximal nombre de bloc
- On fait le choix de l'entrelaceur
- Spécifier le code convolutif utilisé (le nombre de registre v=4 , R=v-1, R=4-1 R=3)
- On défini le $\frac{E_b}{N_0}$ utilisé, (déterminer l'intervalle $\frac{E_b}{N_0}$ utilisés)
- Tout d'abord, En Générée le donnée puis encodées à l'aide d'un turbo code. Ensuite, elles sont transmises via une modulation BPSK (Binary Phase Shift Keying). Les

données modulées entrent dans un canal AWGN (Additive White Gaussian Noise). Après la transmission, une démodulation BPSK est effectuée. Ensuite, les données sont décodées à l'aide du turbo décodage. Enfin, le taux d'erreur binaire (BER) est calculé.

- Nous utilisons l'entrelaceur QPP F(x) =(ax^2+bx) mod N.

IV.7Résultats et discussion

Les résultats de simulation sont donnés par les figures suivantes :

Dans le trame k=64 nous allons étudier 4 cas :

1er cas : en fixant l'étalement S=4 et variant la dispersion, les résultats de simulation sont les suivants tableaux 4.1.

Tableau 4.1 : Les valeurs de BER pour S=4

Dispersion	**Dmin**	**Dmax**
Valeur	0.128968	0.261905
a,b	a=17 b=24	a=47 b=60
Eb/No	0:4.4	0:4.4
BER	1.0063e-01	9.8752e-02
	8.4964e-02	8.2421e-02
	6.9445e-02	6.6549e-02
	5.5492e-02	5.2755e-02
	4.3544e-02	4.0301e-02
	3.2956e-02	2.9760e-02
	2.4333e-02	2.1218e-02
	1.7959e-02	1.4307e-02
	1.2460e-02	9.7762e-03
	8.5202e-03	6.4300e-03
	5.5395e-03	3.9742e-03
	3.6305e-03	2.2827e-03
	2.2714e-03	1.2380e-03
	1.4363e-03	7.7788e-04
	8.1392e-04	4.2281e-04
	6.3747e-04	2.4693e-04
	3.3713e-04	1.2900e-04
	2.3354e-04	5.8516e-05
	1.4595e-04	2.9162e-05
	9.1877e-05	1.5424e-05
	5.8132e-05	9.3309e-06
	2.7471e-05	4.3342e-06
	1.9842e-05	2.4621e-06

Les résultats du tableau4.1 sont représentés sur la figure 4.5 suivante :

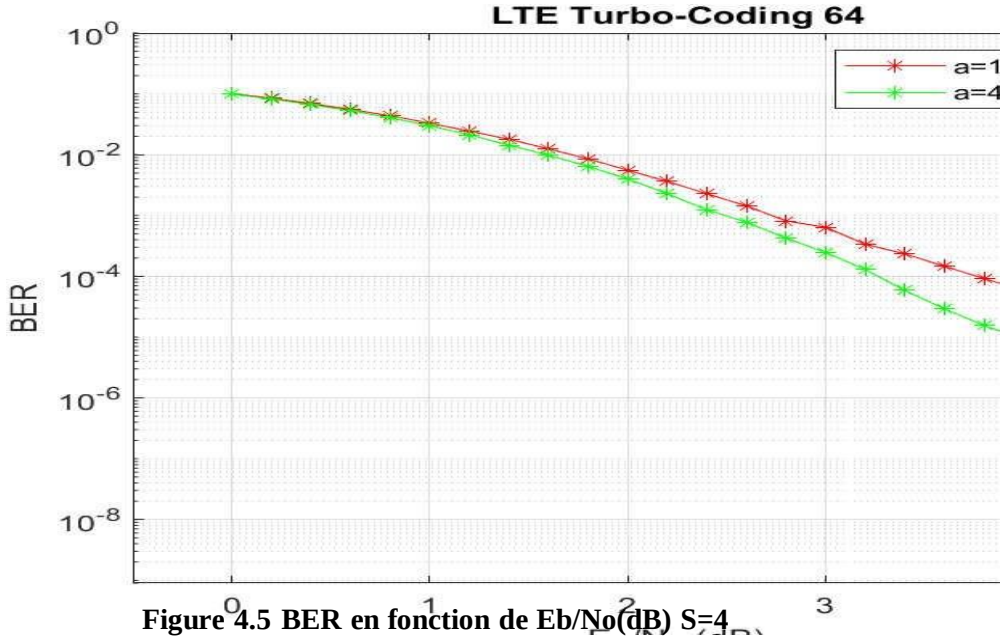


Figure 4.5 BER en fonction de Eb/No(dB) S=4

La figure 4.5montre qu'il y a une amélioration considérable du taux d'erreurs du bloc de code, pour un étalement fixe et une dispersion minimale donne Eb/No=4.4*10^{-6} (dB) et pour une dispersion maximale Eb/No=4* 10^{-6} (dB) on a gagné énergie est égale 0.4 dB alors les performances de turbo code est meilleurs.

2er cas : Nous allons fixer l'étalement S=6 et varie la dispersion, Nous a permet de trouver ces résultats de simulation tableau 4.2 :

Tableau 4.2 : Les valeurs de BER pour S=6

Etalement	**Dmin**	**Dmax**
Valeur	0.0550595	0.087897
a,b	a=37 b=32	a=29 b=48
Eb/No	0:4.4	0:4.4
BER	9.7887e-02	9.8455e-
	8.1909e-02	8.1574e-
	6.6043e-02	6.5362e-
	5.3080e-02	5.2215e-
	3.956	3.9714e-
	2.9188e-02	2.8735e-
	2.0827e-02	1.9743e-
	1.4072e-02	1.3538e-
	9.2700e-03	8.7639e-
	6.0488e-03	5.3770e-
	3.7461e-03	3.7493e-
	2.1261e-03	2.0202e-
	1.2506e-03	1.0603e-
	7.0691e-04	6.4697e-
	3.5200e-04	2.7343e-
	1.8431e-04	1.6517e-
	8.1981e-05	6.8179e-
	3.9596e-05	4.0348e-
	2.0952e-05	1.5036e-
	8.2243e-06	9.9152e-
	4.7162e-06	4.3890e-
	2.2345e-06	2.4425e-
	9.5883e-07	9.3416e-

Les résultats du tableau 4.2 sont représentés sur la figure 4.6 suivante :

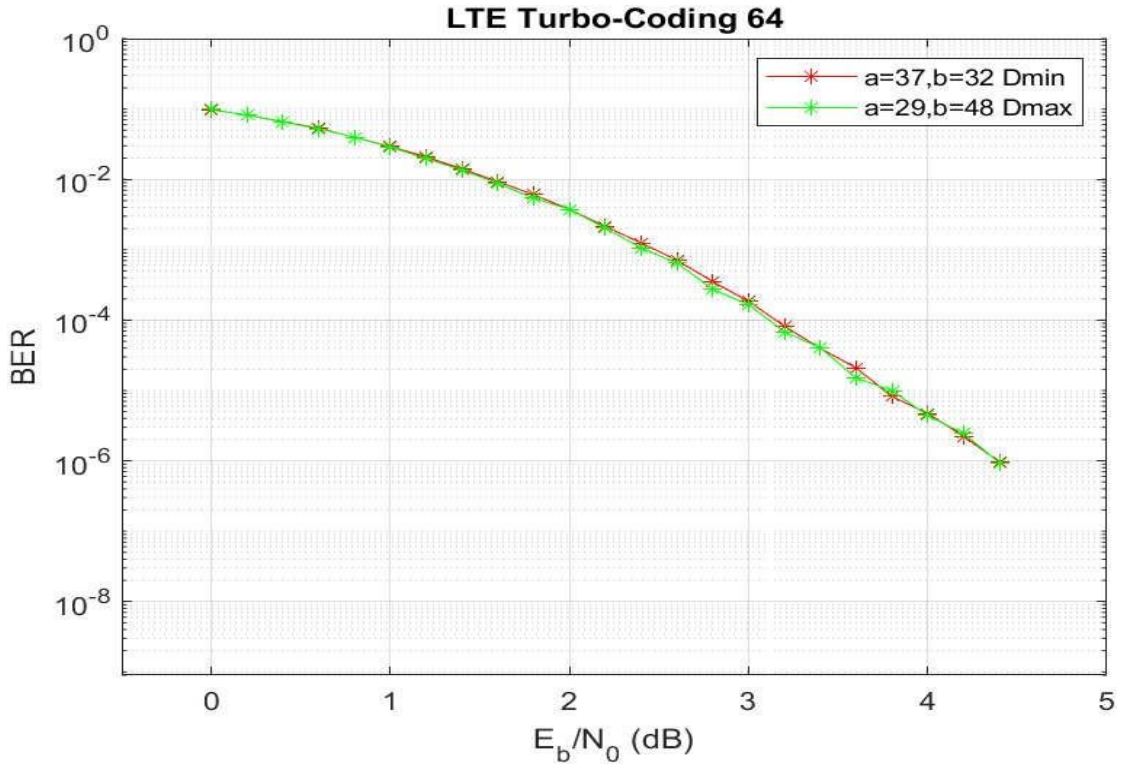


Figure 4.6 : BER en fonction de Eb/No(dB) S=6

Dans cette figure les valeurs de dispersions très proche donc on ne remarque rien pour la comparaison.

3er cas : nous allons fixer la dispersion et varie l'étalement pour S=4 S=6 et S=10, Nous a permet de trouver ces résultats de simulation tableau 4. 3 :

Tableau 4.3 : Les valeurs de BER pour différent étalement

Étalement	**S=4**	**S=6**	**S=10**
dispersion	**D**	**D**	**D**
Valeur	0.0610119	0.0615079	0.0600198
a,b	a=29 b=32	a=19 b=32	a=5 b=32
Eb/No	0:4.8	0:4.6	0:4.6
BER	1.0099e-01	9.8343e-02	9.9383e-02
	8.4461e-02	8.1581e-02	8.3012e-02
	6.8952e-02	6.5431e-02	6.7758e-02
	5.4205e-02	5.1693e-02	5.3839e-02
	4.1976e-02	3.9158e-02	3.9567e-02
	3.1519e-02	2.8807e-02	2.9047e-02
	2.2820e-02	2.0287e-02	2.1143e-02
	1.6141e-02	1.3588e-02	1.4725e-02
	1.0760e-02	9.4700e-03	9.2357e-03
	6.8643e-03	5.9019e-03	5.8145e-03
	4.9178e-03	3.5274e-03	3.7149e-03
	2.8744e-03	2.1121e-03	1.9423e-03
	1.8961e-03	1.2580e-03	1.0260e-03
	1.0164e-03	7.3282e-04	5.1451e-04
	5.4013e-04	2.8924e-04	2.9647e-04
	3.2931e-04	1.4965e-04	1.0532e-04
	2.0580e-04	8.5376e-05	5.9821e-05
	1.3672e-04	3.2857e-05	2.6144e-05
	6.4547e-05	1.6890e-05	1.0061e-05
	4.3319e-05	7.9041e-06	5.6980e-06
	2.6115e-05	4.0059e-06	2.0648e-06
	2.6115e-05	1.9259e-06	7.9649e-07
	9.3416e-07		

Les résultats sont affichés sur la figure 4.7 :

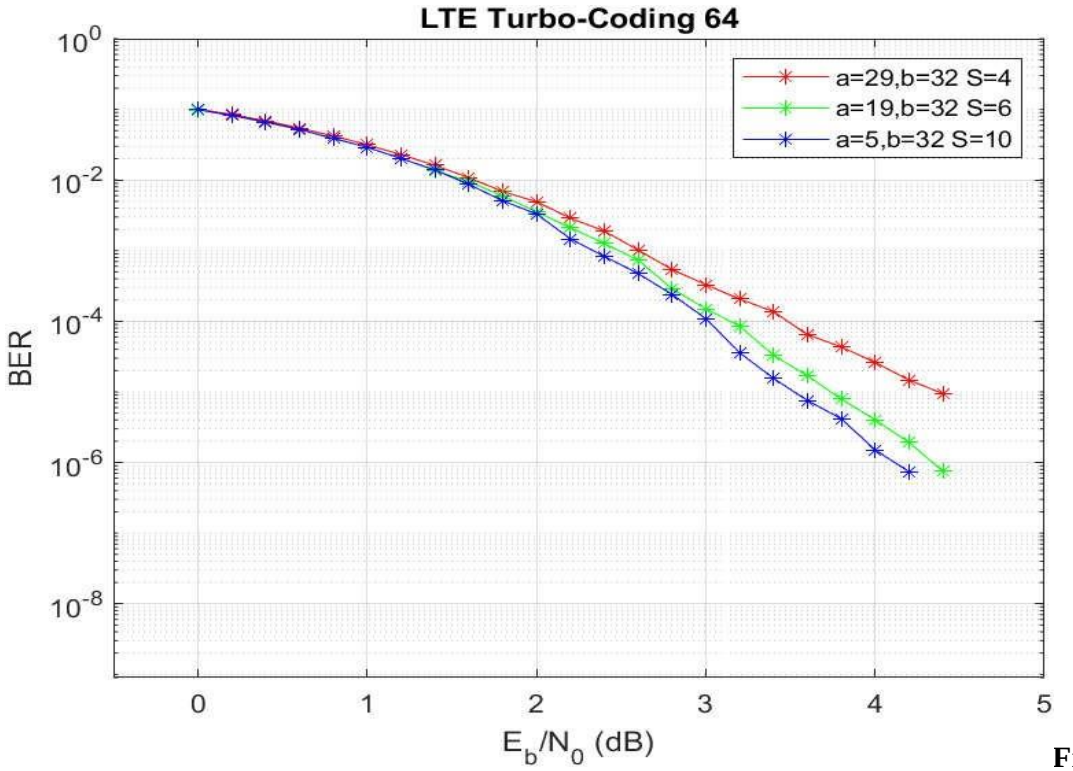


Figure 4.7 : BER en fonction de Eb/No(dB) S=4 S=6 S=10

Nous remarquons après chaque augmentation d'étalement nous enregistrons des bonnes performances quand le BER=10^{-5}et un étalement=4 consommée énergie Eb/No=4.4 (dB) et étalement =6 donne Eb/No= 3.8(dB) et le dernier Eb/No= 3.6(dB)

entre S=4 et S=10 nous avons gagné beaucoup d'énergie égale 0,8(dB).

4er cas : Nous avons prendre un étalement minimal avec dispersion minimale et étalement maximal avec dispersion maximale, Nous a permet de trouver ces résultats de simulation tableau 4. 4 :

Tableau 4.4 : Les valeurs de BER de déférente étalement et dispersion

Etalement	**Dmin**	**Dmax**
Valeur	0.0520833	0.0600198
a,b	a=11 b=32	a=5 b=32
Eb/No	0:4.6	0:4.4
BER	1.0078e-01	9.8618e-02
	8.4218e-02	8.1865e-02
	6.8599e-02	6.6202e-02
	5.4374e-02	5.1855e-02
	4.1376e-02	3.8611e-02
	3.0822e-02	2.9000e-02
	2.2589e-02	2.0083e-02
	1.5969e-02	1.3869e-02
	1.0345e-02	8.6701e-03
	6.9642e-03	5.0646e-03
	4.7053e-03	3.2993e-03
	3.0494e-03	1.4630e-03
	1.7886e-03	8.2768e-04
	8.5539e-04	4.7285e-04
	4.5502e-04	2.3980e-04
	2.8929e-04	1.0856e-04
	1.3945e-04	3.5739e-05
	8.8916e-05	1.5490e-05
	3.6595e-05	7.5590e-06
	1.8096e-05	4.1471e-06
	1.0345e-05	1.5057e-06
	4.4219e-06	7.3970e-07
	1.7702e-06	

Les résultats du tableau 4.4 sont représentés sur la figure 4.8suivante :

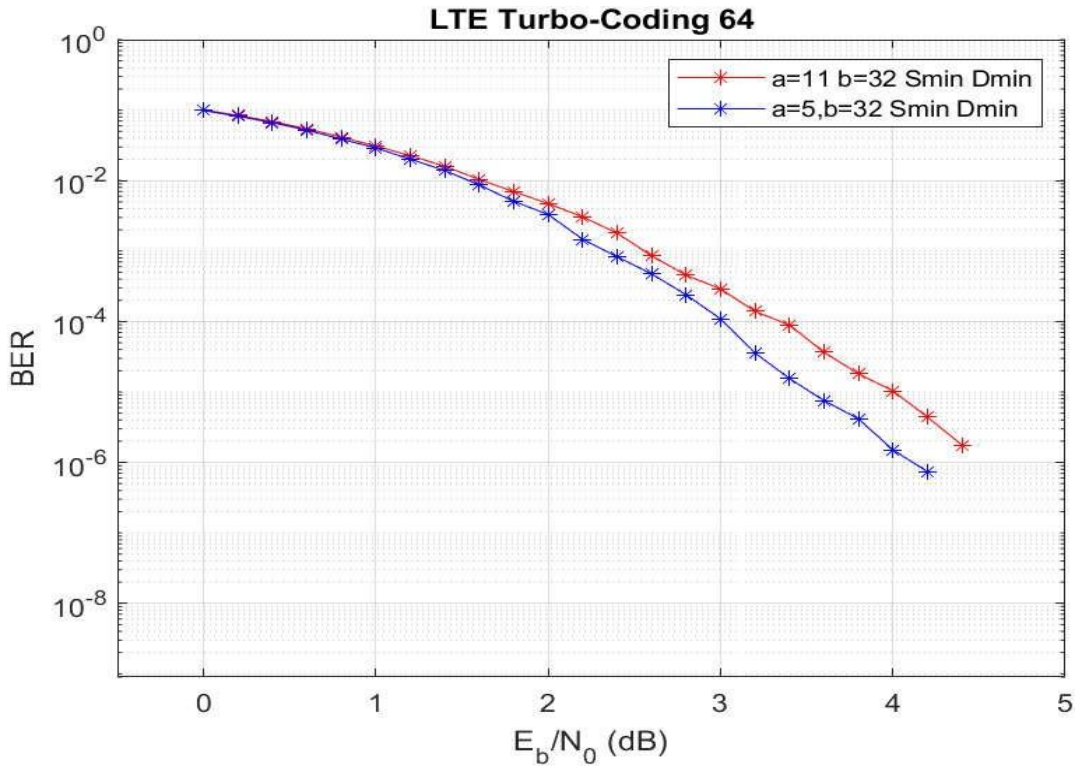


Figure 4.8 : BER en fonction de Eb/No(dB) S=4 S=10

Après la comparaison entre les valeurs minimales et maximales nous remarquons qu'on a arrivé à une meilleure performance de turbo code lorsque l'étalement est maximal avec une dispersion maximale.

Pour la deuxième longueur du trame k=128 nous allons étude 5 cas :

1er cas : On a fixé l'étalement S=4 et varie la dispersion, Nous a permet de trouver ces résultats de simulation tableau 4.5 :

Tableau 4.5 : les valeurs de BER pour S=4 dans trame=128

Etalement	**Dmin**	**Dmax**
Valeur	0.0259596	0.258489
a,b	a=67 b=64	a=31 b=52
Eb/No	0:5.2	0:4.4
BER	9.8043e-02	9.1658e-02
	8.0443e-02	7.1279e-02
	6.4062e-02	5.2215e-02
	4.8650e-02	3.5826e-02
	3.6021e-02	2.3542e-02
	2.5787e-02	1.4897e-02
	1.7831e-02	8.5733e-03
	1.1804e-02	4.6116e-03
	7.6375e-03	2.5401e-03
	4.7459e-03	1.1990e-03
	3.0884e-03	6.0700e-04
	1.8418e-03	2.6553e-04
	1.0786e-03	1.2124e-04
	5.0912e-04	4.9857e-05
	3.1691e-04	2.0081e-05
	1.4284e-04	1.1155e-05
	8.5475e-05	5.8855e-06
	4.9611e-05	2.5577e-06
	2.7354e-05	1.3153e-06
	1.3661e-05	7.3028e-07
	8.5184e-06	
	3.5596e-06	
	1.9263e-06	
	9.5057e-07	

Les résultats du tableau 4.5 sont représentés sur la figure 4.9 suivante :

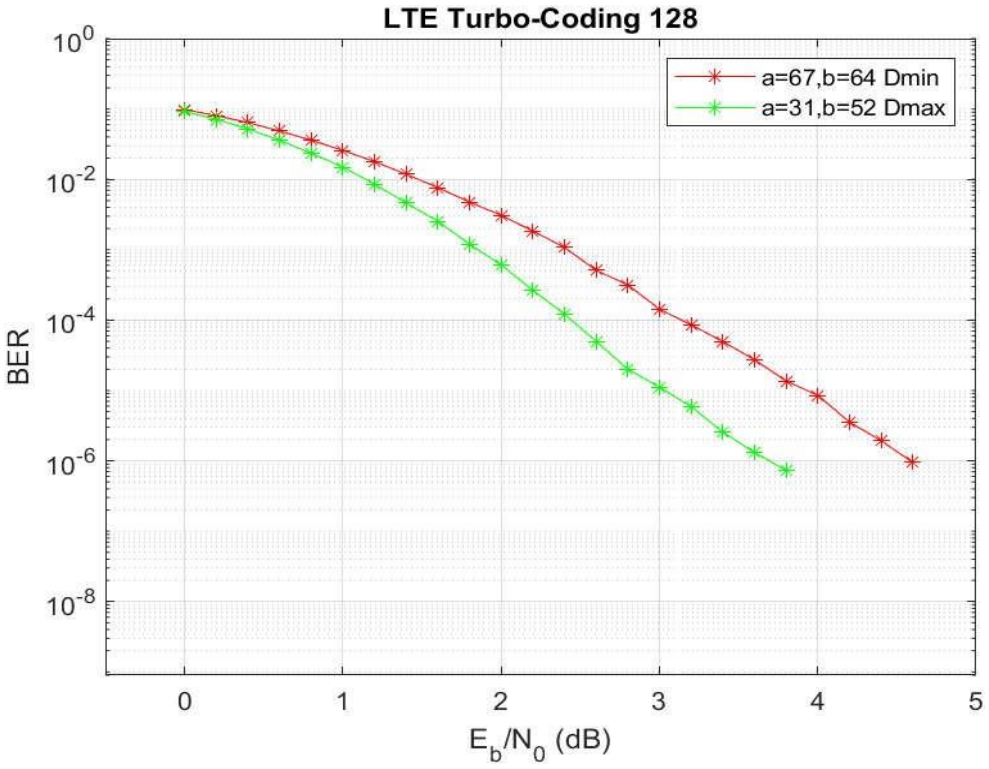


Figure 4.9 : BER en fonction de Eb/No(dB) S=4

Dans cette figure l'étalement fixe S=4 et la dispersion déférente, malgré le trame qui agrandi mais on trouve la même remarque, les performances de turbo code et meilleur pour dispersion maximale.

2eme cas : Nous allons fixer l'étalement S=8 et varie la dispersion nous a permet de trouver ces résultats de simulation tableau 4.6:

Tableau 4.6: Les valeurs de BER pour S=8 et trame128

Etalement	**Dmin**	**Dmax**
Valeur	0.0300197	0.0804626
a,b	a=33 b=64	a=9 b=48
Eb/No	0:4.2	0:3.8
BER	9.2377e-02	9.1507E-02
	7.2382e-02	7.1001E-02
	5.3577e-02	5.2354E-02
	3.6658e-02	3.6355E-02
	2.5087e-02	2.4034E-02
	1.5308e-02	1.4375E-02
	8.9857e-03	7.8953E-03
	5.0537e-03	4.1304E-03
	2.3589e-03	2.0551E-03
	1.2484e-03	8.8438E-04
	6.6678e-04	3.8459E-04
	2.8718e-04	1.4619E-04
	1.2032e-04	5.2377E-05
	5.0422e-05	2.2161E-05
	3.2911e-05	7.4388E-06
	1.5586e-05	3.6211E-06
	7.3769e-06	1.6266E-06
	4.5932e-06	8.1779E-07
	2.1050e-06	
	1.0612e-06	
	4.9579e-07	

Les résultats du tableau 4.6 sont représentés sur la figure 4.10 suivante **:**

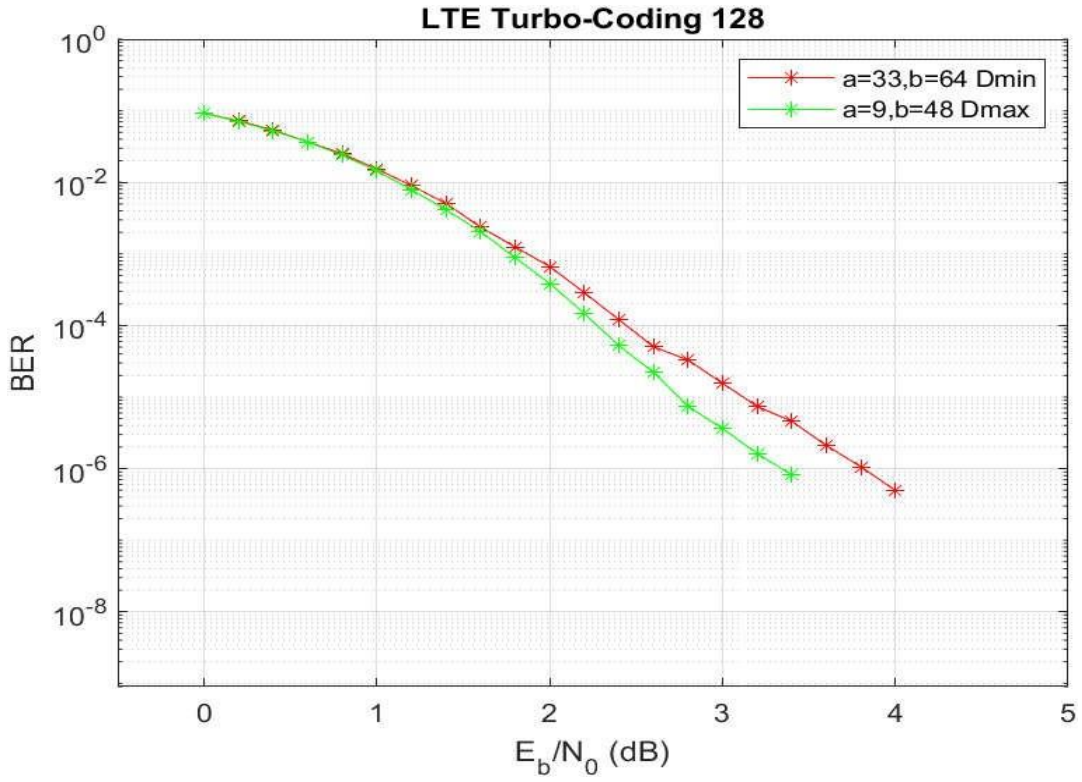


Figure 4.10 : BER en fonction de Eb/No(dB) S=8

Dans cette figure montre qu'il y a une amélioration considérable du taux d'erreurs du bloc de code, pour étalement fixe et BER = 10^{-6}la dispersion minimal donne Eb/No=$4*10^{-6}$ (dB) et dispersion maximale Eb/No=$3.6*10^{-6}$ (dB) on a gagnée énergie est égale 0.4 dB alors les performances de turbo code est meilleurs.

3eme cas : Nous allons fixer l'étalement S=12 et varie la dispersion, Nous a permet de trouver ces résultats de simulation tableau 4.7 :

Tableau 4.7: Les valeurs de BER pour S=12 et trame128

Etalement	**Dmin**	**Dmax**
Valeur	0.0301427	0.0453986
a,b	a=47 b=64	a=15 b=32
Eb/No	0:3.6	0:3.6
BER	9.1986E-02	9.2471e-02
	7.1504E-02	7.0939e-02
	5.2069E-02	5.0551e-02
	3.6210E-02	3.4723e-02
	2.2988E-02	2.2616e-02
	1.3132E-02	1.4219e-02
	7.4688E-03	8.2765e-03
	4.1289E-03	4.0632e-03
	2.0586E-03	2.0721e-03
	9.2806E-04	9.5659e-04
	3.3039E-04	3.7097e-04
	1.4128E-04	1.6802e-04
	5.2343E-05	4.0838e-05
	1.6376E-05	1.3460e-05
	6.6236E-06	5.5251e-06
	2.2517E-06	1.6485e-06
	7.5434E-07	7.5077e-07

Les résultats du tableau 4.7 sont représentés sur la figure 4.11 suivante **:**

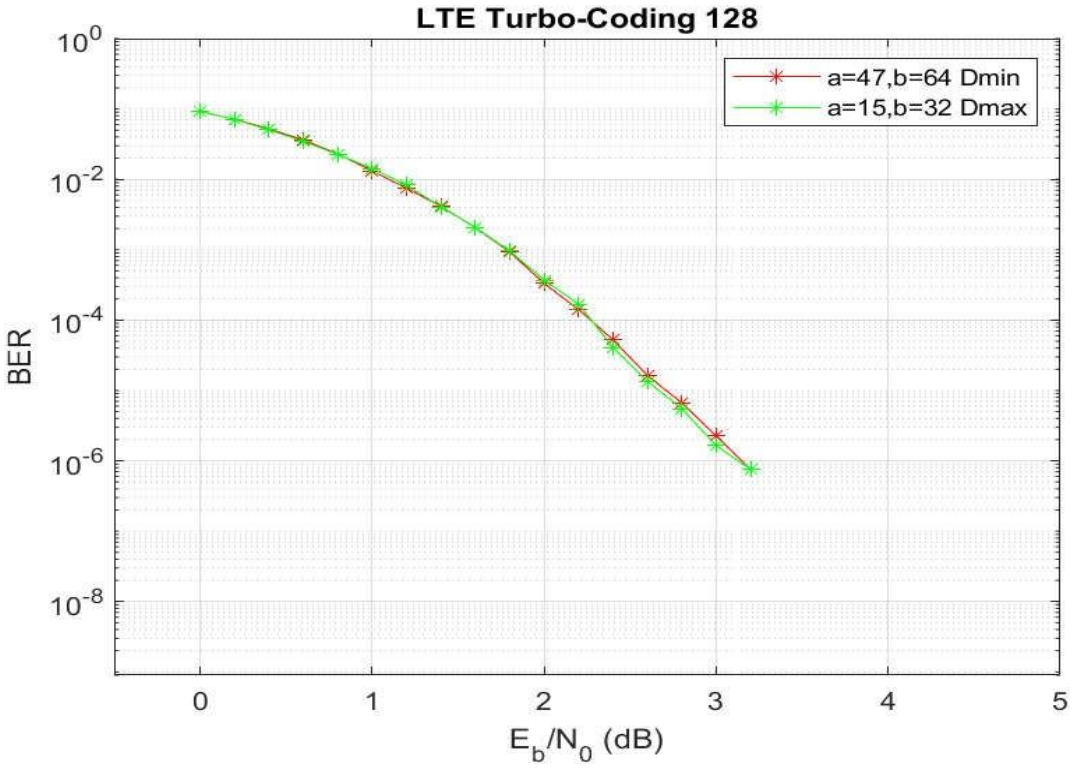


Figure 4.11 BER en fonction de Eb/No(dB) S=12

Dans cette figure les valeurs de dispersions très proche donc on ne remarque rien pour la comparaison.

4eme cas : Nous allons fixera dispersion et varie l'étalement, Nous a permet de trouver ces résultats de simulation tableau 4.8 :

Tableau 4.8: Les valeurs de BER de déférent étalement trame128

Étalement	**S=4**	**S=8**	**S=12**
Dispersion	**D**	**D**	**D**
Valeur	0.0310039	0.0306348	0.0306348
a,b	a=61 b=64	a=95 b=64	a=113 b=64
Eb/No	0:5.6	0.50	0:3.6
BER	9.7738e-02	9.2091e-02	9.1883e-02
	8.0314e-02	7.1353e-02	7.1450e-02
	6.2531e-02	5.2079e-02	5.2827e-02
	4.8022e-02	3.6390e-02	3.5838e-02
	3.5487e-02	2.4439e-02	2.2724e-02
	2.4995e-02	1.5000e-02	1.3921e-02
	1.7042e-02	8.9217e-03	7.5376e-03
	1.1299e-02	4.5631e-03	4.1007e-03
	7.6813e-03	2.3558e-03	1.8159e-03
	4.8928e-03	1.0331e-03	7.0847e-04
	3.1525e-03	5.0144e-04	3.0634e-04
	1.7528e-03	1.9599e-04	1.3117e-04
	1.0842e-03	9.8474e-05	4.7419e-05
	7.2328e-04	4.4118e-05	1.7841e-05
	5.0446e-04	2.1244e-05	5.0260e-06
	2.8998e-04	8.9103e-06	1.6728e-06
	1.7257e-04	4.3718e-06	4.7902e-07
	1.0146e-04	1.8098e-06	
	5.6223e-05	7.9644e-07	
	2.7897e-05		
	1.6855e-05		
	1.1574e-05		
	6.6795e-06		
	3.3493e-06		
	2.0043e-06		
	1.0828e-06		
	4.9225e-07		

Les résultats du Tableau 4.8 sont représentés sur la figure 4.12 suivante :

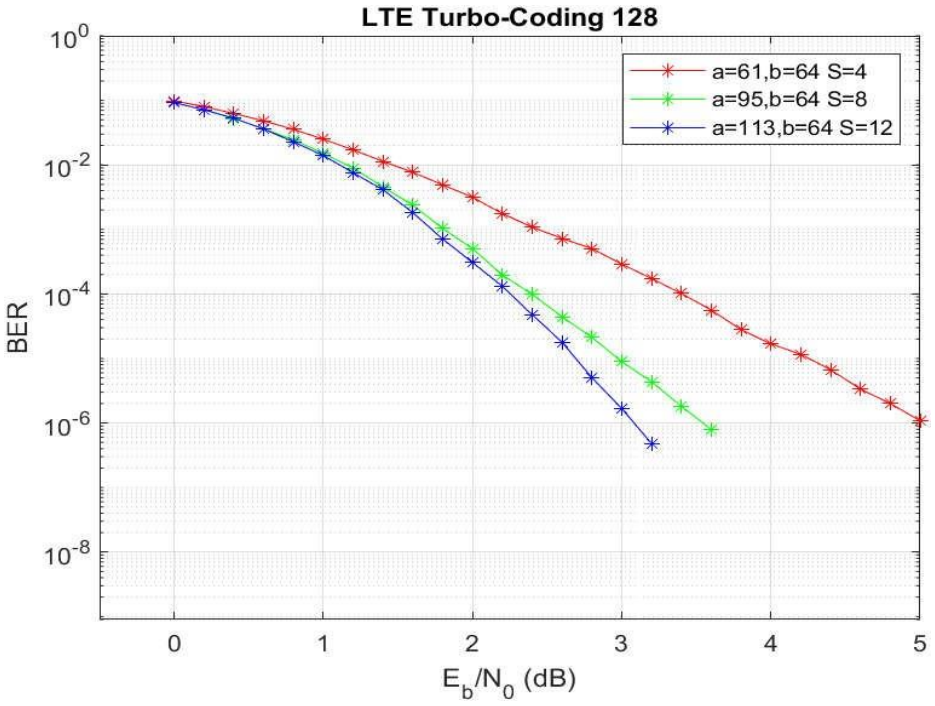


Figure 4.12 : BER en fonction de Eb/No(dB) S=4 S=8 S=12

Après cette exécution nous remarquons après chaque augmentation d'étalement nous enregistrons bon performances quand le BER=2* 10^{-5} étalement=4 consommée énergie Eb/No=4.4 (dB) et étalement=8 donne Eb/No= 3.2(dB) et le dernier Eb/No= 2.8(dB) entre S=4 et S=12 nous avons gagné beaucoup d'énergie égale 1,6(dB).

5eme cas : Nous allons prends étalement maximal avec dispersion maximal et étalement minimal avec dispersion minimal nous permet a de trouver ces résultats de simulation Tableau 4.9 :

Tableau 4.9. Les valeurs de BER de déférente étalement et dispersion trame128

Étalement	**Smin=4**	**Smax=14**
Dispersion	**Dmin**	**Dmax**
Valeur	0.0259596	0.03036348
a,b	a=67 b=64	a=11 b=64
Eb/No	0:5 .	0:3.6
BER	9.8043e-02	9.157400E-02
	8.0443e-02	7.038900E-02
	6.4062e-02	5.146600E-02
	4.8650e-02	3.577300E-02
	3.6021e-02	2.230500E-02
	2.5787e-02	1.365800E-02
	1.7831e-02	7.432900E-03
	1.1804e-02	3.725800E-03
	7.6375e-03	1.702000E-03
	4.7459e-03	8.022400E-04
	3.0884e-03	3.690500E-04
	1.8418e-03	1.475600E-04
	1.0786e-03	5.562100E-05
	5.0912e-04	2.167300E-05
	3.1691e-04	5.035900E-06
	1.4284e-04	2.063800E-06
	8.5475e-05	6.368700E-07
	4.9611e-05	
	2.7354e-05	
	1.3661e-05	
	8.5184e-06	
	3.5596e-06	
	1.9263e-06	
	9.5057e-07	

Les résultats du tableau 4.9 sont représentés sur la figure 4.13 suivante **:**

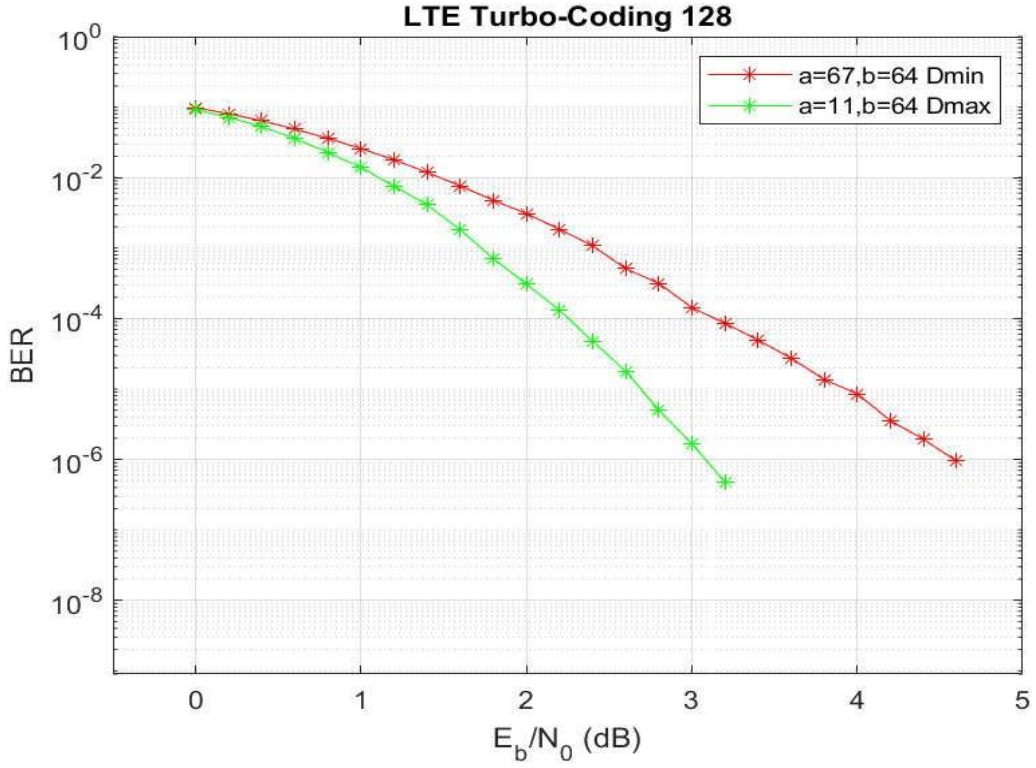


Figure 4.13 : BER en fonction de Eb/No(dB) S=4 S=14

Après la comparaison entre les valeurs minimales et maximales nous remarquons pour meilleur performance de turbo code prends l'étalement maximal avec dispersion maximal.

IV.8 Conclusion

On remarque les points suivants

- On observe que lorsque l'étalement est fixé et que la dispersion varie, la performance s'améliore à chaque augmentation de la dispersion.
- En contraire, lorsque la dispersion est fixée et que l'étalement augmente, la performance s'améliore également.

Conclusion Générale

Conclusion Générale

Ce mémoire a été consacré a l'étude des performances des codes turbo qui sont actuellement parmi les codes les plus performants connus à ce jour.

Dans ce mémoire, nous avons étudié les codes turbo, en particulier les codes turbo convolutionnels parallèles, dans un canal gaussien. Les codes turbo sont parmi les codes correcteurs d'erreur les plus performants connus à ce jour, permettant d'approcher la limite de Shannon.

Dans un premier temps, nous avons concentré notre généralité sur le système de communication numérique et la base pour tout ce qui suit. Nous avons revisité les généralités et les bases de la théorie du codage convolutionnel pour établir un socle solide. Nous nous sommes ensuite concentrés sur les notions fondamentales des turbo codes, en présentant les codeurs convolutionnels systématiques, qu'ils soient récursifs ou non, et leurs propriétés respectives. Nous avons examiné les éléments constitutifs du codage turbo, introduisant la concaténation parallèle des codes convolutionnels systématiques et récursifs, qui forment la base des turbo codes.Nous avons également expliqué ce qui rend les turbo codes uniques : le décodage itératif turbo. L'importance de l'échange d'information entre les décodeurs du système turbo a été analysée en détail.

Les turbo codes représentent une avancée majeure pour les systèmes de codage d'information. Cette technologie est devenue essentielle pour assurer la fiabilité des transmissions numériques de données grâce à son pouvoir de correction très élevé.

A la fin sur la base de la recherche nous pouvons travailler avec des entrelaceurs de grand dispersion et spreading.

Références bibliographiques

Références :

[1] MehtouguiNawel ,Concept de base sur le système de communication par satellite1page .Mémoire de master :Aéronautique , institut d'aéronautique et des études spéciale ,Universités Blida 1,2019

[2]Mr Mehdi HAFSSIL ,Mr Salah BOUHOUN .Introduction au codage de canal :Le Canal à bruit blanc additif Gaussien : modulation BPSK .11page. Mémoire de master : Electronique:Ecole Nationale Polytechnique :2013

[3]BOUHAFS BILAL& YAHIAOUI AMINA .Généralités sur le système de communication numérique :Le codage de canal 6p .Mémoirede master :Électronique :Université SAAD DAHLAB de BLIDA :2017-2018

[4]BOUMENTEL KHEIRA&FEDLAOUI AMINA. Généralités sur le système de communication numérique :Le canal à Bruit Blanc Additif Gaussien (BBAG) 8p.Mémoirede Master : Électronique :Université SAAD DAHLAB de BLIDA :2017/2018.

[5] Mr IFTENE Essedik Etude des structures d'entrelaceurs pour le codage turbo du canal pour l'optimisation des systèmes de communication par satellite ,(Turbo codage) 2015/2016 page 12 , thése de magister Instrumentation spatiale.

[6]ESSELIMANI Chouaib. .ALIOUCHEHouda.Concaténation parallèle des codes CRS :Entrelaceur déterministe 21page .Mémoire de master : Génie Électrique:UNIVERSITE DR. YAHIA FARES DE MEDEA2014-2015

[7]Grégoy Rover.Decodageiteratif et entrelacement :Paramètresd'unentrelaceur 66page .Thèse master :Géniéélctrique :UNIVERSITÉ DE MONTRÉAL .ECole polytechnique montréalNOVEMBRE 1/ 2000.:Maîtrise èsciences appliquées

[8] Silvio A. Abrantes , From BCJR to turbo decoding: MAP algorithms made easier , Simplified versions of the MAP algorithm β , April 2004 ,page8 .

[9]TS 25.212 V2.3.0 (1999-10)UMTS STANDART.3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG)Radio Access Network (RAN);Working Group 1 (WG1);Multiplexing and channel coding (FDD)

[10]ETSI TS 136 212 V11.3.0 (2013-07).LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (3GPP TS 36.212 version 11.3.0 Release 11) ;Turbo code internal interleaver [enligne]. 2013,14page.vol,11,n=3.

[11]Silvio A. Abrantes , From BCJR to turbo decoding: MAP algorithms made easier, Trellis-aidedcalculation of α and β , April 2004 ,page8 .

[12] jorgeCastiñeira Moreira.ESSENTIALS OF ERROR-CONTROL CODING: turbo code : Random Interleavers [enligne] 2006 wiley)] jorgeCastiñeira Moreira.ESSENTIALS OF ERROR-CONTROLCODING: turbo code : Random Interleavers [enligne] 2006,wiley)-libgen.li.521 page

[13]*Lucian Trifina, Daniela Tarniceriu and ValeriuMunteanu.*On Dispersion and Nonlinearity Degree of QPP Interleavers:An Upper Bound for a QPP Interleaver Dispersion. 2012. 398 page Published online: 1 Sep. 2012

[14] International Research Journal of Engineering and Technology (IRJET) .Review paper on study of various Interleavers and their significance.International Research Journal of Engineering and Technology (IRJET).QPPINTERLEAVER.Volume: *Department of Electronics, M.I.T.S, Gwalior, India*05 Issue: 10 | Oct 2018.431page .

[15]Carlos Avenancio-Leon . ANALYSIS OF THE DISPERSION AND SPREADING PROPERTIES OF INTERLEAVERS FOR TURBO CODES:THE DISPERSION FACTOR:THE SPREADING FACTOR.Department of Mathematics, University of Puerto Rico at Humacao

[16] Mohammad Salim, R.P. Yadav, KapilNarwal, Aarti Sharma4.A New Block S-Random Interleaver for ShorterLength Frames for Turbo Codes: Interleavers in Turbo Code:S-Random Interleaver .Article *in* Bulletin of Electrical Engineering and Informatics · November 2013 .Vol. 2, No. 4, December 2013, pp. 293~298 ISSN: 2089-3191 29

[17] Mr IFTENE EssedikEtude des structures d'entrelaceurs pour le codage turbo du canal pour l'optimisation des systèmes de communication par satellite,(Turbo codage) 2015/2016 page 11 , thése de magister Instrumentation spatiale.

[18] Mr IFTENE Essedik Etude des structures d'entrelaceurs pour le codage turbo du canal pour l'optimisation des systèmes de communication par satellite ,(Turbo codage) 2015/2016 page 12 , thése de magister Instrumentation spatiale.

[19] BOUHAFS BILAL et YAHIAOUI AMINAEstimation des performances du TEB des turbo-codes dans un canal AWGN, année2017-2018, Les Codes Systématiques Récursifs (CSR)21 page , Mémoire de MasterSystème des télécommunications.

[21] BOUMENTEL KHEIRA et FEDLAOUI AMINACodageconvolutif concaténé en parallèle pour turbo-code ,Spécialité Systèmes de Télécommunications (Décodage des codes convolutif) 2017-2018 page 18 , Mémoire de Master Mention Électronique .

[20] Mr Mehdi HAFSSIL Mr Salah BOUHOUNCodage de canal : Application des Turbocodes dans un canal satellitaire, Ecole Nationale Polytechnique,(la concaténation parallèle) 47 page , Mémoire en vue de l'obtention du diplôme d'Ingénieur en Electronique, Juin 2013.

[22] ESSELIMANI Chouaib. ALIOUCHE Houda.Etude de la performance de turbo code ,(principe de decodage itératif) page 27, 2 0 1 4 - 2 0 1 5,Systèmes de Télécommunications.

[23]Silvio A. Abrantes , From BCJR to turbo decoding: MAP algorithms made easier , Application of the BCJR algorithm in iterativedecodingApril 2004 ,page8