UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
MOHAMED ZEBDI

ÉTUDE COMPARATIVE DES ALGORITHMES DE CORRECTION D'ERREURS SANS VOIE DE RETOUR POUR LA COMMUNICATION SANS FIL

AVRIL 2004

À ma mère, la plus

formidable des mamans

RÉSUMÉ

Le terme « Codage » est généralement associé à la modélisation de l'information à transmettre en une série de symboles ou nombres pour la réalisation de plusieurs objectifs. Le cryptage ou le codage d'identification est courament utilisé dans la transmission des données en vue d'en assurer une protection contre une mauvaise utilisation de l'information utile. D'autre part, la technique de codage source vise à compresser les informations et le codage canal cherche à immuniser ces mêmes informations contre les distorsions causées par le canal de communication.

La théorie de l'information et le codage canal ont été introduits par C. E. Shannon. Dans son article publié en 1948, Shannon a défini les limites théoriques du codage canal, un domaine encore méconnu durant cette époque. Les premières propositions qui ont suivi la publication de cet article sont basées sur le principe de l'ajout de l'information redondante à l'information utile, permettant plus tard de détecter ou de corriger, les erreurs dues à la distorsion causée par le canal de communication. L'information redondante peut être déterminée par l'utilisation de deux classes différentes de codage. La première classe comporte le codage par bloc, où des blocs indépendants de symboles « mots codes » seront construits par un ajout constant de cette information redondante à l'information utile. La seconde méthode est le codage convolutionnel, où l'information redondante est continuellement

calculée pour la séquence de codes, en utilisant la combinaison (convolution) d'une succession de symboles d'information. Une étude détaillée des différentes méthodes de codage est ainsi présentée dans cette étude comparative.

D'autre part, les algorithmes prévus dans les récepteurs de la troisième génération, permettant d'accroître la capacité d'un système de communication DS-CDMA (*Direct Sequence Code Division Multiple Access*), seront traités dans ce travail. Cette étude permettra de bien introduire les différentes techniques et algorithmes de décodage. Les techniques de décodage par méthodes dures (*Hard Decision Decoding*) et méthodes douces (*Soft DecisionDecoding*) représentent les deux principales classes d'algorithmes de décodage convolutionnel et le turbo décodage respectivement.

La sortie douce (*Soft Output*) d'un algorithme de décodage permet la réalisation d'un processus de décodage itérative appelé turbo décodage. Ainsi, après chaque itération, une amélioration des performances est constatée par rapport à la précédente, grâce à l'introduction du principe de la « conduite d'information ». Cette derniére permet d'assister le processus de décodage durant les prochaines itérations. La structure de turbo code est ensuite présentée afin de déterminer l'influence de certains paramètres, tels que l'encodeur convolutionnel et l'interfoliage (*Interleaving*) sur les performances d'un processus de turbo décodage.

Les résultats de simulation des différents algorithmes de décodage, sont ainsi présentés; les deux variétés de l'algorithme de Viterbi, en l'occurrence (*Viterbi Algorithm*) et l'algorithme SOVA (*Soft Output Viterbi Algorithm*) pour le codage convolutionnel, et les algorithmes MAP (*Maximum Aposteriori Probabilite*), avec ces deux approximations dans le domaine logarithmique (*Log-MAP*) et (*Max-Log-MAP*) pour le turbo code.

Une variation systématique des paramètres d'influences est introduite, permettant à la fois de mettre en évidence les différents choix apportés par le standard et les limites de performances de chaque algorithme de décodage. Tandis que l'algorithme SOVA apporte une légère amélioration à l'algorithme de Viterbi pour le codage convolutionnel, le Max-Log-MAP représente le meilleur compromis entre performance et complexité de calcul dans un processus de turbo décodage.

L'étude comparative est complétée par l'exploration de l'espace d'implémentation, avec une étude comparative entre une approche logicielle et matérielle d'un turbo code, pour conclure enfin sur le codage canal selon le standard 3GPP.

REMERCIEMENTS

Je tiens tout d'abord à remercier le professeur Daniel MASSICOTTE, le directeur de ce travail, pour sa confiance et ses jugements de valeur, qui m'ont permis de m'introduire et d'acquérir des connaissances précieuses dans ce domaine recherche, ainsi que de faire partie de son équipe au Laboratoire des signaux et systèmes intégrés.

Je tiens aussi à remercier mes parents qui m'ont aidé et soutenu durant mes études, ainsi que Messaoud Ahmed Ouameur et Adel-Omar Dahmane, pour leur patience, leur soutien et leurs précieux conseils, afin que je puisse réussir la réalisation de ce travail.

Je dédie ce travail à mes frères (Tarik, Abdelhamid et Adel), mes sœurs (Radia et Samia), mes amis d'Algérie : Slimane, Abdelkader et Kamel.

Table des Matières

Table des illustrations

Liste des tableaux

Liste des symboles

a	Facteur d'évanouissement
a_k	Bit de parité dans l'encodeur
$A_{k-1}(s')$	Logarithme de $\alpha_{k-1}(s')$
$\widetilde{A}_k(s)$	Valeur estimée de $A_k(s)$
b_l	Un sous canal
$B_k(s)$	Logarithme de $\beta_k(s)$
c	Élément du mot code
C	Mot code
Cte	Constante
C_c	Capacité du canal
$d_{\min}$	Distance minimale
d_{free}	Distance libre générée par un encodeur convolutionnel
E_b	Énergie par bit transmis
E_s	Énergie par symbole
$f_c(\delta)$	Fonction d'approximation dans l'algorithme Log-MAP
G	Polynôme générateur
g_i	Élément du polynôme générateur G
$GF(q)$	Calcul dans le domaine de Gaulois de la variable q

H_l	Réponse impulsionnelle d'un sous canal
k	Ordre d'un bit en entrée (Temps discret)
k_0	Bit d'entrée d'ordre k_0
K	Contrainte de Longueur d'un encodeur convolutionnel
L	Longueur de séquence k bits d'entrée
$L(u_k / \mathbf{y})$	LLR sur la probabilité de u_k étant donné la séquence $\mathbf{y}$
$L(u_k)$	Information apriorique sur u_k
L_c	Indice de fiabilité du canal de transmission
l	Profondeur de treillis d'un encodeur convolutionnel
$L_e(u_k)$	Information extrinsèque sur u_k
m	Fenêtre d'encodage
M	Mémoire d'un encodeur convolutionnel
$M(\underline{s}_k^s)$	Valeur métrique à l'état s à l'instant k
N_0	Énergie du bruit
n_0	Sortie du codage d'ordre n_0
n	Nombre d'éléments par mot code
p	Élément du polynôme générateur du mot code $c(p)$
P_i	Un vecteur d'interférence
$\mathbf{r}$	Vecteur du signal reçu
R_c	Rapport de Codage
$\widetilde{r}$	Sortie bruitée

r	Signal reçu
R	Rapport de transmission
s	Valeur de l'état S à l'instant k
S_k	État de transition à l'instant k
$\underline{s}$	Vecteur des états de transitions
T	Période de signalisation
t	Temps continue
u	Séquence de bits d'entrée
u_k	Bit d'entrée
u_k	Bit d'entrée
W	Bande passante
$\mathbf{x}_k$	Vecteur des bits transmis x_k
(X_k, Y_k)	Composantes d'un mot code $C_k (R = 1/2)$.
x_{kl}	Version reçue du bit systématique d'ordre l à l'instant k
$\mathbf{y}$	Séquence binaire reçue
y_{kl}	Version reçue du bit de parité d'ordre l à l'instant k
y_{ks}	Valeur reçue du bit systématique x_k
η	Bruit additionnel
τ	Retard temporel
$\psi(t)$	Signal de durée T
$P(u_k)$	Probabilité la valeur du bit décodé u_k

$\alpha_{k-1}(s')$	Probabilité de transition par l'état s' à l'instant $k-1$
$\gamma_k(s',s)$	Probabilité de transition entre l'état s' et l'état s à l'instant k
$\beta_k(s)$	Probabilité de transition par l'état s à l'instant k
σ^2	Variance du bruit
$\Gamma_k(s',s)$	Logarithme de $\gamma_k(s',s)$
p	Densité de probabilité
Δ_k^s	Différence métrique à l'état s à l'instant k

Liste des abréviations

3G	3rd Generation Partnership Project
3GPP	3rd Generation Partnership Project
3GPP	3rd Generation Partnership Project
AWGN	Aditive White Gaussian Noise
BER	Bit Error Rate
DS-CDMA	Direct Sequence Code Division Multiple Access
FEC	Forward Error Correction
ISI	Inter Symbol Interference
LLR	Log Likelihood Ratio
MAI	Multiple Access Interference
MAP	Maximum Aposteriori Probabilite
MLSE	Maximum Least Square Estimation
RSC	Recursive Systematic Convolutionnel Code
SNR	Signal Noise Ratio
SOVA	Soft Output Viterbi Algorithm
VA	Viterbi Algorithm
ZF	Zero Forcing

1.INTRODUCTION

Dans ce chapitre, les fondements de la théorie de l'information seront introduits pour une meilleure compréhension de la problématique relative à la naissance de la théorie du codage. La présentation des éléments de cette théorie permet ainsi de définir les objectifs, la méthodologie à adopter en vue de réaliser une étude comparative des techniques de décodage d'un système de communication sans fil à accès multiple par code avec une largeur de bande étalée (WCDMA) [HAL01].

1.1 Problématique

Dans son article « *A Mathematical Theory of Communication* » 1948, Shannon avait stimulé un nouvel espace de recherche axé sur les domaines de la théorie de l'information et la théorie du codage. La contribution philosophique fondamentale de était l'application formelle de la théorie de la probabilité dans les études et l'analyse des systèmes de communication. La contribution théorique du travail de Shannon dans le domaine du codage était la mise en œuvre d'une nouvelle définition pratique du terme 'Information', avec l'élaboration de plusieurs 'Théorèmes sur le Codage', donnant naissance à une limite supérieure appelée 'Capacité du Canal' C_c du rapport de transsmission R (*Rate*), définissant le nombre

de bits par symbole transmis pouvant garantir une transmission fiable de 'l'information', par un canal de communication.

Ainsi 'La théorie du codage pour un canal continu, avec une limitation moyenne en puissance', établit que la capacité C_c d'un canal de largeur de bande W munie d'un bruit additionnel Gaussien à bande limitée (*Aditive White Gaussian Noise*), (ce qui caractérise approximativement plusieurs modèles pratiques de systèmes de stockage d'informations ou de communications digitales), peut être exprimé en bits par second (bps) par :

$$C_c = W \log_2(1 + E_s / N_0) \tag{1}$$

En assumant une parfaite signalisation au sens de Nyquist, ou E_s, représente la moyenne de l'énergie du signal pour chaque intervalle de signalisation de durée T, et $N_0/2$ représente les deux parties de la densité spectrale de puissance du bruit, en supposant qu'une quadrature du signal (deux dimensions) est envoyée pour chaque T second d'intervalle de signalisation.

La démonstration de ce théorème [COV91] a permis de mettre en évidence la possibilité d'atteindre une probabilité d'erreur arbitrairement choisie, selon un schéma de codage approprié, pour tout rapport de transmission R inférieur ou égal à la capacité C_c du canal. D'autre part, aucun schéma de codage ne pourrait permettre une performance fiable pour $R > C_c$.

Ce théorème appelé 'Théorème du codage' ne donne aucune orientation sur les méthodes pour trouver un schéma du codage approprié, l'ordre de complexité pour une implémentation, selon une probabilité d'erreur précise, du fait de la nature existentielle de ce théorème. Généralement, le terme 'Codage' est associé à toute modélisation de l'information en une série de symboles ou nombres pour différentes réalisations.

Le cryptage ou codage d'authentification appelé 'Cryptographie' (*Source Coding*) Fig. 1.1, est souvent utilisé dans la transmission de l'information, afin de la protéger d'une éventuelle mauvaise utilisation. Ainsi, le codage de source est une technique de codage permettant de compresser ces mêmes données, en vue d'une optimisation du procédé de la transmission, tandis que le codage canal, communément appelé FEC (*Forward Error Correction*), cherche plutôt à immuniser les informations contre toute distorsion aléatoire.

Un schéma de codage canal consiste à modéliser l'information utile à transmettre par une information redondante, ce qui permet à la reception de corriger l'information endommagée durant sa transmission. À la réception, l'information reçue est ainsi composée d'une série de 'Code Word'. Un 'Code Word' valide est un 'Code Word' dont le décodage se traduit par la recherche d'un autre 'Code Word' qui s'identifie le plus à la série de symboles reçus.

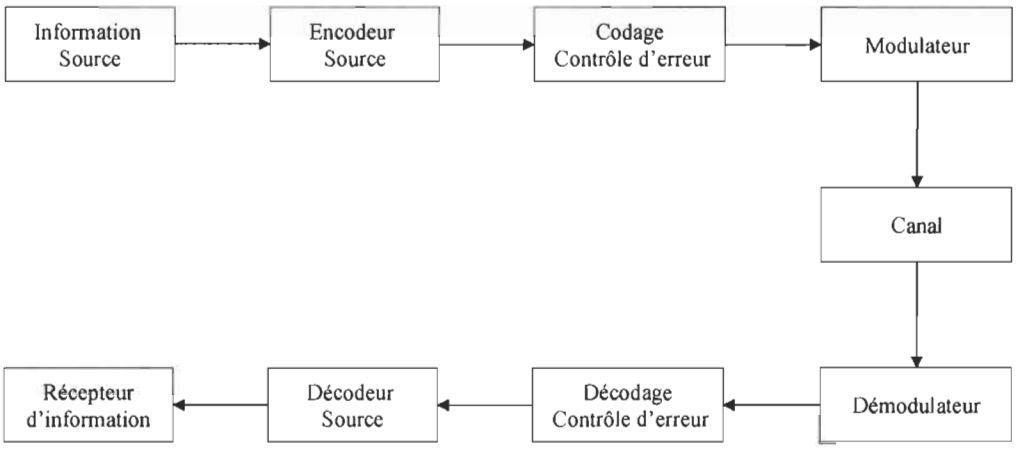


Figure 1.1 Structure d'un Systéme de Communication Numérique

Cette information redondante peut être déterminée par l'utilisation de deux classes de codages possibles. La première classe comprend le codage par Bloc, où des Blocs indépendants de 'Code Word' de symboles sont construits en associant de manière constante l'information utile avec l'information redondante. La seconde méthode est le codage convolutionnel où l'information redondante est continuellement calculée pour la séquence de codage, en utilisant une combinaison (*Convolution*) des symboles successifs. Malgré son jeune âge, le codage canal a acquis une notoriété dans les systèmes de communications, devenu ainsi indispensable pour atteindre des degrés de performance impossible sans son application. Le choix du type de codage à utiliser dépend grandement du canal lui-même.

L'une des premières techniques de codage FEC était un algorithme permettant la correction d'une seule erreur pour chacun des 'Code Word' transmis, connu sous le nom de 'Hamming Code'[HAM50]; une méthode de codage par Bloc proposée en 1950, suivie cinq ans après par la mise en oeuvre du concept du codage

convolutionnel proposé par Elias, avec une méthode de décodage subséquente élaborée par Wozencraft, Reiffen [WOZ61]. Mais l'évènement le plus important dans l'histoire de la correction d'erreur pour un codage convolutionnel reste l'invention par Viterbi en 1967, de l'algorithme MLSE (*Maximum Likelihood Sequence Estimation*) [VIT67].

L'algorithme de Viterbi (*Viterbi Algorithm*) [FOR73] produit une estimation de la séquence la plus probable de bits transmis, plutôt que de minimiser le rapport d'erreurs sur la séquence de bits transmis (*Bit Error Rate*) après décodage. Cependant, cette méthode de décodage a permis d'atteindre un BER minimal, en évaluant la probabilité des 2^k transitions binaires possibles, pour un message de k bits transmis. L'algorithme de recherche du minimum BER proposé par BAHL et al. en 1974, appelé MAP (*Maximum Aposteriori Algorithm*) [BAH74], bien qu'il améliore légèrement la performance réalisée par l'algorithme de Viterbi, cet algorithme était rarement mis en pratique étant donné sa complexité de calcul, jusqu'à l'invention du (*Turbo Code*) par Berrou et al. en 1993 [BERa93][BER96].

Le turbo code permet à un système de communication d'opérer beaucoup plus près de la limite théorique définie par Shannon et de réaliser ainsi un 'Gain en Codage' supérieur à celui déjà réalisé par l'algorithme de Viterbi. Dans son article « Near Shannon Limit Error-Correcting Coding and Decoding »[BERa93], Berrou, Glavieux et Thitimajshima ont utilisé deux encodeurs convolutionnels récursifs, avec sortie systématique RSC (*Recursive Systematic Convolution Code*) en parallèle, avec

l'insertion d'un Interfoliage entre les deux, en utilisant une version modifiée de l'algorithme de décodage MAP dans le processus de décodage.

1.2 Objectifs

Les principaux objectifs du sujet de recherche consistent à l'évaluation des performances des différentes techniques de codage et décodage utilisées selon le standard 3GPP [3GPP01] des communications sans fil, en vue de compléter une étude comparative des performances versus la complexité de calcul.

1.3 Organisation du mémoire

Une recherche bibliographique focalisée sur le codage canal permet en premier lieu de définir le fondement théorique et le développement de la théorie du codage avec le développement des systèmes de communication numérique depuis la publication du premier article de Shannon [SHA48]. Par la suite, nous devons définir les éléments de littérature permettant la réalisation de cette étude comparative, en conformité avec le standard 3GPP tels que les structures du codage, les techniques et algorithmes de décodage et la possibilité d'améliorer les performances du processus de décodage.

Une série de simulations seront introduites par une variation systématique des paramètres d'influence. Cette variation de paramétre sera introduite systématiquement pour le codage convolutionnel et le turbo codage, relativement au

modèle de l'encodeur, algorithmes de décodage, l'interfoliage, longueur de trame ainsi que le modèle de canal utilisé. En s'appliquant sur les résultats de recherches réalisées sur chacun de ces éléments, permettront de présenter en termes de performances l'influence de chaque paramètre sur un processus de décodage et de mettre ainsi en évidence, les différents choix apportés par le standard 3GPP.

D'autre part, l'exploration de l'espace de l'implémentation du turbo code comme étant la structure de décodage qui permet d'atteindre des performances proches de la limite de Shannon, sera réalisée de manière à compléter cette étude comparative sur le codage canal par une étude comparative entre une implémentation logicielle (*Software Implementation*) et matérielle (*Hardware Implementation*) du turbo code. L'étude comparative entre les différentes méthodes codage décodage utilisées dans le standard 3GPP sera effectuée selon la méthodologie suivante :

Dans le chapitre 2, l'étude portera sur les différentes techniques de codage présentées depuis l'apparition de la théorie du codage, ainsi que les récepteurs d'un système DS-CDMA (*Direct Sequence-Code Division Multiple Access*) et leur influence sur le choix de la technique de décodage, dépendamment de la qualité du signal reçu avant décodage. Le chapitre 3 comporte une étude détaillée sur les différentes techniques du codage selon le standard 3GPP, en l'occurrence le décodeur Viterbi comme algorithme de décodage par décision dure (*Hard Decision Decoding*) pour le codage convolutionnel, et SOVA (*Soft Output Viterbi Algorithm*), MAP, Max-Log-MAP et le Log-MAP, pour la version itérative du processus de décodage (Turbo Code).

Le chapitre 4 traite des éléments de structure du turbo code, tels que l'encodeur convolutionnel récursive, l'interfoliage et le turbo décodeur, pour ensuite présenter le résultats des différentes simulations effectuées, avec une variation systématique des paramètres d'influence dans le chapitre 5.

L'étude comparative sera complétée, dans le chapitre 6, par l'exploration de l'espace d'implémentation du turbo code ainsi que les perspectives de recherches dans ce domaine, suivi d'une conclusion sur tous les aspects de cette étude comparative.

2. TECHNIQUES DE CODAGE

Le codage est une modélisation (*Mapping*) d'une série semi-finie de séquences d'éléments $GF(q)$, (Gaulois Field)[BOS99], pour une valeur quelconque du nombre de séquences élémentaires k_0, dans une fenêtre du codage m (*Encoding Window*). Parmi les principales propriétés d'un encodeur (Voir Fig. 2.1), nomons la contrainte de longueur (*Constraint Length*) $K = M + 1$, ainsi que le rapport du codage de l'information R . Ces paramètres réunis définissent la longueur de la séquence reçue après codage (*Word Length*) ou le (*Block Length*) $n = (M + 1)n_0$.

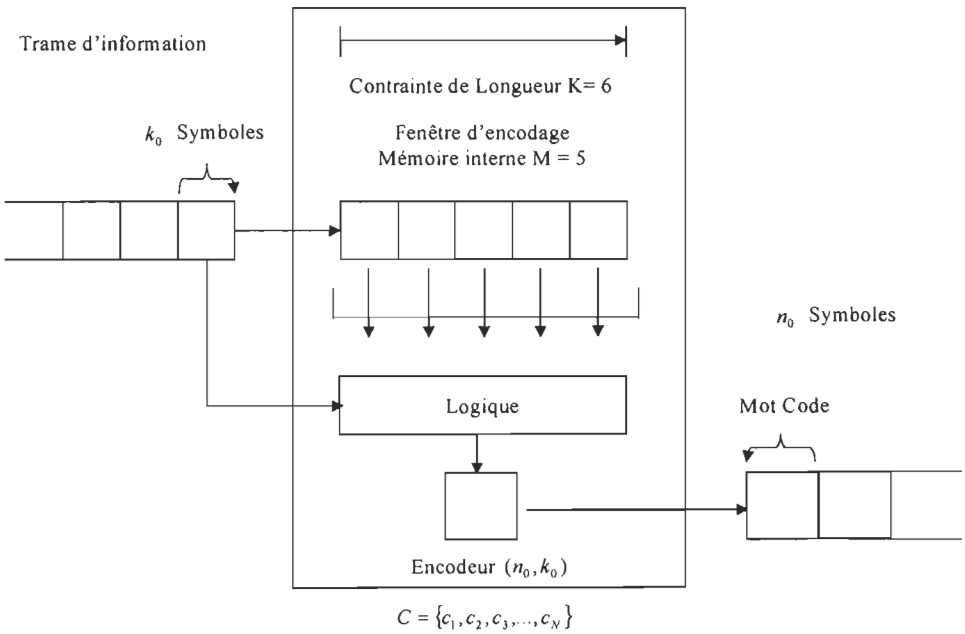


Figure 2.1 Structure d'un Encodeur

Où n_0 définie le nombre d'élements contenus dans un mot code, après codage de k_0 symboles repésentant un seul bit de donnée. La valeur de k_0 déprend du type de modulation choisi pour assurer la transmission des données, cette valeur est réduite à 1 dans le cas d'une modulation (*Binatry Phase Shift Key*).

2.1 Codage Linéaire

2.1.1 Codage Linéaire par Bloc

Un codage linéaire par Bloc(n,k) est complètement défini par $N = 2^k$ séquences binaires, de longueur n nommées mots codes [PRO95]. Un mot code C est constitué de N élements c_i, pour $1 \leq i \leq 2^k$.

$$C = \{c_1, c_2, c_3, ..., c_N\} \tag{2}$$

Chaque séquence de longueur n de c_i est composée de valeurs binaires 0,1. On désigne par la distance de Hamming entre deux codes C_i et C_j, le nombre des composantes sur lesquels, les deux mots codes diffèrent ($d(C_i, C_j)$). Par conséquent, la distance minimale pour un code, au sens de Hamming, est par définition la distance entre n'importe quel deux mots codes (*Code Word*) :

$$d_m = \min_{\substack{i \neq j \\ c_i \neq c_j}} d(C_i, C_j) \tag{3}$$

Le principal objectif, quant à l'utilisation d'un processus de codage pour un système de communication, est de faire augmenter le plus possible la distance Euclédienne entre les signaux transmis, et par conséquent, réduire la probabilité d'erreur relative à niveau de puissance donné. Ceci revient à dire que le meilleur instrument de mesure, permettant une comparaison des performances entre deux codes, revient tout simplement à une comparaison de la distance de Hamming entre les mots codes (*Code Words*) de ces codes.

Toutefois, la comparaison de toutes les distances, entre n'importe quel deux mots codes (*Code Words*), reste difficile et parfois impossible. C'est pourquoi la comparaison entre les différents codes est généralement basée sur le calcul de la distance minimale $(d_{\min})$. Pour une valeur donnée de (n,k), le code ayant une $(d_{\min})$ élevée est plus performant que celui avec une valeur inférieure de distance minimale $(d_{\min})$. Parmi les différentes classes du codage linéaire par Bloc on peut distinguer le codage de Hamming, le codes de correction des erreurs éclats, et le codage cyclique.

2.1.2 Code de Hamming

Un codage linéaire par Bloc défini avec $n = 2^m - 1$, $k = 2^m - m - 1$ et $d_m = 3$, pour un entier $m \geq 2$. Ces codes sont capables de produire une capacité de correction pour des erreurs singulières (sur un seul bit), avant décodage avec un rapport de codage

$$R_c = \frac{2^m - m - 1}{2^m - 1} \tag{4}$$

La valeur du rapport de codage tend vers 1 pour des valeurs élevées de m. Les codes de Hamming sont des codes ayant un rapport de transmission R élevée, avec une distance minimale d_m relativement basse ($d_m = 3$), c'est pourquoi les codes de Hamming ont une capacité de correction d'erreur limitée.

2.1.3 Codes de Correction des Erreurs Éclat

La plupart des codes linéaires par Bloc ont été conçus afin de corriger les erreurs aléatoires (*Random Errors*) totalement indépendantes des autres erreurs survenues sur le canal de transmission. Certains modéles de canal, y compris un canal (*Aditve White Gaussian Noise*), peuvent être catégorisés comme étant un canal à erreurs aléatoires. Cependant, pour d'autres models de canaux physiques, cette hypothèse d'une génération d'erreurs indépendamment du canal n'est pas toujours valide, le canal d'évanouissement (*Fading Channel*) en est le parfait exemple [PRO95]. Si le canal est en grande atténuation (*Deep Fade*), un grand nombre d'erreurs éclat (*Burst Error*) peuvent avoir lieu. On peut utiliser n'importe quel code pour correction d'erreur aléatoire, en vue de corriger les erreurs en éclats , pourvu que le nombre d'erreurs reste inférieur à la moitié de la distance minimale (*Minimum Distance*) du code lui-même.

La méthode effective pour la correction des erreurs en éclats est l'interfoliage des données codées, de manière à ce que les erreurs soient aléatoirement distribuées

sur plusieurs mots codes [BAR94] [DOL95], plutôt que quelques mots codes. Ainsi, les erreurs pouvant apparaître dans un Bloc resteront relativement faibles, ce qui permet l'utilisation d'un code de correction des erreurs aléatoires. Durant la réception, un défoligae (*Deinterleaving*) est utilisé pour neutraliser l'effet de l'interfolieur (*Interleaver*).

2.1.4 Codage cyclique

Les codes cycliques représentent une classe des méthodes de codage linéaire, dont le décalage cyclique d'un « *Code Word* » aléatoirement choisi C est aussi un mot code. Il est d'autant plus simple de représenter chacun de ces mots codes par un polynôme appelé polynôme des mots codes (*Code Word Polynomial*). Un polynôme de mot code correspondant à $C = (c_1, c_2, c_3, ..., c_{n-1}, c_n)$ et défini par :

$$C(p) = \sum_{i=1}^{n} c_i p^{n-1} = c_1 p^{n-1} + c_2 p^{n-2} + ... + c_{n-1} p + c_n \tag{5}$$

Le polynôme de mot code de ($C^{(1)} = (c_2, c_3, ..., c_{n-2}, c_{n-1}, c_n, c_n)$), le décalage cyclique de C est

$$C^{(1)} = c_2 p^{n-1} + c_3 p^{n-2} + ... + c_{n-1} p^2 + c_n p + c_1 \tag{6}$$

Avec une relation entre $C^{(1)}(p)$ et $C(p)$ [BOS99]

$$C^{(1)}(p) = pC(p) \qquad \forall \text{ modulo } (p^n + 1) \tag{7}$$

D'une façon générale, i décalage du polynôme de mot code est :

$$C^{(i)}(p) = p^i C(p) \qquad \forall \text{ modulo } (p^n + 1) \qquad (8)$$

Le fait que n'importe quel polynôme de mot code est un produit d'un générateur polynomial avec le polynôme de la séquence d'information [PRO94] implique que $C = (c_1, c_2, c_3, ..., c_{n-1}, c_n)$, est une convolution discrète des séquences $U = (u_1, u_2, u_3, ..., u_k)$ et $P = (1, p_2, p_3, ..., p_{n-k}, 1)$. Ce qui est très important quant au design d'un encodeur cyclique. En comparant un code cycle par rapport aux autres classes des encodeurs d'un codage linéaire par Bloc, un encodeur cyclique possède une structure interne très facile à implémenter avec de simples registres à décalage.

2.2 Codage Convolutionnel

Un processus de codage linéaire et invariant dans le temps, avec une contrainte de longueur finie, est appelé codage convolutionnel (CC) présenté à la Figure 2.4. La représentation du CC varie selon le stade de mise en œuvre [PRO94]. Dans le cas d'une implémentration VLSI, un registre à décalage permet la réalisation d'un code convolutionnel sur silicium. Tandis que pour un processus de décodage séquentiel, la représentation du (CC) apparaît selon l'un des schémas de représentatins suivants :

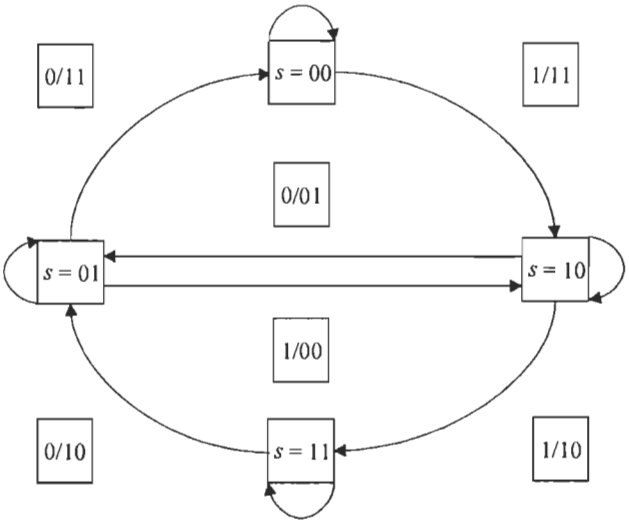


Figure 2.2 Diagramme d'état d'un Encodeur Convolutionnel

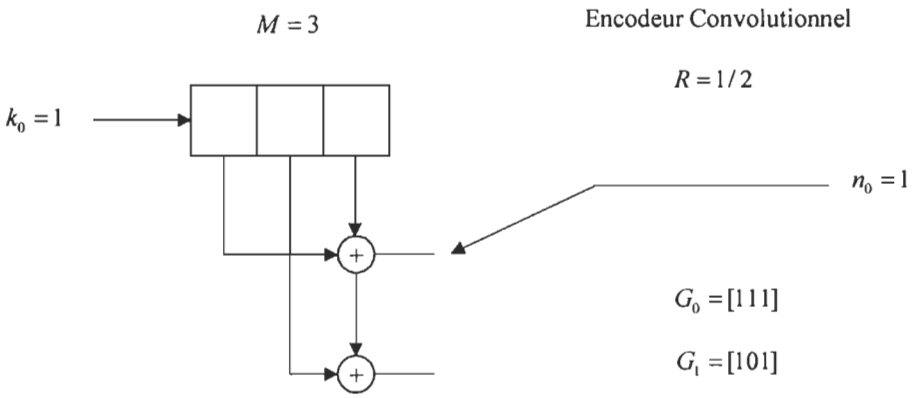


Figure 2.3 Polynôme Générateur (Encodeur Convolutionnel)

a) Treillis : une représentation temporelle des différentes transitions entre états, utilisée principalement par l'algorithme de décodage de Viterbi (Fig. 2.5).

b) Matrice : une representation utulisee lors d'un decodage par syndrome [BOS99] .

c) Polynôme : Pour la vérification de la fiabilité du codage (*Catastrophique/non catastrophique code*) (Fig. 2.3).

d) Diagramme d'état : représentation des propriétés de distance entre tous les états possibles (Fig. 2.2).

L'encodeur convolutionnel simple est une machine déterministe à nombre fini d'états, par conséquent toutes les transitions probables entre ces états peuvent être prédéfinies, condition essentielle pour la mise en œuvre d'un algorithme de décodage.

Pour un encodeur convolutionnel de rapport R = 1/2 et de contrainte de longueur K, l'entrée de l'encodeur en temps k est le bit u_k et le mot code (*Code Word*) correspondant est C_k est le couple binaire (X_k, Y_k) avec :

$$X_k = \sum_{i=0}^{L-1} g_{1i} u_{K-i} \qquad \text{modulo 2} \qquad g_{1i} = 0,1 \tag{9}$$

$$Y_k = \sum_{i=0}^{L-1} g_{2i} u_{K-i} \qquad \text{modulo 2} \qquad g_{2i} = 0,1 \tag{10}$$

Où $G_1 : \{g_{1i}\}, G_2 : \{g_{2i}\}$ sont les deux polynômes générateurs de l'encodeur.

D'autre part, l'information en sortie issue de cette encodeur $C_k(X_k, Y_k)$ Fig. 2.6 n'est pas en fonction du bit d'entrée u_k seule, mais de toute la séquence $u_{k-1}, u_{k-2}, ..., u_{k-M}$ de l'information mémorisée, M représente le nombre de cases mémoires internes.

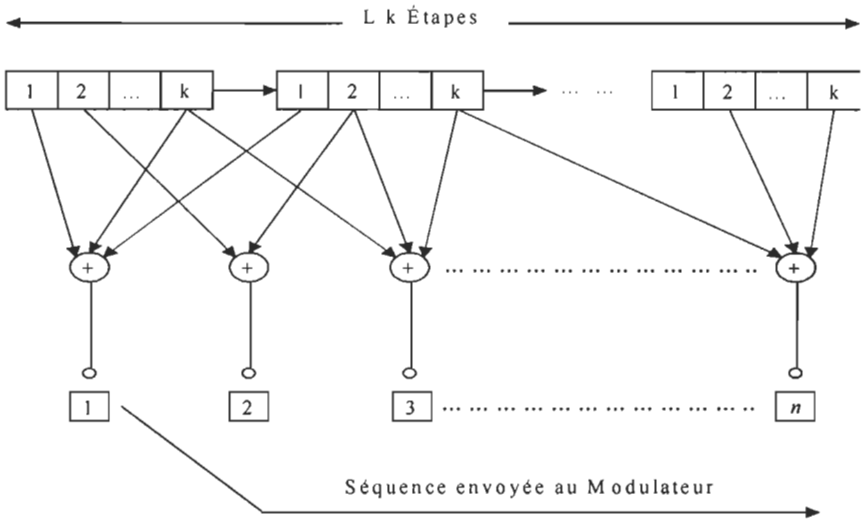


Figure 2.4 Encodeur Convolutionnel

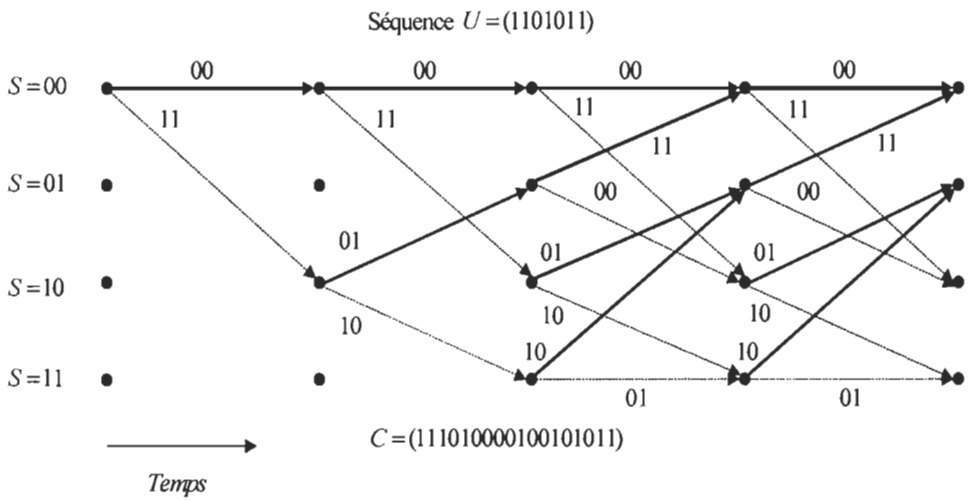


Figure 2.5 Diagramme de Treillis (Encodeur Convolutionnel)

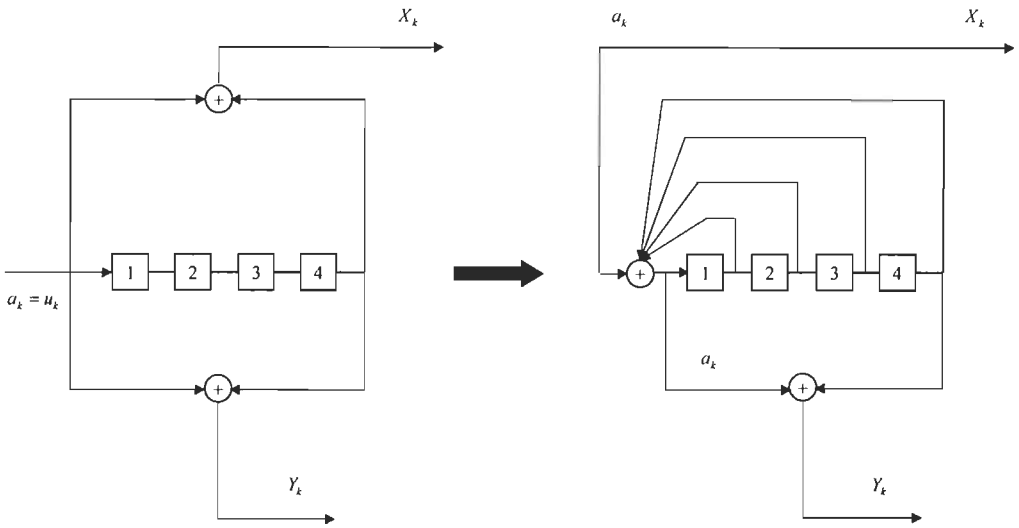


Figure 2.6 Codage Convolutionnel Systématique Récursive (Schéma de principe)

Bien que cet effet de mémoire donne plus de complexité aux différents algorithmes de décodage, le plus grand avantage reste la possibilité de donner une plus grande immunité au bruit, pour les informations codées avant la transmission.

Il est bien connu que BER d'un encodeur convolutionnel simple NSC (*Non Systematic Code*) est inférieur à celui d'un encodeur systématique, d'une valeur de mémoire interne M, pour des valeurs de SNR (*Signal Noise Rate*) élevées. Dans son article [BERa93], Berrou a proposé une nouvelle classe d'encodeur convolutionnel systématique et récursive RSC pour de nouvelles applications. Un RSC binaire de rapport $R=1/2$ est obtenu à partir d'un NSC code, en utilisant une boucle de retour (*Feedback loop*) Fig. 2.6, par l'affectation direct d'une des sorties X_k ou Y_k, vers l'entrée u_k. Pour un RSC code, l'entrée du registre à décalage (Mémoire) n'est plus égale à u_k, mais à un nouveau variable a_k. En substituant a_k à la place de u_k, on a :

$$a_k = u_k + \sum_{i=1}^{L-1} \gamma_i a_{k-i} \tag{11}$$

La structure du treillis est identique pour le codage en NSC ou le RSC, avec la même (*Free Distance*). Cependant, la différence entre les deux codes est que les séquences de sortie $\{X_k\}$, $\{Y_k\}$ ne correspondent plus à la séquence d'entrée $\{u_k\}$, que ce soit pour un NSC ou un RSC.

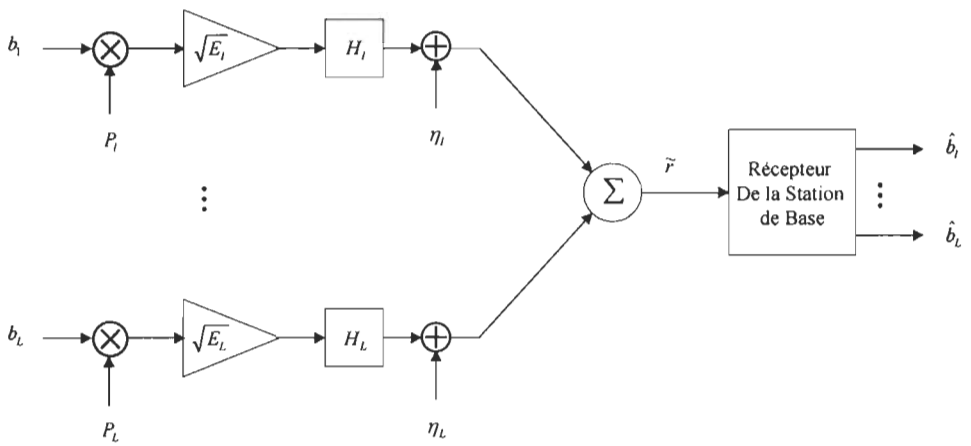
2.3 La Détection Multi-Usagers

La troisième génération (3G) des communications sans fil devrait assurer un débit de 2Mbits/s pour les téléphones cellulaires, 20Mbits/s pour les PCS compatible avec tous les pays, flexible pour l'ajout de nouveaux services, avec l'utilisation des bandes de fréquences qui interfèrent le moins possible avec les systèmes existants. La détection Multi-Usagers a le potentiel d'accroître grandement la capacité des systèmes DS-CDMA, en comparaison avec les approches conventionnelles basées sur le model TDMA [PET95].

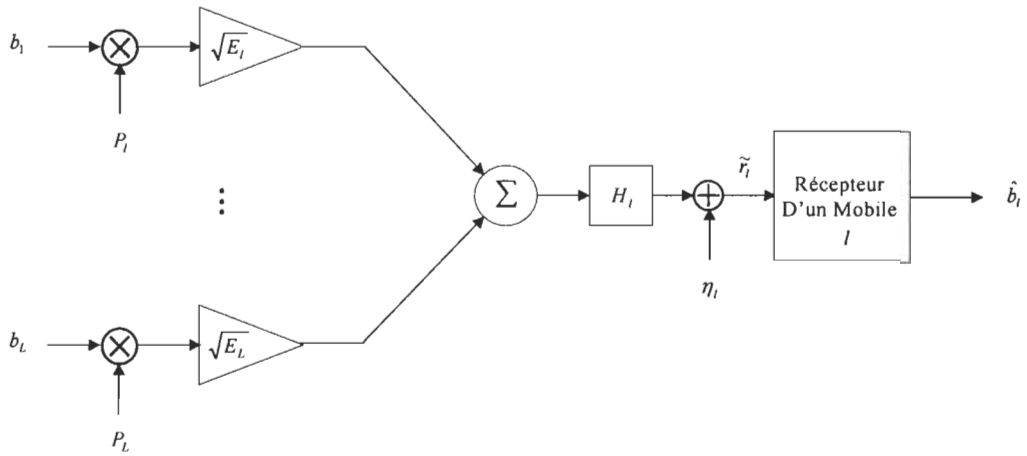
Les algorithmes implémentés dans les récepteurs dépendent de la position de ces derniers; par conséquent, on peut distinguer deux représentations possibles de bande de base d'un système de DS-CDMA selon que le récepteur se trouve dans la station de base (*Uplink*) comme le montre la Figure 2.7-a [WOO99], ou dans le mobile (*Downlink*) comme le montre la Figure 2.7-b [WOO99].

Dans le premier cas, toutes les signatures (Codes Pseudo Aléatoires) des différents utilisateurs sont connues par la station de base, alors que dans le deuxième cas, seule la signature du mobile en question est connue. Cependant, il existe d'autres aspects et d'autres problèmes pour l'implémentation d'un système mobile, nous pouvons citer entre autres le codage canal [HAGa96], [ZHA00] et [TIN00], les antennes intelligentes permettant une optimisation et un contrôle plus strict de la puissance [LEH99], [CHR00] et [KIM00], les boucles de phase et de code [HUA98].

Le récepteur devrait extraire l'information utile dotée de deux types d'interférences éventuelles; l'interférence intersymboles ISI (*InterSymbol Interference*) et les interférences d'accès multiples MAI (*Multiple Access Interference*) [WOO99] provenant des autres usagers, provocant une non-orthogonalité des codes pseudo-aléatoires dans la réception.



a) Mobiles vers une Station de Base (Uplink).



b) Une Station de Base vers un Mobile (Downlink).

Figure 2.7 Représentation en bande de base d'un système DS-CDMA avec b_l l'information à transmettre à l'utilisateur l étalée avec le code pseudo aléatoire p_l. Les gains $\sqrt{E_l}$ sont pour le contrôle de puissance, H_l le sous canal, η_l le bruit thermique.

Trois principales approches distinguent les modèles de récepteurs mis en pratique :

A- Un récepteur sous optimal caractérisé par une banque de filtre appariés (*Matched filters*) [PRO95], [KLE96], en considérant que l'utilisateur est seul. Cette approche reste inefficace, car l'interférence est considérée comme un bruit qui nécessite un contrôle strict de puissance.

B- Une approche utilisant une combinaison de diversité [PET95] pour traiter les trajets multiples *(Multipath)*, ce qui définit un récepteur Rake.

C- Une autre approche utilisant des récepteurs résistants aux problèmes du loin/proche (*near/far*) au niveau de la station de base [WOO99].

Ces récepteurs nécessitent une connaissance des séquences utilisées par les autres utilisateurs ainsi que les délais de leurs signaux.

D- Les algorithmes adaptatifs [RAP94], [LAT00], [WAN99] et [DAS98], consistent à implémenter plusieurs structures de récepteurs à utilisateurs multiples, mais pour des dynamiques rapides, avec une vitesse de convergence lente, les algorithmes adaptatifs peuvent s'avérer inefficaces.

Les différentes approches définissent les principaux modèles de récepteurs mis en pratique.

2.3.1 Le récepteur Rake

Le récepteur *Rake* est constitué de corrélateurs Fig. 2.8 [PRA98]; chacun reçoit un signal multi-trajets, pour être combinés après un désétalement effectué par les corrélateurs. L'amélioration des performances est due au caractère indépendant des évanouissements spectraux, des signaux à trajets multiples reçus à l'entrée des corrélateurs, les coefficients τ_1, τ_2, τ_3 sont les retards accumulés sur chacun des trajets, a_1, a_2, a_3 les facteurs d'atténuations causées par le canal.

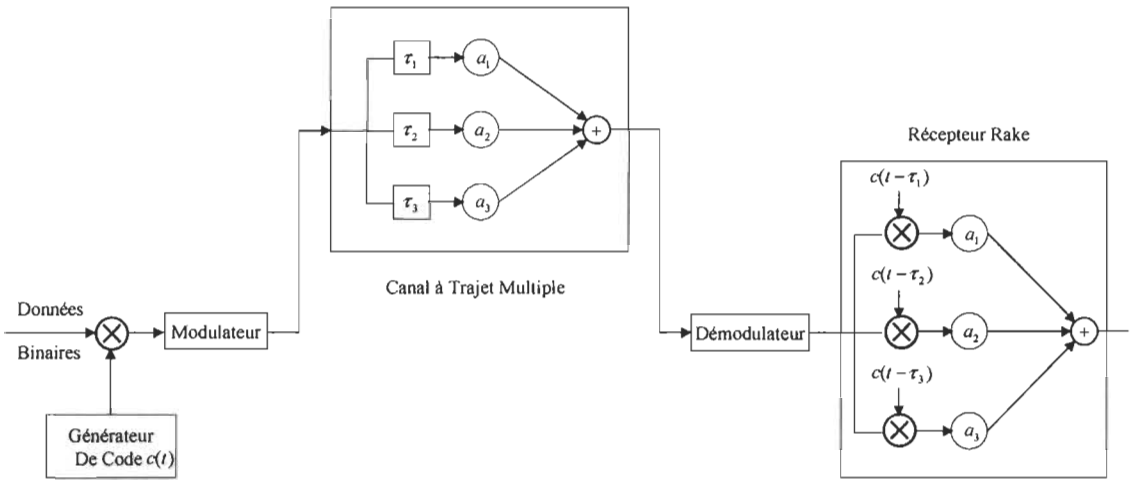


Figure 2.8 Schéma de principe d'un Récepteur Rake.

2.3.2 Récepteur Forçage à Zéro

C'est la combinaison d'un filtre de pré-blanchissement Fig. 2.9, ou de décoloration du bruit, avec les filtres appariés (*Matched filters*) du récepteur Rake, suivie d'un échantillonneur à la cadence des symboles transmis.

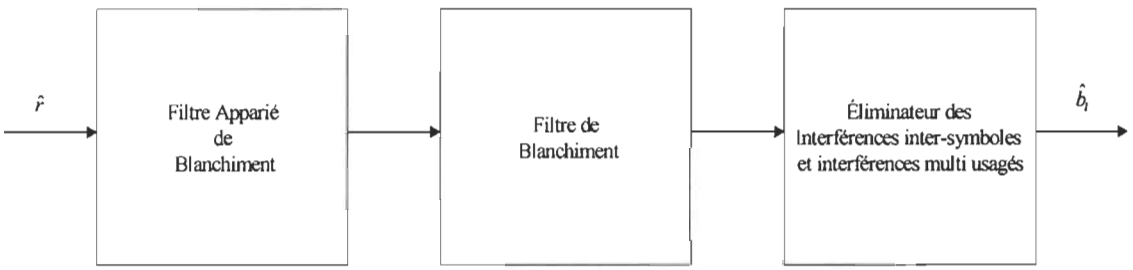


Figure 2.9 Récepteur Forçage à Zéro sous sa forme linéaire.

Les interférences ISI et MAI sont ainsi considérées comme des bruits ; l'égalisateur forçage à zéro (*Zero Forcing*) minimise l'erreur entre la séquence reçue et la séquence équivalente issue des symboles estimés par le filtre [KLE96]. Les performances sont ainsi meilleures que celle du Rake, mais insuffisantes. En effet, la

décoloration entre les symboles envoyés et les interférences ne peut être obtenue, du fait que la technique ne tient pas compte du niveau du bruit dans l'algorithme.

2.3.3 Récepteur MMSE

Le MMSE (*Minimum Mean square Error*) Fig. 2.10 est une extension du récepteur de forçage à zéro par un estimateur de Wiener [KLE96]. La dégradation de performance du récepteur à forçage zéro est améliorée par l'estimateur de Wiener qui minimise l'erreur quadratique moyenne des données b_1, à partir de la sortie du forçage à zéro.

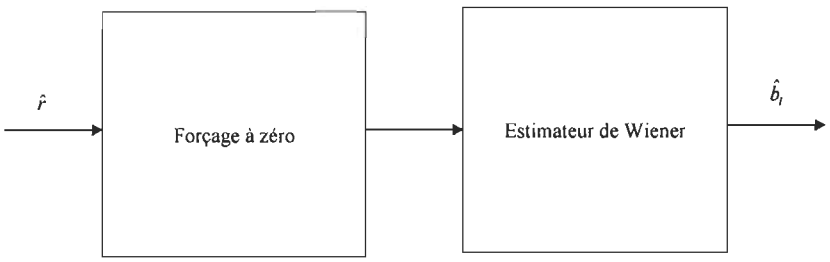


Figure 2.10 Récepteur MMSE optimal.

Le MMSE est souvent cité comme l'égalisateur le plus performant [WOO99], ce qui lui a permis d'avoir plusieurs versions, selon l'endroit du récepteur (Mobile ou station de base) [LAT00].

3. TECHNIQUES ET ALGORITHMES DE DÉCODAGE

Les techniques de décodage dépendent essentiellement de la nature du signal reçu à partir du (*Rake Receiver*). L'ajout de l'information redondante à l'information utile était utilisé comme simple moyen de détection des erreurs, connu sous le nom de (*Automatic Repeat Request)*. (*Forward Error Correction*) est une autre méthode de détecter et de corriger les erreurs sur les données reçues par transmission dans un domaine bruité. Cette méthode permet d'atteindre des gains considérables en codage et de réduire par conséquent le rapport Signal sur bruit. La problématique reliée à une éventuelle application de cette méthode réside dans le fait que cela nécessite la transmission des données, avec un rapport de transmission plus élevé, en vue de bien transmettre les mêmes informations utiles par unité de temps, ce qui nécessite une largeur de bande encore plus grande.

D'autre part, dans les systèmes de communication sans fil, le critère de performances le plus important est sans doute de pouvoir transmettre les données utiles avec le minimum de puissance possible, afin d'assurer une plus grande longueur de vie pour les batteries, ainsi qu'un minimum d'interférence entre les canaux. Selon la théorie du Codage, augmenter la largeur de mot code ou la mémoire de l'encodeur, en utilisant de « *Bons* » codes, permettent théoriquement de s'approcher encore plusde la limite de Shannon [SHA48]. Cependant, trouver ces

codes et l'implémentation en temps réel de ces mêmes encodeurs a été pendant longtemps un domaine de recherche actif.

Turbo Code [BERa93], [BER96] est une technique très efficace pour détecter et corriger les erreurs, ce qui permet de réaliser une communication fiable avec un taux d'erreurs (BER), près de la limite de Shannon. En codage convolutionnel, les performances sont améliorées par l'augmentation de la contrainte de longueur (*Contraint Length*), par contre le facteur essentiel pour l'amélioration des performances d'un turbo code est due essentiellement à l'utilisation des algorithmes de décodage avec des décisions douces (*Soft Input/Soft Out put*), dans son décodeur.

Une version modifiée de l'algorithme de minimisation de l'erreur de décodage sur chaque bit transmis, connu sous le nom de (*Maximum Aposteriory Algorithm*) de Bahl et al. [BAH74] était le premier algorithme utilisé dans un processus de turbo décodage. Depuis, un grand travail de recherche a été élaboré en vue de réduire la complexité du décodeur, comme l'a suggéré Robertson et al. [ROB95], Berrou et al. [BERb93], ou Battail [BAT87].

Le Goff et al. [GOF94], Wachsmann et Huber [WAC95], Robertson et Worz [ROB98] ont suggéré d'utiliser le Codage en jonction avec des schémas de modulation de largeur de bande efficaces. D'autres avancées ont permis une bonne compréhension des excellentes performances des Codes sont dues à Benedetto et Montorsi [BENa96] [BENb96], Perez et al. [PER96]. Hagenauer et al. [HAGb96] et [PYN97] ont élargi le concept vers le codage concaténé par Bloc. Jung et Nabhan [JUNa94] et [JUN96] ont caractérisé les performances du Turbo Codage, sous la

contrainte d'une transmission avec des trames réduites en longueur, ce qui caractérise les systèmes de transmission de la voix. Avec la collaboration de Blanz, ils ont appliqué les Turbo Codes aux Systèmes CDMA en utilisant la détection en jonction avec la diversité d'antenne [JUNb94]. Barbeluscu et Pietrobon ont montré l'influence du choix de l'interfolieur dans les performances d'un Turbo Encodeur [BAR94].

En général, la structure d'un turbo encodeur est constituée de deux encodeurs convolutionnels identiques et récursifs, avec une sortie systématique (RSC). Pour une entrée identique, les deux sorties de ces encodeurs sont différentes. Le Bloc de N bits d'entrée sont encodés en premier lieu par le premier encodeur, ce même Bloc de données est ensuite interfolié pour être encodé par la suite. Le principal objectif de l'utilisation de l'interfoliage est de disperser aléatoirement l'erreur survenue sur un groupe de bits transmis (erreurs2 d'éclats), de manière à être correctement détectée et corrigée. D'autre part, cela permet d'augmenter encore plus la distance libre (*Free Distance*) de l'encodeur. Pour optimiser la transmission, la perforation (*Puncturing*), peut être utilisée selon un model bien déterminé, entièrement défini par le (*Punctering Matrix*) [ROB98].

L'algorithme MAP était le premier algorithme utilisé dans un processus décodage itérative, permettant ainsi d'estimer les (*Aposteriori Probabilities*) de l'état et la transition observés dans la chaîne d'une source de Markov, dans un bruit sans effet de mémoire. Cet algorithme est optimal en termes de minimiser le BER, contrairement à l'algorithme de Viterbi, qui minimise la probabilité d'un chemin incorrect à travers le treillis sélectionné par le décodeur. Ce qui permet de conclure

que l'algorithme de Viterbi sert à minimiser le nombre de groupe de bits associés à ces chemins dans le treillis, plutôt que le nombre actuel de bits qui sont incorrectement décodés. Dans le souci de réduire la complexité du MAP, une première approximation de ce même algorithme est proposée par Koch et Baier [KOC90], Erfanian et al. [ERF94], connu sous le nom de Max-Log-MAP, qui consiste à transférer le calcul récursive de l'algorithme MAP, dans le domaine logarithmique, en invoquant une approximation qui réduit considérablement sa complexité. Cette approximation rend les performances de Max-Log-MAP sous optimales en comparaison avec le MAP. Cependant, Robertson et al. [ROB95] ont suggéré une autre approximation qui corrige l'approximation de l'algorithme Max-Log-MAP et, par conséquent, d'avoir les mêmes performances du MAP avec une fraction de sa complexité. Cet algorithme sera connu sous le nom du Log-MAP.

D'autre part, une version itérative a été proposée de l'algorithme de Viterbi, connu sous le nom de SOVA [BERb93], [HAG89]. Cet algorithme possède deux modifications par rapport à l'algorithme de Viterbi classique, pour permettre son utilisation dans un processus de décodage itérative. La première modification consiste à tenir compte du (*A-priori*) information dans le sélection du chemin le plus probable dans un treillis, qui définit les bits à décoder. L'autre modification permet de produire un (*Soft Output*) sous forme de (*A-posteriori information*) pour chaque bit décodé.

3.1 Techniques de décodage

3.1.1 Décodage par Décision Douce (*Soft Decision Decoding*)

Dans un canal AWGN, une détection optimale est principalement une détection basée sur la minimisation de la distance Euclidienne entre le signal reçu et le signal transmis. En d'autres termes, une fois la sortie du canal reçue, soit passée par les filtres appariés (*Matched Filters*), on doit choisir le signal message ayant le minimum de distance au sens de Hamming avec le signal reçu. Avec l'utilisation d'une modulation (*Phase Shift Key*), pour la transmission des données codées, le mot code $C_i = (c_{i1}, c_{i2}, ..., c_{in})$ est modélisé (*Mapped*), en une séquence $s_i(t) = \sum_{k=1}^{n} \psi_{ik}(t-(k-1)T)$ avec :

$$\psi_{ik}(t) = \begin{cases} \psi(t) & c_{ik} = 1 \\ -\psi(t) & c_{ik} = 0 \end{cases} \qquad (12)$$

$\psi(t)$ est un signal de durée T est d'énergie E, de valeur nulle en dehors de l'intervalle $[0,T]$. La distance Ecludienne entre deux signaux choisis arbitrairement [PRO95] :

$$\left(d_{ij}^{E}\right)^2 = \sum_{\substack{1 \le k \le n \\ k : c_{ik} \neq c_{jk}}} (\pm 2\sqrt{E})^2 = 4 d_{ij}^{H} E \qquad (13)$$

Ce qui donne une simple relation entre la distance Ecludienne et la distance de Hamming, pour une modulation BPSK. Une modulation optimale consiste alors à faire passer le signal reçu $r(t)$, à travers une série de filtres appariés (*Matched*

Filters), en vue d'obtenir à la fin un vecteur $\mathbf{r}$. Un décodage par décision soft, (*Soft Decision Decoding*), consiste alors à trouver le point le plus proche, au sens Ecludien (*Minimum Ecludean Distance*) du vecteur $\mathbf{r}$. Cette méthode de décodage nécessite un traitement en nombre réel des données.

3.1.2 Décodage par Décision Dure (*Hard Decision Decoding*)

Une technique plus simple et plus souvent utilisée par l'attribution de valeurs binaires (*Hard Binary Decision*) aux composantes du vecteur reçues $\mathbf{r}$, et trouver le mot code *(Code Word)*, le plus proche de $\mathbf{r}$, au sens de Hamming. Il existe trois étapes élémentaires dans le processus de décodage par décision douce (*Hard Decision Decoding*). Premièrement, le signal brut reçu $r(t)$ passe par une série de filtres appariés (*Matched Filters*), l'échantillonneur bloqueur, pour obtenir le vecteur $\mathbf{r}$. Deuxièmement, on compare les éléments du vecteur $\mathbf{r}$, avec des niveaux seuils (*Threshold*), et quantifier chaque élément par un des deux niveaux, ce qui définit le nouveau vecteur $\mathbf{y}$. Finalement, le décodage est réalisé par la recherche du mot code le plus proche du vecteur $\mathbf{y}$, au sens de Hamming.

Il est prouvé que la différence entre les performances d'un décodage par décision douce et un décodage par décision dure est autour de 2dB [PRO94] pour un canal AWGN, par conséquent, la probabilité d'erreur d'une décision dure, dont la puissance est supérieure à 2dB, est comparable à celle d'une décision douce. D'autre part, il est encore prouvé qu'au lieu de quantifier tous les éléments du vecteur $\mathbf{r}$, selon deux niveaux distincts, une quantification par huit niveaux (trois bits par élément) réduit

l'écart de performances entre les deux à 0.1dB. Cette quantification à plusieurs niveaux, qui constitue un compromis entre le modèle de décodage par décision douce (Précision Infinie) et le décodage par décision dure.

3.2 Algorithmes de Décodage

3.2.1 Algorithme de décodage Viterbi

On a vu précédemment, dans les différents schémas pour encodage linéaire par Bloc, qu'il existe deux possibilités de procéder au traitement des données à décoder, à savoir une décision douce (*Soft Décision*), décision dure (*Hard décision*). Pour une décision douce, le vecteur $\mathbf{r}$, représentant la sortie d'un filtre apparié (*Matched Filters*), est comparé aux différents points du signal en vue de choisir le point qui lui est le plus proche au sens Euclidean.

Pour une décision douce, ce même vecteur $\mathbf{r}$ est transformé en premier lieu en une séquence binaire $\mathbf{y}$, en procédant à une (Décision Dure) sur chacun de ces éléments, ce qui permet ensuite de choisir le mot code qui lui est le plus proche, dans le sens de Hamming. En plus clair, il est évident que pour les deux approches, la principale tâche reste à déterminer le chemin dans le treillis, à distance minimale par rapport à une séquence donnée, ce problème apparaît dans plusieurs aspects du domaine de la communication, particulièrement pour l'estimation de la séquence la plus probable, transmise dans un canal à bande limitée avec interférence

inter-symboles, reconnaissance de la voix ou d'autres problèmes de classification des models (*Pattern classification*).

Tous ces problèmes sont essentiellement identiques, basés sur un algorithme de recherche optimale dans un treillis, connu sous le nom de l'algorithme de Viterbi . Cet algorithme procure une solution optimale pour tous ces problèmes en même temps. Dans un processus à décision dure (*Hard décision*) de décodage d'un codage convolutionnel, on cherche à trouver un chemin dans le treillis dont le mot code (*Code Word*) C est de distance minimale dans le sens de Hamming, par rapport à la séquence reçue et quantifiée $\mathbf{y}$. Le canal de communication est supposé sans effet de mémoire (du fait que le bruit additionnel qui vient s'ajouter aux informations transmises est supposé être un bruit blanc). En dénotant la séquence de bits correspondant à la i^{eme} branche par c_i et y_i respectivement, avec $1 \le i \le m$ (m le nombre total de branches), et pour chaque valeur C_i et $\mathbf{y}_i$ sont de longueur n. La distance de Hamming entre C et $\mathbf{y}$ peut être exprimée en

$$d(C,\mathbf{y}) = \sum_{i=1}^{m} d(C_i,\mathbf{y}_i) \tag{14}$$

En supposant traiter avec un encodeur convolutionnel de valeur $K = 1$ (Fig. 2.5), le chemin optimal à un certain point passe par l'état S, cet état est connecté par deux chemins différents aux états précédents S_1 et S_2. Si l'on cherche à déterminer lequel des deux chemins minimise la distance du chemin total dans le treillis, on doit en premier lieu additionner toutes les distances métriques relatives aux états S_1 et

l'état S_2 au métriques de branches de connexion avec l'état S. Évidemment, la branche permettant de cumuler une distance métrique minimale jusqu'à l'état S est retenue, le chemin équivalent englobant cette branche est ensuite appelé (*Survivor)*. Ce même processus est appliqué aux états suivants, jusqu'à atteindre l'état tout zéro à la fin de treillis.

3.2.2 Algorithme de décodage MAP

En 1974, un algorithme connu sous le nom de MAP a été proposé par Bahl et al. [BAH74], dans le but d'estimer l'a-posteriori, dans le calcul des probabilités des états de transitions d'une source de Markov, observé dans un canal bruité et sans effet de mémoire.

Bahl et al, ont démontré la possibilité de l'application de l'algorithme au décodage des procédés de codage par Blocs, ou par codage convolutionnel. L'algorithme est optimal en termes de minimisation du BER, de l'information à décoder, contrairement à l'algorithme de Viterbi [FOR73], qui minimise l'erreur de probabilité sur un trajet sélectionné par le décodeur dans le treillis. Néanmoins, pour la plupart des applications, la performance des deux algorithmes reste presque identique [BAH74].

L'algorithme MAP examine tous les chemins possibles à travers un treillis d'un codeur convolutionnel. Du coup, sa mise en œuvre semblait être compromise pour la plupart des systèmes, vu sa complexité de calcul, jusqu'à la découverte de (Turbo Code), ce qui a permis une large utilisation de cet algorithme. Le MAP ne

produit pas seulement une estimation de la séquence de bits transmis, mais aussi une probabilité de véracité pour chaque bit décodé, ceci est primordial pour un décodage itératif, comme celui proposé par Berrou et al [BERa93].

L'algorithme MAP calcule pour chaque bit décodé u_k la probabilité selon laquelle le bit transmis était +1 ou −1, étant donné une séquence $\mathbf{y}$ reçue de symboles transmis. Ceci est équivalent de trouver *a-posteriori* (*Log Likelihood Ratio*)

$$L(u_k / \mathbf{y}) = \log\left(\frac{P(u_k = +1/\mathbf{y})}{P(u_k = -1/\mathbf{y})}\right) \tag{14}$$

Si l'état état précédent $S_{k-1} = s'$ et l'état présent $S_k = s$ sont connus dans la treillis, le bit d'entrée u_k à l'origine de cette transition sera après conséquent connu aussi [WOO00].

En utilisant le théorème de bayes et en tenant compte du fait que les transitions entre les états précédents S_{k-1} et les états présents S_k dans le treillis sont mutuellement exclusifs, (les états S_{k-1} et les états S_k ne peuvent apparaître en même temps), nous pourrons alors écrire :

$$L(u_k / \mathbf{y}) = \log\left(\frac{\sum_{u_u=+1}(s',s) \Rightarrow P(S_{k-1} = s' \wedge S_k = s \wedge \mathbf{y})}{\sum_{u_u=-1}(s',s) \Rightarrow P(S_{k-1} = s' \wedge S_k = s \wedge \mathbf{y})}\right) \tag{15}$$

Où $(s',s) \Rightarrow +1$ représente l'ensemble des transitions entre les états précédents $S_{k-1} = s'$, et les états présents $S_k = s$, pouvons apparaître si le bit transmis $u_k = +1$; et similairement pour $(s',s) \Rightarrow u_k = -1$. En considérant la probabilité $P(S_{k-1} = s' \wedge S_k = s \wedge \underline{y})$, la séquence $\mathbf{y}$ peut être partagée en trois sections; la séquence reçue avant la présente transition $\mathbf{y}_{j<k}$, la séquence reçue correspondant à la présente transition $\mathbf{y}_{j=k}$, la transition reçue après la présente transition $\mathbf{y}_{j>k}$, nous pouvons ainsi écrire :

$$P(s' \wedge s \wedge \mathbf{y}) = P(s' \wedge s \wedge \mathbf{y}_{j<k} \wedge \mathbf{y}_{j=k} \wedge \mathbf{y}_{j>k}) \tag{16}$$

En utilisant le théorème de Bayes et en assumant que le canal de transmission des données est un canal sans effet de mémoire, la séquence future $\mathbf{y}_{j>k}$ ne dépend que de la séquence présente $\mathbf{y}_{j=k}$ et non pas de la séquence précédente $\mathbf{y}_{j<k}$, on pourrait donc écrire :

$$P(s' \wedge s \wedge \mathbf{y}) = \beta_k(s).\gamma_k(s',s).\alpha_{k-1}(s') \tag{17}$$

Où $\alpha_{k-1}(s') = P(S_{k-1} = s' \wedge \mathbf{y}_{j<k})$ représente la probabilité que le treillis est à l'état s' , en temps $k-1$, et $\beta_k(s) = P(\mathbf{y}_{j>k} \backslash S_k = s)$ est la probabilité que la future séquence à recevoir sera $\mathbf{y}_{j>k}$, si le treillis est en état s en temps k. Et finalement, $\gamma_k(s',s) = P(\{\mathbf{y} \wedge S_k = s\} \backslash S_{k-1} = s')$ représente la probabilité que le treillis transite vers l'état s, et la séquence reçue du canal en conséquence de cette transition est $\mathbf{y}_k$,

si le treillis était dans l'état s' en temps $k-1$. Cette équation montre que la probabilité $P(S_{k-1} = s' \wedge \mathbf{y}_{j<k})$, que le treillis prend la transition entre de l'état $S_{k-1} = s'$ vers l'état $S_k = s$ avec une séquence reçue $\mathbf{y}$, peut être exprimée en un produit de trois termes de probabilité, $\alpha_{k-1}(s'), \gamma_k(s', s), \beta_k(s)$.

L'algorithme MAP calcule les probabilités $\alpha_{k-1}(s'), \beta_k(s)$ de tous les états s' du treillis pour $k = 0,1,2....N-1$, et $\gamma_k(s', s)$ de toutes les transitions possibles entre un état $S_{k-1} = s'$ vers l'état $S_k = s$, pour $k = 0,1,2....N-1$, ces valeurs donnent le $LLR(u_k \mid \mathbf{y})$:

$$L(u_k / \mathbf{y}) = \log\left(\frac{\sum_{u_k=+1}(s', s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s', s).\beta_k(s)}{\sum_{u_k=-1}(s', s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s', s).\beta_k(s)}\right) \tag{18}$$

Pour une implémantation d'un algorithme MAP, le calcul des probabilités $\alpha_{k-1}(s'), \gamma_k(s', s), \beta_k(s)$ doit être récursif; en effet, on peut écrire :

$$\alpha_k(s) = P(S_k = s \wedge \mathbf{y}_{j<k+1}) \tag{19}$$

$$= P(s \wedge \mathbf{y}_{j<k} \wedge \mathbf{y}_k) \tag{20}$$

$$= \sum_{s'} P(s \wedge s' \wedge \mathbf{y}_{j<k} \wedge \mathbf{y}_k) \tag{21}$$

En utilisant le théorème de baye's et en assumant encore une fois que le canal de transmission de données est sans effet mémoire alors :

$$\alpha_k(s) = \sum_{s'} P(s \wedge s' \wedge \mathbf{y}_{j<k} \wedge \mathbf{y}_k) \tag{22}$$

$$= \sum_{S'} P\left(\{s \wedge \mathbf{y}_k\} \mid \{s' \wedge \mathbf{y}_{j<k}\}\right) P\left(s' \wedge \mathbf{y}_{j<k}\right) \tag{23}$$

$$= \sum_{S'} P\left(\{s \wedge \mathbf{y}_k\} \mid s'\right) P\left(s' \wedge \mathbf{y}_{j<k}\right) \tag{24}$$

$$= \sum_{s'} \alpha_{k-1}(s').\gamma_k(s', s) \tag{25}$$

Par conséquent, en supposant qu'à l'état initial le treillis se trouve à l'état $S_0 = 0$ avec :

$$\alpha_0(S_0 = 0) = 1 \tag{26}$$

$$\alpha_0(S_0 = s) = 0 \text{ pour tout } s \neq 0 \tag{27}$$

Les valeurs de $\alpha_{k-1}(s')$ seront calculées d'une manière récursive, une fois que les valeurs de $\gamma_k(s', s)$ sont connues.

De façon similaire, les valeurs $\beta_k(s)$ sont calculées en écrivant :

$$\beta_{k-1}(s') = P(\mathbf{y} \mid s') \tag{28}$$

$$= \sum_{s} \beta_k(s).\gamma_k(s', s) \tag{29}$$

Le calcul de $\gamma_k(s',s)$ est effectué à la base de la séquence reçue du canal de transmission, ainsi que les informations préliminaires (*A-priori Information*) disponibles ; en utilisant le théorème de baye's, on a :

$$\gamma_k(s',s) = P(\{\mathbf{y}_k \wedge s\} \mid s') \tag{30}$$

$$= P(\mathbf{y}_k \mid \{s' \wedge s\}).P(s \mid s') \tag{31}$$

$$= P(\mathbf{y} \mid \{s' \wedge s\}).P(u_k) \tag{32}$$

$$= P(\mathbf{y}_k \mid \mathbf{x}_k).P(u_k) \tag{33}$$

u_k est le bit d'entrée nécessaire pour engendrer une transition de l'état $S_{k-1} = s'$ à l'état $S_k = s$, $P(u_k)$ représente l'information préliminaire sur ce même bit d'entrée, et $\mathbf{x}_k$ le mot code transmis associé à cette transition.

L'information préliminaire (*A-priori information*), $P(u_k)$ est collecté selon un procédé de décodage itérative, de la sortie d'un précédant décodeur, et la probabilité $P(\mathbf{y} \mid \mathbf{x})$ est donnée avec la supposition que le canal de transmission est un canal Gaussien, sans effet de mémoire, avec une modulation BPSK [WOO00] :

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{l=1}^{n} P(\mathbf{y}_{kl} \mid \mathbf{x}_{kl}) \tag{34}$$

$$= \prod_{l=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp^{\left(\frac{E_b R}{2\sigma^2}(y_{kl} - \alpha x_{kl})^2\right)} \quad (35)$$

Où x_{kl} , y_{kl} représentent les bits transmis et reçue respectivement, des mots codes $\mathbf{x}$ et $\mathbf{y}$.

n : le nombre de bits dans chaque mot code

E_b : l'énergie par bit transmis.

σ^2 : La variance du bruit.

a : Amplitude d'atténuation ($a = 1$ pour un canal AWGN sans atténuation).

3.2.3 Algorithme de décodage Max Log-MAP

Bien que l'algorithme MAP soit beaucoup plus complexe que l'algorithme de décodage (*Hard décision*) de Viterbi, la performance des deux algorithmes en pratique reste presque identique. La découverte du décodage itérative (Turbo Code) a renouvelé l'importance à l'algorithme MAP, en réalisant que sa complexité pouvait être réduite, sans pour autant nuire à la qualité de ses performances.

L'algorithme Max-Log-MAP était proposé au début par Koch et Baier [KOC90] et Erfanian et al [ERF94]. Cette technique apporte une simplification du MAP, en transformant la récursivité, dans le domaine logarithmique, cette approximation réduit considérablement la complexité de cet algorithme, mais d'autre part, ces performances sont considérées comme étant sub-optimales, en comparaison avec l'algorithme de source MAP. L'algorithme MAP calcule la valeur a-

posteriori $L(u_k \mid \mathbf{y})$, en utilisant les valeurs de $\alpha_{k-1}(s'), \gamma_k(s', s), \beta_k(s)$, calculées de manière récursive. L'algorithme Max-Log-MAP simplifie ce calcul, en le transférant dans le domaine logarithmique par l'approximation :

$$\log\left(\sum_i \exp^{x_i}\right) \approx \max_i(x_i) \tag{41}$$

$\max_i(x_i)$ exprime la valeur maximale de x_i ; par conséquent, on peut définir :

$$A_k(s) \cong \log(\alpha_k(s)) \tag{42}$$

$$B_k(s) \cong \log(\beta_k(s)) \tag{43}$$

$$\Gamma_k(s', s) \cong \log(\gamma(s', s)) \tag{44}$$

En appliquant cette nouvelle définition sur $\alpha_{k-1}(s'), \gamma_k(s', s), \beta_k(s)$, on a :

$$A_k(s) \cong \log(\alpha_k(s)) \tag{45}$$

$$= \log\left(\sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)\right) \tag{46}$$

$$= \log\left(\sum_{s'} \exp[A_{k-1}(s')] + \Gamma_k(s', s)\right) \tag{47}$$

$$\approx \max_i\left(A_{k-1}(s') + \Gamma_k(s', s)\right) \tag{48}$$

Cette équation implique que pour chaque chemin dans le treillis, à partir de chaque état précédent, jusqu'à l'état présent $S_k = s$, l'algorithme additionne la valeur métrique $\Gamma_k(s', s)$, relative à la branche de transition entre les états s' et s, à $A_{k-1}(s')$, afin de donner une nouvelle estimation $\widetilde{A}_k(s)$ correspondant au chemin suivi dans le treillis. La nouvelle valeur de $A_k(s)$ est la valeur maximale des estimations $\widetilde{A}_k(s)$, Cette valeur représente la probabilité que le treillis est dans l'état $S_k = s$ en temps k, ayant reçu la séquence $\mathbf{y}_{j<k}$. A cause de l'approximation dans le Max Log-MAP, uniquement le chemin avec le maximum de probabilité (ML) parcourant l'état $S_k = s$ est considéré, on déduit alors que la valeur de $A_k(s)$, dans l'algorithme Max-Log-MAP, donne la probabilité sur le chemin le plus probable sur le treillis, passant par l'état $S_k = s$, au lieu de donner une valeur de probabilité sur tous les chemins passant par cet état. Cette approximation est une des raisons des performances sub-optimales du Max-Log-MAP.

Le calcul récursive des valeurs de $A_k(s)$ est pratiquement la même manière de calcul récursif pour l'algorithme de Viterbi (pour chaque paire de chemins fusionnant sur l'état $S_k = s$, le chemin survivant (*The Survivor*) est déterminé en utilisant deux additions et une comparaison.

De la même façon, on peut calculer récursivement les valeurs de $B_{k-1}(s')$ et $\Gamma_k(s', s)$:

$$B_{k-1}(s') \cong \log\left(\beta_{k-1}(s')\right) \tag{49}$$

$$\approx \max_{s}\left(B_k(s) + \Gamma_k(s',s)\right) \tag{50}$$

Ceci est équivalent au calcul en récursif utilisé dans Viterbi, sauf la différence que cette récursivité se fait de gauche à droite dans le treillis.

Pour le calcul des valeurs de $\Gamma_k(s',s)$ on a ;

$$\Gamma_k(s',s) \cong \log(\gamma_k(s',s)) \tag{51}$$

$$= Cte + \frac{1}{2}u_k L(u_k) + \frac{L_c}{2}\sum_{l=1}^{n} y_{kl}x_{kl} \tag{52}$$

Où C est une constante indépendante de u_k ou du code transmis. Cependant, la valeur métrique $\Gamma_k(s',s)$ est la même utilisée par Viterbi avec, en plus, une information en priori $u_k L(u_k)$, c'est pourquoi le terme de corrélation $\sum_{l=1}^{n} y_{kl}x_{kl}$ est pondéré par l'indice de fiabilité du canal L_c. Finalement, on peut écrire que l'information en a-posteriori $L(u_k \mid \mathbf{y})$, que calcule l'algorithme Max Log-MAP.

$$L(u_k / \mathbf{y}) = \log\left(\frac{\sum_{u_k=+1}(s',s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}{\sum_{u_k=-1}(s',s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}\right) \tag{53}$$

$$\approx \max_{\substack{(s',s)\Rightarrow \\ u_k=+1}}\left(A_{k-1}(s') + \Gamma_k(s',s) + B_k(s)\right) - \max_{\substack{(s',s)\Rightarrow \\ u_k=-1}}\left(A_{k-1}(s') + \Gamma_k(s',s) + B_k(s)\right) \tag{54}$$

L'algorithme MAP calcule pour chaque bit u_k l'information en a-posteriori $L(u_k \mid \mathbf{y})$, en considérant toute transition entre l'état S_{k-1} et l'état S_k. Ces transitions sont ainsi groupées avec celles pouvant être causées par un bit en entrée $u_k = +1$ et celles causées par l'entrée $u_k = -1$. Avec les deux groupes de transitions possibles représentant la valeur maximale de $\left(A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)\right)$, le LLR (Log Liklehood Ratio) est calculé à partir de ces deux 'meilleures' transitions possibles.

La complexité de l'algorithme Max-Log-MAP n'est pas significativement élevée par rapport à celle de Viterbi. Il est à noter que cela nécessite deux récursivités au lieu d'une pour l'algorithme de Viterbi, la valeur métrique par branche (*Branch Metric*), quand elle nécessite l'ajout d'un terme relatif à l'information en priori $u_k L(u_k)$, c'est pourquoi Viterbi [VIT96], [VIT97] considère que l'algorithme Max-Log-MAP est à peine trois fois plus complexes que l'algorithme de Viterbi, ainsi une réduction par quatre, du temps de calcul des résultats intermédiaires, l'emplacement mémoire nécessaire pour l'algorithme Max-Log-MAP serait identique à celui du Viterbi.

3.2.4 Algorithme de décodage Log-MAP

L'algorithme Max-Log-MAP donne une légère dégradation de performances par rapport à celle du MAP, en raison de l'approximation de (40), elle résulte en une chute de performances de l'ordre de 0.35dB [ROB95] durant un décodage itérative. Cependant, cette approximation peut être ajustée par l'utilisation de l'algorithme Jacobien.

$$\log(e^{x_1} + e^{x_2}) = \max(x_1, x_2) + \log(1 + e^{-|x_1 - x_2|}) \tag{55}$$

$$= \max(x_1, x_2) + f_c(|\, x_1 + x_2 \,|) \tag{56}$$

$$= g(x_1, x_2) \tag{57}$$

Où la fonction f_c est perçue comme un terme de correction dans cette approximation. De la même manière que l'algorithme Max Log-MAP, $A_k(s) \cong \log(\alpha_k(s))$, $B_k(s) \cong \log(\beta_k(s))$ et $\Gamma_k(s', s) \cong \log(\gamma_k(s', s))$ sont calculés de manière récursive, cependant la maximisation dans (48) et (50) sont complétée par la correction dans (57), ce qui signifie que l'algorithme Log-MAP calcule les valeurs exactes de $A_k(s), B_k(s), \Gamma_k(s`, s)$, et non pas une simple approximation.

Le terme $f_c(\delta)$ n'est pas nécessairement calculé pour chaque valeur de δ, Robertson et al [ROB95] ont prouvé qu'il suffit de stocker huit valeurs de δ, entre 0 et 5, dans un tableau comparatif, en vue d'en déduire toutes les valeurs possibles de $f_c(\delta)$. Par conséquent, il est clair que l'algorithme Log-MAP est légèrement plus complexe que Max-Log-MAP, mais donne exactement les mêmes performances que le MAP.

3.2.5 Algorithme de décodage SOVA

L'algorithme SOVA possède deux modifications par rapport à l'algorithme de Viterbi, ce qui permet son utilisation dans un processus de décodage itératif. La

première modification consiste à introduire l'information à priori (*a-priori information*), dans le calcul des valeurs métriques par chemin (*Path Metrics*), durant la sélection du chemin le plus probable ML dans le treillis.

La deuxième modification consiste à produire une sortie douce (*Soft Output*), de l'algorithme SOVA, à partir du calcul de l'information en a-posteriori, sous forme de rapport de probabilité LLR $L(u_k \mid \mathbf{y})$, pour chaque bit décodé. En considérant $\underline{s}_k^s$, l'ensemble des états constituant un chemin survivant (*Survivor*) dans le treillis, à l'état $S_k = s$ en temps k. La probabilité de véracité du chemin décrit par cet ensemble d'états est :

$$p(\underline{s}_k^s \mid \mathbf{y}_{j\le k}) = \frac{p(\underline{s}_k^s \wedge \mathbf{y}_{j\le k})}{p(\mathbf{y}_{j\le k})} \tag{58}$$

La densité de probabilité $p(\mathbf{y}_{j\le k})$ relative à la séquence reçue $\mathbf{y}_{j\le k}$, des transitions d'états dans le treillis, en incluant la $k$éme, est constante pour tous les chemins possibles $\underline{s}_k$ dans le treillis ; par conséquent, la probabilité que le chemin $\underline{s}_k^s$ est le chemin le plus probable dans le treillis est proportionnelle à $p(\underline{s}_k^s \wedge \mathbf{y}_{j\le k})$, les valeurs métriques dans l'algorithme doivent être définies de manière à ce que leur maximisation revienne à la maximisation de la densité de probabilité $p(\underline{s}_k^s \wedge \mathbf{y}_{j\le k})$. Si le chemin $\underline{s}_k^s$ en temps k a l'ensemble des états $\underline{s}_{k-1}^{s'}$ durant les $k-1$ étapes précédentes, avec la supposition que le canal de transmission

est un canal sans effet de mémorisation sur les données, alors en utilisant la définition (8) de $\gamma_k(s',s)$ on a :

$$p(\underline{s}_k^s \wedge \mathbf{y}_{j\le k}) = p(\underline{s}_{k-1}^{s'} \wedge \mathbf{y}_{j\le k-1}).p(s \wedge \mathbf{y}_k \mid s') \tag{59}$$

$$= p(\underline{s}_{k-1}^{s'} \wedge \mathbf{y}_{j\le k-1}).\gamma_k(s',s) \tag{60}$$

On peut alors définir une valeur métrique $M(\underline{s}_k^s)$ pour le chemin $\underline{s}_k^s$ ou

$$M(\underline{s}_k^s) = \log(p(\underline{s}_k^s \wedge \mathbf{y}_{j\le k})) \tag{61}$$

$$= M(\underline{s}_{k-1}^{s'}) + \log(\gamma_k(s',s)) \tag{62}$$

En utilisant l'équation (26) est en supprimant le terme constant C on a :

$$M(\underline{s}_k^s) = M(\underline{s}_{k-1}^{s'}) + \frac{1}{2}u_k L(u_k) + \frac{L_c}{2}\sum_{l=1}^{n} y_{kl}x_{kl} \tag{63}$$

Ainsi, la mise à jour des valeurs métriques dans l'algorithme SOVA de la même manière, en tenant compte de l'information en priori $u_k L(u_k)$, ce qui équivalent à l'équation (48), dans l'algorithme de Max Log-MAP.

Pour la seconde modification, il est évident que, dans un treillis à valeurs binaires, chaque état $S_k = s$ en temps k, est une intersection de deux chemins différents. L'algorithme SOVA calcule les valeurs métriques, pour les deux chemins possibles, suivant l'équation (63) , en vue de sélectionner celui accumulant la plus

grande valeur. Ces deux chemins possibles $\underline{s}_k^s$ et $\underline{\hat{s}}_k^s$, passant par l'état $S_k = s$, ont les valeurs métriques $M(\underline{s}_k^s)$ et $M(\underline{\hat{s}}_k^s)$, et le chemin $\underline{s}_k^s$ est sélectionné en tant que chemin survivant, du fait qu'il possède la valeur métrique supérieure, ce qui nous amène à définir la différence métrique :

$$\Delta_k^s = M(\underline{s}_k^s) - M(\underline{\hat{s}}_k^s) > 0 \tag{64}$$

La probabilité de véracité de prendre la décision sur le chemin $\underline{s}_k^s$, comme étant le chemin survivant en éliminant $\underline{\hat{s}}_k^s$ est alors :

$$P = \frac{P(\underline{s}_k^s)}{P(\underline{s}_k^s) + P(\underline{\hat{s}}_k^s)} \tag{65}$$

En tenant compte de la définition de la valeur métrique dans l'algorithme SOVA :

$$P = \frac{e^{M(\underline{s}_k^s)}}{e^{M(\underline{s}_k^s)} + e^{M(\underline{\hat{s}}_k^s)}} = \frac{e^{\Delta_k^s}}{1 + e^{\Delta_k^s}} \tag{66}$$

Le rapport de probabilité relative à la véracité de la décision sur l'état $S_k = s$ est tout simplement Δ_k^s. Une fois que le chemin le plus probable ML est identifié, en parcourant tout le long d'un treillis, nous aurons besoin d'évaluer la probabilité de la véracité de cette identification et la fiabilité sur les séquences de bits décodés, en calculant le LLR .

Dans l'algorithme de Viterbi, on peut remarquer que tous les chemins survivants, jusqu'à un certain stade l dans le treillis, sont issues d'un même point à un certain stade avant étape l. Le point l est supposé être de δ transitions avant l'étape l, ou δ est généralement cinq fois la contrainte de longueur du code conventionnel. Par conséquent, la valeur du bit u_k, associé à la transition entre l'état $S_{k-1} = s'$ vers l'état $S_k = s$, du chemin le plus probable sélectionné ML, pouvait être différent, si l'algorithme de Viterbi a choisi un des autres chemins dans le treillis, durant les δ dernières transitions. D'autre part, le choix d'un autre chemin, pour les δ transitions, après l'état $S_k = s$, n'affecte pas la valeur de u_k , vu qu'ils divergent par rapport au chemin le plus probable. Il est donc primordial que la sortie douce, produite par l'algorithme SOVA, prenne en considération la probabilité que les chemins immergeant avec le chemin le plus probable, entre les étapes k et $k+\delta$, ont été incorrectement disqualifiés. Ceci est réalisable par le calcul des différences métriques $\Delta_i^{s_i}$, de tous les états s_i, le long du chemin le plus probable, de l'étape $i = k$ et l'état $i = k+\delta$.

Hagenauer [HAG95] a prouvé que le rapport de probabilité LLR peut être calculé par :

$$L(u_k \mid \mathbf{y}) \approx u_k \min_{\substack{i=k\ldots k+\delta \\ u_k \neq u_k^l}} \Delta_i^{s_i} \tag{67}$$

Où u_k est la valeur du bit donnée par le chemin le plus probable ML, u_k^l est la valeur ce bit pour les autres chemins immergeant avec le chemin retenu, et non retenu

dans l'étape i du treillis. Cette minimisation est calculée uniquement pour les chemins non retenus qui auraient une valeur différente du bit u_k, s'ils étaient retenus comme des chemins survivants. Il est évident que les chemins retenus en tant que chemins survivants, ayant la même valeur du bit u_k, n'affectent pas la fiabilité de la décision sur u_k.

L'algorithme SOVA est le plus simple des algorithmes à sortie douce, Robertson et al [ROB95] ont montré que l'algorithme Max-Log-MAP était deux fois plus complexe que l'algorithme SOVA, bien que ce dernier soit considéré comme le moins précis des algorithmes à sortie douce.

4. TURBO CODE

Ce chapitre est consacré à l'étude des éléments d'une structure de turbo décodage ainsi que l'influence de chaque élément sur l'évolution des performances d'une opération de turbo décodage. Cette étude permettra de compléter notre étude comparative et, par conséquent, mettre en évidence la pertinence des différents choix d'éléments établis dans le standard de la troisième génération de la communication sans fil.

4.1 Structure du Turbo Encodeur

Un Turbo encodeur est représenté en deux dimensions Fig. 4.1. Les données en entrée sont organisées sous forme de Blocs de séquences de longueur N. Le premier Bloc de données sera encodé par l'encodeur convolutionnel récursif, de rapport de transmission (R = 1/2) ENC^{-}. Le même Bloc des bits d'informations est interfolié et encodé par l'encodeur $ENC^{|}$ de même rapport de transmission. La sortie de chaque encodeur constitue le bit encodé. L'interfolieur réarrange l'ordre des bits d'informations pour les introduire au deuxième encodeur.

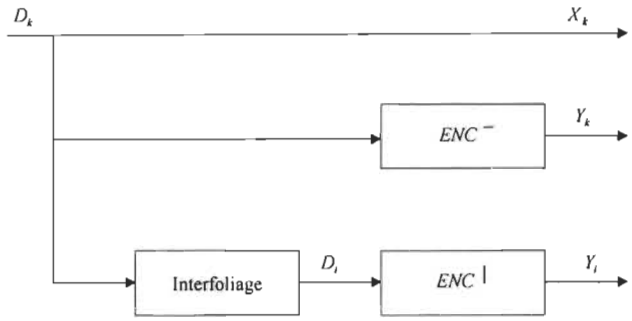


Figure 4.1 Turbo Encodeur (R = 1/3)

En ignorant les délais sur chaque Bloc, on assume une sortie simultanée des deux encodeurs (X_k, Y_K, Y_i), ce triplet d'un turbo encodeur de rapport de transmission (R = 1/3) est ainsi modulé pour être transmis par un canal de communication. Les séquences de bits de parités (Y_k, Y_i) peuvent être perforées (*Punctered*), (éliminer des bits alternativement), de manière à produire un rapport de transmission supérieur (R=1/2) et minimiser ainsi le niveau minimal de bruit par bit (E_b), permettant d'atteindre un même BER, après décodage.

L'objectif principal derrière l'utilisation d'un interfolieur est d'augmenter la valeur de la distance minimale du turbo code, ainsi après chaque correction des données reçues effectuée par le premier encodeur, les erreurs restantes peuvent encore être corrigées de nouveau par le deuxième encodeur. En d'autres termes, l'interfolieur a l'effet d'apparier les mauvaises (de faible poids) séquences de bits de parité, avec les bonnes séquences de bits (de poids fort) dans la plupart des cas, ce qui conduit à la génération d'un minimum de mots de faible poids. Pour des longs Blocs de séquences de bits d'information, des performances assez proches de la limite de Shannon peuvent être atteintes, même si la distance minimale du code généré (d_{free})

pour un turbo code reste relativement faible (d_{free} *est autour de 6*) pour la plupart des méthodes d'interfoliage possible. D'excellentes performances pour des rapports d'erreurs sur chaque bit transmis (BER) modérés, sont dues essentiellement à la réduction draconienne du nombre de mots code du voisinage immédiat, en comparaison avec un codage convolutionnel. Benedetto et Montorsi [BENa96] ont montré que le nombre est réduit par un facteur N, qui représente la longueur des Blocs de séquences en entrée de chaque encodeur, ce facteur est référé comme un « *Un Gain d'Interfoliage* ».

4.2 Structure d'un Turbo Décodeur

À la réception, le processus de décodage itérative est réalisé comme illustré ci-dessous Fig. 4.2, en utilisant deux décodeurs avec une sortie douce pour une seule itération. Le décodeur DEC^{-} produit une sortie douce, qui représente une mesure, ou une information sur la fiabilité de chaque bit décodé. À partir de cette mesure, une information extrinsèque peut être produite, qui ne dépend pas de la séquence d'entrée de l'encodeur.

Après interfoliage, cette information extrinsèque sera introduite dans le décodeur DEC^{I}, et la nouvelle information extrinsèque produite par ce décodeur alimentera l'entrée du premier encodeur DEC^{-} pour une deuxième itération.

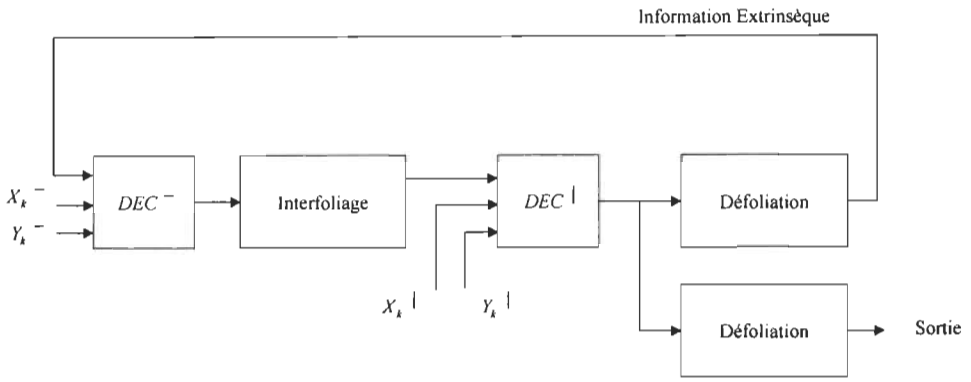


Figure 4.2 Un Turbo Décodeur

L'apparition d'une erreur dans la sortie douce du premier décodeur, due au niveau élevé de bruit en interférence avec des séquences de bits d'information à son entrée, pourrait donc être corrigée par le second décodeur.

4.3 Interfoliage

La structure d'un interfoliage consiste un facteur déterminant, quant aux performances du turbo code. Ainsi les meilleures performances atteintes par un processus de turbo décodage, en comparaison avec un codage convolutionnel, sont rendues possible uniquement par l'utilisation d'un interfoliage de larges séquences de bits d'information. En général, dans le cas des longs Blocs, les interfoliages aléatoires donnent de meilleures performances. D'autre part, pour d'autres applications, il est préférable d'utiliser des méthodes déterministes d'interfoliage, afin de réduire les exigences de mise en œuvre au silicium.

La modélisation d'un interfoliage est basée sur deux critères distincts :

A) La distribution spectrale de distance (*Weight Distribution*) du code généré.

B) La corrélation entre la sortie douce de chaque décodeur issue des bits de parités, et les séquences de bits d'information à son entrée. Ce critère est souvent référé comme un critère approprié au décodage itératif (*Iterative Decodingv System*) [SAD00].

C'est une mesure de l'effectivité d'un algorithme de décodage itérative, le fait que ces deux séquences données soient les moins corrélées, de meilleures performances peuvent alors être atteintes. D'autre part, les performances d'un turbo code pour un rapport d'erreur sur chaque bit décodé (BER) relativement bas, dépendent principalement de la valeur de la distance minimale effective ($d_{\min}$) [BENa96] [DIV96]. Il a été démontré que les performances asymptotiques d'un turbo code approchent l'asymptote relative à la valeur de ($d_{\min}$). Le niveau de bruit qui apparaît, pour des rapports d'énergie par bit transmis sur bruit (SNR) modéré, est le résultat d'un ($d_{\min}$) relativement bas [PER96]. Ce niveau de bruit doit être amélioré soit par la maximisation de la capacité de l'interfoliage ou de la valeur ($d_{\min}$), ce qui donne une importance cruciale au choix d'un interfolieur.

4.3.1 Interfoliage « Ligne-Cologne »

Le plus simple des interfoliages est une mémoire, dans laquelle les données sont écrites ligne par ligne et lues colonne par colonne. Cette méthode d'interfoliage,

est connue comme l'une des méthodes d'interfoliage par bloc (*Block Interleaving*), qui ne permet pas d'atteindre des performances appréciables, pour des rapports d'erreur sur chaque bit décodé (BER) relativement bas.

4.3.2 Interfoliage Pseudo-Aléatoire

Les interfolieurs pseudo-aléatoires sont définis par un générateur de nombre pseudo-aléatoire entre 1 à N (N la longueur du bloc à entrelacer). Cette approche peut conduire à de bons ou mauvais interfolieurs, spécialement pour interfoliage avec des Blocs de séquences relativement petits. Le seul critère de choix entre eux est basé sur les résultats de simulation par ordinateur.

4.3.3 Interfoliage de type S (S – Interleaver)

Il a été démontré en [KHA99] que les séquences de bits d'entrée de poids-2 sont un important facteur du design d'un interfolieur. Le poids d'une séquence binaire d'entrée est de nombre de 1 dans cette séquence. Si l'on choisit arbitrairement un interfolieur de longueur N, la probabilité qu'une séquence particulière de poids-2 soit permutée par l'interfolieur, pour donner une autre séquence de poids-2 de la même forme, est approximativement de l'ordre 2/N, pour N relativement large. Cette probabilité augmente pour des valeurs plus petites de N.

C'est pourquoi, et dans le but d'éviter des permutations identiques, une permutation « S-Random » a été définie comme suit : chaque entier sélectionné arbitrairement est comparé au S entier sélectionné précédemment. Cet entier sera accepté (maintenue) si et seulement s'il est de distance S, de chaque élément des S

entiers sélectionnés précédemment. Le processus est répété pour tous les éléments de Bloc de séquence de longueur N. Pour valeur donnée de N, le maximum facteur S qui doit être utilisé est inférieur à $\sqrt{\frac{N}{2}}$.

4.4 Décodage Itérative (Turbo Décodage)

Le principe du décodage itérative est basé sur l'exploitation d'un décodeur SISO (*Soft Input Soft Output*), comme celui utilisé pour l'algorithme MAP, afin d'améliorer les performances de décodage durant chaque itération. Dans son article [BERa93], Berrou a démontré que pour un codage systématique, tel qu'un code RSC la sortie d'un décodeur MAP pouvait s'écrire :

$$L(u_k / \mathbf{y}) = \log\left(\frac{\sum_{u_k=+1}(s',s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}{\sum_{u_k=-1}(s',s) \Rightarrow \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}\right) \tag{36}$$

$$= L_e(u_k) + L_c y_{ks} + Le(u_k) \tag{37}$$

Avec :

$$L_e(u_k) = \log\left(\frac{\sum_{u_k=+1}(s',s) \Rightarrow \alpha_{k-1}(s').\lambda_k(s',s).\beta_k(s)}{\sum_{u_k=-1}(s',s) \Rightarrow \alpha_{k-1}(s').\lambda_k(s',s).\beta_k(s)}\right) \tag{38}$$

$$\lambda_k(s',s) = \exp\left(\frac{L_c}{2}\sum_{l=2}^{n} y_{kl}x_{yl}\right) \tag{39}$$

$$L_c = \frac{4a}{2\sigma^2} \tag{40}$$

Les (*a-posteriori*) LLR $L(u_k \mid \mathbf{y})$ calculés avec l'algorithme MAP peuvent être vus comme étant la résultante du comportement des trois termes, $L(u_k), L_c(y_{kl}), L_e(u_k)$. $L_e(u_k)$ est l'information priori (*a-priori information*), récoltée de la valeur de $P(u_k)$ dans l'équation de calcul de la probabilité de transition par branche $\gamma_k(s',s)$, cette probabilité doit être issue d'une source indépendante, par rapport aux données d'entrée du décodeur. Dans la plupart des cas, on a aucune information priori sur le bit d'entrée u_k, du coup, $L(u_k)$ sera égal à zéro correspondant à $P(u_k) = 0.5$. Cependant, dans le cas d'un décodage itérative, chaque décodeur peut alimenter un autre décodeur avec une estimation sur le bit en entrée u_k selon la valeur de $L(u_k)$. Le terme $L_c y_{ks}$ représente la sortie douce du canal pour une entrée systématique u_k, directement transmise à travers le canal et reçue en tant que y_{ks}. Si le rapport signal sur bruit (SNR) est élevé, l'indice de fidélité du canal L_c donne au bit d'entrée systématique une large influence, sur la valeur de $L(u_k \mid \mathbf{y})$, et, par conséquent, sur la décision sur u_k. D'autre part, si la valeur du (SNR) reste faible, (un bruit élevé), l'impact de la sortie systématique du canal y_{ks} sera minime sur $L(u_k \mid \mathbf{y})$ produit par l'algorithme MAP.

Finalement la valeur $L_e(u_k)$ traduit le poids des contraintes imposées par l'encodeur, en utilisant l'information a-priori $L(u_n)$, et la séquence reçue $\mathbf{y}$ en excluant la valeur y_{ks} et l'information préliminaire $L(u_k)$ sur le bit d entrée u_k, ce qui justifie son appellation (information extrinsèque), sur le bit d'entrée u_k, l'équation (5) montre que cette information extrinsèque peut être extraite à partir d'un MAP décodeur par la substitution de la valeur de l'information en priori $L(u_k)$ et la valeur $L_c y_{ks}$ reçue du canal à partir de l'entrée systématique de l'encodeur, de la sortie douce (Soft Output) $L(u_k \mid \mathbf{y})$ du décodeur.

En vue de décrire tout le processus d'un décodage itérative, à base d'un algorithme MAP, on considère le premier encodeur durant la première itération, cet encodeur reçoit du canal la séquence $L_c y$, contenant la version transmise du bit systématique $L_c y_{ks}$, et les bits de parité $L_c y_{kl}$ du premier encodeur, ce décodeur peut ainsi traiter cette entrée douce (Soft Input), pour produire une estimation $L(u_k \mid \mathbf{y})$, sur les bits d'entrées $u_k, k = 1,2,3.....,N$. Il faut noter que durant la première itération, le premier encodeur n'a aucune information en priori. Par conséquent, on doit considérer que l'information binaire transmise est équiprobable ($P(u_k)$. Le second décodeur commence ainsi le traitement, une fois il reçoit la séquence $\mathbf{y}$, contenant la version entrelacée, des entrées systématiques et bits de parité, du premier encodeur. Le second encodeur peut, en outre, utiliser le rapport de probabilité

conditionnelle $L(u_k \mid \mathbf{y})$, produit par le premier encodeur, comme une source d'information en priori, sur tous les bits u_k à décoder.

Cette information est vue comme une autre '*opinion indépendante*' sur chaque bit u_k, afin de minimiser le plus possible, les erreurs sur le décodage. Durant un décodage itératif (Turbo Code), l'information extrinsèque $L_e(u_k)$ issue d'un autre décodeur est utilisée, après entrelacement, en vue d'avoir le même ordre des données en entrée du deuxième encodeur, comme un rapport de probabilité (LLR), de l'information en priori. Toutes ces informations réunies $(L(u_k), L_e(u_k), \mathbf{y})$, permettent au second encodeur de produire à son tour une nouvelle estimation $L(u_k \mid \mathbf{y})$, ce qui constitue la dernière étape de la première itération. Durant la second itération, le premier encodeur procédera encore une fois au traitement de la séquence $\mathbf{y}_1$, mais maintenant avec de l'information en priori $L(u_k)$ produite par l'information extrinsèque $L_e(u_k)$,de l'information a-posteriori $L(u_k \mid \mathbf{y})$ calculée par le deuxième encodeur durant la première itération, ce qui permet de produire une valeur améliorée de $L(u_k \mid \mathbf{y})$, par le premier encodeur par rapport à la première itération et, par conséquent, une version améliorée de l'information en priori $L(u_k)$, utilisée avec la séquence reçue $\mathbf{y}_2$, par le second encodeur, pour le calcul de $L(u_k \mid \mathbf{y})$.

Le processus de décodage itératif continue à faire baisser la valeur du BER, avec chaque itération. Généralement, [SKL97] pour des raisons de complexité de mise en oeuvre, uniquement huit à neuf itérations sont utilisées durant le décodage

itérative, étant donné que cette diminution du BER est de plus en plus réduite, au fur et à mesure que le nombre d'itérations augmente. Le tableau 4.1 présente le résumé d'une étude comparative, entre le codage convolutionnel et le turbo code.

Tableau 4.1 Comparaison entre Code convolutionnel et Turbo Code

Paramètre	**Code Convolutionnel**	**Turbo code**
Large contrainte de longueur	Bon	Mauvais
Large distance libre	Bon	Indifférent
Rapport de codage bas	Indifférent	Bon
Encodeurs récursifs	Indifférent	Bon
Décodeurs à sortie douce	Indifférent	Bon

5. RÉSULTATS DE SIMULATION

Dans ce chapitre, l'étude portera sur les résultats obtenus par simulation, en vue de réaliser une étude comparative relative aux différents algorithmes de décodage. Les simulations ont été réalisées sur (Matlab), conformément au schéma de principe d'un système de communication numérique Fig. 1.1, en assumant une modulation BPSK de la donnée utile, et une réception parfaite des données transmises à travers un canal sans effet de mémoire. Sauf indication contraire, le canal de transmission est un bruit blanc Gaussien Additionnel AWGN, caractérisé par la superposition d'un bruit de distribution Gaussienne, à l'information transmise. Le standard 3GPP [3GPP01] représente le cadre général des simulations, pour le code convolutionnel (Fig. 5.1) (Fig. 5.2) et le turbo code (Fig. 5.3).

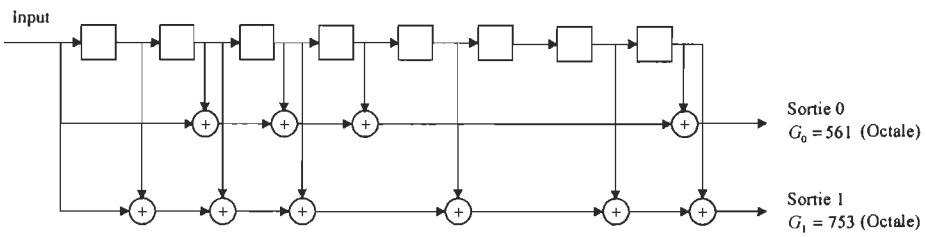


Figure 5.1 Encodeur Convolutionnel 3GPP (R = 1/2)

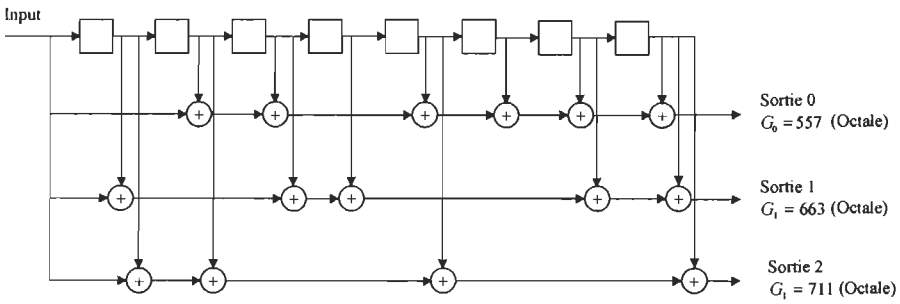


Figure 5.2 Encodeur Convolutionnel 3GPP (R = 1/3)

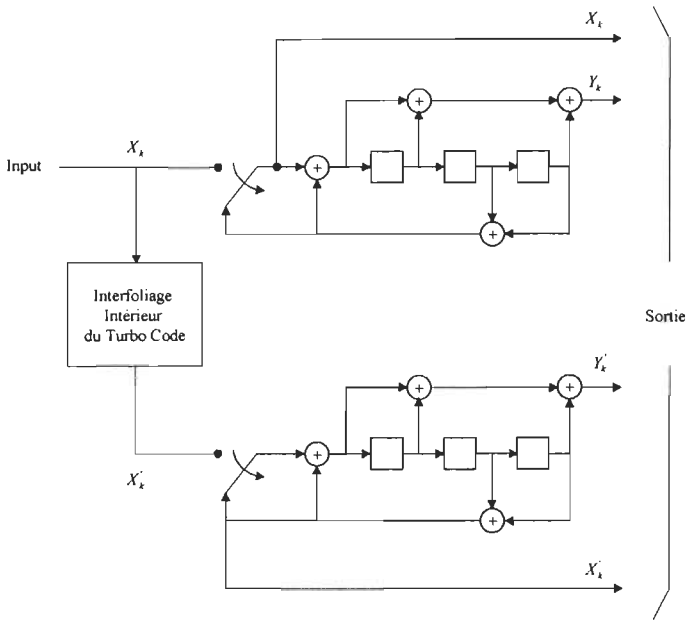


Figure 5.3 Tubo Encodeur 3GPP

Dans le but de bien compléter l'étude comparative, des simulations additionnelles ont été effectuées en vue de bien élucider tous les paramètres d'influence intervenant dans la structure du codage (Ex : Contrainte de longueur, entrelacement, ...), le service (Perforation), ou le choix de l'algorithme de décodage (*Ex : MAP, Log-MAP,...*). Le codage convolutionnel parallèle et enchaîné (*Parallel Concatrenated Convolutionnel Code*), constitué de deux encodeurs convolutionnels récursifs et identiques, avec un interfolieur intercalé entre leurs entrées respectives, est le modèle de simulation adopté pour le turbo codage. Le résultat d'un processus de turbo décodage sera donc une décision dure (*Hard Decision*), effectuée après chaque itération sur la sortie douce du second décodeur.

5.1 Codage Convolutionnel

La Figure (5.4) montre les performances du décodage avec l'algorithme de Viterbi selon le standard 3GPP. Le gain en performance apporté par l'application de l'algorithme de Viterbi (3,1,9), en comparaison avec le Viterbi (2,1,9), est de l'ordre de 2.1dB, pour une valeur du ($BER=10^{-3}$), (Valeur de référence en matière de transmission de la voix). Ce gain en performance permet une réduction de près de 40%, la valeur de la puissance minimale pour chaque bit transmis. Dans les deux cas, atteindre une valeur du ($BER=10^{-5}$) nécessite un rapport d'énergie par bit transmis, sur puissance de bruit supérieure à 6dB, ce qui traduit un niveau de communication peu fiable, pour des niveaux de bruit très élevés.

5.2 Turbo Code

5.2.1 Algorithmes de Turbo Décodage

La Figure 5.8 permet de voir une comparaison de performances entre un turbo décodeur utilisant l'algorithme MAP et les algorithmes de décodage produits par l'approximation du MAP, dans le domaine logarithmique. Pour un rapport d'erreur de décodage ($BER=10^{-5}$), la dégradation de performance observée, pour l'approximation de l'algorithme MAP apportée par le Log-MAP, est au voisinage de

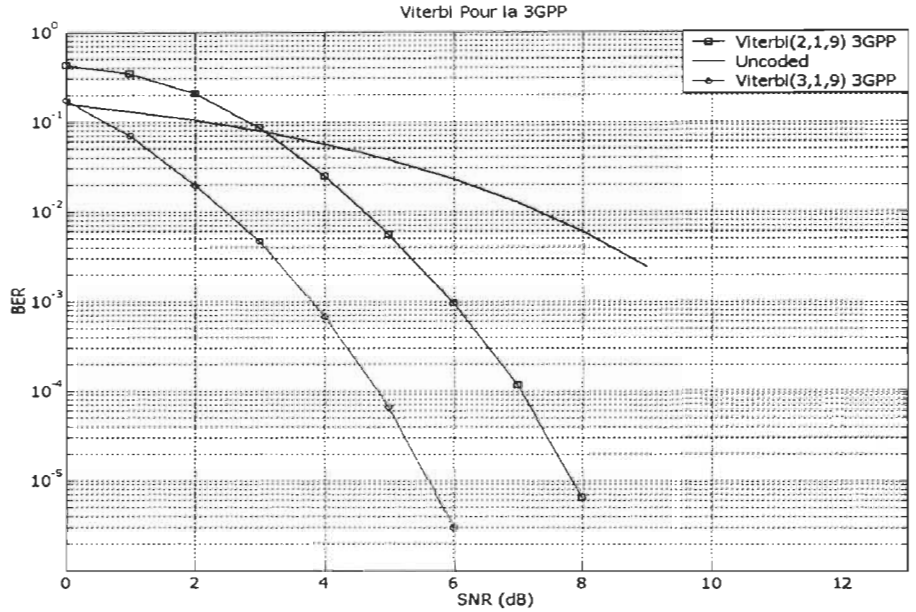


Figure 5.4 Algorithme de Viterbi 3GPP

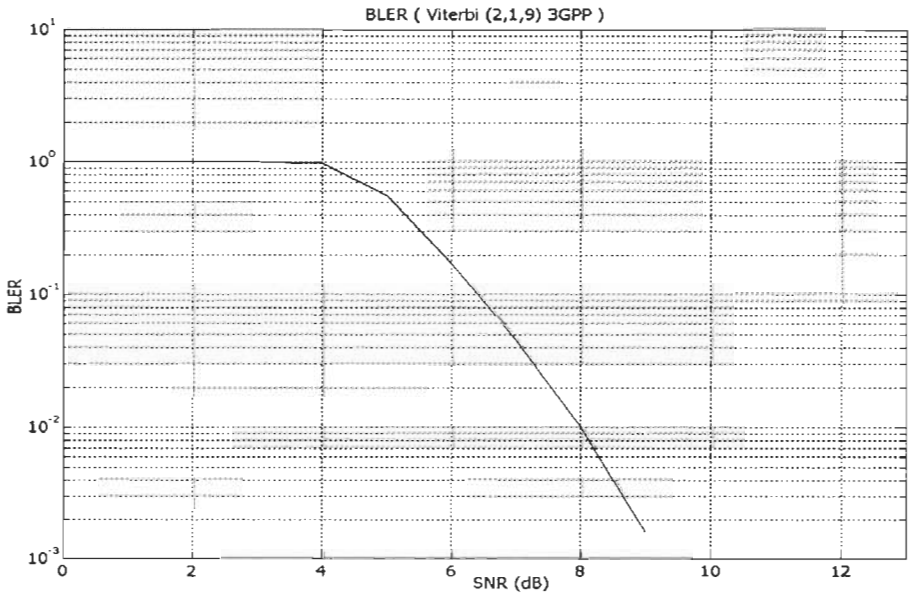


Figure 5.5 (BLER) (Décodage Convolutionnel 3GPP)

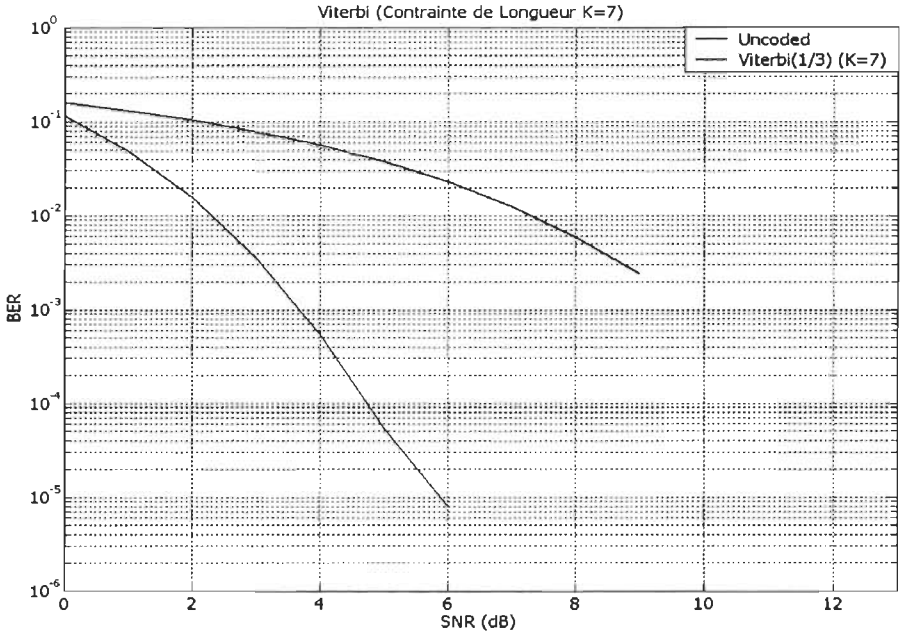


Figure 5.6 Encodeur Convolutionnel K=7

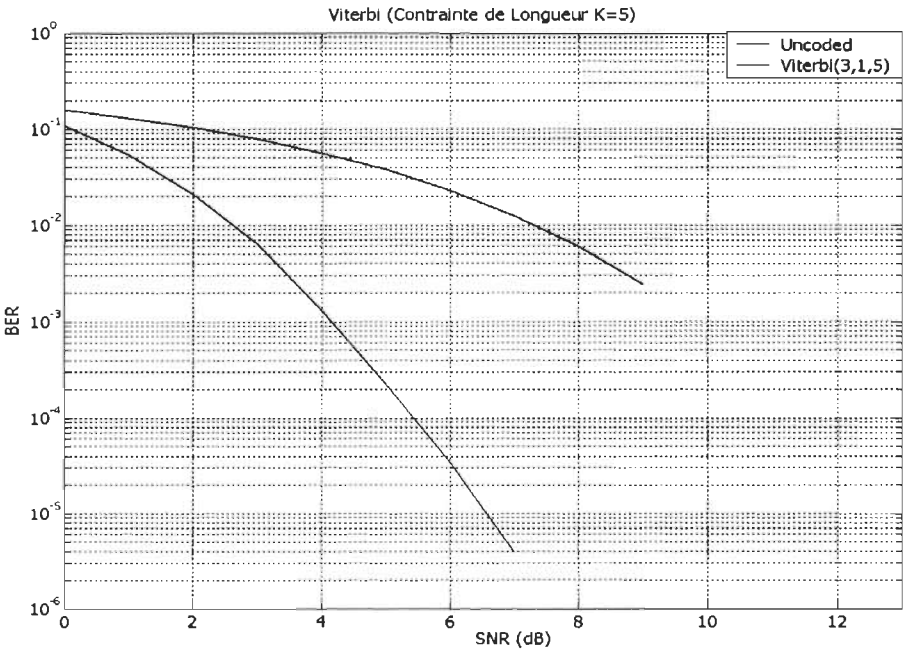


Figure 5.7 Encodeur Convolutionnel K=5

0.1dB. Par contre l'approximation du MAP par l'algorithme Max-Log-MAP, produit une dégradation des performances, pour le même niveau d'erreur ($BER = 10^{-5}$), de l'ordre de 0.5dB, une perte d'énergie par bit transmis, pour un même niveau de bruit au voisinage de 3dB. D'autre part, la Figure 5.9 montre les performances du turbo SOVA, appliqué sur le même canal, qui démontre déjà un taux d'erreur sur le décodage ($BER = 5.10^{-4}$), pour un rapport d'énergie par bit transmis sur énergie du bruit ($Eb / No = 1.8dB$). Cette performance montre l'importance du choix de l'algorithme, dans le processus de décodage, en tenant compte de sa complexité du calcul et de sa flexibilité durant une mise en œuvre sur silicium. Il est important de rappeler que les simulations, pour les différents algorithmes de décodage, ont été réalisés sur le même canal 64kbps, sans perforation, selon le standard 3GPP [3GPP01], avec un nombre de Bloc réduit pour le turbo SOVA (50 Blocs).

D'autre part, la Figure 5.10 montre la variation de la moyenne, de la sortie douce du second décodeur du turbo Max-Log-MAP, selon une multitude de niveaux de bruit. Cette valeur moyenne du LLR représente un indice de confiance sur le résultat de décodage durant chaque itération.

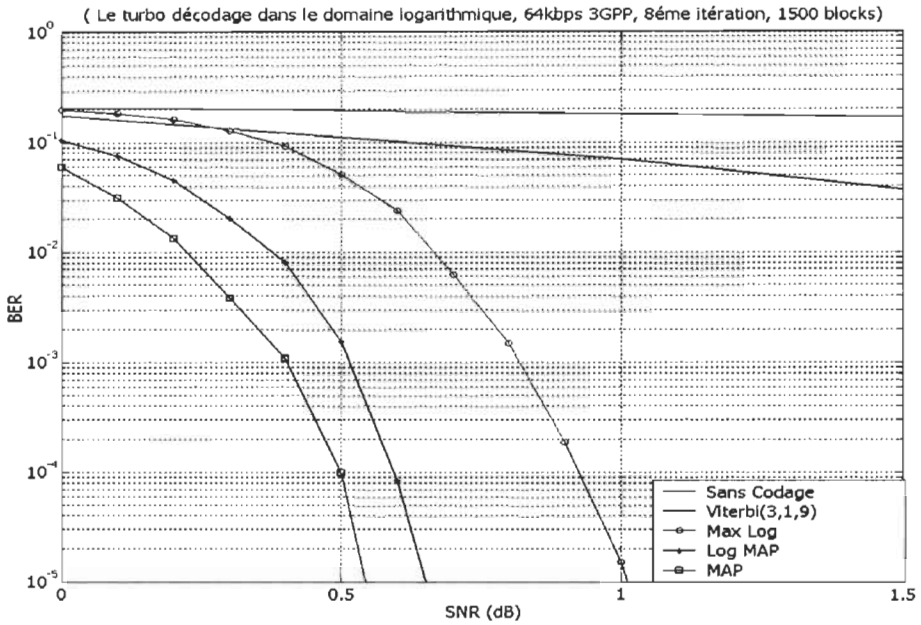


Figure 5.8 Le Turbo Décodage dans le domaine Logarithmique (Standard 3GPP)

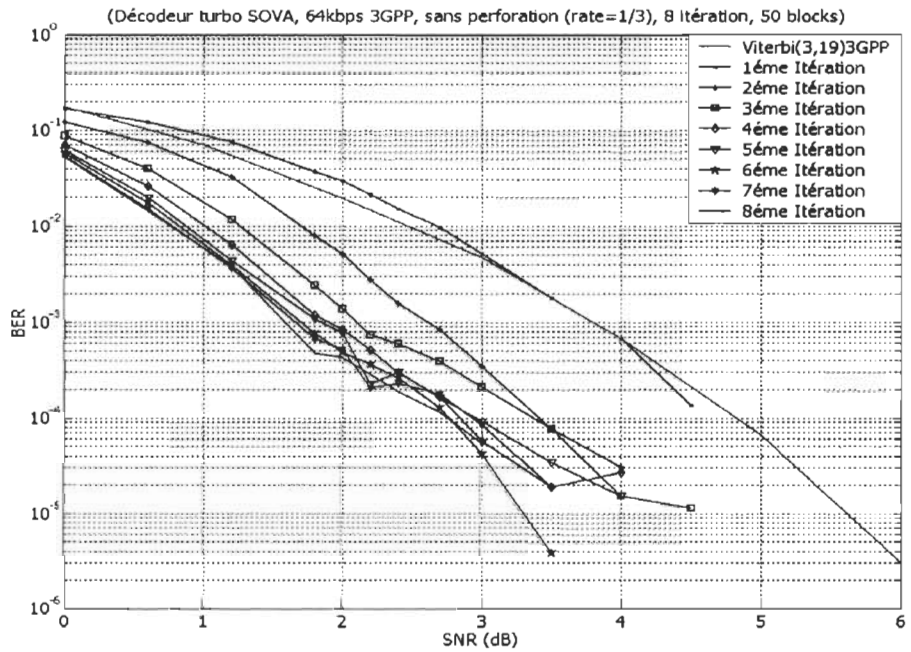


Figure 5.9 Le Turbo SOVA (Standard 3GPP)

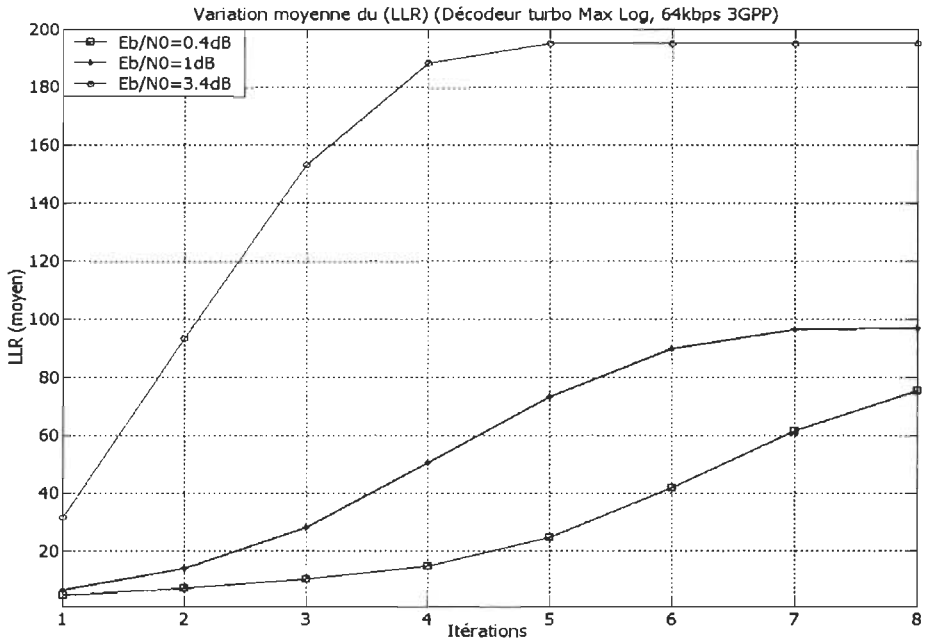


Figure 5.10 Variation de la moyenne du LLR (Max-Log-MAP Standard 3GPP)

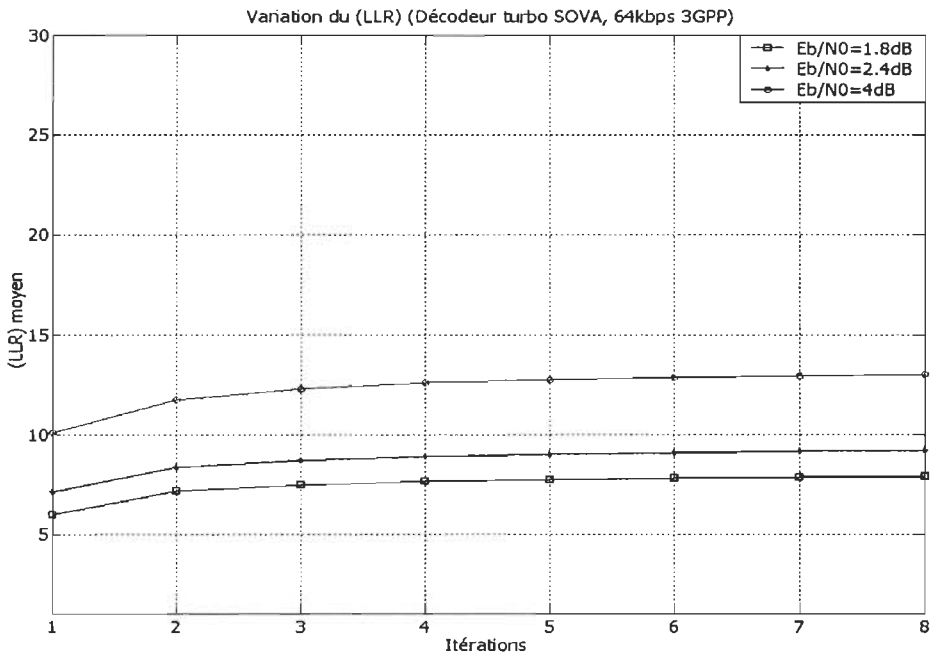


Figure 5.11 Variation de la moyenne du LLR (SOVA Standard 3GPP)

Pour des niveaux de bruit relativement élevés ($SNR = 0.4dB$), le turbo Max-Log-MAP n'a toujours pas atteint le niveau de saturation à la huitième itération, ce qui justifie le recours à un supplément d'itérations dans ce cas, pour le turbo Max-Log-MAP. Par contre, ce niveau de saturation peut être atteint, dés la $5^{iéme}$ itération pour un niveau de bruit relativement bas ($SNR = 3.4dB$), pour le même algorithme.

L'indice de confiance du turbo SOVA Fig. 5.11 a atteint la saturation avec des valeurs relativement faibles, dès la deuxième itération. Cette constatation est sensiblement la même pour les différents niveaux de bruit, traduisant ainsi les limites de l'utilisation de cet algorithme, dans un processus de turbo décodage. La variation moyenne de la sortie douce du second décodeur représente ainsi un outil supplémentaire de comparaison entre les différents algorithmes de décodage qui permet de justifier le choix d'un algorithme de turbo décodage.

5.2.2 Nombres d'itérations

La Figure 5.12 montre les performances d'un turbo décodage, avec l'algorithme Max-Log-MAP, pour le canal 64kbps du standard 3GPP, en fonction du nombre d'itérations employées. Le (BER) des données non codées ainsi que le codage convolutionnel (3,1,9) appliqué au canal 12.2kbps du même standard sont introduits pour la comparaison. Les performances du turbo décodage après une itération sont sensiblement similaires au codage convolutionnel pour des faibles valeurs du (SNR), mais augmentent plus rapidement pour des valeurs du (SNR) plus élevées. L'augmentation du nombre d'itérations permet au turbo décodeur d'atteindre

de meilleures performances. Cependant, après six itérations, on observe une légère amélioration de cette performance pour chaque itération supplémentaire.

La Figure 5.12 montre clairement un gain de 0.06dB, pour un rapport d'erreur de décodage (BER=10^{-5}), en utilisant huit itérations au lieu de sept. Les mêmes résultats sont observés pour l'utilisation de l'algorithme de SOVA, mais pour des raisons de complexité, huit itérations seront généralement utilisées dans la pratique.

5.2.3 Longueur de Trame

Des résultats impressionnants ont été présentés dans plusieurs publications pour l'application d'un processus de turbo décodage sur des longues trames de séquences de bits en entrée de l'encodeur. Cependant, pour plusieurs modèles d'application (Ex : la transmission de la voix), les longs délais causés par l'application de ces longues trames en entrée sont pratiquement inacceptables. Par conséquent, un grand effort de recherche a été entrepris pour permettre d'approcher les résultats déjà réalisés avec les longues trames. La Figure 5.13 démontre la grande influence de la longueur de trame sur les performances d'un turbo code.

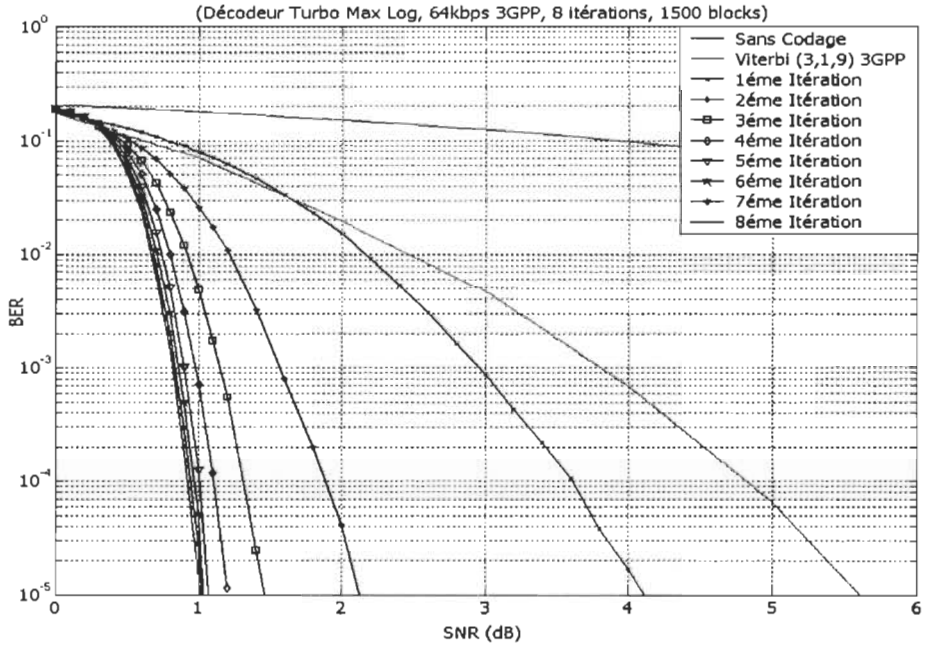


Figure 5.12 Turbo Max Log-MAP 3GPP

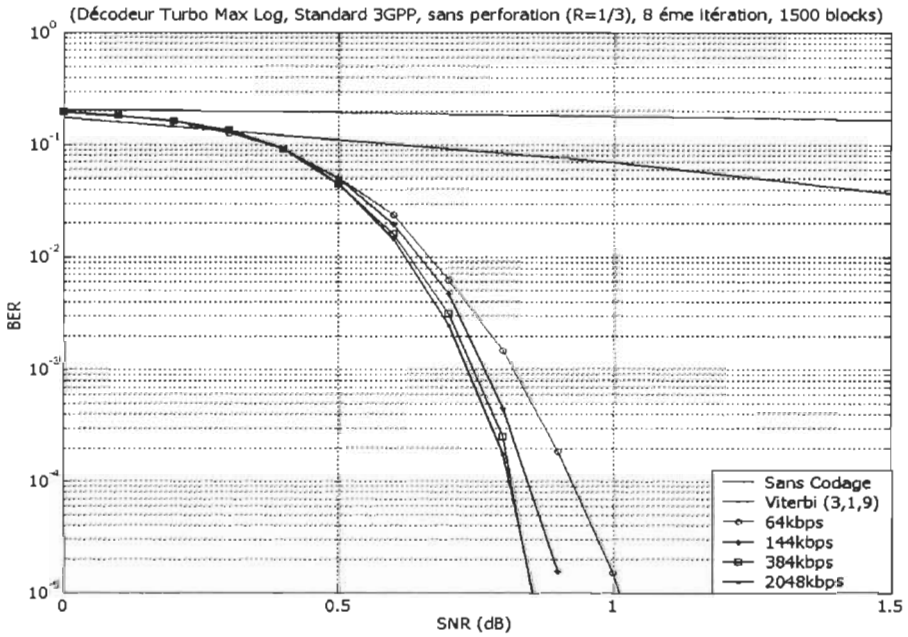


Figure 5.13 Variation de trame en Turbo Code (3GPP)

L'application du Max-Log-MAP sur un canal de 64kbps du standard 3GPP inflige une dégradation des performances de l'ordre de 0.155dB pour l'utilisation du canal 2084 kbps dans les même conditions. Cette dégradation est justifiée par une meilleure décorrélation par interfoliage, réalisée par l'interfolieur du même standard, entre la sortie douce de chaque décodeur correspondant à ces bits de parité et la séquence d'information dans l'entrée du même décodeur. Cette dégradation est réduite à 0.095dB et 0.085dB, pour l'utilisation du canal 384 kbps et 144kbps respectivement, par rapport à un canal de 64kbps du standard de 3GPP.

La comparaison avec le codage convolutionnel (3,1,9) du standard 3GPP est justifiée par le fait qu'il représente le même ordre de complexité qu'un processus de turbo décodage avec 8 itérations dans le domaine logarithmique [WOO00], en ignorant le fait que seules les informations aprioriques (*A-priori Information*) qui changent après chaque itération. Il est donc important de noter que la complexité d'un processus de turbo décodage de 8 itérations peut être réduite par le stockage des informations utilisées durant la première itération, en vue de leur réutilisation pour les itérations futures. La Figure 5.13 montre les performances supérieures d'un turbo code par rapport au codage convolutionnel (3,1,9), pour les différentes longueurs de trame.

5.2.4 Contraintre de Longueur

La contrainte de longueur est un important paramètre dans le choix d'un encodeur convolutionnel récursif et systématique (RSC) d'un turbo code. Souvent c'est le polynôme générateur produisant le maximum de distance libre (*Free*

Distance) qui est choisi pour le turbo code, malgré la présence de l'interfoliage qui remet en cause cette évidence. La Figure 5.14 montre le résultat de la huitième itération de simulation d'un turbo code, en utilisant l'algorithme Max-Log-MAP, pour l'encodeur convolutionnel récursif ($G_0 = 7, G_1 = 5$ en notation octale), avec une contrainte de longueur $K = 3$. Il est évident que l'augmentation de la mémoire interne de l'encodeur convolutionnel récursif et, par conséquent, la contrainte de longueur, permet théoriquement un gain en performance par l'augmentation de la quantité d'information autour de chaque bit u_k transmis relatif aux bits de parité produits par les contraintes de l'encodeur. Cette quantité d'information sera capitale durant le processus de turbo décodage, étant donné qu'elle détermine la valeur l'information extrinsèque relative à u_k utilisée pour l'amélioration des performances après chaque itération. Néaumoins, la complexité de l'implémantation d'un turbo décodeur croit exponentielement avec la valeur de la contrainte de longueur K, par conséquent, pour des raisons pratiques, l'utilisation d'un turbo décodeur avec une contriante de longueur relativement elevée, n'est nullement recommandé.

On observe, dans notre cas, un gain de près de 0.03dB, entre l'utilisation de l'encodeur convolutionnel et récursif du standard 3GPP et celui avec une contrainte de longueur $K = 3$. Il est cependant important de noter que le choix de la valeur octale assurant la régénération (*Feedback*), est aussi important pour un encodeur convolutionnel récursif, afin de maximiser la distance minimale effective (*Effective Maximum free Distance*) du turbo code [ROB94].

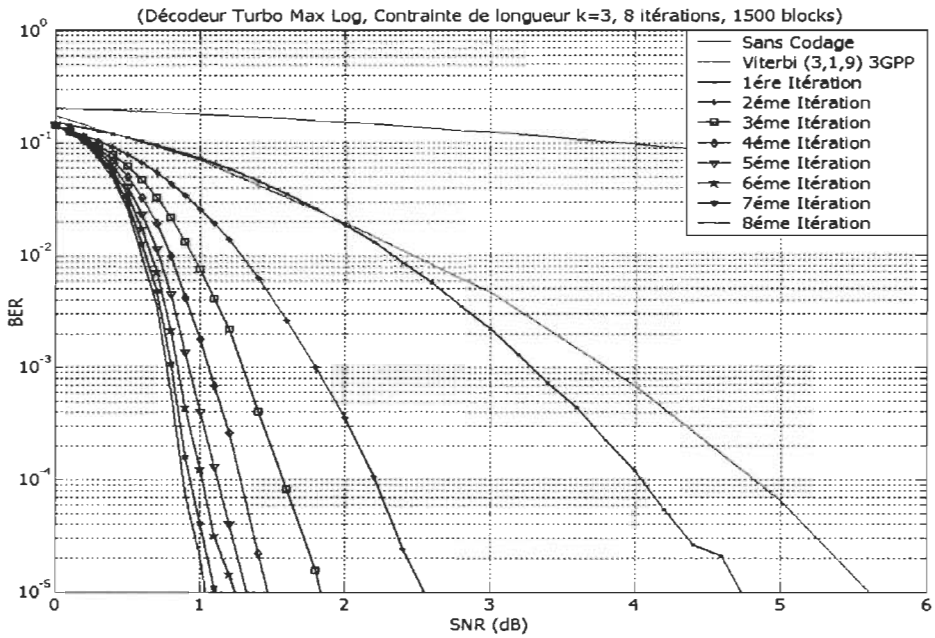


Figure 5.14 Turbo Max Log-MAP (K=3)

Le gain en performance produit par l'augmentation de la contrainte de longueur de l'encodeur convolutionnel récursif ne peut être réalisé au détriment de la complexité du turbo décodage, qui peut alors doubler ou même quadrupler. Un compromis performance complexité de calcul dans le choix de la contrainte de longueur est donc en vigueur.

5.2.5 Sensibilité au bruit

Selon l'algorithme choisi dans un processus de turbo décodage, l'évaluation de l'influence d'une éventuelle estimation du bruit à la réception, sur les performances du turbo codage, est un des éléments clé dans le processus d'implémantation du turbo code. La Figure 5.15 montre la sensibilité des performances de la huitième itération du turbo décodage avec l'algorithme MAP par

rapport à une erreur d'estimation du bruit à la réception (*Offset*). Comme prévu, on remarque l'influence du niveau de bruit sur cette sensibilité qui présente une erreur de près de 10^{-1} sur le (BER), correspondant à une erreur d'estimation du bruit (*Offset*) de 1dB, pour un ($E_b / N_0 = 0dB$), jusqu'à atteindre la valeur du 10^{-4} d'erreur sur le (BER), correspondant à 0.5dB d'erreur d'estimation du bruit, pour un ($E_b / N_0 = 0.5dB$). Cette variation est justifiée par le fait que l'équation du calcul des métriques des branches, élément de base pour le calcul de la sortie douce de l'algorithme MAP, est une fonction non linéaire du bruit (35). L'approximation apportée par l'algorithme Jacobien dans le Log-MAP produit un terme de correction qui est aussi une fonction non linéaire du bruit. La sensibilité du Log-MAP au bruit est donc maintenue, étant donné qu'une estimation du bruit reste nécessaire pour cette algorithme, à la fois pour le calcul des métriques de branches et la pondération des informations constituant la sortie douce de chaque encodeur [WOR00]. La Figure 5.16 montre la non-sensibilité au bruit, de l'algorithme Max-Log-MAP, quel que soit le niveau de bruit. Cette conclusion est une des conséquences de l'approximation de l'algorithme MAP qui permet le calcul des métriques des branches par une fonction linaire du bruit (52).

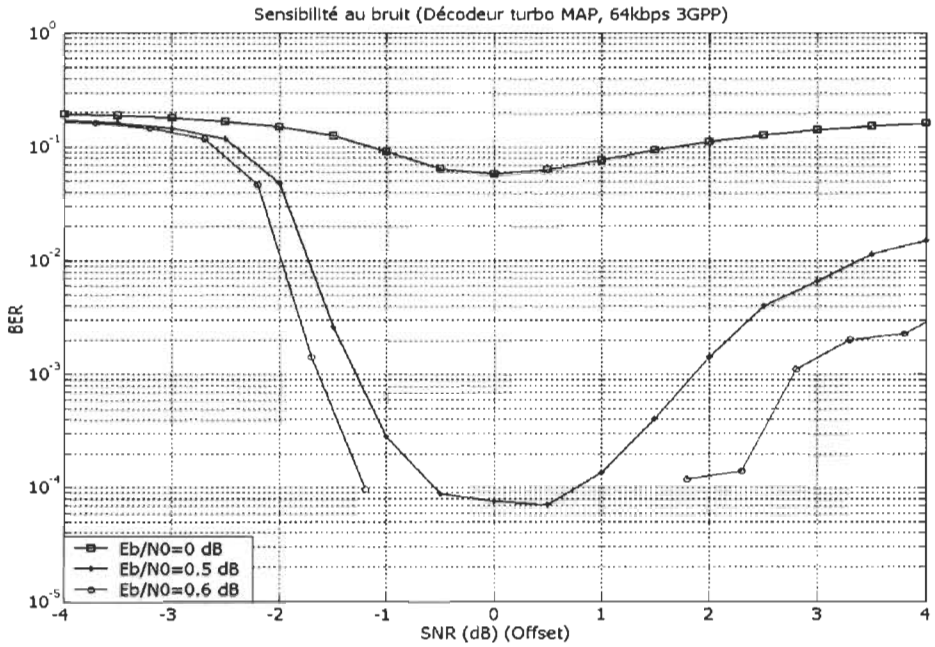


Figure 5.15 Sensibilité au Bruit (Turbo MAP 3GPP)

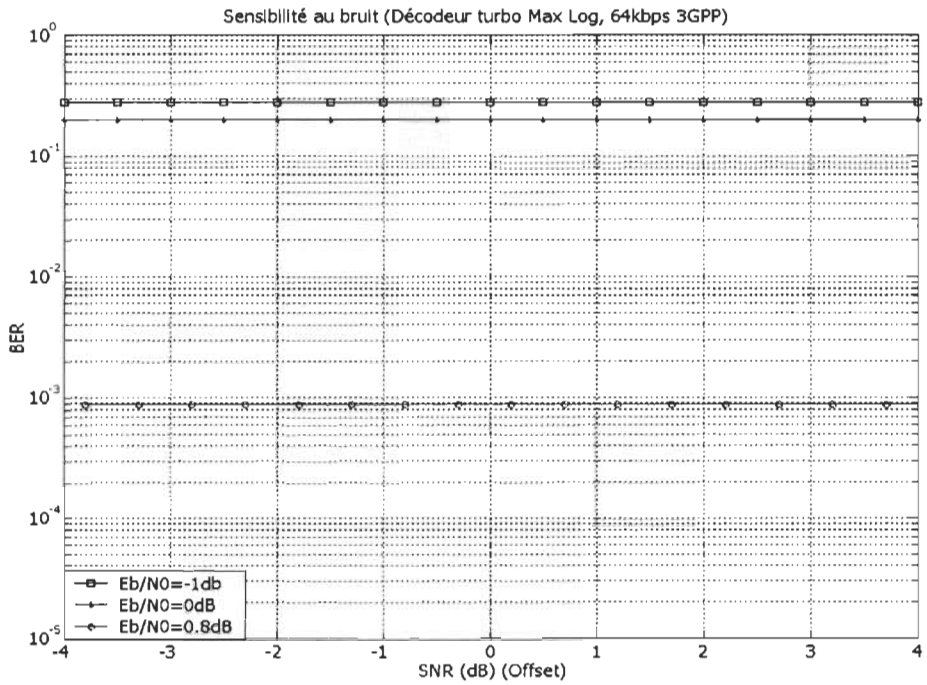


Figure 5.16 Sensibilité au Bruit (Turbo Max Log-MAP)

5.2.6 Interfoliage

Il est évident de constater l'importance de l'interfoliage sur les performances d'un turbo décodeur. La conception de l'interfolieur et le polynôme générateur utilisés dans la structure d'encodage, ainsi que la perforation utilisée au niveau du décodage a un effet crucial sur la distance minimale du turbo code. Plusieurs propositions ont été présentées pour définir les critères de conception d'un « bon » interleaver [SAD00] basé sur la maximisation de la distance libre minimale (*Maximum Free Distance*), mais le processus reste très complexe et les interfolieurs résultants ne sont pas nécessairement optimaux. La Figure 5.17 présente le résultat de simulation d'un décodage par l'algorithme Max-Log-MAP, avec un interfoliage (Ligne-Colonne). En outre, la dégradation des performances par rapport à l'interfoliage pseudo aléatoire du standard 3GPP, on remarque l'influence de la faible distance libre minimale ($d_{\min}$), causant une saturation à partir de la valeur (BER=10^{-4}). Cette saturation traduit une grande corrélation entre la sortie douce de chaque décodeur relative à ces bits de parité et la séquence des bits d'information à son entrée. La Figure 5.18 montre le résultat de la même simulation avec un interfoliage (S-aléatoire), (Pseudo Aléatoire) et l'interfoliage aléatoire, avec un gain de performances de 0.07dB, par rapport à l'entrelacement pseudo aléatoire du standard 3GPP, pour une valeur de (BER=10^{-4}). Par conséquent, un faible rapport

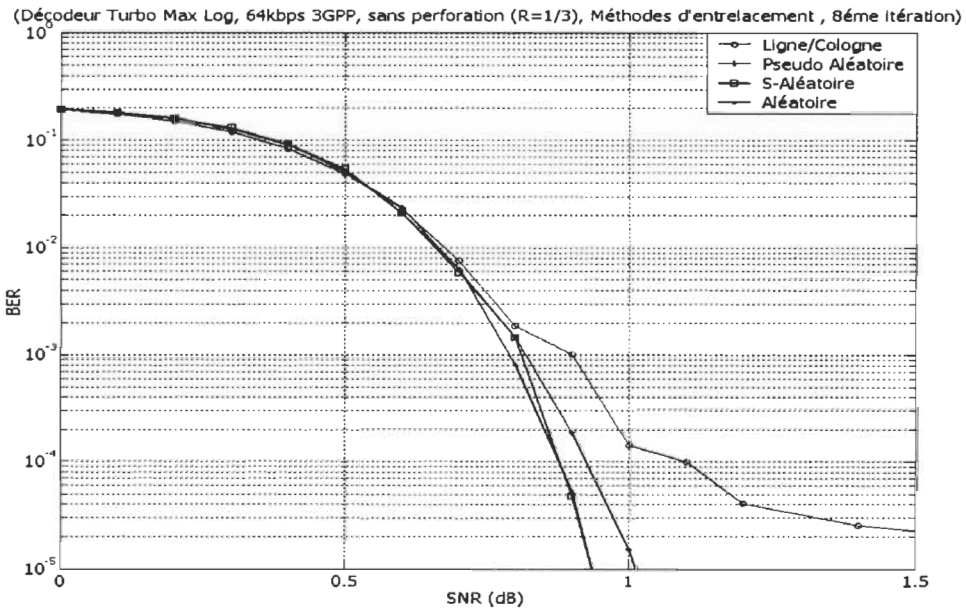


Figure 5.17 Interfoliage en turbo Décodage 3GPP

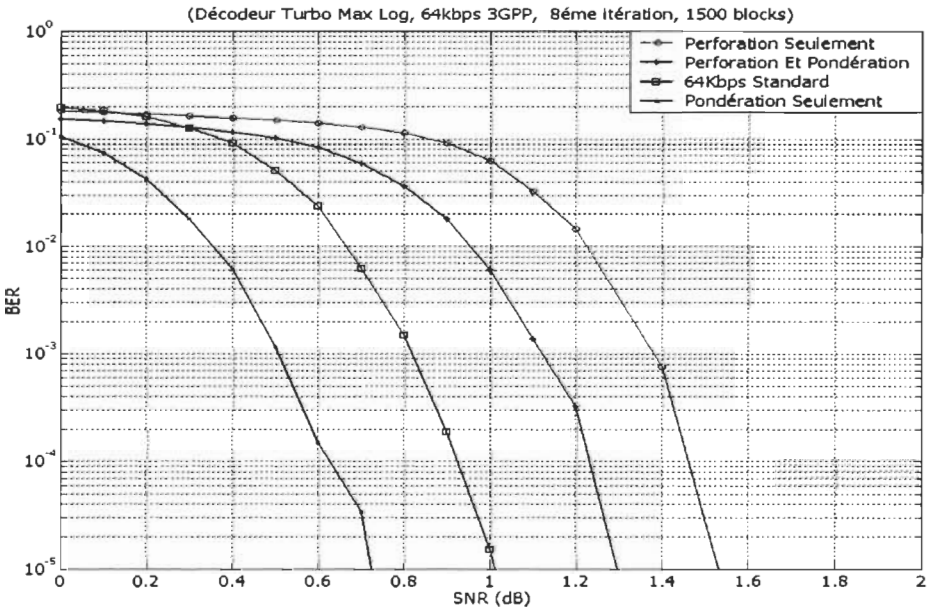


Figure 5.18 Pondération/Perforation en Turbo Décodage 3GPP

d'erreur par bit décodé (BER), pour un rapport d'énergie par bit transmis sur niveau de bruit (E_b / N_0) faible est réalisable uniquement avec un interfoliage aléatoire.

5.2.7 Perforation

En turbo décodage, deux ou plusieurs encodeurs peuvent être utilisés pour la génération des bits de parité (*Parity Bits Informations*), à partir des bits de données en entrée. Selon le standard 3GPP, deux encodeurs convolutionnels récursif sont seulement utilisés dans un processus de turbo décodage, pour un rapport de codage (R = 1/3). Dans le but de produire un rapport de codage (R = 1/2), la moitié des bits de parité de chaque encodeur seront perforés, selon l'arrangement utilisé par Berrou dans son article original [BERa93]. Toutefois, il est possible d'omettre cette perforation et transmettre tous les bits de parité de chaque encodeur. La Figure 5.18 représente le résultat de simulation d'un turbo décodeur appliquant l'algorithme Max-Log-MAP, sur le canal 64kbps du standard 3GPP, sans et avec perforation. La transmission de tous les bits de parité résulte en un gain de presque 0.5dB pour (BER), ce qui représente un gain de puissance en transmission canal, au voisinage de 3dB.

5.2.8 Amélioration des performances

Une surestimation a été observée de l'information extrinsèque produite par chaque décodeur, à partir des bits de parités introduits à son entrée. Une pondération adéquate de cette information, avant sa transmission à un autre décodeur, permet d'améliorer considérablement les performances du turbo décodage, pour des

algorithmes de décodage insensibles au bruit [VOG00]. La Figure 5.18 montre un gain de performance de près de 0.7 dB, par rapport au standard 3GPP (Sans Pondération), en utilisant un coefficient de pondération de l'information extrinsèque du Max-Log-MAP ($\alpha = 0.7$), ce qui permet un gain en puissance de près de 5dB.

D'autre part, la Figure 5.18 montre aussi l'amélioration de la sortie douce du décodeur apporté par cette pondération, en comparaison avec celle relative au standard 3GPP sans pondération. On observe ainsi une saturation dès la cinquième itération, y compris pour les niveaux de bruit relativement élevés, traduisant la capacité de l'algorithme à atteindre les valeurs de sortie maximales, avec un nombre minimal d'itérations. La perforation des bits de parités à la réception, avec les mêmes conditions de simulation (Algorithme Max Log-MAP avec Pondération), montre aussi un gain de performances à la huitième itération de près de 0.2dB, apporté par le coefficient de pondération ($\alpha = 0.7$) Fig. 1.16, et réduit ainsi les pertes en performances causées par la perforation à 0.3dB seulement.

5.2.9 Canal d'évanouissement (Rayleigh)

Dans la section précédente, on a discuté des performances du turbo codage avec des séquences de bits transmises en modulation BPSK, à travers un canal AWGN. Dans cette section, ces mêmes séquences seront transmises à travers un canal d'évanouissement (Rayleigh), en assumant que le récepteur possède une estimation exacte des amplitudes et phase d'évanouissement infligé par le canal. Cette supposition est justifiée par plusieurs techniques, (Ex. modulation assistée par

des symboles pilotes PSAM) [WOO00], qui donnent des performances proches de celles obtenues avec une connaissance parfaite du canal.

La Figure 5.19 montre la dégradation de performances, pour un canal d'évanouissement (*Rayleigh Fading Channel*) parfaitement interfolié, en assumant une parfaite connaissance à la réception des amplitudes et phases introduites par le canal. Les pertes en performances pour une valeur du ($BER=10^{-5}$) varient de l'ordre de 0.197dB, pour une vitesse de $3Km/h$, jusqu'à atteindre les 0.227dB, pour une vitesse de $50Km/h$. Une autre conséquence de la transmission des données codées à travers un canal d'évanouissement (*Rayleigh*) est présentée dans la Figure 5.20 où l'on observe une nette diminution de la moyenne de l'indice de certitude sur chaque bit décodé, pour tous les niveaux de bruit, en comparaison avec celle d'un canal AWGN en raison des déformations infligées par les évanouissements du canal.

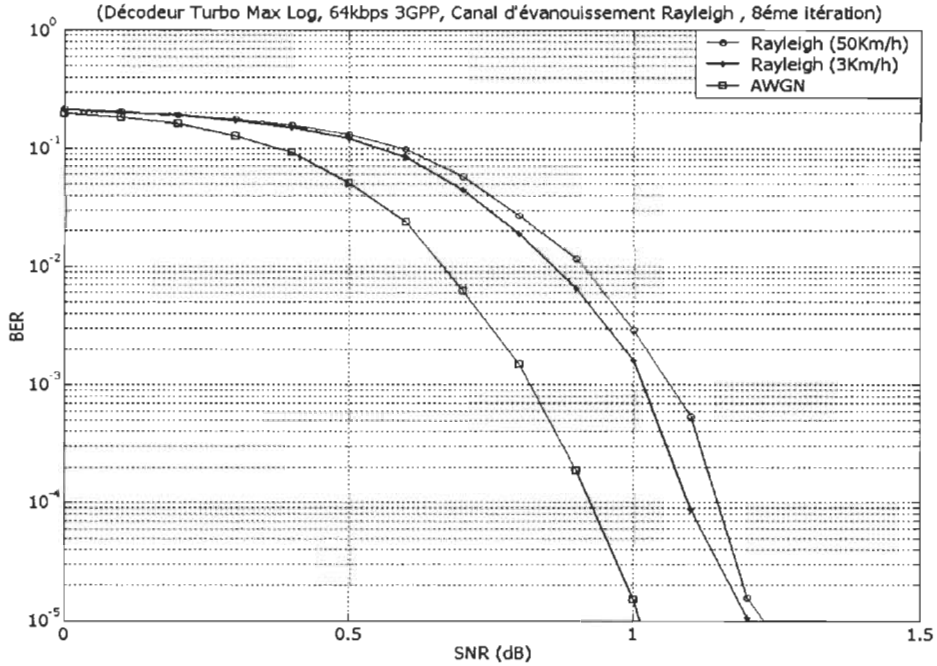


Figure 5.19 Turbo Max Log-MAP (Canal d'Évanouissement Rayleigh 3GPP)

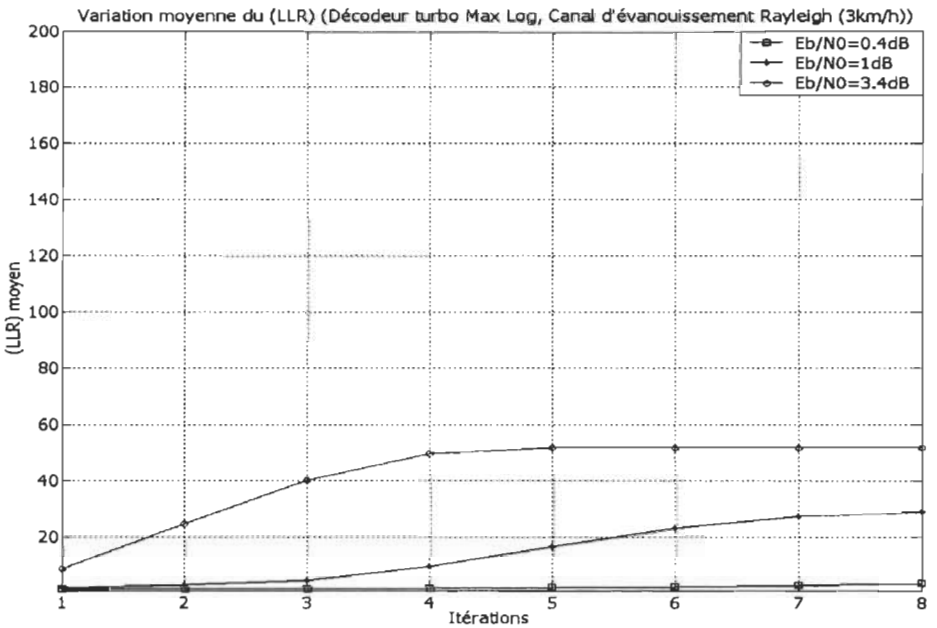


Figure 5.20 Variation Moyenne Turbo Max Log-MAP 3GPP

6. IMPLÉMENTATION DU TURBO CODE

6.1 Complexité Algorithmique

En raison de la complexité de calcul, l'implémentation directe de l'algorithme MAP n'est pas réalisable dans un système à temps réel. En vue de minimiser cette complexité, des algorithmes de calcul des métriques d'états (*State Metrcis*) dans le domaine logarithmique sont adoptés. Ainsi, les opérations de multiplications sont converties en addition (*Log-MAP Algorithm*), avec comme inconvénient, le traitement du logarithme de la somme d'exponentiels, un terme de correction de l'approximation issue de l'application de l'algorithme Jacobien (55). Une table de recherche (*Look Up Table*) peut donc être utilisée pour l'estimation de cette fonction de correction (56), bien que dans la plupart des implémentations, ce terme est souvent ignoré (57), (*Max-Log-MAP*).

Le Tableau 6.1 représente une analyse de complexité des algorithmes de turbo décodage, pour un encodeur convolutionnel (n,k) muni d'une mémoire v. De toute évidence, l'algorithme Log-MAP est approximativement trois fois plus complexe que l'algorithme SOVA, alors que le Max Log-MAP est approximativement deux fois plus complexe que le SOVA.

Tableau 6.1 Comparaison de Complexité de calcul des Algorithmes de Turbo Décodage

Operation	MAP	Log-MAP	ML-MAP	SOVA
Addition	$2\times 2^k \times 2^v + 6$	$6\times 2^k \times 2^v + 6$	$4\times 2^k \times 2^v + 8$	$4\times 2^k \times 2^v + 9$
Multiplication	$5\times 2^k \times 2^v + 8$	$2^k \times 2^v + 6$	$2\times 2^k \times 2^v$	$2^k \times 2^v$
Max Ops.		$4\times 2^v - 2$	$4\times 2^v - 2$	$2\times 2^k - 2$
Tab. Rech		$4\times 2^v - 2$		
Exponentiel	$2\times 2^k \times 2^v$			

6.2 Espace de Conception d'un Système de Turbo Décodage

Malgré la diminution significative de complexité produite par le décodage itérative par rapport au décodage optimal, l'implémentation d'un turbo décodeur reste une tâche relativement difficile, étant donné la réutilisation systématique et coûteuse de chaque décodeur. La simplification algorithmique apportée par le Max-Log-MAP permet une amélioration considérable des performances en exécution, mais une estimation de premier ordre de la complexité de calcul révèle qu'approximativement 1500 MOPS sont nécessaires pour assurer un débit de transmission de données de l'ordre de 2Mb/s, en cinq itérations, avec une contrainte de longueur de l'encodeur convolutionnel ($K = 9$). Une réduction de complexité est par conséquent envisagée à tous les niveaux d'abstraction (Système, Architecture, Transfère de Registre, Porte, Niveau Transistor) [BER99]; une implémentation efficace d'un turbo décodeur requiert une exploration de l'espace de conception des systèmes, avant toute

schématisation matérielle ou implémentation sur DSP, des spécifications algorithmiques. L'espace de conception d'un système de turbo décodage Fig. 6.1 est composé de deux composantes principales; une partie qui dépend du service à assurer et l'autre qui dépend directement de l'implémentation. Les éléments du turbo encodeur définissent directement la première partie (Structure de l'encodage, Perforation, Entrelacement), bien que le nombre d'itérations dépende en grande partie de l'implémentation, il dépend également de la qualité du service à réaliser. Les codes convolutionnels sont décodés avec l'algorithme MAP ou SOVA ; ainsi, le concepteur devrait choisir entre plusieurs options qui lui permettent de réduire la complexité de calcul, augmenter le débit ou de réduire la consommation de puissance, de même pour le traitement de l'information extrinsèque selon la méthode originale de Berrou [BERa93], ou la méthode directe Robertson [ROB94].

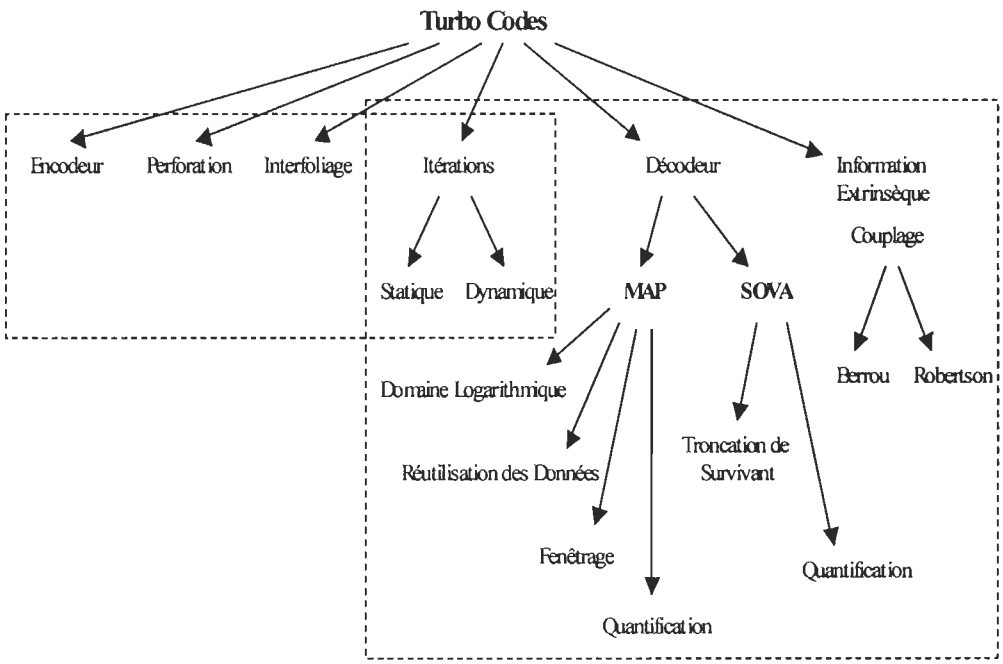


Figure 6.1 Espace de Conception d'un Système de Turbo Décodage

6.3 Exploration de l'Espace de Conception

Durant une correction d'erreur par décodage FEC, le taux d'erreurs dans la transmission des données (BER) est en fonction du niveau de bruit (SNR), cependant l'optimisation en implémentation influence cette fonction, par conséquent, l'exploration de la partie relative à l'implémentation du turbo code nécessite le recours à des modèles de simulation pour différents niveaux d'abstraction, en adoptant la méthodologie de conception de deux étapes :

1- La première étape utilise le modèle de référence comme entrée, en vue d'obtenir une version dite à bit réel (*Bit-true Model*), par l'optimisation et la quantification algorithmique.

2- Le modèle de bit réel est utilisé ensuite dans la deuxième partie, durant la conception matérielle, pour définir une structure de transfert registre, et développer ainsi un codage DSP optimisé.

Le modèle à bit réel du décodeur diffère selon le modèle d'architecture ciblé (Matériel, Logiciel, combinaison, Matériel/Logiciel), reflétant ainsi certaines exigences spécifiques, (La largeur du bit est une grandeur imposée dans une architecture DSP, et un critère d'optimisation dans une conception matérielle personnalisée). Un système de simulation VHDL doit être incorporé dans l'environnement de simulation pour la validation d'un modèle VHDL.

6.4 Optimisation

La complexité d'un turbo code est en fonction du décodeur et du nombre d'itérations utilisées. D'autre part, la complexité du décodeur dépend également de la puissance de l'opérateur, le nombre de données à réutiliser, le parallélisme et l'effet de quantification. L'implémentation de l'algorithme de SOVA à été proposée selon deux structures différentes : retraçage en arrière (*Trace-Back*) et la structure d'échange inter-registre. Différentes architectures peuvent donc être envisagées, toutefois les performances supérieures présentées par l'algorithme MAP [ROB95], avec un niveau de complexité similaire, ont grandement contribué à favoriser son implémentation dans un processus de turbo décodage, malgré les difficultés d'une représentation numérique des probabilités, les fonctions non linéaires et les opérations de multiplication et addition combinées de ces valeurs, nécessaires pour sa mise en oeuvre. La transformation de l'algorithme MAP dans le domaine arithmétique et la substitution des logarithmes restant par l'algorithme Jacobien permettent de réduire sensiblement les problèmes numériques du MAP, et facilite ainsi son implémentation. Des simplifications supplémentaires produisent l'algorithme (*Max-Log-MAP*), par l'omission du facteur de correction dans l'algorithme Jacobien, provoquant une dégradation des performance par rapport à l'algorithme de base (*MAP*).

En contraste avec le (Max Log-MAP), les boucles de calculs récursives l'algorithme (*Log-MAP*), comprennent des opérations d'addition-comparaison-sélection (*Add Compare Select*), avec une évaluation supplémentaire du terme de correction durant l'opération d'addition. Bien que cela nécessite l'allocution d'une

mémoire additionnelle, ce terme de correction est calculé par le bais d'une table d'allocution (*Look Up Table*), implémentée sous forme d'un Bloc de logique combinatoire, afin d'éviter l'augmentation de la consommation de puissance et la diminution de la vitesse de décodage, engendrées par l'utilisation d'un autre modèle d'architecture. Les opérations de calcul indépendantes du nombre d'itérations peuvent être calculées une fois seulement, durant tout le processus du turbo décodage; par conséquent, les résultats intermédiaires seront réutilisés durant chaque itération.

Le parallélisme inhérent dans le turbo décodage peut être exploité, en établissant un compromis entre la surface à utiliser d'une part, et la vitesse d'exécution, consommation de puissance, d'autre part. Dans les niveaux supérieurs d'implémentation, les unités fonctionnelles des éléments du décodeur, assurant le calcul des branches métriques, la récursivité en avant et en arrière (*Forward and Backward Recursion*) et la sortie douce, peuvent être parallélisés, ce qui permet de doubler le débit, en comparaison avec une autre solution en série. Le nombre d'amortisseurs (*Buffers*) dépend ainsi de la profondeur utilisée [BER99]. À d'autres niveaux, le parallélisme peut être introduit en observant que le décodage d'un Bloc de données reçues peut être divisé en plusieurs sous Blocs. Cette méthode de décodage est appelée technique de décodage par fenêtre glissante (*Sliding Window Technique*) [DAV95], ce qui permet d'augmenter le débit tout en minimisant la mémoire requise. D'autre part, la quantification permet de réduire la largeur du bit, produit par une formulation en point fixe du turbo code. Chaque bit de réduction se traduit par un impact significatif sur la surface et la consommation de puissance.

6.5 Implémentation du Turbo Code

Le turbo Max-Log-MAP a été parfaitement implémenté sur le DSP56603 de Motorola, pour l'exploitation des performances de l'algorithme du turbo décodage.

Le DSP56603 est un processeur numérique du signal (DSP) de 16 bits, pour traitement des données en point fixe, spécialement optimisé pour les applications dans le domaine de la communication sans fil (*Wireless Communication*), avec une vitesse de traitement de 80MIPS. Son architecture permet l'exécution parallèle de certaines paires d'instructions, (Ex. les opérations au niveau de l'unité arithmétique et logique, ou les transferts de données avec les registres de mémoire). Mais en raison des limites technologiques, une exploitation optimale du parallélisme au niveau d'instruction du processeur nécessite un développement manuel du code assembleur du DSP, produisant une nette diminution des performances (*48kb/s, 80MHz DSP56603*). Le schéma du stockage des données dépend essentiellement de l'architecture adoptée. Les résultats ont montré que le transfert des données constitue le principal obstacle de l'implémentation d'un turbo code, en dépit de la complexité de calcul. L'objectif d'atteindre la vitesse de traitement de 2Mb/s nécessite ainsi l'assemblement en parallèle de 40 DSP. Un autre modèle d'architecture, en vue d'une implémentation sur matériel configurable, est aussi possible. Une implémentation sur FPGA munie d'une horloge de 20MHz produit une sortie de 20kb/s. Ainsi, un traitement de 2Mb/s ne peut être assuré que par une architecture parallèle constituée de dix FPGA [BER99].

6.6 Conception Combinée (Logicielle / Matérielle)

Les performances d'une implémentation purement logicielle d'un turbo code sont certainement en dessous des attentes en matière de vitesse de décodage. Par contre, malgré le manque de flexibilité, une implémentation matérielle produit de meilleures performances, ce qui permet de constater la nécessité d'envisager une implémentation combinée (Matérielle/Logicielle). Un modèle d'implémentation qui présente à la fois l'avantage de la flexibilité d'une implémentation logicielle (Adaptation facile des exigences des différents modèles de canal de transmission) d'une part, et les performances d'une implémentation matérielle d'autre part. Cependant, Deux approches différentes peuvent être envisagées :

1- Pour une approche privilégiant la performance, une implémentation matérielle est appliquée au traitement développé pour chaque itération, tandis que les éléments de calcul, développés communément pour chaque bloc de données, seront implémentés selon le modèle purement logiciel.

2- Pour une approche privilégiant la flexibilité, une implémentation logicielle est assurée aux éléments de calcul nécessitant plus de flexibilité, alors que les autres parties fixes seront réalisées selon un processus matériel.

CONCLUSION

Dans cette étude comparative, les différentes techniques utilisées, les éléments de structures et les performances résultantes d'un système de décodage, selon le standard de la troisième génération d'un système de communication sans fil, ont été présentés. Cette description nous a permis de passer en revue les méthodes de codage linéaire, convolutionnel et le turbo codage.

Malgré la possibilité de pouvoir décoder un turbo code de manière optimale, par une seule étape non itérative [BRE00], le décodage itératif d'un turbo code qui nous permet de se rapprocher encore plus de la limite de Shannon par l'augmentation du gain de codage après chaque itération, est souvent préféré pour des raisons de complexité. Ainsi, nous avons défini le modèle mathématique des algorithmes de décodage MAP, Log-MAP et Max-Log-MAP utilisés en turbo décodage, en vue d'évaluer leur complexité de calcul, leur sensibilité au bruit leurs limites de performances par simulation ainsi que l'influence des éléments de la structure de turbo décodage, sur l'évolution de ces performances. L'algorithme Max-Log-MAP présente le meilleur compromis, performance complexité de calcul, apporté par la simplification de l'algorithme optimal d'origine MAP. Cette simplification a réduit considérablement la sensibilité du MAP au bruit; cependant, la perte de performance

causée par cette approximation est compensée par une pondération adéquate de l'information extrinsèque dans la sortie douce de chaque décodeur.

La recherche continue dans plusieurs domaines, notamment dans l'espoir d'améliorer le compromis entre performance et complexité de calcul, ainsi que l'approche de la limite de Shannon dans le cas d'un canal de communication sans fil, dispersive ou d'évanouissement (*Ex : Évanouissent Rayleigh*).

Une implémentation logicielle (*Software Implementation*) d'un turbo code est clairement en dessous des performances exigées en matière de vitesse de traitement des données. Bien que les performances d'une implémentation matérielle (*Hardware Implementation*) soient meilleures, le manque de flexibilité de cette méthode, nécessaire pour une adaptation aux différents modèles de canaux de transmission, ou services exigés (Ex : Perforation), a stimulé aussi le domaine de recherche dans l'implémentation mixte d'un turbo code.

BIBLIOGRAPHIE

[3GPP01] 3GPP Technical specification, "Technical Specification Group Radio Acces Networks; UE Radio Transmission And Reception (FDD)," *3GPP TS 25.101 v5.0.0 (2001-09)*.

[BAH74] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimum Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 248-287, March 1974.

[BAR94] A. S. Barbelescu and S. S. Pietrobon, "Interleaver Design for Turbo Codes," *Electron. Lett.*, vol. 30, no. 25, pp. 2107-2108, Dec. 1994.

[BAT87] G. Battail, "Ponderation des Symbols de Codes par l'Algorithme de Viterbi," Ann. Telecommun., vol. 42, no. 1-2, pp. 31-38, Jan. 1987.

[BENa96] S. Benedetto and G. Montorsi, "Design of Parallel Concatenation Convolutional Codes," *IEEE Trans. Commun.*, vol. 44, pp. 591-600, May 1996.

[BENb96] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409-428, Mar. 1996.

[BER96] C. Berrou, A. Glavieux, "Near Optimum Error Correcting Coding and Decoding : Turbo-Codes," *IEEE Trans. On Communication*, vol. 44, pp. 1261-1271, Oct. 1996.

[BER99] F. Berens, A. Worm, H. Michel,N. Wehn, "Implementation Aspects of Turbo-Decoders for Future Radio Applications," *In Proc. VTC '99 Fall*, pages 2601-2605, Sept. 1999.

[BERa93] C. Berrou, A. Galvieux and P. Thitimajshima " Near Shannon Limit Error-Correcting Coding and Decoding :(Turbo Codes),". *École Nat. Sup de Télécommunications De Bretagne France.1993.*

[BERb93] C. Berrou, P. adde, E. Angui, and S. Faudeil, " A low Complexity Soft-Output Viterbi Decoder Architecture," *in Proc. int. Conf. Communications*, May 1993, pp. 737-740.

[BOS99] M. Bossert, "Channel Coding for Telecommunications," *John Wiley And Sons Ltd, 1999.*

[BRE00] M. Breiling, L. Hanzo, "The super-trellis structure of turbo codes," *IEEE Trans. Inform. Theory*, Sept. 2000.

[CHR00] M. Chryssomallis, "Smart Antennas," *IEEE Antennas and Propagation Magazine*, vol. 42, No. 3, 2000, pp 129-136.

[COV91] T. M. Cover and J. A. Thomas, "Elements of Information Theory," Wiley-Interscience, August 1991.

[DAS98] K. Das, and S. D. Morgera, "Adaptative Interference Cancellation for DS-CDMA Systems using Neural Networks Techniques," *IEEE Journal on Selected areas in Communications*, vol. 16, No. 9, 1998, pp. 1774-1784.

[DAV95] H., David, H. Meyer, "Real-Time Algorithms and VLSI Architectures for Soft Output MAP Convolutionnel Decoding," *The Sixth International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* vol. 1, pp. 193-197, Sept. 1995.

[DIV96] D. Divsalr, and R. J. McEliece, "Effective free distance of Turbo codes," *Electronics Letters*, vol. 32, no. 5, pp. 445-445, Feb. 1996.

[DOL95] S. Dolinar and D. Divsalar, "Weight Distribution for Turbo Codes using Random and Non-Random Permutations," *Communications Systems and Research section* TDA report 42-122 August 1995.

[ERF94] J. A. Erfanian, S. Pasupathy, and G. Gulak, "Reduced Complexity Symbol Detectors with Parallel Structures Fir ISI Channels," *IEEE Trans. Commun.*, vol. 42, pp. 1661-1671, 1994.

[FOR73] G. D. Forney, "The Viterbi Algorithm," *Proc. of the IEEE*, vol. 61, pp. 268-278, Mar 1973.

[GOF94] S. Le Goff , A. Glavieux, and C. Berrou, "Turbo-Codes and High Spectral Effenciency Modulation," *Proc. IEEE Int. Conf Communications,* pp. 645-649, 1994.

[HAG89] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *IEEE Globecom*, pp. 1680-1686, 1989.

[HAG95] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43 pp. 2449-2457, Sept. 1995.

[HAGa96] J. Hagenauer, "Forward error correcting for CDMA systems," *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications* Proceedings, vol. 2, 1996, pp 566-569.

[HAGb96] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, pp. 429-445, Mar. 1996.

[HAL01] H. Haloma, A Toskala, "WCDMA for UMTS Radio Access For Third Generation Mobile Communications," Revised Edition, Wiley-Interscience, 2001.

[HAM50] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell syst. Tech. J.,* vol. 29., pp. 147-150, 1950.

[HUA98] W. huang, I. Andonovic et M. Nakagawa, "PLL Performance of DS-CDMA Systems in the Presence of Phase Noise, Multiuser Interference, and Additive Gaussian Noise", *IEEE Transactions on Communications*, vol. 46, No. 11 1998, pp 1507-1515.

[JUN96] P. Jung, "Comparaison of Turbo Code Decoders Applied to Short Frame Transmission Systems," *IEEE J. Select. Areas Commun.*, pp. 530-537, 1996.

[JUNa94] P. Jung and M. Nabhan, "Performance Evaluation of Turbo Codes for Short Frame Transmission Systems," *IEEE Electron. Lett.*, pp. 111-112, Jan. 1994

[JUNb94] P. Jung and M. Nabhan, and J. Blanz, "Application of Turbo-Codes to a CDMA Mobile Radio System using Joint Detection and Antenna Diversity," Proc. IEEE Conf. Veh. Technol., pp. 770-774, 1994.

[KAH99] A. K. Khandani, "Optimization of the interleaver structure for turbo codes," *Proceeding of the 1999 Canadian workshop on information theory*, pp. 25-28, Jun. 1999.

[KIM00] Y-H. Kim, W-S Yoon, "Efficiently Combined Structure of Smart Antenna and Interference Canceller using Symbol Reliability," Electronics Letters, vol. 36, No. 18, 2000, pp 1583-1584.

[KLE96] S. Klein, G. K. Kaleh, P. W. Baier, "Zero Forcing and Minimum Mean Square-Error Equalization for Multiuser Detection in Code-Division Multiple-Access Channels," *IEEE Transactions on Vehicular Technology*, vol. 45, No. 2, Mai 1996, pp. 276-287.

[KOC90] W. Koch and A. Baier, "Optimum and Sub-Optimum Detection of Coded Data Disturbed by Time-Varying Inter-Symbol Interference," *IEEE Globecom*, pp. 1679-1684, Dec. 1990.

[LAT00] M. Latva-aho, M. J. Juntti, "LMMSE Detection for DS-CDMA Systems in Fading Channels," *IEEE Transactions on Communications*, vol. 48, No. 2, 2000, pp 194-199.

[LEH99] H. Lehne, et M. Petterson, "An Overview of Smart Antenna Technology for Mobile Communications Systems," *IEEE Communicatrions Surveys*, vol. 2, No 4, 1999, pp. 2-13.

[PER96] L.C. Perez, J. Seghers, and D. J. Costello, "A Distance Spectrum Interpretation of Turbo Codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698-1709, Nov. 1996.

[PET95] L. R. Peterson, R. E. Zeimer, et D. E. Borth, "Introduction to Spread Spectrum Communications," Prentice-Hall, 1995.

[PRA98] R. Prasad, T. Ojanpera, "An Overview of CDMA Evolution Toward Wideband CDMA," *IEEE Communications Surveys*, vol. 1, 1998,

[ROB95] P. Robertson, E Villeburn, and P. Hoeher, "A Comparaison of Optimal and Sub-optimal MAP Decoding Algorithms Operating in the Log Domain," *in Proc. Int. Conf. Communications*. June 1995, pp. 1009-1013.

[PRO94] J. G. Proakis, M. Salehi, "Communication Systems Engineering," Prentice-Hall, 1994.

[PRO95] J. G. Proakis, "Digital Communications," Third Edition, MacGraw Hill, 1995.

[PYN97] R. Pyndiah, "Iterative Decoding of Product Codes: Block Turbo Codes," *in proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 1997, pp. 71-79.

[RAP94] P. B. Rapajic, et B. S. Vucetic, "Adaptative Receiver Structures for Asynchronous CDMA systems," *IEEE Journal on Selected areas in Communications*, vol. 12, No. 4, 1994, pp 684-697.

[ROB94] P.Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," *In. Proc. Globecom 94'*, pages 1298-1303, Dec. 1994.

[ROB98] P. Robertson and T. Worz, "Bandwidth-Efficient Turbo Trellis-Coded Modulation using Punctured Components Codes," *IEEE J.Select. Areas Commun*, vol. 16, pp 206-218 Feb 1998.

[SAD00] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi, and G. Nebe, "Interleaver Design for Turbo Codes" *Prod. AT&T Shannon Labs*, Nov. 2000.

[SHA48] C. E. Shannon, "A Mathematical Theory of Communication," *Bell syst. Tech. J.*, vol. 27., pp. 379-423, 1948.

[SKL97] B. Sklar, "A Premier on Turbo Code Concepts," *IEEE Communications Magazine*, pp. 0163-6804. Dec 1997.

[TIN00] J. Tingfang et W. E. Stark, "Turbo-Coded ARQ Schemes for DS-CDMA Data Networks over Fading and Shadowing channels; Throughput, Delay, and Energy Efficiency," *IEEE Journal on Selected areas in Comuunications*, vol. 18 No. 8, 2000, pp 1355-1364.

[VIT67] A. J. Viterbi, "Error Bounds for Convolutionnel Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.

[VIT96] A. Viterbi, "Approching The Shannon Limit : Theorist's Dream and practitioner's challenge," *in Proc. int. conf. Millimiter Wave and Far In-frared Science and Technology*, 1996, pp. 1-11.

[VIT97] A. J. Viterbi, "An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutionnel Codes," *IEEE J. Select. Areas Commun.*, pp. 260-264, Feb. 1997.

[VOG00] J. Vogt, A. Finger, "Improving the max-log-MAP turbo decoder," Electronics Letters. Vol. 36. No. 23. Nov. 2000

[WAC95] U. Wachsmann and J. Huber, "Power and Bandwidth Efficient Digital Communications using Turbo Codes in Multilevel Codes," *Eur, Trans. Telecommun.*, vol. 6, pp. 557-567, Sept.-Oct 1995.

[WAN99] X. wang, et H. V. Poor, "Blind Joint Equalization and Multiuser Detection for DS-CDMA in unknown Correlated Noise," *IEEE Transactions on Circuits and Systems 2 : Analog and Digital Signal Processing*, vol. 46, No. 7, 1999, pp. 886-895.

[WOO00] P. J. Woodard, L. Hanzo, "Comparative Study of Turbo Decoding Techniques : An Overview," IEEE Trans. On Vehicular Technology. Vol. 49, No. 6. Nov 2000.

[WOO99] G. K. Woodward, "Adaptative Detection for DS-CDMA," These de Doctorat en Génie Électrique, University of Sydney 1999.

[WOR00] A. Worm, P. Hoener, N. Wehn, "Turbo-Decoding Without SNR Estimation," IEEE Communications lettre, vol.4 NO. 6 Jun. 2000.

[WOZ61] J. M. Wozencraft and B. Reiffen, "Sequencial Decoding," *Cambridge, MA:* MIT press, 1961.

[ZHA00] Y. Zhang, R. S. Blum, "Multistage multiuser detection for CDMA with Space-Time Coding," *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Arry* Processing, 2000, pp 1-5.

Analyse des Décodeurs Turbo pour les Systèmes UMTS

Mohamed ZEBDI, Messaoud AHMED OUAMEUR,
Adel Omar DAHMANE et Daniel MASSICOTTE
Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières
C.P. 500, Trois-Rivières, Québec, Canada, G9A 5H7
Tel. : (819)-376-5011 (3918), Fax : (819)-376-5219, Web: http://lssi.uqtr.uquebec.ca
E-mail : {Mohamed_Zebdi, Daniel_Massicotte}@uqtr.uquebec.ca

Abstract

Cette contribution consiste à présenter l'apport du Codage Canal dans un Système de Communication sans fil (WCDMA), pour la Troisième Génération. L'étude portera en premier lieu sur le Système de Codage Conventionnel, pour introduire une étude détaillée sur le Turbo Code. Ainsi, une étude comparative sera élaborée entre le Codage Convolutionnel et le Turbo Codage d'une part, et entre les différents algorithmes de décodage itératif (Turbo décodage), selon le standard de la Troisième Génération, en mettant en évidence l'apport des méthodes de décodage douce (Soft Decoding Decoding) par rapport au méthodes de décodage dure (Hard Decision Decoding), en l'occurrence l'algorithme de Viterbi.

Une série de Paramètres sera systématiquement introduite en vue de compléter l'étude comparative en mettant en évidence la pertinence des normes 3GPP.

Le Canal AWGN (12kbps) et (64kbps) du standard 3GPP seront ainsi utilisés respectivement pour le Codage Convolutionnel et le Turbo Codage.

1. Introduction

Le Turbo Codage était introduit en 1993 par Berrou, Galvanieux, et Thitimajashima [1],[2], comme une méthode de Codage permettant d'atteindre, avec de longues trames, des performances exceptionnelles, jamais atteinte avant avec un encodeur Convolutionnel [3]. Depuis sont invention, le Turbo Codage a bénéficié d'une recherche intensive pour atteindre un stade de maturité dans l'espace que quelques années seulement. Par Conséquent, le Turbo Codage a trouvé une place de choix dans le Standard de la Communication sans fil pour la Troisième Génération [7]. Des gains aussi impressionnant peuvent être atteint avec l'aide du Turbo Codage surtout avec les Systèmes de Communication dont le délai associé, est moins critique que celui d'un Système de Communication sans fil .

Dans leur article, Berrou et al. [1] et [2] ont utilisé deux Encodeurs Convolutionnel Récursives et Systématiques (*RSC*) Codes, avec un interfoliage (*Interleaver*) entre les deux encodeurs. Cette structure permet de minimiser le rapport de transmission des données, augmenter la distribution des poids du Turbo Encodeur , ce qui permet d'atteindre de meilleurs performances durant le Turbo Décodage [4] [5].

Contrairement au Algorithmes de décodage basés sur les décision dures (*Hard Décision Decoding*) [6] comme le Viterbi, utilisés pour les encodeurs Convolutionnel simples, les algorithmes de décodage pour les systèmes du Turbo Codage, sont des algorithmes de décodage avec décision douce (*Soft Input/Soft Output*), ce qui permet de procéder à un processus de décodage itérative. Une étude comparative est ainsi réalisée, entre les différents algorithmes de décodage itératif, comparaison avec l'algorithme de Viterbi, selon le standard de la 3GPP.

2. Codage Convolutionnel

La configuration du Codage Convolutionnel pour la Troisième Génération est présenté dans la figure 1. Les encodeurs (a), (b) sont de Contrainte de Longueur $K = 9$, et un rapport d'encodage ($R=1/2$), ($R=1/3$) respectivement. La figure 2, montre un gain en performance apporté par l'application de l'algorithme de

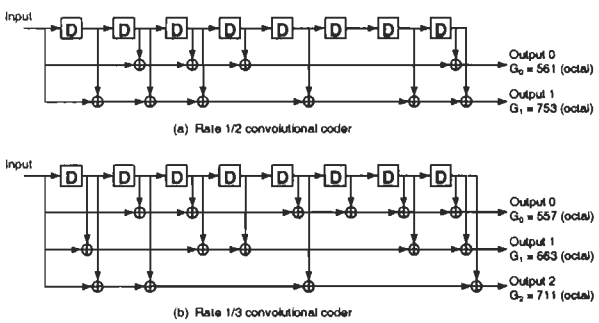


Fig.1 Codage Convolutionnel 3GGP

l'algorithme de Viterbi (3,1,9), en comparaison avec le Viterbi (2,1,9), de l'ordre de 2.1dB, pour un (BER=10^{-3}), (Valeur de référence en matière de

transmission de la voix) . Ce gain en performance permet une réduction de la puissance d'un rapport de prés de 1.6, pour chaque bit transmis. Dans les deux cas, atteindre une valeur du ($BER=10^{-5}$), nécessite un rapport d'énergie par bit transmit, sur puissance de bruit supérieure à 6dB, ce qui traduit un niveau de communication peu fiable, pour un niveau de bruit très élevé.

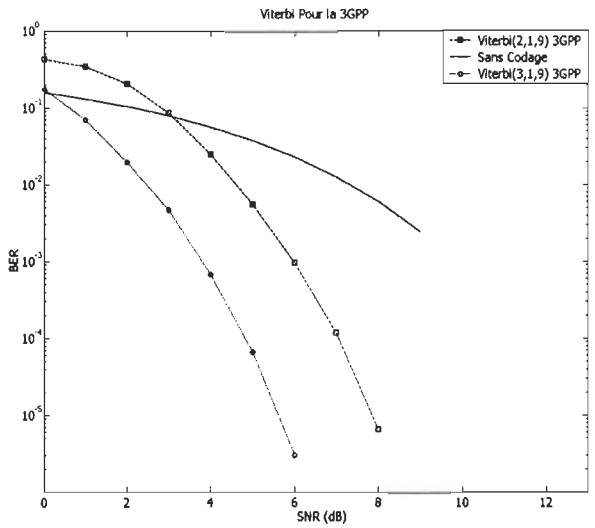


Fig.2 Algorithme de Viterbi pour la 3GGP

3. Turbo Code

La structure du Turbo Code adoptée par le standard 3GPP fig.3, est constituée de deux encodeurs convolutionnels parallèles et concaténés (*PCCC*) [5], de contrainte de longueur $K=4$, avec les polynômes générateurs respectives :

$$g_0(D) = 1 + D^2 + D^3 \quad (1)$$

$$g_1(D) = 1 + D + D^3 \quad (2)$$

munie d'un interfolieur (Interleaver) interne, assurant un rapport de codage (Rate=1/3).

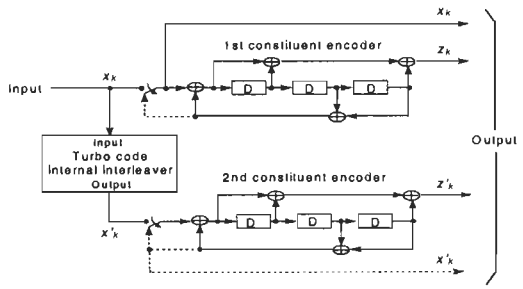


Fig.3 Turbo Encodeur 3GGP

Les simulations ont été réalisées, en assumant une modulation BPSK de la donnée utile [6], avec une réception parfaite des données transmises à travers un canal sans effet de mémoire. Sauf indication contraire, le canal de transmission est un canal (*AWGN*), caractérisé par la superposition d'un bruit additionnel de distribution Gaussienne, à l'informations transmise. Le standard 3GPP [7], représente le cadre général des simulations, à la fois pour le Code Convolutionnel, et le Turbo Code.

3.1 Algorithmes de Turbo Décodage

La figure 4 présente, une comparaison de performances entre un turbo décodeur utilisant l'algorithme MAP, et les algorithmes de décodage produits par l'approximation du MAP, dans le domaine logarithmique. Pour un rapport d'erreur de décodage ($BER=10^{-5}$), la dégradation de performance observée, pour l'approximation de l'algorithme MAP, apportée par le Log MAP et le Max Log MAP est au voisinage de 0.1dB et 0.5dB respectivement.

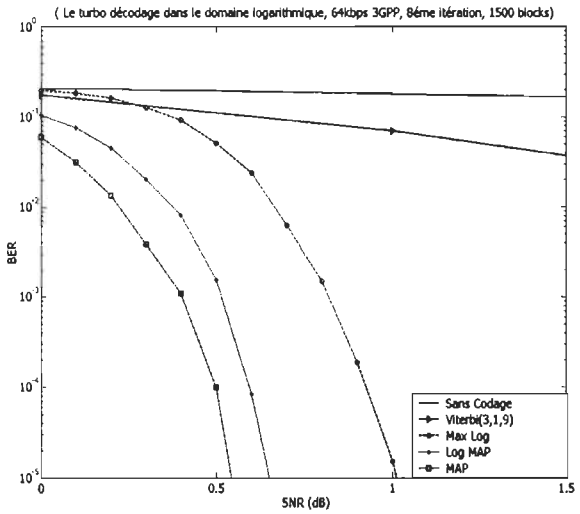


Fig.4 Turbo Décodage dans le domaine Logarithmique pour la 3GGP

En d'autre part, la Figure 5 montre la variation de la moyenne, de la sortie douce du second décodeur du turbo Max Log MAP, selon une multitude de niveaux de bruit. Cette valeur moyenne du *LLR* (*Log Likelihood Ratio*), représente un indice de confiance sur le résultat de décodage, durant chaque itération.

Pour des niveaux de bruit relativement élevés ($SNR=0.4dB$), le turbo Max Log MAP n'a toujours pas atteint le niveau de saturation à la huitième itération, ce qui justifie le recours à un supplément d'itérations. Par contre, ce niveau de saturation peut être atteint, dés la 5éme itération pour ($SNR=3.4dB$), pour les algorithmes Max Log MAP et MAP (fig. 6).

Les valeurs relativement supérieurs des indices de confiance, montre la puissance de l'algorithme Turbo

MAP, en comparaison au méthodes sous optimales, produites par son approximation dans le domaine logarithmique.

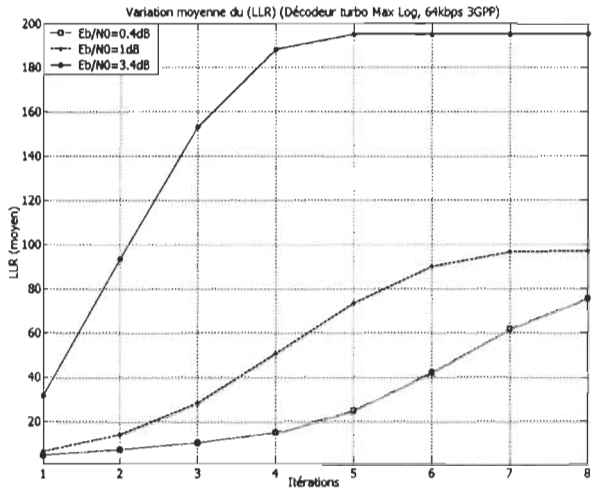


Fig.5 Sortie Douce du Turbo Max Log MAP pour la 3GPP

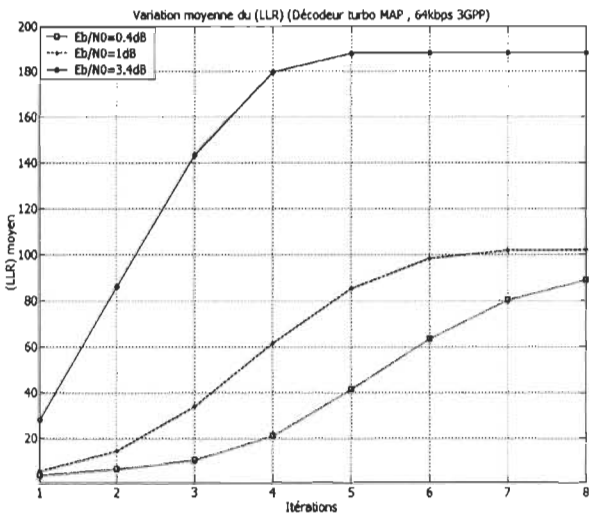


Fig.6 Sortie Douce du Turbo MAP pour la 3GPP

La variation moyenne de la sortie douce du second décodeur (*Soft Output*), représente ainsi un outil supplémentaire de comparaison entre les différents algorithmes de décodage, qui permet de justifier le choix d'un algorithme de turbo décodage.

3.2 Longueur de Trame

Des résultats impressives ont été présentés dans plusieurs publications, pour l'application d'un processus de turbo décodage, sur des longues trames, de séquences de bits en entrée de l'encodeur. Ce pendant, pour plusieurs modèles d'application, (*ex : la transmission de la voix*), les longs délais causés par l'application de ces longues trames en entrée, est pratiquement inacceptable. Par conséquent, un grand effort de recherche a été entrepris, pour permettre d'approcher les résultats déjà réalisés avec les longs trames [9]. La Figure 7, démontre le grande influence de la longueur du trame, sur les performances d'un turbo code. L'application du Max Log MAP sur un canal de 64kbps du standard 3GPP, inflige une dégradation des performances de l'ordre de 0.155dB, pour l'utilisation du canal 2084 kbps, dans les même conditions. Cette dégradation est justifiée par une meilleure décorrélation par interfoliage, réalisée par un (*Interleaver*) du même standard, entre la sortie douce (*Soft Output*) de chaque décodeur correspondant à ces bits de parité, et la séquence d'information dans l'entrée du même décodeur [10]. Cette dégradation est réduite à 0.095dB et 0.085dB, pour l'utilisation du canal 384 kbps et 144kbps respectivement, par rapport à un canal de 64kbps, du standard de 3GPP.

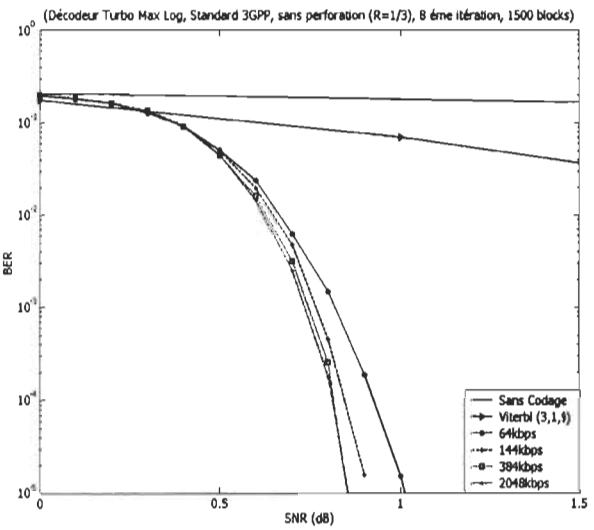


Fig.7 8éme itération Turbo Max Log 3GPP

La comparaison avec le Codage Convolutionnel (3,1,9) du standard 3GPP, est justifié par le fait qu'il représente le même ordre de complexité, qu'un processus de turbo décodage de 8 itérations, avec l'algorithme Log MAP, on ignorant le fait que seul les informations apriorique (*A-priori Information*), qui changent après chaque itération [11]. Il est donc important de noter que la complexité d'un processus de turbo décodage de 8 itérations, peut être réduite par le stockage les informations utilisées durant la première itération, en vue de leur réutilisation pour les itérations future. La figure 7 montre, le gain en codage d'un turbo code, par rapport au Codage Convolutionnel (3,1,9), pour les différentes longueurs de trames, du standard 3GPP.

3.3 Sensibilité au bruit

Selon l'algorithme choisi dans un processus de turbo décodage, l'évaluation de l'influence d'une éventuelle estimation du bruit à la réception, sur les performances du turbo codage, est un des éléments clé dans le processus d'implémentation du turbo code. La Figure 8, montre la sensibilité des performances, de la huitième itération du turbo décodage avec l'algorithme MAP, par rapport à une erreur d'estimation du bruit à la réception (*Offset*). On remarque une erreur de prés de 10^{-1} sur le (*BER*), correspondant à une erreur d'estimation du bruit (*Offset*) de 1dB, pour ($E_b / N_0 = 0dB$), jusqu'à atteindre la valeur de ($BER = 10^{-4}$), correspondant à 0.5dB d'erreur d'estimation du bruit, pour un ($E_b / N_0 = 0.5dB$). Cette variation est justifiée par le fait que l'équation du calcul des métriques des branches (*Branch Metrics*), élément de base pour le calcul de la sortie douce (*Soft Output*) de l'algorithme MAP, est une fonction non linéaire du bruit [12] .

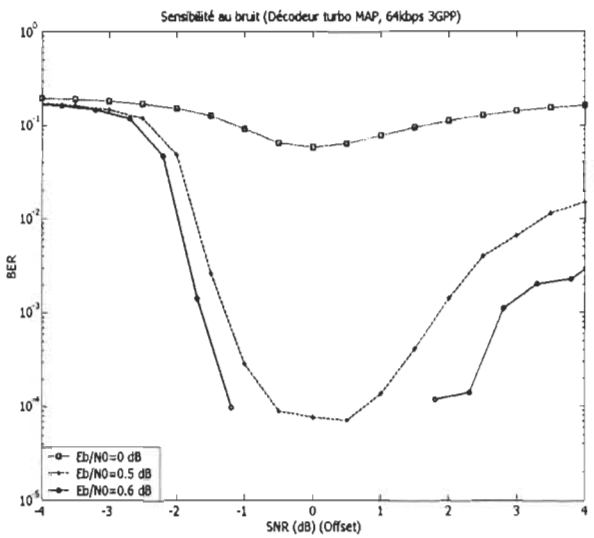


Fig.8 Sensibilité au Bruit (Turbo MAP 3GPP)

L'approximation apportée par l'algorithme Jacobien dans le Log MAP, produit un terme de correction, qui est aussi une fonction non linéaire du bruit [11]. La sensibilité du Log MAP au bruit est donc maintenue, étant donnée qu'une estimation du bruit reste nécessaire pour cette algorithme, à la fois pour le calcul des métriques de branches, et la pondération des informations constituant la sortie douce de chaque encodeur [12]. La Figure 9, montre l'insensibilité au bruit, de l'algorithme Max Log MAP, quelque soit le niveau de bruit. Cette conclusion, est une des conséquences de l'approximation du l'algorithme MAP, qui permet le calcul des métriques des branches, par une fonction linaire du bruit [11].

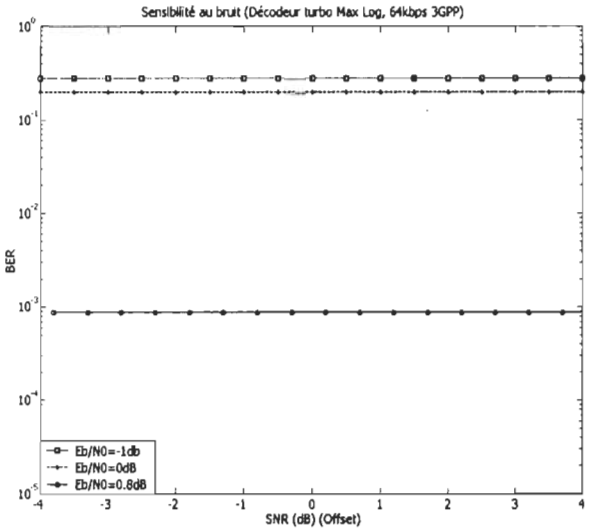


Fig.9 Sensibilité au Bruit (Turbo Max Log 3GPP)

3.4 Interfoliage

Il est évident de constater l'importance de l'entrelacement, sur les performances d'un turbo décodeur. La conception de l'entrelaceur et le polynôme générateur , utilisés dans la structure d'encodage, ainsi que la perforation utilisée au niveau du décodage à un effet crucial, sur la distance minimale du turbo code [13].Plusieurs propositions ont été présentés, pour définir les critères de conception d'un « bon » interfoliage [14], basé sur la maximisation de la distance libre minimale (Maximum Free Distance), mais processus reste très complexe, et les interfoliages résultants ne sont pas nécessairement optimaux. La figure 10 présente, le résultat de simulation d'un décodage par l'algorithme Max Log MAP, avec un interfoliage (*Ligne-Cologne*). En outre la dégradation des performances par rapport à l'entrelacement (*Pseudo-aléatoire*) du standard 3GPP, on remarque l'influence de la faible distance libre minimale (d_{min}), causant une saturation à partir de la valeur ($BER = 10^{-4}$). Cette saturation, traduisant une grande corrélation entre, la sortie douce de chaque décodeur relative à ces bits de parité, et la séquence des bits d'information, à son entrée [14].

La figure 10 montre aussi, le résultat de la même simulation avec un interfoliage (*S-aléatoire*), (*Pseudo-aléatoire*), et l'interfoliage aléatoire. Un gain de performances de 0.07dB, par rapport à l'interfoliage pseudo aléatoire du standard 3GPP, pour une valeur de ($BER = 10^{-4}$). Par conséquent, un plus faible rapport d'erreur par bit décodé (*BER*), pour rapport d'énergie par bit transmit sur niveau de bruit (E_b / N_0) faible, est réalisable uniquement avec un interfoliage aléatoire.

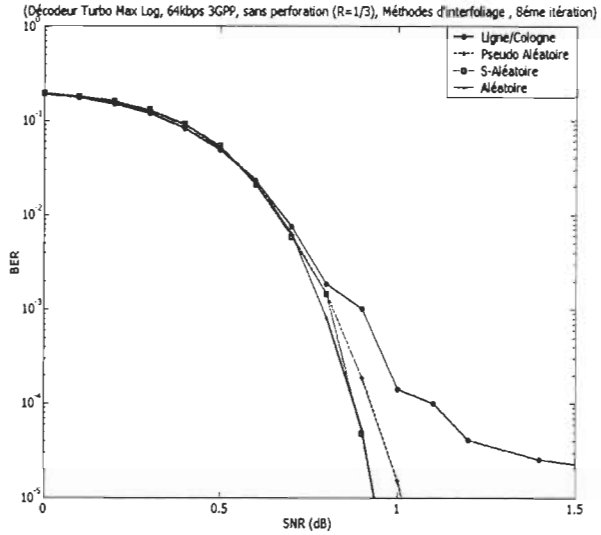


Fig.10 Interfoliage en turbo Décodage

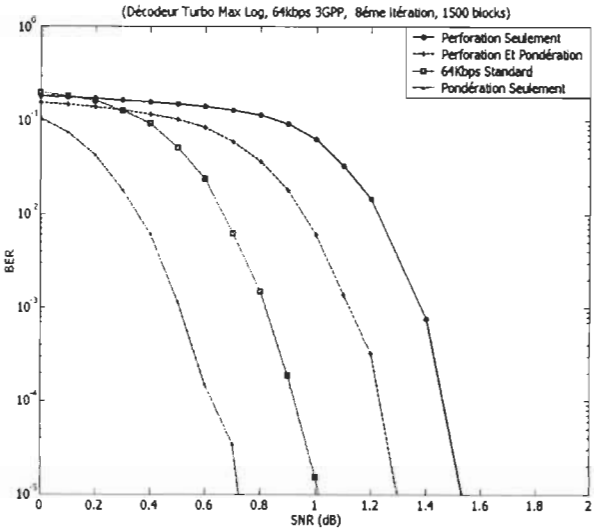


Fig.11 Perforation/Pondération (Turbo Max Log 3GPP)

3.5 Perforation

En turbo décodage, deux ou plusieurs encodeurs peuvent être utilisés, pour la génération des bits de parité (*Parity bits informations*), à partir des bits de données en entrée. Selon le standard 3GPP, deux encodeurs convolutionnels récursives sont seulement utilisés dans un processus de turbo décodage, pour un rapport de codage (Rate=1/3). Dans le but de produire un rapport de codage (Rate=1/2), la moitié des bits de parité de chaque encodeur seront perforées (*Punctered*), selon l'arrangement d'utilisée par Berrou dans son article original [1]. Toute fois il est possible d'omettre cette perforation, et transmettre tous les bits de parité de chaque encodeur. La figure 11 représente le résultat de simulation d'un turbo décodeur, appliquant l'algorithme Max Log MAP, sur le canal 64kbps du standard 3GPP, sans et avec perforation. La transmission de tous les bits de parité, résulte en un gain de presque 0.5dB, pour ($BER = 10^{-5}$), ce qui représente un gain de puissance en transmission canal, au voisinage de 3dB.

3.6 Pondération

Une surestimation a été observée, de l'information extrinsèque produite par chaque décodeur, à partir des bits de parités introduits à son entrée. Une pondération adéquate de cette information, avant ça transmission à un autre décodeur, permet d'améliorer considérablement les performances du turbo décodage, pour des algorithmes de décodage insensible au bruit [15]. La figure 11 montre un gain de performance de prés de 0.7 dB, par rapport au standard 3GPP (*sans pondération*), en utilisant un coefficient de pondération de l'information extrinsèque du Max Log MAP ($\alpha = 0.7$), ce qui permet un gain en puissance de prés de 5dB.

3.7 Canal d'évanouissement (Rayleigh)

Dans cette section, les mêmes séquences seront transmises à travers un canal d'évanouissement (Rayleigh), en assumant que le récepteur possède une estimation exacte des amplitudes et phase d'évanouissement, infligé par le canal.

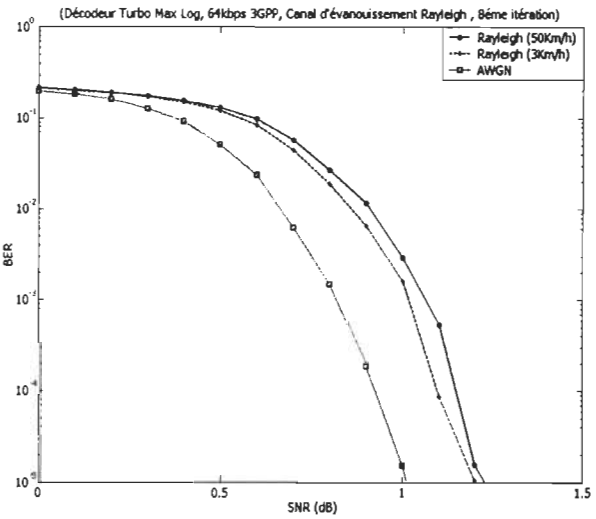


Fig.12 Perforation/Pondération (Turbo Max Log 3GPP)

La figure 12 montre la dégradation de performances, pour un canal d'évanouissement (*Rayleigh Fading Channel*), parfaitement interfolié, en assumant une parfaite connaissance à la réception, des amplitudes et phases introduite par le canal. Les pertes en performances pour une valeur du (BER=10^{-5}), varient de l'ordre de 0.197dB, pour une vitesse de $3Km/h$, jusqu'à atteindre les 0.227dB, pour une vitesse de $50Km/h$. Une autre conséquence de la transmission des données codées à

travers un canal d'évanouissent (*Rayleigh*), est présenté dans la figure 13, où on observe une nette diminution de la moyenne de l'indice de certitude, sur chaque bit décodé, en comparaison avec celle d'un canal (*AWGN*).

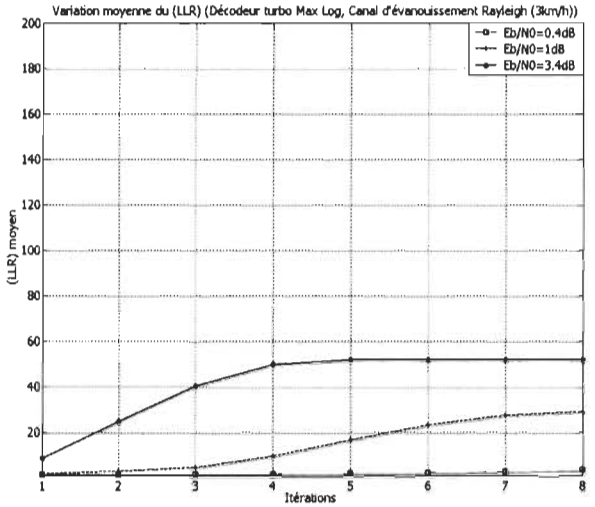


Fig.13 Évolution de la Sortie Douce avec un Canal d'Évanouissent Rayleigh (Turbo Max Log 3GPP)

4. Conclusion

Dans cette étude comparative, les différentes techniques utilisées, éléments de structures adoptées, et les performances résultantes d'un système de décodage, selon le standard de la troisième génération d'un système de communication sans fil, ont été présentées.

Cette description nous a permis de passer en revue les méthodes de codage convolutionnel, et le turbo codage qui nous permet de se rapprocher de la limite de Shannon, par l'augmentation du gain de codage après chaque itération, en vue d'évaluer leur complexité de calcul, sensibilité au bruit, et limites de performances par simulation, ainsi que l'influence des éléments de la structure de turbo décodage, sur l'évolution de ces performances. L'algorithme Max Log MAP, présente le meilleur compromis, performance complexité de calcul, apporté par la simplification de l'algorithme optimal d'origine MAP. Cette simplification a réduit considérablement la sensibilité du MAP au bruit. Cependant, la perte de performance causée par cette approximation, est compensée par une pondération adéquate de l'information extrinsèque dans la sortie douce (*Soft Output*) de chaque décodeur. La recherche continue dans plusieurs domaines, notamment dans l'espoir d'améliorer le compromis entre performance et complexité de calcul, ainsi que l'approche de la limite de Shannon, dans le cas d'un canal de communication sans fil, dispersive ou d'évanouissement.

5. Bibliographie

[1] C. Berrou, A. Galvieux and P. Thitimajshima " Near Shannon Limit Error-Correcting Coding and Decoding :(Turbo Codes),". *École Nat. Sup de Télécommunications De Bretagne France.1993.*

[2] C. Berrou, A. Glavieux, "Near Optimum Error Correcting Coding and Decoding : Turbo-Codes," *IEEE Trans. On Communication*, vol. 44, pp. 1261-1271, Oct. 1996.

[3] G. D. Forney, "The Viterbi Algorithm," *Proc. of the IEEE*, vol. 61, pp. 268-278, Mar 1973.

[4] A. S. Barbelescu and S. S. Pietrobon, "Interleaver Design for Turbo Codes," *Electron. Lett.*, vol. 30, no. 25, pp. 2107-2108, Dec. 1994.

[5] P.Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," *In. Proc. Globecom 94'*, pages 1298-1303, Dec. 1994.

[6] J. G. Proakis, M. Salehi, "Communication Systems Engineering," Prentice-Hall, 1994.

[7] 3GPP Technical specification, "Technical Specification Group Radio Acces Networks; UE Radio Transmission And Reception (FDD)," *3GPP TS 25.101 v5.0.0 (2001-09).*

[8] P. Robertson, E Villeburn, and P. Hoeher, "A Comparaison of Optimal and Sub-optimal MAP Decoding Algorithms Operating in the Log Domain," *in Proc. Int. Conf. Communications.* June 1995, pp. 1009-1013.

[9] P. Jung, "Comparaison of Turbo Code Decoders Applied to Short Frame Transmission Systems," *IEEE J. Select. Areas Comm.*, pp. 530-537, 1996.

[10] L.C. Perez, J. Seghers, and D. J. Costello, "A Distance Spectrum Interpretation of Turbo Codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698-1709, Nov. 1996.

[11] P. J. Woodard, L. Hanzo, "Comparative Study of Turbo Decoding Techniques : An Overview," IEEE Trans. On Vehicular Technology. Vol. 49, No. 6. Nov 2000.

[12] A. Worm, P. Hoener, N. Wehn, "Turbo-Decoding Without SNR Estimation," IEEE Communications Letter, vol.4 NO. 6 Jun. 2000.

[13] S. Benedetto and G. Montorsi, "Design of Parallel Concatenation Convolutional Codes," *IEEE Trans. Commun.*, vol. 44, pp. 591-600, May 1996.

[14] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi, and G. Nebe, "Interleaver Design for Turbo Codes" *Prod. AT&T Shannon Labs*, Nov. 2000.

[15] J. Vogt, A. Finger, "Improving the max-log-MAP turbo decoder," Electronics Letters. Vol. 36. No. 23. Nov. 2000.