

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département d'Electronique



Projet de Fin d'Etude

Mémoire en vue de l'obtention du diplôme d'Ingénieur en Electronique

Thème :

Codage de canal : Application des Turbocodes dans un canal satellitaire

Réalisé par :

Mr Mehdi HAFSSIL

Mr Salah BOUHOUN

Encadré par :

Pr. D. BERKANI (*Promoteur, l'ENP*)

Pr. K. GHANEM (*Co-promotrice, CDTA*)

Juin 2013

E.N.P- 10, Avenue Hassen Badi, 16200 El Harrach, Alger

ملخص:

هذا العمل هو دراسة حول الترميز توربو (Turbo-codes) وتطبيقه على قناة اتصال مع قمر اصطناعي. بعد تقديم سريع حول نظرية الاتصال وترميز القناة، تطرقنا إلى عرض حول الاتصالات الفضائية. بعد ذلك مررنا إلى الهدف من هذه المذكرة وهو فك الترميز المعتمد على العملية التكرارية للترميز المسمى توربو (Turbo-codes) خوارزمية MAP التي تستعمل في فك الترميز توربو تتسم بتعقيد كبير جدا. استعملنا في عملنا هذا أحد الفروع المبسطة للخوارزمية MAP المسماة log-MAP. أحد النقص في هذه الخوارزمية هو وجوب معرفة تباين الضجيج القناة في هذا العمل نقوم بالتحليل والتقييم مستعملين محاكاة بالحاسوب لاختيار بعض الإعدادات التي تؤثر في جودة مفكك الترميز توربو الذي يعمل بخوارزمية log-MAP. استخدمنا لذلك عدة أنظمة ذات عقبات مختلفة الأطوال، ورسائل ذات أطوال مختلفة وتبديلات متنوعة. قمنا بنمذجة قناة القمر الصناعي بقناة ذات ضجيج أبيض جوس مضاف (AWGN). في عملنا استعملنا فقط التشكيل BPSK.

الكلمات المفتاحية: ترميز القناة، الترميز توربو، سعة القناة، خوارزمية log-MAP، ترتيب.

Résumé : Ce travail est une étude qui porte sur les Turbo-codes et leurs applications sur un canal satellite. Après une brève introduction sur la théorie de l'information et le codage de canal, une présentation sur les communications satellitaire. A partir de la, nous allons passer au but de notre mémoire qui porte sur le décodage itératif des Codes dits Turbo.

L'algorithme MAP qui est utilisé pour le décodage des Codes Turbo souffre d'une très grande complexité. Une variante simplifiée du MAP, qui porte le nom du Log-MAP est utilisée dans notre travail. L'inconvénient de l'utilisation du MAP est la nécessité de l'estimation de la variance du bruit dans le canal. L'aspect de ce travail est d'analyser et d'évaluer par simulation sur ordinateur pour le choix de certaines paramètres qui agissent sur les performances du décodeur Turbo utilisant l'algorithme Log-MAP pour différentes longueurs de contrainte, un taux de codage $R_t = 1/3$, et pour des blocs de petite à moyenne taille avec différent type de permutation. Le canal satellitaire est modélisé par un canal à bruit blanc additif et gaussien. Dans nos simulations, seule la modulation cohérente BPSK est considérée.

Mots clés : Codage de canal, Turbocode, capacité du canal, Concaténation, Permutation, Décodage itératif, Modulation, l'algorithme Log-MAP.

abstract : This work is a study that focuses on the Turbo-codes and their applications on a satellite channel. After a brief introduction to information theory and channel coding and a presentation about satellite communications, we'll go to the purpose of our memory that focuses on iterative decoding of Turbo Codes said.

The MAP algorithm is used for decoding the Turbo codes suffers from high complexity. A simplified version of the MAP, which named Log-MAP is used in our work. The disadvantage of the use of MAP is the need to estimate the variance of the noise in the channel. Medskip The aspect of this work is to analyze and evaluate by computer simulation for the choice of certain parameters witch influence on the performance of Turbo decoder using the Log-MAP algorithm. The analysis and the evaluation are done for different constraint lengths, a coding rate $R_t = 1/3$ and for blocks of small to medium size with different functions of Interleaving. The satellite channel is modeled by a channel additive white Gaussian noise. In our simulations, only the coherent BPSK modulation is considered. Medskip

Key words : Channel coding, Turbocode, channel capacity, Concatenation, interleaving, iteratif Decoding, Modulation, Log-MAP algorithm.

Dédicaces

A toi mon PERE, si loin de moi depuis mon voyage dans l'au-delà et si près de moi par la pensée, TU DEMEURES A JAMAIS AVEC MOI

A toi ma MERE, si loin de moi depuis ton lieu de résidence et si près de moi par la pensée, TU ME MANQUES

A mes soeurs

A tous mes proches

A Monsieur : D. BERKANI

A Madame : K. GHNEM

A mes amis : Salima, Islem,...

A tous mes amis de l'ENP.

A tous ceux qui ont su croire en moi

A tous ceux qui me sont chers

Je dédie ce modeste travail

Mehdi HAFSSIL

Dédicaces

A mes très chers parents

A mes frères et soeur

A mes oncles et tantes

A tous mes proches

A Monsieur : D. BERKANI

A Madame : K. GHNEM

A mes amis : Amine, Ahmed, Abdennour ...

A tous mes amis de l'ENP.

A tous ceux qui ont su croire en moi

A tous ceux qui me sont chers

Je dédie ce modeste travail

Salah BOUHOUN

Remerciement

Nous tenons à remercier Dieu le tout Puissant et Maître de l'univers, qui nous a donné la force nécessaire, la forte volonté et la patience, afin d'accomplir ce travail.

Tout d'abord, Nous remercions particulièrement et très chaleureusement notre promoteur, Pr D. BERKANI, avec qui nous avons eu tant de plaisir à travailler et de profiter de son expérience, ses conseils précieux, ses remarques pertinentes.

Nous tenons également à exprimer notre profonde reconnaissance au Pr K. GHANEM pour tous ses efforts consentis pour la réussite de notre stage au Centre du Développement des Technologies Avancées.

Nous exprimons notre gratitude et nos remerciements sincères envers le Pr Z. TERRA qui nous a fait l'honneur de présider le jury. Nous remercions aussi Dr A. MOUSSAOUI d'avoir accepté d'être membre du jury et d'examiner notre travail.

Que tous nos professeurs qui ont contribué à notre formation trouvent ici notre plus profonde gratitude.

Enfin, Nous souhaitons dédier ce mémoire à nos parents. Rien n'aurait été possible sans leur soutien, confiance et générosité.

Table des matières

1	Introduction au codage de canal	3
1.1	Introduction	3
1.2	Généralités	4
1.2.1	Chaîne de transmission	4
1.2.2	Le codage de source	4
1.2.3	Le codage de canal	5
1.2.4	La modulation	5
1.2.5	Le canal de communication	9
1.2.6	Information mutuelle et capacité d'un canal	11
1.2.7	Le théorème fondamental du codage de canal	11
1.3	Les codes correcteurs d'erreurs	12
1.3.1	Mesure des performances d'un code correcteur d'erreurs	12
1.4	Conclusion	14
2	Introduction aux communications satellitaires	16
2.1	Introduction	16
2.2	Principe des communications satellitaires	16
2.2.1	Le secteur spatial	16
2.2.2	Secteur terrien	18
2.3	Gammes de fréquences utilisées par les satellites	18
2.4	Services de télécommunications par satellite	19
2.4.1	La téléphonie	19

2.4.2	La télédiffusion	20
2.4.3	La transmission de données	20
2.5	Structure du système d'émission	21
2.5.1	Filtre Cosinus surélevé	21
2.5.2	Amplificateur Haute Puissance	21
2.5.3	Antenne d'émission	22
2.6	Structure du système de réception	23
2.6.1	Antenne de réception	23
2.6.2	Amplificateur à faible bruit(Préamplificateur)	23
2.6.3	Filtre adapté	23
2.7	Les pertes	24
2.7.1	Propagation en espace libre	24
2.7.2	influence du troposphère	24
2.7.3	Atténuation dûe à la pluie	24
2.7.4	L'évanouissement multi trajets (Multipath fading)	24
2.7.5	L'ombrage (Shadowing)	25
2.7.6	Effet Doppler	26
2.8	Bilan de liaison	26
3	Généralités sur les codes convolutifs	30
3.1	Introduction	30
3.2	Les codes en bloc	30
3.2.1	les codes en blocs linéaires	31
3.2.2	les codes en blocs linéaires à symboles binaires	31
3.3	Codage convolutionnel	31
3.3.1	Codeur convolutionnel non récursif (RFI)	31
3.3.2	Codeur convolutionnel récursif (RII)	32
3.3.3	Représentation polynômiale des codes convolutionnels	33
3.3.4	Représentations graphiques des codeurs convolutifs	33
3.4	Distance libre	36

3.5	Décodage des codes convolutif	37
3.5.1	Algorithme de Viterbi	37
3.5.2	Simulation	40
3.6	Conclusion	40
4	Les turbo codes	43
4.1	Introduction	43
4.2	Les permutations	44
4.2.1	l'entrelacement par bloc	44
4.2.2	l'entrelacement par convolution	45
4.3	La concaténation	46
4.3.1	Concaténation série (CS)	46
4.3.2	La concaténation parallèle (CP)	47
4.4	Fermeture du treillis	48
4.4.1	A l'aide de bits de bourrage	49
4.4.2	Un entrelacement permettant une fermeture auto- matique du treillis	49
4.4.3	Un codage circulaire	49
4.5	Turbo codage	50
4.6	Turbo décodage	50
4.6.1	Principe de Turbo Décodage	50
4.6.2	Logarithme de rapport de vraisemblance	52
4.6.3	L'algorithme de maximum probabilité à posteriori	53
4.6.4	L'algorithme SOVA	57
4.7	Conclusion	59
5	Simulations et résultats	61
5.1	Conditions et paramètres de la simulation	61
5.1.1	Effet de nombre d'itération sur la performance du Turbocode	62
5.1.2	Effet de la taille de bloc	63

5.1.3	Effet du polynôme générateur	63
5.1.4	Effet du type de permutation	65
6	Conclusions	69
A	Notions sur les corps de Galois	71

Table des figures

1.1	Système de communication spatiale [1]	4
1.2	Forme d'un signal modulé BPSK	7
1.3	Comparaison du BER de la modulation BPSK avec différents types de modulation	8
1.4	Comparaison du BER de la modulation BPSK avec différents ordre de modulation	8
1.5	Constellation du signal BPSK	9
1.6	Modulateur BPSK [2]	9
1.7	Graphe de transition dans un (CBS)	10
2.1	Composantes d'un système de télécommunications par satellite	17
2.2	Architecture d'un système de diffusion par satellite	19
2.3	La caractéristique d'amplificateur $OBO = f(IBO)$	22
2.4	Bilan de puissance d'une liaison satellitaire	29
3.1	Codeur RFI non systématique de matrice génératrice $G(D) = [1 + D^2, 1 + D + D^2]$	32
3.2	Codeur RII systématique de matrice génératrice $G(D) = [1, 1 + D + D^2/1 + D^2]$	32
3.3	Diagramme d'état du codeur RIF $G(D) = [1 + D^2, 1 + D + D^2]$	34
3.4	Représentation d'un codeur sous forme d'arbre [3]	35
3.5	Représentation d'un codeur sous forme de treillis [3]	36
3.6	Codeur convolutif non systématique $[(7, 5), (5)]$	37

3.7	Comparaison des performances entre les codes correcteurs les plus utiliser	40
4.1	Principe de fonctionnement d'un entrelaceur par bloc [4]	45
4.2	(a)Entrelaceur par convolution, (b)Disentrelaceur par convolution [5]	46
4.3	Schéma bloc type de codes concaténés série [4]	47
4.4	Schéma bloc type de codes concaténés parallèle [4]	48
4.5	Exemple d'une sequence de transition avec fermeture par bits de bourrage [6]	50
4.6	Algorithme de décodage du Turbo code [6]	51
4.7	Principe des itérations et de calcul des L a posteriori	55
5.1	Algorithme de notre système de communication simulé	61
5.2	Effet de nombre d'itération sur la performance du Turbo-code N=512 bits	63
5.3	Effet de la taille de bloc sur la performance du Turbo-code	64
5.4	Effet du type de polynôme générateur sur la performance du Turbo-code	65
5.5	Effet de la longueur de contrainte de code sur la performance du Turbo-code pour $G1=(3,[7,5],5)$ et $G2=(7,[171,133],133)$	66
5.6	Effet de la longueur de contrainte de code sur la performance du Turbo-code pour $G1=(2,[2,3],3)$ et $G2=(7,[171,133],133)$	66
5.7	Effet de la longueur de contrainte de code sur la performance du Turbo-code	67
5.8	Effet de la longueur de contrainte de code sur la performance du Turbo-code	67
5.9	Effet de la longueur de contrainte de code sur la performance du Turbo-code	68

Liste des tableaux

1.1	Les applications des différentes méthode de modulation [7]	6
1.2	Limite théorique de l'efficacité spectrale de modulation [7]	7
2.1	gammes de fréquences utilisées dans les télécommunications	18
2.2	Les différents types de satellite de télécommunications . . .	20
2.3	Affaiblissement consécutif à la pluie	25
2.4	Les fréquences de travail montante et descendante selon le type de liaison	26
3.1	Les états internes d'un CR a 2 mémoires	38

Liste des abréviations

BER TEB	Taux d'erreur binaire
QEF	Transmission quasiment sans erreurs
PSK	Phase shift keying
BPSK	Binary phase shift keying
CBS	Canal binaire symétrique sans mémoire
AWGN BBAG	Bruit blanc additif gaussien
TTC	Tracking telemetry and command
VSAT	Very small aperture terminal
RCVO	Receive only
MEO	Medium earth orbit
LEO	Low earth orbit
GEO	Geostationary earth orbit
DVB-S	Digital video broadcasting satellite
DAB	Digital audio broadcasting
OBP	On board processing
OBO	Output back off
IBO	Input back off
PIRE	Puissance Isotrope Rayonnée Effective
RFI	Reponse finie à une impulsion
RII	Reponse infinie à une impulsion
CR	Code récursif
CRC	Code récursif systématique
NRNSC	Code non récursif non systématique
CCSDS	Consultative comited for space data systems
MAP	Maximum a posteriori
DMH	Distance minimal de Hamming
CS	Concatination série
CP	Concatination parallèle
RTZ	Retoure à zéro
SISO	Soft in soft out
LLR	Logarithm likelihood ratio
SOVA	Soft Output Viterbi Algorithm

Introduction Générale

En 1948, Claude Shannon a montré qu'il existe une limite au débit d'information transmis en présence de bruit, appelée la capacité du canal, mais il n'a pas explicité les moyens de l'approcher. Même si le caractère asymptotique de cette limite ne laisse aucun espoir de l'atteindre, la communauté de la théorie de l'information ne cesse d'effectuer des recherches afin de pouvoir atteindre cette limite. Après plus de 40 ans de recherche, où "des demi-vérités ont fini par devenir des dogmes", C. Berrou, A. Glavieux et P. Thitimajshima ont montré comment réussir à s'approcher de cette fameuse limite avec une complexité raisonnable, et cela, par l'application d'un principe analogue à celui de la contre-réaction bien connu en électronique. Ce principe, baptisé "Principe Turbo" par analogie aux moteurs en mécanique, consiste à réinjecter à l'entrée du système une partie de l'information de sortie. Cette découverte a permis d'ouvrir une grande course dans le domaine des codes correcteurs d'erreur et les appliquer dans plusieurs domaines de communication, dont les communications par satellite. Depuis la première tentative de communiquer en utilisant un satellite (Spoutnik) en 1957, les recherches ont connu un grand avancement pour pouvoir transmettre un message à travers la liaison satellite avec le moins d'erreur. Avec l'augmentation des services utilisant les communications satellite, et la nécessité d'un débit et des performances élevés, plusieurs techniques sont développées pour répondre à ces besoins, sachant que la liaison satellite provoque des dégradations très importantes au signal.

Organisation du document

Ce mémoire comporte 5 chapitres :

Le premier chapitre est consacré à l'introduction aux concepts généraux liés à la théorie de l'information et au codage de canal. Dans ce chapitre, on a défini chaque bloc de la chaîne de communication numérique, et quelques outils de mesure de performance.

Le deuxième chapitre est une introduction à la communication satellite. On va voir dans ce chapitre le principe de communication par satellite, les services de télécommunications par satellites, les fréquences utilisées. On a comptabilisé aussi les gains et les pertes pour établir le Bilan de liaison.

Le chapitre trois contient des généralités sur les codes convolutifs, et les principaux notions utilisées, codage et décodage.

Le chapitre quatre est une description du Turbo-code, les notions de permutations, la concaténation, et les différentes méthodes de fermeture de treillis, après on entame la partie de codage puis le décodage du Turbo-codes. Dans le décodage, l'algorithme Log-MAP est décrit, avec une bref description de l'algorithme SOVA.

Le cinquième chapitre contient les simulations et l'analyse des résultats. Utilisant un simulateur d'un système de communication numérique programmé en Matlab. Enfin, une conclusion générale sur ce travail

Chapitre 1

Introduction au codage de canal

1.1 Introduction

Le taux d'erreur binaire (TEB) est une donnée d'une très grande importance dans une chaîne de transmission numérique. Afin d'assurer une transmission fiable, le TEB doit rester en dessous d'un seuil donné qui est fonction de l'application. Les applications multimédia, en particulier, nécessitent une transmission quasiment sans erreurs (QEF).

D'une manière générale, le taux d'erreur binaire, dépend directement du rapport signal sur bruit, et plus précisément du rapport signal sur bruit par bit utile E_b/N_0 . La solution la plus naturelle qui permet d'améliorer la fiabilité de la transmission consiste alors à augmenter le niveau de la puissance transmise au niveau de l'émetteur. Cette solution n'est pas toujours applicable et se heurte souvent à plusieurs contraintes. D'une part, les réglementations imposées par les différents organismes de télécommunication (UIT par exemple) limitent toujours le niveau de la puissance maximale transmise par l'émetteur afin de limiter d'une part les interférences avec les autres utilisateurs et d'autre part avec les autres systèmes opérant à des fréquences voisines. Encore mieux, il se trouve dans certains cas, que la puissance disponible à l'émetteur soit limitée. C'est le cas d'un émetteur à bord d'un satellite mais aussi celui d'un terminal mobile.

On comprend alors qu'il va falloir trouver une solution qui permette une exploitation optimale de la puissance disponible. Cette solution est donnée par le codage canal. Cette technique consiste à associer à la séquence binaire transmise (aléatoire) une nouvelle séquence binaire corrélée d'une façon contrôlée. Cette nouvelle séquence, plus longue que la séquence d'origine, contient, bien entendu, toute l'information utile à transmettre. La corrélation introduite entre les différents bits codés offre une protection aux bits utiles contre les perturbations du canal de transmission.

Ce chapitre a pour objectif de présenter des principes et des concepts fondamentaux qui nous seront utiles pour la suite. Dans un premier temps, nous donnerons un exemple typique d'une chaîne de transmission et nous expliquerons le rôle de chaque bloc de

la chaîne. Après une petite introduction présentant les différentes catégories de codes correcteurs qui seront étudiés en détail dans le chapitre suivant, les méthodes de calcul visant à évaluer les performances d'un code correcteur, sont introduites.

1.2 Généralités

1.2.1 Chaîne de transmission

Une chaîne de transmission modélise les différentes étapes autorisant le transfert d'une information d'une source vers un destinataire. Une chaîne de transmission peut différer grandement selon la nature du système de transmission, qui peut aussi bien être un système de stockage de données qu'un système de télécommunication [8]. Néanmoins certaines étapes sont applicables dans tous les contextes (Figure(1.1)). Ainsi, le couple codeur-décodeur de source peut être employé pour réduire la taille du message à transmettre par une opération de compression-décompression réalisée avec ou sans perte d'information. A l'inverse, le couple codeur-décodeur de canal augmente la taille du message à transmettre afin d'accroître la qualité de l'information transmise sur le canal. Le canal, justement, comprend toutes les étapes entre la sortie du codeur et l'entrée du décodeur. Cela intègre le lieu physique de la transmission, dans lequel le signal transmis est bruité d'une façon aléatoire, et le couple modulateur-démodulateur, qui est utilisé dans la plupart des systèmes pour transformer l'information numérique en un signal continu compatible avec le milieu de transmission et vice-versa.

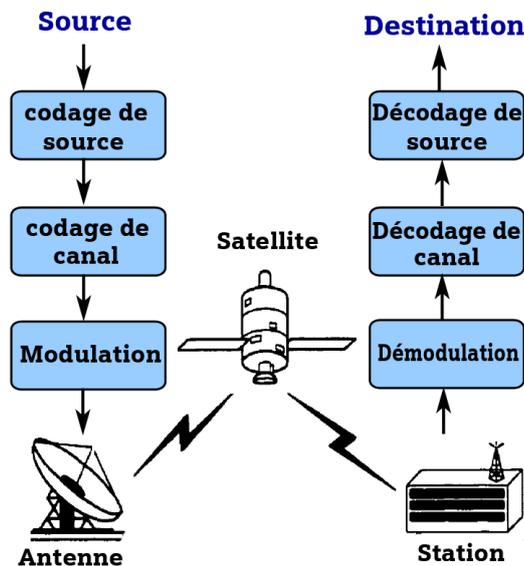


Figure 1.1 – Système de communication spatiale [1]

La suite de cette section illustre un peu plus en détail chacune des étapes d'une chaîne de transmission en se contraignant au cas de la chaîne de communication sans fil présentée dans la figure Figure(1.1).

1.2.2 Le codage de source

Le codage de source vise à la concision maximale de l'information. L'usage d'un canal coûte d'autant plus cher que le message est long, le verbe "coûter" devant s'étendre ici en un sens très général, celui d'exiger l'emploi de ressources limitées telles que le temps, la puissance et la bande passante [9]. Pour diminuer ce coût, on cherche à représenter le message avec le moins de bits possibles, c'est-à-dire le compresser. Pour ce faire, on essaye d'éliminer la redondance contenue dans le message transmis par la source.

Un point essentiel dans le codage de source est le critère de fidélité. Ce critère varie selon l'application. Les applications où la compression de données doit se faire sans perte, utilisent un codage de source appelé *réversible*¹, tandis que, les applications où les pertes sont tolérables, utilisent un codage dit *non-réversible*².

1.2.3 Le codage de canal

L'objectif du codage de canal est de protéger le message contre les perturbations du canal, et cela, en introduisant une redondance à l'information utile dans le message à transmettre. La redondance et l'information utile sont liées par une loi donnée. A la réception, le décodeur de canal exploite la redondance produite par le codeur dans le but de détecter, puis de corriger lorsque possible, les erreurs introduites lors de la transmission [9]. Ce point sera détaillé encore plus dans les sections suivantes.

1.2.4 La modulation

Afin de transmettre le message au travers du milieu de transmission, le modulateur génère un signal porteur, dont la forme d'onde peut-être soit une suite d'impulsions soit une onde sinusoïdale. Dans le cas de la modulation numérique, le modulateur transpose chaque ensemble de m bits du message entrant dans le modulateur au débit binaire D_b en un signal physique de durée $T = \frac{m \cdot R}{D_b}$. Les 2^m signaux physiques possibles forment ce que l'on appelle la constellation de la modulation. Dans le cas d'une onde sinusoïdale, le message peut être porté par la phase, l'amplitude ou/et la fréquence (OOK, PSK, FSK, CP FSK, QAM, ...).

1. Réversible : le message envoyé sera exactement restitué au niveau du récepteur (sans perte ou distorsion).

2. Par exemple, dans la norme MPEG4, la compression des données multimédia est faite avec une distorsion tolérable par l'observateur (critère de fidélité), et cela à cause des défauts de l'oeil (persistance rétinienne) et de l'oreille humaine (l'effet de masquage).

Type de modulation	Application
MSK, GMSK	GSM, CDPD
BPSK	Deep space telemetry, cable modems
QPSK, $\pi/4$ DQPSK	Satellite, CDMA, NADC, TETRA, PHS, PDC, LMDS, DVB-S, cable (return path), cable modems, TFTS
OQPSK	CDMA, satellite
FSK, GFSK	DECT, paging, RAM mobile data, AMPS, CT2, ERMES, land mobile, public safety
8, 16 VSB	North American digital TV (ATV), broadcast, cable
8PSK	Satellite, aircraft, telemetry pilots for monitoring broadband video systems
16 QAM	Microwave digital radio, modems, DVB-C, DVB-T
32 QAM	Terrestrial microwave, DVB-T
64 QAM	DVB-C, modems, broadband set top boxes, MMDS
256 QAM	Modems, DVB-C (Europe), Digital Video (US)

Table 1.1 – Les applications des différentes méthode de modulation [7]

Les caractéristiques principales d'une modulation sont : sa taille m , sa constellation et sa largeur de bande, qui d'après le critère de *Nyquist* est supérieure à $\frac{1}{2T}$ [8]. Par ailleurs, on mesure les performances d'une modulation grâce à l'efficacité spectrale η , qui fournit le nombre de bits d'information transmis par unité de temps et par unité de bande, soit :

$$\eta = \frac{D_b}{B} = \frac{mR}{BT} \text{ bit/Hz/s} \quad (1.1)$$

La Table(1.2) illustre quelques types de modulation, les plus utiliser dans la télécommunication et leurs limites théoriques de l'efficacité spectrale données en (bit/sec/Hz). Ainsi à un rapport signal à bruit donné, une modulation sera d'autant plus efficace que l'efficacité spectrale sera proche de la limite imposée par le théorème de *Shannon*.

Le démodulateur effectue le travail inverse du modulateur, c'est-à-dire qu'il convertit le signal reçu du milieu dans le format d'entrée du décodeur de canal. En pratique, le démodulateur doit détecter le signal entrant et fournir une mesure de fiabilité pour chaque point de la constellation. De ces mesures et de la détection dépend la probabilité d'erreur à un rapport signal à bruit donné.

La Table(1.1) cite les différents formats de modulation utilisés dans les communications sans fil et vidéo.

• Modulation BPSK :

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

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

Table 1.2 – Limite théorique de l'efficacité spectrale de modulation [7]

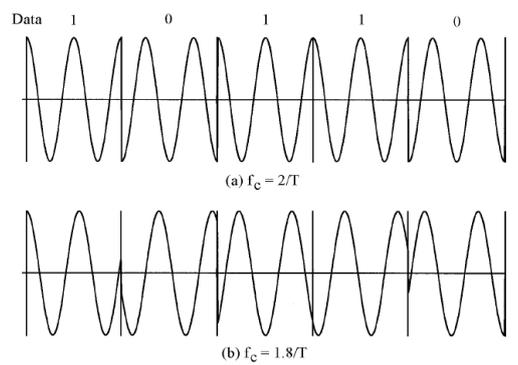


Figure 1.2 – Forme d'un signal modulé BPSK

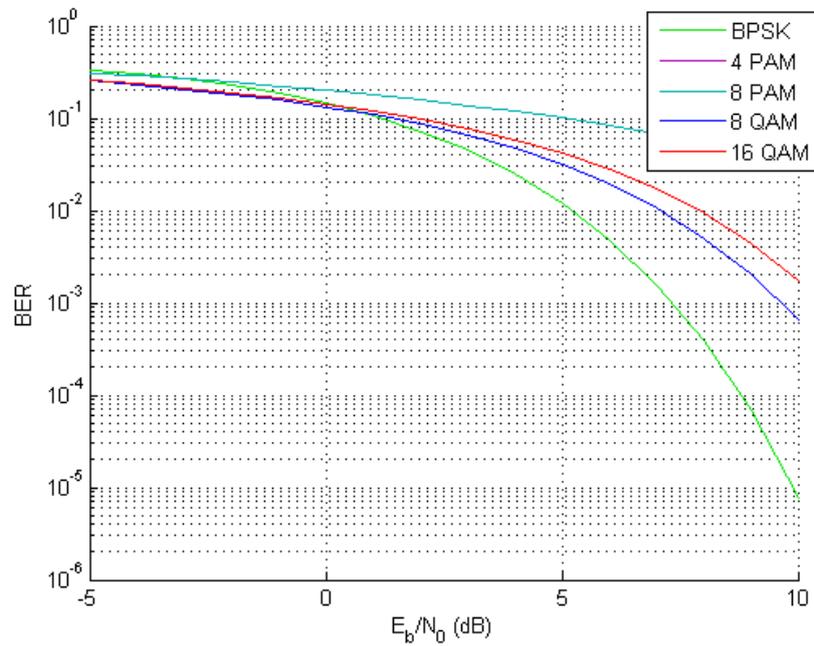


Figure 1.3 – Comparaison du BER de la modulation BPSK avec différents types de modulation

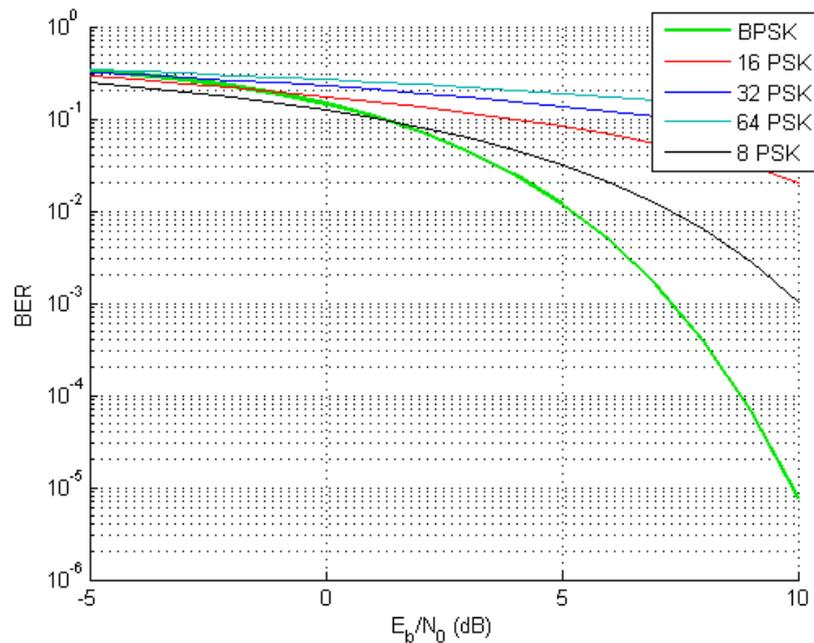


Figure 1.4 – Comparaison du BER de la modulation BPSK avec différents ordre de modulation

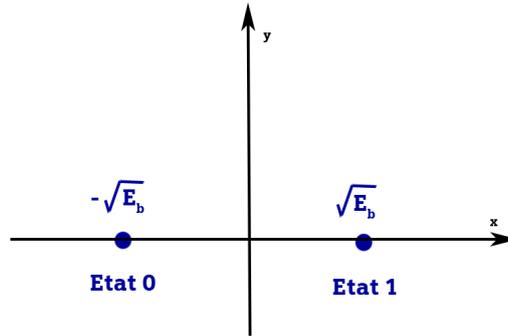


Figure 1.5 – Constellation du signal BPSK

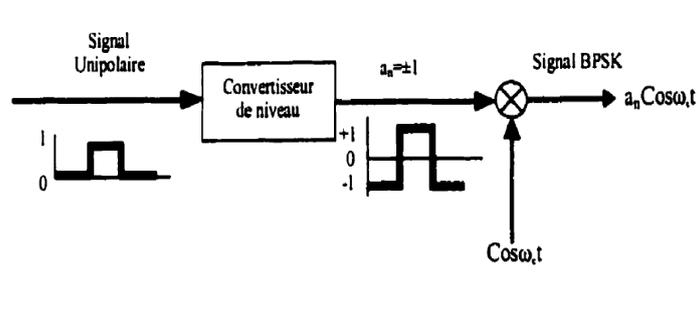


Figure 1.6 – Modulateur BPSK [2]

La modulation BPSK (Binary Phase Shift Keying) est parmi les modulations numériques les plus utilisées. D'après les simulations qu'on a fait, en comparant la modulation BPSK avec autre type de modulation (Figure(1.3)) et la modulation BPSK avec le même type de modulation PSK (Phase Shift Keying) mais avec des ordres de modulation plus grands (Figure(1.4)), on a remarqué qu'elle est la plus performante au niveau de la probabilité d'erreur.

En BPSK, les phases opposées de la porteuse (0 et π) sont transmises toutes les T secondes (si on considère que la durée d'un bit est T) et sont basées sur les symboles d'information $+1$ ou -1 . BPSK peut être considéré comme un type de modulation d'amplitude où la porteuse est multipliée par -1 ou $+1$ suivant le bit de parité comme le montre la Figure(1.2) où les données $+1$ et -1 sont représentées respectivement par $\cos(\omega_c t)$ et $\cos(\omega_c t + \pi)$. Nous pouvons représenter ceci sous forme de constellation dans un plan x / y .

BPSK se limite à deux signaux sur l'axe des x tel qu'il est montré à la Figure(1.5) où E_b est l'énergie d'un bit. Contrairement à la structure du modulateur BPSK, montrée à la Figure(1.6), le démodulateur est beaucoup plus complexe. En effet le récepteur BPSK doit fournir une phase de référence cohérente afin de démoduler le signal reçu.

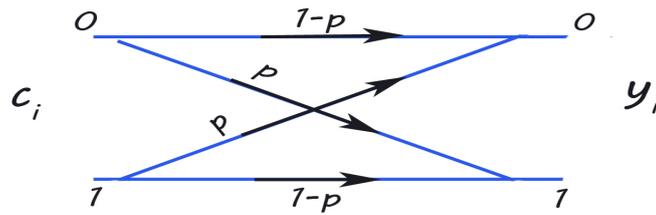


Figure 1.7 – Graphe de transition dans un (CBS)

Ou encore, un bit de synchronisation permet de retrouver la phase de référence [10]. Mentionnons finalement que le BER de la modulation BPSK cohérente dans un canal AWGN est donné par la formule suivante :

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (1.2)$$

ou

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp^{-y^2/2} dy$$

1.2.5 Le canal de communication

Le canal, au sens des communications numériques, et comme représenté à la Figure(1.1), inclut le milieu de transmission (lien physique entre l'émetteur et le récepteur : câble, fibre, espace libre,...), le bruit (bruit de fond, bruit impulsif, parasite, évanouissement, distorsion, défaut, panne), et les interférences (provenant des autres utilisateurs du milieu de transmission, de brouilleurs intentionnels ou non). Par la suite, aussi bien dans la présentation des résultats de la théorie de l'information que dans cette introduction au codage de canal, nous nous placerons principalement dans le cas discret et nous utiliserons un modèle de canal plus global, incluant une partie de l'émetteur et du récepteur.

a- Le canal binaire symétrique sans mémoire :

Le canal binaire symétrique sans mémoire (CBS) est le modèle de base de la théorie des codes correcteurs. Il y sert de référence et de point de comparaison. Chaque symbole binaire émis peut être reçu soit correctement avec la probabilité $(1-p)$, soit d'une manière erronée avec la probabilité p [11]. Ceci est schématisé par un graphe de transition dans la Figure(1.7), Le caractère sans mémoire caractérisé par :

Si une erreur perturbe un symbole donné c_i , émis à l'instant i , elle sera indépendante des erreurs affectant d'autres symboles c_j , où $i \neq j$.

b- Le Canal à bruit blanc additif Gaussien :

Le canal à bruit blanc additif Gaussien (BBAG) est un canal à entrée binaire et sortie analogique tel que chaque symbole sortant du canal est la somme du symbole émis X et d'un bruit Gaussien centré de variance σ^2 définie comme : $\sigma^2 = N_0/2$, où N_0 représente la densité spectrale de puissance du bruit. Autrement dit, si on note x_k la variable qui représente le signal modulé correspondant à un bit d'information à l'instant k , le signal correspondant à la sortie du canal sera :

$$y_k = x_k + n \quad (1.3)$$

Dans ces conditions, la probabilité conditionnelle que la sortie Y corresponde au symbole x_i est :

$$P(Y/X = x_i) = \frac{1}{\sigma \sqrt{2\pi}} \exp -\frac{(x_i - Y)^2}{2\sigma^2} \quad (1.4)$$

- l'adjectif "additif" dans AWGN signifie que l'impact du bruit sur le signal transmis peut être modélisé comme une variable aléatoire n qui s'additionne au signal modulé.
- Le terme "blanc" indique que la densité spectrale de puissance N_0 du bruit est considérée unilatérale (c.à.d de la fréquence 0 à inf) et constante.

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

$$y_k = x_k + n \quad (1.5)$$

c- Canal de Rayleigh :

Dans les liaisons radiomobiles, les canaux de transmission évoluent en fonction du temps à cause des déplacements aléatoires des entités communicantes et l'existence d'obstacles entre l'émetteur et le récepteur. Il peut en résulter que le signal émis suit plusieurs trajets avant d'arriver au récepteur, conduisant à une variabilité importante du signal reçu due à l'addition de plusieurs signaux déphasés. Lorsque le débit de transmission est suffisamment faible, chaque symbole ne se superpose qu'avec lui-même, au moins sur une portion significative de sa durée. Un canal de Rayleigh permet de prendre en compte ces effets : réflexions multiples, évanouissements, fluctuations à grande et petite échelle et effet Doppler. L'amplitude du signal reçu apparaît comme une variable aléatoire qui suit une loi de Rayleigh (équation 1.6). Ce modèle est particulièrement adapté à une représentation statistique d'un canal radiomobile.

$$P(x) = \frac{x^2}{\sigma^2} \exp -\left(\frac{x^2}{2\sigma^2}\right) \quad [12] \quad (1.6)$$

1.2.6 Information mutuelle et capacité d'un canal

Si on considère une communication à temps discret : $X_k \rightarrow Y_k$ avec X_k v.a. à valeurs dans $\{x_i\}$ et Y_k v.a. à valeurs dans $\{y_j\}$, l'information mutuelle moyenne représente l'information apportée par l'observation de la sortie Y_k sur la valeur prise par la source X_k , et est donnée par :

$$I(X, Y) = \sum_{i,j} P(x_i, y_j) \log_2 \frac{P(x_i / y_j)}{P(x_i)} \quad [3] \quad (1.7)$$

Si le canal est très bruité alors $P(x_i / y_j) = P(x_i)$ et l'information mutuelle est nulle, Si le canal est parfait (sans bruit) $P(x_i / y_j) = 1$ pour $i = j$ et 0 pour $i \neq j$ l'information mutuelle est donc égal à l'entropie de la source.

On montre que :

$$I(X, Y) = H(X) - H(X/Y) = H(Y) - H(Y/X)$$

La capacité d'un canal est définie comme le maximum de l'information mutuelle, ce maximum étant pris sur toutes les distributions de probabilité des symboles possibles. La capacité est donc donnée par :

$$C = \max_{P(x_i)} \{ I(X, Y) \} = \max(H(X) - H(X/Y)) \quad [3] \quad (1.8)$$

1.2.7 Le théorème fondamental du codage de canal

La théorie du codage, introduite par Shannon en 1948, associe à chaque canal une capacité représentant le maximum d'information transmissible sur ce canal (exprimée en bits par seconde). La théorie repose sur le théorème suivant également énoncé par Shannon :

"Pour une source à débit d'information R_D et un canal de capacité C , si $R_D < C$, il existe un code ayant des mots d'une longueur N tel que sa probabilité d'erreur soit arbitrairement petite" [13].

Dans sa démonstration, ce théorème assure l'existence d'un code (le code aléatoire) permettant une transmission fiable, bien qu'en pratique ce code est trop complexe à décoder. Depuis, la communauté scientifique s'efforce de trouver des codes correcteurs d'erreurs de longueur finie ayant une complexité raisonnable et s'approchant le plus près possible de la capacité.

1.3 Les codes correcteurs d'erreurs

Il existe une grande variété de codes correcteurs d'erreurs, dont les performances et les applications sont variables. Mais, le principe de base reste le même : ajouter de la

redondance intelligemment et utiliser cette sur-information pour déterminer la fiabilité du message (détection d'erreur), puis, si c'est possible reconstruire le message d'origine au mieux (correction d'erreur).

Mais en revanche, l'ajout de la redondance dans le message à transmettre, entraîne une perte d'efficacité du système. En effet, les bits de redondance introduits ne véhiculent pas de l'information utile. Cependant, cette perte est à mettre en balance avec le gain de qualité obtenu par l'utilisation du codage [9].

1.3.1 Mesure des performances d'un code correcteur d'erreurs

a- Rapport signal sur bruit :

Connaître la puissance du bruit N n'a un intérêt que si on peut la comparer à celle du signal P_s et en déduire son impact sur la dégradation du signal. C'est pourquoi on utilise généralement un rapport de puissance appelé rapport signal sur bruit (Signal Noise Ratio SNR) :

$$SNR = \frac{P_s}{N} \quad (1.9)$$

Le rapport signal sur bruit se rapporte toujours au niveau nominal du signal. Le plus souvent, celui-ci est exprimé en dB (équation(1.10)).

$$SNR(dB) = 10 \log\left(\frac{P_s}{N}\right) \quad [12] \quad (1.10)$$

Celui-ci va donc permettre d'apprécier la qualité d'un signal et déterminer la sensibilité d'un dispositif pour une densité spectrale du bruit donnée. Le rapport signal à bruit est une donnée surtout intéressante pour des signaux analogiques, puisqu'il va permettre d'estimer la dégradation subit par ce dernier. En effet, plus le rapport signal à bruit est faible, plus le signal est dégradé par le bruit et plus il sera difficile de supprimer l'influence du bruit sur le signal. Il est nécessaire de garantir un rapport signal à bruit important pour s'assurer que le signal reçu reste une « copie fidèle » du signal transmis.

Exemple : [12]

- Téléphonie classique : $SNR \geq 47\text{dB}$ ($B=3.1\text{ KHz}$), bruit à peine perceptible, bruit à 30dB très gênant ;
- Transmission de musique : $SNR \geq 50\text{dB}$ ($B=15\text{ KHz}$), plus sévère que les exigences en téléphonie puisque largeur de bande plus grande ;
- Transmission de télévision : $SNR \geq 52\text{dB}$ ($B=5\text{ MHz}$) ;
- Système GSM : $SNR \geq 8\text{dB}$ ($B=200\text{KHz}$), le bruit thermique étant de -120dBm à 290°K, le premier étage d'amplification ajoutant un bruit de 10dB, la sensibilité du récepteur est de -102dBm (63pW) ! Cette sensibilité permet de garantir un taux d'erreur binaire d'au plus 1 pour 100 bits.

b- Cas d'un signal numérique - Rapport signal à bruit par bit :

Les signaux numériques sont sensibles au bruit, mais ne sont pas aussi sensibles que les signaux analogiques. Contrairement à un signal analogique, la qualité d'un signal numérique ne se mesure pas à la distorsion du signal, mais à la possibilité pour un circuit digital de détecter correctement l'état binaire transmis. Alors que la principale contrainte d'une communication analogique est le rapport signal à bruit qui est directement relié à la distorsion du signal, celle d'une communication numérique est le taux d'erreur binaire.

Alors que les exigences en termes de rapport signal à bruit pour les transmissions analogiques sont très élevées (plusieurs dizaines de dB!), celles-ci sont beaucoup plus faibles pour des communications numériques. Les niveaux de bruit nécessaires pour induire une erreur binaire doivent être très grand et du même ordre que l'amplitude du signal. En général, il est possible de recevoir un signal numérique avec une qualité acceptable avec un rapport signal à bruit légèrement négatif.

Ainsi, le rapport signal sur bruit n'est pas la meilleure métrique pour qualifier la qualité d'un signal numérique. On préfère employer un rapport signal à bruit normalisé appelé rapport signal à bruit par bit noté E_b/N_0 . Il s'agit du rapport entre l'énergie véhiculée par un bit E_b et la densité spectrale en puissance du bruit N_0 , cette grandeur est directement reliée au taux d'erreur binaire (BER), et fixer une contrainte en termes de taux d'erreur binaire revient à fixer une contrainte sur le rapport E_b/N_0 .

Le rapport signal à bruit et le rapport signal à bruit par bit sont reliés par l'équation :

$$\frac{S}{N} = \frac{E_b}{N_0} \cdot \frac{F_b}{B} \quad [12] \quad (1.11)$$

ou

S : puissance du signal (W) ;

N : puissance du bruit (W) ;

E_b : énergie par bit (W.s/bit) ;

N_0 : densité spectrale de bruit (W/Hz) ;

F_b : débit binaire (bits/s) ;

B : bande passante du canal de transmission (Hz).

b- Taux d'erreur binaire :

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

numérique, on utilise la notion de taux d'erreur binaire ou Bit Error Rate (BER). Il s'agit du taux d'erreur mesuré à la réception d'une transmission numérique, et se calcule à l'aide de l'équation :

$$BER = \frac{\text{nombre de bits erronés}}{\text{nombre total de bits reçus}} \quad [12] \quad (1.12)$$

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

Remarque : Le BER n'est pas la seule métrique utilisée pour qualifier la qualité d'un signal numérique. On retrouve aussi le Frame Error Rate (FER) ainsi que le Block Error Rate (BLER) qui indiquent respectivement la probabilité d'erreur par trame et la probabilité d'erreur par blocs de données.

1.4 Conclusion

Dans ce chapitre, nous avons introduit, dans un premier temps, le schéma fondamental d'une communication numérique, en expliquant brièvement chacune de ses parties. Ensuite, nous avons présenté quelques notions fondamentales sur le codage de canal, en introduisant le théorème fondamental de codage de canal et la définition d'un code correcteur d'erreurs.

A travers les différentes simulations présentées dans ce chapitre, nous avons remarqué que la modulation BPSK présente de bonnes performances de point de vue BER, en justifiant avec ça le choix de la modulation BPSK dans le reste de notre travail. Dans le chapitre suivant nous allons introduire les systèmes de communication satellitaire et plus précisément la modélisation du canal satellitaire qui est le canal de transmission sur lequel nous allons appliquer le codeur du canal turbo.

Chapitre 2

Introduction aux communications satellitaires

2.1 Introduction

Les informations transmises par l'intermédiaire des canaux satellitaires ou autre dans le système de communication subissent des affaiblissements et des distorsions lors de leurs transmission. Pour obtenir des communications fiables, il est clair qu'un effort au niveau de la technique de communication et du modèle retenu pour le canal doit être mis afin de combattre les effets néfastes du canal. Dans ce qui suit, nous donnerons un aperçu d'un système de communication satellitaire puis décrivons brièvement les différents blocs constituant ce système. Ensuite, nous aborderons le concept principal présenté dans ce chapitre à savoir le bilan de puissance, et ce après avoir balayé les différents types de bruit introduits par le canal. On verra que le bilan de puissance va nous permettre de déterminer le rapport signal à bruit, une notion fort importante en communications telle que mentionnée dans le chapitre 1.

2.2 Principe des communications satellitaires

Les satellites fournissent des canaux de communication tout comme le font les systèmes basés sur les technologies terrestres : faisceaux hertziens et câbles (coaxiaux et fibres optiques). La Figure(2.1) illustre les différentes composantes d'un système de télécommunications par satellite que l'on a subdivisé en deux méga-blocs :

- Le secteur spatial ;
- Le secteur terrien.

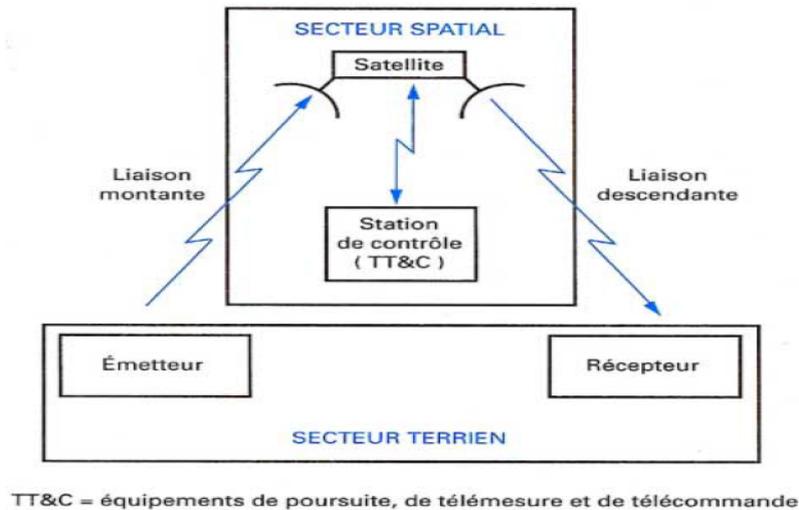


Figure 2.1 – Composantes d'un système de télécommunications par satellite

2.2.1 Le secteur spatial

Le secteur spatial comporte le satellite et l'ensemble des moyens de contrôle situés au sol, c'est-à-dire l'ensemble des stations de poursuite, de télémétrie et de télécommande (TTC : tracking, telemetry and command), ainsi que le centre de contrôle du satellite, où sont décidées toutes les opérations liées au maintien à poste et vérifiées les fonctions vitales du satellite.

Le satellite est constitué d'une charge utile et d'une plate-forme. La charge utile comporte les antennes de réception et d'émission (canaux, amplificateur ...), et l'ensemble des équipements électroniques assurant la transmission des signaux. La plate-forme comporte l'ensemble des sous systèmes permettant à la charge utile de fonctionner. On y trouve :

- la structure ;
- l'alimentation électrique ;
- le contrôle thermique ;
- le contrôle d'altitude et d'orbite ;
- les équipements de propulsion ;
- les équipements de poursuite, de télémétrie et de télécommande (TTC).

Pour atteindre le satellite, la station émettrice emprunte un faisceau montant (up-link) et pour atteindre la station réceptrice, le satellite emprunte un faisceau descendant (down-link). Les principales caractéristiques qui déterminent les performances d'un système satellite sont les suivantes :

- la ou les zones couvertes par le satellite.
- la gamme de fréquences utilisées.

- la puissance et la capacité de transmission.
- la taille des antennes des stations terriennes.
- les techniques de modulation et de codage utilisées pour la transmission des signaux.
- les techniques d'accès aux capacités de transmission embarquées sur le satellite.

2.2.2 Secteur terrien

Le secteur terrien est constitué par l'ensemble des stations terriennes, le plus souvent raccordées aux terminaux des usagers par un réseau terrestre, ou directement dans le cas de petites stations (VSAT : very small aperture terminal) et de stations mobiles. Les stations se distinguent par leur taille, qui varie selon le volume de trafic à acheminer sur la liaison spatiale, et selon le type de trafic (téléphone, télévision, données). Les plus grandes sont équipées d'antennes de 30 m de diamètre (standard A du réseau Intelsat) et les plus petites d'antennes de 0,6 m (stations de réception de la télévision directe). On distingue en outre les stations fixes, les stations transportables, les stations mobiles. Certaines stations sont à la fois émettrices et réceptrices. D'autres sont réceptrices uniquement (RCVO : receive only) : c'est le cas par exemple des stations de réception d'un système de radiodiffusion par satellite, ou d'un système de distribution de signaux de télévision ou de données.

2.3 Gammes de fréquences utilisées par les satellites

Les gammes de fréquences utilisées par les satellites sont désignées par des lettres et résultent d'accords internationaux. Deux modes d'exploitation ont été définis : télécommunication et télédiffusion directe. Le Tableau(2.1) présente ces gammes de fréquences utilisables pour les communications par satellite. La bande C fut la première à être mise en exploitation commerciale et elle est aujourd'hui très encombrée. De plus , cette bande est aussi utilisée par des systèmes terrestres de transmission hertzienne provoquant d'importants risques d'interférence.

L'usage de fréquences élevées permet d'utiliser des antennes de petite dimension mais la transmission est perturbée par des phénomènes naturels tels que la pluie. Les conditions climatiques de la zone à desservir privilégient l'usage de certaines bandes de fréquences. Les plus basses fréquences offrent une meilleure résistance à la rencontre de fortes pluies et les plus hautes permettent de réduire la taille des antennes au sol.

2.4 Services de télécommunications par satellite

Les satellites de télécommunications peuvent être classés en fonction des services qu'ils sont appelés à rendre. On distinguera ainsi les services de téléphonie, les services

Bande	Fréquences	Services
L	1 - 2 GHz	Communications avec les mobiles.
S	2 - 3 GHz	Communications avec les mobiles.
C	4 - 6 GHz	Communications civiles nationales et internationales, télévision.
x	7 - 8 GHz	Communications militaires.
Ku	11 - 14 GHz	Communications civiles nationales et internationales, télévision.
Ka	20 - 30 GHz	nouveaux systèmes d'accès aux réseaux large bande.
EHF	21 - 45 GHz	Communications militaires.

Table 2.1 – gammes de fréquences utilisées dans les télécommunications

de télédiffusion et enfin les services de transmission de données. Une telle classification pourrait être rendue obsolète dans quelques années par le développement attendu des systèmes satellitaires multimédia qui pourraient, en théorie, véhiculer n'importe quel service.

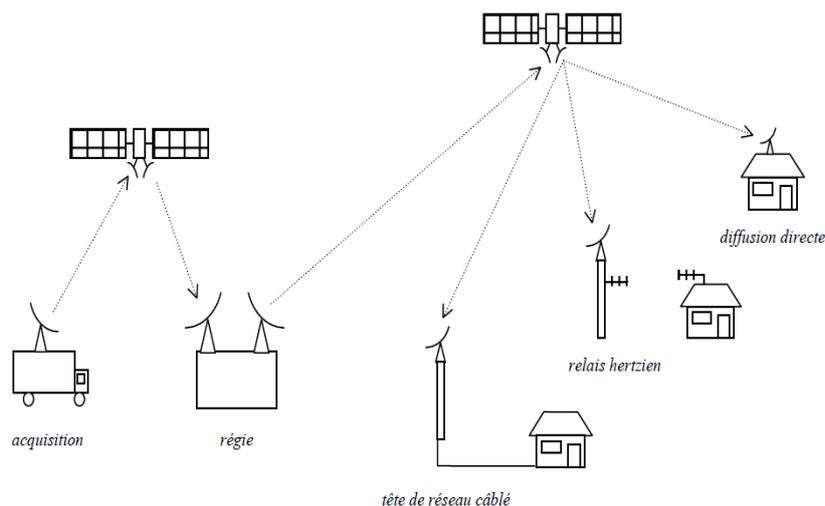


Figure 2.2 – Architecture d'un système de diffusion par satellite

2.4.1 La téléphonie

On peut distinguer les systèmes de téléphonie fixe par satellite et les systèmes de téléphonie mobile par satellite. La téléphonie fixe par satellite concerne essentiellement les appels internationaux qui mettent en jeu un satellite. Les satellites utilisés sont de type géostationnaire. Ces systèmes sont de plus en plus remplacés, lorsque c'est rentable, par des câbles sous marins.

A coté de la téléphonie fixe par satellite on trouve la téléphonie mobile par satellite et dans ce domaine, deux famille de systèmes existent. Les systèmes de téléphonie mobile

	GEO	MEO	LEO
Altitude	36 000 km	13 000 km	640 à 1 600 km
Temps de transit	0,25 s	0,1 s	0,05 s
Applications	Diffusion radio et TV	- Téléphonie mobile ; - Données à faible débit.	- Téléphonie mobile ; - Données à faible et à haut débit.
Débit binaire	jusqu'à 155 Mbit/s	de 9,6 à 38,4 kbit/s.	- Petits : 2,4 à 300 kbit/s ; - Gros : 2,4 à 9,6 kbit/s ; - Large bande : de 16kbit/s à 155 Mbit/s.

Table 2.2 – Les différents types de satellite de télécommunications

par satellite géostationnaires et les systèmes de téléphonie mobile par satellites MEO¹ et LEO². Parmi les systèmes utilisant des satellites GEO³ on peut citer *Inmarsat* qui a été le premier système de communications mobiles par satellites d'abord pour la desserte en mer puis en avion et en terrestre. Des systèmes sont actuellement en développement. On peut ainsi citer le cas récent de *Thuraya* qui offre un service de téléphonie mobile sur l'Asie centrale, le Moyen Orient, l'Afrique centrale et l'Europe. Ce système a la particularité d'offrir de la téléphonie mobile au moyen d'un satellite Géostationnaire. Le terminal de taille raisonnable (à peine plus volumineux qu'un terminal GSM) a une antenne omnidirectionnelle mais réussit cependant à "passer" le bilan de liaison d'un système GEO.

2.4.2 La télédiffusion

La transmission vidéo par satellite est la première application des satellites de télécommunications. Elle est estimée à 60% de la capacité du secteur spatial. La transmission peut être faite en utilisant les normes vidéo analogiques (PAL, SECAM, NTSC) mais dans la plupart des systèmes actuels ce sont les normes vidéo numériques qui sont utilisées. On utilise essentiellement la norme DVB-S (Digital Video Broadcasting Satellite) qui utilise un codage MPEG-2 du flux vidéo et permet le transfert du signal entre les systèmes à satellites, les systèmes hertziens et les systèmes par câbles. Tous les bouquets des satellites européens utilisent (obligation réglementaire) la norme DVB-S. La norme numérique audio correspondante pour la diffusion de radio numérique s'appelle le DAB (Digital

1. Les systèmes MEO (Medium Earth Orbit) qui correspondent à des satellites évoluant sur l'orbite médiane de 5.000 à 15.000 km et au dessus de 20.000 km.

2. Les systèmes LEO (Low Earth Orbit) qui correspondent à des satellites évoluant en orbite basse de 700 à 1.500 km

3. Les systèmes GEO (Geostationary Earth Orbit) qui correspondent à des satellites évoluant sur l'orbite géostationnaire.

Audio Broadcasting).

2.4.3 La transmission de données

La transmission de données constitue la troisième grande application des satellites de télécommunication.

a- Les réseaux d'entreprise VSAT :

Les réseaux d'entreprise par satellite sont apparus vers 1980 grâce à la diminution de la taille et donc du coût des stations terriennes. Cette diminution a permis au satellite de concurrencer efficacement les infrastructures terrestres pour des réseaux d'au moins quelques dizaines de sites. On parle de VSAT (Very Small Aperture Terminal) pour des stations terriennes dont le diamètre d'antenne est inférieur à 2,4 mètres.

b- La radiomessagerie :

La radiomessagerie par satellites est un service offert sur plusieurs satellites GEO comme Inmarsat-C et D, OmniTracs ou encore EutelTracs. Elle existe aussi sur des constellations en orbite basse telle que OrbComm.

c- L'Internet par satellite :

L'Internet par satellite est un sujet très important actuellement et le satellite fait partie des solutions d'accès local au même titre que le LMDS, l'ADSL, les solutions câblés ou d'autres technologies. Le programme SkyBridge d'Alcatel fait partie de ce type de systèmes.

2.5 Structure du système d'émission

Pour étudier une liaison satellite il faut d'abord étudier les différents blocs qui constituent la chaîne de transmission. Le satellite constitue l'élément émetteur de cette chaîne. Il est constitué de plusieurs blocs, les principaux blocs sont : le filtre en cosinus surélevé (Raised Cosine), Amplificateur Haute Puissance et antenne d'émission.

2.5.1 Filtre Cosinus surélevé

Les informations transmises par le satellite ont un débit le plus élevé possible. La contrainte est que le satellite dispose d'une bande étroite de fréquences. Pour émettre ces informations dans cette bande de fréquence il faut filtrer ce signal, afin d'éviter les interférences inter-symboles. Un filtre Raised Cosine est utilisé. Il est caractérisé par son

facteur Roll-off β . La bande passante utilisée est en fonction du facteur β et du débit symbolique.

2.5.2 Amplificateur Haute Puissance

Une des principales action d'un répéteur de satellite consiste à amplifier un signal avant de le transmettre. Dans le cas de satellites transparents, ce signal n'est autre que le signal reçu filtré et transposé. Dans le cas de satellites avec un traitement bord (OBP : On Board Processing), il s'agit d'un signal généré par la charge utile. L'amplification est ainsi au coeur du système et les problèmes de non linéarité causée par sa saturation demandent à être étudiés avec la plus grande attention.

Les courbes de caractéristiques des amplificateurs présentent en général l'Input Back-Off (IBO) en abscisse et l'Output Back-Off (OBO) en ordonnée. La saturation de l'amplificateur se traduit par un infléchissement de la courbe $OBO = f(IBO)$ (Figure(2.3)). On définit le point à $1dB$ de compression comme le lieu où la tangente à la courbe de gain aux faibles puissances s'écarte de $1dB$ de la courbe de gain réel. Ce point est très utile pour déterminer la proportion de la courbe de gain où l'amplificateur peut être considéré comme linéaire.

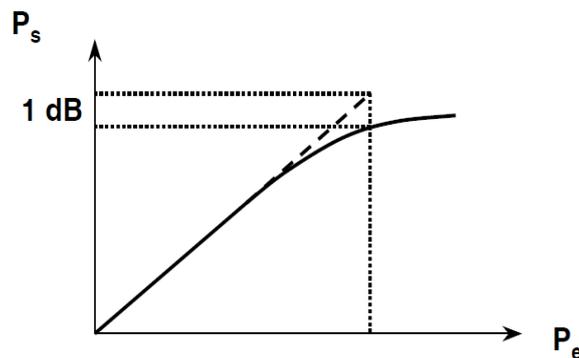


Figure 2.3 – La caractéristique d'amplificateur $OBO = f(IBO)$

Le modèle de Saleh, qui est considéré comme le plus précis pour la modélisation d'un amplificateur haute puissance, a été utilisé pour traduire cette non-linéarité qui affecte et l'amplitude et la phase du signal. Dans ce modèle le gain et la phase à la sortie de l'amplificateur sont décrits par :

$$g(r) = \frac{\alpha_g \cdot r}{1 + \beta_g \cdot r^2} \quad (2.1)$$

$$\phi(r) = \frac{\alpha_\phi \cdot r^2}{1 + \beta_\phi \cdot r^2} \quad (2.2)$$

où α et β sont des facteurs qui caractérisent la non linéarité, et r est le signal d'entrée.

2.5.3 Antenne d'émission

L'antenne est l'interface entre le guide d'onde du satellite et l'espace libre. Il transforme l'énergie électrique en une énergie magnétique qui se propage dans l'espace libre. Dans le domaine du satellite, les antennes sont choisies pour avoir le maximum d'information dans la station terrienne, pour minimiser les distorsions du signal dans l'espace.

Une antenne est en général caractérisée par son gain, et sa directivité. Une antenne jugée appropriée pour les communications satellitaires, devrait avoir un grand gain, et un angle à 3dB très étroit. Plusieurs types d'antennes sont utilisés pour remplir ses critères, dont les antennes à réflecteur parabolique, les antennes à lentille, et les réseaux d'antennes. Les antennes hélicoïdales sont plus particulièrement utilisées dans le domaine militaire. Le gain d'une antenne dépend de son diamètre, son rendement et de la fréquence utilisée tel que démontré par l'équation suivante.

$$G = \eta \left(\frac{\pi D}{\lambda} \right)^2 \quad (2.3)$$

où

D : le diamètre de l'antenne ;

η : le rendement de l'antenne (généralement varie entre 0.5-0.6) ; λ : la longueur d'onde qui est définie comme : $\lambda = C/f$; avec C : la célérité de la lumière et f : la fréquence du signal.

La zone de couverture du satellite, dépend de l'angle d'ouverture à (3dB) du lobe principal de l'antenne. Les zones de couvertures différent selon l'utilisation du satellite, la PIRE (Puissance Isotrope Rayonnée Effective ,On rappelle que la PIRE est la puissance rayonnée par rapport à une antenne isotrope pour laquelle $G_T = 1$) dépend de la couverture désirée : Dans les couvertures globales elle est de l'ordre de 15 – 25dBW (ex :télédiffusion), de l'ordre de 25 – 35dBW (ex :communication téléphonique) dans les couvertures zonales et de l'ordre de 35 – 45dBW pour les couvertures ponctuelles [1].

2.6 Structure du système de réception

2.6.1 Antenne de réception

La station terrienne doit avoir une antenne de réception sensible pour recevoir le signal émis par le satellite. L'antenne doit avoir un gain élevé, et un lobe très étroit pointé vers le satellite. Un lobe non directif n'est pas utile car la destination est ponctuelle. Un système de tracking est nécessaire si le satellite n'est pas géostationnaire.

2.6.2 Amplificateur à faible bruit(Préamplificateur)

Le signal arrivé à l'antenne est très faible (de l'ordre de nW ou pW), une amplification

peut distordre le signal, ce qui diminue le rapport signal sur bruit. Un amplificateur à faible bruit est utilisé afin d'amplifier le signal tout en gardant le rapport signal sur bruit élevé. Un amplificateur à faible bruit est caractérisé par son facteur de bruit. Il faut que le facteur de bruit soit petit pour ne pas distordre le signal. Le facteur de bruit est le rapport du rapport signal sur bruit à l'entrée sur le rapport de signal sur bruit à la sortie.

2.6.3 Filtre adapté

Pour reconstituer les symboles à la réception, on utilise un filtre adapté. Le filtre adapté prend des échantillons à des périodes fixes.

2.7 Les pertes

Pour déterminer l'ensemble des erreurs intervenant dans les mesures sans risquer d'en oublier une, une bonne méthode consiste à suivre le trajet de l'onde ou du signal depuis le satellite jusqu'à l'observateur et d'examiner tout ce qui est susceptible de modifier la phase, la fréquence ou la vitesse de propagation.

2.7.1 Propagation en espace libre

N'importe quelle onde électromagnétique se propageant dans le vide subit une atténuation dans l'espace. Cette atténuation dépend de la fréquence de l'onde et du trajet parcouru.

$$L_p = \left(\frac{4 \pi f_c}{V} \right)^2 \quad (2.4)$$

f_c : la fréquence du signal ;

V : La vitesse de la lumière.

La distance parcourue par le signal dans l'espace (le vide) est plus grande que celle parcourue dans l'atmosphère, donc on peut remplacer V par C qui est la vitesse de la lumière dans le vide .

2.7.2 influence du troposphère

Le phénomène spécifique aux liaisons par satellites est la présence de l'ionosphère entre 70 et 1000 *km* d'altitude. Cette couche est réfléchissante en dessous de 30 *MHz* (exploitée pour les ondes courtes). Au-delà, elle est transparente, mais crée néanmoins des perturbations :

- distorsion de temps de propagation de groupe, donc de phase ;
- rotation de la polarisation, à cause du champ magnétique terrestre.

Bande	Temps	Europe	Afrique
L	1 %	0	0
	0.1 %	0	0
S	1 %	0	0
	0.1 %	0	0
C	1 %	0	0
	0.1 %	0	De 0 à 0.3 dB
Ku	1 %	0.1 dB	De 0.2 à 2 dB
	0.1 %	0.8 dB	De 2 à 16 dB

Table 2.3 – Affaiblissement consécutif à la pluie

L'emploi de code correcteurs d'erreurs est fréquent sur les liaisons numériques par satellites, ainsi que les techniques numériques d'égalisation adaptative et d'annulation d'écho.

2.7.3 Atténuation due à la pluie

La propagation des ondes centimétriques est perturbée par la pluie. Le degré de cette perturbation est fonction de la fréquence utilisée. Le Tableau(3.1) montre l'atténuation due à la pluie dans deux régions (Europe et Afrique équatoriale) pour deux durées : 1% et 0.1% du temps. On constate que le signal est plus perturbé dans les régions pluvieuses (Afrique équatoriale). On constate aussi que les fréquences élevées sont très perturbée (Bande Ku) et cela peut même entraîner des interruptions de service dans la Bande Ka qui n'est pas représentée dans le tableau.

2.7.4 L'évanouissement multi trajets (Multipath fading)

Ce phénomène est provoqué par les combinaisons constructives et destructives des répliques du signal arrivant avec un délai au récepteur dû au temps de propagation dans les différents chemins entre l'émetteur et le récepteur. L'évanouissement multi trajet est encore appelé évanouissement sélectif en fréquence parce que il dépend fortement de la fréquence. Même lorsque la puissance reçue globalement est importante, cet évanouissement sélectif reste gênant car il provoque une déformation du signal émis qui peut grandement perturber la détection au niveau du récepteur. Pour étudier la faisabilité d'une transmission, nous avons donc besoin d'une mesure de ces délais. Mais l'influence d'un chemin de propagation sur le signal dépend aussi de son amplitude et de l'évaluation de l'écart entre le premier et le dernier chemin observé qui ne fournit pas une information suffisante. Cet écart est

Liaison	Montante	Descendante
Télémessure / Télécommande	2 GHz	2,2 GHz
Télécommunications	6 GHz	4 GHz
	8 GHz	7 GHz
	14 GHz	11 GHz
Nouveaux systèmes	30 GHz	20 GHz
Radiodiffusion / Télévision	10,7 à 11,7 GHz	11,7 à 12,7 GHz
Liaison avec les mobiles	1,6 GHz	1,5 GHz

Table 2.4 – Les fréquences de travail montante et descendante selon le type de liaison

dans la pratique caractérisé par l'étalement efficace du délai défini par :

$$\tau_{RMS} = \sqrt{\frac{\sum_{i=0}^{R-1} (\tau_i - \tau_0 - \bar{\tau})^2 |\beta_i|^2}{\sum_{i=0}^{R-1} |\beta_i|^2}} \quad (2.5)$$

où $\bar{\tau}$ est le délai excédentaire moyen dont la valeur est obtenue par :

$$\bar{\tau} = \frac{\sum_{i=1}^{R-1} (\tau_i - \tau_0) |\beta_i|^2}{\sum_{i=1}^{R-1} |\beta_i|^2} \quad (2.6)$$

Différents procédés existent pour limiter les effets de l'évanouissement multi trajet. On peut agir à l'émission en adaptant la mise en forme des signaux pour les rendre moins sensibles aux déformations. Il existe aussi une possibilité de doter le récepteur de moyens d'analyse des déformations afin d'en tenir compte dans ses décisions, dits à maximum de vraisemblance, ou tout au moins d'y apporter des corrections avant de fournir le signal à un organe de décision simplifié. On parle alors d'égalisation.

2.7.5 L'ombrage (Shadowing)

Ce terme fait référence aux variations de puissance reçues dues à la présence d'obstacles de grandes dimensions. En première approximation, on considère qu'elles agissent uniformément sur la toute la largeur de bande de transmission. En règle générale, elle peut être considérée comme très lente à l'échelle des débits de transmission. Le phénomène d'ombrage est d'ailleurs souvent désigné sous l'appellation d'évanouissement lent (Slow Fading), cela veut dire que la durée d'un symbole est très inférieure au temps de cohérence du canal, et le signal reçu ne varie donc pas ou très peu sur cette durée.

2.7.6 Effet Doppler

L'effet Doppler est le décalage en fréquence dû à la mobilité, il est fonction de :

- La Direction du déplacement ;

- La vitesse de déplacement.

Si en prend par exemple un satellite **LEO**, sa vitesse de défilement v est d'environ 7 km/s. L'écart de fréquence Δf qui résulte de cette vitesse est égal à \mathbf{v}/\mathbf{c} , où \mathbf{c} est la vitesse de la lumière.

$$f_{reelle} = f_0(1 + \frac{v}{c}) \quad (2.7)$$

Pour $f_0 = 1,5GHz$ (GPS), $\Delta f = 35kHz$. Cet écart de fréquence induit une rotation de la constellation de $2\pi\Delta fT_s$ où T_s est la durée d'un symbole.

Si l'on accepte une certaine dégradation du rapport E_b/N_0 , la modulation différentielle permet de compenser la rotation due à l'effet Doppler jusqu'à $\Delta fT_s = 10^{-2}$.

2.8 Bilan de liaison

Lors de la définition d'un système de communications avec un satellite, il est nécessaire de déterminer la taille des antennes d'émission et de réception, la puissance d'émission et le rapport signal à bruit nécessaire pour pouvoir effectuer la transmission avec la qualité requise. Evaluer tous ces paramètres constitue le bilan de liaison (Figure(2.4)).

Considérons une antenne omnidirectionnelle, dite isotrope (Isotropic Antenna) rayonnant une puissance de P_T Watts. La densité de puissance à une distance d de l'antenne qui rayonne dans une sphère de surface $4\pi d^2$ est alors égale à :

$$\frac{P_T}{4\pi d^2} \quad W/m^2 \quad (2.8)$$

Supposons maintenant que l'antenne est directive et rayonne principalement dans une direction définie par son azimuth et son élévation (θ_0, ϕ_0) . Par rapport à l'antenne omnidirectionnelle la densité de puissance dans cette direction sera multipliée par un coefficient $G_T(\theta_0, \phi_0)$ qui représente le gain de l'antenne dans cette direction.

$$\frac{G_T P_T}{4\pi d^2} \quad W/m^2 \quad (2.9)$$

Le produit $G_T P_T$ est appelé la Puissance Isotrope Rayonnée Effective : PIRE

Une antenne de réception dirigée dans la direction de rayonnement principal de l'antenne d'émission va recevoir une fraction de la puissance rayonnée. Cette fraction est proportionnelle à la surface de l'antenne de réception et à son orientation par rapport à la direction de propagation de la puissance émise. En supposant les antennes d'émission et de réception parfaitement alignées, la puissance reçue s'écrit :

$$P_R = \frac{G_T P_T A_R}{4\pi d^2} \quad (2.10)$$

A_R : est la surface effective de l'antenne de réception. Pour une antenne parabolique de diamètre \mathbf{D} , on a :

$$A_R = \eta \frac{\pi D^2}{4} \quad (2.11)$$

η : l'efficacité de l'antenne de réception. elle varie généralement entre 50% et 70%.

Le gain d'une antenne parabolique de diamètre D s'exprime quant à lui par l'équation :

$$G_R = \eta \left(\frac{\pi D}{\lambda} \right)^2 \quad (2.12)$$

La puissance reçue par l'antenne s'écrit finalement :

$$P_R = \frac{G_T P_T G_R}{\left(\frac{4\pi d^2}{\lambda} \right)^2} \quad (2.13)$$

On introduit alors le facteur $L_s = \left(\frac{\lambda}{4\pi d^2} \right)^2$ qui est appelé la perte en espace libre (free-space path loss). La puissance reçue s'écrit alors :

$$P_R = G_T P_T G_R L_s \quad (2.14)$$

En prenant en compte les pertes de propagation atmosphérique sous la forme d'un terme L_a , la puissance reçue devient :

$$P_R = G_T P_T G_R L_s L_a \quad (2.15)$$

Prise en dB cette expression devient :

$$(P_R)_{db} = (G_T)_{db} + (P_T)_{db} + (G_R)_{db} + (L_s)_{db} + (L_a)_{db} \quad (2.16)$$

Pour terminer le bilan de liaison il faut prendre en compte le bruit additif du canal et du récepteur. Le bruit thermique est défini par sa densité monolatérale de puissance :

$$N_0 = kT \quad W/Hz \quad (2.17)$$

avec :

k : constante de Boltzmann : $k = 1,38 \cdot 10^{-23} JK^{-1}$;

T : température du bruit en Kelvin.

La puissance de bruit P_n dans une bande de fréquence W est alors égale à :

$$P_n = N_0 W \quad (2.18)$$

En introduisant l'énergie par bit E_b dans la bande de réception et le débit binaire R_b , il vient :

$$P_R = E_b R_b \quad (2.19)$$

Le rapport $\frac{E_b}{N_0}$ est alors égal à :

$$\frac{E_b}{N_0} = \frac{P_R}{R_b N_0} \quad (2.20)$$

Pour obtenir un taux d'erreurs spécifié lors de codage du canal, il est nécessaire de définir un rapport $\frac{E_b}{N_0}$ requis noté $\left(\frac{E_b}{N_0} \right)_{req}$, et d'ajuster les puissances d'émission et les tailles des antennes afin que :

$$\frac{P_R}{N_0} = R_b \left(\frac{E_b}{N_0} \right)_{req} \quad (2.21)$$

En remplaçant P_R et N_0 par leur expression respective, on obtient :

$$\frac{P_R}{N_0} = \frac{G_T P_T L_s L_a G_R}{k T} \quad (2.22)$$

On voit alors faire apparaître le terme $\frac{G_R}{T}$ appelé facteur de mérite du récepteur qui est une caractéristique très importante pour qualifier la chaîne de réception d'un système satellitaire.

Remarque Dans cet exposé du bilan de liaison on a supposé une chaîne de réception parfaite sans dégradation du rapport signal sur bruit. Une chaîne "réelle" sera caractérisée par son facteur de bruit qu'il faudra donc soustraire au rapport signal sur bruit idéal.

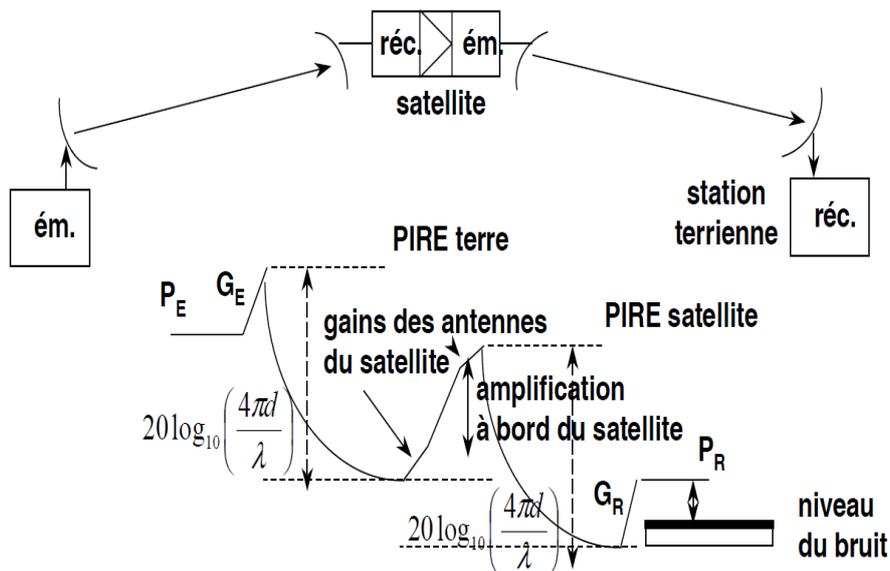


Figure 2.4 – Bilan de puissance d'une liaison satellitaire

Chapitre 3

Généralités sur les codes convolutifs

3.1 Introduction

Un turbocode est une mosaïque qui a pour but d'imiter un code convolutif de grande mémoire ν . Il est construit par association de plusieurs petits codes systématiques récurrents (CSR) dont les décodages particuliers sont de complexité raisonnable [4].

Dans ce chapitre nous nous alignerons avec l'ordre chronologique du développement des codes correcteurs pour organiser la présentation des concepts de telle sorte à décrire d'abord les codes en bloc, ensuite les codes convolutionnels en général et leurs représentations et enfin les codes récurrents comme cas particulier. Nous terminerons par l'introduction de la notion d'espace libre qui est une caractéristique importante pour évaluer le gain de performance offert par le code convolutif.

3.2 Les codes en bloc

On appelle code en bloc de longueur n sur F un sous ensemble C de F^n . Il est d'usage, en codage algébrique, de supposer que F est un corps fini, afin d'avoir accès à la structure d'espace vectoriel. Ceci implique que la cardinalité $q = |F|$ est une puissance m -ième d'un nombre premier p .

La dimension du code est par définition :

$$k = \log_q (|C|) \tag{3.1}$$

Dans les cas pratiques, la dimension k est un nombre entier, elle représente le nombre de symboles d'information par mot de code. Le rapport $R = k / n$ est appelé le taux ou rendement du code.

3.2.1 les codes en blocs linéaires

On dit que le code en blocs de paramètres (n, k) , noté $C(n, k)$, est un code en bloc linéaire si les mots de code constituent un sous-espace vectoriel de F_q^n , ce qui veut dire que la somme de deux mots de code est un mot de code, et que le mot nul constitué de n symboles à zéro est toujours un mot de code [4].

3.2.2 les codes en blocs linéaires à symboles binaires

Dans les codes en blocs à symboles binaires, $n = 2$ c'est à dire que les éléments du sous ensemble C sont à valeurs dans F^2 . Comme la fonction g qui représente la sortie en fonction de l'entrée est une application linéaire, nous pouvons décrire l'opération de codage simplement comme le résultat de la multiplication d'un vecteur de k symboles représentant les données à coder par une matrice représentative du code considéré, appelée matrice génératrice du code [4].

3.3 Codage convolutionnel

Un codeur convolutionnel binaire (n, k) est un dispositif qui accepte des k -tuples binaires en entrée et produit n -tuples binaires en sortie. Il existe deux types de codeurs convolutionnels. Le premier est appelé **codeur convolutionnel RFI** (Réponse Finie à une Impulsion et connu sous le nom de codeur convolutionnel non récursif). Le deuxième type est appelé **codeur convolutionnel RII** (Réponse Infinie à une Impulsion ou mieux connu sous le nom de codeur convolutionnel récursif). Un codeur convolutionnel est **systematique** si l'information présente à l'entrée est produite à la sortie. Dans le cas contraire le codeur est dit **non-systematique**. [1]

3.3.1 Codeur convolutionnel non récursif (RFI)

Un codeur est dit **RFI** si sa sortie peut s'exprimer comme une combinaison linéaire de l'entrée courante et d'un nombre fini des entrées précédentes. La combinaison linéaire est exprimée en fonction des bits d'entrée et des séquences génératrices du codeur. Soit la séquence génératrice $\{g_{i,o,l}\}$ qui relie une séquence d'entrée particulière $\{d_{t-l}\}$ à une séquence de sortie particulière $\{X_t^o\}$. Une valeur particulière de $g_{i,o,l}$ signale la présence ou bien l'absence d'une connexion entre le l ème élément de mémoire $l \in [1, \nu_i]$ du i ème entrée $i \in [1, k]$ à la o ème sortie $o \in [1, n]$. Les n équations de sortie ont la forme :

$$X_t^o = \sum_{i=1}^k \sum_{l=1}^{\nu_i} g_{i,o,l} \cdot d_{t-l} \quad (3.2)$$

Les mémoires de chacune des k entrées sont énumérées par le vecteur mémoire (v_1, v_2, \dots, v_k) . Dans la Figure(3.1) et (3.2), les codeurs possèdent une seule entrée donc $k = 1$, deux sorties donc $n = 2$ et enfin $v_1 = 2$ (deux délais) est la valeur de la mémoire. [1]

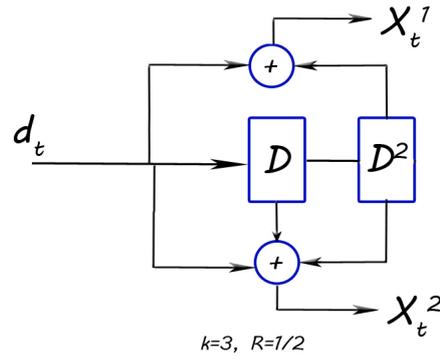


Figure 3.1 – Codeur RFI non systématique de matrice génératrice $G(D) = [1 + D^2, 1 + D + D^2]$

3.3.2 Codeur convolutionnel récursif (RII)

Un codeur convolutionnel est dit *RII* si les n équations de sortie sont en fonction des précédentes entrées et sorties. ce qui implique que la sortie dépend d'une infinité d'entrées précédentes. Un exemple de codeur *RII* est présenté dans la Figure (3.2). Ce dernier est aussi systématique du fait que l'entrée du codeur se retrouve à sa sortie.

Dans le cas des turbo codes, l'intérêt se porte sur les codeurs convolutionnels *RII* et systématiques. D'où la nécessité de transformer un codeur RFI spécifique en un codeur *RII* systématique. La condition à satisfaire est que le codeur RFI soit non catastrophique [14].

Un codeur est dit non catastrophique si et seulement si le seul chemin qui produit exclusivement des zéros à la sortie est la boucle unique qui part de l'état initial du codeur (en général l'état zéro c.a.d les registres du codeur sont supposés remplis par des zéros) y retourne instantanément [10].

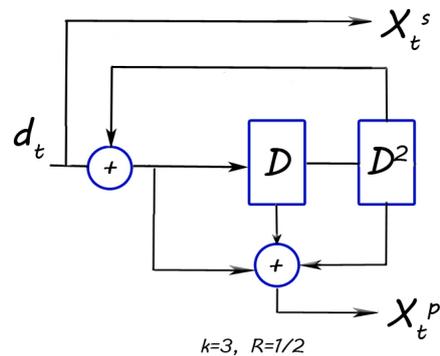


Figure 3.2 – Codeur *RII* systématique de matrice génératrice $G(D) = [1, 1 + D + D^2 / 1 + D^2]$

3.3.3 Représentation polynômiale des codes convolutifs

En définissant D le délai séparant deux instants successifs, les sorties des codeurs *RFI* peuvent s'exprimer en fonction des entrées par l'équation matricielle suivante :

$$\begin{aligned} \vec{X}(D) &= [X_1(D), X_2(D), \dots, X_n(D)] \\ &= [d_1(D), d_2(D), \dots, d_n(D)] \begin{pmatrix} g_{1,1}(D) & g_{1,2}(D) & \cdots & g_{1,n}(D) \\ g_{2,1}(D) & g_{2,2}(D) & \cdots & g_{2,n}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1}(D) & g_{k,2}(D) & \cdots & g_{k,n}(D) \end{pmatrix} \end{aligned} \quad (3.3)$$

où

$$d_u(D) = \sum_t d_t^u D^t \quad (3.4)$$

La matrice polynomiale $G(D)$ est appelée la matrice génératrice du codeur. Dans le cas d'un codeur *RFI*, les termes de la matrice génératrice sont des polynômes de la forme :

$$g_{i,o}(D) = \sum_{l=1}^{\nu_i} g_{i,o,l} D^l \quad (3.5)$$

dont le degré est inférieur ou égal ν_i . En appliquant les définitions précédentes au codeur de la Figure(3.1), nous obtenons la matrice génératrice suivante :

$$G(D) = \begin{bmatrix} 1+D^2 & 1+D+D^2 \end{bmatrix} \quad (3.6)$$

Pour le cas du codeur systématique récursif (RSC) de la Figure(3.2), celui-ci est construit à partir du codeur non systématique non récursif (NRNSC) de la Figure(3.1), et donc de façon similaire au codeur *RFI*, le codeur convolutionnel *RII* peut être décrit par une matrice génératrice. A la différence du cas non récursif, les éléments de la matrice génératrice sont des fonctions rationnelles de la variable D . Ajoutons, que dans le cas où le codeur est aussi systématique, au moins le premier terme de $G(D)$ doit être égal à 1. La matrice génératrice du codeur systématique récursif représenté dans la Figure(3.2) s'écrit sous la forme :

$$G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix} \quad (3.7)$$

3.3.4 Représentations graphiques des codeurs convolutifs

Il y a trois alternatives pour représenter un code convolutionnel graphiquement : **l'arbre**, **le treillis** et le **diagramme d'état**. Notons que le treillis peut être compacté sous une autre forme qu'on nomme le diagramme d'état.

a- Diagramme d'état :

Le diagramme d'état est un graphe qui représente les états possibles du codeur et les transitions d'un état à un autre. Le diagrammes d'état du codeur précédent (Figure(3.1)) est présenté dans la (Figure(3.3)). S_0 , S_1 , S_3 et S_4 correspondent respectivement aux états 00, 01, 10 et 11. Les lignes en gras indiquent que le bit d'information est un "1" et celles en pointillés indiquent que c'est un "0". La structure des diagrammes d'états est identique pour les codeurs *RFI* de la Figure(3.1) et *RII* de la Figure(3.2) [15]. Cependant, les séquences de sorties des codeurs *RFI* et *RII* sont différentes pour la même séquence d'entrée. Nous pouvons remarquer ceci en comparant les deux diagrammes d'états, par exemple le passage de l'état S_1 à l'état S_0 a été provoqué par le bit d'entrée "0" pour le codeur de la Figure(3.1) tandis que ce même passage est provoqué par le bit "1" pour le codeur de la Figure(3.2).

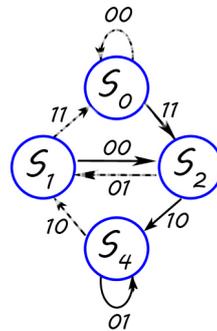


Figure 3.3 – Diagramme d'état du codeur RIF $G(D) = [1 + D^2, 1 + D + D^2]$

b- Représentation sous forme d'arbre :

La représentation des codes convolutifs qui s'est révélée la plus féconde, notamment pour la conception d'algorithmes de décodage, est celle issue de théorie des graphes. La représentation sous forme d'arbre qui fait partie des représentations graphiques est construite de la manière suivante :

- Le temps s'écoule de la gauche vers la droite.
- Chaque décalage du registre, après l'instant initial donne lieu à une bifurcation. La branche montante est adoptée si le symbole introduit est égal à "0", et la descendante est retenue s'il est "1". Le point de bifurcation qui est un trait horizontal s'appelle un "noeud" et donne naissance à deux branches dans ce cas-ci (2^k dans le cas général). On repère chaque branche par le bloc de "n" symboles émis après chaque décalage du registre qui était initialement zéro.

La suite codée correspondant à un message, est alors représentée par un chemin, c'est-à-dire une succession de branches dans l'arbre. Chaque bloc de n symboles émis lorsqu'un bloc de r symboles d'information est introduit dans le codeur dépend :

- De ce bloc.
- Des k blocs précédents, c'est-à-dire du contenu des $k.r$ derniers étages du registre.

Pour chaque état initial du codeur on peut donc tracer un arbre. Tous les états possibles sont réalisés après $(k+1)$ décalages de r symboles, puisque c'est le nombre nécessaire pour évacuer le contenu initial du registre. Les arbres partiels tracés à partir de la $(k+2)$ ème ligne de noeuds ne feront donc que répéter les arbres partiels différents que d'états, soit $2^{(k.r)} = 4$ dans notre exemple. Il suffit d'arrêter la construction à ces partiels.

Dans notre exemple le début de l'arbre associé au code de rendement $1/2$ donne :

1. Racine :

- 0 entre \rightarrow sortie : 00 sur le noeud N_{s1}
- 1 entre \rightarrow sortie : 11 sur le noeud N_{i1}

2. Premier noeud supérieur N_{s1} (avec un 0 dans la première cellule) :

- 0 entre \rightarrow sortie : 00 sur le noeud N_{s11}
- 1 entre \rightarrow sortie : 11 sur le noeud N_{i12}

3. Premier noeud inférieur N_{i1} (avec un 1 dans la première cellule) :

- 0 entre \rightarrow sortie : 10 sur le noeud N_{s11}
- 1 entre \rightarrow sortie : 01 sur le noeud N_{i12}

Etc.

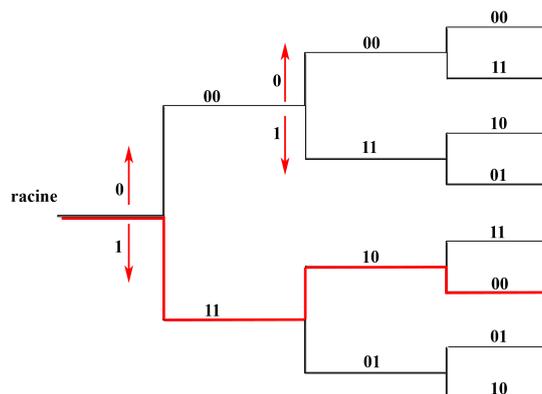


Figure 3.4 – Représentation d'un codeur sous forme d'arbre [3]

Ainsi par exemple, si l'on figure sur cet arbre le mot d'information "101", on peut immédiatement lire sur les noeuds correspondants aux évolutions des "0" et des "1", le mot

codé qui est donné ici par le chemin représenté en rouge sur la Figure(3.4), qui restitue la séquence :

111000

Bien entendu cette représentation devient rapidement inexploitable dès que $2^{k.r}$ devient important.

c- Représentation sous forme de treillis :

Le nombre de noeuds se dédouble à chaque étage, l'arbre, bien que très simple, devient donc rapidement très encombrant. Par ailleurs on remarque qu'au bout de trois étages les moitiés supérieure et inférieure de chaque noeud sont bien symétriques et dans le même ordre que prévu.

Dans le cas de notre exemple on a (00) contre (11) près de la racine, (00 11) contre (10 01) à l'étage suivant, puis (00 11 10 01) contre (11 00 01 10) et enfin (00 11 10 01 11 00 01 10) se répétant deux fois.

Cette symétrie devenant permanente à partir de cet étage. Ce comportement est à rapprocher du fait que le codeur comporte un registre à décalage de trois bits et le système ne garde donc aucun souvenir de ce qui s'est passé plus de trois coups d'horloge plus tôt.

Afin de rendre la représentation plus exploitable, tout en tenant compte de la récursivité de l'arbre, l'idée a été de réunir branche par branche et étage par étage, les noeuds équivalents. En somme « on tresse » les branches de l'arbre en « treillis ».

La disposition change alors un peu, la racine se place en haut et les branches descendent de telle manière que les mêmes états se retrouvent sur une même horizontale. A partir du troisième étage (dans notre exemple), certaines branches remontent et à partir du quatrième, à chaque noeud aboutissent deux branches et de chaque noeud partent deux branches. La représentation devient alors périodique.

Dans notre exemple, le rendement du code est de $\frac{1}{2}$ et le treillis a quatre états est représenté dans la Figure(3.5).

Le principe du codage dans un treillis suit les mêmes étapes que celle de l'arbre. On monte pour un « 0 » et l'on descend pour un « 1 », et chaque branche entre deux états est liée avec un symbole qui sera mis en sortie lorsque cette branche est empruntée sauf que dans ce cas, le treillis ne diverge pas comme l'arbre mais se replie sur lui-même à partir de $i=3$ (dans notre exemple).

Par exemple le codage du mot d'information « 1011 (00) » y compris la queue du

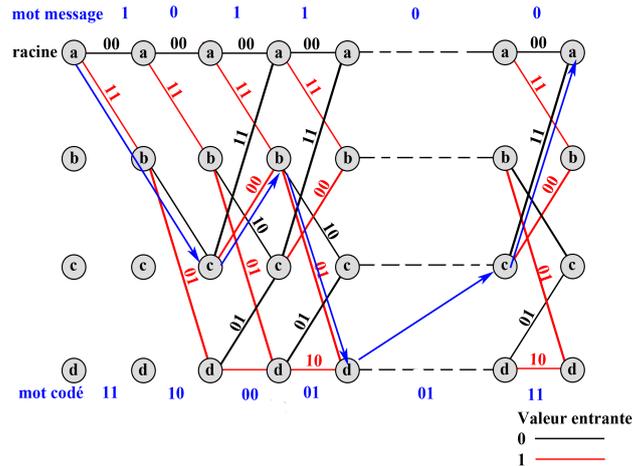


Figure 3.5 – Représentation d’un codeur sous forme de treillis [3]

message (00) est alors la suivant (voir la Figure(3.5)) :

111000010111

3.4 Distance libre

La notion de distance pour les codes convolutifs est semblable à celle des codes en blocs. Elle est liée à la distance de Hamming entre les séquences codées. Dans le cas convolutif ces séquences sont de longueur variable, deux séquences peuvent diverger sur quelques instants puis converger puis diverger à nouveau. On définit donc la **distance libre** d’un code convolutif comme la plus petite distance de Hamming existant entre deux chemins (suites d’états) qui divergent puis convergent de nouveau. Comme un code convolutif est linéaire, la distance libre est encore égale au poids du chemin à plus faible poids divergeant du chemin « tous zéros » et re-convergeant vers ce chemin de poids nul. [11].

3.5 Décodage des codes convolutif

Il y a une multitude d’algorithmes qui sont utilisés pour le décodage des codes convolutifs, on a pris à titre d’exemple l’algorithme de Viterbi puisque il détecte et il corrige l’erreur d’une façon simple et pratique.

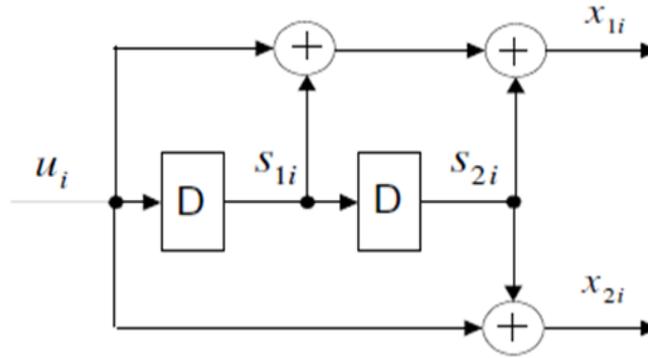


Figure 3.6 – Codeur convolutif non systématique $[(7, 5), (5)]$

Etat interne	S_{1i}	S_{2i}
a	0	0
b	0	1
c	1	0
d	1	1

Table 3.1 – Les états internes d’un CR a 2 mémoires

3.5.1 Algorithme de Viterbi

Bien que l’algorithme de Viterbi ne soit pas au sens strict utilisé dans le décodage des turbo codes qui sont l’objet de notre étude, le mécanisme de décodage de ces derniers fait intervenir des algorithmes très semblables, bien que légèrement plus complexes. Nous allons donc désormais développer brièvement l’algorithme de Viterbi, qui est en réalité un algorithme très efficace mais également très intuitif de décodage des codes convolutifs à l’aide de leur treillis (Cet algorithme permet d’ailleurs de décoder tous les codes pouvant être décrits par un treillis, ce qui est beaucoup plus général que les simples codes convolutifs). Pour expliquer le principe de fonctionnement d’un décodeur de code convolutif qui travail suivant l’algorithme de Viterbi , nous allons prendre l’exemple suivant :

- Soit le codeur convolutif non récursif non systématique de rendement 1/2 défini par sa matrice génératrice

$$G(D) = (1 + D + D^2 , 1 + D^2) \tag{3.8}$$

- Notre codeur contient deux case mémoire => 4 états internes notés a,b,c et d représente (voir Tableau(3.1)).

- Le treillis élémentaire du codeur convolutif présent dans la Figure(??).

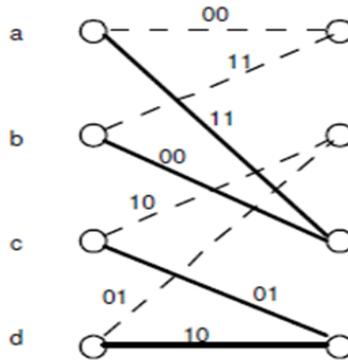


Figure 3.7 – Le treillis élémentaire du codeur convolutif $[(7, 5), (5)]$

- On considère la séquence d'information suivante :

1001

- La séquence codée est donc :

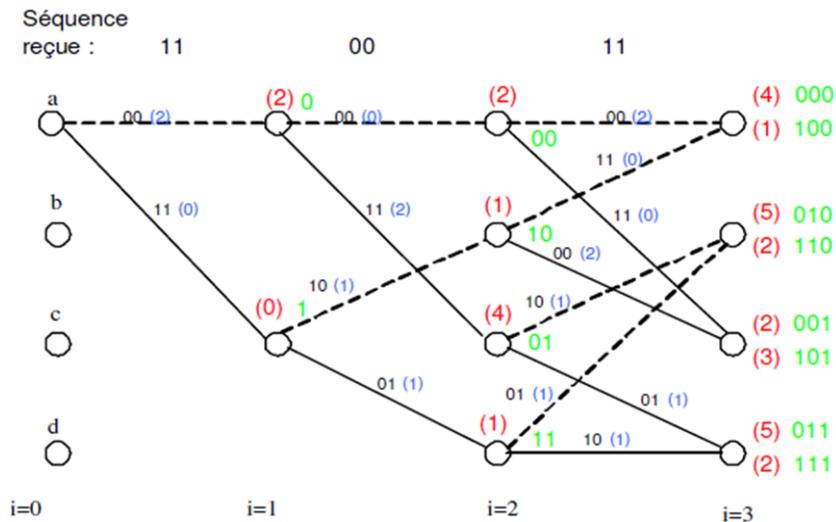
11101111

- Une erreur survient dans la transmission du troisième bit ;

- La séquence reçue est donc :

11001111

- On décode en utilisant l'algorithme de Viterbi :



(erreur de transition de l'état précédent vers l'état actuel)

(la somme des erreurs, métrique de branche)

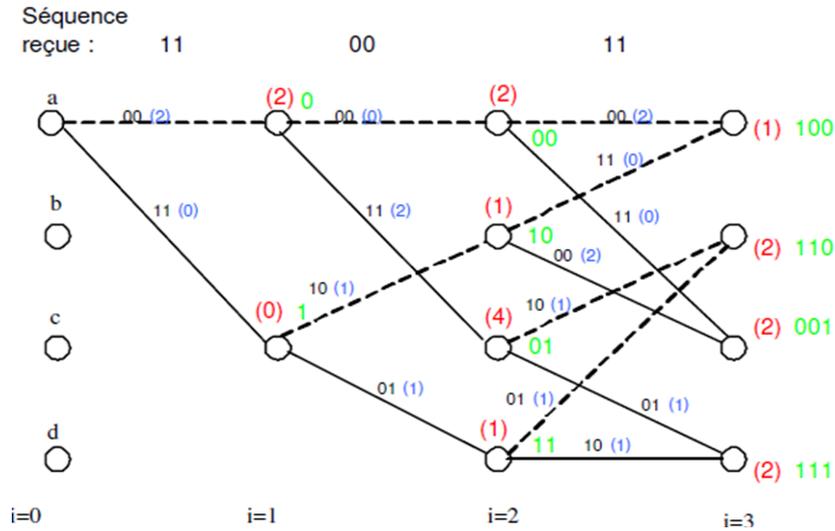
(le chemin, la séquence décodé)

$i = 0$: initialisation du treillis ;

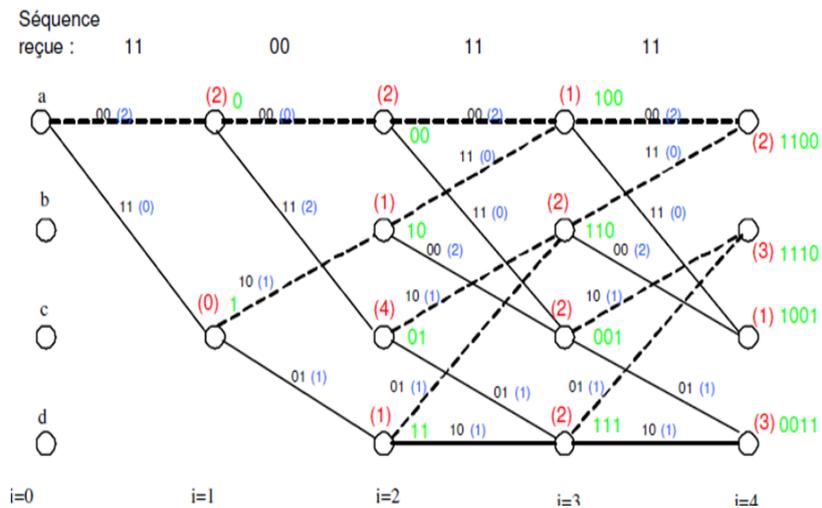
$i = 1$: calcul des métriques de branche ;

$i = 2$: calcul des métriques de branche ;

$i = 3$: A cette instant, il y a au moins deux chemin qui arrive au même état, après le calcul des métriques de branche ; on ne stock que la branche qui a la plus petite métrique et qui sera appelé "la branche survivante" les autres chemines seront abandonnés (elles meurent). Dans le cas ou deux branches ont la même métrique une seule branche sera sélectionné arbitrairement.



$i = 4$: calcul des métriques de branche ; stockage des séquences survivantes



Lorsque la séquence de notre message est décode, les métriques mesurer et présélectionner sont stockées, il nous reste que de choisir une séquence parmi les séquences survivantes la plus probable qui est dans notre cas :

1001

en remarque bien que le message décodé est corrige et il correspond bien a la sequence transmise.

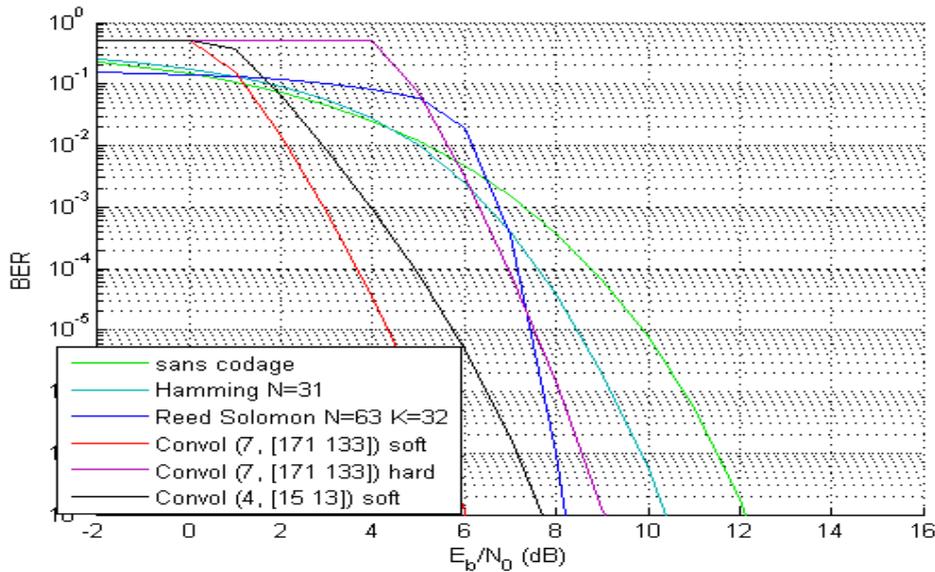


Figure 3.8 – Comparaison des performances entre les codes correcteurs les plus utilisés

Cet algorithme est très utilisé car il permet un décodage à faible coût en termes de mémoire que de nombre d'opérations. Cependant, le parcours du treillis monte rapidement en complexité avec la longueur de contrainte, rendant le décodage par treillis difficile sur des codes à longueur de contrainte élevée.

3.5.2 Simulation

Avant de clôturer ce chapitre, nous allons comparer les performances du point de vue du BER en fonction du SNR entre les codes correcteurs les plus utilisés.

Dans le cas d'un canal à bruit blanc additif et gaussien AWGN, et modulation BPSK la Figure(3.7), montre que le code convolutif d'un polynôme générateur $G(D) = (7, [171 \ 133])$ avec soft décision donne des performances meilleures par rapport au code de Hamming et le code Reed Solomon. Pour une probabilité d'erreur égal à 10^{-5} , le code convolutif apporte un gain de codage de $2.7dB$, et le code Reed Solomon ($N = 63k = 32$) apporte $1.3dB$, tandis que le code de Hamming ($N = 31$) apporte uniquement $0.8dB$ par rapport au cas sans codage. Malgré cela, les performances de ces codes restent toujours médiocres par rapport aux critères exigés dans les communications satellitaires (Les détails sur les turbocodes seront illustrés dans le chapitre suivant).

3.6 Conclusion

Dans ce chapitre, nous avons introduit les différentes classes de code correcteur d'erreurs. puis en a passé par les représentations des codes convolutifs les plus utilisés,

vers la fin en a explique avec un simple exemple le principe du décodage des codes convolutionnels avec l'algorithme de Viterbi.

A la fin de ce chapitre, nous avons présenté quelques résultats sur les performances des codes correcteurs les plus connus. En analysant ces résultats, nous avons constaté que les techniques de codage classique sont très loin de la limite de Shannon et ils donnent de mauvaise résultats pour des valeurs de SNR inférieur a 3 dB . Pour ce qui suit, notre étude sera focalisée sur les turbocodes.

Chapitre 4

Les turbo codes

4.1 Introduction

Les turbo codes ont été découverts par C. Berrou et al. en 1993 [15]. Cette famille de codes offre des grandes performances en puissance permettant de s'approcher de la limite théorique de Shannon [13]. Ils marquent une révolution dans la communauté de la théorie de l'information. Depuis leur introduction, de très nombreux travaux de recherche relatifs aux turbo codes, et plus généralement aux codes concaténés, ont été effectués afin de comprendre et d'analyser leur comportement et d'améliorer leurs performances. Jusqu'à nos jours, aucune explication théorique et mathématique n'arrive à expliquer parfaitement les performances de cette famille des codes.

Coté application, les turbo codes ont été adoptés par quelques standards émergents telles que le DVB-RCS pour les liaisons satellitaires montantes. Des schémas de codage turbo ont été aussi proposés par la comité CCSDS (Consultative Committee for Space Data Systems) pour les applications de télémétrie et de télémétrie spatiales [16]. Cette nouvelle technique de codage de canal qui est le résultat de travaux pratiques n'a pu battre les autres techniques sans le traitement probabiliste dans les récepteurs.

Certains éléments et structures de bases doivent être introduits avant d'aborder les turbocodes eux-même, mis à part les codes convolutifs récurrents qui ont déjà été vu dans le chapitre précédent. Basé sur cela, dans ce chapitre, nous allons décrire le principe d'entrelacement qui est une partie intégrante des turbocodes, le concept de concaténation et les différentes méthodes de fermeture d'un treillis. Finalement, les algorithmes MAP et SOVA largement utilisés dans le processus de décodage des turbocodes, sont présentés.

4.2 Les permutations

Les codes utilisés pour des canaux satellitaires sont conçus pour combattre des erreurs indépendantes et surtout espacées dans le temps [10]. D'autres types de canaux (radio

mobile, téléphonique...) sont soumis à des erreurs qui arrivent en salves.

L'utilisation de la technique d'entrelacement a pour objet d'augmenter la distance minimale de Hamming ce qui permet d'améliorer la capacité de correction du code et de lutter contre les erreurs en salves [17].

Nous pouvons définir ainsi la fonction de permutation comme un processus de réarrangement de l'ordre d'une séquence de données dans un format déterminé [5]. La permutation inverse qui est le processus inverse de la permutation consiste à modifier l'ordre reçu à son ordre original.

considérons maintenant une permutation de la taille N . Nous supposons que la séquence de donnée présente à l'entrée de l'entrelaceur est binaire et donnée par :

$$c = (c_1, c_2, c_3, \dots, c_N) \quad [5] \quad (4.1)$$

où $c_i \in (0, 1), 1 \leq i \leq N$ la séquence présente à la sortie de l'entrelaceur est

$$c^{\sim} = (c_1^{\sim}, c_2^{\sim}, c_3^{\sim}, \dots, c_N^{\sim}) \quad [5] \quad (4.2)$$

où $c_i^{\sim} \in (0, 1), 1 \leq i \leq N$

c^{\sim} est constituée avec les éléments de c mais dans un ordre différent, il existe donc une une-à-une correspondance entre chaque élément de c_i et c_j^{\sim} . Si l'on définit un ensemble A comme $A = \{1, 2, 3, \dots, N\}$, la permutation peut être aussi définie par une fonction d'indexation :

$$\pi(A \rightarrow A) : j = \pi(i) \quad i, j \in A \quad [5] \quad (4.3)$$

où i et j sont respectivement l'index d'un élément dans l'ordre original dans c et l'ordre permuté dans c^{\sim} , la fonction de permutation peut être représentée par un vecteur :

$$\pi_N = (\pi_1, \pi_2, \dots, \pi_N) \quad (4.4)$$

Il existe deux types ou familles d'entrelacement : l'entrelacement par "bloc" et l'entrelacement "Convolutionnel".

4.2.1 l'entrelacement par bloc

L'entrelacement par bloc ou permutation régulière peut être définie sous deux formes différentes. La première suppose que le bloc contenant k bits peut être organisé comme un tableau de M lignes et N colonnes. L'entrelacement consiste alors à écrire la séquence ligne par ligne, et lorsque le tableau est plein de le lire colonne par colonne [4]. le désentrelaceur consiste à construire un tableau de même dimension où l'on écrit la séquence reçue colonne par colonne et on lit le tableau ligne par ligne (voir Figure(??)).

La seconde forme de l'entrelaceur s'applique sans hypothèse sur la valeur de k . Après l'écriture des données dans une mémoire linéaire (adresse $i, 0 \leq i \leq k - 1$), le bloc

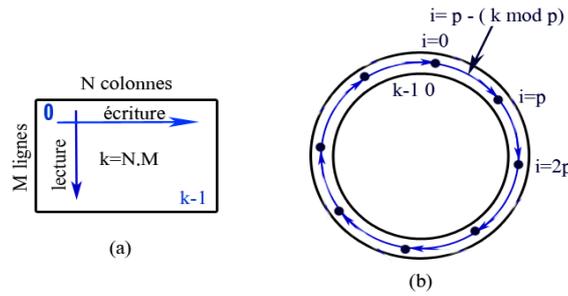


Figure 4.1 – Permutation régulière sous formes rectangulaire (a) et circulaire (b) [4].

est assimilé à un cercle [4], les deux extrémités ($i = 0$ et $i = k-1$) étant alors contigues (Figure(??)). Les données binaires sont alors extraites de telle sorte que la j -ème donnée lue ait été préalablement écrite a la place i , de valeur :

$$i = \Pi(j) = P_j + i_0 \text{ mod } k \quad [4] \tag{4.5}$$

où $k = M.N$, P est un entier premier avec k et i_0 est l'indice de départ

Exemple : Dans l'exemple ci-dessous dans la Figure(4.1), un message quelconque est scindé en paquets de même taille (écriture ligne par ligne)(1), lu en vertical (lecture colonne par colonne)(2) et transmis en horizontal. Les erreurs de transmission sont alors ajoutées et à la réception l'opération inverse est effectuée(4) afin de récupérer le message original mais affecté d'erreurs (7).

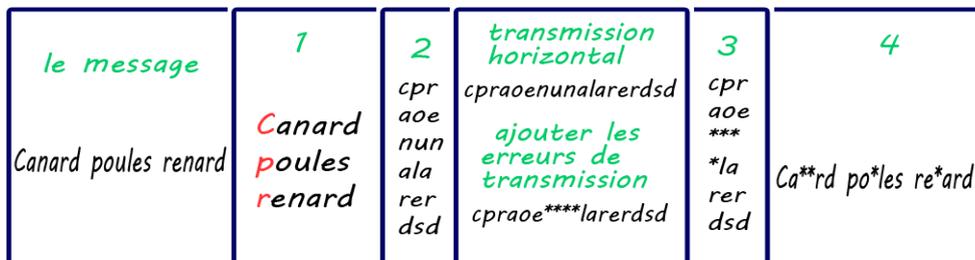


Figure 4.2 – Principe de fonctionnement d'un entrelaceur par bloc [4]

Il est clair dans cette figure que l'entrelacement a permis de répartir les erreurs sur plusieurs paquets. Donc, le code correcteur peut facilement corriger les erreurs générées à cause des effets adverses du canal, et ce à l'échelle de la transmission. Sans entrelacement, l'erreur aurait porté sur des mots complets que l'on n'aurait pas pu récupérer.

4.2.2 l'entrelacement par convolution

Il y a un nombre inimaginable de permutations irrégulières. Dans notre travail nous avons plutôt retenu ce qui est, du moins présentement, le type de permutation à la fois le

plus simple et le plus performant, à savoir les permutations circulaires presque régulières [5]. La séquence de symboles à permuer n'est plus limitée à un bloc, mais elle peut avoir une longueur quelconque. Chaque symbole est aiguillé par un commutateur vers un registre à décalage différent. Un commutateur, synchrone du premier, prélève les symboles sortants des registres pour les aiguiller vers la sortie. Lorsque le nombre des registres est épuisé l'opération recommence. il y' aura L registres dont la longueur variera entre B et $(L-1).B$, plus un registre dégénéré de longueur zéro (une simple connexion)(Figure(??)).

le générateur inverse a exactement la même structure que celle décrite précédemment (Figure(??)), mais avec la longueur des registres prise dans l'ordre décroissant. Bien entendu les commutations doivent être synchrones.

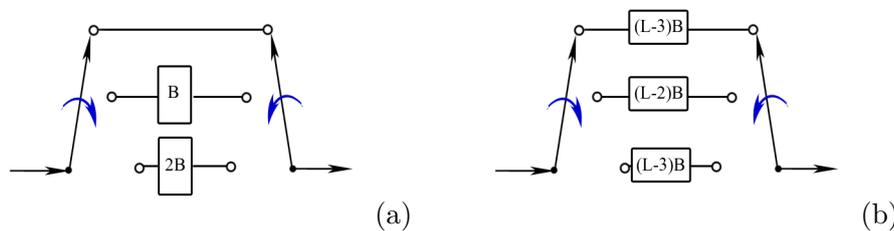


Figure 4.3 – (a)Entrelaceur par convolution, (b)Disentrelaceur par convolution [5]

Remarque : Comme le but de l'entrelacement est de rendre la distribution des erreurs uniforme dans le cadre des codes correcteurs d'erreurs, cela nécessiterait en théorie une profondeur mémoire infinie pour y parvenir [17]. Ainsi un entrelaceur est d'autant plus performant que sa profondeur mémoire est importante au détriment de la complexité et du temps de calcul.

4.3 La concaténation

La plupart des codes élémentaires telles que BCH, Reed-Solomon ou CSRC sont asymptotiquement bons. Leurs distances minimales de Hamming (DMH) peuvent être rendues aussi grandes que l'on veut, en augmentant suffisamment le degré des polynômes générateurs. Le problème est que pour des degrés de polynômes qui garantiraient les DMH requises par les applications pratiques, la complexité des décodeurs correspondants peut devenir prohibitive et rend sa réalisation coûteuse.

Une solution fut proposée par *Forney* durant son travail de thèse en 1965, et qu'il a appelé la concaténation de codes, où il introduisit pour la première fois la notion du codage composite connu aujourd'hui sous le nom de *concaténation série* (CS) [18].

4.3.1 Concaténation série (CS)

La concaténation série est un moyen simple pour obtenir des codes à grande DMH et néanmoins aisément décodables par combinaison de plusieurs codes élémentaires de taille raisonnable, de telle sorte que le code global résultant possède un pouvoir de correction élevé [4].

Le schéma présenté à la Figure(4.3) donne la représentation en bloc du codage composite. Selon cette figure, un premier codeur, dit codeur extérieur, fournit un mot de code qui est ensuite recodé par un deuxième codeur, dit codeur intérieur. Si les deux codes sont systématiques, le code concaténé est lui-même systématique.

Il a été rapporté que le choix le plus judicieux de ces codes constituants serait un code de type algébrique, typiquement Reed-Solomon, pour le code extérieur, et un code convolutif pour le code intérieur. Le décodeur intérieur serait alors un décodeur de Viterbi, qui tire aisément profit des valeurs souples fournies par le démodulateur, tandis qu'un décodeur extérieur, qui opérerait sur des symboles de plusieurs bits (par exemple 8 bits), peut s'accommoder d'erreurs en rafales à la sortie du premier décodeur. Une fonction de permutation ou d'entrelacement insérée entre les deux codeurs, et sa fonction inverse placée entre les deux décodeurs, peuvent augmenter très nettement la robustesse du code concaténé. Un tel schéma de codage a rencontré beaucoup de succès dans des applications aussi variées que les transmissions en espace lointain et la diffusion de télévision numérique, satellitaire et terrestre. C'est en fait le schéma de codage adopté dans de nombreux pays pour la télévision numérique terrestre [6.2].

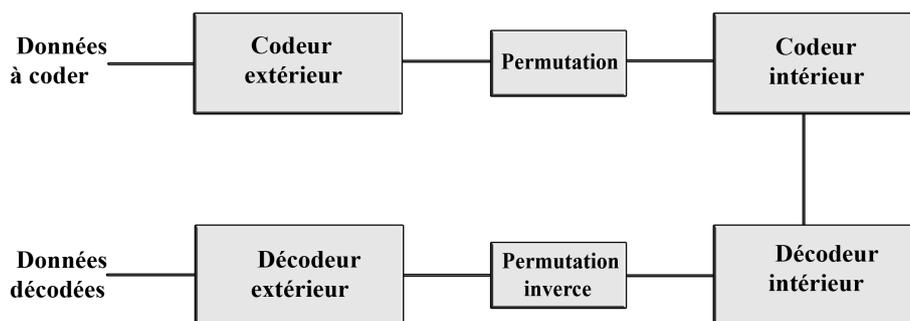


Figure 4.4 – Schéma bloc type de codes concaténés série [4]

Cependant, l'inconvénient majeur d'une concaténation de type série est que son décodage. En effet, même si, localement, les deux décodeurs élémentaires sont optimaux, la mise en série simple de ces deux décodages n'est pas globalement optimale, le décodeur intérieur ne tirant pas profit de la redondance produite par le code extérieur. D'où l'importance de la concaténation parallèle

4.3.2 La concaténation parallèle (CP)

Une nouvelle forme de concaténation, dite concaténation parallèle (CP), a été introduite au début des années 1990 pour élaborer les turbocodes [19]. La Figure(4.4) représente une CP de dimension 2, qui est la dimension classique des turbocodes. Dans ce schéma, le message est codé deux fois, dans son ordre naturel et dans un ordre permuté. La partie redondante du mot de code est formée par la réunion des sorties redondantes des deux codeurs. La CP diffère de la CS par plusieurs aspects, détaillés dans la section suivante. En se limitant à la dimension 2, la concaténation qui associe deux codes de rendement

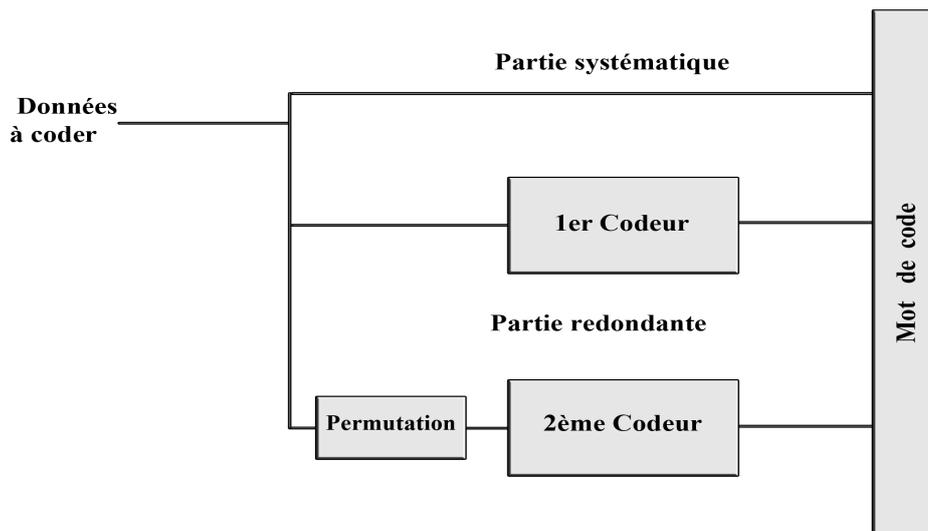


Figure 4.5 – Schéma bloc type de codes concaténés parallèle [4]

élémentaire R_1 (code C_1) et R_2 (code C_2), a un rendement global de codage :

Dans le cas d'une concaténation série :

$$R_s = R_1 \cdot R_2 \quad (4.6)$$

Pour la concaténation parallèle :

$$R_p = \frac{R_1 \cdot R_2}{R_1 + R_2 - R_1 \cdot R_2} = \frac{R_1 \cdot R_2}{1 - (1 - R_1) \cdot (1 - R_2)} \quad (4.7)$$

Pour des valeurs identiques de R_1 et R_2 , le rendement global R_s est plus élevé. On en déduit aussi que lorsque la dimension du code composite augmente, l'écart entre R_p et R_s augmente également. Si on prend en exemple, trois codes élémentaires de rendement $1/2$, ils formeront un code concaténé de rendement global $1/4$ en parallèle et $1/8$ en série. Il n'est donc pas avantageux de porter la dimension d'un code concaténé en série au-delà de 2, sauf pour des rendements très proches de l'unité. Et même si la concaténation parallèle n'apporte pas d'augmentation du pouvoir de correction, elle induit une réduction de la

complexité de réalisation car à performance égale, elle manipule des codes élémentaires de contraintes moindres. De plus, les décodeurs sont moins importants, les matrices d'entrelacement sont de dimension plus réduite et les paquets d'erreurs produits sont de plus petite taille [3].

4.4 Fermeture du treillis

Pour un turbocode, si nous ne faisons rien de particulier concernant les états terminaux, les informations situées à la fin du bloc, aussi bien dans l'ordre naturel que dans l'ordre permuté, seront moins bien protégées. Ceci conduira à une diminution du gain asymptotique, une dégradation, fonction de la taille du bloc, qui peut être tolérée dans certaines applications, mentionnons-le. Il est à noter que la non-fermeture des treillis pénalise plus fortement le TEP (Taux d'Erreurs de Paquets) que le TEB [4]. C'est la raison pour la quelle nous devons prendre en compte la fermeture des deux treillis en adoptant diverses solutions dont :

4.4.1 A l'aide de bits de bourrage

Les standards CCSDS et UMTS utilisent cette technique. Les bits assurant la fermeture d'un des deux treillis ne sont pas utilisés dans l'autre codeur [4]. Ces bits ne sont donc pas turbocodés ce qui conduit mais dans une moindre mesure, aux mêmes inconvénients que ceux présentés dans le cas précédent. De plus, la transmission des bits de fermeture entraîne une diminution du rendement de codage et donc, de l'efficacité spectrale.

4.4.2 Un entrelacement permettant une fermeture automatique du treillis

Il est possible de fermer automatiquement [4], sans ajout de bits de fermeture, le treillis d'un turbocode en transformant légèrement le schéma de codage (auto-concaténation) et en utilisant un entrelacement respectant certaines règles de périodicité. Cette solution ne diminue pas l'efficacité spectrale mais impose des contraintes sur l'entrelacement qui rendent difficile la maîtrise des performances à faibles taux d'erreurs.

4.4.3 Un codage circulaire

un codeur de code convolutif circulaire garantit que l'état initial et l'état final du registre sont identiques [4]. Le treillis prend alors la forme d'un cercle ce qui, du point de vue du décodeur, peut être considéré comme un treillis de longueur infinie. Ce procédé de fermeture, déjà connu sous le nom de *tail-biting* pour les codes non récursifs, offre deux avantages majeurs :

- Contrairement à d'autres techniques, la fermeture circulaire ne présente aucun effet de bord, tous les bits du message sont protégés de la même manière et tous sont

doublément codés par le turbocode. Il n'y a donc pas lieu, lors de la conception de la permutation, d'accorder une importance particulière à tel ou tel bit, ce qui conduit à des modèles de permutation plus simples.

- Les séquences qui ne sont pas RTZ ont une influence sur l'ensemble du cercle : un symbole de parité sur deux, en moyenne, est modifié tout le long du bloc. Pour des valeurs typiques de k (quelques centaines ou plus), le poids de sortie correspondant est donc très élevé et ces motifs d'erreur ne contribuent pas à la DMH du code, comme on l'a déjà mentionné à la fin de la section précédente. Sans terminaison ou avec une terminaison par bits de bourrage, seule la partie du bloc après le début de la séquence non RTZ a une incidence sur les symboles de parité.

À ces deux avantages, s'ajoute bien sûr l'intérêt de n'avoir à transmettre aucune information supplémentaire sur la terminaison et donc de ne rien perdre en efficacité spectrale. La technique de terminaison circulaire a été retenue dans les standards DVB-RCS et DVB-RCT [4].

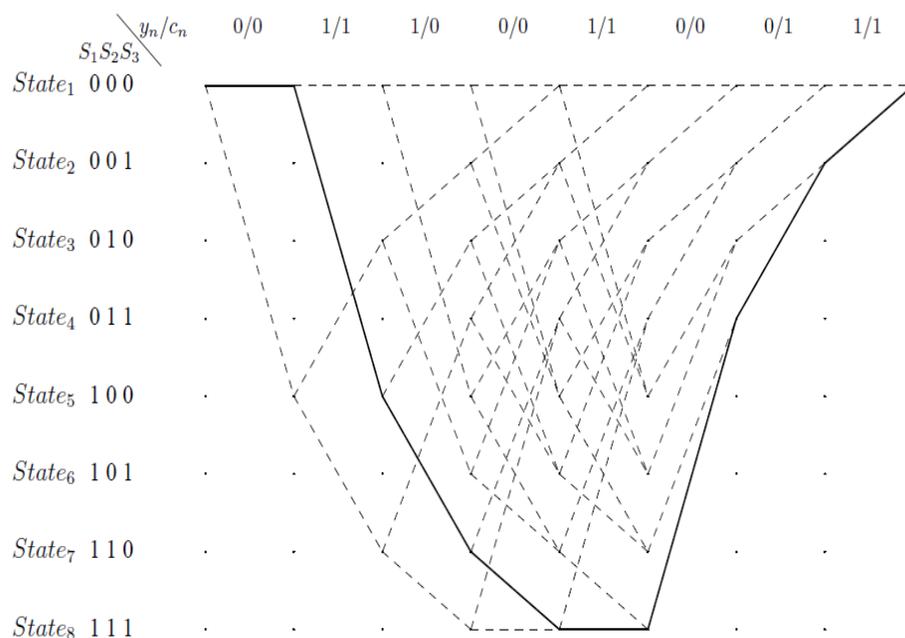


Figure 4.6 – Exemple d'une séquence de transition avec fermeture par bits de bourrage [6]

4.5 Turbo codage

La Figure(4.4) représente un exemple d'un turbo codeur standard de rendement $R = 1/3$. Un turbo codeur résulte de la concaténation de deux codeurs élémentaires récursifs systématiques (RSC ou RII) C_1 et C_2 , qui ont la même taille de contrainte k et le même polynôme générateur $G(D)$. Les deux codeurs RSC C_1 et C_2 reçoivent le même

message informatif $\{d_k\}$ mais les séquences sont arrangées dans un ordre différent par le biais de l'entrelaceur Π_1 .

Lorsque l'on injecte à l'entrée de notre turbo codeur la séquence d_k , On reçoit à $t = k$ (t : temps), à la sortie du turbocodeur, puisqu'il est systématique, la séquence d'entrée d_k suivie des séquences C_k^1 et C_k^2 , générées des codeurs élémentaires C_1 et C_2 , respectivement [20].

4.6 Turbo décodage

4.6.1 Principe de Turbo Décodage

Le principe de décodage d'un turbocode est illustré dans Figure(4.6). Il contient deux décodeurs SISO en parallèle [6]. Chaque décodeur reçoit deux informations : une *information intrinsèque* issue du canal de transmission, et une *information extrinsèque* pour la correction. Le résultat est une sortie souple des bits décodés. La sortie souple est représentée par ce que l'on appelle les *logarithmes de rapport de vraisemblance* (LLRs : Log-Likelihood Ratio). Le décodage est effectué itérativement. Le décodeur DEC1, à entrée souple sortie souple, fait un décodage et envoie le résultat au deuxième décodeur DEC2 passant par le désentrelaceur. DEC2 exploite la valeur reçue de DEC1 (information extrinsèque) et la valeur reçue du canal (l'information intrinsèque) pour décoder l'information. A son tour DEC2 envoie le résultat à DEC1 à travers un entrelaceur. DEC1 utilise l'information reçue du DEC2 et l'information reçue du canal pour le décodage, la sortie est renvoyé à DEC2, et ainsi de suite. Le taux d'erreur binaire diminue en augmentant le nombre d'itérations.

Généralement l'opération se termine lorsque l'erreur entre deux sorties souples successives est presque constante.

Dans la figure ci-haut, les notations suivantes sont adoptées :

- \tilde{a}^c : la séquence de bits systématiques reçue ;
- \tilde{b}^c : la séquence de bits systématiques entrelacée ;
- \tilde{c}^c : la séquence de bits reçue codée par le codeur 1 ;
- \tilde{d}^c : la séquence de bits reçue codée par le codeur 2 ;
- \tilde{e}^c : la séquence de bits de fermeture du codeur 1 ;
- \tilde{f}^c : la séquence de bits de fermeture du codeur 2.

Chaque décodeur reçoit deux informations : la première étant la décision souple de la séquence codée, qui est reçue directement du canal (\tilde{c}^c , \tilde{d}^c), l'autre étant la séquence décodée par l'autre décodeur, qui est l'addition de l'information à priori fournie par le deuxième décodeur et la séquence systématique reçue. L'information à priori est

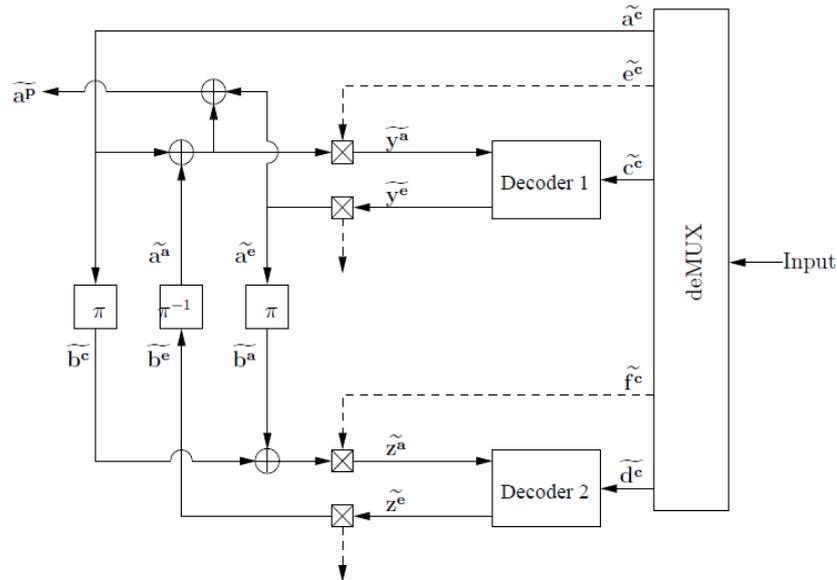


Figure 4.7 – Algorithme de décodage du Turbo code [6]

l'information extrinsèque générée par le deuxième décodeur après un entrelacement ou désentrelacement.

Dans la figure ci-haut, le décodeur DC1 reçoit à son entrée $y^{\sim a}$ du deuxième décodeur, qui est la somme des deux séquences $a^{\sim c}$ et $a^{\sim a}$ suivie par la séquence $e^{\sim c}$. L'ajout de la séquence de bits de fermeture est indispensable pour que le décodeur opère de façon adéquate, mais cette séquence n'est pas échangée entre les décodeurs, étant spécifique à chaque décodeur. D'un autre côté l'information extrinsèque $y^{\sim e}$ générée par le décodeur DC1 est transmise au décodeur DC2, après la suppression de la séquence de terminaison $e^{\sim c}$ et l'entrelacement de la séquence d'information. Le décodeur DC2 effectue les mêmes opérations que le décodeur DC1, il reçoit à son entrée l'information extrinsèque $z^{\sim a}$ qui est la somme de l'information reçue du décodeur DC1 entrelacée $b^{\sim a}$ ($a^{\sim e}$ entrelacé) et l'information systématique entrelacée $b^{\sim c}$ suivie par la séquence de bits de fermeture $f^{\sim c}$. L'information extrinsèque générée par le deuxième décodeur $z^{\sim e}$ est transmise au premier décodeur après la suppression de la séquence de bits de fermeture et le désentrelacement. Dans la première itération $b^{\sim e}$ est initialisé par 0 ce qui signifie que l'information est totalement inconnue, et les valeurs sont équiprobables. Par conséquent, $a^{\sim a}$ est aussi initialisé automatiquement à 0. La seule information extrinsèque considérée est donc l'information systématique $a^{\sim c}$. Dans la figure précédente deux entrelaceurs et un désentrelaceur sont utilisés entre les deux décodeurs pour obtenir le bon ordre de séquence à l'entrée de chaque décodeur.

L'algorithme de MAP utilise les informations extrinsèques générées par les décodeurs. L'information a posteriori est obtenue par la somme de l'information a priori $a^{\sim a}$ et

l'information extrinsèque y^e du décodeur DC1. A la fin des itérations la décision ferme est prise en basant sur la décision souple à la sortie.

4.6.2 Logarithme de rapport de vraisemblance

Chaque décodeur doit envoyer une décision souple pondérée. Le *logarithme de rapport de vraisemblance* (LLR : Log-Likelihood Ratio) est la valeur appropriée associée à la décision de chaque bit décodé. Les LLRs sont échangées pour améliorer la fiabilité de la décision dans l'opération de décodage itératif.

L'utilisation du logarithme a deux avantages principaux Li09. Premièrement, l'algorithme MAP utilise un bon nombre de multiplications, ce qui fait que le circuit implémenté résultant est complexe. L'utilisation du logarithme MAP (Log MAP) par ailleurs transforme les multiplications au domaine logarithmique, où elles deviennent des sommes. Deuxièmement la gamme des valeurs des décisions souples peut être grand voire illimité (théoriquement), qui implique une utilisation d'une mémoire de grande capacité. L'utilisation du Log MAP réduit la gamme de ces valeurs. Par conséquent, le temps de décodage est réduit en utilisant des opérations simples, et peu de mémoires.

Les opérations de base dans l'algorithme MAP sont l'addition et la multiplication. La multiplication dans le domaine logarithmique revient à une addition :

$$\ln(a.b) = \ln(a) + \ln(b) = A + B \quad (4.8)$$

L'addition peut être résolue par le logarithme Jacobien, qui est définie comme suit :

$$\ln(a + b) = \max(a, b) + \ln(1 + \exp^{-|a-b|}) \quad (4.9)$$

Dans la pratique, le terme $(\ln(1 + \exp^{-|a-b|}))$ est implémenté dans une table dans la mémoire. Cette fonction peut être réalisée par une sélection de valeurs dans ce tableau. Le décodeur utilise les probabilités à priori reçues du deuxième décodeur (qui sont nulles initialement), et les probabilités à posteriori reçues du canal de transmission pour améliorer la décision.

Le logarithme de rapport de vraisemblance LLR d'un bit décodé (u_k), est défini comme :

$$L(u_k) = \ln \frac{P_r \{u_k = 1\}}{P_r \{u_k = 0\}} \quad (4.10)$$

où $P_r\{d_k = i\}$, $i = 0, 1$, est la probabilité à priori du bit u_k [15]. On peut calculer les probabilités pour que $u_k = 0, 1$ sachant que $P_r\{u_k = 1\} = 1 - P_r\{u_k = 0\}$ en fonction de $L(u_k)$.

$$P(u_k = i) = \frac{e^{i.L(u_k)}}{1 + e^{L(u_k)}} \quad i = 0, 1 \quad (4.11)$$

On définit aussi les LLR basés sur les probabilités conditionnelles. Le premier LLR est le rapport des probabilités à posteriori du bit $u_k = 0, 1$ conditionné par la séquence y_k .

$$L(u_k/y_*) = \ln \frac{P(u_k = 1/y_*)}{P(u_k = 0/y_*)} \quad (4.12)$$

Ce LLR est fournit par le décodeur SISO utilisé dans le décodage du Turbo-code. Le deuxième LLR d'intérêt est le rapport des probabilités à posteriori du bit reçu y_k conditionné par le bit transmis $x_k = 0, 1$ et donné par

$$L(y_k/x_k) = \ln \frac{P(y_k/x_k = 1)}{P(y_k/x_k = 0)} \quad (4.13)$$

Ce LLR est la valeur reçue du canal de transmission. La probabilité à posteriori du bit reçu y_k sachant le bit transmis $x_k = 0, 1$ dans un canal Gaussien utilisant une modulation BPSK est :

$$P(y_k/x_k = \pm 1) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-E_b}{2\sigma^2}(y_k - (\pm 1))^2\right) \quad (4.14)$$

Le LLR dans (4.13) devient donc :

$$L(y_k/x_k = \pm 1) = \frac{E_b}{2\sigma^2} 4 y_k = L_c \cdot y_k \quad [21] \quad (4.15)$$

Donc le LLR est le produit de y_k à la sortie du canal et L_c qui est connu sous le nom de la fiabilité du canal.

4.6.3 L'algorithme de maximum probabilité à posteriori

En 1974, BAHL et al. ont introduit un nouvel algorithme qui est connu sous le nom Maximum probabilité à posteriori . Leur travail est l'estimation de la probabilité a posteriori des états de transitions d'une source de Markov observée à travers un canal discret sans mémoire. Cet algorithme est appliqué au turbocode pour réduire au minimum le taux d'erreur. La différence entre cet algorithme et l'algorithme de Viterbi, est que l'algorithme de Viterbi prend le maximum de vraisemblance pour minimiser l'erreur de la séquence de bit entièrement, tandis que l'algorithme MAP prend le maximum de vraisemblance de chaque bit avec les informations à priori, pour choisir le chemin de décodage dans le treillis. D'après [22], cet algorithme n'est pas un bon concurrent à l'algorithme de Viterbi à cause de sa complexité. Mais il est devenu intéressant avec les turbocodes, ou il a donné de bons résultats après quelques modifications [15]. En pratique on utilise l'algorithme MAP approximé.

L'algorithme MAP consiste à calculer trois valeurs essentielles, pour fournir la probabilité du bit (égal à 1 ou -1) si la séquence reçue est y_* . ceci est équivalent à calculer le LLR $L(u_k/y_*)$. La probabilité de $P(u_k = +1)$ est la somme des probabilités de toutes les transitions possibles T de l'état s_{i-1} à l'état s_i du treillis quand un bit +1 est envoyé.

De façon analogue, la probabilité de $P(u_k = -1)$ est la somme des probabilités de tous les transitions possibles T de l'état s_{i-1} à l'état s_i quand un bit -1 est envoyé. On peut écrire donc le LLR comme suit :

$$L(u_k/y_*) = \ln \frac{\sum_{((s',s),u_k=+1)} P(S_{(k+1)} = s' \wedge S_k = s \wedge y_*)}{\sum_{((s',s),u_k=-1)} P(S_{(k+1)} = s' \wedge S_k = s \wedge y_*)} \quad (4.16)$$

On peut diviser la séquence à traiter en trois sections : la transition présente, la séquence reçue avant la transition présente, et la séquence reçue après la séquence présente. Avec la règle de Bayes, on trouve après calculs et simplifications :

$$\sigma_k(s', s) = P(S_{(k+1)} = s'_k = s_*) = P(y_{j>k} / s).P(y_k \wedge s / s').P(s' \wedge y_{j<k}) \quad (4.17)$$

$$P(s' \wedge s \wedge y_*) = \beta_k(s) \cdot \gamma_k(s', s) \cdot \alpha_{k-1}(s') \quad [22] \quad (4.18)$$

α , est la métrique réursive en avant, définie par la probabilité [22] :

$$\alpha_{k-1}(s') = P(S_{k-1} = s' \wedge y_{*j<k}) \quad (4.19)$$

Elle correspond à la probabilité que le treillis est à la position s' à l'instant $k - 1$, $y_{*j<k}$ dans l'équation précédente étant la séquence reçue du canal avant cet instant.

β , est la métrique réursive en arrière, définie par la probabilité [22] :

$$\beta_k(s') = P(y_{*j>k} / S_k = s) \quad (4.20)$$

C'est la probabilité que la séquence future à recevoir est $y_{*j>k}$, sachant que s est l'état du treillis à l'instant k .

γ , est la métrique de branche définie par la probabilité [22] :

$$\gamma_k(s', s) = P[(y_k \wedge S_k = s) / S_{k-1} = s'] \quad (4.21)$$

C'est la probabilité que le treillis se déplace vers s à l'instant k et la séquence reçue est y_k , sachant que l'état précédent est s' .

La sortie du décodeur est donnée en calculant les valeurs LLR. Si cette valeur est positive la décision est un 1, sinon elle est nulle. L'équation qui donne les LLR en fonction des métriques présentées préalablement :

$$L(u_k) = L^e(u_k) + L(y_k / x_k = \pm 1) + L(u_k = \pm 1 / y_*) \quad (4.22)$$

Le premier terme est la valeur a priori du bit décodé reçu du deuxième décodeur. Pour la première itération, comme aucune information n'est disponible, elle est égale à 0 (équiprobabilité $p=0.5$). Le deuxième terme est le LLR du canal et donne une indication sur le SNR du canal. Il est constant et ne change pas d'une itération à une autre, car le

terme L_c (la capacité du canal) et la séquence de bit systématiques sont constants. La troisième valeur est le LLR à posteriori, et doit être calculé pour chaque itération. Donc on peut écrire que :

$$L(u_k) = L_{\text{a priori}} + L_{\text{canal}} + L_{\text{a posteriori}} \quad (4.23)$$

A chaque itération, seulement le $L_{\text{a posteriori}}$ est calculé. A noter que le $L_{\text{a posteriori}}$ à la sortie du décodeur est le $L_{\text{a priori}}$ à l'entrée du deuxième décodeur (Figure 4.7).

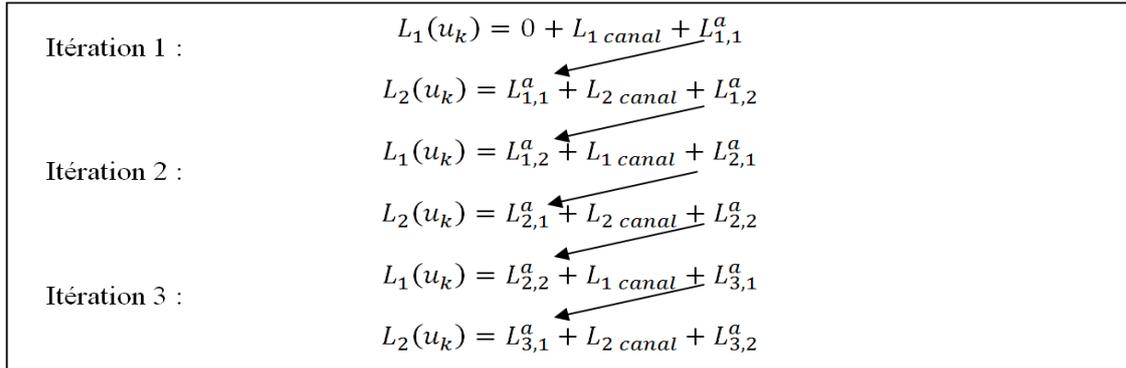


Figure 4.8 – Principe des itérations et de calcul des L a posteriori

Il s'en suit que dans le processus de décodage, le $L(u_k)$ n'est pas calculé à chaque itération, seul le terme $L_{\text{a posteriori}}$ est calculé jusqu'à un nombre défini d'itérations après quoi le $L(u_k)$ est finalement calculé.

Les valeurs α_k et β_k , sont calculées d'une manière récursive.

a- La métrique récursive en avant α_k :

α_k est calculé en utilisant la dernière métrique et la métrique de branche complète.

$$\alpha_k = \sum_s \alpha_{k-1}(s') \cdot \gamma_k(s', s) \quad (4.24)$$

Les conditions aux limites pour $k = 0$ sont $\alpha_0(0) = 1$ et $\alpha_0(m) = 0$ pour $m \neq 0$. Cela découle du fait que le codage commence toujours par l'état 1, et donc on initialise ces valeurs de telle sorte à signifier que le seul état possible est 1.

Les autres métriques sont calculées récursivement pour chaque état. A chaque fois, avant de passer à la métrique suivante, on normalise la métrique calculée pour éviter le dépassement de gamme. Pour normaliser on divise par la somme des métriques récursives en avant et l'équation suivante est obtenue :

$$\alpha_k^{\sim}(s) = \frac{\sum_{s'} \alpha_{k-1}^{\sim}(s') \cdot \gamma_k(s', s)}{\sum_s \sum_{s'} \alpha_{k-1}^{\sim}(s') \cdot \gamma_k(s', s)} \quad (4.25)$$

b- La métrique réursive en arrière β_k :

Elle est calculée d'une manière réursive en commençant par la fin. On suppose aussi que l'état final est 1, puisque le codage a utilisé les bits de fermeture pour forcer le treillis à zéro, d'où $\beta_\tau(0) = 1$, et $\beta_\tau(m) = 0$ pour $m \neq 0$, τ étant la longueur du treillis.

Les autres métriques sont calculées réursivement pour chaque état en commençant par la fin. Rappelons qu'avant de passer à la métrique suivante, on devra normaliser la métrique calculée, en divisant par la somme des métriques réursives en avant. L'équation résultante est la suivante :

$$\beta_{k-1}^{\sim}(s) = \frac{\sum_{s'} \beta_k^{\sim}(s') \gamma_k(s', s)}{\sum_s \sum_{s'} \alpha_{k-2}^{\sim}(s') \gamma_{k-1}(s', s)} \quad (4.26)$$

c- La métrique de branches γ_k :

Pour calculer les métriques réursives il faut calculer d'abord la métrique de branche complète γ . Elle est définie comme étant la corrélation entre le signal reçu et les valeurs du treillis. Si les valeurs reçues sont identiques (même signe) à celles du treillis la corrélation sera donc élevée. Pour chaque décodeur deux métriques sont évaluées : une métrique de branche complète qui contient les bits systématiques ainsi que les bits codés, et une métrique de branche partielle qui contient seulement la séquence de bits partielle.

c.1- La métrique de branche complète :

Elle est définie par l'équation :

$$\gamma(s', s) = \exp\left(\frac{1}{2} \cdot L^e(u_k) \cdot c_k^1 + L_c \cdot \frac{1}{2} \cdot y_k^{1,s} \cdot c_k^1\right) \exp\left(\sum_{i=2}^q L_c \cdot \frac{1}{2} \cdot y_k^{1,p} \cdot c_k^i\right) \quad (4.27)$$

On utilise ici, L^e la probabilité à priori, les valeurs du treillis de chaque branche c_k^i , les bits systématiques et de parité reçus $y_k^{1,s}$ $y_k^{1,p}$, et la capacité du canal L_c . Les valeurs du treillis sont les valeurs de sortie qui correspondent à chaque transition. Ces valeurs sont les bits systématiques et les bits de parité de chaque codeur.

c.2- La métrique de branche partielle :

Elle est basée sur les bits de parité et est utilisée dans le calcul des LLR extrinsèques.

$$\gamma^e(s', s) = \exp\left(\sum_{i=2}^q L_c \cdot \frac{1}{2} \cdot y_k^{1,p} \cdot c_k^i\right) \quad (4.28)$$

Après avoir calculé les valeurs des métriques de branches et des états, les probabilités à postériori sont calculées. Ces probabilités, comme mentionné avant, sont des produits de trois valeurs. On peut les noter par σ_k .

$$\sigma_k(s', s) = \alpha_{k-1}^{\sim} \cdot \gamma_k^e(s', s) \cdot \beta_k^{\sim} \quad (4.29)$$

$$L(u_k = \pm 1/y_*) = \ln \frac{\sum_{u=+1}(\sigma_k(s', s))}{\sum_{u=-1}(\sigma_k(s', s))} \quad (4.30)$$

Pour résumer, les étapes de l'algorithme sont :

1. Initialisation des valeurs $\alpha(0)$, et $\beta(n)$ à 0, (équiprobabilité) ;
2. Calcul des valeurs de $\gamma_k(s', s)$ et de $\gamma_k^e(s', s)$ selon les équations (4.27) et (4.28) ;
3. Calcul des valeurs de $\alpha_{k-1}(s')$ selon l'équation (4.25) ;
4. Calcul des valeurs de $\beta_k(s)$ selon l'équation (4.26) ;
5. Calcul des valeurs $L(u_k = \pm 1/y_*)$ selon l'équation (4.30) en utilisant l'équation (4.29) ;
6. Calcul de l'erreur à chaque itération et comparaison avec celle de l'itération précédente, si elle converge le processus d'itération se termine ;
7. Si le nombre d'itération est défini le processus se termine après avoir complété le nombre d'itération ;
8. on calcule la valeur de sortie finale en utilisant l'équation (4.22)

La décision est prise à partir du LLR de sortie après l'étape 8 via une décision dure.

4.6.4 L'algorithme SOVA

L'algorithme de Viterbi, est modifié pour donner à la sortie non seulement la séquence des états qui a la grande probabilité, mais aussi les probabilités a posteriori pour chaque ou la valeur de fiabilité. Cet algorithme modifié appelé Soft Output Viterbi Algorithm (SOVA), est conçu pour la première fois pour l'utiliser dans plusieurs applications comme le décodage des codes à concaténation, comme un décodeur interne (inner decoder) [23]. La sortie du décodeur est souple, ce qui augmente la performance du décodeur.

Cet algorithme a deux modifications qui permet d'être exploitable par le décodeur (SISO) du turbo décodeur. Premièrement la métrique du chemin est modifiée pour tenir compte de l'information à priori quand on sélectionne le chemin de maximum de vraisemblance (ML) à travers le treillis.

Deuxièmement l'algorithme est modifié pour fournir des sorties soft sous la forme de LLR

$$L(u_k/y^*)$$

pour chaque bit décodé. On définit la probabilité que la séquence d'états jusqu'à l'instant k est correct à travers le treillis est donnée par :

$$P(s_k^s/y_{j \leq k}) = \frac{P(s_k^s \wedge y_{j \leq k})}{P(y_{j \leq k})} \quad (4.31)$$

La probabilité $P(y_{j \leq k})$ est constante pour tous les chemins à travers treillis jusqu'à l'instant k .

La probabilité $P(s_k^s / y_{j \leq k})$ que le chemin est correct à travers le treillis est proportionnelle à la probabilité $P(s_k^s \wedge y_{j \leq k})$. Donc elles sont proportionnelles. On peut écrit :

$$P(s_k^s \wedge y_{j \leq k}) = P(s_{k-1}^{s'} \wedge y_{j \leq k-1}) \cdot P(S_k = s \wedge y_k / S_{k-1} = s') \quad (4.32)$$

On définit $M(s_k^s)$ la métrique de chemin pour s_k^s , où :

$$M(s_k^s) = \ln \left(P(s_k^s \wedge y_{j \leq k}) \right) = M(s_{k-1}^{s'}) + \ln \left(P(S_k = s \wedge y_k / S_{k-1} = s') \right) \quad (4.33)$$

On peut constater que le deuxième terme est la métrique de branche utilisée dans l'algorithme MAP. L'équation peut s'écrire donc :

$$M(s_k^s) = \ln \left(P(s_k^s \wedge y_{j \leq k}) \right) = M(s_{k-1}^{s'}) + \ln \left(\gamma_k(s', s) \right) \quad (4.34)$$

On a :

$$\ln \left(\gamma(s', s) \right) = \left(\frac{1}{2} \cdot L^e(u_k) \cdot c_k^1 + \frac{1}{2} \cdot L_c \cdot y_k^{1,s} \cdot c_k^1 \right) + \left(\sum_{i=2}^q \left(\frac{1}{2} \cdot L_c \cdot y_k^{1,p,i} \cdot c_k^i \right) \right) \quad (4.35)$$

$$\ln \left(\gamma(s', s) \right) = \left(\frac{1}{2} \cdot L^e(u_k) \cdot c_k^1 \right) \cdot L_c \cdot \frac{1}{2} \cdot \left(\sum_{i=1}^q \left(y_k^{1,i} \cdot c_k^i \right) \right) \quad (4.36)$$

La métrique devient :

$$M(s_k^s) = M(s_{k-1}^{s'}) + \left(\frac{1}{2} \cdot L^e(u_k) \cdot c_k^1 \right) \cdot \left(L_c \cdot \frac{1}{2} \cdot \sum_{i=1}^q \left(y_k^{1,i} \cdot c_k^i \right) \right) \quad (4.37)$$

La métrique dans l'algorithme SOVA est mise en forme comme dans le cas de l'algorithme de Viterbi, avec le terme additionnel $\left(\frac{1}{2} L^e(u_k) c_k^1 \right)$ qui est l'information à priori prise en compte.

Dans un treillis binaire, on a deux chemins pour atteindre l'état à l'instant k , s_k^s et \hat{s}_k^s , qui les deux métriques de chemin respectivement $M(s_k^s)$ et $M(\hat{s}_k^s)$. Pour déterminer le chemin le plus correct (le plus probable) on calcul la différence de métrique Δ_k^s :

$$\Delta_k^s = M(s_k^s) - M(\hat{s}_k^s) \quad (4.38)$$

Si $\Delta_k^s < 0$ donc le chemin s_k^s est le chemin le plus probable, sinon le chemin \hat{s}_k^s , est le plus probable. La sortie sera donc une sortie souple qui est le LLR du bit u_k . Δ_k^s est la valeur du LLR du bit u_k .

Cet algorithme est l'algorithme moins complexe que l'algorithme de MAP, mais sa performance n'est pas mieux que celle de l'algorithme de MAP [21].

4.7 Conclusion

Dans ce chapitre, nous avons mis la lumière sur le Turbo codeur qui est constitué de codeurs convolutionnels récurrents et systématiques. Nous avons aussi décrit les conditions ou les hypothèses qui nous serviront comme base de départ de nos simulations. Le principe d'entrelacement bloc classique et aléatoire a été décrit. Comme nous l'avons signalé dans la description du système de communications : un système de codage ou un codeur ne peut pas fonctionner seul. Nous avons donc besoin d'un décodeur d'où la nécessité d'un algorithme de décodage qui pourrait remplir cette tâche. L'algorithme Log MAP est l'algorithme par excellence qui est désigné ou choisi par la majorité des chercheurs pour le décodage des Turbo codes et on a expliqué pour quoi on a choisi cet algorithme dans notre travail.

Dans le suivant chapitre nous allons démontrer pratiquement avec les simulations l'influence du nombre d'itération, le choix du polynôme générateur et le type de permutation sur les performances de notre encodeur.

Chapitre 5

Simulations et résultats

5.1 Conditions et paramètres de la simulation

Dans notre étude, on a simulé une chaîne de communication et ses différents blocs comme il est décrit dans la figure(5.1), pour qu'on puisse évaluer par la suite les performances de notre codeur de canal (turbo-code). Nos programmes qui étaient inspirés des scripts Matlab créés par Robert G. Maunder sous une Licence GNU ont été implémentés sur MatLab.

Dans le cas d'une communication réelle le message informatif est très long et on ne peut pas attendre jusqu'à ce que l'acquisition du message sera terminée pour le traiter, nous allons procéder au traitement par bloc, des blocs qui auront la même taille que celle du entrelaceur, une fois ce bloc est rempli nous allons le traiter d'une façon simultanée. Dans le cas de nos simulations comme nous ne disposons pas d'une base de données nous faisons appel à une fonction MatLab (randi) qui génère un vecteur (contient que 0 et 1) aléatoire gaussien de taille similaire à celle de l'entrelaceur que nous allons l'adopter comme le message informatif. un codeur de canal qui fonctionne avec un turbo-code, un modulateur BPSK, Canal satellitaire modélisé par un canal AWGN. Dans la partie réception, on a un démodulateur BPSK et un décodeur itératif SISO. Le codeur utilise deux blocs RSC concaténés en parallèle, un entrelaceur sépare les deux blocs. Le décodeur utilise l'algorithme Log-MAP. La condition d'arrêt de la simulation (génération et transmission des messages) pour chaque valeur du SNR est d'atteindre au moins 100 erreurs pour un message d'une taille minimale de 10000 bits. si la taille du message transmis atteint les 10000 bits et nous n'enregistrons pas 100 erreurs, nous continuerons à générer et transmettre des blocs de messages jusqu'à la satisfaction de cette dernière condition. Les paramètres qui identifient notre codeur sont :

- un code de rendement $R = 1/3$ (dans toutes les simulations) ;
- La taille de bloc N : on a pris pour notre étude $N=40,512$ et 1024 bits ; ($N=40$ la référence)
- Le Polynôme générateur : on a choisi le polynôme de la norme UMTS $G(D) =$

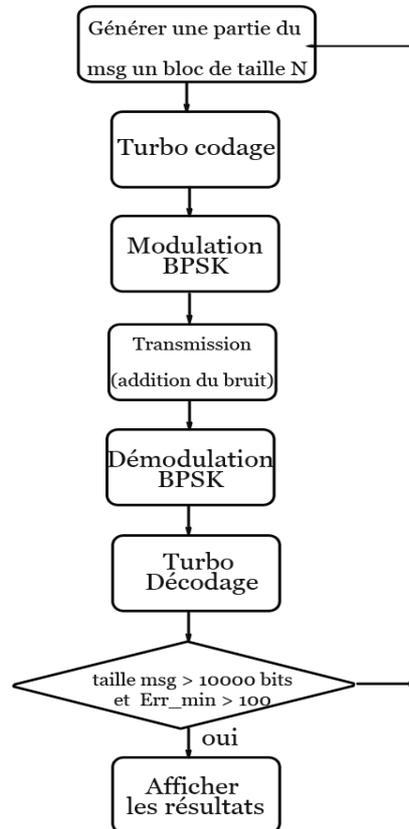


Figure 5.1 – Algorithme de notre système de communication simulé

$[4, (15, 13), 13]$ et d'autres polynômes pour la comparaison ; $(G(D) = [4, (15, 13), 13]$ la référence)

- L'entrelaceur : plusieurs méthodes sont présentes : linéaire, non linéaire, aléatoires, UMTS... (UMTS la référence) ;
- Nombre d'itérations : dans notre travail on utilisé que 6 itérations, qui est suffisante dans notre étude.

Nous présentons la probabilité d'erreur par bit BER (Bit Error Probability) en fonction du rapport E_b/N_0 . Des résultats des simulations, nous pouvons déduire les comportements suivants :

5.1.1 Effet de nombre d'itération sur la performance du Turbocode

Dans cette première étape nous allons évaluer les performances du processus itératif du décodage des Turbo-codes. Pour différent nombre d'itération (de 1 jusqu'au 6 itérations), le polynôme générateur des deux codeurs CRS du turbocodeur est $G(D) = (4, [15, 13], 13)$, le nombre min d'erreur 100 bits, la taille du bloc 512, le SNR varie de 0 à 1.5 dB.

Comme il est mentionné dans la section 4.6 le taux d'erreur binaire se diminue en

augmentant le nombre des itérations. Il permet d'améliorer l'information a priori $L_e(u_k)$ des bits à chaque itération. La figure(5.2). nous permet de voir bien l'effet de nombre d'itération sur la performance. Le gain entre la première itération et la deuxième itération est de 0.5 dB.

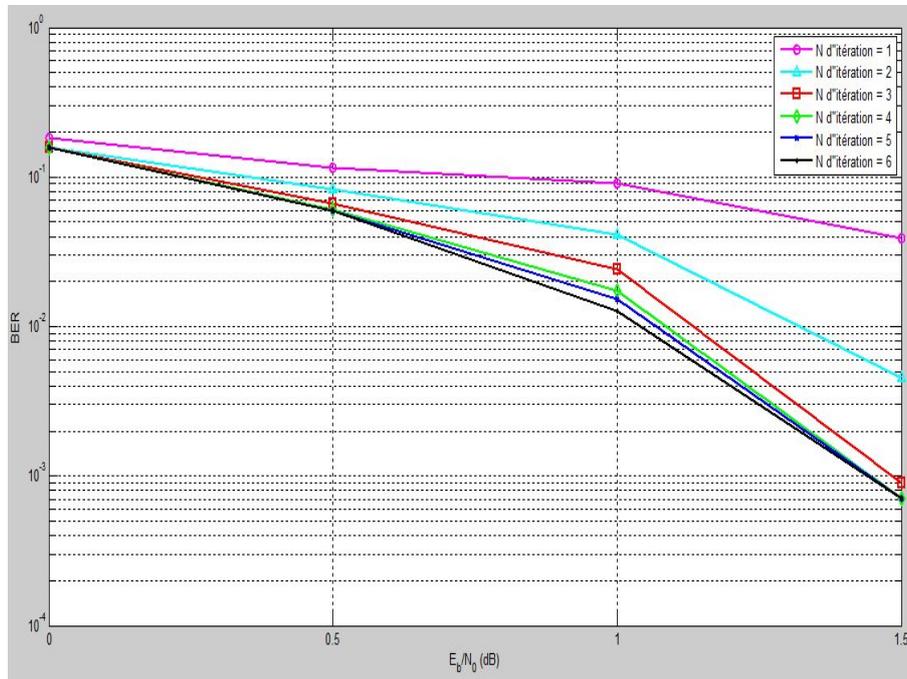


Figure 5.2 – Effet de nombre d'itération sur la performance du Turbo-code N=512 bits

Après un nombre défini d'itérations les courbes se convergent, et l'amélioration s'arrête. Après 4 itérations l'amélioration est petite, et l'arrêt de processus à cette étape est préféré pour gagner du côté temps d'exécution.

5.1.2 Effet de la taille de bloc

La figure(5.3) illustre bien le rôle qu'elle joue la taille du bloc sur le taux d'erreur binaire BER. nous constatons une amélioration du BER lorsque la taille du bloc croît (donc la taille de l'entrelaceur car nous avons pris la taille de l'entrelaceur identique à celle de la longueur du bloc d'information). Cette amélioration est de l'ordre de $1/N^1$ [24]. Par exemple en comparant les résultats de simulations pour $N = 512$ et $N = 1024$ (figure(5.3)) le BER s'améliore d'un facteur de 2, ce qui n'est que le rapport entre 1024 et 512 ($1024/512 = 2$) ce qui vau un gain en dB égale à 0.5 a $BER = 2.510^{-3}$.

le choix de taille de bloc est pris en tenant compte du BER voulu, de temps de calcul et la complexité de circuit.

1. ceci a été démontré théoriquement par Benedetto dans le cas de l'entrelacement bloc classique

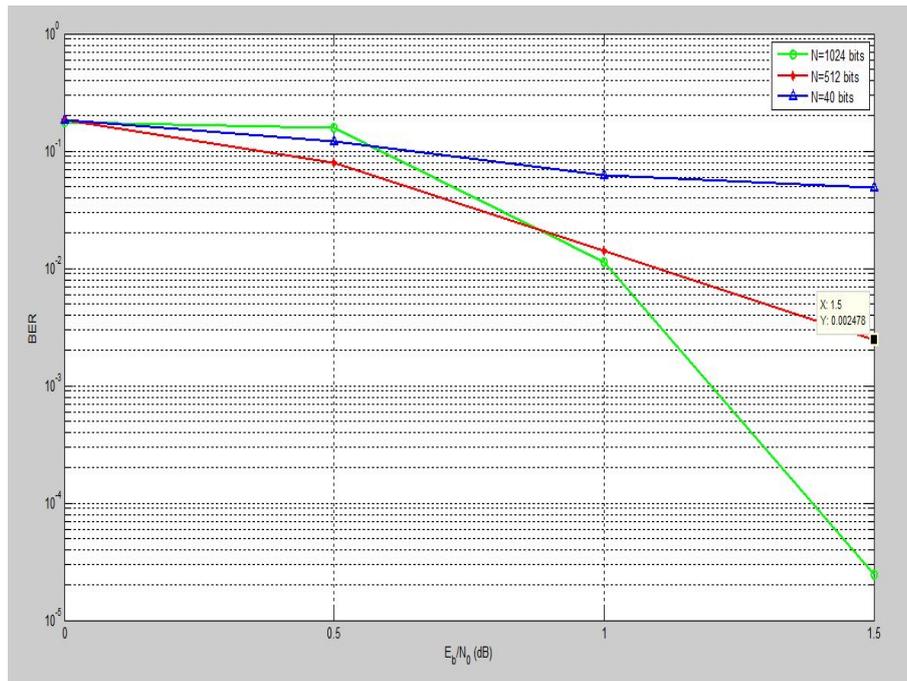


Figure 5.3 – Effet de la taille de bloc sur la performance du Turbo-code

5.1.3 Effet du polynôme générateur

Dans cette partie nous analysons l'influence du choix du polynôme générateur sur les performances du codeur faisant une comparaison entre deux codeurs ayant la même longueur de contrainte, un codeur muni d'un polynôme générateur quelconque et le second appartenant au corps de Galois, les résultats présentés dans la figure(5.4) montrent bien qu'avec un polynôme générateur quelconque les performances sont médiocres et le codeur ne joue presque aucun rôle.

Dans la seconde sous-partie nous analysons, aussi à travers des simulations, l'influence de la longueur de contrainte de code (taille de mémoire). La taille de l'entrelaceur est de 40. En pratique, l'écart des performances favorable aux codes ayant une distance libre plus élevée (taille de mémoire plus grande), n'est effectif que pour les fortes valeurs de E_b/N_0 . Ce fait est illustré sur la figure(5.5) et figure(5.6). Comme nous pouvons le voir dans l'exemple de la figure(5.5), le code du polynôme générateur $G(D) = (3, [7, 5], 5)$ ($K = 3$) offre les meilleures performances à faible rapport E_b/N_0 par rapport au deuxième code ($G(D) = (7, [171, 133], 133)$) ayant une plus grande longueur ($K = 7$) sachant que les deux polynômes générateurs des deux codeurs appartiennent tous les deux au corps de Galois.

Ce comportement est contradictoire avec les performances théoriques données par les distances libres des codes. Cet écart est dû à une vitesse de convergence plus élevée dans le cas des codes avec une faible longueur de contrainte [25].

On sait que la pente asymptotique de la courbe du TEB est fonction de la distance

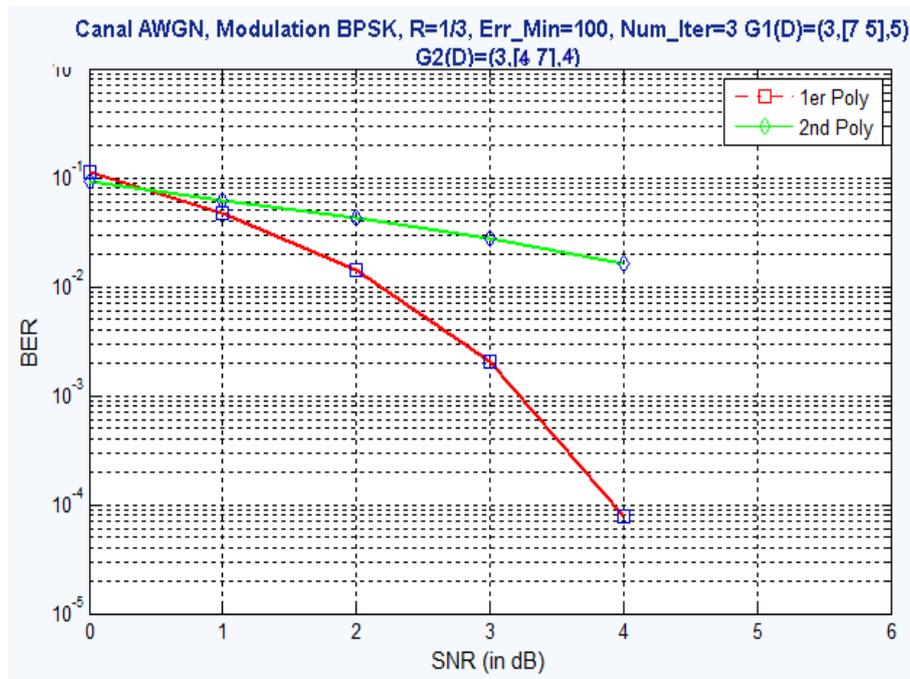


Figure 5.4 – Effet du type de polynôme générateur sur la performance du Turbo-code

libre du code convolutif considéré. Ce fait signifie qu'il existe une valeur seuil du SNR à partir de laquelle le codeur ($G(D) = (7, [171, 133], 133)$) aura de meilleures performances que le code $G(D) = (3, [7, 5], 5)$ et c'est ce que nous remarquons (figure(5.5)), au delà de $SNR = 3.4dB$ le codeur $G(D) = (3, [7, 5], 5)$ devient moins performant que le deuxième codeur.

Autres que ses performances, le code $G(D) = (3, [7, 5], 5)$ conduit à une très faible complexité quand on le compare avec les codes ayant une longueur de contrainte supérieure, donc si la BER désirée est faible, il n'y a aucun avantage à augmenter la longueur de contrainte, surtout pour des blocs de petite taille (inférieurs à 400).

5.1.4 Effet du type de permutation

Comme nous l'avons déjà vu la technique d'entrelacement augmente la distance minimale de Hamming, ce qui permet d'améliorer la capacité de correction du code et de lutter contre les erreurs en salves on les rendant indépendantes et surtout espacées dans le temps.

Les simulation montre que pour le même polynôme générateur, un rendement fixe de 1/3 et $E_b/N_0 \in [0 \ 1.5] dB$ le type de permutation n'a pas d'influence lorsque la taille de l'entrelaceur est petite $K = 40$ (figure(5.6)) mais ce n'est pas le cas pour les moyennes et grandes tailles $k = 512$ et $K = 1024$. La figure(5.7) et figure(5.8) montre que le type de permutation joue aussi un rôle sur les performances de turbo codeur,

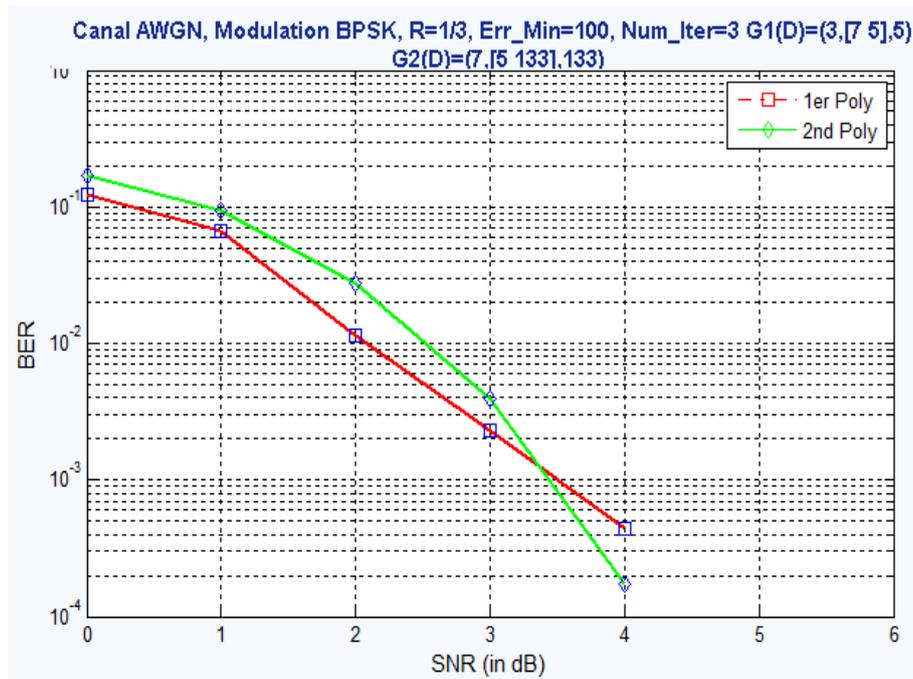


Figure 5.5 – Effet de la longueur de contrainte de code sur la performance du Turbo-code pour $G1=(3,[7,5],5)$ et $G2=(7,[171,133],133)$

et plus cette permutation est aléatoire (dispersion temporelle) le codeur devient plus performant. Si on prend l'exemple de la figure(5.8) pour une taille de bloc $K = 1024$ la permutation aléatoire et UMTS apporte un gain de $0.3dB$ à $BER = 10^{-2}$ par rapport à la permutation régulière.

En remarque aussi que l'entrelaceur bloc UMTS fournit des performances d'erreurs équivalentes à celui de l'entrelaceur aléatoire, ce dernier consomme plus de mémoire (en doit sauvegarder la nouvelle position de chaque bit) c'est la raison pour la quelle nous avons adopté l'entrelacement bloc de type UMTS dans la totalité de nos simulations.

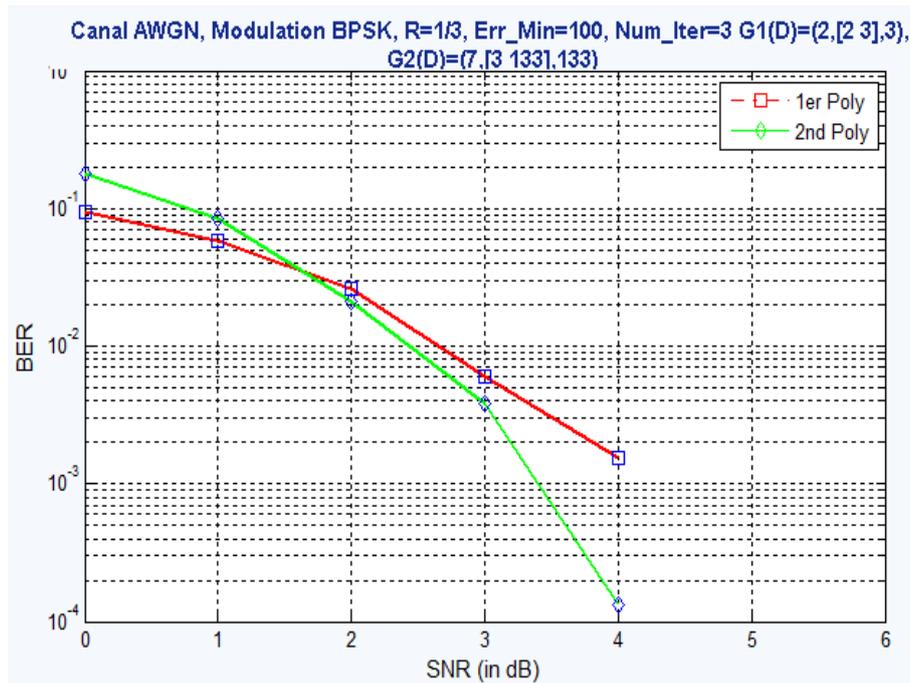


Figure 5.6 – Effet de la longueur de contrainte de code sur la performance du Turbo-code pour $G1=(2,[2,3],3)$ et $G2=(7,[171,133],133)$

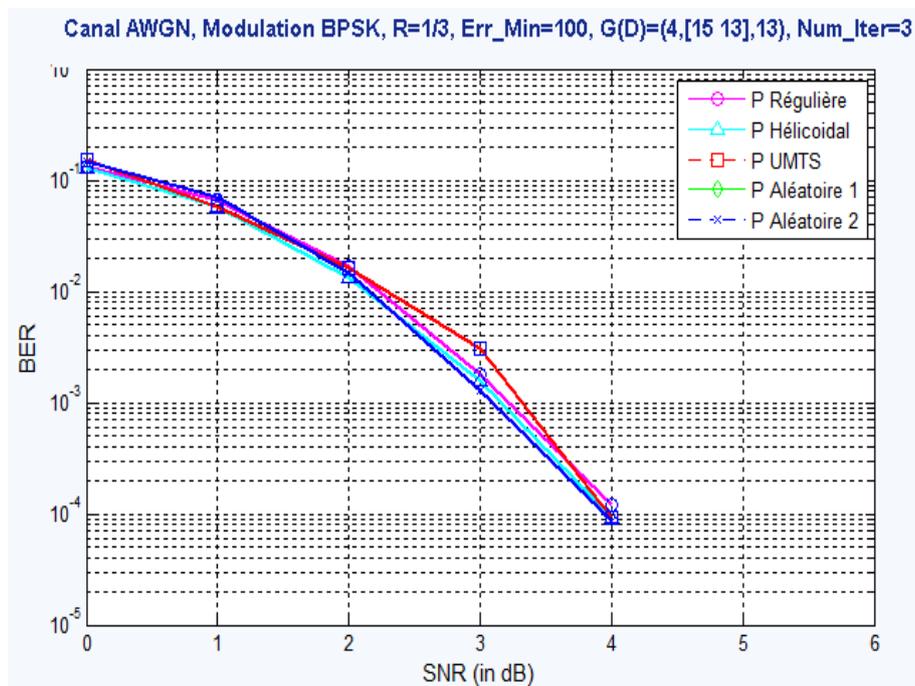


Figure 5.7 – Effet de la longueur de contrainte de code sur la performance du Turbo-code

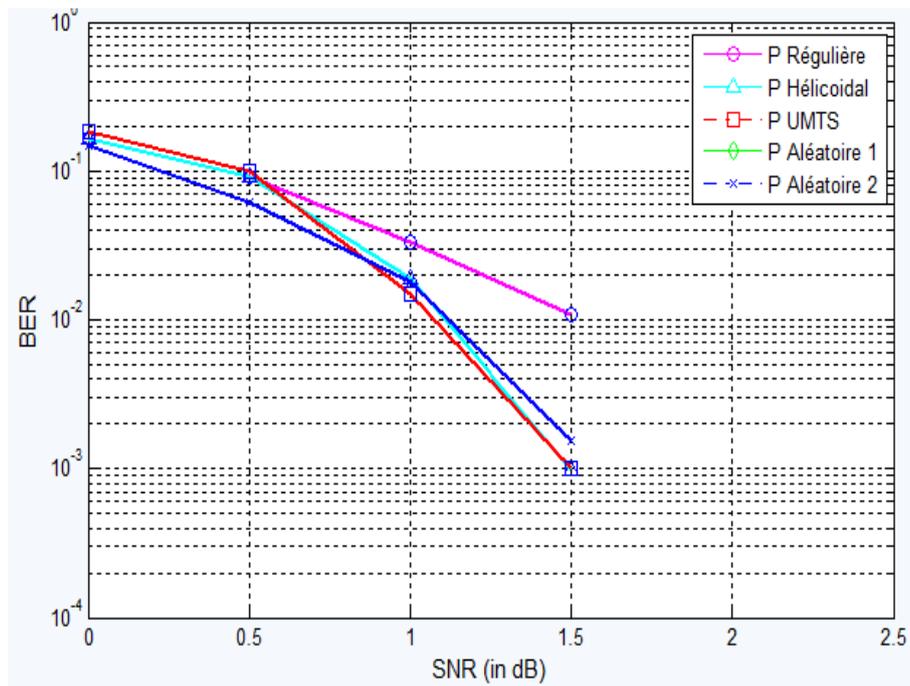


Figure 5.8 – Effet de la longueur de contrainte de code sur la performance du Turbo-code

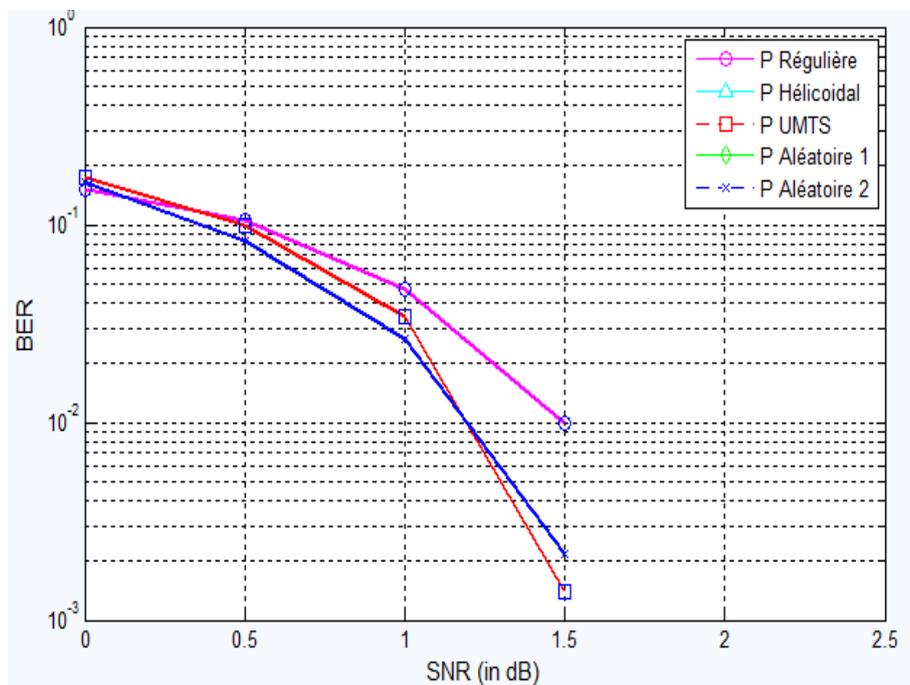


Figure 5.9 – Effet de la longueur de contrainte de code sur la performance du Turbo-code

Chapitre 6

Conclusions

Afin de proposer une qualité de service compétitive avec les systèmes terrestres, les futurs systèmes de communication par satellite en bande Ka doivent franchir certains handicaps. Les fortes atténuations du canal, causées principalement par les précipitations, constituent un défi qu'il va falloir relever. Même si plusieurs techniques de codage de canal et modulation ont été proposées afin de lutter contre les dégradations du canal, l'adoption d'un codeur de canal est nécessaire pour une exploitation optimale des ressources du canal. Dans ce mémoire, les turbocodes ont été adoptées pour la conception d'un codeur de canal, les principales motivations de ce choix peuvent se résumer par les points qui suivent :

- Le codeur utilise la fonction de permutation : cette propriété offre à cette classe de codeur de canal une parfaite immunité contre les erreurs qui arrivent en salves.
- La diversité des paramètres influencent sur le codeur : cette propriété permet une plus grande souplesse pour modifier les performances du codeur en l'adaptant aux conditions de propagation (SNR). L'adaptativité se fait par la modification d'un ou plusieurs paramètres de turbocodeur.
- L'application d'un processus de décodage itératif lors d'une concaténation parallèle avec un code convolutif : ce processus de décodage offre des bonnes performances. Il permet une communication fiable même à faible E_b/N_0 .

Après avoir introduit dans un premier temps, Les systèmes de communication numérique. Nous avons détaillé un peu dans le chapitre suivant les systèmes de communications satellitaire, présentant par la suite les différentes classes des codes correcteurs : bloc et convolutif ainsi que leurs présentations polynomiale et graphique. En comparant leurs performances dans un canal gaussien AWGN, nous avons constaté que les codes convolutifs sont les plus performants. A partir de ce résultat, nous avons considéré le cas des codes convolutifs pour le reste de notre étude. Nous avons, ensuite choisi le turbo codeur est abordé la partie décodage et encodage. Concernant la partie décodage, nous avons illustré l'algorithme Log-MAP qui est un décodeur SISO et

on a expliqué le processeur itératif. Et concernant la partie encodage, nous avons présenté l'algorithme d'un encodeur présenté par Claude Berrou. Quant à la partie expérimentale, nous avons tout d'abord commencé par introduire la chaîne de transmission, les conditions et hypothèses de simulations, ensuite, nous avons présenté les résultats de simulations, à partir desquelles nous avons tiré les conclusions suivantes :

- Les codes turbocodes donnent de meilleures performances par rapport aux codes convolutifs classiques.
- Augmentation de la taille du turbocode entraîne une amélioration des performances, cette amélioration est remarquable surtout pour $N = 1024$.
- L'augmentation du nombre d'itérations améliore les performances de décodage jusqu'à un nombre d'itérations maximum où l'algorithme de décodage converge.
- Plus la permutation est aléatoire plus la protection contre les erreurs est meilleure (une grande diversité temporelle).

Cette étude a permis de mettre en évidence l'importance du choix de certains paramètres sur les performances du Turbocode et en se basant sur ces résultats pour le choix de ces paramètres nous pouvons avoir un meilleur compromis entre la complexité et la performance.

Annexe A

Notions sur les corps de Galois

Un corps de Galois noté $GF(Q)$ (pour Galois Field) est un ensemble fini qui possède Q éléments. Un corps de Galois défini à partir d'un élément primitif est tel que le nombre d'éléments Q du corps de Galois s'écrit $Q = q^m$ où q est un nombre premier. La définition de $GF(q^m)$ repose sur un polynôme irréductible $p(x)$ de degré m , à coefficients dans $GF(q)$. Comme $p(x)$ est irréductible, il ne possède pas de racine dans $GF(q)$. Cependant, si on suppose que ce polynôme possède m racines dans un autre ensemble et qu'on note α l'une de ces racines, cette racine est alors définie comme l'élément primitif du corps de Galois $GF(q^m)$. Le polynôme $p(x)$ dont α est une racine est appelé le polynôme primitif de $GF(q^m)$ puisqu'il est essentiel à l'obtention des éléments du corps. En effet à partir de ce polynôme et de α , tous les éléments du corps de Galois sont obtenus :

$$GF(q^m) = \{0, \alpha^0, \alpha, \dots, \alpha^{q^m-2}\} \quad (\text{A.1})$$

Les opérations de multiplication et d'addition dans $GF(q^m)$, sont réalisées comme suit :

- pour l'addition, on réalise la somme bit par bit modulo q ;
- pour la multiplication, on utilise la notation en puissances de α c'est-à-dire pour tout $i; j$ entiers :

$$\alpha^i \cdot \alpha^j = \alpha^{(i+j)(q^m-1)}.$$

La définition des codes correcteurs d'erreurs plus précisément BCH ou RS repose sur des corps de Galois définis à partir d'un élément primitif avec $q = 2$ et tel que $m \leq 3$. Le corps de Galois s'écrit $GF(2^m)$ et contient 2^m éléments. Les additions dans le corps de Galois ont donc lieu bit à bit *modulo*(2), et ainsi l'opérateur $+$ est égal à l'opérateur $-$. De plus, les 2^m éléments du corps, appelés *symboles* sont composés des éléments de base '0' et '1'. Pour en avoir 2^m il suffit de construire des symboles de *m* bits. Ces symboles *m*'aires sont utiles pour la définition des polynômes générateurs des codes BCH ou RS. En effet, ils sont construits comme étant des polynômes de degré $N - K$ et possédant comme racines les $2t$ puissances de l'élément primitif $\alpha : \{\alpha^i\}, i \in [1, 2t]$ où t correspond

au nombre maximal d'erreurs corrigibles par le code considéré. De part cette propriété, ces symboles *m'aires* sont manipulés lors du décodage des données pour localiser et évaluer les erreurs de transmission tant pour les codes BCH que pour les codes RS

Bibliographie

- [1] O. A. Hani, “Le dÉcodage des codes turbo,” Master’s thesis, ÉCOLE POLYTECHNIQUE DE MONTRÉAL, 1999.
- [2]
- [3] Y. Mori, *Electronique pour le traitement du signal théorie de l’information et du codage*. Lavoisier, 2006.
- [4] C. Berrou, *Codes et turbo codes*. Springer collection IRIS, 2007.
- [5] R. P. Michel Jézéquel, *Turbo codes, error-correcting codes of widening application*. HPS, 2001.
- [6] L. Li, *Analysis of Low Power Implementational Issues of Turbo-like Codes in Body Area Networks*. PhD thesis, UNIVERSITY OF SOUTHAMPTON, 2009.
- [7] *Digital modulation in communications systems An introduction Application Note 1298*.
- [8] O. Muller, “Architectures multiprocesseurs monopuces génériques pour turbo-communications haut-débit,” Master’s thesis, Université de Bretagne-Sud, 2007.
- [9] M. T. Elias, “Etude des codes ldpc et application dans un système mimo,” Master’s thesis, ECOLE NATIONALE POLYTECHNIQUE, 2009.
- [10] S. B. Wicker, *Error Control Systems for Digital Communications and Storage*. Englewood Cliffs : Prentice Hall, 1995.
- [11] P. G. G. COHEN, J.-L. DORNSTETTER, *Codes correcteurs d’erreurs : une introduction au codage algébrique*. MASSON, 1992.
- [12] A. Boyer, “Canaux de transmissions bruités,” Master’s thesis, INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE TOULOUSE, 2010.
- [13] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical*, 1948.
- [14] W. S. B. Heegard, C., *Turbo Coding*. Kluwer Academic Publishers, 1999.
- [15] A. G. C. Berrou and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding : Turbo-codes,” *ICC’93, Geneva, Switzerland*, 1993.
- [16] C. . b 6, *telemetry channel coding*. blue book. issue 6, 2002.

- [17] H. G. Claude Berrou and S. Kerouédan, "Serial/parallel (s/p) turbo codes for low error rates," *IEEE Communications Society*, 2004.
- [18] G. D. Forney, "Performance of concatenated codes," *IEEE Press*, pp. 90–94, 1974.
- [19] G. M. S. Benedetto, D. Divsalar and F. Pollara, "Serial concatenation of interleaved codes : performance analysis, design, and iterative decoding," *IEEE Trans. Info. Theory*, vol. 44, no. 3, 1998.
- [20] A. G. e. C. B. Stkphane Le Goff, "Turbo-codes and high spectral efficiency modulation," *IEEE*, 1994.
- [21] M. Mansour, "Application du turbo-code au codage de canal," Master's thesis, Ecole Nationale Polytechnique, 2008.
- [22] F. J. L. R. Bahl, J.Cocke and J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions On Information Theory*, 1974.
- [23] J. P.Hoehner, "A viterbi algorithm with soft-decision outputs and its applications," *IEEE Globecom*, 1989.
- [24] S. G.Montorsi, "Unveiling turb codes : Some results on parraillel concatenated coding schemes," *IEEE Trans Inform Theory*, 1996.
- [25] P. Moqvist., *Multiuser Serially Concatenated Continuous Phase Modulation*. PhD thesis, Chalmers University of Technology Goteborg Sweden, 2002.
- [30] G. Maral, M. Bousquet, *Satellite Communications Systems, third ed, Wiley*
- [31] N. Sennequier, *Les satellites de télécommunications, Collection Que sais-je, PUF* 2000.
- [32] J. G. Proakis, *Digital Communications*, third ed. Mac Graw Hill
- [33] Shannon C.E., *A Mathematical Theory of Communications, Bell Syst Tech J*, vol. 27, pp 623-656, October 1949
- [34] M. Oerder, H. Meyr. *Digital Filter and Square Timing Recovery. IEEE Trans. Comm.* vol 36 n°5, May 1988.
- [35] F. Gardner. *Interpolation in Digital Modems-Part I : Fundamentals. IEEE Trans. Comm.* Vol 41, n°3, March 1993.
- [36] F. Gardner. *Interpolation in Digital Modems-Part II Implementation and Performance. IEEE Trans. Comm.* Vol 41, n°6, June 1993.
- [37] C. Bergogne, *Synchronisation pour transmission en mode paquet à faible rapport signal sur bruit, Thèse de Doctorat ENST*, 1996