

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Mention Électronique  
Spécialité Systèmes de Télécommunications

présenté par

BOUMENTEL KHEIRA

&

FEDLAOUI AMINA

# Codage convolutif concaténé en parallèle pour turbo-code

Proposé par : Dr. ANOU Abderrahmane

Année Universitaire 2017-2018

# *Remerciements*

*Tout d'abord, je vœux remercie Dieu pour ses bénédictions qui m'ont aidé à réaliser ce travail.*

*Nous tenons à remercier notre encadreur Mr Anou Abderrahmane, son précieux conseil et son aide durant toute la période du travail.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Nous pouvons remercier mon cher père et ma chère mère qui m'ont toujours supportés et aidés et nous espérons que ma soutenance sera le meilleur cadeau pour eux.*

*“nous voudrions exprimer nous reconnaissances envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de ma démarche. Un grand merci à AMINA, HALIMA, IMAN, NANI, MAYSA, NOURA, SOULEF, HAMID, et surtout IMAD pour ses conseils concernant le style de mon mémoire.”*

# *Dédicace*

*“Maintenant pour ma famille, La première personne que nous tenons à remercier à ma famille c’est mon grand-père ‘Si moh’, pour leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui m’a permis de réaliser les études que je voulais et par conséquent ce mémoire.”*

*“Un grand dédicace pour mon père, son soutien, et pour l’orientation, la confiance, la patience sa présence surtout dans la période satisfaisante.”*

*“Je souhaite particulièrement dédicace a ma collègue Mme BOUMENTEL KHEIRA pour sa précieuse aide à la relecture et à la patiente de mon mémoire. ”*

*“ Je dédie ce travaille à maman, pour sa fatigue avec moi et de leur coopération avec moi toute la durée de ma vie je l'appelle avec miséricorde, que Dieu la bénisse”.*

*Nous dédions ce travail à mes chères sœurs et mes chers frères.*

*, Fédlaoui amina*

---

**ملخص:** يركز هذا المشروع على دراسة وتقييم الأداء للهيكلة الأساسية للتشفير Turbo الموازية من أجل تحسين تصحيح الأخطاء في أنظمة الاتصالات الرقمية. يتم إجراء تشفير Turbo باستخدام تسلسل متوازي لإثنين من الرموز التلافيفية تكرارية ومنهجية عبر entrelaceur. فك التشفير هو تكراري ويتم تنفيذه باستخدام اثنين من أجهزة فك التشفير متسلسلين لتحسين التصحيح من خلال عدد التكرارات. ولهذه الغاية، أجريت محاكاة لقناتين رقميتين عبر القناة Gaussien باستخدام برنامج Matlab/Simulink. أظهرت نتائج المحاكاة قوة تشفير التوربو وكفاءتها لتصحيح الأخطاء.

**كلمات جوهرية:** كود الثنائي؛ Turbo؛ تسلسل متوازي؛ الرموز التلافيفية؛ كود متكرر منهجي؛ مفك تكراري؛ معدل الخطأ؛ نسبة الإشارة إلى الضوضاء.

---

**Résumé :** Ce projet porte sur l'étude et l'évaluation des performances de la structure de base des codes turbo parallèles dans le but 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. le décodage est itératif et s'effectue à l'aide de deux décodeurs en série. Afin d'améliorer la correction au fil des itérations. Pour cela, une simulation de deux chaînes de transmission numérique sur canal Gaussien a été effectuée en utilisant le logiciel Matlab/Simulink. Les résultats de simulations ont montré la robustesse du turbo codes et leur efficacité de correction d'erreurs.

**Mots clés :** codes turbo ; concaténation parallèle ; codes convolutifs ; code systématique récursif ; décodage itératif ; TEB ; rapport signal sur bruit.

---

**Abstract:** This project focuses on the study and performance evaluation of the basic structure of parallel turbo codes in order to improve error correction in digital communication systems. Turbo coding is done using a parallel concatenation of two recursive and systematic convolutional encoders across an interleaver. The decoding is iterative and is performed using two decoders in series. To improve the correction over the iterations. For this, a simulation of two Gaussian channel digital transmission channels was carried out using the Matlab / Simulink software. The results of simulations showed the robustness of the turbo codes and their error correction efficiency

**Keywords :** turbo codes; parallel concatenation; convolutional codes; recursive systematic code; iterative decoding; BER; signal to noise ratio.

---

# Liste des Symboles et abréviations

## Liste des Symboles

$B$	Bande passante
$C$	Capacité de canal
$P_E$	la probabilité d'erreur de décodage
$\phi_k$	La phase a l'instant $k$
$\sigma^2$	Variance du bruit dans le canal
$N_0$	la densité spectrale de puissance de bruit
$f(n)$	La densité de probabilité du bruit additif blanc gaussien
$d_k$	Bit d'information à l'instant $k$
$m$	Mémoire d'un code convolutionnel
$m + 1$	Longueur de contrainte d'un code convolutionnel
$g_{ij}$	polynôme générateur
$R$	Taux de codage de code convolutionnel
$L(d_k)$	Le logarithme de rapport de vraisemblance
$I$	Entrelaceur
$I^-$	Désentrelaceur
$P$	Matrice de perforation
$E_b$	Energie du signal par bit
$N_0$	Densité spectrale de puissance de bruit
$X_k^s$	Symbole systématique à la sortie de l'encodeur turbo
$X_k^p$	symbole de parité à la sortie de l'encodeur turbo

## Liste des abréviations

<b>BPSK</b>	Binary Phase Shift Keying
<b>AWGN</b>	Additive White Gaussian Noise
<b>BBAG</b>	Canal à Bruit Blanc Additif Gaussien
<b>CNS</b>	Codes Non Systématique
<b>NSC</b>	Non Systematic Convolutional code
<b>CSR</b>	Codes convolutionnels Systématiques Récursifs
<b>RSC</b>	Code convolutif récursif systématique
<b>LRV</b>	Logarithmes de Rapport de Vraisemblance
<b>ESA</b>	European Space Agency
<b>NASA</b>	National Aeronautics and Space Administration
<b>ADSL</b>	Asymmetric digital subscriber line
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>DVB-S</b>	Digital Video Broadcasting – Satellite
<b>HDTV</b>	High-definition television
<b>TEB</b>	Taux d'Erreur Binaire
<b>BER</b>	Bit Error rate
<b>CBER</b>	Code Block Error rate
<b>APP</b>	Probabilité à Posteriori

# Table de matières

Introduction générale .....	1
Chapitre 1 :Généralités sur les systèmes de communications numériques.....	3
1.1. Introduction.....	3
1.2. Théorie de l’information.....	3
1.3. Système de transmission de l’information numérique .....	4
1.3.1 .Notion de message numérique .....	5
1.4 .Le codage.....	5
1.4.1. Le codage de source.....	5
1.4.2. Le codage de canal.....	5
1.5. La modulation .....	6
1.6. Le canal de transmission .....	7
1.7. Conclusion .....	8
Chapitre 2 :Turbo-code.....	9
2.1. Introduction.....	9
2.2. Les codes convolutionnels.....	9
2.2.1. Principe de base .....	9
2.2.2 .Type de codeur convolutionnel .....	12
2.2.3 Représentation graphique des codes convolutifs.....	14
2.2.3. Décodage des codes convolutif .....	17
2.3. Association de plusieurs codes : concaténation.....	18
2.4. L’entrelacement .....	18
2.5. Turbo-code.....	19
2.5.1. Turbo-code série .....	19
2.5.2. Turbo-code parallèle .....	19
2.6. Applications des turbo-codes.....	24
2.6. Conclusion .....	24

Chapitre 3 :Simulation et résultats .....	25
3.1. Introduction.....	25
3.2. Méthodologie de simulation .....	25
3.3. Simulations et résultats.....	28
3.3.1. Simulink à blocs de code de taille fixe .....	28
3.3.2. Simulink à blocs de code de taille variable .....	31
3.4. Conclusion .....	33
Conclusion général .....	34
Bibliographie.....	36



## Liste des figures

Figure 1. 1: l'exposant de l'erreur. ....	4
Figure 1. 2 : Principe d'une chaîne de communication numérique. ....	5
Figure 1. 3: Diagramme de constellation pour BPSK .....	7
Figure 1. 4 : Modèle de Canal gaussien. ....	8
Figure 2. 1: Principe d'un codeur convolutionnel. ....	10
Figure 2. 2: codeur convolutionnel $R = 1/2$ . ....	11
Figure 2. 3: Codeur non systématique CNS. ....	13
Figure 2. 4 : Codeur convolutionnel récursif systématique. ....	13
Figure 2. 5 : Représentation en arbre de l'encodeur. ....	15
Figure 2. 6 Diagramme en treillis. ....	16
Figure 2. 7. Diagramme d'état de l'encodeur de la figure 2.3. ....	17
Figure 2. 8 : Le principe de la concaténation de deux codes. ....	18
Figure 2. 9 : Schéma de principe turbo-code série. ....	19
Figure 2. 10 : Schéma de principe d'un encodeur Turbo. ....	20
Figure 2. 11 : Principe de décodage itératif. ....	21
Figure 2. 12 : Schéma de principe d'un décodeur turbo. ....	23
Figure3. 1: La procédure de simulation. ....	27
Figure3. 2: Modèle d'une chaîne de transmission avec un turbo-code parallèle à blocs de code de taille fixe. ....	29
Figure3. 3 : le bloc de paramétrage. ....	30
Figure3. 4: BER en fonction de $E_b/N_0$ . ....	31
Figure3.5: Modèle d'une chaîne de transmission avec un turbo-code parallèle à blocs de code de taille variable. ....	32
Figure3. 6 : CBER en fonction de $E_b/N_0$ . ....	33

## Liste des tableaux

Tableau 2. 1: codage d'une séquence d'information donnée. ....	12
----------------------------------------------------------------	----

# **Introduction générale**

# Introduction générale

La communication numérique a transformé le monde en un petit village en raison de la rapidité de la communication à tout moment et de n'importe quelle zone, il est devenue un outil incontournable pour toute association et répond à de nombreux besoins : gestion de l'association, travail en équipe, communication interne et externe.

L'évolution des systèmes des communications amène leurs concepteurs à devoir traiter des volumes d'information en constante augmentation pour assurer à l'utilisateur une qualité de service acceptable. Cette évolution motive la recherche de nouvelles techniques de communications, performantes et peu coûteuses. Parmi ces techniques, on s'intéresse au codage correcteur d'erreurs. On a plus spécifiquement, mené l'étude sur la technique de codage turbo, qui est permet d'avoisiner la limite théorique de correction.

La théorie de l'information est basée principalement sur les travaux de **ClaudShannon [2]** en 1948. Il établit la limite théorique de performance pour les systèmes de communication. Il démontre qu'on peut atteindre cette limite utilisant des codes correcteurs d'erreurs performantes. Il a fallu des dizaines d'années pour approcher à ces limites fixées il y a presque un demi-siècle. Ce n'est qu'en 1993 que nous avons pu approcher de cette limite grâce à l'invention de **turbo-code**.

Ces codes inventés par Claude Berrou et Alain Glavieux[5] sont obtenus par concaténation de deux ou plusieurs codes convolutifs de faible complexité, séparés par une fonction d'entrelacement introduisant de la diversité. Leur décodage fait appel à un processus itératif (ou turbo) utilisant deux ou plusieurs décodeurs élémentaires qui s'échangent des informations de fiabilité, appelées informations extrinsèques, afin d'améliorer la correction au fil des itérations.

Dans ce mémoire, nous présentons essentiellement le principe de fonctionnement des codes convolutifs concaténé en paires pour les turbos codes et leurs performances sur un canal Gaussien.

Ce mémoire comporte trois chapitres :

Dans le premier chapitre on expose quelques généralités relatives aux systèmes de communications numérique, et on présente ses composants notamment le codage de canal. Vers la fin de ce chapitre, on mentionne le type de canal de transmission utilisé.

Le deuxième chapitre on introduit le principe du codage convolutionnel et on décrit en particulier les codes convolutionnels récurrents et systématiques, qui sont à la base des turbos codes. Par la suite, on présente le principe de fonctionnement des turbos codes parallèles et le principe de décodage itératif.

Le troisième chapitre expose les résultats de simulation et analyse les performances des turbos codes selon différents paramètres en utilisant le Simulink de Matlab. Il consiste à simuler la structure de base des turbo-codes parallèle, on utilise deux structures de chaînes de transmission à blocs de codes de taille fixe et de blocs de codes de taille variable.

Enfin, ce mémoire est achevé par une conclusion générale inspirée de l'étude menée tout au long de ce travail.

# **Chapitre 1**

## **Généralité sur les systèmes de communications numériques**

# Chapitre 1

## Généralités sur les systèmes de communications numériques

### 1.1. Introduction

Le but principal d'un système de communication numérique est de transmettre l'information avec un maximum d'efficacité, mais l'existence des bruits dans les canaux de communication causent des erreurs pendant la transmission du signal donc le signal reçu peut occasionnellement différer de celui transmis.

La probabilité de ces erreurs étant fonction du rapport signal à bruit  $E_b/N_0$ . Pour améliorer la qualité d'une transmission numérique on peut accroître le rapport signal à bruit comme première approche, c'est-à-dire par exemple : augmenter la puissance émise ou diminuer le facteur de bruit du récepteur. Malheureusement cette solution est souvent très coûteuse, une seconde approche, consiste à coder le message à transmettre [1].

### 1.2. Théorie de l'information

La théorie de l'information fondée par Claude E. Shannon en 1948, fournit un ensemble de développements théoriques qui permettent l'évaluation de la capacité des canaux de communication [2].

Shannon assure dans ce théorème que :

Pour une source à débit d'information  $R$  bit/s et un canal de capacité  $C$  bit/s, si  $R < C$  il y aura un code ayant des mots d'une longueur  $n$ , de sorte que la probabilité d'erreur de décodage  $P_E$  soit :

$$P_E \leq 2^{-nE(R)} \quad (1.1)$$

Avec :

- $n$  : La longueur du mot code
- $E(R)$ : Fonction non négatif (figure 1.1) appelée l'exposant de l'erreur

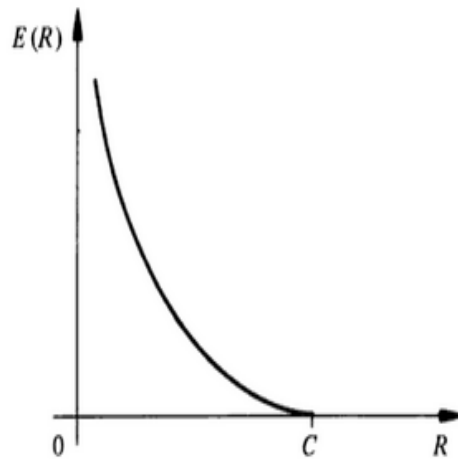


Figure 1. 1: l'exposant de l'erreur [2].

Shannon a démontré qu'il était possible de transmettre un message sur un canal de transmission avec une probabilité d'erreur arbitrairement faible si le débit d'information à l'entrée du canal était inférieur à sa capacité. On cherche donc à construire des codes ayant un taux le plus élevé que possible (pour des raisons de temps et de coût) et permettant une grande fiabilité [13].

### 1.3. Système de transmission de l'information numérique

Le rôle d'un système de transmission de l'information numérique est d'envoyer une information d'un milieu à un autre pour cela on a besoin de deux blocs; le premier est un émetteur qui permet de coder l'information et de l'envoyer, le second est un récepteur qui permet de capter et décoder l'information [6]. Pour une chaîne de transmission numérique, la donnée à transmettre passe par plusieurs blocs qui sont schématisés par la figure 1.2 :

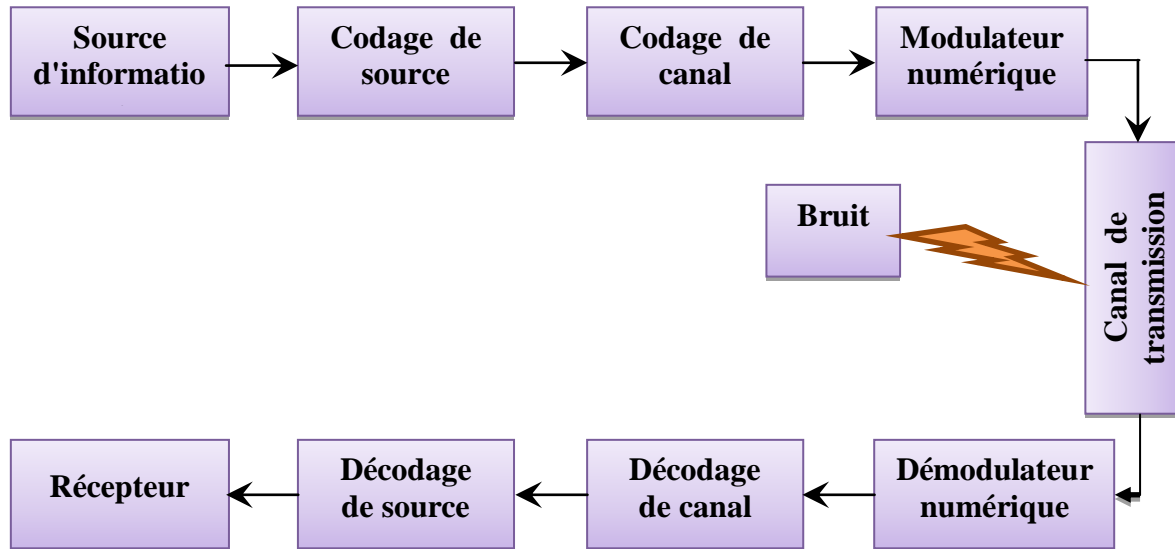


Figure 1. 2 : Principe d'une chaîne de communication numérique [6].

### 1.3.1 .Notion de message numérique

Est définie comme une suite d'éléments pouvant prendre un permis  $Q$  valeurs possible ; on appelle alphabet l'ensemble de ces valeurs qui peuvent être considéré comme des variables aléatoires discrètes, on dit que les symboles sont  $Q$ -aires (cas particulier et fréquent) où l'alphabet est constitué uniquement de deux valeur, notées traditionnellement 0 et 1, les éléments sont dit binaire [1].

## 1.4 .Le codage

Le processus de codage dans un système de communication numérique s'effectue en deux étapes : codage de source (compression de données) et codage de canal [3].

### 1.4.1. Le codage de source

La première opération qui est effectuée sur la source des données est le **codage de source**, ou compression des données est principalement utilisée pour réduire la redondance dans la séquence d'information issue de la source numérique [13].

### 1.4.2. Le codage de canal

Le codage de canal, appelé aussi codage correcteur d'erreurs est utilisé pour protéger l'information contre les erreurs introduites par le canal de communication bruité, c'est pour transmettre l'information avec le maximum de fiabilité. Le principe consiste à insérer dans le message des éléments binaires sont les bits de redondant pour permettre à la réception de



détecter et éventuellement corriger les erreurs de transmission. Cette opération conduite à une augmentation de débit binaire de transmission.

Il est classifié en deux catégories, la première est celle des codes en blocs où une séquence de  $k$  bits d'information est codée dans un bloc de  $n$  symboles  $n > k$ . La deuxième est celle des codes convolutionnels utilisés pour la transmission des symboles d'information qui arrivent en série sans structure en bloc. Dans notre travail, seul les codes convolutionnels sont étudiés, puisque d'une part, les principaux inconvénients du codage en bloc peuvent être évités par codage convolutif, et d'autre part ces derniers sont la base des Turbo-code [3].

## 1.5. La modulation

La modulation a pour objectif d'adapter le message au canal de transmission. Le principe est d'utiliser une onde porteuse qui sert à transporter les données par une modification d'un ou de plusieurs caractéristiques de cette onde [9].

$$s(t) = A \cos(\omega_0 t) + \varphi_0 \quad (1.2)$$

Les paramètres modifiables sont :

- L'amplitude :  $A$
- La fréquence :  $f_0 = \frac{\omega_0}{2\pi}$
- La phase :  $\varphi_0$

### 1.5.1. La modulation BPSK

La modulation BPSK (Binary Phase Shift Keying) est une modulation très utilisée dans l'étude théorique des communications numériques [4]. Elle est la plus performante au niveau de la probabilité d'erreur et donc la plus adéquate pour les systèmes de communications. Cette modulation consiste à faire varier la phase de la porteuse en fonction du signal [6].

C'est une modulation binaire (un seul bit est transmis par période  $T$ ), et sont basées sur les symboles d'information  $+1$  ou  $-1$  [9].

Si on a une porteuse  $A \cos(\omega_0 t)$ , le signal modulé est :

$$m(t) = A \cos(\omega_0 t + \phi_k) \quad (1.3)$$

Avec :  $k = 0, 1, \dots, M - 1$ .

Pour la modulation BPSK ( $M = 2$ )

$$\phi_k = 0 \text{ ou } \pi \quad (1.4)$$

Alors les deux états de la phase  $\phi_k$  sont 0 et  $\pi$  [4].

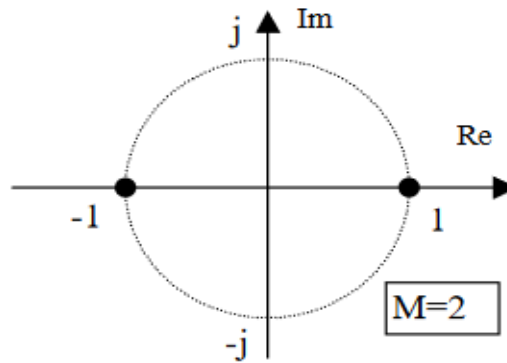


Figure 1. 3: Diagramme de constellation pour BPSK[6].

## 1.6. Le canal de transmission

Pour transiter de l'émetteur au récepteur, l'information utilise un média appelé, canal de transmission, il est pratiquement constitué par un câble bifilaire, un câble coaxiale une fibre optique ou bien l'espace libre [3].

Le canal est un véritable problème pour les transmissions de données. En effet, il est source de bruits de toutes sortes. Il existe des modèles pour identifier ce bruit et nous allons utiliser le modèle **AWGN** (Additive **White Gaussian Noise**) [1].

### 1.6.1. Le canal à Bruit Blanc Additif Gaussien (BBAG)

Le canal BBAG ou bien (AWGN) est un canal à entrée binaire et sortie analogique. Ce modèle implique que le bruit du canal est une variable aléatoire  $n$  qui s'additionne au signal modulé (figure 1.4). Dans ce cas, la variable  $n$  est gaussienne, de moyenne nulle et de variance  $\sigma^2$  tel que  $\sigma^2 = N_0/2$ , où  $N_0$  représente la densité spectrale de puissance de bruit. Mathématiquement nous pouvons écrire la densité de probabilité du bruit additif blanc gaussien [1].

$$f(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n^2}{2\sigma^2}} \quad (1.5)$$

Conséquemment, si  $x_k$  est la variable qui représente le signal modulé correspondant à un bit d'information à l'instant  $k$ , alors à la sortie du canal nous recueillons :

$$r_k = x_k + n \quad (1.6)$$

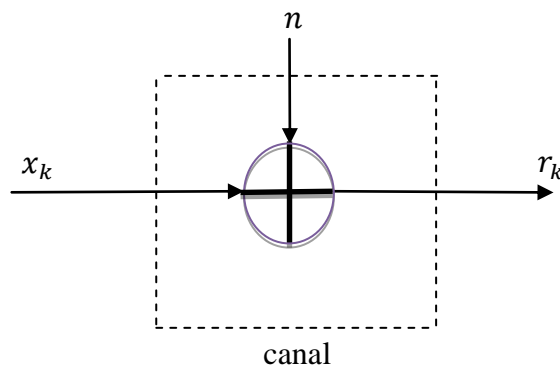


Figure 1. 4 : Modèle de Canal gaussien [1].

Le modèle de bruit blanc additif gaussien est très pratique, pour les calculs théoriques, Il reflète également très bien la réalité dans les communications spatiales [1].

### 1.7. Conclusion

Dans ce chapitre, nous avons tout d'abord introduit les systèmes de communication de façon générale. 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 et 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 2**

## **Turbo-codes**

## Chapitre 2

### Turbo-code

#### 2.1. Introduction

Le but de la recherche dans le domaine de la théorie de l'information est de trouver un système de contrôle d'erreur qui s'approchera le plus de la limite de Shannon, la solution été inventer en 1993, c'est les codes turbos.

Dans ce chapitre, on va exposer une étude détaillée sur les turbo-codes convolutif et leur décodage. Il traite tout d'abord le codage convolutif qu'est la base de ces codes. Puis les différents types de turbo-code et en particulier turbo-code parallèle.

#### 2.2. Les codes convolutifs

C'est en 1955 que Peter Elias introduit la notion de code convolutif, qui sont des codes correcteurs d'erreur les plus utilisés dans le domaine des télécommunications sans fil où une probabilité d'erreur par bit faible est requise.

Les codes convolutifs forment une classe extrêmement souple et efficace de codes correcteurs d'erreurs. Théoriquement, ils ont les mêmes caractéristiques que les codes en blocs sauf pour la valeur de leur dimension et leur longueur. Ils s'appliquent sur des séquences infinies de symboles d'information et génèrent des séquences infinies de symboles codés [6].

Les codes convolutionnels sont basés sur l'introduction de la redondance à l'aide de registres à décalage [3].

##### 2.2.1. Principe de base

Un codeur convolutionnel constituée d'un registre à décalage de  $m + 1$  étages, de  $n$  additionneurs modulo 2 qui sont reliés à certaines des cellules de registre et d'un commutateur qui sélectionne séquentiellement les sorties des  $n$  additionneurs.

La valeur  $m + 1$  est appelée la longueur de contrainte du codeur .Le mot de code dépend ainsi non seulement du message d'information à son entrée  $k$ , 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 donc les code convolutifs introduisent par conséquent un effet de mémoire d'ordre  $m$  [3].

Le principe d'un codeur convolutionnel est illustré par le schéma de la figure 2.1.

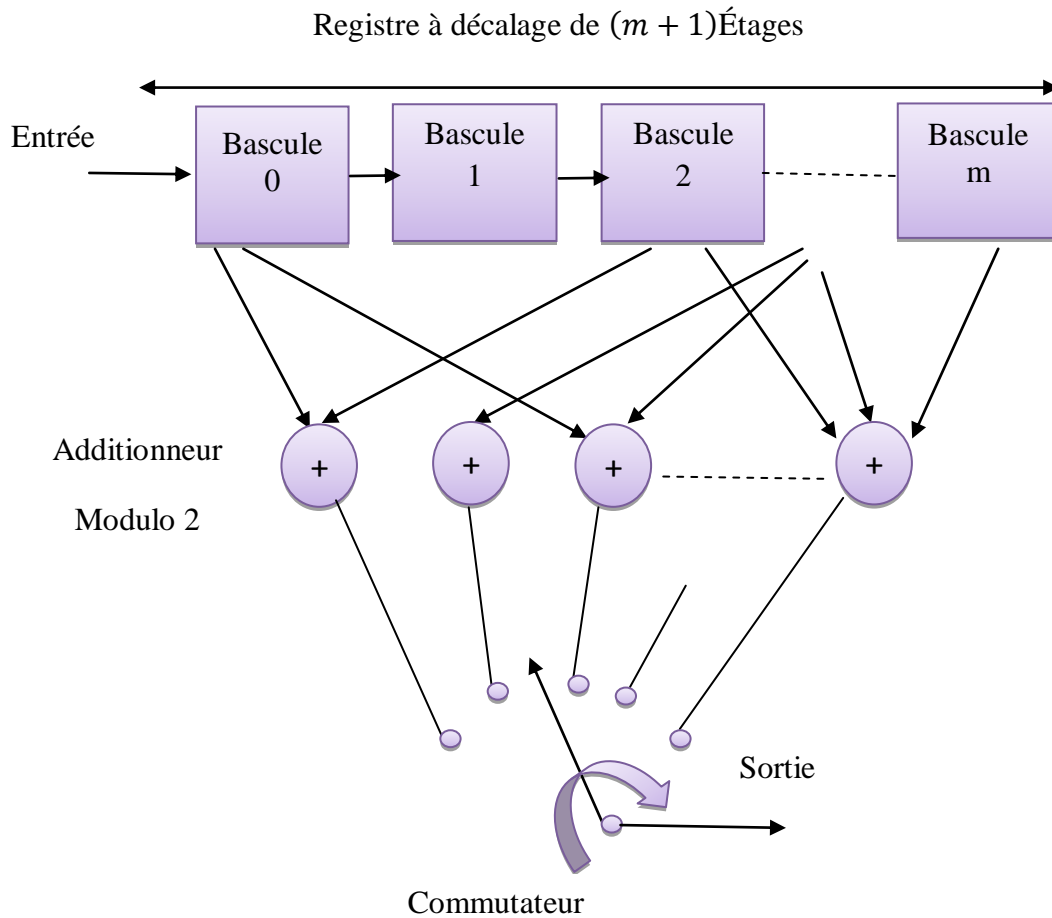


Figure 2. 1: Principe d'un codeur convolutionnel [1].

Il transmet les symboles d'information en série et 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 précèdent ce dernier (d'où l'expression " **convolutif** ").

Les trois principaux paramètres qui caractérisent un codeur convolutionnel sont :

- la longueur de contrainte  $m + 1$ .
- le taux de codage  $R = k/n$ .
- le polynôme générateur.

Chaque polynôme générateur est défini par le paramètre binaire  $g_{ij}$  .

Avec :

$$g_{ij} = \begin{cases} 1 & \text{s'il y a une connexion à l'additionneur} \\ 0 & \text{s'il n'y en a pas} \end{cases} \quad (2.1)$$

Pour illustrer mieux les codes convolutionnels, nous allons présenter un exemple d'un codeur de taux de codage  $R = 1/2$  et de longueur de contrainte  $m + 1 = 3$ .

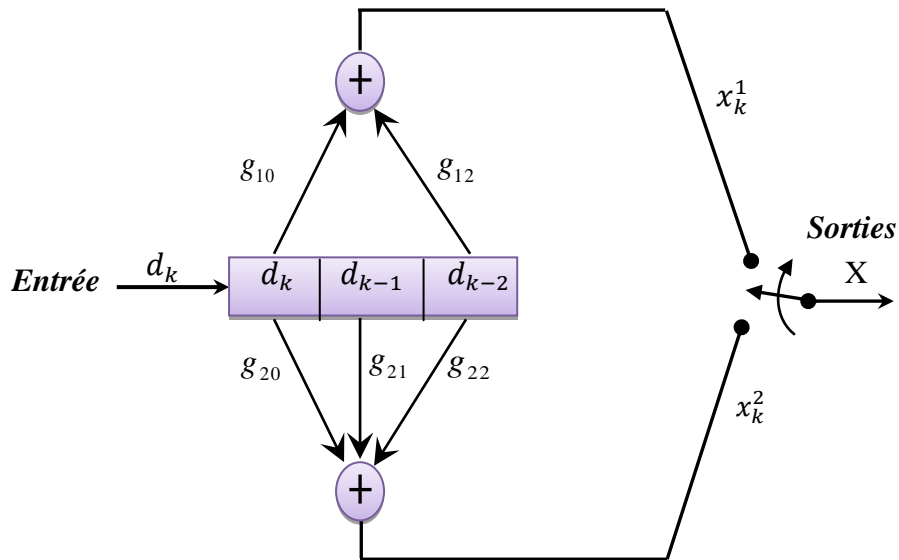


Figure 2. 2: codeur convolutionnel  $R = 1/2[4]$ .

Pour le codeur de la figure (2.2), les sorties  $x_k^i$ ,  $i=1, 2$ , étant égales à :

$$x_k^i = \sum_{j=0}^m g_{ij} d_{k-j} \quad (2.2)$$

Avec  $g_{ij} \in \{0,1\}$

Les deux séquences génératrices sont :

$$\begin{aligned} g_1 &= [g_{10}, g_{11}, g_{12}] \\ g_2 &= [g_{20}, g_{21}, g_{22}] \end{aligned} \quad (2.3)$$

En général on exprime les séquences génératrices en octal et ainsi, pour le codeur de la figure (2.2) nous avons  $g_1 = [1, 0, 1]$  où  $[101] = 5$  octal et  $g_2 = [1, 1, 1]$  où  $[111] = 7$  octal.

Les codes convolutifs sont aussi définis à partir de leurs polynômes générateurs exprimés en fonction de la variable  $D$  (Delay) équivalente à la variable  $Z^{-1}$  de la transformée en  $Z$  :

$$\begin{aligned} g_1 &\rightarrow G_1(D) = g_{10} + g_{11}(D) + g_{12}(D^2) \\ g_2 &\rightarrow G_2(D) = g_{20} + g_{21}(D) + g_{22}(D^2) \end{aligned} \quad (2.4)$$

Soit encore :

$$\begin{aligned} G_1(D) &= 1 + D^2 \\ G_2(D) &= 1 + D + D^2 \end{aligned} \quad (2.5)$$

➤ Exemple d'application

Considérons le codeur convolutif de la figure (2.2), et supposons que la séquence d'information soit :  $d_k = 01110011$

Voici les étapes de codage de la séquence  $d_k$ :

Entrée ( $d_k$ )	Etat ( $d_{k-1}, d_{k-2}$ )	Contenu du registre	Sortie( $x_k^1, x_k^2$ )
0	00	000	00
1	00	100	11
1	10	110	10
1	11	111	00
0	11	011	10
0	01	001	11
1	00	100	11
1	10	110	10

Tableau 2. 1: codage d'une séquence d'information donnée.

Le mot de code trouvé s'écrit :

$$X = 00\ 11\ 10\ 00\ 10\ 11\ 11\ 10$$

### 2.2.2 .Type de codeur convolutif

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



- **Les Codes Non Systématique (CNS)**

Un code convolutif est dit systématique si les bits source apparaissent directement dans les bits de sortie. Au contraire, dans un code convolutif non-systématique ou NSC (Non Systematic Convolutional code), tout bit de sortie est la combinaison de plusieurs bits d'entrées [8].

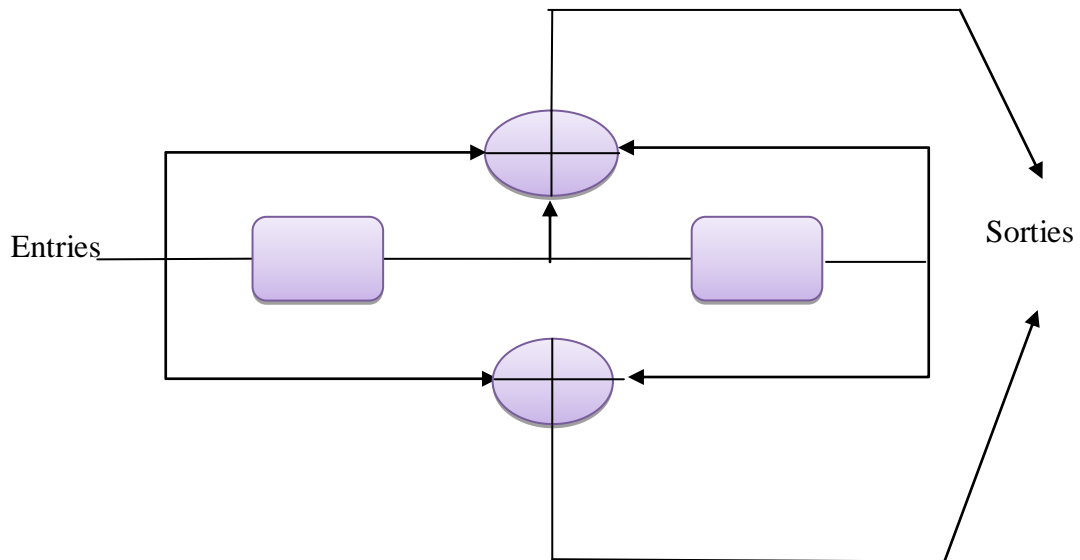


Figure 2. 3:Codeur non systématique CNS [6].

- **Les Codes Systématiques Récurifs (CSR)**

Un code convolutif est dit récurif si certains bits de sortie du codeur sont réinjectés en entrée. Un code convolutif récurif systématique ou RSC peut être construit à partir d'un code systématique en réinjectant dans l'entrée l'une des sorties (figure 2.3). Les codes convolutifs systématiques récurifs sont à la base de la construction des turbo-codes [8].

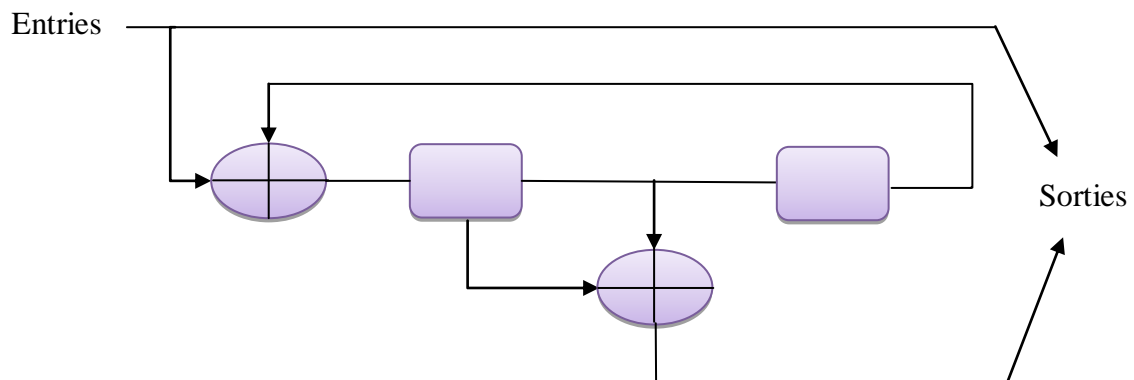


Figure 2. 4 : Codeur convolutionnel récurif systématique [6].

### 2.2.3 Représentation graphique des codes convolutifs

La représentation des codes convolutifs par les polynômes n'est pas réellement adaptée au décodage. Aussi pour réaliser les décodeurs, on utilise des représentations graphiques de ces codes sous forme d'un arbre, d'un treillis et diagramme d'état. Nous allons utiliser le codeur convolutif non systématique de la figure (2.2) [1].

- **Diagramme en arbre**

Pour la représentation du diagramme en arbre (figure 2.5) nous avons adapté les conventions suivantes :

- Le temps s'écoule de la gauche vers la droite
- Lorsque l'élément binaire à l'entrée du codeur est égal à 0 (respectivement à 1), le couple binaire en sortie du codeur (noté entre parenthèse) est porté par une branche montant (respectivement descendante) du diagramme en arbre. Les branches montantes et descendantes se séparent en un point appelé nœud. Plus généralement, en considérant des blocs de  $k$  éléments binaires d'information à l'entrée du codeur chaque nœud donne naissance à  $2^k$  branches.

Pour une séquence binaire donnée à l'entrée du codeur, la séquence correspondance en sortie du codeur est représentée par un chemin dans l'arbre, constitué par une suite de branches. Chaque bloc de  $n = 2$  éléments binaires en sortie du codeur dépend du bloc de  $k = 1$  élément binaire présent à son entrée mais aussi des  $m = 2$  blocs de  $k$  éléments binaires contenus dans sa mémoire. Ces  $mk = 2$  éléments binaires définissent l'état du codeur. Nous noterons les quatre états possibles de ce codeur de la façon suivante :

00; 01; 10; 11.

Quel que soit l'état initial du codeur, après  $(m + 1) = 3$  décalage à l'entrée du codeur, tous les états peuvent être atteints dans l'arbre [1].

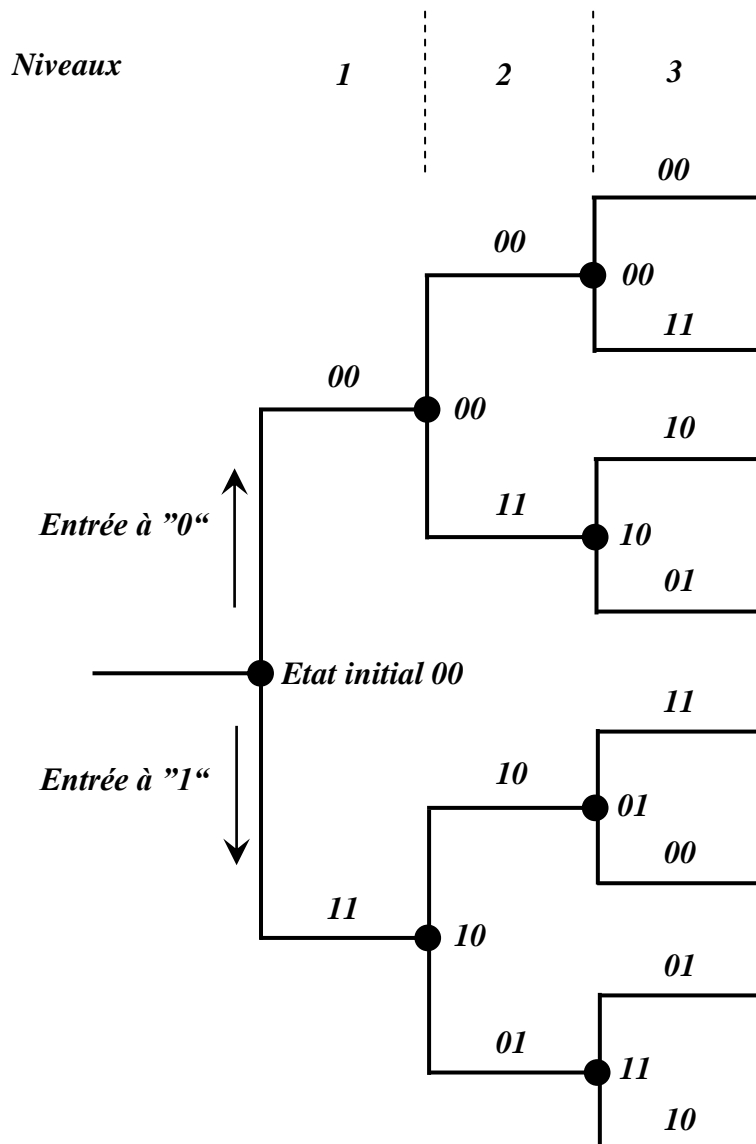


Figure 2. 5 : Représentation en arbre de l'encodeur [1].

- **Diagramme en treillis**

La sortie du codeur dépendant uniquement de son entrée et de son état, il est donc possible d'utiliser une représentation plus concise que l'arbre, appelé diagramme en treillis il est la représentation la plus courante d'un code convolutif. Dans ce diagramme sont pris en compte les états différents du codeur et la façon dont ils communiquent en fonction du temps. Sur la figure (2.6), nous avons représenté le treillis associé au codeur convolutif de la figure (2.2) en faisant l'hypothèse que l'état initial du codeur était 00.

Les branches en traits pointillés correspondent à la présence d'un élément binaire d'information égal à 0 à l'entrée du codeur et les branches en trait plein, à un élément binaire

égal à 1. À chaque branche nous avons associé la valeur du couple binaire disponible en sortie du codeur.

Après  $(m + 1)$  décalage, quel que soit l'état initial du codeur, le motif du treillis se répète. De chaque nœud partent  $2^k$  branches (ici 2), et en chaque nœud convergent  $2^k$  branches [3].

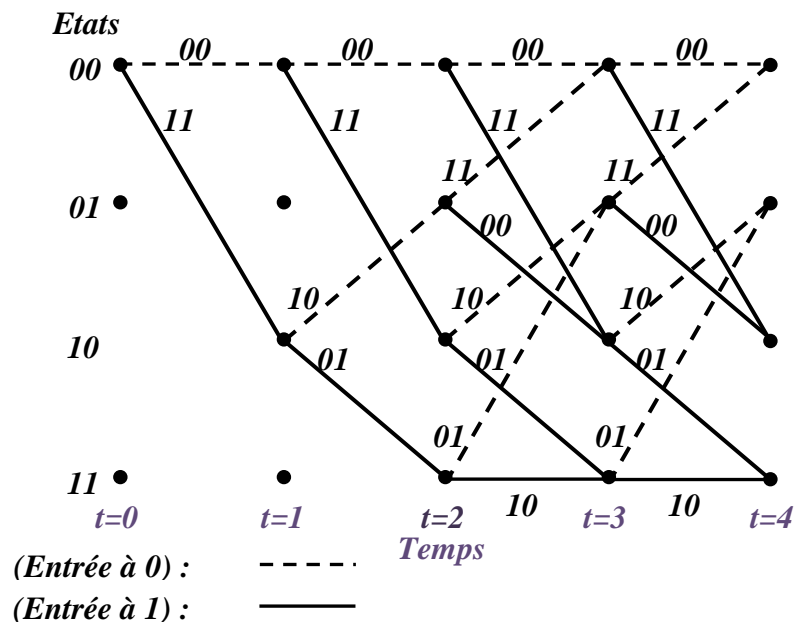


Figure 2. 6 : Diagramme en treillis [3].

### 2.2.3.3. Diagramme d'état

Le diagramme d'état est une autre représentation de la fonction d'un codeur convolutif, ne faisant pas apparait explicitement le temps. Ce diagramme ne retient que les différents états du codeur et la façon dont ils communiquent [3].

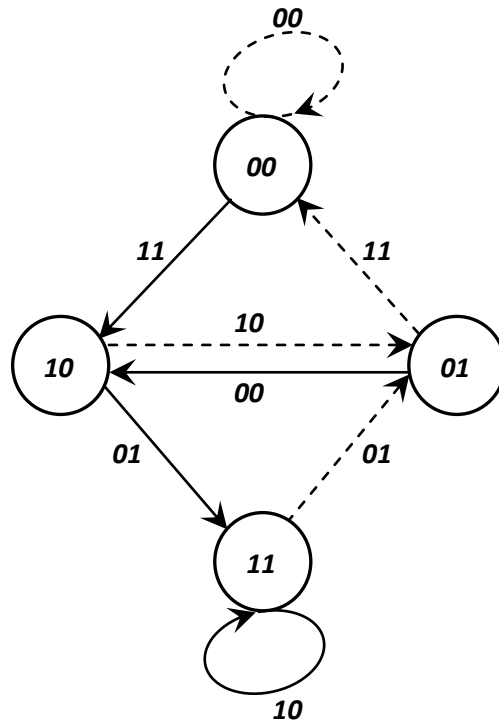


Figure 2. 7. Diagramme d'état de codeur de la figure 2.3 [3].

Avec :

- Les mots inscrits sur les transitions représentent les sorties. Tandis que ceux qui sont encerclés représentent les états.
- Les transitions en traits pleins correspondent à l'entrée de "1" dans le codeur. Celles en pointillés correspondent à l'entrée de "0" [4].

### 2.2.3. Décodage des codes convolutif

Le décodage d'un code convolutif va consister à rechercher, dans l'arbre ou dans treillis, la séquence binaire (correspondant à un chemin particulier) la plus proche de la séquence reçue. Cette séquence sera appelée séquence la plus vraisemblable. Le nombre de séquence possible étant généralement très important ( $N$  grand).

Avec :

$N$  : Nombre de mots d'une séquence

L'application de cette règle de décodage est donc d'une complexité excessive. Des algorithmes permettant de contourner cette difficulté ont été développés, parmi ces algorithmes [3] :

- **L’algorithme séquentiel**

Utiliser l’arbre pour rechercher la séquence la plus vraisemblable. Cet algorithme très performant peut revenir en arrière dans l’exploration de l’arbre si cela s’avéré nécessaire. Cette possibilité de reconsidérer certaine étapes du décodage, conduit malheureusement à un volume de calculs variable pour le décodage d’une séquence. Cet algorithme nécessite donc l’utilisation d’une mémoire tampon pour enregistrer les données reçues à l’entrée du décodeur, afin d’éviter des phénomènes de saturation [3].

- **L’algorithme Viterbi**

L’algorithme de Viterbe est la méthode la plus couramment utilisée pour le décodage à maximum de vraisemblance des codes convolutifs de faible longueur de contrainte (typiquement  $\leq 8$ ). Il permet d’apporter une réduction notable de la complexité de calcul. Il est basé sur l’idée que, parmi l’ensemble des chemins du treillis qui convergent en un nœud à un instant donné, seul le chemin le plus probable peut être retenu pour les étapes suivantes de recherche [7].

### 2.3. Association de plusieurs codes : concaténation

La concaténation consiste à associer deux codes ou plus afin d’obtenir de bonnes performances avec une complexité raisonnable de manière à ce que les éléments binaires du première codeur, dit extérieur, soient appliqués à l’entrée d’un second codeur, dit intérieur [3].



Figure 2. 8 : Le principe de la concaténation de deux codes [3].

### 2.4. L’entrelacement

L’entrelacement ou opération de génération de permutations consiste à changer selon l’application l’ordre des symboles du mot d’information ou du code, de manière déterministe afin de pouvoir les remettre dans l’ordre lors du décodage. La réorganisation des bits permet d’étaler les erreurs qui surviennent par paquets lors d’une transmission. L’utilisation par exemple d’un entrelacement avec un code convolutif diminue sa susceptibilité aux paquets d’erreurs. Selon la norme définie, la taille du bloc d’entrelacement doit être égale au nombre total des bits codés [6].

## 2.5. Turbo-code

L'invention du turbo-code n'est pas l'aboutissement d'un développement mathématique. Elle est le résultat d'une démarche intuitive et expérimentale dont l'origine est à trouver dans les travaux de quelques chercheurs européens.

Les turbo-codes ont été inventés en 1991, et présentés à la communauté scientifique en 1993, par une équipe de l'Ecole Nationale Supérieure des Télécommunications (l'ENST) de Bretagne dirigée par Claude Berrou et Alain Glavieux [5].

Les turbo-codes sont des codes correcteurs d'erreurs qui permettent d'avoisiner la limite théorique de correction sont une concaténation de deux ou plusieurs codes convolutifs de faible complexité. Leur décodage fait appel à un processus itératif «ou turbo», séparés par des blocs d'entrelacements [7].

### 2.5.1. Turbo-code série

Un turbo-code convolutif série (proposé par Forney, durant son travail de thèse en 1965) est obtenu par la concaténation série de codes convolutifs séparés par des entrelacements (figure 2.9). Cette concaténation la plus classique consiste à relier un codeur de Reed Solomon avec un codeur convolutionnel, séparés par un entrelaceur où un codeur au moins doit utiliser la sortie codée de l'autre. Elle a permis une décroissance exponentielle de la probabilité d'erreur au déprimant d'une complexité accru de décodage [6][7].

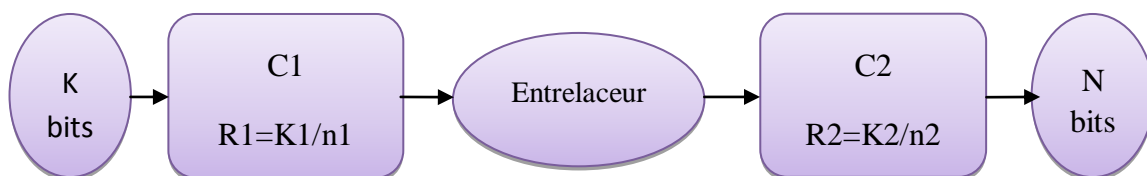


Figure 2. 9 : Schéma de principe turbo-code série [6].

### 2.5.2. Turbo-code parallèle

- **L'encodeur Turbo**

L'encodeur turbo est constitué de deux ou plusieurs codeurs convolutionnels systématiques récurrents (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. Nous concéderons dans la suite que la

concaténation parallèle de deux codeurs CSR identiques dont le schéma est donné à la figure 2.10.

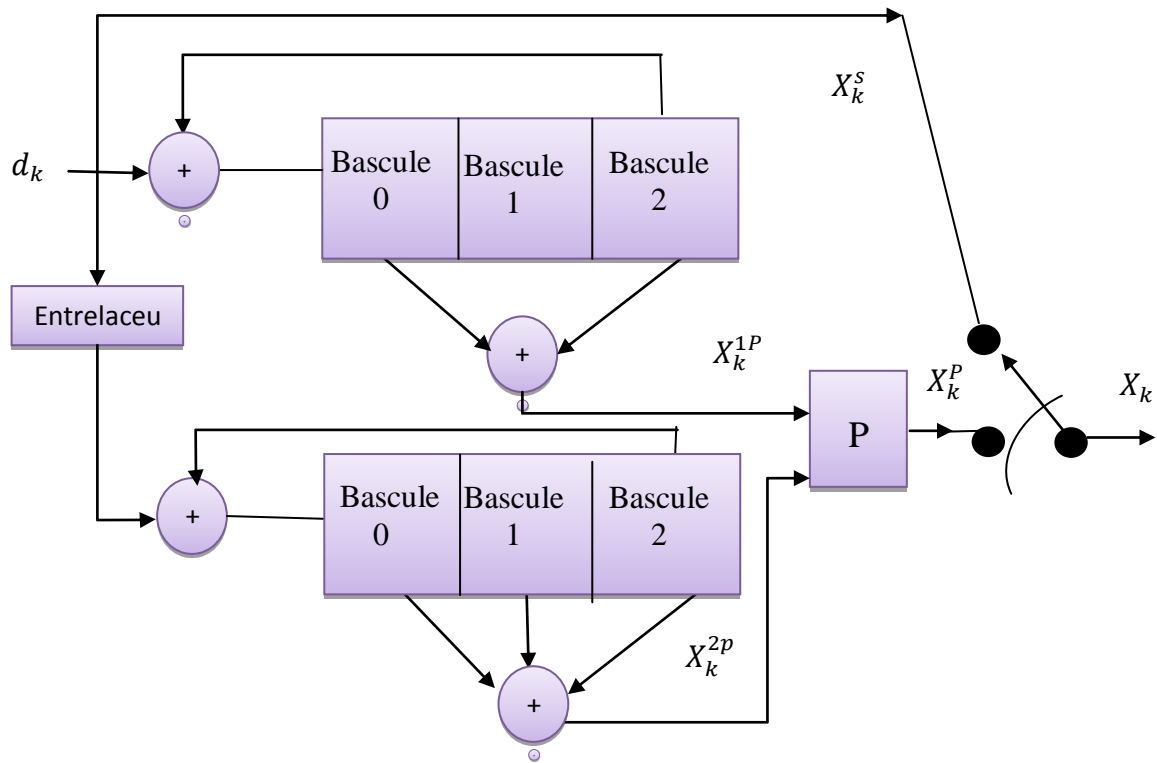


Figure 2. 10 : Schéma de principe d'un encodeur Turbo [1]

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 le codage convolutionnel de cette séquence. Elle produit alors une séquence de symboles de parité  $X_k^{1P}$ . La séquence d'entrée passe en parallèle par le codeur inférieur après avoir été entrelacée. Ce deuxième codeur produit une séquence de symboles de parité  $X_k^{2P}$ . 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'information  $X_k^S$ .

Le but de la perforation est de supprimer certains symboles de parité afin de faire varier le taux de codage [1]. Le taux de codage, noté  $R$ , de l'encodeur turbo obtenu à la suite d'une concaténation parallèle d'un premier codeur de taux de codage  $R_1$  et d'un deuxième codeur de taux de codage  $R_2$  vérifie la relation :

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} - 1 = \frac{R_1 R_2}{R_1 + R_2 - R_1 R_2} \quad (2.6)$$



Dans le cas de la figure 2.10,  $R_1 = R_2 = 1/2$ . La relation 2.6 nous donne un taux global sans perforation de  $R = 1/3$ .

- **Décodage d'un turbo-code**

Le décodage d'un turbo-code binaire utilise la technique de décodage itératif présenté sur le schéma de principe de la figure (2.11)

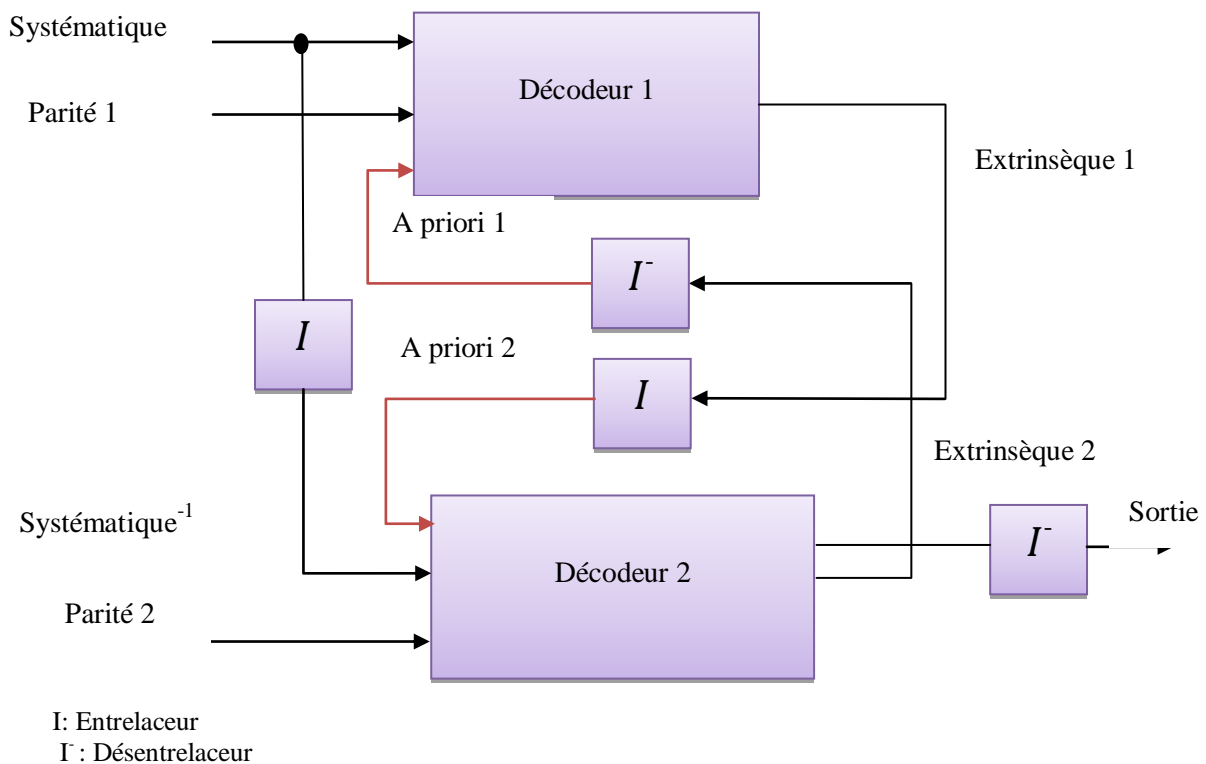


Figure 2. 11 : Principe de décodage itératif [1].

La boucle permet à chaque décodeur de tirer profit de l'ensemble des informations disponibles. Les grandeurs considérées à chaque nœud du montage sont des LRV (Logarithmes de Rapport de Vraisemblance), les opérations de décodage étant effectuées dans le domaine logarithmique [7].

En 1966, Chang et Hancock ont développé un algorithme minimisant la probabilité d'erreur par symbole. Cet algorithme est appelé MAP (Maximum a Posteriori), qui utilise de la règle du maximum a posteriori c.à.d. de prendre le maximum des probabilités a posteriori (APP) du symbole ou bit envoyé dans le canal. Le calcul de l'APP se base sur l'observation de la séquence reçue, d'où la nomination "a posteriori" [4].

- **Logarithmes de Rapport de Vraisemblance**

Ce rapport LRV est donné par la relation suivante :

$$L(d_k) = \log \frac{p(d_k = 1)}{p(d_k = 0)} \quad (2.7)$$

Avec

$$d_k = \begin{cases} 1, & L(d_k) \geq 0 \\ 0, & L(d_k) < 0 \end{cases} \quad (2.8)$$

Il faut noter que si le LRV est positif, la valeur de  $d_k = 1$  est plus probable que celle de  $d_k = 0$ , et inversement. Cette propriété sera exploitée plus tard dans le décodeur [4].

Le LRV conditionnel de  $d_k$  en fonction de la séquence reçue  $y$  peut être défini comme :

$$L(d_k/y) = \log \frac{P(d_k = 1/y)}{P(d_k = 0/y)} \quad (2.9)$$

Les probabilités conditionnelles  $P(d_k = 1/y)$  et  $P(d_k = 0/y)$  sont connues comme étant des probabilités à posteriori du bit décodé  $d_k$  [6].

- **Principe de décodage itératif**

La technique de décodage illustrée par la figure (2.11) se fait par itérations de deux étapes. Contrairement au codage, les blocs décodeurs sont en série et non en parallèle. Chaque décodeur agit l'un après l'autre. A partir des séquences reçues le 1<sup>er</sup> décodeur décode le bloc reçu, il transmet une information extrinsèque liée au logarithme du rapport de vraisemblance. Cette information est une évaluation de la fiabilité du décodage du 1<sup>er</sup> décodeur. Le 2<sup>ème</sup> décodeur disposant de cette information de fiabilité, ainsi que les symboles de parité issu du deuxième codeur. Si certaines erreurs ne sont pas corrigées par le 1<sup>er</sup> décodeur, elles peuvent être corrigées par le 2<sup>ème</sup>. Ces deux étapes se répètent par une boucle de contre réaction [1].

Le décodage itératif fonctionne au niveau des bits d'information et non au niveau des mots de code. L'objectif est l'estimation de l'APP de chaque bit d'information [6].

- **Structure d'un turbo décodeur**

Le décodage d'un turbo-code binaire s'appuie sur le schéma de principe de la figure 2.12. Le LRV en sortie d'un décodeur du code systématique peut être vu comme la somme de deux termes :

- l'information intrinsèque (en provenance du canal de transmission),
- l'information extrinsèque.

L'information extrinsèque obtenue à la sortie d'un décodeur devient une information à prior pour l'autre décodeur après avoir été entrelacée (figure 2.11).

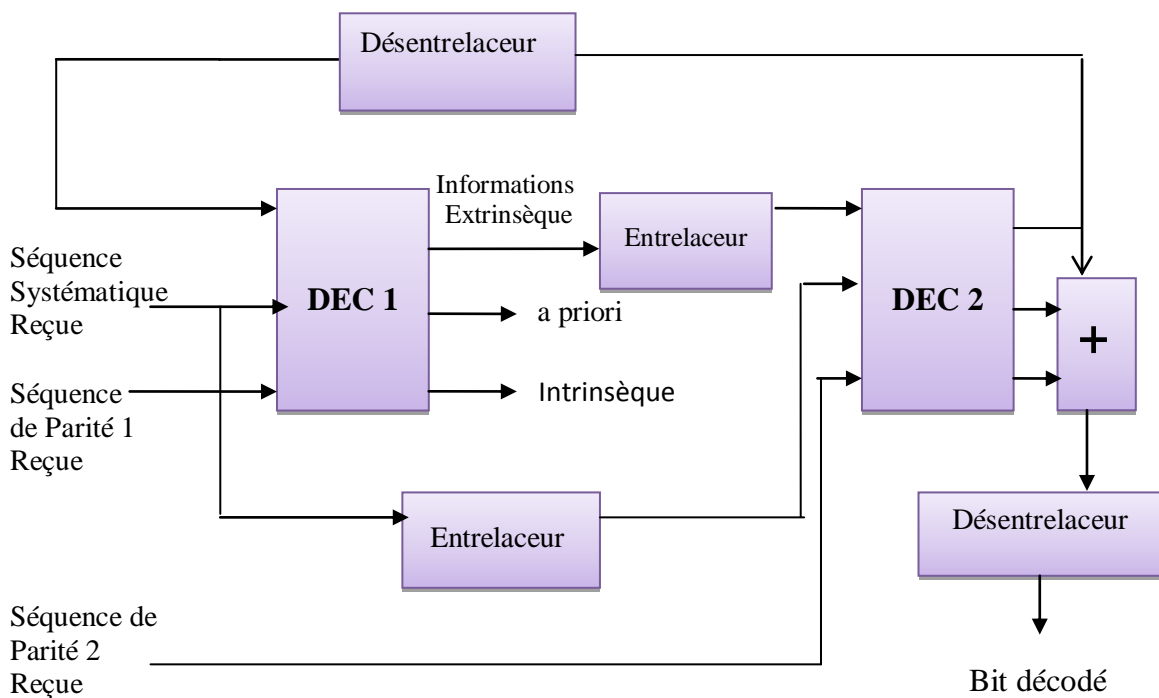


Figure 2. 12 : Schéma de principe d'un décodeur turbo.

- A la  $i^{\text{ème}}$  itérations : Pour générer les informations extrinsèques qui seront entrelacées avant d'être transmises au DEC2 comme entrée a priori. Le DEC1 reçoit l'information systématique, les informations de parité 1 et a priori.
- A la première itération : Nous n'avons pas besoin d'ajouter l'information a priori.

Le DEC2 reçoit l'information extrinsèque entrelacée, de l'information systématique entrelacée, et de la parité du  $2^{\text{ème}}$  codeur pour générer la  $2^{\text{ième}}$  information extrinsèque qui sera délacée à son tour, puis retournée, via la boucle de retour vers le DEC1 [1].

Après un nombre fixe d'itérations, la décision finale sur les bits du message est basée sur les valeurs extrinsèques des deux décodeurs et les valeurs du canal. Il est important de noter que seules les valeurs extrinsèques sont mises à jour à chaque itération, les valeurs du canal correspondant aux bits du message restent toujours fixes [4].

L'objectif de l'entrelacement est de disperser les erreurs qui arrivent en rafale difficilement détectable par les décodeurs MAP [1].

## **2.6. Applications des turbo-codes**

En raison de leurs performances, les turbo-codes ont été intégrés dans de nombreux standards de communication et en particulier pour les communications spatiales. L'ESA (European Space Agency) a été la première à les utiliser pour la sonde lunaire Smart-1 (lancement le 27 septembre 2003, orbite autour de la Lune le 15 novembre 2004 et alunissage en septembre 2006). La NASA les utilise depuis 2003 pour toutes ses sondes spatiales. Les turbo-codes sont utilisés de manière intensive pour les communications par flot en général : ADSL-2, la téléphonie mobile (UMTS, 3G) ou encore le nouveau standard DVB-S2 de télévision haute définition (HDTV) normalisé en mars 2005 [8].

## **2.6. Conclusion**

Dans la première partie de ce chapitre nous avons rappelé le principe de fonctionnement des codes convolutionnels. Les deux types de codages convolutionnel non récursif et systématique récursif ont été aussi présentés. Par la suite nous nous sommes intéressés à la concaténation en parallèle de deux codes CSR ou plus communément le turbo-code et leur structure de l'encodeur et le principe du décodage itératif. À la fin de ce chapitre, nous avons décrit les principales applications des turbo-codes dans le domaine des télécommunications. Une évaluation des performances par simulation d'un code turbo selon différents paramètres fera l'objet du prochain chapitre.

# **Chapitre 3**

## **Simulation et résultats**

# Chapitre 3

## Simulation et résultats

### 3.1. Introduction

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 taux d'erreur binaire «TEB».

L'objectif essentiel de ce chapitre est d'évaluer par simulation les performances des Turbo-codes parallèle dans une chaîne de transmission numérique sur une plage de valeurs  $E_b/N_0$ , à l'aide de composants de la Communications System Toolbox™ de Matlab. Nous avons choisi les spécifications LTE (Long Term Evolution) pour les paramètres des composants constitutifs. Cette simulation nous permet d'analyser et d'évaluer l'influence ou l'effet de quelques paramètres tels que : le nombre d'itérations, et la longueur du bloc de code (la longueur du bloc d'entrelacement).

### 3.2. Méthodologie de simulation

Les étapes de la simulation suivent la procédure suivante :

1. Générer aléatoirement les bits d'information.
2. Encoder les bits d'information par le codage turbo.
3. Utilisez la modulation BPSK pour convertir la donnée à la sortie de code turbo, en signaux complexes.
4. Introduire un bruit pour simuler les erreurs sur le canal de transmission AWGN.
5. A la réception, les opérations inverses sont effectuées pour démoduler et décoder la séquence des symboles reçues.
6. Comptez le nombre des bits erronés en comparant la séquence des bits décodés avec la séquence des bits transmis.
7. Calculer le taux d'erreurs binaire (TEB).
8. Tracer les performances (TEB en fonction de  $E_b/N_0$ ).

La simulation est initialisée avec une valeur de  $E_b/N_0 = -0.5 \text{ dB}$ . La valeur de  $E_b/N_0$  est augmentée après chaque simulation par  $0.5 \text{ dB}$ .

- La simulation se fait en exécutant le modèle Matlab/Simulink pour chaque valeur de  $E_b/N_0$  dB et on calcule la valeur de taux d'erreur binaire. Ensuite les valeurs de taux d'erreur binaire correspondant aux différentes simulations sont stockées dans l'espace de travail de MATLAB, puis les graphes représentant TEB en fonction  $E_b/N_0$  sont tracés, comme il est expliqué dans l'organigramme suivant :

- L'organigramme

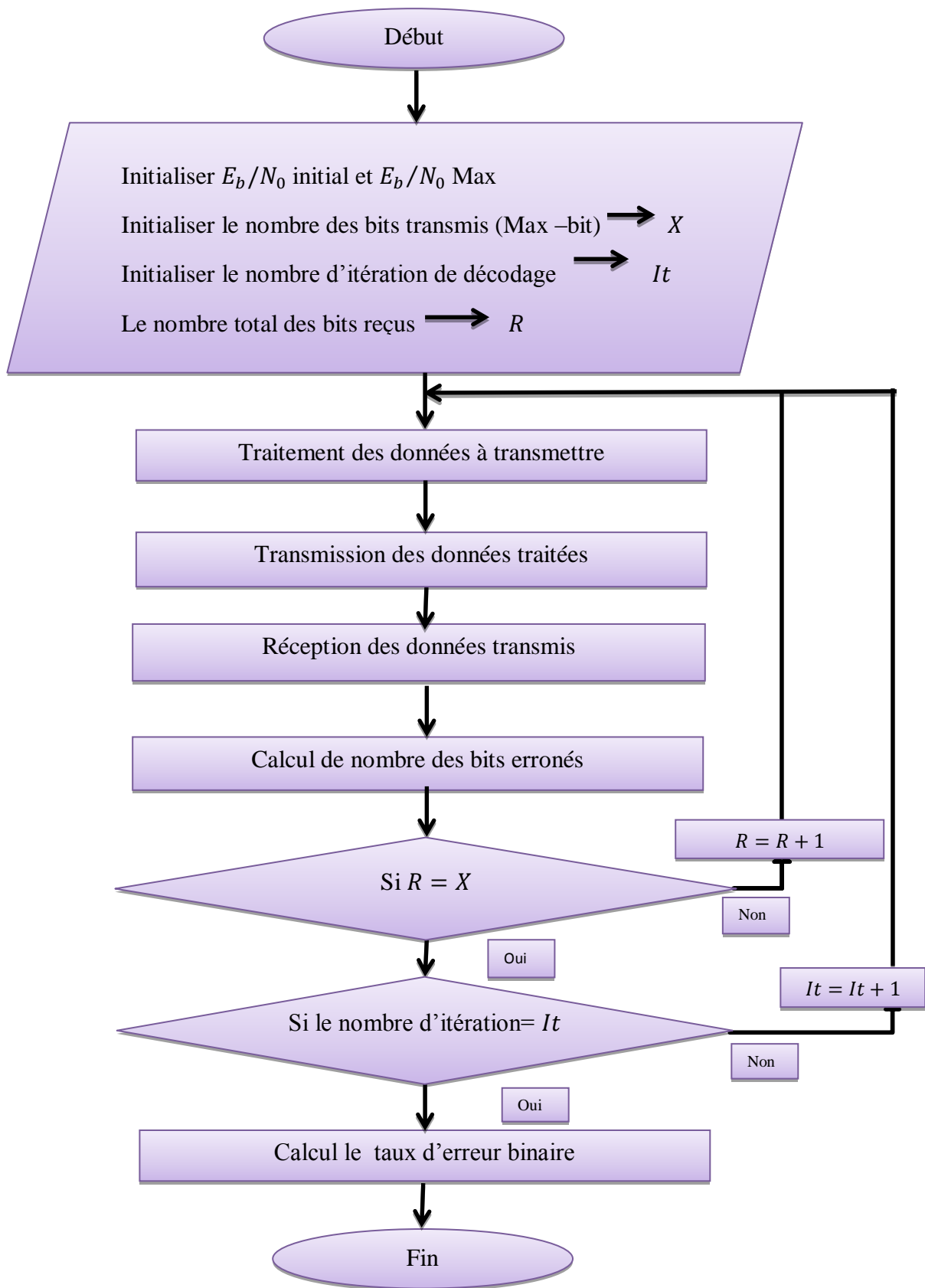


Figure3. 1: La procédure de simulation



### **3.3. Simulations et résultats**

Le code turbo est le seul type de code correcteur d'erreur qui se rapproche assez près de la limite de Shannon. Cependant, son comportement dépend de beaucoup de paramètres. Dans notre travail, nous allons présenter l'effet de différentes longueurs de blocs de code, l'influence de nombre d'itération sur la performance des codes turbo. Tous les résultats de simulations présentés sont basés sur le décodage itératif en utilisant le décodeur de probabilité a posteriori (APP). Notre travail consiste à simuler la structure de base des codes turbo utilisent deux structures de chaînes de transmission différents.

#### **3.3.1. Simulink à blocs de code de taille fixe**

La première partie illustrée sur la figure suivante, utilise l'implémentation Simulink à blocs de code de taille fixe pour décrire les détails du schéma de codage.

## Une chaine de transmission avec un turbo-code parallèle à blocs de code de taille fixe

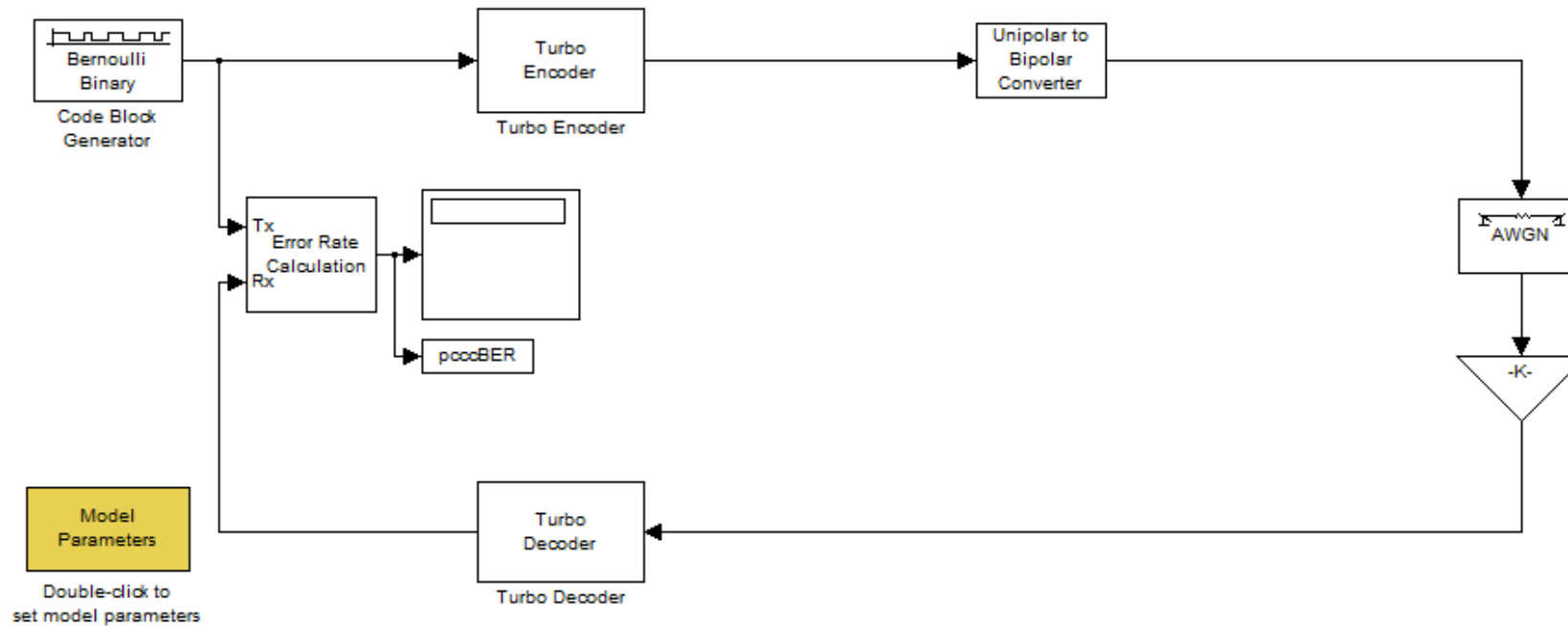


Figure3. 2: Modèle d'une chaine de transmission avec un turbo-code parallèle à blocs de code de taille fixe.

Cette chaîne de transmission adoptée par notre simulation 1 est constituée des blocs suivants :

- **Un bloc de source Bernoulli Binary Generator** : qui génère d'une façon aléatoire une suite binaire.
- **Un bloc Turbo Encoder** : qui réalise le codage de l'information, basé sur un bloc système utilise deux codeurs convolutifs systématiques récursifs identiques à huit états, ces deux encodeurs sont séparés par un entrelaceur.
- **Un bloc unipolar to bipolar converter** : qui permet d'effectuer l'opération de modulation
- **Un bloc turbo-décoder** : pour décodage itératif. Il utilise le décodeur de probabilité a posteriori (APP) comme composant de décodeur constituant.
- **Un bloc Error Rate Calculation** : qui sert à compter les bits ou les symboles erronés .Il produit les trois valeurs suivantes :
  - le taux d'erreur binaire du signal reçu.
  - le nombre total d'erreurs.
  - le rapport entre le nombre total d'erreurs et le nombre total de bits testés.

Dans cette étape de simulation on a présenté la probabilité d'erreur par bit appelée « Bit Error rate » (BER) en fonction du rapport  $E_b/N_0$ .

Pour illustrer parfaitement le rôle dominant du nombre d'itérations et la longueur du bloc de code dans la détermination des performances des Turbo-Codes, on va simuler la chaîne de transmission pour différents nombre d'itérations et différents longueurs du bloc qui peut être changé à partir de bloc Paramètres de modèle (model parameters) suivant :

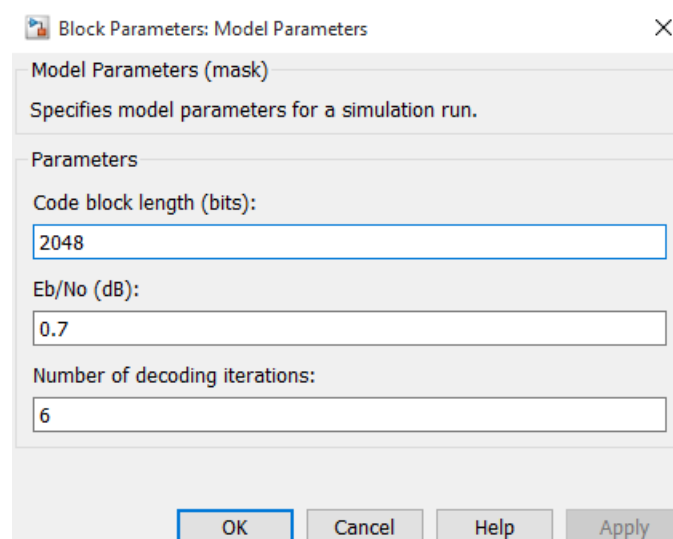


Figure3. 3 : le bloc de paramétrage.

Les résultats de simulation sont donnés par la figure (3.4) pour les tailles de blocs 2048 bits et 512 bits, et les nombre d'itérations 3 et 6.

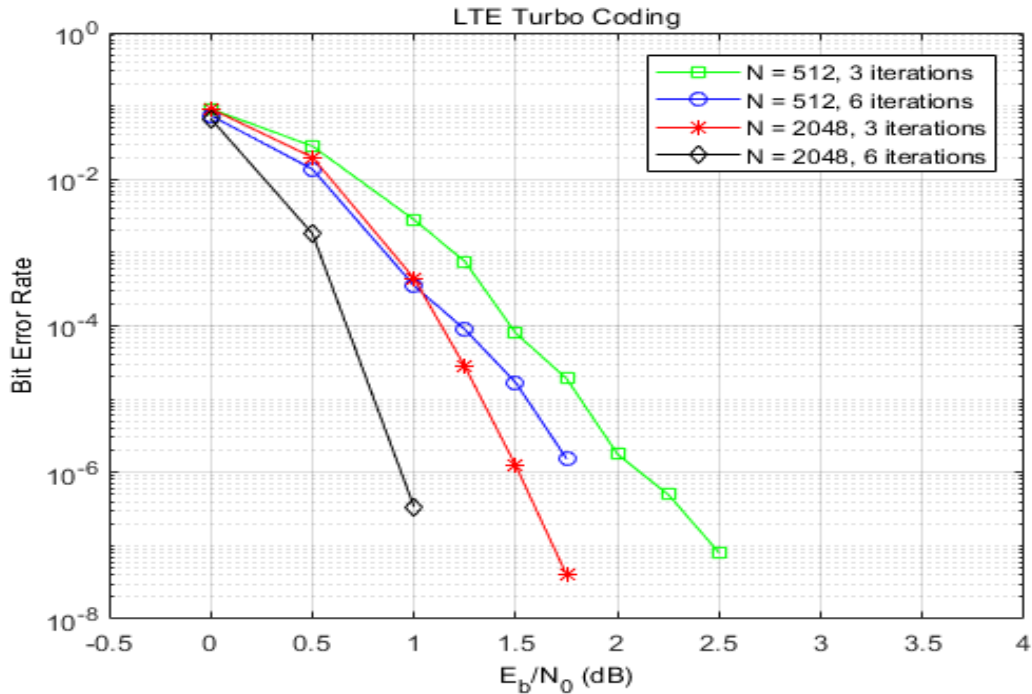


Figure3. 4: BER en fonction de  $E_b/N_0$ .

D'après cette figure (ci-dessus), il est clairement montré que l'augmentation du nombre d'itérations est suivie d'une amélioration considérable du taux d'erreurs binaire.

Selon la même figure, on constate une amélioration considérable du taux d'erreurs binaires avec l'augmentation de la longueur du bloc de code. Donc plus on augmente la taille du bloc de code, plus la qualité de transmission s'améliore.

Donc pour obtenir un meilleur résultat on augmente le nombre d'itération et la taille de bloc de code on même temps avec une longueur du bloc de code égale à 2048 bits et 6 itération. La figure (3.4), illustre ce scénario.

### 3.3.2. Simulink à blocs de code de taille variable

La deuxième partie utilise l'implémentation Simulink à blocs de code de taille variable. Le modèle (figure 3.5) est configuré pour exécuter deux longueurs de bloc de code, qui varient selon le signal de commande sélectionné. Les indices d'entrelaceur par longueur de bloc et la variance de bruit sont calculés par de 0,5 du rapport  $E_b/N_0$ (dB). À l'aide du détecteur de syndrome CRC, le modèle affiche le taux d'erreur du bloc de code.

## Une chaîne de transmission avec un turbo-code parallèle à blocs de code de taille variable

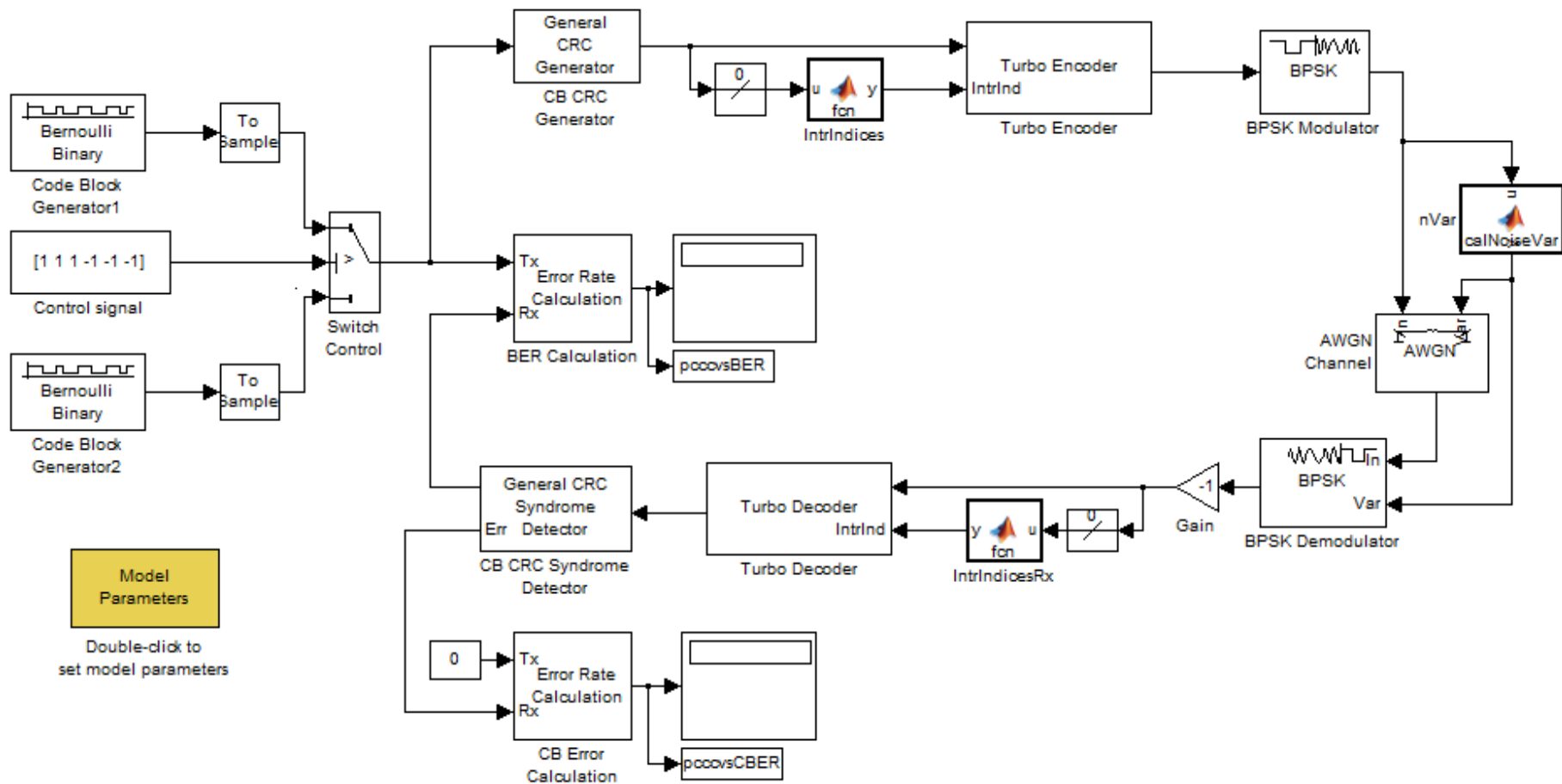


Figure3.5: Modèle d'une chaîne de transmission avec un turbo-code parallèle à blocs de code de taille variable.

La simulation de la chaîne de transmission pour différents nombre d'itérations (3 et 6 itération) et différentes longueurs du bloc (512 et 2048 bits). Nous a permet de trouver ces résultats de simulation (figure 3.6).

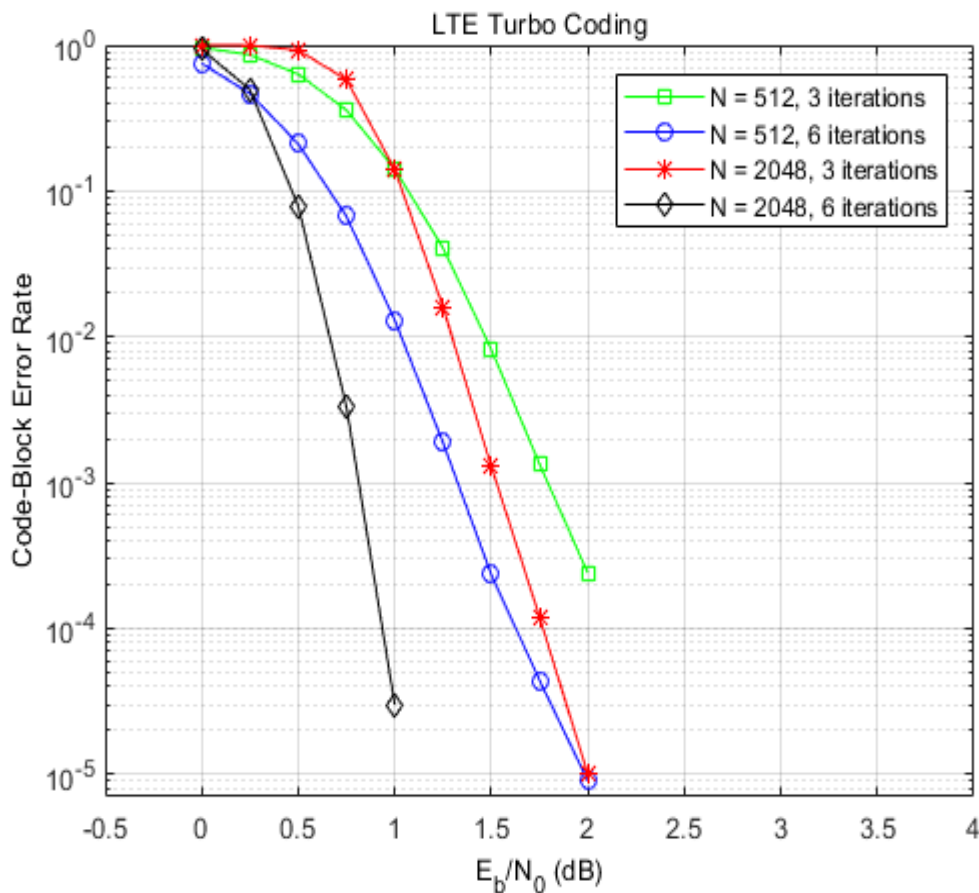


Figure 3.6 : CBER en fonction de  $E_b/N_0$ .

Cette figure montre qu'il y a une amélioration considérable du taux d'erreurs du bloc de code, si on augmente la longueur de l'entrelacement et le nombre d'itération. Donc plus on augmente la longueur du bloc et le nombre d'itération la qualité de transmission s'améliore malgré l'utilisation de blocs de code de tailles différentes. Sachant que la variance de la source d'information n'affecte pas les performances du turbo code.

### 3.4. Conclusion

Dans ce chapitre, nous avons évalué les performances d'une transmission numérique en termes la correction d'erreurs, nous avons utilisé le turbo code parallèle. D'après les résultats exposés dans ce chapitre, on constate que la qualité du message reçu augmente avec le nombre d'itérations. De même, l'augmentation de la taille du bloc de code engendre une amélioration surprenante au niveau des performances d'erreurs.

# **Conclusion générale**

## Conclusion général

Dans ce mémoire, nous nous sommes intéressés à l'étude des codes turbo et en particulier au turbo code convolutif parallèle dans un canal Gaussien. Les codes turbo font partie des codes correcteurs d'erreur les plus performants connus à ce jour, et qui permettent d'approcher la limite de Shannon.

L'objectif essentiel de notre travail est d'évaluer par simulation les performances des Turbo-codes parallèle dans une chaîne de transmission numérique. Cette simulation nous permet d'analyser et d'évaluer l'influence ou l'effet de quelques paramètres tels que : le nombre d'itérations, et la longueur du bloc de code.

Nous avons effectué des simulations de la structure de base des codes turbo à blocs de code de taille fixe et de blocs de code de taille variable en utilisant l'outil informatique Matlab\ Simulink .

La simulation a montré la performance du taux d'erreur sur les bits pour code de taille fixe, et le taux d'erreur du bloc de code pour des blocs de code de taille variable du schéma de codage concaténé parallèle dans un canal AWGN sur une plage de valeurs  $E_b/N_0$ .

Nous avons essayé d'améliorer les performances de notre turbo-code en jouant sur ses paramètres tels que : le nombre d'itérations, la taille du bloc de code pour code de taille fixe dans le but d'améliorer la qualité de transmission en fonction de BER.

Les résultats de simulation obtenus montrent qu'avec l'augmentation des deux paramètres cités ci-dessus, la capacité de correction du turbo code s'améliore. Donc les turbo-codes ont prouvé leur robustesse et leur capacité de correction.

Ensuite, on a testé les performances de notre turbo code avec des blocs de code de taille variable pour les mêmes paramètres de turbo code en fonction de CBER on obtenu les mêmes améliorations des performances avec l'augmentation du nombre d'itérations, la taille du bloc de code mais la complexité de chaîne aussi augmente.

En fin de compte, pour améliorer la qualité de transmission en assurant une bonne protection du signal émis, nous avons appliqué le turbo codage basé sur le décodage itératif. Il est



possible d'obtenir des résultats encore mieux en modifiant autre paramètres du turbo code comme la longueur de constraint, le rendement..., etc.

Un système basé sur le turbo codage offre des perspectives plus intéressantes.

# **Bibliographie**

## Bibliographie

- [1] IfteneEssedik « Etude des structures d'entrelaceur pour le codage turbo du canal pour l'optimisation des systèmes de communication par satellite », mémoire de magister, université Mohamed Boudiaf d'Oran, 2016.
- [2] AlexandruSpataru, « Fondement de la théorie de la transmission de l'information », Presses Polytechnique romandes, 1987.
- [3] Michel Joindot, Alain Glavieux, « introduction aux communications numériques », IUT Ecole d'ingénieurs, Edition DUNOD, 2007,335 pages.
- [4] Djamel Slimani « Décodage des turbo codes applique aux données multimédias pour une communication moderne » mémoire de magister de l'université Mohamed Boudiaf d'Oran, 2009.
- [5] Berrou, Glavieux et Thitirnajshima, "Near Shannon limit error correcting coding and decoding: turbo codes", IEEE transmission and communication, proceedings of patents, Geneva, Switzerland, May 1993.
- [6] SlamaAmina, HadjGuenaouiHafsa « performances d'un turbo code parallèle à rendement élevé » mémoire de master, université Saad dahlab Blida, 2011.
- [7] Claude Berrou, « code et turbo-codes », école nationale supérieure des télécommunications de Bretagne CS 8381829238 Brest Codex 3, France, 2007,408p.
- [8] Jean –Guillaume Dumas, Jean-Louis Roch, Eric Tannier, Sébastien Verrette « théorie des codes Compression, cryptage, correction », Edition Dunod, Paris, 2007,354p.
- [9] Patrice Kadionik, « bases de transmissions numériques les modulations numérique », EditionENSEIRB, 2000,41 pages.
- [10] A.Migan, S.argenterir « transmission de l'information : les codes convolutif », université Pierre et Marie Curie, Paris, 2012, 67p.
- [11] Belkacem Benmahammed, « Les turbo codes dans un contexte OFDM », thèse de master université Saad dahlab Blida, 2017.
- [12] Mechri Nawal, « l'effet du turbo code sur une transmission, MIMO-OFDM>, thèse de master université Saad dahlab Blida ,2013.
- [13] [www.unilim.fr/pages\\_perso/vahid/.../theorie\\_de\\_information](http://www.unilim.fr/pages_perso/vahid/.../theorie_de_information), pdf, p23.