MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DJILLALI LIABES SIDI BEL-ABBES

FACULTE DE GENIE ELECTRIQUE

Thèse

Présentée par
MOSTARI LATIFA

Pour obtenir le titre de Docteur en Science en Electronique
Option : Télécommunications

Turbo LDPC codes non-binaires associés à des modulations d'ordre élevé

Soutenue le 21/06/2016
devant le jury composé de :

TALEB Nasreddine	Pr., UDL, Sidi Bel-Abbès	Président
BOUNOUA Abdennacer	Pr., UDL, Sidi Bel-Abbès	Directeur de thèse
TALEB-AHMED Abdelmalik	Pr., UVHC, Valenciennes	Co-Directeur de thèse
MAHDJOUB Zoubir	Pr., UDL, Sidi Bel-Abbès	Examinateur
MERIAH Sidi Mohamed	Pr., UABB, Tlemcen	Examinateur
BENTOUTOU Youcef	Dir. Rech., Oran	Examinateur

Année universitaire 2015/2016

Je dédie ce modeste travail à:

mes parents,

mes frères Abdellatif, Djamel rahimaoullah et Aissa,

et mes sœurs Zahira et Nassima

Remerciements

Tout d'abord, je voudrais remercier ALLAH tout puissant qui m'a donnée le courage et la patience pour faire ce modeste travail.

Je tiens à remercier Mr. Abdelmalik TALEB-AHMED, professeur à l'université de Valenciennes, pour m'avoir accueilli dans son laboratoire LAMIH, pour ses conseils, ses lectures enrichissantes et son suivi continu.

Je remercie également Mr. Abdennacer BOUNOUA, Professeur à l'université de Sidi Bel-Abbès qui a accepté d'être le rapporteur de cette thèse et pour la confiance qu'il m'a accordée.

Mes remerciements vont également aux membres de jury : Pr. Nasreddine TALEB, Pr. Zoubir MAHDJOUB, Pr. Sidi Mohamed MERIAH et Dir. Rech. Youcef BENTOUTOU pour m'avoir honoré par leur évaluation de ce travail.

Mes remerciements aussi à tous ceux qui m'ont aidée et encouragée.

Résumé

Les codes **LDPC** (Low-Density Parity-Check) binaires ont une capacité s'approchant de la limite de Shannon pour des blocs de donnée de grande taille. Cependant, ces codes ont l'inconvénient d'être moins efficaces pour des blocs de donnée de taille faible ou moyenne. De plus, l'association de codes **LDPC** binaires avec des modulations d'ordre élevé le rendre aussi moins efficaces. Les codes **LDPC** non-binaires, définis sur des corps de Galois d'ordre $q > 2$, permettent de résoudre ces problèmes.

Le décodeur **LDPC** doit fonctionner en décisions douces calculées à l'aide du **LLR** (Log-Likelihood Ratio) ou de l'**APP** (A Posteriori Probability) selon le type d'algorithme de décodage utilisé. Le calcul exact de ces décisions pour des modulations d'ordre élevé implique des opérations compliquées. Plusieurs algorithmes ont été introduits afin de simplifier le calcul du **LLR** pour les codes binaires. Dans ce travail, nous utilisons ces algorithmes pour simplifier le calcul du **LLR** pour les codes **LDPC** non-binaires. Ainsi, une méthode pour simplifier le calcul de l'**APP** est introduite. Elle est programmée afin d'adapter le plus parfaitement possible le système de transmission au type de canal considéré. Cette méthode conduit à simplifier la mise en œuvre du système.

Bien que les codes **LDPC** non-binaires soient de bons codes correcteurs d'erreurs pour un système utilisant une modulation d'ordre élevé, la concaténation des codes **LDPC** avec un décodage itératif est encore attrayante pour construire des codes correcteurs d'erreurs puissants. Dans ce travail, nous étudions une concaténation de deux codes **LDPC** binaires, réguliers et identiques, disposés en parallèle à travers la structure turbo-code proposée par Berrou et autres. Ainsi, nous proposons, sous la même structure proposée par Berrou et autres, un nouveau code correcteur d'erreurs appelé turbo **LDPC** code non-binaire.

Mots-clés **:** ***LDPC*** *code non-binaire, turbo-code, entrelacement, décodage itératif, codage de Gray, constellation,* ***LLR****,* ***APP****,* ***MAQ****, canal Gaussien, canal de Rayleigh.*

Abstract

Binary **LDPC** codes (Low-Density Parity-Check) have a capability approaching the Shannon limit for large data blocks. However, these codes have the disadvantage of being less effective for small or medium data blocks. Moreover, the binary **LDPC** codes in combination with higher order modulations also make it less effective. The non-binary **LDPC** codes defined on the Galois field of order $q > 2$, can solve these problems.

The **LDPC** decoder must operate on soft decisions calculated using: **LLR** (Log-Likelihood Ratio) or **APP** (A Posteriori Probability) according to type of decoding algorithm used. The exact calculation of these decisions for higher order constellations involves complicated operations. Several algorithms have been introduced to simplify the calculation of the **LLR** for the binary codes. In this work, we use these algorithms to simplify the **LLR** calculation for non-binary LDPC codes. Thus, a method to simplify the **APP** calculation is introduced. It is programmed to adapt as perfectly as possible the transmission system to the type of channel in question. This method leads to simplify the implementation of the system.

Although the non-binary **LDPC** codes are good error-correcting codes for a system using a high order modulation, the concatenation of **LDPC** codes with iterative decoding is still attractive to construct powerful error correcting codes. In this work , we study a concatenation of two identical regular binary **LDPC** codes disposed in parallel through the turbo-code structure proposed by Berrou and all. Thus, we propose, in the same structure proposed by Berrou and all, a new error correcting code called non-binary turbo **LDPC** code.

Keywords*: non-binary **LDPC** code , turbo-code, interleaving, iterative decoding, Gray mapping, constellation, **LLR, APP**, **QAM**, Gaussian channel, Rayleigh channel.*

ملخص

رموز **LDPC** (مراجعة التكافؤ قليلة الكثافة) الثنائية لديها قدرة تقترب من حد شانون لكتل بيانات من الحجم الكبير. ومع ذلك، هذه الرموز لديها عيب كونها أقل فعالية لكتل بيانات من الحجم الصغير أو المتوسط .وعلاوة على ذلك، تركيب رموز **LDPC** الثنائية مع تحويرات ذات درجة عالية تجعلها أقل فعالية .رموز **LDPC** غير الثنائية يمكن أن تحل هذه المشاكل.

فك **LDPC** يجب أن تعمل بقرارات ناعمة التي تحسب باستخدام **LLR** أو **APP** تبعا لنوع الخوارزمية المستخدمة. الحساب الدقيق لهذه القرارات لأجل تحويرات ذات درجة عالية ينطوي على عمليات معقدة .وقد أدخلت عدة خوارزميات لتبسيط حساب **LLR** من أجل الرموز الثنائية .في هذا العمل، نستخدم هذه الخوارزميات لتبسيط حساب **LLR** من أجل رموز **LDPC** غير الثنائية .أيضا، تم إدخال طريقة لتبسيط حساب **APP**، التي تم برمجتها لتكييف قدر الممكن نظام الاتصالات مع نوع القناة. يؤدي هذا الأسلوب إلى تبسيط تركيب النظام.

على الرغم من أن رموز **LDPC** غير الثنائية هي رموز جيدة لتصحيح الأخطاء للنظام باستخدام تحويرات ذات درجة عالية، جمع رموز **LDPC** مع فك تكرارية لا تزال جذابة لبناء رموز قوية لتصحيح الأخطاء .في هذا العمل، ندرس جمع اثنين من رموز **LDPC** الثنائية ،منتظمة ومتماثلة، مرتبة بالتوازي من خلال هيكل رمز توربو مقترحة من طرف Berrou وآخرين. أيضا، نقترح، في نفس الهيكل المقترح من قبل Berrou وآخرين، قانون جديد لتصحيح الخطأ مسمى رمز توربو **LDPC** غير الثنائي.

الكلمات الدالة *: رمز* ***LDPC*** *غير الثنائي، رمز توربو، التداخل، فك متكرر، ترميز غراي، كوكبة،* ***LLR، APP، QAM،*** *قناة جوس، قناة رايلي.*

Table des Matières

Liste des Figures

Liste des Tableaux

Liste des Abréviations

LDPC : Low-Density Parity-Check

DVB-S2 : Digital Video Broadcasting-Satellite

WIMAX : Worldwide Interoperability for Microwave Access

DSL : Digital Subscriber Line

W-LAN : Wireless Local Area Network

PCGC : Parallel Concatenated Gallager Code

SP : Sum-Product

FFT-SP : Fast Fourier Transform Sum-Product

MS : Min-Sum

EMS : Extended Min-Sum

SMS : Simplified Min-Sum

MAQ : Modulation d'Amplitude de deux porteuses en Quadrature

LLR : Log-Likelihood Ratio

APP : A Probability a Posteriori

MAP : Maximum A Posteriori

TEB : Taux d'Erreurs Binaires

MDA : Modulation par Déplacement d'Amplitude

MDP : Modulation par Déplacement de Phase

MDF : Modulation par Déplacement de Fréquence

AWGN : Additive White Gaussian Noise

Introduction Générale

"Il fat toujousr visrr la lune,
car myme en csa d'échec,
on atetrrit dans les étoiles"

A la lecture de la phrase ci-dessus vous avez effectué une correction d'erreurs. La phrase sus-indiquer a été corrompue par des erreurs. Vous étiez en mesure d'effectuer une correction d'erreurs. Vous utilisez d'autres lettres pour trouver le mot attendu. C'est le principe de la redondance : la phrase contient plus de lettres que d'informations. C'est ce principe qu'utilise la théorie des codes correcteurs d'erreurs : il s'agit d'ajouter des redondances qui n'apportent pas d'information, mais qui permettront de détecter les erreurs de l'information transmise, voire de les corriger.

Les codes correcteurs d'erreurs constituent une des disciplines nées des travaux de Shannon sur les communications numériques. En 1948, Shannon [22] prouve qu'il existe une efficacité spectrale limite que l'on ne peut pas dépasser si l'on souhaite une transmission sans erreurs. Shannon était capable de donner une limite sans donner le code qui permet la correction des erreurs.

Afin de réaliser la solution de codage, des efforts intensifs de recherches ont été faits dans le monde entier. L'essentiel est de réaliser un code permettant de se rapprocher à la limite de Shannon, et aussi d'atteindre un bon compromis performance/complexité.

Jusqu'aux années 80, le code permettant d'atteindre la limite de Shannon avec une complexité raisonnable n'était pas encore introduit. Deux grandes familles de code correcteur d'erreurs ont été imposées sont : les codes en bloc qui se subdivisent en plusieurs types et les codes convolutifs [72].

Les performances d'un code binaire augmentent avec sa longueur de bloc N, tandis que la complexité de décodage d'un code binaire de dimension N est de l'ordre de $O(N)$ [73, 74]. Une approche de ce problème est de réduire la complexité de décodage. Une façon d'aborder

le problème, est de construire de bons codes qui présentent une complexité de décodage raisonnable, est d'utiliser des codes concaténés. La stratégie de la concaténation de codes est de construire des codes correcteurs d'erreurs puissants en combinant, en parallèle, en série ou hybride, deux ou plusieurs codes correcteurs d'erreurs, en blocs ou convolutifs, de longueurs faibles à modérées avec une complexité de décodage raisonnable.

En 1993 Berrou [18] a montré que les performances des codes concaténés peuvent être améliorer avec une technique itérative utilisée pour leur décodage. Ce nouveau schéma de code, appelé turbo-code, permet d'atteindre la limite de Shannon. Les turbo-codes peuvent être des turbo-codes en bloc ou des turbo-codes convolutifs [18, 75] selon le type des codes concaténés. Ainsi, selon le type de la concaténation, parallèle ou série, on peut avoir des turbo-codes parallèles ou séries.

Après la puissance du décodage itératif qui a été mise en évidence grâce à l'invention des turbo-codes. Les codes **LDPC** (Low-Density Parity-Check) binaires, qui ont été négligés, à cause de leur complexité, pendant de longues années depuis leur introduction par Gallager en 1962 [23, 76], ont été redécouverts par Mackay [77] en 1995 et Spieleman et autres [78] en 1996.

Les codes **LDPC** sont des codes en blocs, basés sur des matrices de contrôle de parité creuses, c'est-à-dire le nombre d'éléments non-nuls dans la matrice et plus faible que le nombre de zéros, et leur décodage se fait selon le principe de décodage itératif. Après quelques avancées théoriques fondamentales, les codes **LDPC** binaires ont été proposés dans des standards tels que **DVB-S**2, **WIMAX**, **DSL**, **W-LAN**....Une contribution significative a été introduite par Luby et d'autres en 1997 [79] qui ont introduit et paramétré les codes **LDPC** irréguliers. Ces derniers ont pour principale caractéristique de présenter de meilleures performances que les codes réguliers.

Ainsi, bien que les codes **LDPC** soient de bons codes correcteurs d'erreurs, la concaténation de ces codes avec un décodage itératif est encore attrayante pour construire des codes correcteurs d'erreurs puissants [20, 21, 80] avec une complexité raisonnable. Les codes **LDPC** concaténés en parallèle originaux **PCGC**s (Parallel Concatenated Gallager Codes), ont été introduits dans [67] comme une classe de codes concaténés dans lesquels deux codes **LDPC**, qui sont des codes irréguliers ayant des paramètres différents, interagissent en parallèle à travers la structure turbo-code sans entrelaceurs. L'entrelaceur exécute comme une

permutation, il change la répartition du poids du code. Il est donc utile dans l'augmentation de la distance minimale du code. Dans [68, 69] une concaténation série de codes **LDPC** binaires irréguliers, est également introduite.

Les auteurs dans [67] ont montré comment les différents codes **LDPC** composants avec différents paramètres affectent les performances globales dans un canal gaussien. Bien qu'ils limitent leur description des **PCGC** à un rendement égal à 1/3 en combinant deux codes **LDPC** de rendement égal à 1/2, ils ont prédit que les conclusions sont facilement étendues au cas où trois ou plusieurs codes sont utilisés, tel que présenté dans [68]. Egalement dans [67] les auteurs ont montré que, l'entrelaceur n'est pas nécessaire lorsque le code **LDPC** est concaténé avec un autre, afin d'étudier l'effet de l'entrelaceur entre les deux codes **LDPC** composants, un **PCGC** a été modifié pour utiliser un entrelaceur pour permuter les bits d'information comme dans le turbo-code tel que présenté dans [14] pour des codes irréguliers.

Bien que les codes **LDPC** irréguliers soient plus performants que les codes réguliers, les codes **LDPC** irréguliers ont un plancher d'erreurs et une complexité de codage plus élevés que les codes réguliers. Dans ce travail, nous étudions une concaténation de deux codes **LDPC** réguliers identiques disposés en parallèle à travers la structure turbo-code proposée par Berrou et autres [18], en utilisant un entrelaceur entre les deux codes **LDPC** qui le composent.

En 2002 Davey et Mackey [25] ont étudié les codes **LDPC** non-binaires. Ces codes sont définis sur un corps de Galois fini d'ordre q $GF(q)$. Les codes **LDPC** non-binaires offrent de meilleures performances que leurs équivalents binaires lorsque le bloc codé est de longueur faible à modérée, ou lorsque la modulation utilisée est à grand nombre d'états.

Cependant, les avantages de l'utilisation de codes **LDPC** non-binaires impliquent une augmentation importante de la complexité de décodage. Plus l'ordre de corps de Galois est élevé plus la complexité devient importante. Pour un code défini dans un corps de Galois $GF(q)$, la complexité est d'ordre $O(q^2)$. De même, la mémoire requise pour le stockage des messages est d'ordre $O(q)$. Plusieurs algorithmes ont été développés dans le but d'atteindre un algorithme avec une bonne performance et complexité réduite de sorte qu'ils deviennent implémentables.

Les auteurs dans [25] ont proposé le premier algorithme de décodage itératif pratique pour les codes **LDPC** non-binaires. Cet algorithme, appelé l'algorithme Somme-Produit (**SP**),

appelé aussi l'algorithme de propagation de croyance, est un algorithme de décodage itératif optimal mais avec une complexité de calcul élevée.

Plusieurs algorithmes ont été proposés pour réduire la complexité de l'algorithme **SP** non-binaire : l'algorithme **FFT-SP** [34] dans le domaine fréquentiel sur la base de la transformée de Fourier rapide (**FFT**); dans le domaine logarithmique [35] tel que l'algorithme Min-Sum (**MS**); dans le domaine mixte [36] où on peut profiter de ces deux domaines. Pour réduire les besoins de complexité et de la mémoire deux algorithmes simplifiés, basés sur l'algorithme **MS**, ont été proposés: l'algorithme **EMS** (Extended Min-Sum) [37] et l'algoritthme **MM** (Min-Max) [39]. De l'algorithme **EMS** l'algorithme **SMS** (Simplified Min-Sum) est dérivé.

Pour une performance de décodage optimisée, non seulement l'algorithme est important, mais également la structure du code et la construction de la matrice de contrôle de parité jouent un rôle important. Dans le but d'atteindre un code **LDPC** non-binaire avec une bonne performance et complexité réduite de sorte qu'ils deviennent implémentables, tout en conservant inchangé l'algorithme de décodage, un turbo **LDPC** code non-binaire est proposé. Nous appliquons les codes **LDPC** non-binaires au schéma d'un turbo-code proposé dans [18].

Étant donné que le nombre croissant d'applications nécessitent des transmissions à haut débit sans augmenter la bande passante du canal de transmission, c'est la raison pour l'utilisation d'un système combinant une modulation d'ordre élevé à un code correcteur d'erreurs performant. Il est intéressant d'examiner les performances des codes proposés, lorsqu'ils sont associés à des modulations d'ordre élevé (**MAQ**-16, **MAQ**-64, **MAQ**-256) utilisant le codage de Gray. Le codage de Gray est largement appliqué aux constellations d'ordre élevé afin de réduire les erreurs de bit car deux points de constellation adjacentes diffèrent dans un seul bit.

Le décodeur **LDPC** doit fonctionner en décisions douces calculées à l'aide du **LLR** (Log-Likelihood Ratio) ou de l'**APP** (A Probability a Posteriori). Le calcul exact de ces décisions a des problèmes d'opérations compliquées. Plusieurs algorithmes ont été introduits pour simplifier le calcul exact du **LLR** pour les codes binaires tels que l'algorithme pragmatique et l'algorithme max-log-**MAP** (max-log Maximum A Posteriori). Dans ce travail, nous appliquons ces simplifications pour les codes **LDPC** non-binaires. Ainsi, nous proposons une méthode pour simplifier le calcul exact de l'**APP**. Elle est programmée afin

d'adapter le plus parfaitement possible le système de transmission au type de canal considéré. Cette méthode conduit à simplifier la mise en œuvre du système.

L'objectif de cette thèse consiste à simplifier la combinaison des code **LDPC** non-binaires avec des constellations d'ordre élevé dans deux types de canaux de transmission: gaussien et de Rayleigh, en simplifiant le calcul des décisions pondérées à l'entrée du décodeur. Et d'étudier le nouveau code correcteur d'erreurs, turbo **LDPC** code, permettant d'améliorer les performances de ce système.

Ce travail s'est déroulé au sein du laboratoire de "Réseaux de Communication, Architecture et Multimédia (RCAM)," de l'université Djillali Liabès de Sidi Bel-Abbés. Cette thèse est composée comme suit :

Le premier chapitre discute les notions de base des codes en bloc linéaires. Ainsi, une description d'un système de transmission dans lequel le code correcteur d'erreurs est inséré.

Dans le deuxième chapitre, on donne une description détaillée des codes **LDPC** binaires et non-binaires. Ensuite, nous détaillons aussi les principaux algorithmes de décodage associés à cette classe de code correcteur d'erreurs.

Le chapitre III vise à montrer une nouvelle méthode permettant la simplification aussi bien du calcul de l'**APP** pour les codes **LDPC** non-binaires que la mise en œuvre du système combinant un code **LDPC** et une constellation d'ordre élevé, ainsi que la manière de la réaliser. Ensuite, une description de la procédure permettant d'adapter le calcul simplifié du **LLR**, introduit pour les codes binaires, aux codes **LDPC** non-binaires. Enfin, on présente les différents résultats de simulation de l'effet de ces simplifications sur les performances d'un code **LDPC** non-binaire associé à une **MAQ**-16, **MAQ**-64, **MAQ**-256, sur canal gaussien et sur un canal de Rayleigh.

Le chapitre IV introduit un nouveau code correcteur d'erreurs: turbo **LDPC** code. La première partie de ce chapitre décrit le principe d'un turbo **LDPC** code à concaténation parallèle et leur décodage. Ensuite, nous évaluons les performances de ce code proposé, associé à des constellations d'ordre élevé, sur un canal gaussien et sur un canal de Rayleigh.

On termine par une conclusion générale et les perspectives.

Chapitre I

Système de Transmission Numérique et Code en Bloc Linéaire

I.1 Introduction

Le mathématicien Shannon était capable de montrer théoriquement une transmission sans erreurs sans donner le code correcteur d'erreurs qui permet de réaliser la solution de correction des erreurs. Après cette théorie, des recherches ont été faites afin de réaliser le code correcteur d'erreurs. Dans le but d'atteindre la limite théorique de Shannon et d'avoir un bon compromis performance/complexité. Deux grandes familles de codes correcteurs d'erreurs ont été introduites : les codes en blocs et les codes convolutifs [1, 2].

Il existe deux catégories de codes en bloc : les codes en bloc linéaires et non-linéaires. Les codes en bloc non-linéaires ne sont jamais employés dans les applications pratiques et ne sont pas beaucoup étudiés. Les codes en bloc linéaires se subdivisent en plusieurs codes tels que les codes **LDPC**.

Dans ce travail, on s'intéresse aux codes **LDPC** qui satisfont à la preuve de Shannon. Pour ceci, on discute, dans ce chapitre, les codes en bloc linéaires. L'étude des codes correcteurs d'erreurs a nécessité de les situer à l'intérieur d'un système de transmission numérique. On a pour cela, dans la première partie et sous forme introductive, identifié les différentes fonctions qui sont présentées dans un tel système. La deuxième partie est axée sur l'étude des codes en bloc linéaires.

I.2 Description générale d'un système de transmission numérique

Un système de transmission numérique véhicule de l'information entre une source à un destinataire. La source d'information et le destinataire sont en général séparés par une distance considérable. Le canal de transmission, en même temps qu'il assure la connection entre ces deux entités, dégrade le signal transmis. Il faut alors mettre en place un système d'émission-réception pour minimiser l'effet du canal sur le signal.

Le schéma le plus simple d'un système de transmission numérique est donné à la figure I.1.

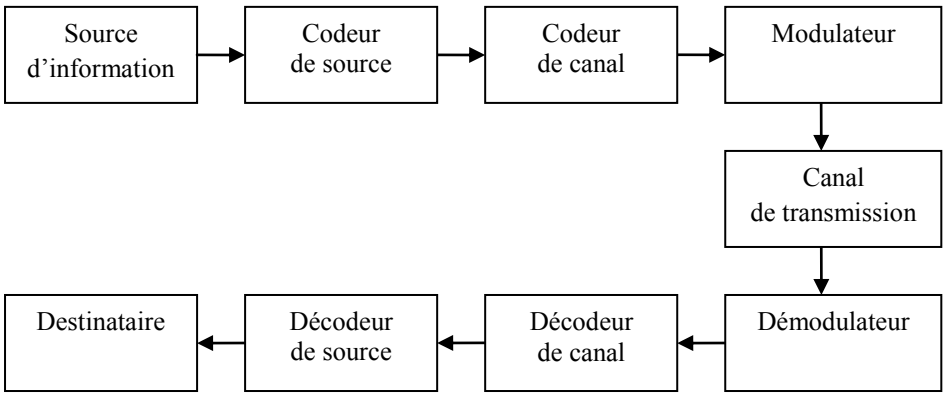


Figure I.1- Schéma d'un système de transmission numérique

I.2.1 Source d'information

La source d'information génère le message à transmettre. Le message transporté peut être soit directement d'origine numérique comme les fichiers d'ordinateur, soit d'origine analogique (parole, image...) mais converti sous une forme numérique.

I.2.2 Codeur de source

Plus le message est long plus l'usage d'un canal coûte cher. Pour diminuer ce coût, le codeur de source convertit la séquence de l'information de la source en une séquence alternative avec une représentation plus efficace d'informations, c'est-à-dire avec un peu de symboles. Par conséquent, cette opération s'appelle souvent la compression. Selon la source, la compression peut être sans perte (par exemple, pour des fichiers de données d'ordinateur) ou avec perte (par exemple, pour la vidéo, les images immobiles, ou la musique) où la perte peut être faite pour être imperceptible ou acceptable [3].

I.2.3 Codeur de canal

Pour transmettre le message avec la fiabilité maximale, le codeur de canal également appelé codeur correcteur d'erreurs fait en quelque sorte l'inverse du codeur de source puisqu'il consiste à ajouter des symboles de redondances au message qui ne porteront pas

d'information mais qui permettront de détecter et/ou de corriger les erreurs de transmission à la réception.

- **Efficacité d'un codeur de canal**

L'efficacité d'un codeur de canal se reflète dans le gain de codage. Le gain de codage est la différence entre le rapport signal à bruit (E_b/N_0 (dB) où E_b étant l'énergie reçue par bit d'information transmis et N_0 la densité spectrale de puissance du bruit contenue dans la bande) requis pour atteindre un certain Taux d'Erreurs Binaires (**TEB**) sur les bits transmis avec un système codé et le rapport signal à bruit requis pour atteindre le même **TEB** avec un système non codé, comme l'indique la figure I.2. Le **TEB** est défini par :

$$\boldsymbol{TEB} = \frac{nombre\ \ de\ bits\ erron\acute{e}s}{nombre\ \ de\ bits\ transmis} \tag{I.1}$$

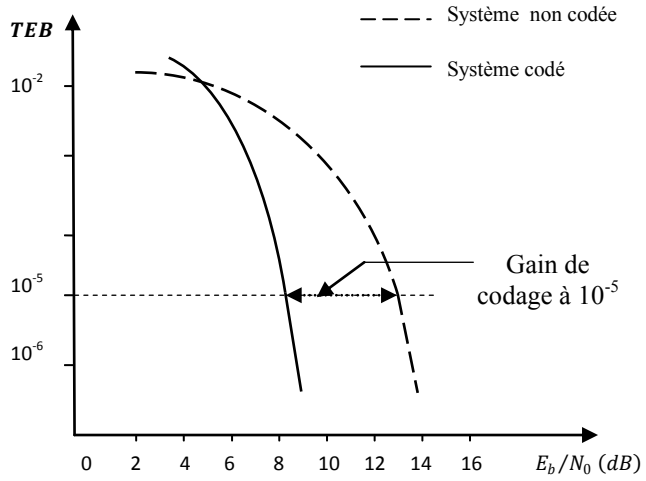


Figure I.2- Illustration du gain de codage pour un $\boldsymbol{TEB} = 10^{-5}$

I.2.4 Modulateur

Le modulateur convertit l'information numérique en formes d'ondes adaptées aux caractéristiques du canal. Il transpose la gamme de fréquence occupée par le signal dans une autre bande propre à la transmission.

Les formes d'ondes peuvent varier selon leur amplitude (Modulation par Déplacement d'Amplitude (**MDA)**), leur phase (Modulation par Déplacement de Phase (**MDP**)) ou la combinaison des deux (Modulation d'Amplitude de deux porteuses en Quadrature (**MAQ**)), et selon leur fréquence (Modulation par Déplacement de Fréquence (**MDF**)).

I-2-5 Canal de transmission

Le canal de transmission constitue le lien physique entre l'émetteur et le récepteur. Il emploie deux modes de transmission : filaire (câble, fibre optique,…) et non-filaire (espace libre). Le dispositif de stockage est également un genre de canal, lequel peut être envoyé à... par écrit et être reçu de...par lecture.

Les deux modèles de canaux utilisés dans notre étude sont présentés au paragraphe suivant.

I.2.5.1 Canal de Rayleigh

Le canal de Rayleigh constitue un cas particulier des canaux à trajets multiples (également appelés canaux à évanouissements).

Un canal à trajets multiples introduit des affaiblissements et des retards variables avec le temps [4]. Ce phénomène provient de la réflexion et de la diffraction du signal émis dans un environnement changeant. Dans les communications radio-mobiles [5] par exemple, l'atténuation du signal varie avec la vitesse du véhicule (qui est le récepteur) et la nature des obstacles.

Les chemins reflétés sont habituellement plus longs que le chemin direct, qui signifie que ces signaux atteignent le récepteur plus tard que ceux du chemin direct. Par conséquent, les signaux de différents chemins peuvent arriver au récepteur avec différents retards et à différentes heures [6, 7] comme représenté sur la figure I.3.

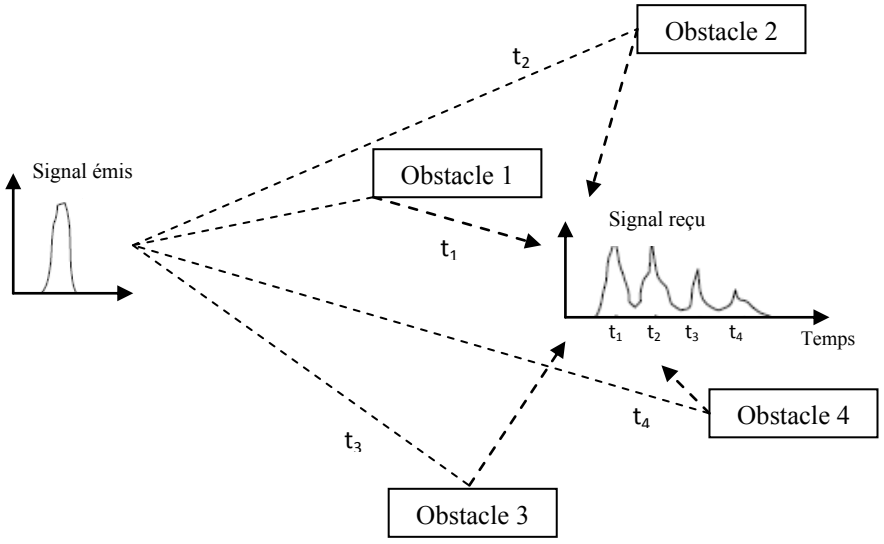


Figure I.3- Illustration du temps de retard des trajets multiples

La figure I.4 montre l'affaiblissement d'un signal en fonction de la distance de séparation entre l'émetteur et le récepteur, en utilisant deux fréquences porteuses $f_c = 1500\ MHz$ et $f_c = 500\ MHz$.

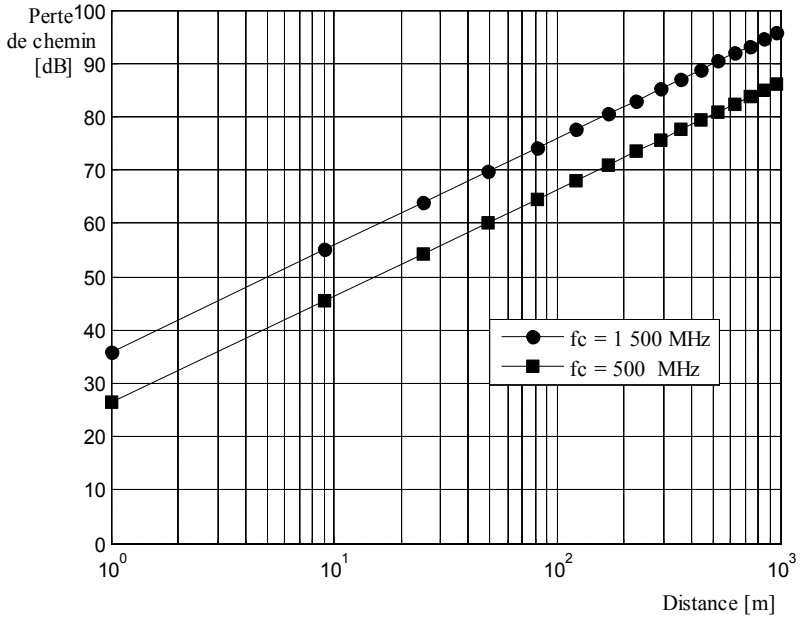


Figure I.4- Affaiblissement d'un signal, en fonction de la distance de séparation entre l'émetteur et le récepteur, pour deux fréquences $f_c = 1500\ MHz$ et $f_c = 500\ MHz$

Dans le cas d'un canal à trajets multiples, la sortie du canal (figure I.5), à l'instant nT, est modélisée à l'aide de l'expression suivante :

$$Y_n = \alpha_n X_n + z_n \tag{I.2}$$

Où :

- X_n représente le signal transmis à l'instant nT ;
- Y_n est le signal reçu correspondant ;
- α_n est l'atténuation aléatoire caractérisant le canal à l'instant nT;
- z_n est un bruit gaussien centré et indépendant (**AWGN)** de variance σ^2.

Si le canal est tel qu'aucun trajet direct n'est visible entre l'émetteur et le récepteur (transmission radio-mobile en milieu urbain par exemple), alors la variable α_n suit une distribution de Rayleigh.

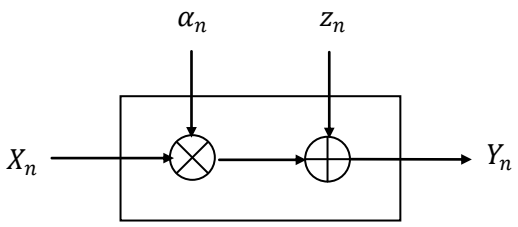


Figure I.5- Modèle d'un canal de Rayleigh

I.2.5.2 Canal gaussien

Ce type de canal ne déforme pas le signal mais lui ajoute un bruit **AWGN**. Ce modèle de bruit est le plus simple et en plus il fournit un modèle presque parfait dans certains systèmes de communication. Par exemple, pour les canaux satellitaires où les communications sont en vue directe, le modèle **AWGN** est exact. Mentionnons que l'adjectif "additif" dans **AWGN** signifie que l'impact du bruit sur le signal transmis peut être modélisé comme une variable aléatoire z_n qui s'additionne au signal transmis. La variable z_n est supposée gaussienne de moyenne nulle et de variance σ^2.

Dans le cas d'un canal gaussien l'expression (I.2) devient égale à :

$$Y_n = X_n + z_n \tag{I.3}$$

Dans le cas d'un canal gaussien α_n est une constante égale à 1.

I.2.6 Récepteur

Le récepteur doit à partir de signal reçu identifier le signal émis. Les blocs à la réception font essentiellement les fonctions inverses de ceux de l'émission.

Dans ce qui suit, on va exposer un type de codes correcteur d'erreurs: le code en bloc linéaire.

I.3 Codage d'un code en bloc

Le codage s'effectue par bloc de symboles, la séquence d'information est segmentée en bloc de longueur fixe, il y a *K* symboles dans chaque bloc. On va ajouter de la redondance, on va ajouter $M = N - K$ symboles, c'est-à-dire que le bloc final à transmettre possède *N* $(N > K)$ symboles. Les *M* symboles introduits par le codeur sont appelés symboles de contrôle ou de parité. Ces symboles offrent une grande diversité.

Un code en bloc est une application biunivoque entre des blocs de *K* symboles issus de la source d'informations, appelés message ou mot d'information, et des blocs de *N* symboles, appelés mot de code. La figure I.6 montre le principe de codage en bloc.

Chaque code en bloc est défini par la quantité suivante qui s'appelle le rendement:

$$R = \frac{nombre\ \ de\ symboles\ \ d'entr\acute{e}e}{nombre\ \ de\ symbole\ \ de\ sortie} = \frac{K}{N} \tag{I.4}$$

Les symboles prennent leurs valeurs dans un corps fini à *q* éléments, appelé corps de Galois $GF(q)$, dont les principales propriétés sont données dans l'annexe A. Dans ce chapitre nous allons considérer que les symboles sont binaires et prennent leur valeur dans le corps $GF(2)$ à deux éléments 0 et 1. Ce corps est le plus petit corps de Galois.

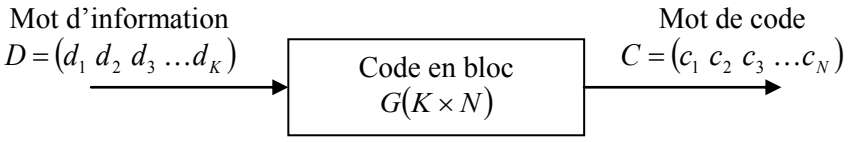


Figure I.6- Codage en bloc

G est la réponse du code définissant aussi les codes en bloc, et celle-ci est déterminée par une matrice génératrice de taille $K \times N$, où *K* : le nombre de colonnes et *N* : le nombre de lignes.

En effet, puisque le code est linéaire (la correspondance entre la sortie et l'entrée est une fonction linéaire), la convolution temporelle correspond à la multiplication polynomiale. On passe du message *M* au mot de code *C* par la matrice génératrice *G* [8] :

$$C = M.G \tag{I.5}$$

Avec *K* symboles d'entrée, on peut construire 2^K messages, le codeur va donc utiliser 2^K mots de code choisis parmi 2^N possibilités [9].

La matrice génératrice est toujours équivalente à une forme particulièrement simple, construisant des codes appelés codes systématiques. C'est le cas le plus utilisé, dont la matrice systématique s'obtient à partir de la matrice génératrice du code non-systématique.

- **Code en bloc systématique**

Un code est systématique si les symboles d'entrées sont toujours retrouvés intacts dans la séquence de symboles codés à la sortie du code. Le mot de code est représenté sous la forme suivante :

$$C = \left(c_{K+1} \ldots\ldots c_N, d_1 \ldots\ldots d_K\right) \tag{I.6}$$

La matrice génératrice *G* prend alors la forme suivante [10] :

$$G = \begin{pmatrix} q_{11} & \cdots & q_{1,N-K} & 1 & 0 & \cdots & 0 \\ q_{21} & \cdots & q_{2,N-K} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{K1} & \cdots & q_{K,N-K} & 1 & 0 & \cdots & 1 \end{pmatrix} \tag{I.7}$$

La matrice *G* peut se mettre sous la forme [10] :

$$G = \left(Q, I_K\right) \tag{I.8}$$

Où I_K est la matrice unité à K lignes et à K colonnes et Q une matrice à K ligne et $(N-K)$ colonnes qui caractérise le code. La figure I.7 illustre le schéma d'un code en bloc sous la forme systématique.

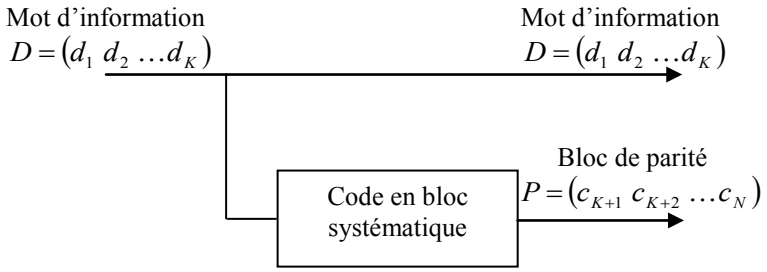


Figure I.7- Code en bloc systématique

Le codage est supposé systématique dans la suite de ce travail.

A partir de la matrice génératrice on peut déterminer une matrice définissant le décodeur d'un code en bloc et elle possède encore une forme particulièrement simple. Cette matrice est appelée matrice de contrôle de parité. Elle permet comme nous allons le voir de détecter les erreurs de transmission.

- **Matrice de contrôle de parité**

Pour une matrice systématique G de taille $K \times N$, avec K lignes indépendantes, on peut toujours trouver une matrice H de dimension $M \times N$, non nulle qui vérifie la relation suivante [11] :

$$G.H^T = 0 \tag{I.9}$$

Où l'indice T indique la transposition.

Compte-tenu de la matrice G, la matrice H est égale à :

$$H = \begin{pmatrix} 1 & 0 & \ldots & 0 & q_{11} & \ldots & q_{1,K} \\ 0 & 1 & \ldots & 0 & q_{21} & \ldots & q_{2,K} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & q_{N-K,1} & \ldots & q_{N-K,K} \end{pmatrix} \tag{I.10}$$

Qui peut encore s'écrire :

$$H = \left(I_{N-K}, Q^T\right) \tag{I.11}$$

Où I_{N-K} est une matrice unité à *M* ligne et à *M* colonnes.

Les mots de code doivent être suffisamment différents les uns des autres pour avoir de meilleures performances. Cette distinction est caractérisée par un paramètre important qui explique les performances d'un code correcteur d'erreurs en terme de probabilité d'erreur, appelé distance minimale.

I.4 Distance minimale

La distance minimale $d_{\min}$, est définie comme la plus petite distance de Hamming existant entre deux mots de code.

Rappelons que la distance de Hamming entre deux mots de code est égale au nombre de symboles binaires codés différents entre les séquences associées à chacun des deux mots de code.

La recherche de la distance minimale bénéficie de la propriété de linéarité des codes en bloc, qui permet de se concentrer uniquement sur les distances entre les mots de code potentiels et le mot de code zéro. Le mot de code "zéro" est le message codé contenant une séquence de zéros, générée par le codage d'une série de bits d'informations 'zéro'.

De ce fait, la distance minimale est aussi égale au poids minimal w_H des mots de code non-nuls [8]. Le poids d'une séquence est le nombre de bit égal à 1.

$$d_{\min} = \min\left\{w_H(C_i) \middle| C_i \neq 0\right\}; \quad i \in \left\{1, \ldots, 2^K - 1\right\} \tag{I.12}$$

Lorsque le nombre de mots de code est très élevé, la recherche de la distance minimale peut s'avérer laborieuse. Une première solution pour contourner cette difficulté est de déterminer la distance minimale à partir de la matrice de contrôle de parité [10].

Nous avons vu que la distance minimale est égal au poids de Hamming minimal des mots de code non-nuls. Considérons un mot de code de poids w_H, la propriété d'orthogonalité

$C.H^T = 0$ implique que la somme de $d_{\min}$ colonnes de la matrice de contrôle de parité est nulle. Ainsi $d_{\min}$ correspond au nombre minimal de colonnes de la matrice de contrôle de parité linéairement dépendantes.

Un bon code est un code avec une grande distance minimale, mais il doit pouvoir être bien décodé. Cette distance nous permet de déterminer la capacité de détection et correction de code.

I.5 Capacité de détection et correction d'un code en bloc

Pour une distance minimale $d_{\min}$ d'un code en bloc, le code pourra alors détecter t_d erreurs et corriger t_c erreurs :

- Le nombre d'erreurs à détecter est :

$$t_d = d_{\min} - 1 \tag{I.13}$$

- Le nombre d'erreurs à corriger est:

$$t_r = \begin{cases} \dfrac{d_{\min} - 1}{2} & si\ d_{\min}\ est\ impair \\ \dfrac{d_{\min} - 2}{2} & si\ d_{\min}\ est\ pair \end{cases} \tag{I.14}$$

La distance minimale de notre exemple est égale à 4. Dans ce cas, le code pourra détecter 3 erreurs et corriger 2 erreurs.

Les équations (I.13) et (I.14) montrent bien que la distance minimale détermine les performances d'un code en bloc.

I.6 Principe de la détection des erreurs

Considérons le modèle de transmission de la figure I.8.

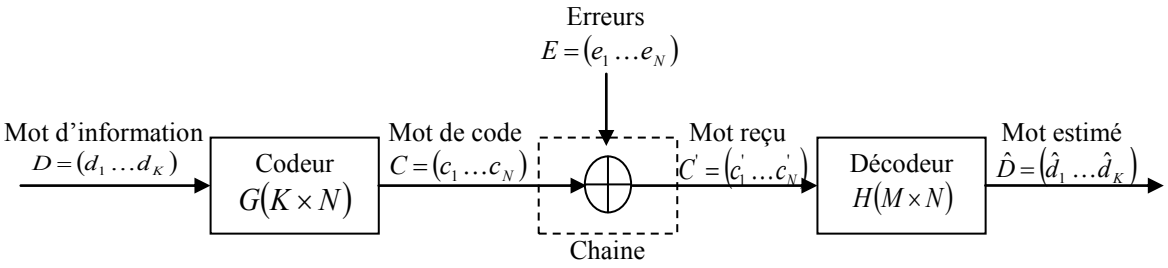


Figure I.8- Modèle de transmission simplifié utilisant le codeur et le décodeur en bloc

Soit C' le mot reçu, constitué de N symboles binaires lorsque le mot de code C est émis :

$$C' = C + E \tag{I.15}$$

Où E est un mot de N symboles binaires qui représente les éventuelles erreurs de transmission.

En utilisant la matrice de contrôle de parité pour détecter les erreurs de transmission. Une propriété importante pour n'importe qu'elle type de code en bloc est donnée par :

$$C'.H^T \neq 0 \Rightarrow C' \; n'est \; pas \; un \; mot \; de \; code \tag{I.16}$$

Si on multiplie le mot reçu C' par la matrice de contrôle de parité transposée, on obtient :

$$S = C'.H^T = C.H^T + E.H^T = M.G.H^T + E.H^T \tag{I.17}$$

En tenant compte que $G.H^T = 0$ (équation (I.9)), on obtient :

$$S = E.H^T \tag{I.18}$$

S s'appelle le syndrome, il permet de détecter les erreurs de transmission. Le syndrome S peut être nul ou pas, notons que H est une matrice non-nulle :

- **1er cas :** $S = 0$ signifie deux possibilités

- E est identiquement nul alors il n'y a pas d'erreurs de transmission ;
- E est égal à un mot de code. Dans ce cas on ne peut pas détecter les erreurs de transmission.

- **2ème cas :** $S \neq 0$

- E est différent de zéro et les erreurs de transmission sont détectées.

I.7 Principe de la correction des erreurs

La correction des erreurs consiste à rechercher le mot de code émis C c'est-à-dire à réaliser un décodage de C à partir du mot reçu C' . Deux stratégies sont possibles :

- Un décodage à entrée ferme ou pondéré et à sortie ferme ;

- Un décodage à sortie pondérée.

Un décodage à entrée ferme, c'est-à-dire l'entrée du décodeur est constituée par des symboles binaires, la règle de correction consiste à choisir pour un mot de code émis, celui qui est à la distance de Hamming minimale du mot reçu.

Un décodage à entrée pondérée, c'est-à-dire l'entrée du décodeur est constituée par des échantillons analogiques, on remplace la distance de Hamming par la distance Euclidienne.

Chaque type de codes en bloc à ses propres algorithmes de décodage. Le bon code est celui qui corrige plus d'erreurs et à moins de complexité de décodage.

La tendance actuelle étant à l'augmentation des débits de transmission et les bandes passantes disponibles étant limitées. Les codes à rendement élevé sont, en terme de débit, les meilleurs codes puisqu'ils introduisent moins de redondances mais en même temps perdent leurs intérêts d'un point de vue capacité de correction. Pour obtenir un code à rendement élevé

et au pouvoir de correction convenable, il faudrait modifier la structure du codeur et ceci en augmentant alors la complexité du décodeur.

De ce fait, il existe une technique simple, appelée poinçonnage [12, 13, 14], qui permettra d'augmenter le rendement d'un code sans pour autant augmenter la complexité du décodeur. Cette technique permet d'obtenir des codes ayant des performances très voisines de celles obtenues avec des codes non-poinçonnés de même rendement, tout en présentant l'avantage d'avoir une réduction de la complexité de décodage.

I.8 Code en bloc poinçonné

A l'aide de la technique de poinçonnage, le codeur et le décodeur sont identiques pour l'ensemble des rendements de codage. Seule la fonction de poinçonnage est paramétrable en fonction de rendement.

La technique de poinçonnage est placée à la sortie du codeur. Elle permet d'effacer, c'est-à-dire de ne pas transmettre, certains symboles codés. En réception, un dispositif, nommé dépoinçonnage, insère des zéros analogiques en entrée du décodeur aux places correspondantes.

La figure I.9 représente le schéma de principe d'un système de transmission utilisant la technique de poinçonnage.

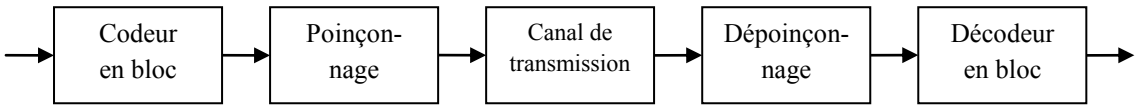


Figure I.9- Schéma d'un système de transmission numérique utilisant le poinçonnage

La technique de poinçonnage permet d'augmenter le rendement d'un code sans augmenter la complexité de décodage. Par l'intermédiaire de la concaténation de codes, le rendement d'un code peut être abaissé avec moins de complexité de décodage. Notons que le poinçonnage et la concaténation de codes sont utilisés pour n'importe quel type de code correcteur d'erreurs.

I.9 Concaténation de codes

La concaténation de codes est la combinaison de deux ou plusieurs codes de faible longueur de bloc de données afin d'obtenir un code puissant avec moins de complexité qu'un seul code atteignant ces performances. Ainsi, le décodeur est constitué de deux ou plusieurs décodeurs de complexité modérée.

Les concaténations les plus célèbres sont : la concaténation parallèle [15], la concaténation série [16]. Il est possible de fabriquer des concaténations hybrides [17] en combinant la concaténation série et la concaténation parallèle. Les structures hybrides sont peu intéressantes car un très grand gain en performance est déjà atteint par les structures classiques. Dans la chapitre IV on va utiliser la concaténation parallèle.

La concaténation parallèle a été introduite par Berrou et autres [18] pour élaborer les turbo-codes. Les turbo-codes permettent d'avoisiner la limite théorique de Shannon (Voir l'annexe B pour la démonstration de la capacité du canal déterminée par Shannon).

Les turbo-codes sont obtenus par la concaténation parallèle [18], série [19] ou hybride de deux ou plusieurs codes correcteurs d'erreurs de faible complexité. Leur décodage fait appel à un processus itératif (ou turbo) qui permet de tirer la puissance des codes concaténés. Les turbo-codes peuvent être des turbo-codes en bloc [20, 21] ou des turbo-codes convolutifs [18] selon le type des codes concaténés.

I.10 Conclusion

Un système de transmission numérique a pour but de reproduire à un point, un message qui a été délivré à partir d'un autre point. Le concepteur d'un tel système doit garantir la fiabilité de la communication. Cette fiabilité est mesurée sous forme de taux d'erreurs binaires pour un certain niveau de rapport signal à bruit.

Le code en bloc a été examiné, dans ce chapitre, sous une forme générale en guise d'introduction à l'étude d'un cas particulier de codes dont il sera question dans le prochain chapitre.

Chapitre II

*Code **LDPC** **B**inaire et **N**on-**B**inaire*

II.1 Introduction

Parmi tous les codes existants permettant d'approcher la limite de Shannon [22] sont les codes **LDPC** et les turbo-codes. Les codes **LDPC** ont été réintroduits après la puissance du décodage itératif qui a été mise en évidence grâce à l'invention des turbo-codes.

Ce chapitre s'intéresse aux codes **LDPC**. Ces codes sont des codes en bloc linéaires. Les codes **LDPC** binaires sont proposés par Gallager en 1962 [4, 34], et redécouverts par Mackay en 1995 [23, 24]. Malheureusement, les codes **LDPC** binaires montrent un affaiblissement de performances quand les blocs de données sont petits ou moyens et la modulation utilisée pour la transmission est à grand nombre d'états. De ce fait, Les codes **LDPC** non-binaires, définis dans un corps de Galois fini $GF(q)$, sont étudiés par Davey and Mackay [25] pour éviter cette affaiblissement.

Dans ce chapitre, on étudie le principe d'un code **LDPC** binaire et non-binaire et leur décodage. Tout d'abord, on explique les codes **LDPC** réguliers et irréguliers. Ensuite, la représentation graphique des codes **LDPC** est représentée. Puis, on discute les algorithmes de décodage des codes **LDPC** binaires et non-binaires.

II.2 Définition d'un code LDPC

Les codes **LDPC** sont des codes en bloc linéaires, basés sur des matrices de contrôle de parité de faible densité, ou creuse, c'est-à-dire le nombre des éléments différents de zéro de la matrice est beaucoup inférieur au nombre de 0.

Les éléments différents de zéro de la matrice peuvent être des éléments binaires ou non-binaires. Par conséquent, nous avons des codes **LDPC** binaires et non-binaires. Un code **LDPC** binaire est défini sur un corps de Galois fini d'ordre 2 $GF(2)$. Tandis qu'un code **LDPC** non-binaire est défini sur un corps de Galois fini d'ordre q $GF(q)$.

On considère, dans ce travail, les caractéristiques de ce corps qui possèdent des puissances de deux: $q = 2^p$, où p est un entier positif.

Les codes **LDPC** peuvent être réguliers [26] ou irréguliers selon la distribution régulière ou irrégulière de ces éléments dans la matrice [27].

II.3 Code LDPC régulier

Un code **LDPC** est défini par une matrice de contrôle de parité H de dimension $M \times N$. Le nombre de colonnes, représenté par N, définit la longueur de code. Le nombre de lignes, représenté par M, détermine le nombre d'équations de contrôle de parité de code.

La matrice de contrôle de parité d'un code **LDPC** régulier a des poids de colonnes et de lignes constants. Le poids de la colonne, noté w_c, et le poids de la ligne, noté w_l, représentent respectivement le nombre d'éléments non-nuls par colonne et par ligne.

Un code **LDPC** régulier est noté code **LDPC** régulier (w_c, w_l). Le rendement de ce code est donné par [28] :

$$R = \frac{N - M}{N} = 1 - \frac{w_c}{w_l} \tag{II.1}$$

On a la relation suivante:

$$N = \frac{w_l}{w_c} M \tag{II.2}$$

Dans le cas des codes **LDPC** non-binaires, les valeurs non-nulles de la matrice de contrôle de parité sont choisies uniformément dans $GF(q)$.

La matrice définie dans l'équation (II.3) est une matrice de contrôle de parité H de dimension *4×6*, d'un code **LDPC** régulier $(2, 3)$.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & 0 & 0 & 0 \\ 0 & h_{22} & 0 & h_{24} & h_{25} & 0 \\ h_{31} & 0 & 0 & h_{34} & 0 & h_{36} \\ 0 & 0 & h_{43} & 0 & h_{45} & h_{46} \end{bmatrix} \tag{II.3}$$

Où $h_{ij}, i = 1...4, j = 1...6$ sont les éléments non-nuls de *H* de la ligne *i* et de la colonne *j*.

Prenons un exemple d'une matrice de contrôle de parité *H* de dimension *5*×10, d'un code **LDPC** régulier $(2, 4)$, défini dans $GF(4)$:

$$H = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 1 \\ 3 & 0 & 1 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 3 & 3 & 0 \end{pmatrix} \quad \text{(II.4)}$$

II.4 Code LDPC irrégulier

L'irrégularité d'un code **LDPC** irrégulier se traduit par la distribution non-uniforme des éléments non-nuls sur les colonnes et/ou sur les lignes.

La matrice définie dans l'équation (II.5) est une matrice de contrôle de parité *H* de dimension 4×8, d'un code **LDPC** irrégulier.

$$H = \begin{bmatrix} h_{11} & 0 & 0 & 0 & h_{15} & 0 & 0 & 0 \\ 0 & h_{22} & 0 & 0 & h_{25} & h_{26} & 0 & 0 \\ 0 & 0 & h_{33} & 0 & 0 & h_{36} & h_{37} & 0 \\ 0 & 0 & 0 & h_{44} & 0 & 0 & h_{47} & h_{48} \end{bmatrix} \quad \text{(II.5)}$$

Où $h_{ij}, i = 1...4, j = 1...8,$ sont les éléments non-nuls de *H* de ligne *i* et de colonne *j*.

Un code **LDPC** irrégulier est défini par deux séquences λ_i et ρ_j. Les valeurs λ_i comptabilisent le nombre de valeurs non-nulles associées à une colonne de poids *i*. Les valeurs ρ_j sont définies de la même façon pour les lignes.

Ces paramètres sont directement reliés aux performances des codes **LDPC**. De plus il est possible de les relier à la proportion de colonnes de poids *i* et à la proportion de lignes de poids *j*. Cette dernière paramétrisation facilite les algorithmes de construction des matrices de parité.

Naturellement les poids maximaux des lignes et des colonnes doivent être limités pour que l'ensemble des matrices de parité ainsi paramétrées restent des matrices creuses.

Une paramétrisation usuelle d'un code **LDPC** irrégulier est déterminée par :

- Profil d'irrégularité des nœuds de variable [29] :

$$\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1} \tag{II.6}$$

Avec λ_i est égal au rapport entre le nombre cumulé de valeurs non-nulles des colonnes de degré *i* et le nombre total de valeurs non-nulles de la matrice *H*.

- Profil d'irrégularité des nœuds des parités [29] :

$$\rho(x) = \sum_{j \geq 2} \rho_j x^{j-1} \tag{II.7}$$

Avec ρ_j est égal au rapport entre le nombre cumulé de valeurs non-nulles des lignes de degré *j* et le nombre total de valeurs non-nulles de la matrice *H*.

Le rendement d'un code **LDPC** irrégulier est donné par :

$$R = 1 - \frac{\sum_{j \geq 2} \rho_j / j}{\sum_{i \geq 1} \lambda_i / i} \tag{II.8}$$

Il y a également une autre paramétrisation, désignée par :

- Profil d'irrégularité des nœuds de variable :

$$\widetilde{\lambda}(x) = \sum_{i \geq 1} \widetilde{\lambda}_i x^{i-1} \text{, avec } \widetilde{\lambda}_i = \frac{\lambda_i / i}{\sum_k \lambda_k / k} \tag{II.9}$$

- Profil d'irrégularité des nœuds des parités

$$\widetilde{\rho}(x) = \sum_{j \geq 2} \widetilde{\rho}_j x^{j-1} \text{ , avec } \widetilde{\rho}_j = \frac{\rho_j / j}{\sum_k \rho_k / k} \tag{II.10}$$

Les codes **LDPC** irréguliers sont ainsi paramétrés par $(N, \lambda(x), \rho(x))$ ou $(N, \tilde{\lambda}(x), \tilde{\rho}(x))$.

Exemple

Le code défini par la matrice de contrôle de parité de l'équation (II.5) a une distribution des degrés égale à :

$$\begin{aligned}\lambda(x) &= \lambda_1 + \lambda_1 + \lambda_1 + \lambda_1 + \lambda_2 x + \lambda_2 x + \lambda_2 x + \lambda_1 \\ &= 5\lambda_1 + 3\lambda_2 x \\ &= 5\frac{1}{11} + 3\frac{2}{11}x \\ &= \frac{5}{11} + \frac{6}{11}x\end{aligned} \tag{II.11}$$

$$\begin{aligned}\rho(x) &= \rho_2 x + \rho_3 x^2 + \rho_3 x^2 + \rho_3 x^2 \\ &= \rho_1 x + 3\rho_3 x^2 \\ &= \frac{2}{11}x + 3\frac{3}{11}x^2 \\ &= \frac{2}{11}x + \frac{9}{11}x^2\end{aligned} \tag{II.12}$$

Si la distribution de l'irrégularité est bien choisie, alors les codes **LDPC** irréguliers présentent de meilleures performances que les codes réguliers [30].

Les codes **LDPC** sont définis par leur matrice de contrôle de parité. Ainsi, ils peuvent être décrits par une représentation graphique, appelée graphe Tanner [31]. Cette représentation est utilisée comme un support par le décodeur [32] qui est exécuté par plusieurs algorithmes et décodé itérativement.

II.5 Représentation graphique d'un code LDPC : graphe de Tanner

Les graphes de Tanner, sont des graphes bipartites, composés de deux types de nœuds : les nœuds de variable représentant les symboles du mot de code et les nœuds de parité représentant les équations de contrôle de parité. Ces deux types de nœuds sont connectés par des branches selon les éléments non-nuls de la matrice de contrôle de parité.

Le nombre des nœuds de variable $N_m \left(m \in \{1,\ldots,N\}\right)$ et des nœuds de parité $M_n \left(n \in \{1,\ldots,M\}\right)$ correspondent, respectivement au nombre de colonnes *N* et au nombre de lignes *M* de la matrice *H*.

La figure (II.1) représente un exemple de graphe de Tanner correspond à la matrice *H* de l'équation (II.3)

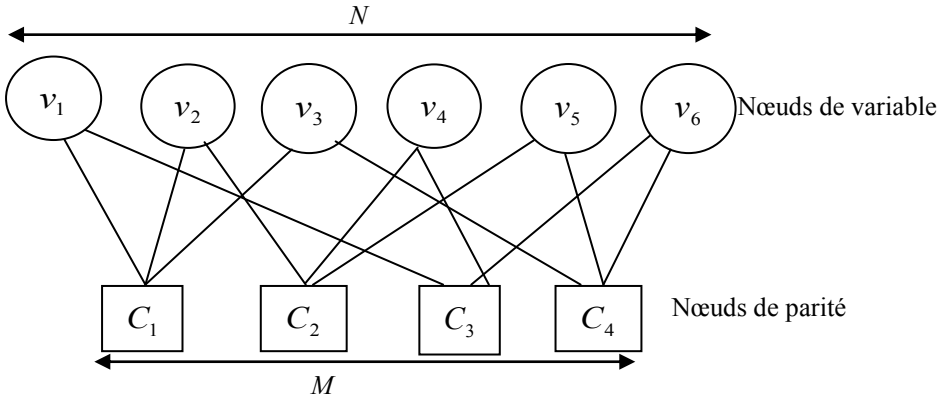


Figure II.1- Graphe de Tanner correspond à la matrice *H* de l'équation (II.3)

Dans la figure (II.1), il y a 6 nœuds de variable et 4 nœuds de parité correspondent respectivement au nombre de colonnes et au nombre de lignes de la matrice de l'équation (II.3). La connexion entre ces deux types de nœuds correspond aux valeurs non-nulles $h_{ij}, i = 1...4, j = 1...6$. La *iéme* nœud de parité et la *jéme* nœud de variable sont connectées si $h_{ij} \neq 0$.

Si on a une distribution non-uniforme du nombre de branches connectées aux différents types de nœuds sur le graphe de Tanner, le code est un code **LDPC** irrégulier.

La matrice de contrôle de parité *H* nous a permis de déterminer le graphe de Tanner qui est utilisé comme un support pour le décodeur. Ainsi, cette matrice est utilisée pour faire le codage des codes **LDPC**.

II.6 Principe de codage d'un code LDPC

Il existe plusieurs techniques de codage et les chercheurs essayent de proposer toujours des techniques qui permettent de réduire la complexité calculatoire de cette opération. L'une des méthodes classiques est celle du codage par décomposition LU. Ce type de codage est systématique. Cela signifie que le mot de code est sous la forme d'une concaténation d'un mot d'information C_I, de taille *N-M* symboles, et d'un bloc de redondance C_R :

$$C = \begin{bmatrix} C_R \\ C_I \end{bmatrix} \tag{II.13}$$

Pour une matrice de contrôle de parité *H,* les mots de code doivent vérifier la relation suivante:

$$C.H = 0 \tag{II.14}$$

L'opération de codage utilise cette relation. On décompose H en deux sous-matrices H_1 et H_2 telle que H est la concaténation de H_1 et H_2.

$$H = [H_1 \quad H_2] \tag{II.15}$$

H_1 est une matrice carrée de dimension $M \times M$ et occupe les M premières colonnes de H, tandis que H_2 est de dimension $M \times (N - M)$ et occupe les *N-M* dernières colonnes.

En appliquant les équations (II.13) et (II.15) dans la relation (II.14), on obtient :

$$C_R.H_1 + C_I.H_2 = 0 \tag{II.16}$$

C_I est déjà connu, c'est le mot d'information. Reste à calculer C_R par l'opération suivante:

$$C_R = inv\,(H_1).C_I.H_2 \tag{II.17}$$

Avec $inv\,(H_1)$ est l'inverse de H_1.

La détermination de $inv\,(H_1)$ est faisable mais cette opération est lourde. En plus, comme H_1 est creuse, algébriquement on peut démontrer que $inv\,(H_1)$ sera dense autrement-

dit contient beaucoup de 1. Par conséquent, la multiplication par $inv\ (H_1)$ prend plusieurs temps. Pour éviter ce calcul, on écrit H_1 sous la forme :

$$H_1 = L.U \tag{II.18}$$

Avec L et U sont respectivement matrice triangulaire inferieure et matrice triangulaire supérieure. C'est la décomposition LU.

Par conséquent, l'équation (II.16) devient :

$$\begin{aligned} C_R..L.U + C_I.H_2 = 0 &\Rightarrow C_R.L.U = C_I.H_2 \\ &\Rightarrow L\underbrace{(C_R.U)}_{Y} = \underbrace{C_I.H_2}_{Z} \\ &\Rightarrow L.Y = Z \\ &\Rightarrow Y = Z/Y \end{aligned} \tag{II.19}$$

Donc le bloc de redondance est :

$$C_R = Y/U \tag{II.20}$$

Exemple

Prenons la matrice de l'équation (II.4), pour un bloc d'information de longueur 5 :

$$C_I = [2\ 3\ 0\ 1\ 2]^T \tag{II.21}$$

La décomposition de H, nous donne :

$$H_1 = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},\ H_2 = \begin{pmatrix} 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 0 & 1 & 3 & 3 & 0 \end{pmatrix} \tag{II.22}$$

La factorisation LU de H_1, nous donne :

$$L=\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\3&3&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix},U=\begin{pmatrix}1&3&1&1&0\\0&3&0&0&0\\0&0&2&2&0\\0&0&0&0&2\\0&0&0&0&1\end{pmatrix} \quad \text{(II.23)}$$

En utilisant les équations (II.19) et (II.20), on calcule le bloc de redondance C_R :

$$Z=C_I.H_2=\begin{bmatrix}2\\1\\2\\1\\0\end{bmatrix},Y=Z/L=\begin{bmatrix}2\\1\\0\\1\\0\end{bmatrix}\text{ et }C_R=Y/U=\begin{bmatrix}1\\2\\2\\1\\0\end{bmatrix} \quad \text{(II.24)}$$

Donc le mot de code est :

$$C=\begin{bmatrix}C_R\ C_I\end{bmatrix}^T=\begin{bmatrix}1&2&2&1&0&2&3&0&1&2\end{bmatrix}^T \quad \text{(II.25)}$$

II.7 Algorithmes de décodage d'un code LDPC

Le principe des algorithmes de décodage d'un code correcteur d'erreurs consiste à estimer la séquence émise. L'estimation se base sur le maximum de vraisemblance. Le maximum de vraisemblance permet d'identifier la séquence la plus probable par la comparaison des séquences transmises possibles pour une séquence reçue.

Cependant, pour des longs blocs de données cette comparaison devient plus compliquée. Plusieurs solutions ont été proposées pour rendre cette tache moins complexe, et exploite la structure des codes afin d'accélérer le décodage, comme pour les codes **LDPC** qui ne sont pas un maximum de vraisemblance mais qui peuvent être très bien exécutés avec une multitude de complexité réduite.

Une classe des algorithmes utilisés pour décoder les codes **LDPC** sont communément appelés des algorithmes de propagation de message [33] puisqu'ils emploient le graphe de Tanner comme un support et leurs opérations peuvent être expliquées par le passage des

messages le long des branches d'un graphe de Tanner. Les algorithmes de propagation de message sont également connus en tant qu'algorithmes de décodage itératif.

Le premier algorithme de décodage itératif pratique, est l'algorithme Somme-Produit (**SP**), appelé aussi l'algorithme de propagation de croyance, est un algorithme de décodage itératif optimal mais avec une complexité de calcul élevée.

Plusieurs algorithmes ont été proposés pour réduire la complexité de l'algorithme **SP**: l'algorithme **FFT-SP** [34] dans le domaine fréquentiel basé sur la transformé de Fourier rapide (**FFT**); l'algorithme **SP** dans le domaine logarithmique [35] tels que : l'algorithme **MS** pour les codes **LDPC** binaires et l'algorithme **EMS** pour les codes **LDPC** non-binaires [36, 37, 38]; l'algorithme **SP** dans le domaine mixte [39] : domaines fréquentiels et logarithmiques, afin de prendre l'avantage de ces deux domaines.

Les messages échangés dans l'algorithme **SP** et ses versions peuvent être mesurées par la probabilité a posteriori (A Posteriori Probability ou **APP** en anglais) ou par le rapport de vraisemblance logarithmique (Log-Likelihood Ratio ou **LLR** en anglais) selon le type d'algorithme.

Dans ce qui suit, on décrit le principe de l'algorithme de propagation de message. Ensuite, on présente l'algorithme **SP** et ses quelques versions dérivées pour les codes **LDPC** binaires et non-binaires.

II.7.1 Principe de l'algorithme de propagation de message

L'algorithme propagation de message peut se décomposer en plusieurs étapes :

a. Initialisation des nœuds de variables par les messages d'entrée du décodeur.

b. A chaque itération les étapes suivantes se répètent :

1. Chaque nœud de parité reçoit les messages arrivant des nœuds de variable qui lui sont reliées par des branches, puis calcule et envoie le message résultant qui est lié à tous les messages arrivant des nœuds de variable excepté le message où le message de sortie sera envoyé ;

2. Chaque nœud de variable reçoit les messages arrivant des nœuds de parité qui lui sont reliées par les branches, puis calcule et envoie le message résultant qui est lié à tous les messages arrivant des nœuds de parité excepté le message où le message de sortie sera envoyé ;

3. Après, l'information a posteriori associée à chaque nœud de variable est calculée avant la prise de décision.

Enfin, après un certain nombre d'itérations ou dans le cas où le syndrome est nul (voir le chapitre I), l'algorithme s'arrête.

L'organigramme de l'algorithme propagation de message est présenté dans l'annexe C.

II.7.2 Algorithmes de décodage des codes LDPC binaires

II.7.2.1 Algorithme SP

Cet algorithme exécute les opérations suivantes:

- Initialisation

Les nœuds de variable v_n, $n\epsilon\{1,\dots,N\}$, sont initialisés par les probabilités $Pr(v_n = x|c_n^{'})$, $x \in GF(2)$. Avec $Pr(v_n = x|c_n^{'})$ est la probabilité du $n^{\text{ième}}$ symbole de mot de code et est égale à x sachant que le $n^{\text{ième}}$ symbole $c_n^{'}$ est reçu.

Les calculs des nœuds de variable sont représentés dans une matrice V de taille $(M \times N)$:

$$V = \begin{pmatrix} \mu_{11}(x) & \mu_{12}(x) & \dots & \mu_{1N}(x) \\ \mu_{21}(x) & \mu_{22}(x) & \dots & \mu_{2N}(x) \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{M1}(x) & \mu_{M2}(x) & \dots & \mu_{MN}(x) \end{pmatrix} \tag{II.26}$$

Où $\mu_{mn}(x) = Pr(v_n = x|c_n^{'})$, $x \in GF(2)$, $m\epsilon M_n$ $(M_n = \{1,\dots,M\})$ et $n\epsilon N_m$ $(N_m = \{1,\dots,N\})$, est un vecteur colonne de longueur 2, représente le calcul de chaque nœud de variable [40] :

$$\mu_{mn}(x) = \begin{pmatrix} Pr(v_n = 0|c_n^{'}) \\ Pr(v_n = 1|c_n^{'}) \end{pmatrix} \tag{II.27}$$

Dans l'initialisation toutes les lignes de la matrice V sont égales à la première ligne.

La probabilité $Pr(v_n = x|c_n^{'})$, $x \in GF(2)$, dépend des informations du canal : le type du canal et la modulation utilisée (voir le chapitre III).

- Calcul et mis à jour des nœuds de parité

Les nœuds de parité calculent les probabilités $Pr(C_m = 0|v_n = x, c_n^{'})$, qui dépendent de tous les symboles qui participent dans la $m^{\text{ième}}$ équation de parité C_m sauf le $n^{\text{ième}}$ symbole $\{v_{n'} = x_{n'} : n' \in N_m/n\}$. Ces calculs sont représentés dans une matrice B, de taille $(M \times N)$, définissant les nœuds de parité :

$$B = \begin{pmatrix} \beta_{11}(x) & \beta_{12}(x) & \dots & \beta_{1N}(x) \\ \beta_{21}(x) & \beta_{22}(x) & \dots & \beta_{2N}(x) \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{M1}(x) & r_{M2}(x) & \dots & \beta_{MN}(x) \end{pmatrix} \tag{II.28}$$

Où $\beta_{mn}(x) = Pr(C_m = 0|v_n = x, c_n^{'})$, $x \in GF(2)$, $m\epsilon\{1, \dots, M\}$ et $n\epsilon\{1, \dots, N\}$, est un vecteur colonne de longueur 2, représente le calcul de chaque nœud de parité:

$$\beta_{mn}(x) = \begin{pmatrix} Pr(C_m = 0|v_n = 0, c_n^{'}) \\ Pr(C_m = 0|v_n = 1, c_n^{'}) \end{pmatrix} \tag{II.29}$$

La probabilité $Pr(C_m = 0|v_n = x, c_n^{'})$ peut être calculée comme suit [11] :

$$\begin{aligned} \beta_{mn}(x) &= Pr(C_m = 0|v_n = x, c_n^{'}) \\ &= \sum_{\{x_{n'}:n' \in N_m/n\}} Pr(C_m = 0, \{v_{n'} = x_{n'}, : n^{'} \in N_m/n\}|v_n = x, c_n^{'}) \\ &= \sum_{\{x_{n'}:n' \in N_m/n\}} Pr(C_m = 0|v_n = x, \{v_{n'} = x_{n'}, : n^{'} \in N_m/n\}, c_n^{'}) \times \\ &\quad Pr(v_{n'} = x_{n'}, : n^{'} \in N_m/n|\, c_n^{'}) \\ &= \sum_{\{x_{n'}:n' \in N_m/n\}} Pr(C_m = 0|v_n = x, \{v_{n'} = x_{n'}, : n^{'} \in N_m/n\}) \times \\ &\quad \prod_{l' \in N_m/n} Pr(v_{l'} = x_{l'} \mid c_n^{'}) \\ &= \sum_{\{\{x_{n'}:n' \in N_m/n\}:x=\sum_l x_l\}} \prod_{l \in N_m/n} Pr(v_l = x_l|\, c_l^{'}) \end{aligned} \tag{II.30}$$

En appliquant $\mu_{mn}(x) = Pr(v_n = x|c_n^{'})$ dans l'équation précédente, on obtient :

$$\beta_{mn}(x) = \sum_{\{\{x_{n'}:n'\in N_m/n\}:x=\sum_l x_l\}} \prod_{l\in(N_m/n)} \mu_{mn}(x_l) \quad \text{(II.31)}$$

▪ Calcul et mis à jour des nœuds de variable

Les nœuds de variable calculent les probabilités $Pr(v_n = x|c_n^{'},\ \{C_{m'} = 0, m' \in M_n/m\})$, $x \in GF(2)$, $m\epsilon\{1,\dots,M\}$ et $n\epsilon\{1,\dots,N\}$. Avec $Pr(v_n = x|c_n^{'},\ \{C_{m'} = 0, m' \in M_n/m\})$ est la probabilité du $n^{\text{ième}}$ symbole de mot de code est égale x conditionnel le $n^{\text{ième}}$ symbole $c_n^{'}$ du mot reçu et toutes les équations de parité qui produisent v_n sauf l'équation C_m.

Donc, la matrice V de l'équation (II.26), définissant les nœuds de variable, prend ces calculs. Où chaque élément $\mu_{mn}(x)$ de cette matrice est calculé comme suit [11]:

$$\begin{aligned}\mu_{mn}(x) &= Pr(v_n = x|c_n^{'}, \{C_{m'} = 0, m' \in M_n/m\}) \\ &= \frac{Pr(v_n=x,\{C_{m'}=0,m'\in M_n/m\}|c_n^{'})}{Pr(\{C_{m'}=0,m'\in M_n/m\}|c_n^{'})} \\ &= \frac{Pr(v_n=x|c_n^{'})Pr(\{C_{m'}=0,m'\in M_n/m\}|v_n=x,c_n^{'})}{Pr(\{C_{m'}=0,m'\in M_n/m\}|c_n^{'})} \\ &= \frac{Pr(v_n=x|c_n^{'})\prod_{m'\in M_n/m} Pr(C_{m'}=0|v_n=x,c_n^{'})}{Pr(\{C_{m'}=0,m'\in M_n/m\}|c_n^{'})}\end{aligned} \quad \text{(II.33)}$$

Avec $Pr(\{C_{m'} = 0, m' \in M_n/m\}|c_n^{'}) = \sum_x Pr(v_n = x|c_n^{'}) \prod_{m'\in M_n/m} Pr(C_{m'} = 0|v_n = x, c_n^{'})$

$$\mu_{mn}(x) = \frac{Pr(v_n=x|c_n^{'})\prod_{m'\in M_n/m} Pr(C_{m'}=0|v_n=x,c_n^{'})}{\sum_x Pr(v_n=x|c_n^{'})\prod_{m'\in M_n/m} Pr(C_{m'}=0|v_n=x,c_n^{'})} \quad \text{(II.33)}$$

En appliquant $\beta_{m'n}(x) = Pr(C_{m'} = 0|v_n = x, c_n^{'})$ dans l'équation (II.39), et on met $\eta = \frac{1}{\sum_x Pr(v_n=x|c_n^{'})\prod_{m'\in M_n/m} Pr(C_{m'}=0|v_n=x,c_n^{'})}$, l'équation (II.39) devient:

$$\mu_{mn}(x) = \eta . Pr(v_n = 0|\, c_n^{'}) \prod_{m'\in M_n/m} \beta_{m'n}(x) \quad \text{(II.34)}$$

▪ Information a posteriori

L'information a posteriori est aussi déterminée par la probabilité $Pr(v_n = x|c_n^{'},\ \{C_m = 0, m \in M_n\})$. Avec $Pr(v_n = x|c_n^{'},\ \{C_m = 0, m \in M_n\})$ est la

probabilité du $n^{\text{ième}}$ symbole de mot de code est égale x conditionnel le $n^{\text{ième}}$ symbole $c_n^{'}$ de mot reçu et toutes les équations de parité C_m qui produisent v_n.

Les probabilités $\mu_n(x) = Pr(v_n = x|c_n^{'},\ \{C_m = 0, m \in M_n\})$ sont représentées dans un vecteur ligne Λ de longueur N :

$$\Lambda = [\mu_1(x)\ \mu_2(x) \dots\ \mu_N(x)] \tag{II.35}$$

Où chaque élément $\mu_n(x)$ contient :

$$\mu_n(x) = \begin{pmatrix} \mu_n(0) \\ \mu_n(1) \end{pmatrix} \tag{II.36}$$

- Décision

A partir des probabilités a postériori $\mu_n(x)$, $x \in GF(2)$, on peut estimer le mot de code transmis :

$$\hat{v}_n = arg\max_x\left(\mu_n(x)\right) \tag{II.37}$$

Donc le mot de code estimé est :

$$\hat{v} = \left(arg\max_x\left(\mu_1(x)\right)\ arg\max_x\left(\mu_2(x)\right)\right) \dots\ arg\max_x\left(\mu_N(x)\right) \tag{II.38}$$

Afin de simplifier la complexité des calculs, on utilise le logarithme des rapports de vraisemblance (**LLR**) des probabilités échangées entre les nœuds. Pour ceci, on discute dans la partie suivante l'algorithme **SP** utilisant le **LLR**. Ainsi, on présente sa dérivée l'algorithme **MS**. Puis, L'algorithme **FFT-SP** [41] est discuté pour les codes **LDPC** binaires et non-binaires.

II.7.2.2 Algorithme LLR-SP

Cet algorithme exécute les opérations suivantes [42]:

- Initialisation des nœuds de variable

$$\gamma_n(x) = log\frac{Pr\left(v_n=1|c_n^{'}\right)}{Pr\left(v_n=0|c_n^{'}\right)}, \qquad m\epsilon\{1,\dots,M\} \text{ et } n\epsilon\{1,\dots,N\} \tag{II.39}$$

$$\mu_{mn}(x) = \gamma_n(x) \tag{II.40}$$

- Itération

▪ Calcul des nœuds de parité

$$\beta_{mn} = 2 \times \tanh^{-1}\left(\prod_{n' \in N_m / n} \tanh(\mu_{mn'}/2) \right) \tag{II.41}$$

▪ Calcul des nœuds de variable

$$\mu_{mn} = \gamma_n + \sum_{m' \in M_n / m} \beta_{m'n} \tag{II.42}$$

▪ Information a posteriori

$$\widetilde{\gamma} = \gamma + \sum_{m \in M_n} \beta_{mn} \tag{II.43}$$

▪ Décision

$$\hat{c}_n = \arg\max_x (\widetilde{\gamma}_n) \tag{II.44}$$

L'algorithme **LLR-SP** atteints des meilleures performances pour les codes **LDPC**. Mais les opérations de $\tanh^{-1}$ et $\tanh$ sont trôp complexe à réaliser. Autrement, l'algorithme **MS** fait des approximations afin de simplifier les calculs des nœuds de parité. Cet algorithme est utilisé pour les codes **LDPC** binaires. L'extension de cet algorithme (l'algorithme **EMS**) est utilisée pour les codes **LDPC** non-binaires.

II.7.2.3 Algorithme MS

L'algorithme **MS** exécute les opérations suivantes [43]:

▪ Initialisation des nœuds de variable

$$\gamma_n(x) = log \frac{Pr(v_n=1|c_n')}{Pr(v_n=0|c_n')}, \qquad m\epsilon\{1, \dots, M\} \text{ et } n\epsilon\{1, \dots, N\} \tag{II.45}$$

$$\mu_{mn}(x) = \gamma_n \tag{II.46}$$

- Itération

$$\mu_{mn} = \mu_{mn}^{*} \times \mu_{mn}^{**} \text{ , avec } \mu_{mn}^{*} = sign(\gamma_n) \text{ et } \mu_{mn}^{**} = |\gamma_n| \tag{II.47}$$

- Itération

▪ Calcul des nœuds de parité

$$\beta_{mn} = \prod_{n' \in N_m / n} \mu_{mn'}^{*} \times 2 \times \tanh^{-1}\left(\prod_{n' \in N_m / n} \tanh\left(\mu_{mn'}^{**}/2\right) \right) \text{ pour l'algorithme } \mathbf{SP} \tag{II.48}$$

$$\beta_{mn} = \prod_{n' \in N_m / n} \mu_{mn'}^{*} \times \min_{n' \in N_m / n} \left(\mu_{mn'}^{**}\right) \text{ pour l'algorithme } \mathbf{MS} \tag{II.49}$$

▪ Calcul des nœuds de variable

$$\mu_{mn} = \gamma_n + \sum_{m' \in M_n / m} \beta_{m'n} \tag{II.50}$$

▪ Information a posteriori

$$\widetilde{\gamma} = \gamma + \sum_{m \in M_n} \beta_{mn} \tag{II.51}$$

▪ Décision

$$\hat{c}_n = \arg\max_{x}\left(\widetilde{\gamma}_n\right) \tag{II.52}$$

Finalement, l'algorithme arrête si le nombre maximum d'itérations est atteint ou si le syndrome *S* est nul.

II.7.3 Algorithmes de décodage des codes LDPC non-binaires

L'algorithme de décodage pour les codes **LDPC** binaires peut être prolongé pour décoder les codes **LDPC** non-binaires, définis dans un corps de Galois $GF(2^p)$.

Les éléments $\mu_{mn}(x)$, $\beta_{mn}(x)$ et $\mu_n(x)$ des matrices respectivement *V* (équation (II.26)), *B* (équation (II.28)) et Λ (équation (II.35)) sont de longueur 2^p :

$$\mu_{mn}(x)=\begin{pmatrix}\mu_{mn}(0)\\ \mu_{mn}(1)\\ \vdots\\ \mu_{mn}(2^p-1)\end{pmatrix},\beta_{mn}(x)=\begin{pmatrix}\beta_{mn}(0)\\ \beta_{mn}(1)\\ \vdots\\ \beta_{mn}(2^p-1)\end{pmatrix},\mu_n(x)=\begin{pmatrix}\mu_n(0)\\ \mu_n(1)\\ \vdots\\ \mu_n(2^p-1)\end{pmatrix} \quad \text{(II.53)}$$

Dans l'initialisation $\mu_{mn}(x)$ est égale à :

$$\mu_{mn}(x)=\begin{pmatrix}Pr(v_n=0|C_m)\\ Pr(v_n=1|C_m)\\ \vdots\\ Pr(v_n=2^p-1|C_m)\end{pmatrix} \quad \text{(II.54)}$$

Avec $\{0,1,\dots,2^p-1\}$ sont les éléments de corps $GF(2^p)$. Pour un corps de Galois $GF(4)$ les éléments décimales de ce corps sont $\{0,1,2,3\}$, où 0, 1, 2 et 3 dans $GF(4)$ corresponds respectivement à 00, 01, 10 et 11 dans $GF(2)$. Donc :

$$\mu_{mn}(x)=\begin{pmatrix}Pr(v_n=0|C_m)\times Pr(v_n=0|C_m)\\ Pr(v_n=0|C_m)\times Pr(v_n=1|C_m)\\ Pr(v_n=1|C_m)\times Pr(v_n=0|C_m)\\ Pr(v_n=1|C_m)\times Pr(v_n=1|C_m)\end{pmatrix} \quad \text{(II.55)}$$

Afin de simplifier le calcul des nœuds de parité, une permutation des probabilités et une dépermuation sont réalisées respectivement avant et après le calcul des nœuds de parité.

- **Nœuds de permutation**

En générale, les équations de contrôle de parité sont de la forme :

$$C_m=\sum_{n=1}^{N}h_{mn}\,c_n,\quad m\in\{1,\dots,M\} \quad \text{(II.56)}$$

Avec h_{mn} et $c_n\in GF(2^p)$, $i\in\{1,\dots,M\}$, $j\in\{1,\dots,N\}$. L'équation de contrôle parité C_i est satisfait quand :

$$h_{i1}c_1+h_{i2}c_2+\cdots+h_{iN}c_N=0 \quad \text{(II.57)}$$

Dans ce cas le calcul des nœuds de parité est plus compliqué que le cas binaire puisque nous devons maintenant considérer les éléments non-binaires de la matrice de contrôle de parité *H*.

Les éléments non-nuls h_{mn} de la matrice H sont les éléments $\{0, \alpha^0, \alpha^1, \dots, \alpha^{2^p-2}\}$ d'un corps de Galois $GF(2^p)$, ces élément correspondent aux nombres décimales $\{0, 1, 2, \dots, 2^p - 1\}$.

Quand le symbole codé c_n est multiplié par un élément non-binaire de contrôle de parité h_{mn} , nous pouvons compenser ceci en décalant cycliquement en bas les probabilités du vecteur $\mu_{mn}(x)$, excepté la première probabilité, correspondant à la probabilité du symbole codé étant zéro.

Le nombre de décalages circulaires est égal à la puissance de l'élément primitif α qui est multiplié avec le symbole codé [40].

$$nombre\ de\ décalges = \begin{cases} 0 & si\ h_{mn} = \alpha^0 \\ 1 & si\ h_{mn} = \alpha^1 \\ \vdots & \vdots \\ 2^p - 1 & si\ h_{mn} = \alpha^{2^p-1} \end{cases} \tag{II.58}$$

Selon les éléments non-nuls de la matrice de contrôle de parité H, les décalages sont illustrés comme suit :

$$\alpha^0 v_j \Rightarrow \mu_{mn}(x) = \begin{pmatrix} \mu_{mn}(0) \\ \mu_{mn}(1) \\ \mu_{mn}(2) \\ \vdots \\ \mu_{mn}(2^p - 1) \end{pmatrix}, \alpha^1 v_j \Rightarrow \mu_{mn}(x) = \begin{pmatrix} \mu_{mn}(0) \\ \mu_{mn}(2^p - 1) \\ \mu_{mn}(1) \\ \vdots \\ \mu_{mn}(2^p - 2) \end{pmatrix}, \alpha^2 v_j \Rightarrow \mu_{mn}(x) = \begin{pmatrix} \mu_{mn}(0) \\ \mu_{mn}(2^p - 2) \\ \mu_{mn}(12^p - 1) \\ \vdots \\ \mu_{mn}(2^p - 3) \end{pmatrix}, \dots.$$

Ce décalage circulaire des probabilités, ou permutation, transforme l'équation de contrôle de parité de l'équation (II.57) à :

$$c_1 + c_2 + \cdots + c_N = 0 \tag{II.59}$$

Cette équation qui est plus similaire aux équations de contrôle de parité binaires. L'inverse d'une permutation, ou un dépermutation, où les probabilités sont cycliquement décalés vers le haut, encore excepté la première probabilité.

Dans la section suivante, on présente l'algorithme **FFT-SP** qui peut être utiliser pour les codes **LDPC** binaires et non-binaires. Puis, on présente l'algorithme **EMS**. Ces deux algorithmes sont utilisés dans notre travail.

II.7.3.1 Algorithme FFT-SP

Dans cet algorithme, on utilise la transformé de Fourier rapide (**FFT**) afin de réduire la complexité de calcul des nœuds de parité [41]. L'algorithme **FFT-SP** est le suivant [40]:

▪ Initialisation des nœuds de variable

$$\gamma_n(x) = Pr(v_n = x|c_n^{'}), \qquad x \in GF(2^p),\ m\epsilon\{1,\dots,M\} \text{ et } n\epsilon\{1,\dots,N\} \tag{II.60}$$

Dans le cas binaire $p = 1$.

$$\mu_{mn}(x) = \gamma_n(x) \tag{II.61}$$

- Itération

▪ Calcul des nœuds de parité

$$\beta_{mn}(h_{mn} \otimes c_n) = FFT^{-1}\left(\prod_{n' \in N_m/n} FFT\left(\mu_{mn'}(h_{mn'} \otimes c_{n'})\right)\right) \tag{II.62}$$

Avec $\mu_{mn'}(h_{mn} \otimes c_n)$ signifie la permutation des probabilités $\mu_{mn'}$, $\beta_{mn'}(h_{mn'} \otimes c_{n'})$ signifie la dépermutation des probabilités β_{mn}.

Dans le cas binaire, on ne fait pas les deux opérations : permutation et dépermutation.

▪ Calcul des nœuds de variable

$$\mu_{mn}(x) = \delta_{mn}\gamma_n(x)\prod_{m' \in M_n/m}\beta_{m'n}(x), \text{ avec } \delta_{m,n} = \frac{1}{\sum_a \gamma_n \prod_{m' \in M_n/m}\beta_{m'n}(x)} \tag{II.63}$$

▪ Information à posteriori

$$\tilde{\gamma}_n(x) = \delta_n\gamma_n(x)\prod_{m' \in M_n}\beta_{m'n}(x) \tag{II.64}$$

▪ Décision

$$\hat{c}_n = \arg\max_x\left(\tilde{\gamma}_n(x)\right) \tag{II.65}$$

Finalement, l'algorithme s'arrête si le nombre maximum d'itérations est atteint ou si le syndrome *S* est nul.

II.7.3.2 Algorithme EMS

L'algorithme **EMS** exécute les opérations suivantes [39]:

▪ Initialisation

$$\gamma_n(x) = log\frac{Pr(v_n=x|c_n^{'})}{Pr(v_n=0|c_n^{'})}, \qquad x \in GF(2^p),\ m\epsilon\{1,\dots,M\} \text{ et } n\epsilon\{1,\dots,N\} \tag{II.66}$$

$$\mu_{mn}(x) = \gamma_n(x) \tag{II.67}$$

- Itération

▪ Traitement des nœuds de parité

$$\beta_{mn}(h_{mn}\otimes c_n) = \min_{(x_n)_{n'\in N_m/n}}\left(\sum_{n'\in N_m/n}\mu_{mn'}(h_{mn'}\otimes c_{n'})\right) \tag{II.68}$$

▪ Traitement des nœuds de variable

$$\mu_{mn}^{'}(x) = \gamma_n(x) + \sum_{m'\in M_n/m}\beta_{m'n}(x) \tag{II.69}$$

$$\mu_{mn}(x) = \mu_{mn}^{'}(x) - \mu_{mn}^{'}(0) \tag{II.70}$$

▪ Information a posteriori

$$\tilde{\gamma}(x) = \gamma(x) + \sum_{m\in M_n}\beta_{mn}(x) \tag{II.71}$$

▪ Décision

$$\hat{c}_n = \arg\max_x(\tilde{\gamma}_n) \tag{II.72}$$

Finalement, l'algorithme arrête si le nombre maximum d'itérations est atteint ou si le syndrome *S* est nul.

II.8 Conclusion

Ce chapitre a été consacré aux code **LDPC** binaires et non-binaires. On a étudié leur principe de codage et on a montré que le codage se fait à l'aide de leur matrice de contrôle de parité. Ainsi, on a illustré leur algorithme de décodage à décision pondérée, l'algorithme **SP**, où a partir de cet algorithme plusieurs algorithmes ont été dérivés.

On a aussi montré que les messages échangés dans le décodeur peuvent être calculer à l'aide du **LLR** ou de l'**APP** selon le type d'algorithme de décodage utilisé. Par conséquent, les messages à l'entrée du décodeur doivent être calculer selon le type d'algorithme utilisé.

Le calcul des messages d'entrées, se fait à la sortie du canal, dépend de la constellation utilisée et de type de canal considérée. Cependant, le nombre d'opérations effectuées pour faire ce calcul augment avec l'ordre de la constellation. Ainsi, le calcul se changent en fonction de type de canal.

Le chapitre suivant est consacré à la simplification de ce calcul pour les codes **LDPC** non-binaires.

Chapitre III

*Calcul Simplifié de l'**APP** et du **LLR** pour un Code **LDPC** Non-Binaire*

III.1 Introduction

Les constellations d'ordre élevé peuvent obtenir une transmission à haut débit améliorée sans augmenter la bande passante [44]. Pour cette raison, la modulation d'amplitude en quadrature (**MAQ**), qui a été adoptée par diverses normes de communication [45-48], est fortement recommandée comme une constellation d'ordre élevé. Cependant, les systèmes de communication utilisant la **MAQ** nécessitent un fort rapport signal à bruit. Afin de remédier à cet inconvénient, il est intéressant de combiner avec la **MAQ** des codes correcteurs d'erreurs performants tels que les codes **LDPC** non-binaires.

Etant donné que le décodeur doit fonctionner en décisions douces calculées à l'aide du **LLR** ou de l'**APP**. Le calcul exact de ces décisions ont des problèmes d'opérations compliquées. Plusieurs algorithmes ont été introduits pour simplifier le calcul exact du **LLR** pour les codes binaires.

Dans ce chapitre, nous adaptons ces simplifications sur les codes **LDPC** non-binaires. Ainsi, nous proposons une méthode pour rendre facile le calcul exact de l'**APP**. Elle est programmée afin d'adapter le plus parfaitement possible le système de transmission au type de canal considéré. Cette méthode conduit à simplifier la mise en œuvre du système.

Au début de ce chapitre, le système combinant un code **LDPC** et une **MAQ** est présenté. Après, une description de la méthode introduite. Ensuite, la simplification du calcul exact de l'**APP** est examinée, et ce, pour un canal gaussien et un canal de Rayleigh. Puis, les calculs simplifiés du **LLR** et de l'**APP** pour les codes **LDPC** non-binaires sont discutées. Enfin, on présente les différents résultats de simulation de l'influence de la simplification du calcul de l'**APP** et du **LLR** sur les performances d'un code **LDPC** non-binaire associé à une **MAQ**.

III.2 Mise en œuvre d'un système combinant une modulation et un code LDPC

Les modulations à *M* états ($M = 2^m$) utilisent un ensemble de *M* signaux de durée *T* pour transmettre *m* symboles $\{u_{n,i}\}, i \in \{1, \dots, m\}$, toutes les *T* secondes. D'un point de vue théorique, l'opération de modulation à l'instant *nT* consiste à représenter, dans un espace à deux dimensions, les *M* signaux par un ensemble de *M* points appelé constellation, et de faire correspondre l'ensemble de *m* symboles à chaque point de la constellation repéré par son abscisse a_n et son ordonnée b_n. La constellation permet de différencier chaque type de modulation.

A titre d'exemple, considérons le cas de la modulation **MAQ**-16 (*M*=16, *m*=4) dont la constellation ainsi que le codage de Gray associé sont représentés sur la figure III.1.

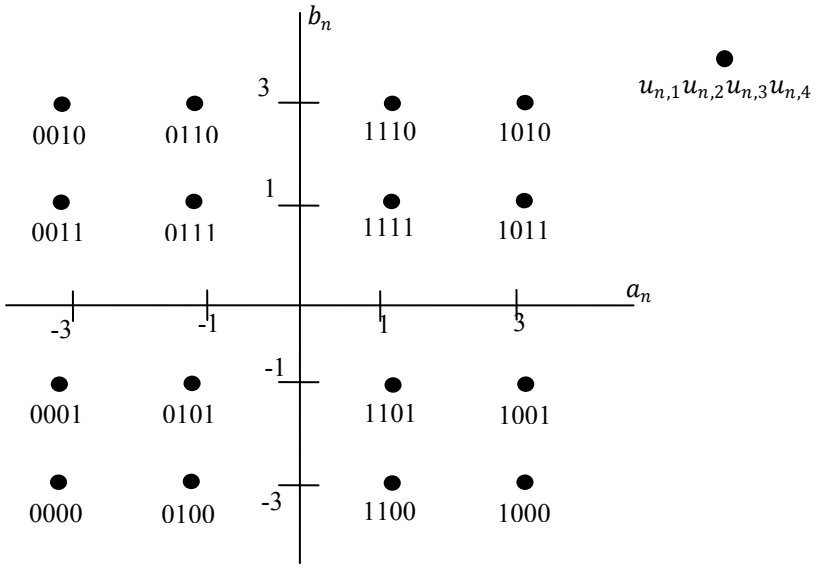


Figure III.1- Constellation d'une **MAQ**-16 avec codage de Gray

Pour la **MAQ**: $a_n, b_n \in [\pm 1, \pm 3, \pm 5, \dots, (m-1)]$;

Donc chaque ensemble de *m* symboles binaires est associé à chaque instant *nT* à un couple de symboles (a_n, b_n). Après le passage dans le canal de transmission, l'observation relative au couple (a_n, b_n) est représentée par un couple (a_n', b_n'). Les symboles transmis vont

mieux suivre un code binaire de type Gray, il permet d'affirmer qu'il existe généralement un seul symbole erroné [49].

Le schéma le plus simple d'un système de transmission numérique dans le cadre de l'association d'un code **LDPC** et d'une constellation à *M* états, en particulier **MAQ**-*M*, où *M*= 2^m états, est donné à la figure III.2.

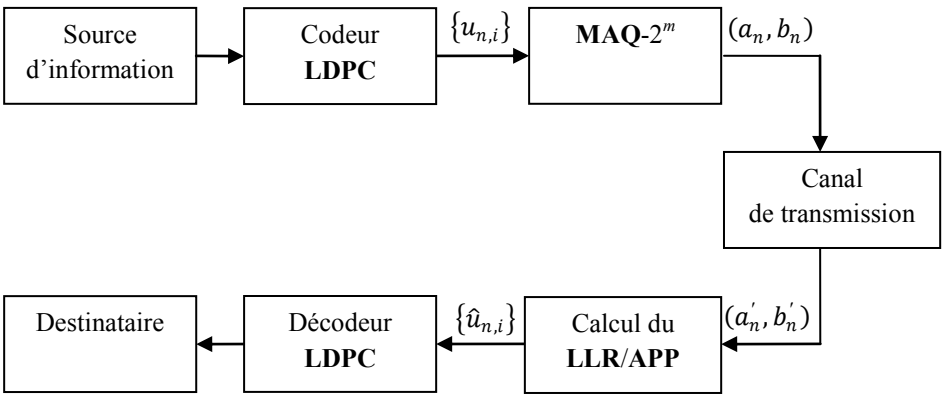


Figure III.2- Schéma d'un système de transmission numérique dans le cadre de l'association d'un code **LDPC** et une **MAQ**-2^m

A la réception, on traite les couples (a'_n, b'_n) représentatifs du couple (a_n, b_n) afin d'extraire *m* échantillons $\{\hat{u}_{n,i}\}, i \in \{1, \dots, m\}$, chacun représentatif d'un symbole binaire $u_{n,i}$ associé au même signal lors de la modulation.

Quelle que soit la constellation à 2^m états employée, pour chaque couple (a'_n, b'_n) reçu à l'instant *nT*, l'échantillon $\hat{u}_{n,i}$ est obtenu à l'aide de deux relations: $\boldsymbol{LLR}(u_{n,i})$ (Log-Likelihood Ratio) ou $\boldsymbol{APP}(u_{n,i})$ (A Posteriori Probability):

- $\boldsymbol{APP}(u_{n,i})$ est calculée comme suit [11]:

$$\boldsymbol{APP}(u_{n,i} = 0) = Pr\{u_{n,i} = 0/(a'_n, b'_n)\}, \qquad i \in \{1, \dots, m\} \qquad \text{(III.1.a)}$$

$$\boldsymbol{APP}(u_{n,i} = 1) = 1 - \boldsymbol{APP}(u_{n,i} = 0), \qquad i \in \{1, \dots, m\} \qquad \text{(III.1.b)}$$

Où $Pr\{u_{n,i} = w/(a_n^{'}, b_n^{'})\}$ représente la probabilité pour que le symbole $u_{n,i}$ émis possède la valeur w (w = 0 ou 1) sachant que le couple disponible en sortie du canal est $(a_n^{'}, b_n^{'})$.

En utilisant la règle de Bayes, l'expression (III.1) devient [11]:

$$\boldsymbol{APP}(u_{n,i} = 0) = \frac{Pr\{(a_n^{'},b_n^{'})\,/u_{n,i}=0\}}{Pr\{(a_n^{'},b_n^{'})\,/u_{n,i}=0\}+Pr\{(a_n^{'},b_n^{'})\,/u_{n,i}=1\}}, \qquad i \in \{1,\dots,m\} \qquad \text{(III.2.a)}$$

$$\boldsymbol{APP}(u_{n,i} = 1) = 1 - \boldsymbol{APP}(u_{n,i} = 0), \qquad i \in \{1,\dots,m\} \qquad \text{(III.2.b)}$$

Où $Pr\{(a_n^{'}, b_n^{'})\,/u_{n,i} = w\}$ désigne à présent la probabilité pour que le couple disponible soit $(a_n^{'}, b_n^{'})$ sachant que le symbole binaire $u_{n,i}$ est égal à w.

- $\boldsymbol{LLR}(u_{n,i})$ est calculé comme suit [50]:

$$\boldsymbol{LLR}(u_{n,i}) = log\left[\frac{Pr\{u_{n,i}=1/(a_n^{'},b_n^{'})\}}{Pr\{u_{n,i}=0/(a_n^{'},b_n^{'})\}}\right], \qquad i \in \{1,\dots,m\} \qquad \text{(III.3)}$$

En utilisant deux fois de suite la règle de Bayes, l'expression (III.3) devient [33]:

$$\boldsymbol{LLR}(u_{n,i}) = log\left[\frac{Pr\{(a_n^{'},b_n^{'})\,/u_{n,i}=1\}}{Pr\{(a_n^{'},b_n^{'})\,/u_{n,i}=0\}}\right], \qquad i \in \{1,\dots,m\} \qquad \text{(III.4)}$$

Equation (III.4) est le calcul exact du **LLR** qui représente l'algorithme log-**MAP** (Maximum A Posteriori) [51, 52, 53]. Cependant, il implique des opérations compliquées. Plusieurs algorithmes ont été introduit afin de simplifier le calcul exact du **LLR**. L'algorithme pragmatique (section III.4.2.1.b), introduit dans [54, 55], tente de simplifier le calcul en supposant que les valeurs de vraisemblance sont des variables gaussiennes. L'algorithme max-log-**MAP** (section III.4.2.1.a) est la simplification la plus populaire de l'algorithme log-**MAP** [56].

Cependant, les simplifications du **LLR** sont utilisées que pour des décodeurs basés sur **LLR**. Tandis que pour les décodeurs basés sur **APP**, nous devons calculer l'**APP**, et ce dernier sont aussi complexe. De ce fait, nous proposons une méthode permettant d'appliquer les simplifications du **LLR** et de les adapter au décodeur **LDPC** basé sur l'**APP**, et de même permettant de simplifier le calcul de l'**APP**. Ceci à condition d'insérer un module

supplémentaire permettant de faire la conversion du **LLR** à l'**APP**, comme le montre la figure III.3.

L'algorithme peu complexe de l'**APP** obtenu sur le canal gaussien peut être réutilisé efficacement sur un canal de Rayleigh (figure III.3), ceci à condition d'insérer un module supplémentaire permettant de tenir compte, à chaque instant nT, de l'atténuation α_n du canal.

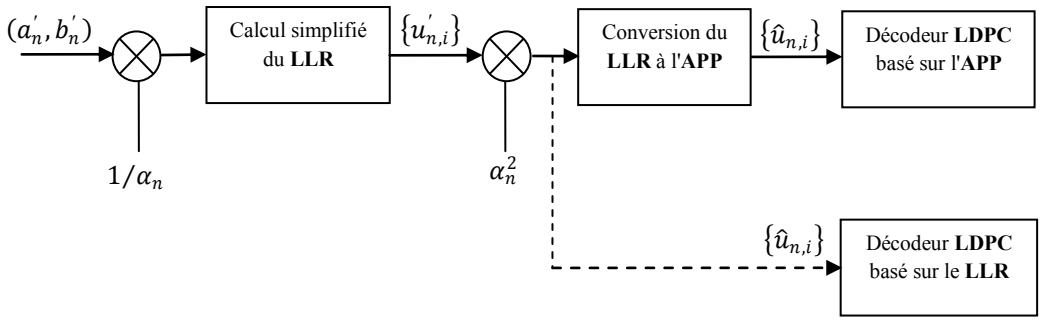


Figure III.3-Principe de fonctionnement du calcul simplifié de l'**APP** permettant d'adapter le système destiné au canal gaussien à un canal de Rayleigh

Le bloc du calcul simplifié du **LLR** est identique pour l'ensemble des algorithmes de décodage à décision pondérée, quelque soit l'entrée du décodeur: **LLR** ou **APP**. Seule le module de la conversion du **LLR** à l'**APP** des messages est dépend de l'entrée du décodeur. Sa présence sur la figure III.3, se justifie par le besoin d'appliquer le calcul simplifié du **LLR** au décodeur **LDPC** basé sur **APP**. Leur présence n'est pas indispensable dans le cas d'un décodeur **LDPC** basé sur **LLR**.

En effet, il est aisé de modifier un algorithme de décodage basé sur **LLR** à un algorithme basé sur **APP**, tout en conservant inchangés le calcul simplifié du **LLR**.

La figure (III.3) montre bien que la méthode proposé permet de simplifier la mise on œuvre du système. Dans ce qui suit, on va montrer que cette méthode permet de simplifier le calcul exact de l'**APP**. On va expliquer la méthode pour une **MAQ** à 2^{2p} états, avec $p \in \mathbb{N}$.

III.3 Calcul exact de l'APP

Une modulation **MAQ** à 2^{2p} états conventionnelle utilise une constellation carrée (cas des **MAQ**-16, **MAQ**-64, et **MAQ**-256). Une telle modulation présente la particularité de

pouvoir se résumer à deux modulations d'amplitude à 2^P états (**MDA-**2^P) indépendantes agissant sur deux porteuses en phase et en quadrature [57].

D'après la propriété précédente (le cas d'une constellation carrée), les p expressions relatives à la voie en phase, obtenues à partir de la relation (III.2), sont par conséquent les suivantes:

$$\boldsymbol{APP}\left(u_{n,i}=0\right)=\frac{Pr\{a_n^{'}/u_{n,i}=0\}}{Pr\{a_n^{'}/u_{n,i}=0\}+Pr\{a_n^{'}/u_{n,i}=1\}},\qquad i\in\{1,\dots,p\}\qquad \text{(III.5.a)}$$

$$\boldsymbol{APP}\left(u_{n,i}=1\right)=1-\boldsymbol{APP}\left(u_{n,i}=0\right),\qquad i\in\{1,\dots,p\}\qquad \text{(III.5.b)}$$

Et les p expressions relatives à la voie en quadrature, obtenues à partir de la relation (III.2), sont les suivantes:

$$\boldsymbol{APP}\left(u_{n,i}=0\right)=\frac{Pr\{b_n^{'}/u_{n,i}=0\}}{Pr\{b_n^{'}/u_{n,i}=0\}+Pr\{b_n^{'}/u_{n,i}=1\}},\qquad i\in\{p+1,\dots,2p\}\qquad \text{(III.6.a)}$$

$$\boldsymbol{APP}\left(u_{n,i}=1\right)=1-\boldsymbol{APP}\left(u_{n,i}=0\right),\qquad i\in\{p+1,\dots,2p\}\qquad \text{(III.6.b)}$$

Pour un canal de transmission gaussien, les p relations relatives à la voie en phase aboutissent finalement aux expressions suivantes:

$$\boldsymbol{APP}\left(u_{n,i}=0\right)=\frac{\sum_{j=1}^{2^{p-1}}exp\left\{-\frac{1}{2\sigma^2}\left(a_n^{'}-a_{i,j}^{0}\right)^2\right\}}{\sum_{j=1}^{2^{p-1}}exp\left\{-\frac{1}{2\sigma^2}\left(a_n^{'}-a_{i,j}^{0}\right)^2\right\}+\sum_{j=1}^{2^{p-1}}exp\left\{-\frac{1}{2\sigma^2}\left(a_n^{'}-a_{i,j}^{1}\right)^2\right\}},i\in\{1,\dots,p\}\qquad \text{(III.7.a)}$$

$$\boldsymbol{APP}\left(u_{n,i}=1\right)=1-\boldsymbol{APP}\left(u_{n,i}=0\right),\qquad i\in\{1,\dots,p\}\qquad \text{(III.7.b)}$$

Avec:

- $a_n^{'}$: échantillon représentatif du symbole a_n émis, disponible en sortie du canal ;
- $a_{i,j}^{k}$: valeurs que peut prendre le symbole a_n lorsque le symbole binaire $u_{n,i}$ à transmettre possède la valeur k (k = 0 ou 1) ;
- σ^2 : variance du bruit présent en sortie du canal.

De la même façon, pour un canal gaussien, les p relations relatives à la voie en quadrature aboutissent finalement aux expressions suivantes:

$$\boldsymbol{APP}(u_{n,i}=0)=\frac{\sum_{j=1}^{2^{p-1}} exp\left\{-\frac{1}{2\sigma^2}\left(b_n^{'}-b_{i,j}^{0}\right)^2\right\}}{\sum_{j=1}^{2^{p-1}} exp\left\{-\frac{1}{2\sigma^2}\left(b_n^{'}-b_{i,j}^{0}\right)^2\right\}+\sum_{j=1}^{2^{p-1}} exp\left\{-\frac{1}{2\sigma^2}\left(b_n^{'}-b_{i,j}^{1}\right)^2\right\}}, i\in\{p+1,\dots,2p\} \quad \text{(III.8.a)}$$

$$\boldsymbol{APP}(u_{n,i}=1)=1-\boldsymbol{APP}(u_{n,i}=0), \qquad i\in\{p+1,\dots,2p\} \quad \text{(III.8.b)}$$

Avec :

- $b_n^{'}$: échantillon représentatif du symbole b_n émis, disponible en sortie du canal ;
- $b_{i,j}^{k}$: valeurs que peut prendre le symbole b_n lorsque le symbole binaire $u_{n,i}$ à transmettre possède la valeur k (k = 0 ou 1) ;

Les relations (III.7) et (III.8), représentant le calcul exact de l'**APP**, reflète une information complète pour tous les symboles **MAQ** possibles. Par conséquent, il est nécessaire un grand nombre de calculs pour calculer exactement l'**APP** dans le cas d'une constellation d'ordre élevé, comme **MAQ-**16, **MAQ-**64, **MAQ-**256.

Par la suite, on va simplifier ce calcul, en utilisant la méthode proposé, lorsque le canal est gaussien. Ensuite, on va montrer que les équations peu complexes obtenues sur un canal gaussien peuvent être réutilisés efficacement sur un canal de Rayleigh, ceci à condition d'insérer un module supplémentaire permettant de tenir compte, à chaque nT, de l'atténuation α_n du canal.

III.4 Calcul simplifié de l'APP

III.4.1 Cas du canal de Gauss

En suivant la figure (III.3), la simplification des relations (III.7) et (III.8) est obtenu après le calcul simplifié du **LLR** puis la conversion du **LLR** à l'**APP**.

III.4.1.1 Calcul simplifié du LLR

Comme le canal de transmission est gaussien, et la modulation utilise est une constellation carrée, les expressions obtenues d'après l'équation (III.4) sont bien approchées, à l'aide deux algorithmes: max-log-**MAP** et pragmatique.

III.4.1.1.a Algorithme max-log-MAP

L'algorithme Max-log-**MAP**, introduit dans [56], démontre que les *p* relations relatives à la voie en phase, sont données par:

$$\boldsymbol{LLR}(u_{n,i}) = \frac{\left(\min_{j\in\{1,\dots,2^{p-1}\}}\left(a_n^{'}-a_{i,j}^{0}\right)\right)^2-\left(\min_{j\in\{1,\dots,2^{p-1}\}}\left(a_n^{'}-a_{i,j}^{1}\right)\right)^2}{2\sigma^2}, i\in\{1,\dots,p\} \qquad \text{(III.9.a)}$$

Et les *p* relations relatives à la voie en quadrature, sont données par:

$$\boldsymbol{LLR}(u_{n,i}) = \frac{\left(\min_{j\in\{1,\dots,2^{p-1}\}}\left(b_n^{'}-b_{i,j}^{0}\right)\right)^2-\left(\min_{j\in\{1,\dots,2^{p-1}\}}\left(b-b_{i,j}^{1}\right)\right)^2}{2\sigma^2}, i\in\{p+1,\dots,2\,p\} \qquad \text{(III.9.b)}$$

III.4.1.1.b Algorithme pragmatique

L'algorithme pragmatique, introduit dans [54], démontre que les *p* relations relatives à la voie en phase, sont données par:

$$\begin{aligned}
&\boldsymbol{LLR}(u_{n,1}) = a_n^{'}\\
&\boldsymbol{LLR}(u_{n,2}) = -\left|\boldsymbol{LLR}(u_{n,1})\right| + 2^{p-1}\\
&\vdots\\
&\boldsymbol{LLR}(u_{n,i}) = -\left|\boldsymbol{LLR}(u_{n,i-1})\right| + 2^{p-i+1} \qquad \text{(III.10.a)}\\
&\vdots\\
&\boldsymbol{LLR}(u_{n,p}) = -\left|\boldsymbol{LLR}(u_{n,p-1})\right| + 2
\end{aligned}$$

Et les *p* relations relatives à la voie en quadrature, sont données par:

$$\begin{aligned}
&\boldsymbol{LLR}(u_{n,p+1}) = b_n^{'}\\
&\boldsymbol{LLR}(u_{n,p+2}) = -\left|\boldsymbol{LLR}(u_{n,p+1})\right| + 2^{p-1}\\
&\vdots\\
&\boldsymbol{LLR}(u_{n,p+i}) = -\left|\boldsymbol{LLR}(u_{n,p+i-1})\right| + 2^{p-i+1} \qquad \text{(III.10.b)}\\
&\vdots\\
&\boldsymbol{LLR}(u_{n,2p}) = -\left|\boldsymbol{LLR}(u_{n,2p-1})\right| + 2
\end{aligned}$$

Pour une bonne approximation, on multiplie les relations (III.10) par un facteur constant $f = 2/\sigma^2$, on obtient:

$$\begin{aligned}
&\boldsymbol{LLR}(u_{n,1}) = f \times a_n^{'} \\
&\boldsymbol{LLR}(u_{n,2}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,1})\right| + 2^{p-1}\right) \\
&\vdots \\
&\boldsymbol{LLR}(u_{n,i}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,i-1})\right| + 2^{p-i+1}\right) \\
&\vdots \\
&\boldsymbol{LLR}(u_{n,p}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,p-1})\right| + 2\right)
\end{aligned} \qquad \text{(III.11.a)}$$

Et

$$\begin{aligned}
&\boldsymbol{LLR}(u_{n,p+1}) = f \times b_n^{'} \\
&\boldsymbol{LLR}(u_{n,p+2}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,p+1})\right| + 2^{p-1}\right) \\
&\vdots \\
&\boldsymbol{LLR}(u_{n,p+i}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,p+i-1})\right| + 2^{p-i+1}\right) \\
&\vdots \\
&\boldsymbol{LLR}(u_{n,2p}) = f \times \left(-\left|\boldsymbol{LLR}(u_{n,2p-1})\right| + 2\right)
\end{aligned} \qquad \text{(III.11.b)}$$

III.4.1.2 Conversion du LLR à l'APP

La dérivation de l'**APP** a partir du **LLR** simplifié [58], conduit aux relations simplifiées suivantes:

$$\boldsymbol{APP}(u_{n,i} = 0) = \frac{1}{1+exp\left(\boldsymbol{LLR}(u_{n,i})\right)}, \qquad i \in \{1, \dots, 2p\} \qquad \text{(III.12.a)}$$

$$\boldsymbol{APP}(u_{n,i} = 1) = 1 - \boldsymbol{APP}(u_{n,i} = 0), \qquad i \in \{1, \dots, 2p\} \qquad \text{(III.12.b)}$$

Ainsi, il est bien remarquable que le calcul simplifié de l'**APP**, les relations (III.12), sont moins nombre de calculs que le calcul exact de l'**APP**, les relations (III.7) et (III.8) .

A titre d'illustration, considérons par exemple le cas de la modulation **MAQ**-16 (p=2), les fonctions reliant les sorties $\hat{u}_{n,i}$ $\left(i \in \{1, 2, 3, 4\}\right)$ aux entrées $a_n^{'}$ et $b_n^{'}$, en utilisant les deux algorithmes pragmatique et max-log-**MAP**, sont indiquées sur les figures III.4.

Sur chacune des figures III.4, on a représenté la fonction dont la forme est donnée par les relations (III.7) et (III.8), ainsi que la fonction simplifiée correspondante issue des relations (III.12), ceci pour un rapport signal à bruit égal à 4 dB.

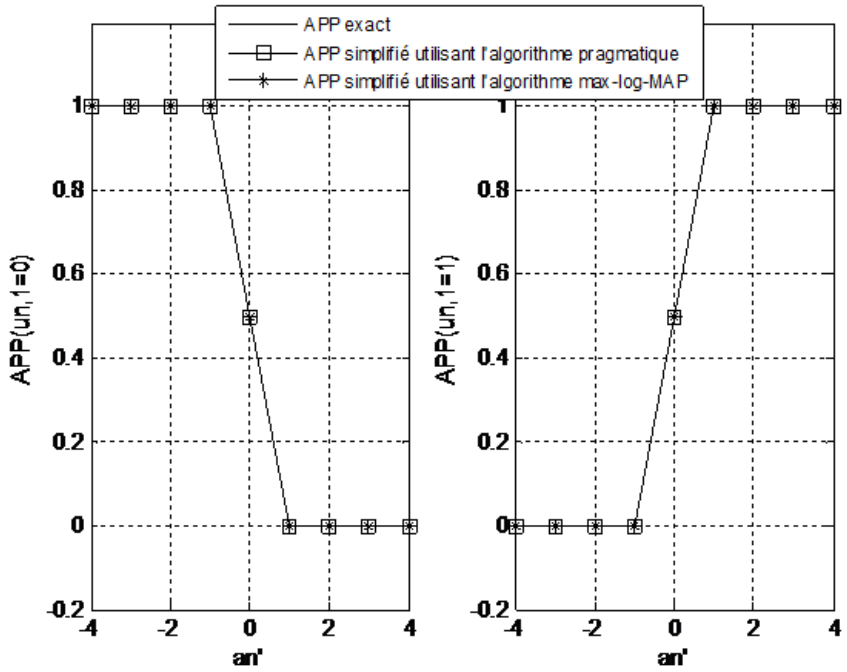


Figure III.4.a- Courbes permettant d'obtenir l'échantillon $\boldsymbol{APP}(u_{n,1})$ en fonction de l'échantillon $a_n^{'}$, pour un $E_b/N_0 = 4\ dB$

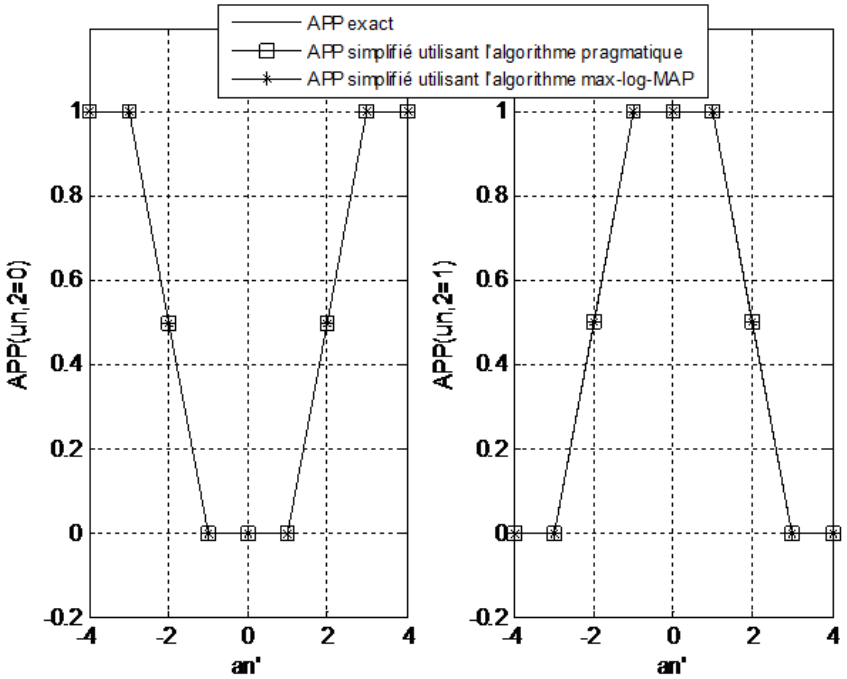


Figure III.4.b- Courbes permettant d'obtenir l'échantillon $\boldsymbol{APP}(u_{n,2})$ en fonction de l'échantillon $a_n^{'}$, pour un $E_b/N_0 = 4\ dB$

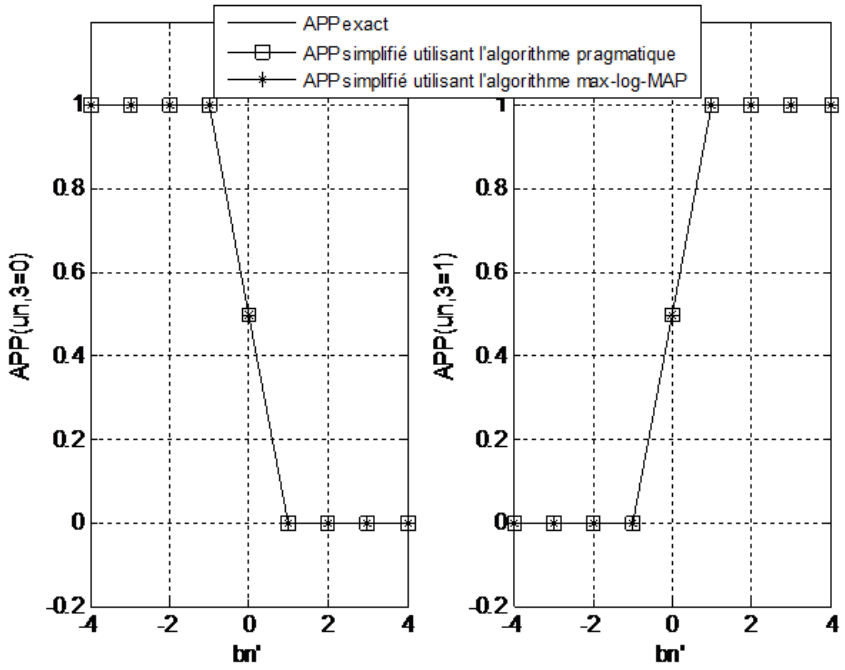


Figure III.4.c- Courbes permettant d'obtenir l'échantillon $\boldsymbol{APP}(u_{n,3})$ en fonction de l'échantillon $b_n^{'}$, pour un $E_b/N_0 = 4\ dB$

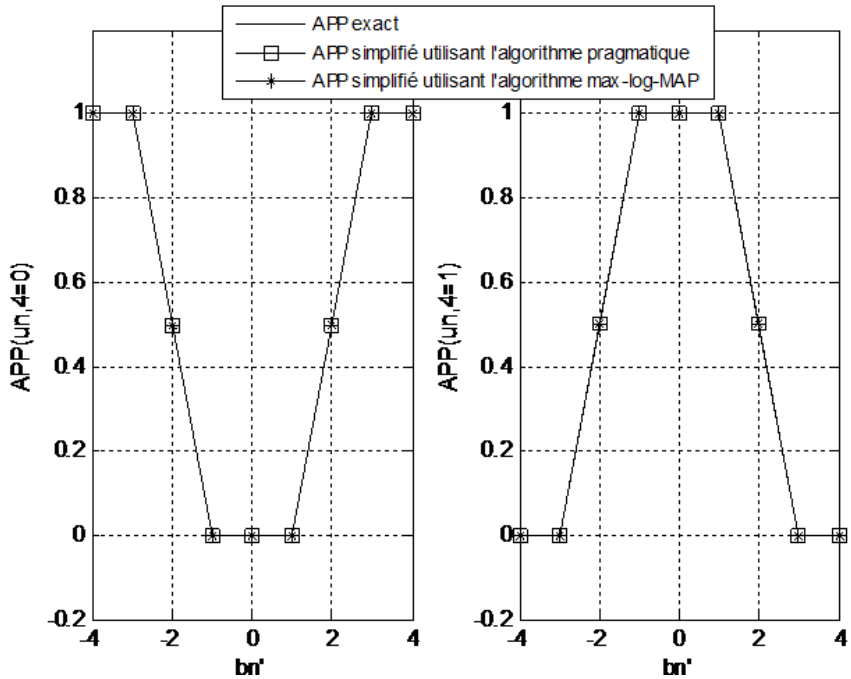


Figure III.4.d- Courbes permettant d'obtenir l'échantillon $\boldsymbol{APP}(u_{n,4})$ en fonction de l'échantillon $b_n^{'}$, pour un $E_b/N_0 = 4\ dB$

Dans tous les cas, ces courbes permettent de vérifier que, malgré leur grande simplicité, les approximations réalisées par les relations (III.12) sont excellentes, et qu'elles peuvent donc être utilisées avantageusement en remplaçant les expressions (III.7) et (III.8).

III.4.2 Cas du canal de Rayleigh

Dans cette partie, on fait l'hypothèse que le canal de transmission est un canal de Rayleigh. De plus, on suppose que l'atténuation α_n du canal à l'instant nT est parfaitement connue par le récepteur [59].

On va montrer qu'il n'est pas nécessaire de recommencer tout le raisonnement effectué à propos du canal gaussien pour obtenir les algorithmes permettant de mettre en œuvre les simplifications réalisées sur un canal de Rayleigh [60].

En premier lieu, rappelons qu'en sortie du canal à l'instant nT, l'information disponible relative au couple (a_n, b_n) est telle que :

$$a_n^{'} = \alpha_n a_n + z_n \tag{III.13}$$

$$b_n^{'} = \alpha_n b_n + z_n \tag{III.14}$$

Où z_n est un bruit gaussien, centré, de variance σ^2 et la variable α_n caractérise l'atténuation du signal émis.

Comme la variable α_n à l'instant nT est connue, il est possible de diviser les deux échantillons $a_n^{'}$ et $b_n^{'}$ disponibles en sortie du canal par α_n [55]. Les échantillons $a_n^{''}$ et $b_n^{''}$ ainsi obtenus s'expriment sous la forme :

$$a_n^{''} = \frac{a_n^{'}}{\alpha_n} = a_n + z_n^{'} \tag{III.15}$$

$$b_n^{''} = \frac{b_n^{'}}{\alpha_n} = b_n + z_n^{'} \tag{III.16}$$

Où $z_n^{'}$ est un bruit gaussien, centré, de variance σ_n^2 égale à σ^2/α_n^2 .

Etant donné que les échantillons $a_n^{''}$ et $b_n^{''}$ sont modélisés par des variable gaussiennes, il est possible d'appliquer, directement sur les échantillons $a_n^{''}$ et $b_n^{''}$ disponibles, des

algorithmes simplifiés du **LLR** strictement identiques à ceux utilisés quand le canal de transmission est gaussien, et ceci quelque soit la modulation à 2^m états employée.

Afin d'appliquer au mieux ces résultats obtenus avec une modulation au système proposé dans ce chapitre, il apparaît donc nécessaire de multiplier par α_n^2 les échantillons $u'_{n,i} = \boldsymbol{LLR}(u_{n,i})$, avant de faire la conversion (figure (III.3)).

Dans la section suivante, nous montrons que la simplification de l'**APP** introduit précédemment est utilisée facilement pour les codes **LDPC** non-binaires. Ainsi, nous montrons qu'on peut adapter le calcul simplifié du **LLR** sur les codes **LDPC** non-binaires.

III.5 Calcul simplifié du LLR et de l'APP pour les codes LDPC non-binaires

Dans l'algorithme de décodage des codes **LDPC** non-binaires, défini dans un corps de Galois $GF(2^s)$ avec *s* est un entier positif, les messages échangés, sont des **APPs** ou des **LLRs**, calculés sur les symboles non-binaires *a*, $a \in GF(2^s)$, du mot de code.

- $\boldsymbol{APP}(a)$ est calculée comme suit :

$$\boldsymbol{APP}(a = v) = Pr\{a = v/(a'_n, b'_n)\} \quad \text{(III.17)}$$

Où $Pr\{a = v/(a'_n, b'_n)\}$ représente la probabilité pour que le symbole *a* émis possède la valeur *v*, $v \in GF(2^q)$, sachant que le couple disponible en sortie du canal est (a'_n, b'_n).

Pour un corps de Galois $GF(2^s)$, les symboles non-binaires *a* du mot de code appartiennent à l'ensemble $\{0, 1, \dots, 2^s - 1\}$, où chaque symboles *a* dans $GF(2^s)$ correspond à la suite binaire $\{u_{n,1}, u_{n,2}, \dots, u_{n,s}\}$ dans $GF(2)$. Par conséquent, l'équation (III.17) devient:

$$\boldsymbol{APP}(a = v) = \prod_{i=1}^{s} Pr\{u_{n,i} = w/(a'_n, b'_n)\} \quad \text{(III.18)}$$

Où la valeur binaire *w* dépend de la valeur *v*.

En utilisant la règle de Bayes, l'expression (III.18) devient :

$$\boldsymbol{APP}(a = v) = \prod_{i=1}^{s} \boldsymbol{APP}(u_{n,i} = w) \quad \text{(III.19)}$$

- $\boldsymbol{LLR}(a)$ est calculé comme suit :

$$\boldsymbol{LLR}(a) = log\left[\frac{Pr\{a=v/(a_n^{'},b_n^{'})\}}{Pr\{a=0/(a_n^{'},b_n^{'})\}}\right] \quad \text{(III.20)}$$

En utilisant la règle de Bayes, l'expression (III.20) devient :

$$\boldsymbol{LLR}(a) = log\left[\frac{Pr\{(a_n^{'},b_n^{'})\ /a=v\}}{Pr\{(a_n^{'},b_n^{'})\ /a=0\}}\right] \quad \text{(III.21)}$$

En appliquant l'équation (III.19) dans (III.21), on obtient:

$$\boldsymbol{LLR}(a) = log\left[\frac{\prod_{i=1}^{s} Pr\{u_{n,i}=w/(a_n^{'},b_n^{'})\}}{\prod_{i=1}^{s} Pr\{u_{n,i}=0/(a_n^{'},b_n^{'})\}}\right] \quad \text{(III.22)}$$

On peut simplifier l'équation rigoureuse (III.22) comme suit:

$$\boldsymbol{LLR}(a) = log\left[\frac{Pr\{u_{n,1}=w/(a_n^{'},b_n^{'})\}\times Pr\{u_{n,2}=w/(a_n^{'},b_n^{'})\}\times\ldots\times Pr\{u_{n,s}=w/(a_n^{'},b_n^{'})\}}{Pr\{u_{n,1}=0/(a_n^{'},b_n^{'})\}\times Pr\{u_{n,2}=0/(a_n^{'},b_n^{'})\}\times\ldots\times Pr\{u_{n,s}=0/(a_n^{'},b_n^{'})\}}\right] \quad \text{(III.23)}$$

$$\boldsymbol{LLR}(a) = log\left[\frac{Pr\{u_{n,1}=w/(a_n^{'},b_n^{'})\}}{Pr\{u_{n,1}=0/(a_n^{'},b_n^{'})\}}\right] + log\left[\frac{Pr\{u_{n,2}=w/(a_n^{'},b_n^{'})\}}{Pr\{u_{n,2}=0/(a_n^{'},b_n^{'})\}}\right] + \cdots + log\left[\frac{Pr\{u_{n,s}=w/(a_n^{'},b_n^{'})\}}{Pr\{u_{n,s}=0/(a_n^{'},b_n^{'})\}}\right] \quad \text{(III.24)}$$

$$\boldsymbol{LLR}(a) = \boldsymbol{LLR}\left(u_{n,1}\right) + \boldsymbol{LLR}\left(u_{n,2}\right) + \cdots + \boldsymbol{LLR}\left(u_{n,s}\right) \quad \text{(III.25)}$$

Donc, l'équation (III.20) peut être simplifier à l'équation (III.26):

$$\boldsymbol{LLR}(a) = \sum_{i=1}^{s} \boldsymbol{LLR}\left(u_{n,i}\right) \quad \text{(III.26)}$$

Par conséquent, en appliquant le calcul simplifié du $\boldsymbol{LLR}\left(u_{n,i}\right)$, l'équation (III.26) devient plus simple. Dans la section suivant on montre l'effet de la simplification de l'**APP** et du **LLR** sur les performances d'un code **LDPC** non-binaire.

III.6 Effet de la simplification de l'APP sur les performances d'un code LDPC non-binaire

Dans cette partie, on va illustrer l'effet de la simplification du calcul de l'**APP** sur les performances d'un code **LDPC** non-binaire, défini dans un corps de Galois $GF(4)$, de rendement égal à 1/2 et d'une matrice de contrôle de parité de taille 512×1024, et pour un

algorithme de décodage utilisant l'**APP** à son entrée: le **FFT-SPA**, où le nombre d'itérations est fixé à quatre.

La chaîne de transmission pour laquelle nous avons évalué le Taux d'Erreurs Binaire (**TEB**) après le décodage est celle qui a été représentée sur la figure III.2, pour un code **LDPC** non-binaire, associé à deux constellations carrées: **MAQ**-16 et **MAQ**-64, ainsi que le codage de Gray associé, et sur deux canaux de transmission: gaussien et de Rayleigh.

La figure III.5 montre des comparaisons de performances, sur un canal gaussien, entre un code **LDPC** non-binaire utilisant le calcul exact de l'**APP**, les relations (III.7) et (III.8), un code utilisant le calcul simplifié (les relations (III.12)) en appliquant l'algorithme pragmatique (les relations (III.11)) et un code utilisant le calcul simplifié (les relations (III.12)) en appliquant l'algorithme max-log-**MAP** (les relations (III.9)).

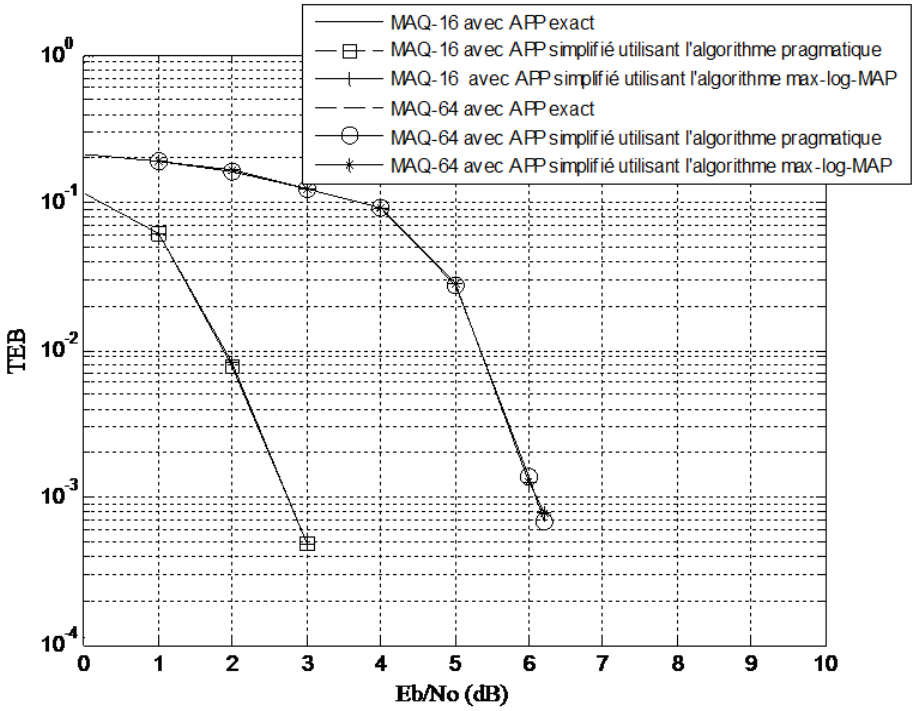


Figure III.5- Performances sur un canal gaussien d'un code **LDPC** non-binaire associé à une **MAQ**-16 et une **MAQ**-64, avec trois calculs de l'**APP**

La figure III.5 montre bien que le calcul simplifié de l'**APP** sur un canal gaussien, pour un algorithme max-log-**MAP** et un algorithme pragmatique, n'a aucun effet sur les performances d'un code **LDPC** non-binaire.

Afin d'étudier l'influence du calcul simplifié sur les performances d'un code **LDPC** non-binaire sur un canal de Rayleigh, les mêmes comparaisons de performances de la figure

III.5 sont effectuées sur un canal de Rayleigh, dans deux figures III.6 et III.7 pour bien montrer la différence

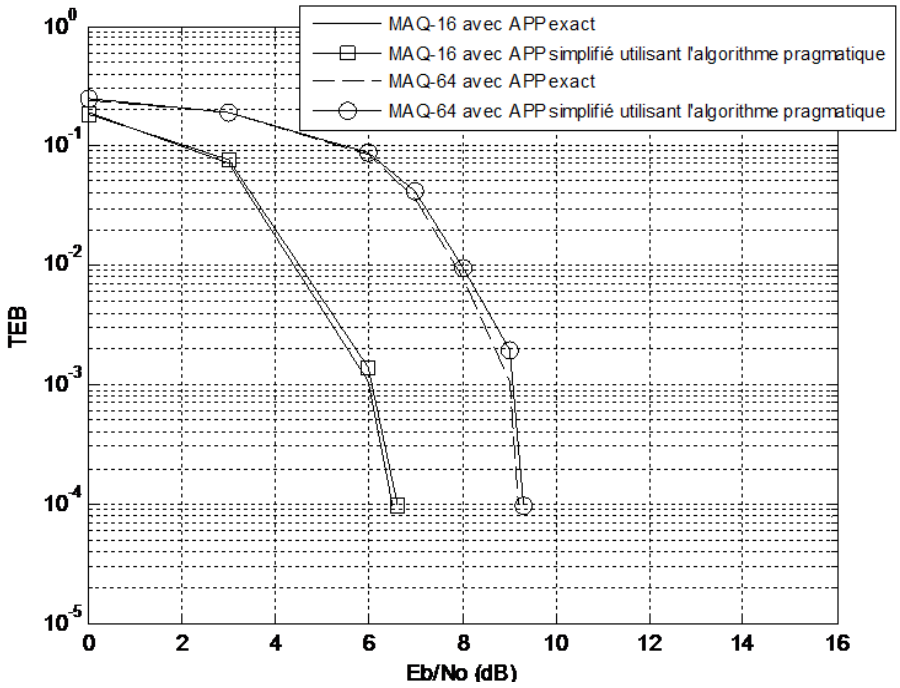


Figure III.6- Performances sur un canal de Rayleigh d'un code **LDPC** non-binaire, avec deux calculs de l'**APP**: exact et simplifié utilisant l'algorithme pragmatique

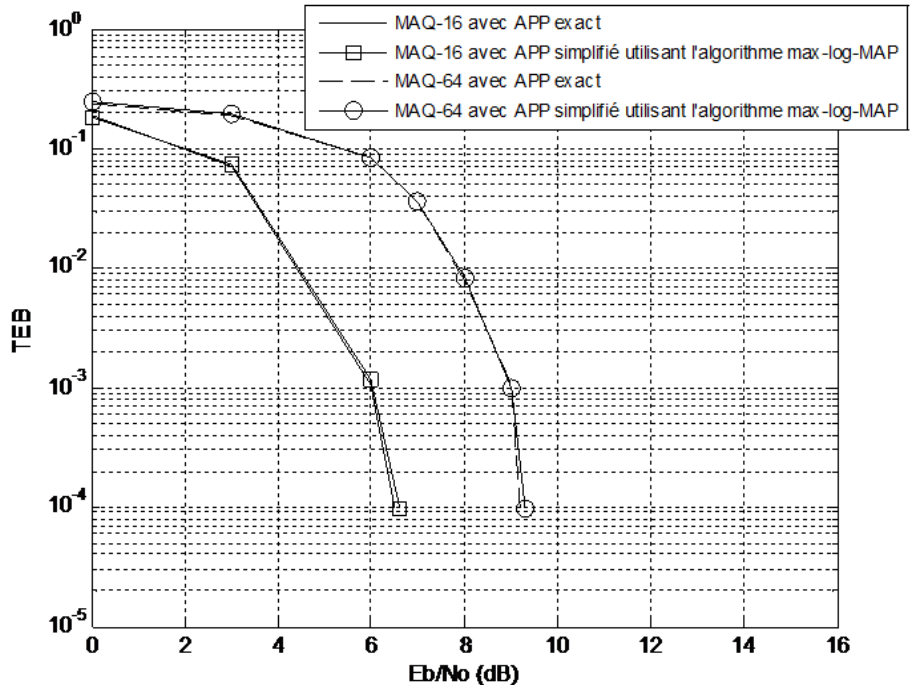


Figure III.7- Performances sur un canal de Rayleigh d'un code **LDPC** non-binaire, avec deux calculs de l'**APP**: exact et simplifié utilisant l'algorithme max-log-**MAP**

Les résultats obtenus sur un canal de Rayleigh montre que, pour un algorithme simplifié de l'**APP** utilisant l'algorithme pragmatique (figure III.6), il existe une très petite dégradation de performances d'un code **LDPC** non-binaire. Tandis que pour un algorithme simplifié de l'**APP** utilisant l'algorithme max-log-**MAP** (figure III.7), il existe une légère dégradation à fort E_b/N_o.

III.7 Effet de la simplification du LLR sur les performances d'un code LDPC non-binaire

Dans cette section, on va présenter l'effet de la simplification du calcul du **LLR** sur les performances d'un code **LDPC** non-binaire, défini dans un corps de Galois $\boldsymbol{GF}(4)$, de rendement égal à 1/2 et d'une matrice de contrôle de parité de taille 128×256, et pour un algorithme de décodage utilisant le **LLR** à son entrée: l'algorithme **EMS**. Le code **LDPC** non-binaire est associé à trois constellations carrées: **MAQ-**16, **MAQ**-64 et **MAQ**-256, ainsi que le codage de Gray associé, et sur un canal gaussien.

La figure III.8 montre des comparaisons de performances, sur un canal gaussien, entre un code **LDPC** non-binaire utilisant le calcul exact du **LLR** et un code utilisant le calcul simplifié en appliquant l'algorithme pragmatique.

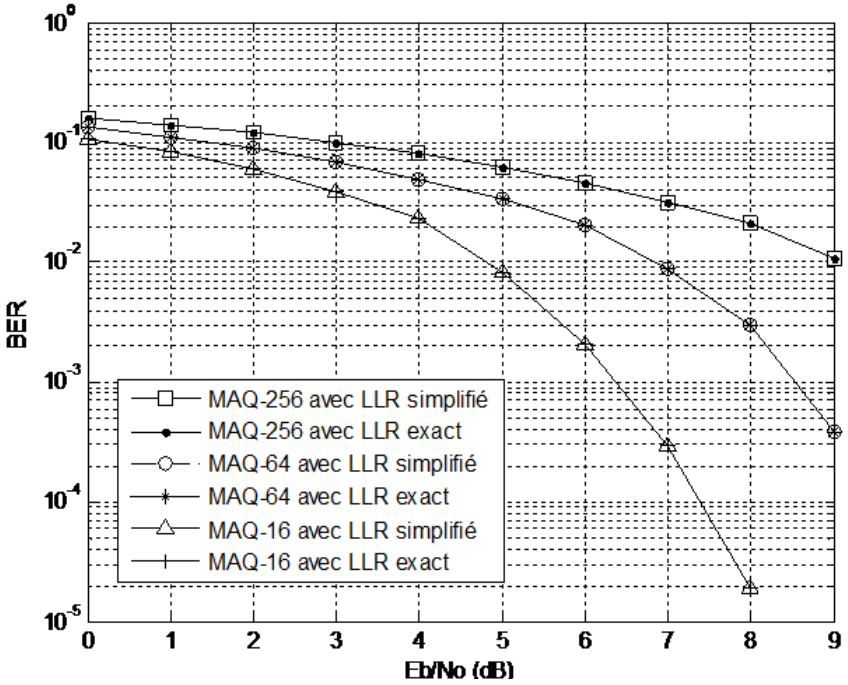


Figure III.8- Performances sur un canal de Gauss d'un code **LDPC** non-binaire, avec deux calculs du **LLR**: exact et simplifié

La figure III.8 montre bien que le calcul simplifié du **LLR** sur un canal gaussien, pour un algorithme pragmatique, n'a aucun effet sur les performances d'un code **LDPC** non-binaire.

III.8 Conclusion

Dans ce chapitre, nous avons adapté le calcul simplifié du **LLR**, introduit pour les codes binaires, sur les codes **LDPC** non-binaires. Ainsi, nous avons utilisé ce même calcul pour faciliter le calcul de l'**APP** pour les codes **LDPC** non-binaires.

La méthode proposée, pour faire ces simplifications, met un système combinant un code **LDPC** non-binaire et une constellation d'ordre élevée simple en œuvre. Elle est programmée afin d'adapter le plus parfaitement possible le système au type de canal considéré. Aussi, elle permet de garantir un décodage efficace quel que soit le type de canal considéré.

Dans le chapitre suivant, on s'intéresse à l'amélioration d'un système combinant une constellation d'ordre élevé et un nouveau code correcteur d'erreurs puissant.

Chapitre IV

*Turbo **LDPC** Code **B**inaire et Non-**B**inaire*

IV.1 Introduction

Bien que les codes **LDPC** sont des bonnes codes correcteurs d'erreurs, et les codes **LDPC** non-binaires offrent de meilleures performances que leurs équivalents binaires lorsque le bloc codé est de longueur faible à modérée, ou lorsque la modulation utilisée est à grand nombre d'états, l'amélioration de ces codes est intéressante lorsqu'ils sont associés à des modulations d'ordre élevées.

Dans ce chapitre nous utilisons le schéma des turbo-codes parallèles proposé par Berrou et autres, et nous appliquons les codes **LDPC** sur ce schéma. Ainsi, nous évaluons les performances de ce code proposé, associé à des constellations d'ordre élevés, sur un canal gaussien et sur un canal de Rayleigh. Tout d'abord, nous présentons les turbo **LDPC** codes parallèles pour les codes binaires et non-binaires. Ensuite, nous discutons les résultats de simulation du code proposé associé à des constellations d'ordre élevé sur un canal gaussien et sur un canal de Rayleigh.

IV.2 Principe du codage d'un turbo LDPC code

La figure IV.1 représente la structure d'un turbo **LDPC** code à concaténation parallèle, construit à partir de deux codes **LDPC** élémentaires ENC_1 et ENC_2, séparés par un entrelaceur noté π introduisant de la diversité. Ce dernier doit permettre d'augmenter les distances libres des codes concaténés [61].

Le rendement d'un turbo-code à concaténation parallèle R_{cp} est donné par [62]:

$$R_{cp} = \frac{R_1.R_2}{R_1+R_2-R_1.R_2} \tag{IV.1}$$

Avec R_1 est le rendement du premier code élémentaire ENC_1 et R_2 est le rendement du deuxième code élémentaire ENC_2.

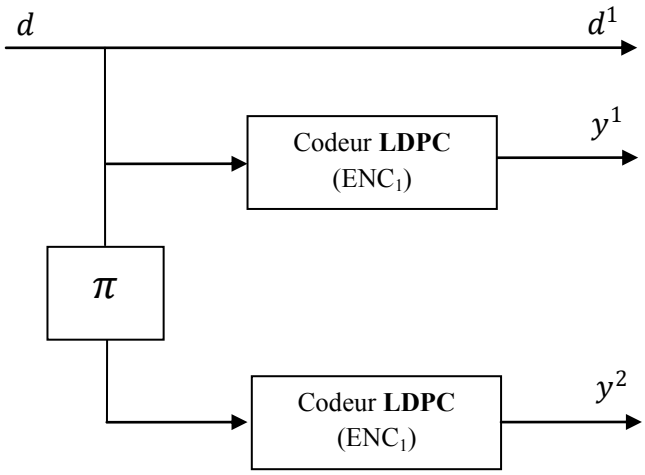


Figure IV.1- Turbo **LDPC** codeur parallèle

Les deux codes élémentaires ENC_1 et ENC_2 utilisent les mêmes blocs d'entrées, mais suivant des séquences différentes. Ceci est rendu possible par la présence de l'entrelaceur [10].

L'entrelacement est une opération qui change l'ordre des symboles à l'émission pour les remettre en ordre à la réception. Elle a pour but d'espacer les symboles consécutifs et de transférer un canal à paquets d'erreurs en un canal à erreurs indépendants. On peut utiliser l'entrelacement dans la concaténation de codes, et aussi avec un seul code correcteur d'erreurs dans la chaine de transmission pour les canaux à évanouissement [55].

Le premier code élémentaire ENC_1 utilise le bloc d'information d de taille N-M, $d = [d_1\ d_2\ \dots d_{N-M}]$, avec une matrice de contrôle de parité H de taille $M \times N$, et génère le bloc d'information codé de taille N:

$$[y^1\ d^1] = [y_1^1\ y_2^1\ \dots y_M^1\ d_1^1\ d_2^1\ \dots d_{N-M}^1] \tag{IV.2}$$

Avec d^1 est le bloc systématique $d^1 = d$, et y^1 est le bloc de parité.

Le deuxième code élémentaire ENC_2 utilise le bloc d'information entrelacé $d_{entrelac\ é}$, et génère le bloc codé de taille N:

$$[y^2\ d^2] = [y_1^2\ y_2^2\ \dots y_M^2\ d_1^2\ d_2^2\ \dots d_{N-M}^2] \tag{IV.3}$$

Avec d^2 est le bloc d'information entrelacé $d^2 = d^1_{entrelac\ é} = d_{entrelac\ é}$, et y^2 est le bloc de parité.

Puisque les deux codes élémentaires sont systématiques et opèrent la même séquence de symboles d'entrée, on n'a pas besoin de transmettre l'entrée du deuxième code ENC_2, et ceci augmente le rendement de turbo-code. Elle est utile seulement pour doubler la diversité en présence d'évanouissements [19].

Par conséquent, pour un bloc d'information de taille *N-M*, $d = [d_1\ d_2 \dots d_{N-M}]$, le turbo **LDPC** codeur génère le bloc d'information codé de taille *N:*

$$[y^2 y^1 d^1] = [y_1^2\ y_2^2 \dots y_M^2\ y_1^1\ y_2^1 \dots y_M^1\ d_1^2\ d_2^1 \dots d_{N-M}^1] \quad \text{(IV.4)}$$

La concaténation parallèle de plus de deux codes [63] de faible complexité donne des turbo-codes de faibles rendements.

IV.3 Principe du décodage d'un turbo LDPC code

Le turbo-décodage se fait selon le principe de décodage itératif [64] ou turbo basé sur l'utilisation de décodeurs à entrée et à sortie pondérée ou **SISO** (Soft-Input Soft-Output) [65] qui s'échangent des informations de fiabilité, appelées informations extrinsèques, par le biais d'une contre-réaction, afin d'améliorer la correction au fil des itérations.

Le terme TURBO (Toggle Until Regenerations Bring Optimality) vient de la notion de bouclage semblable à celle utilisée dans les moteurs turbo.

Dans un circuit électronique le processus turbo s'explique comme ceci :

On fait un premier traitement, les résultats ont été mis en mémoire et on reprend ce traitement en faisant bénéficier le traitement en amont des résultats du traitement en aval. On procède plusieurs fois ce type d'opération dans le cas du turbo-décodage.

Un turbo **LDPC** décodeur parallèle présenté à la figure IV.2, est constitué de deux décodeurs élémentaires **SISO** DEC_1 et DEC_2 associés respectivement au code ENC_1 et ENC_2 disposés en parallèle, de deux entrelaceurs et d'un désentrelaceur noté π^{-1}.

Le turbo **LDPC** décodeur contient deux décodeurs **LDPC** décodés de manière itérative. Par conséquent, chaque itération $iter_{turbo}$ de turbo **LDPC** code contient de multiples itérations $iter_{ldpc}$. Les performances d'un turbo code **LDPC** peuvent être améliorer avec l'augmentation du nombre des itérations $(iter_{ldpc}, iter_{turbo})$.

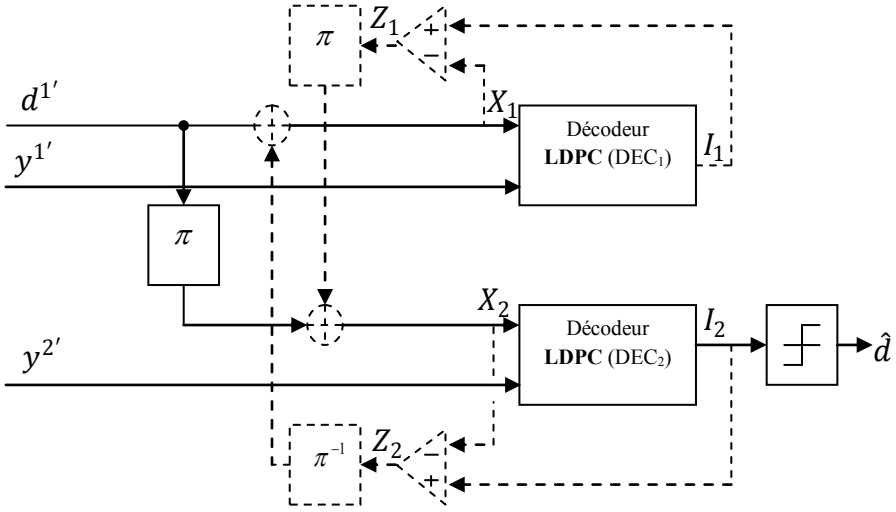


Figure IV.2- Turbo **LDPC** décodeur parallèle

Le turbo-décodeur reçoit les observations $[y^{2'}\, y^{1'}\, d^{1'}]$ en provenance du canal et estime le message émis.

Les deux décodeurs DEC_1 et DEC_2 travaillent conjointement, de telle manière que le décodeur DEC_1 puisse tirer bénéfice de $y^{2'}$ et le décodeur DEC_2 de $y^{1'}$. Ils fournissent une première estimation, chacun communique ses résultats à l'autre pour une nouvelle passe. Ils fournissent ensuite une seconde estimation. Après, chacun communique ses résultats à l'autre et ainsi de suite. Le décodage s'arrête au bout d'un nombre fixe d'itération, et la décision finale peut venir du DEC_1 ou du DEC_2.

Le turbo **LDPC** décodeur reçois les observations pondérées $[y^{2'}\, y^{1'}\, d^{1'}]$. A chaque itération, les deux décodeurs DEC_1 et DEC_2 travaillent conjointement et utilisent respectivement les blocs d'entrée $[y_1', X_1]$ et $[y_2', X_2]$ avec X_1 et X_2 sont donnés par:

$$X_1 = \begin{cases} d^{1'} & pour\ la\ première\ itération \\ d^{1'} + Z_{2\ désentrelac\ é} & pour\ les\ autres\ itérations \end{cases} \qquad \text{(IV.5)}$$

$$X_2 = \begin{cases} d^{1'}_{entrelacé} & pour\ la\ première\ itération \\ d^{1'}_{entrelacé} + Z_{1\ entrelac\ é} & pour\ les\ autres\ itérations \end{cases} \quad \text{(IV.6)}$$

Avec Z_1 et Z_2 sont égales à:

$$Z_1 = I_1 - X_1 \quad \text{(IV.7)}$$

$$Z_2 = I_2 - X_2 \quad \text{(IV.8)}$$

Le soustracteur placé à la sortie de chaque décodeur DEC_1 et DEC_2 est utilisé pour ne garder que la contribution de chacun (information extrinsèque).

La présence de l'entrelaceur π et de désentrelaceur π^{-1} respectivement à la sortie du décodeur DEC_1 et celle du décodeur DEC_2 ont pour rôle de décorréler les décisions pondérées en sortie de chaque décodeur [66].

Plusieurs méthodes d'entrelacement sont possibles. Cependant, le choix de la structure d'un entrelaceur est un facteur clé qui détermine les performances d'un turbo **LDPC** code, dans le sens qu'il modifie leur propriété de distance libre.

Afin que ce turbo **LDPC** décodeur s'effectue correctement même après plusieurs itérations de décodage, l'entrelacement et le désentrelacement doivent s'effectuer de manière pseudo-aléatoire ou aléatoire.

Grâce à ces deux types d'entrelacement, les turbo **LDPC** codes paraissent aléatoires au canal (notons que les codes aléatoires sont ceux qui ont été utilisés par Shannon), ce qui constitue une caractéristique dont le décodage peut tirer bénéfice.

IV.4 Performances d'un turbo LDPC code binaire et non-binaire

Les résultats présentés dans la littérature [67-70, 71] montrent que la concaténation des codes **LDPC** binaires irréguliers, est plus performante qu'un seul code **LDPC** binaire. Notons que les codes **LDPC** irréguliers ont pour principale caractéristique de présenter de meilleures performances que les codes réguliers. Cependant, les codes **LDPC** irréguliers ont un plancher d'erreurs et une complexité de codage plus élevés que les codes réguliers.

De ce fait, nous proposons l'utilisation des codes élémentaires réguliers identiques, pour les turbo **LDPC** codes binaires, avec un entrelaceur entre eux afin d'introduire la diversité.

Nous présentons dans les figures IV.3, IV.4 et IV.5, sur un canal gaussien, respectivement avec une **MAQ**-16, une **MAQ**-64 et une **MAQ**-256 utilisant le codage de Gray, des comparaisons de performances d'un turbo **LDPC** code binaire de rendement 1/3, composé de deux code **LDPC** réguliers binaires de rendement 1/2 et d'une matrice de contrôle de parité de taille 512×1024, avec un seul code **LDPC** de rendement 1/3 et d'une matrice de contrôle de parité de taille 1024×1536.

Nous remarquons que le code proposé présente aussi de meilleures performances qu'un seul code **LDPC** binaire. Nous constatons que, comme indiqué précédemment, les performances d'un turbo **LDPC** code peuvent être améliorées avec l'augmentation du nombre des itérations $(iter_{ldpc}, iter_{turbo})$.

Dans [67] les auteurs ont montré que, l'entrelaceur n'est pas nécessaire lorsque le code **LDPC** binaire est concaténé avec un autre code, afin d'étudier l'effet de l'entrelaceur entre les deux codes **LDPC** composants, un **PCGC** (Parallel Concatenated Gallager Codes) proposé dans [67] a été modifié pour utiliser un entrelaceur afin de permuter les bits d'information tel que présenté dans [70] pour des codes irréguliers.

Afin d'étudier l'effet de l'entrelacement sur les performances d'un turbo **LDPC** code binaire, où les codes constituants sont des codes réguliers identiques, des comparaisons de performances entre un turbo **LDPC** code binaire avec et sans entrelacement sont effectuées sur la figure IV.3. Nous montrons que l'entrelaceur à un effet positif sur les performances d'un turbo **LDPC** code binaire.

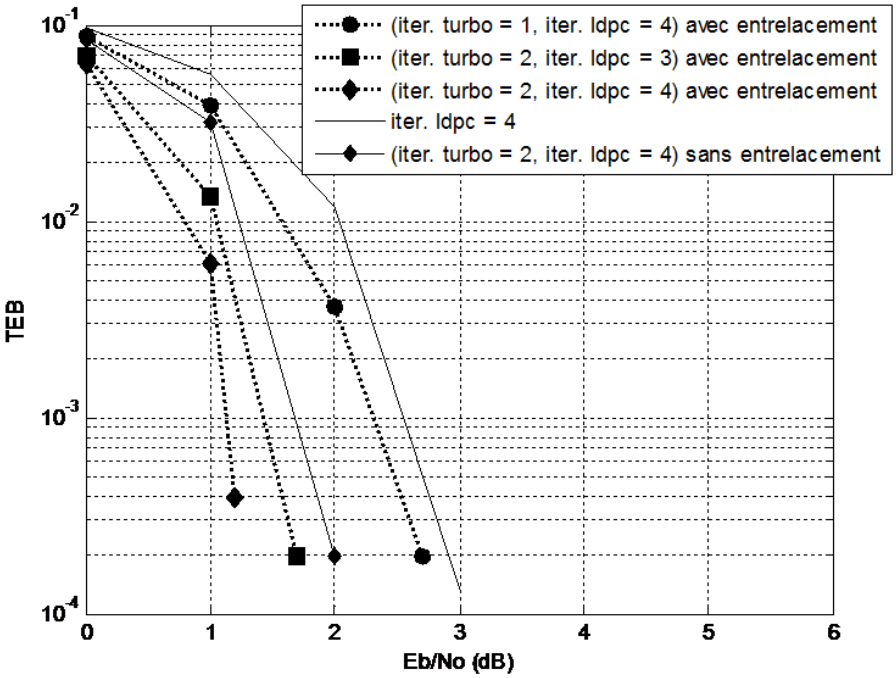


Figure IV.3- Comparaisons de performances, sur un canal gaussien, d'un turbo **LDPC** code binaire, avec et sans entrelacement, et un seul code **LDPC** binaire de même rendement égal à 1/3, pour une **MAQ-**16

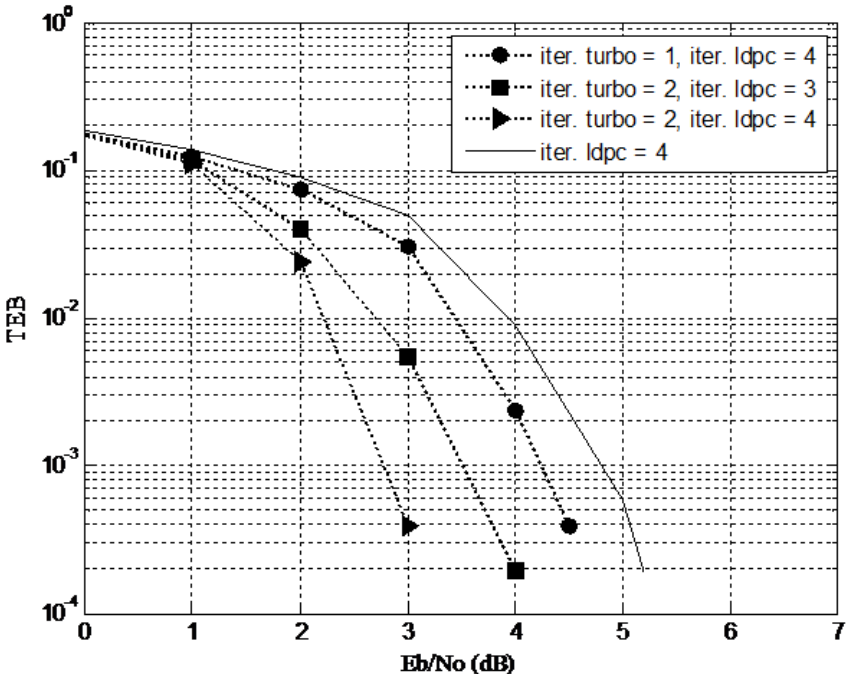


Figure IV.4- Comparaisons de performances, sur un canal gaussien, d'un turbo **LDPC** code binaire, et un seul code **LDPC** binaire de même rendement égal à 1/3, pour une **MAQ-**64

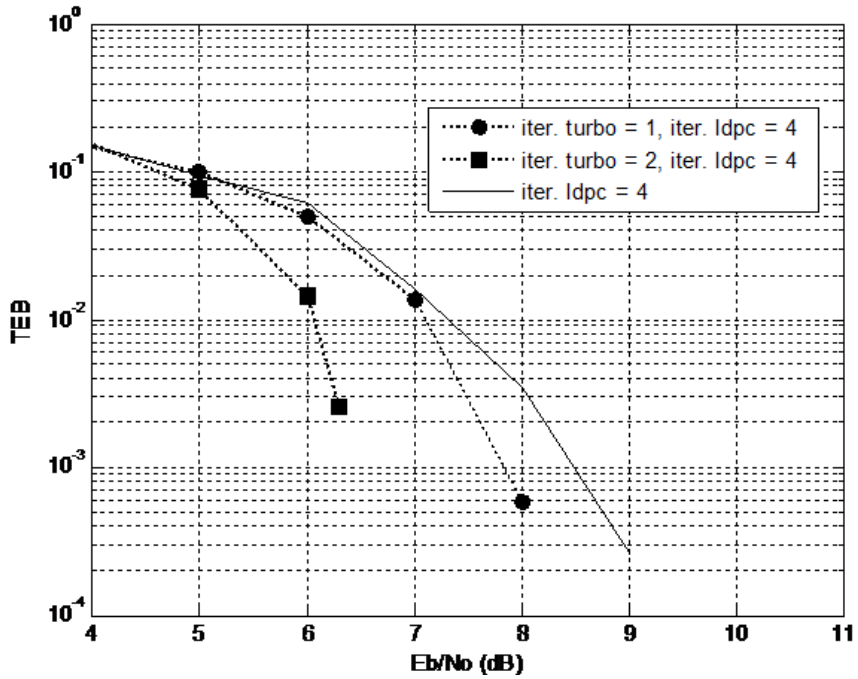


Figure IV.5- Comparaisons de performances, sur un canal gaussien, d'un turbo **LDPC** code binaire, et un seul code **LDPC** binaire de même rendement égal à 1/3, pour une **MAQ**-256

Dans le but d'étudier les performances du code proposé sur un canal de Rayleigh, une comparaison de performances est effectuée, sur un canal de Rayleigh avec une **MAQ**-16, dans la figure IV.6. Ceci pour un turbo **LDPC** code binaire de rendement 1/3, composé de deux codes **LDPC** réguliers binaires de rendement 1/2 et d'une matrice de contrôle de parité de taille 512×1024, avec un seul code **LDPC** de rendement 1/3 et d'une matrice de contrôle de parité de taille 1024×1536.

Nous remarquons que pour un canal de Rayleigh, on obtient un gain de codage, entre un turbo **LDPC** code et un seul code, plus élevé que le gain obtenu pour un canal de Gauss. Ceci est dû à l'effet d'entrelacement.

Les résultats de la figure IV.7 montrent aussi qu'on peut obtenir un gain de codage entre un turbo **LDPC** code et un seul code, en augmentant le rendement du code. Ces résultats sont obtenus, sur un canal gaussien avec une **MAQ**-16 utilisant le codage de Gray, pour un turbo **LDPC** code parallèle binaire de rendement 1/2, composé de deux codes **LDPC** réguliers binaires de rendement 2/3 et d'une matrice de contrôle de parité de taille 256×768,

avec un seul code **LDPC** de rendement 1/2 et d'une matrice de contrôle de parité de taille $512 \times 1($

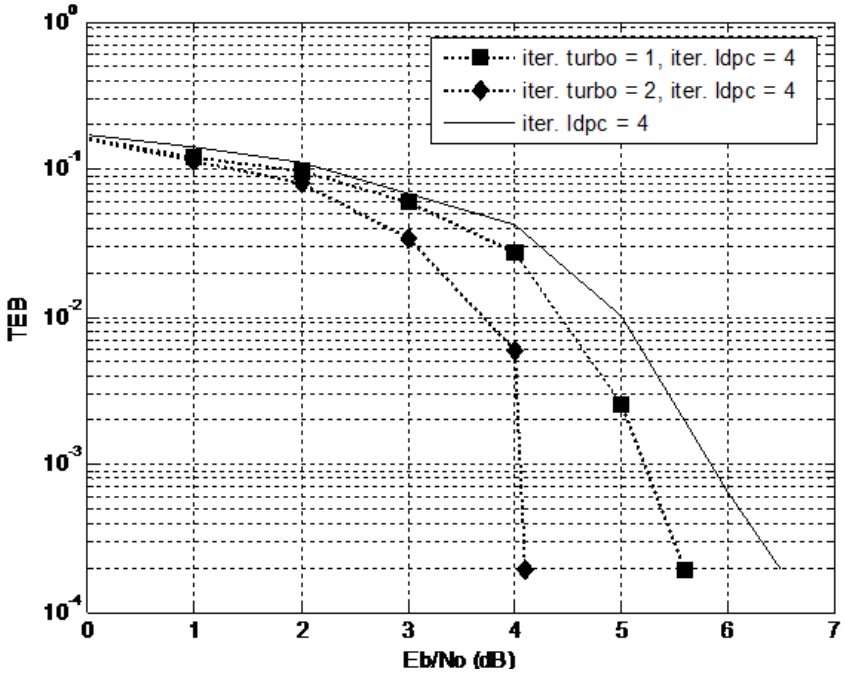


Figure IV.6- Comparaisons de performances, sur un canal de Rayleigh, d'un turbo **LDPC** code binaire, et un seul code **LDPC** binaire de même rendement égal à 1/3, pour une **MAQ-**16

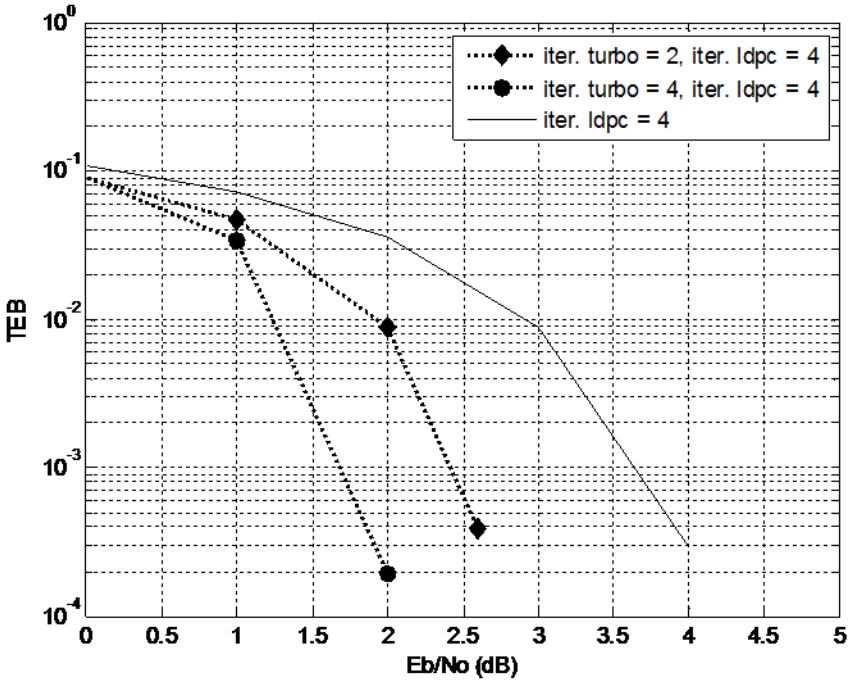


Figure IV.7- Comparaisons de performances, sur un canal gaussien, d'un turbo **LDPC** code binaire, et un seul code **LDPC** binaire de même rendement égal à 1/2, pour une **MAQ-**16

Bien que les codes **LDPC** non-binaires offrent de meilleures performances que leurs équivalents binaires lorsque le bloc codé est de longueur faible à modérée, ou lorsque la modulation utilisée est à grand nombre d'états, la concaténation de ces codes avec un décodage itératif est encore attrayante pour construire des codes correcteurs d'erreurs puissants. Dans notre travail nous introduisons un turbo **LDPC** code non-binaire avec les codes élémentaires sont des codes **LDPC** réguliers non-binaires, avec un entrelaceur entre eux.

Les figures IV.8 et IV.9 illustrent, avec une **MAQ**-16 utilisant le codage de Gray, respectivement sur un canal gaussien et sur un canal de Rayleigh, les comparaisons de performances d'un turbo **LDPC** code non-binaire de rendement 1/3, composé de deux codes **LDPC** non-binaires, définis dans un corps de Galois $GF(4)$, de rendement 1/2 et d'une matrice de contrôle de parité de taille 512×1024, avec un seul code **LDPC** de rendement 1/3 et d'une matrice de contrôle de parité de taille 1024×1536 .

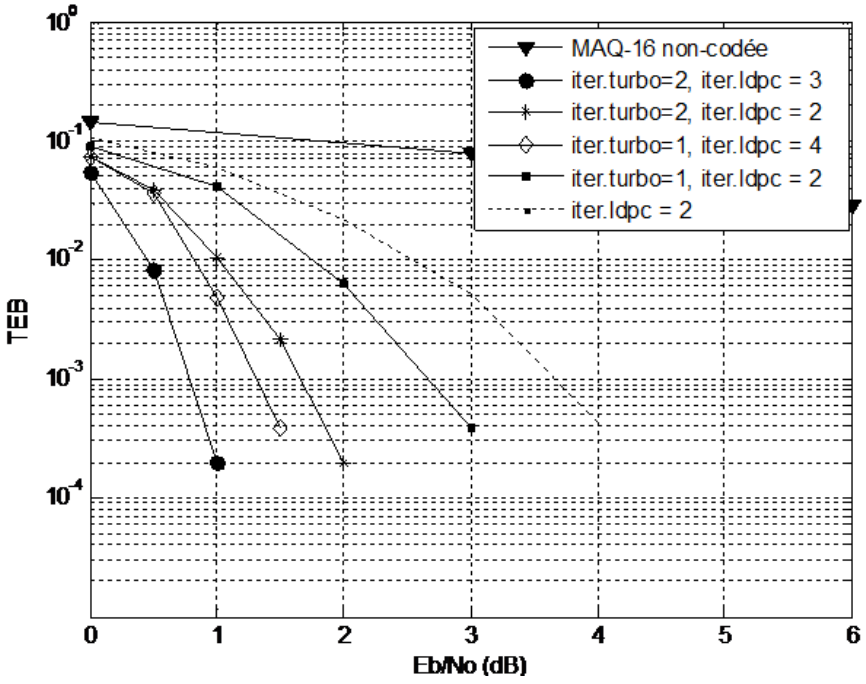


Figure IV.8- Comparaisons de performances, sur un canal gaussien, d'un turbo **LDPC** code non-binaire, et un seul code **LDPC** non-binaire de même rendement égal à 1/3, pour une **MAQ**-16

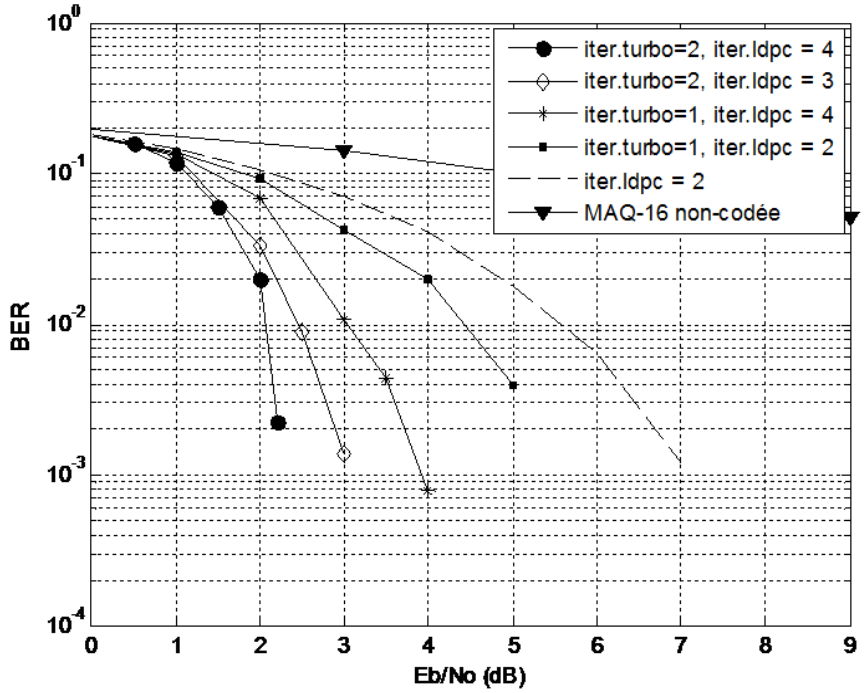


Figure IV.9- Comparaisons de performances, sur un canal de Rayleigh, d'un turbo **LDPC** code non-binaire, et un seul code **LDPC** non-binaire de même rendement égal à 1/3, pour une **MAQ**-16

Vus des résultats obtenus, dans les figures IV.8 et IV.9, le code proposé présente de meilleurs performances qu'un seul code **LDPC** non-binaire. Nous montrons dans ces figures, les améliorations de performance d'un turbo **LDPC** code non-binaire de rendement 1/3 avec plusieurs valeurs d'itérations.

Nous remarquons aussi qu'un seul code est moins performant qu'un turbo **LDPC** code avec même itération $iter_{turbo}$. Les performances de ce dernier peuvent être augmentées avec le nombre des itérations de ses codes élémentaires $iter_{ldpc}$.

Ces résultats montrent bien le fait que dans le cas d'un canal de Rayleigh, le gain de codage entre un turbo **LDPC** code et un seul code est plus élevé que le gain obtenu pour un canal de Gauss. Cela signifie que l'entrelacement a un bon effet sur les canaux de Rayleigh.

IV.5 Conclusion

Dans ce chapitre, nous avons introduit un nouveau schéma de code correcteur d'erreurs, baptisé turbo **LDPC** code, permettant d'améliorer les performances d'un code **LDPC** de même rendement et de même longueur de bloc d'entrée.

Conclusion Générale et Perspectives

L'objectif de cette thèse était d'étudier un nouveau code correcteur d'erreurs, appelé turbo **LDPC** code. Et de simplifier la combinaison de codes **LDPC** non-binaires avec des constellations d'ordre élevé, dans deux types de canaux de transmission: gaussien et de Rayleigh, en simplifiant le calcul des décisions pondérées à l'entrée du décodeur.

L'étude d'un tel sujet a nécessité d'étudier les codes **LDPC** non-binaires qui sont des codes en bloc linéaires, et de les situer à l'intérieur d'un système de transmission numérique. On a pour cela, au premier chapitre et sous forme introductive, identifié les différentes fonctions qui sont présentes dans un tel système. Ensuite, on a discuté sous forme introductive les codes en blocs linéaires. Le deuxième chapitre a été axé sur l'étude en détail des codes **LDPC** binaires et non-binaires.

Le chapitre trois était orienté à l'étude de la simplification du calcul de l'**APP** et du **LLR** pour une constellation carrée d'ordre élevé. On a montré que la méthode proposée, permettant de simplifier le calcul de l'**APP**, permet aussi de simplifier la combinaison d'un code **LDPC** non-binaire avec une constellation d'ordre élevé.

Ensuite, les résultats de la simulation montrent que la simplification du calcul de l'**APP** n'a aucun effet sur les performances d'un code **LDPC** non-binaire pour un canal gaussien. Tandis que, pour un canal de Rayleigh, il y a une très petite dégradation de performances presque nulle. Aussi, les résultats montrent que la simplification du calcul du **LLR** n'a aucun effet sur les performances d'un code **LDPC** non-binaire pour un canal gaussien.

Dans le chapitre IV, nous avons étudié les turbo **LDPC** codes binaires et non-binaires. Ainsi, nous avons évalué les performances de ces codes proposés, associés à des constellations d'ordre élevé, sur un canal gaussien et sur un canal de Rayleigh. Les résultats de la simulation montrent que les turbo **LDPC** codes offrent de meilleures performances qu'un seul code **LDPC**. Ces résultats montrent bien le fait que dans le cas d'un canal de Rayleigh, le gain de codage entre un turbo **LDPC** code et un seul code est plus élevé que le gain obtenu

pour un canal de Gauss. Ceci est dû à la présence de l'entrelacement entre les codes constituants. Nous avons remarqué aussi que les performances d'un turbo **LDPC** code peuvent être améliorées avec le nombre des itérations $iter_{turbo}$ et $iter_{ldpc}$. Ainsi, on a montré qu'on peut obtenir un gain de codage entre un turbo **LDPC** code et un seul code si on augmente le rendement du code.

Par la suite, il serait intéressant d'optimiser les turbo **LDPC** codes, en jouant sur les paramètres de la matrice de contrôle de parité pour trouver le meilleur code. Ainsi, il serait intéressant d'optimiser ces codes pour une transmission radio-mobile.

Annexe A

Notions Sur les Corps de Galois $GF(q)$

Les corps de Galois font partie d'une branche particulière des mathématiques qui modélise les fonctions du monde numérique. La dénomination « corps de Galois » provient du mathématicien français Galois qui en a découvert les propriétés fondamentales.

Il y a deux types de corps, les corps finis et les corps infinis. Les corps de Galois finis sont des ensembles d'éléments fermés sur eux-mêmes. L'addition et la multiplication de deux éléments du champ donnent toujours un élément du corps fini.

A.1 Définition

Un corps de Galois à $q = 2^m$ éléments noté $GF(q)$, où m est un entier positif. Le corps de Galois consiste en un ensemble de nombres, ces nombres sont constitués à l'aide de l'élément de base α comme suit :

$$GF(q) = \{0, \alpha^0, \alpha^1, \ldots, \alpha^{q-2}\} \text{ avec } \alpha^{q-1} = 1 \qquad \text{(A.1)}$$

Avec α est l'élément primitif du corps de Galois $GF(q)$.

$GF(q)$ est formé à partir du corps de base $GF(2)$, où $GF(2) = \{0, 1\}$.

Exemple 1 : $GF(4) = \{0, \alpha^0, \alpha^1, \alpha^2\} = \{0, 1, \alpha, \alpha^2\}$

A.2 Polynôme primitif

Le polynôme primitif, ayant α comme racine, est utilisé pour construire le corps $GF(q)$, il est sous la forme suivante :

$$P(\alpha) = p_0 + p_1.\alpha + p_2.\alpha^2 + \ldots + p_m.\alpha^m \qquad \text{(A.2)}$$

Le polynôme primitif doit être :

- irréductible c'est-à-dire être non factorisable dans $GF(2)$ (autrement dit 0 et 1 ne sont pas racines de $P(\alpha)$;
- de degré m ;
- à coefficients dans $GF(2)$ (c-à-d $p_i = 0\ ou\ 1$, avec $i = 0,\ldots,m$) ;
- diviseur de $x^{2^m-1}+1$, où l'opération de la division se faite modulo 2.

❖ **Détermination du polynôme primitif**

Prenons un exemple d'un corps de Galois $GF(4)$, avec $m=2$, et on détermine son polynôme primitif comme suit :

Pour $m=2$ on peut avoir $2^m=4$ polynômes de degré m :

$$\begin{aligned} P_1(x) &= 1+x+x^2 \\ P_2(x) &= 1+x^2 \\ P_3(x) &= x+x^2 \\ P_4(x) &= x^2 \end{aligned} \qquad \text{(A.3)}$$

Pour déterminer le polynôme irréductible, on remplace 0 et 1 dans chaque polynôme et on prend le polynôme où 0 et 1 ne sont pas des racines (c-à-d $P(0)\neq 0\ et\ P(1)\neq 0$). On trouve que $P_1(x)=1+x+x^2$ est un polynôme irréductible.

On fait la division polynomiale : la division de $x^{2^m-1}+1 = x^{2^2-1}+1 = x^3+1$ par $P_1(x)=1+x+x^2$**.** On trouvé le reste = 0. Donc, le polynôme $P_1(x)$ est diviseur $x^{2^m-1}+1$.

Par conséquent : le polynôme $P_1(x)=1+x+x^2$ est un polynôme primitif du corps $GF(4)$.

Comme mentionné ci-dessus, l'élément α est racine du polynôme primitif, on peut en déduire :

$$P(x) = 1 + x + x^2$$
$$P(\alpha) = 1 + \alpha + \alpha^2 \qquad \text{(A.4)}$$
$$0 \quad = 1 + \alpha + \alpha^2 \Rightarrow \alpha^2 = \alpha + 1$$

Toutes les opérations dans le corps de Galois sont faites modulo 2 et modulo polynôme primitif. Donnons pour le corps $GF(4)$, à titre d'exemple, les règles régissant les opérations d'addition (Tableau A.1) et de multiplication (Tableau A.2). Les opérations sont faites modulo 2 et modulo $\alpha^2 = \alpha + 1$.

+	0	1	α	α^2
0	0	1	α	α^2
1	1	0	$1+\alpha=\alpha^2$	$1+\alpha^2=\alpha$
α	α	$1+\alpha=\alpha^2$	0	$\alpha+\alpha^2=1$
α^2	α^2	$1+\alpha^2=\alpha$	$\alpha+\alpha^2=1$	0

Tableau A.1- Adition dans $GF(4)$

×	0	1	α	α^2
0	0	0	0	0
1	0	1	α	α^2
α	0	α	α^2	$\alpha^3=1$
α^2	0	α^2	$\alpha^3=1$	$\alpha^4=\alpha$

Tableau A.2- Multiplication dans $GF(4)$

A.3 Représentation des éléments du $GF(q)$

Les éléments du corps de Galois $GF(q)$ sont définis modulo polynôme primitif. Chaque élément de ce corps peut être représenté par deux types de représentations polynomiale et binaire :

▪ **Représentation polynômiale** : l'élément est représenté par un polynôme de degré égal à (m-1) et à coefficient dans $GF(2)$, sous la forme :

$$\alpha^{m-1}x^{m-1} + \alpha^{m-2}x^{m-2} + \ldots + \alpha.x + \alpha^0 \qquad \text{(A.5)}$$

Avec : $\alpha^{m-1} \ldots \alpha^0$ éléments dans $GF(2)$

▪ **Représentation binaire** : l'élément est représenté par une séquence des éléments dans $GF(2)$, sous la forme :

$$\alpha^{m-1}\alpha^{m-2}\alpha^{m-3}\ldots\alpha^0 \qquad \text{(A.6)}$$

Où $\alpha^{m-1} \ldots \alpha^0$ correspondent aux éléments de l'équation (A.2).

Le tableau (A.3) illustre les différentes représentations des éléments du corps de Galois $GF(16)$.

Eléments du corps	Représentation polynomiale	Représentation binaire
0	0	0000
1	1	0001
α	α	0010
α^2	α^2	0100
α^3	α^3	1000
α^4	$\alpha+1$ (polynôme primitif)	0011
α^5	$\alpha.\alpha^4=\alpha.(\alpha+1)=\alpha^2+\alpha$	0110
α^6	$\alpha.\alpha^5=\alpha.(\alpha^2+\alpha)=\alpha^3+\alpha^2$	1100
α^7	$\alpha.\alpha^6=\alpha.(\alpha^3+\alpha^2)=\alpha^4+\alpha^3=\alpha^3+\alpha+1$	1011
α^8	$\alpha^4.\alpha^4=(\alpha+1)(\alpha+1)=\alpha^2+1+2.\alpha=\alpha^2+1$	0101
α^9	$\alpha.\alpha^8=\alpha.(\alpha^2+1)=\alpha^3+\alpha$	1010
α^{10}	$\alpha.\alpha^9=\alpha.(\alpha^3+\alpha)=\alpha^4+\alpha^2=\alpha^2+\alpha+1$	0111
α^{11}	$\alpha.\alpha^{10}=\alpha.(\alpha^2+\alpha+1)=\alpha^3+\alpha^2+\alpha$	1110
α^{12}	$\alpha.\alpha^{11}=\alpha.(\alpha^3+\alpha^2+\alpha)=\alpha^4+\alpha^3+\alpha^2=\alpha^3+\alpha^2+\alpha+1$	1110
α^{13}	$\alpha.\alpha^{12}=\alpha.(\alpha^3+\alpha^2+\alpha+1)$ $=\alpha^4+\alpha^3+\alpha^2+\alpha=\alpha+1+\alpha^3+\alpha^2+\alpha=\alpha^3+\alpha^2+1$	1101
α^{14}	$\alpha.\alpha^{13}=\alpha.(\alpha^3+\alpha^2+1)$ $=\alpha^4+\alpha^3+\alpha=\alpha+1+\alpha^3+\alpha=\alpha^3+2.\alpha+1=\alpha^3+1$	1001

Tableau A.3- Représentations des éléments du corps de Galois $GF(16)$

A.4 Polynôme minimal à coefficients dans $GF(2)$ associé à un élément d'un corps de Galois $GF(q)$

Le polynôme minimal $m_\beta(x)$ à coefficient dans $GF(2)$ associé à un élément quelconque β d'un corps de Galois $GF(q)$, est un polynôme de degré au plus égal à

$m=\log_2(q)$, ayant β comme racine :

$$\begin{aligned} m_\beta(x) &= \prod_{i=1}^{m}\left(x+\beta^{2^{(i-1)}}\right) \\ m_\beta(x) &= (x+\beta)\times\left(x+\beta^2\right)\times\left(x+\beta^4\right)\times\ldots.\times\left(x+\beta^{2^{(m-1)}}\right) \end{aligned} \tag{A.7}$$

Exemple 2

La détermination du polynôme minimal associé à $\beta=\alpha$ dans $GF(16)$, se fait comme suit :

$$m=4 \Rightarrow\ m_\beta(x)=(x+\beta).\left(x+\beta^2\right).\left(x+\beta^4\right).\left(x+\beta^8\right) \tag{A.8}$$

On remplace β par α, $m_\beta(x)$ devient :

$$\begin{aligned} m_\beta(x) &= \left[(x+\alpha).\left(x+\alpha^2\right)\right].\left[\left(x+\alpha^4\right).\left(x+\alpha^8\right)\right] \\ m_\beta(x) &= \left[x^2+\alpha^2x+\alpha.x+\alpha^3\right].\left[x^2+\alpha^8x+\alpha^4.x+\alpha^{12}\right] \\ m_\beta(x) &= \left[x^2+\left(\alpha^2+\alpha\right).x+\alpha^3\right].\left[x^2+\left(\alpha^8+\alpha^4\right).x+\alpha^{12}\right] \end{aligned} \tag{A.8}$$

Avec : $\begin{aligned} \alpha^2+\alpha &= \alpha(\alpha+1)=\alpha.\alpha^4=\alpha^5 \\ \alpha^8+\alpha^4 &= \alpha^4\left(\alpha^4+1\right)=\alpha^4.\alpha=\alpha^5 \end{aligned}$ (polynôme primitif $\alpha^4=\alpha+1$)

L'équation (A.8) devient :

$$\begin{aligned} m_\beta(x) &= \left[x^2+\alpha^5.x+\alpha^3\right].\left[x^2+\alpha^5.x+\alpha^{12}\right] \\ &= x^4+\left(\alpha^5+\alpha^5\right).x^3+\left(\alpha^{12}+\alpha^{10}+\alpha^3\right).x^2+\left(\alpha^{17}+\alpha^8\right).x+\alpha^{15} \end{aligned} \tag{A.9}$$

Avec $\begin{aligned} \alpha^5+\alpha^5 &= 0 \\ \alpha^{17} &= \alpha^2.\alpha^{15}=\alpha^2.1=\alpha^2 \end{aligned}$; où $\alpha^{q-1}=\alpha^{16-1}=\alpha^{15}=1$ (voir l'équation A.1)

L'équation (A.9) devient :

$$m_\beta(x)=x^4+\left(\alpha^{12}+\alpha^{10}+\alpha^3\right).x^2+\left(\alpha^2+\alpha^8\right).x+1 \tag{A.10}$$

Pour faire la sommation $\alpha^{12}+\alpha^{10}+\alpha^3$ et $\alpha^2+\alpha^8$: on remplace chaque élément par sa représentation binaire et on fait la sommation.

α^{12} : 1111
+
α^{10} : 0111
+
α^{3} : 1000
= 0000

et

α^{2} : 0100
+
α^{8} : 0101
= 0001

La représentation binaire 0000 correspond à l'élément du corps 0 et 0001 correspond à l'élément 1. Donc, le polynôme minimal associé à $\beta = \alpha$ dans $GF(16)$ est $m_{\beta}(x) = x^4 + x + 1$

On remarque que le polynôme minimale associé à α est le polynôme primitif. De la même manière on calcule le polynôme minimale associé à α^3 ou α^4 …

Annexe B

Démonstration de la capacité du canal Gaussien $C = Blog\left(1+\frac{S}{N}\right)$

Pour un canal gaussien d'une bande *B* perturbée par un bruit de puissance *N* où les signaux transmis sont limités à une certaine puissance *S*, alors les signaux reçus ont une puissance moyenne *S*+*N*. La capacité *C* est la plus grande quantité d'information moyenne qu'il peut transmettre [22] :

$$C = H(y) - H(b) \tag{B.1}$$

Où $H(y), H(b)$ représentent respectivement l'entropie par seconde (l'information moyenne) de la sortie et celle du bruit.

Avec $H = 2BH'$ où *H'* représente l'entropie par échantillon, l'équation (B.1) devient égale à :

$$C = 2B\left(H'(y) - H'(b)\right) \tag{B.2}$$

L'entropie par échantillon de la sortie et celle du bruit est donnée respectivement par :

$$H'(y) = -\lim_{n\to\infty}\frac{1}{n}\int\ldots\int p(y_1,\ldots,y_n)\log p(y_1,\ldots,y_n)dy_1\ldots dy_n \tag{B.3}$$

$$H'(b) = -\lim_{n\to\infty}\frac{1}{n}\int\ldots\int p(b_1,\ldots,b_n)\log p(b_1,\ldots,b_n)db_1\ldots db_n \tag{B.4}$$

Où $p(y_1,\ldots,y_n)$ et $p(b_1,\ldots,b_n)$ sont respectivement la densité du signal reçu et celle du bruit.

- La densité du signal reçu sera déterminée par :

$$p(y_1,\ldots,y_n)=p(y)=\frac{1}{(2\pi(S+N))^{n/2}}\exp-\frac{1}{2(S+N)}\sum_{i=1}^{n}y_i^2 \tag{B.5}$$

et

$$-\log P(y_1,\ldots,y_n)=\frac{n}{2}\log 2\pi(S+N)+\frac{1}{2(S+N)}\sum_{i=1}^{n}y_i^2 \tag{B.6}$$

- Et celle du bruit par :

$$p(b_1,\ldots,b_n)=p(b)=\frac{1}{(2\pi N)^{n/2}}\exp-\frac{1}{2N}\sum_{i=1}^{n}b_i^2 \tag{B.7}$$

et

$$-\log P(b_1,\ldots,b_n)=\frac{n}{2}\log 2\pi N+\frac{1}{2N}\sum_{i=1}^{n}b_i^2 \tag{B.8}$$

- Pour le calcul de l'entropie (par seconde) du bruit, en remplaçant (B.8) dans (B.4), on obtient :

$$H'(b)=\lim_{n\to\infty}\frac{1}{n}\int\ldots\int\frac{n}{2}p(b_1,\ldots,b_n)\log 2\pi N+\lim_{n\to\infty}\frac{1}{n}\int\ldots\int\frac{1}{2N}p(b_1,\ldots,b_n)\sum_{i=1}^{n}b_i^2 \tag{B.9}$$

$$H'(b)=\frac{1}{2}\log 2\pi N+\frac{N}{2N} \tag{B.10}$$

$$H'(b)=\frac{1}{2}\log 2\pi N+\log\sqrt{e} \tag{B.11}$$

Où e représente l'exponentielle et $\log e=1$.

$$H'(b)=\log\sqrt{2\pi eN} \tag{B.12}$$

Puis, en remplaçant (B.12) dans l'équation $H=2BH'$, on obtient :

$$H(b) = B \log 2\pi eN \tag{B.13}$$

- L'entropie (par seconde) de l'ensemble reçu est calculée de la même façon :

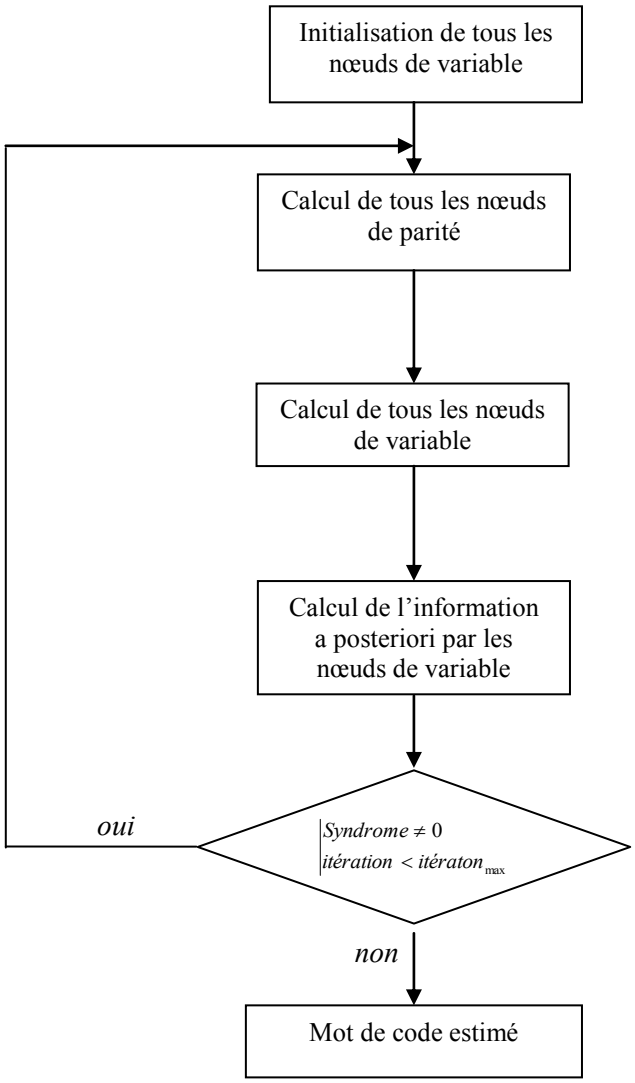
$$H = B \log 2\pi e(S + N) \tag{B.14}$$

La capacité du canal est donc égale à :

$$C = H(y) - H(b) = B \log\left(1 + \frac{S}{N}\right) \tag{B.15}$$

Annexe C

Organigramme de l'Algorithme de Propagation de Message



Bibliographie

[1] Y. Jiang, “A Practical Guide to Error-Control Coding using MATLAB”, Artech House, 2010.

[2] D. Manetti, “Contributions au décodage des codes convolutifs utilisés dans les systèmes de communication mobile UMTS,” Thèse de Doctorat de l'Université de Neuchâtel, Institut De Microtechnique, Septembre 2005.

[3] W. E. Ryan and S. Lin, “Channel Codes: Classical and Modern”, Cambridge University, 2009.

[4] Y. S. Cho, J. Kim, W. Y. Yang and C. G. Kang, “MIMO-OFDM Wireless Communications with Matlab”, John Wiley & Sons (Asia) Pte Ltd, 2010.

[5] T. S. Rappaport, “Wireless Communications: Principles and Practice”, Second Edition, 2002.

[6] J. G. Proakis, “Digital Communications”, Fourth Edition, 2001.

[7] M. Weeks, “Digital Signal Processing using MATLAB”, Infinity Science Press LLC, 2006.

[8] Marco Baldi, “QC-LDPC Code-Based Cryptography”, Springer, 2014.

[9] J. C. Moreira and P. G. Farrell, “Essentials of Error-Control Coding”, John Wiley & Sons Ltd, 2006.

[10] C. Berrou, “Codes et Turbo-Codes”, Springer, 2007.

[11] T. K. Moon, “Error Correction Coding Mathematical Methods and Algorithms”, A John Wiley & Sons, Inc, 2005.

[12] R. Liu, P. Spasojevié et E. Soljanin, “Punctured Turbo-Code Ensembles”, Information Theory Workshop, Proceedings, IEEE, pp. 249 – 252, Paris, France, du 31 Mars au 4 Avril 2003.

[13] S. Le Goff et C. Berrou, "Les turbo-codes de Rendements Elevés", Quatorzième Colloque Gretsi-juan-les-pins- du 18 au 21 Septembre 1993.

[14] C. Douillard et C. Berrou, "Turbo-Codes with rate-*m*/(*m*+1) Constituent Convolutional Codes", IEEE Transactions on Communications, vol. 53, n° 10, pp. 1630-1638, Octobre 2005.

[15] S. Benedetto et G. Montorsi, "Design of Parallel Concatenated Convolutional Codes", IEEE Transactions on Communications, vol. 44, n° 5, pp 191-600, may 1996.

[16] S. Benedetto, G. Montorsi, D. Divsalar et F. Pollara "Serial Concatenation of Interleaved Code: Performance, Analysis, Design and Iterative Decoding", IEEE Transactions on Information Theory, vol. 44, n° 3, pp. 909-926, Mai 1998.

[17] S. Benedetto et G. Montorsi, "Generalized Concatenated Codes with Interleavers", International Symposium on Turbo-Codes and Related Topics, pp. 32-39, Brest, France, Septembre 1997.

[18] C. Berrou, A. Glavieux et P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo-Codes", IEEE ICC'93, vol. 2/3, pp. 1064-1070, Geneva, Mai 1993.

[19] J. J. Boutros, "Les Turbo-Codes Parallèles et Séries", Polycopié de cours, Octobre 1998.

[20] S. Al Muaini, A. AlDweik et M. AlQutayri, "BER Performance of Turbo Product LDPC Codes with Non Sequential Decoding", Wireless and Mobile Networking Conference, pp. 1-6, 2013.

[21] jui-hui hung, jin-shun shyu and sau-gee chen, "A New High-Performance and Low-complexity Turbo-LDPC Code", International Conference on Multimedia and Signal Processing, pp. 68-72, 2011.

[22] C. E. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423 et 623-656, Juillet et Octobre 1948.

[23] R. Gallager, "Low-Density Parity-Check Codes", IEEE Transaction Information Theory, vol. 8, no. 1, pp. 21–28, 1962.

[24] D. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices", IEEE Transaction Information Theory, vol. 45, no. 2, pp. 399–431, Mars 1999.

[25] M. Davey et D. MacKay, "Low-Density Parity Check Codes over GF(*q*)", IEEE Communications Letters, vol. 2, no. 6, pp. 165–167, 2002.

[26] HAN Jian-Bing, HE Chen et HE He Yun, "Research on Regular LDPC Codes with better Performance than Turbo Codes," International Conference *on* Information Engineering ICIE ′09. WASE, 2009.

[27] Su-Chang Chae and Yun-Ok Park, "Low Complexity Encoding of Improved Regular LDPC Codes," Vehicular Technology Conference IEEE, 2004.

[28] V. Mannoni, "Optimisation des Codes LDPC pour les Communications Multi-Porteuses," Thèse de Doctorat de l'Université de Cergy-Pontoise, Juin 2004

[29] P. Ma et D. Yuan, "Irregular LDPC Coded BICM in Image Transmission over Rayleigh Fading Channel," Consumer Communication and Networking Conference, 2005.

[30] S.J. Johnson, S.R. Weller, "Regular Low-Density Parity-Check Codes from Combinatorial Designs", IEEE Information Theory Workshop (ITW'2001), Cairns 2001.

[31] Tanner RM, "A Recursive Approach to Low Complexity Codes," IEEE Transactions on Information Theory, IT-27(5): 533–547, 1981

[32] F. R. Kschischang, B. J. Frey, H. andrea Loeliger, "Factor Graphs and the Sum-Product Algorithm", IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 498–519, Février 2001.

[33] S. Johnson, "Iterative Error Correction Turbo, Low-Density Parity-Check And Repeat-Accumulate Codes," Cambridge University Press, 2010.

[34] L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF(2*q)*", in in Proceeding IEEE Information Theory Workshop, pp. 70-73, 2003.

[35] H. Wymeersch, H. Steendam and M. Moeneclaey, "Log-Domain Decoding of LDPC Codes over *GF*(*q*)," in Proceeding IEEE International Conference Communication, pp. 772-776, Paris, France, Juin. 2004.

[36] C. Spagnol, E. Popovici, and W. Marnane, "Hardware Implementation of *GF*(*2m*) LDPC Decoders," IEEE Transactions Circuits Systems I, Reg. Papers, vol. 56, no. 12, pp. 2609-2620, Décembre 2009.

[37] D. Declercq and M. Fossorier, "Decoding Algorithms for Non binary LDPC Codes over GF(*q*)," IEEE Transactions on Commuications, vol. 55(4), pp. 633-643, Avril 2007.

[38] A. Voicila, D. Declercq, F. Verdier, M. Fossorier and P. Urard, "Low-Complexity Decoding for Non-Binary LDPC Codes in High Order Fields," IEEE Transactions on Communications, vol. 58(5), pp. 1365-1375, Mai 2010.

[39] Valentin Savin, "Min-Max Decoding for Non Binary LDPC Codes," in Proceeding IEEE International Symposium. Information Theory, pp. 960-964, Juillet 2008.

[40] R. A. Carrasco and M. Johnston, "Non-Binary Error Control Coding for Wireless Communication And Data Storage", Wiley & Sons (Asia) Pte Ltd, 2008.

[41] L. Safarnejad and M. Sadeghi, "FFT Based Sum-Product Algorithm for Decoding LDPC Lattices," IEEE Communications Letters, vol. 16, no. 9, Septembre 2012.

[42] S. J. Johnson, "Low-Density Parity-Check Codes: Design and Decoding," Chapter in Wiley Encyclopedia of Telecommunications, 2002.

[43] J. Chen. A.Dholakia, E. Eleftheriou, Marc P. C. Fossorier, Xiao-Yu Hu, "Reduced-Complexity Decoding of LDPC Codes," IEEE Transactions on Communications, vol. 53, no. 8, pp. 1288-1299, Août 2005.

[44] Q. Wang, Q. Xie, Z.Wang, S. Chen and L. Hanzo, "A Universal Low-Complexity Symbol-to-Bit Soft Demapper", IEEE Transactions on Vehicular Technology, pp. 119-130, vol. 63, no. 1, Janvier 2014.

[45] Digital Video Broadcasting (DVB); Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2), ETSI EN Std. 302 755 V1.3.1, Avril. 2012.

[46] Digital Video Broadcasting (DVB); Frame Structure Channel Coding and Modulation for a Second Generation Digital Transmission System for Cable Systems (DVB-C2), ETSI EN Std. 302 769 V1.2.1, Avril 2012.

[47] Unified High-Speed Wireline-Based Home Networking Transceivers System Architecture and Physical Layer Specification ITU-T Std. G.9960, Dec. 2011.

[48] Very High Speed Digital Subscriber Line Transceivers 2 (VDSL2) ITU-T Std. G.993.2, Dec. 2011.

[49] J. Tan, Q. Wang, C. Qian, Z. Wang, S. Chen and L. Hanzo, "A Reduced-Complexity Demapping Algorithm for BICM-ID Systems", IEEE Transactions on Vehicular Technology, 31 Octobre, 2014.

[50] Md.Z. Alam, C. S. Islam et M. A. Sobhan, "Using Log Likelihood Relation for BER Analysis of QAM in Space Diversity", journal of communications, vol. 4, no. 6, pp.371-379, Juillet 2009.

[51] F. Tosato and P. Bisaglia, "Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HIPERLAN/2", in IEEE International Conference Communication, vol. 2, pp. 664–668 7 Août, 2002.

[52] K. Hyun and D. Yoon, "Bit Metric Generation for Gray Coded QAM Signals", Proc. Inst. Elect. Eng., Commun., vol. 152, no. 6, pp. 1134–1138, Dec. 2005.

[53] L. Wang, D. Xu, and X. Zhang, "Low Complexity Bit Metric Generation for PAM Signals Based on Non-Linear Function", Electron. Lett., vol. 17, pp. 966–967,18 Août, 2011.

[54] S. Le Goff, C. Berrou et A. Glavieux, "Turbo-Codes and High Spectral Efficiency Modulations", IEEE International Conference on Communications (ICC'94), vol. 2/3, pp.645-649, New Orleans, Mai 1994.

[55] S. Y. Le Goff, "Bit-Interleaved Turbo-Coded Modulations for Mobile Communications", European Signal Processing Conference (EUSIPCO), Finlande, 2000.

[56] Xiang Liu and Martin Kosakowski, "Max-Log-MAP Soft Demapper with Logarithmic Complexity for M-PAM Signals", IEEE Signal Processing Letters, pp. 50-53, vol. 22, no. 1, Janvier 2015.

[57] R. Ghaffar and R. Knopp, "Low Complexity Metrics for BICM SISO and MIMO Systems," in 2010 IEEE 71st Vehicular Technology Conf. (VTC 2010-Spring), pp. 1–6, 6–19 Mai, 2010.

[58] Sang Hoon Lee, Ji Ae Seok, et Eon Kyeong Joo, "Serially Concatenated LDPC and Turbo Code with Global Iteration," Proceedings of the 18th International Conference on Systems Engineering (ISCEng'05), 2005.

[59] E. Zehevi, "8-PSK trellis codes for a Raleigh Channel," IEEE Transaction on Communications, vol. 40, n° 05, pp. 873-884, Mai 1992.

[60] S. Le Goff, "Les Turbo-Codes et leurs Applications aux Transmissions à Forte Efficacité Spectrale," Thèse de Doctorat de l'Université de Bretagne Occidentale, Brest, Novembre 1995.

[61] K. Wu, Hui Li, Yumin Wang, "The influence of Interleaver on the Minimum Distance of Turbo-Code," International Conference on Communication Technology, 22-24 Octobre, Beijing, China, 1998.

[62] E. Boutillon, C. Douillard et G. Montorsi, "Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues," Proceeding of IEEE, vol. 95, no. 6, Juin 2007.

[63] D. Divsalar et F. Pollara, "Multiple Turbo-Codes for Deep-Space Communications," Jet Propulsion Lab., Passadena, CA TDA Progress Report 42-121, Mai 1995.

[64] P. Hoeher, "New Iterative "Turbo" Decoding Algorithms," International Symposium on Turbo-Codes, pp. 63-70, Brest, France, 1997.

[65] S. Benedetto, G. Montorsi, D. Divsalar et F. Pollara, "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo-Codes," JPL TDA Progress Report 42-124, 15 Février, 1996.

[66] S. Rekh et S. S. rani et A. Shanmugam, "Optimal Choice of Interleaver for Turbo-Codes," Academic Open Internet Journal, vol. 15, 2005.

[67] H. Behairy and S. C. Chang, "Parallel Concatenated Gallager Codes", Electronics Letters, vol. 36, pp. 2025-2026, 2000.

[68] H. M. Behairy et M. Benaissa, "Multiple Parallel Concatenated Gallager Codes: Code Design and Decoding Techniques," IETE Journal of research, vol. 59, 2014.

[69] Z. Wang et M. Zhang, "A Serial Concatenated Scheme for LDPC Code to Achieve Better Error Correction Performance", 2nd International Conference on Cosumer Electronics, Communications and Networks, pp. 1587-1589, 21-23 Avril 2012.

[70] B. Belgheit, A. Boukelif, A. Moulay Lakhder and M. Kamline, "Parallel Concatenated Gallager Codes Matrix and the Effect of Interleaver," International Journal of Electronics, vol. 99, no. 9, pp. 1281-1289, 9 Septembre, 2012.

[71] H. Behairy and S. C. Chang, "Parallel Concatenated Gallager Codes for CDMA Applications," Global Telecommunications Conference GLOBECOM'01. IEEE, vol. 2, 2001.

[72] P. Elias, "Coding for noisy channels", *IRE conv. Rec.*, vol. 3, pt. 4, pp. 37-46, 1955.

[73] M. Beermann, E. Monz, L. Schmalen, and P. Varyx, "High Speed Decoding of Non-Binary Irregular LDPC Codes Using GPUs," in Proc. IEEE SiPS, 2013.

[74] B. Moision, "Decoding Complexity and Performance of Short-Block LDPC Codes Over GF (q) ," IPN Progress Report 42-194, August 15, 2013.

[75] R. Pyndiah, "Near Optimum Decoding of Product Codes: Block Turbo-Codes", IEEE Transactions on Communications, vol. 46, no. 8, pp. 1003-1010, Août 1998.

[76] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Department of Electrical Engineering, M.I.T., Cambridge, Mass., Juillet 1963.

[77] D. J. C. MacKay, R. M. Neal, "Good codes based on very sparse matrices", in Cryptography and Coding, 5th IMA Conference, Décembre 1995.

[78] M. Sipser et D. A. Spieleman, "Expender Codes", IRE Transaction on Information Theory, vol. 42, no. 06,pp. 1710-1722, Novembre 1996.

[79] M. G. Luby, M. Mitzenmachery, M. A. Shokrollahiz, D. A. Spielmanx et V. Stemann, "Practical Loss-Resilient Codes," STOC'97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, pp. 150-159, 1979.

[80] R. P. Kumar and R. S. Kshetrimayum, “An Efficient Methodology for Parallel Concatenation of LDPC Codes with Reduced Complexity and Decoding Delay,” in Proceding National Conference on Communications (NCC), New Delhi, India, Février 2013.