

THÈSE

présentée à

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

Mention : *Electronique*

par

Laura CONDE CANENCIA

Turbocodes et modulations à grande efficacité spectrale

soutenue le 15 juin 2004 devant la commission d'examen :

Composition du Jury :

Président : Gilles BUREL
Rapporteurs : Marie-Laure BOUCHERET
Pierre DUHAMEL
Examineurs : Léon Claude CALVEZ
Catherine DOUILLARD
Alain GLAVIEUX
Invité : Jean-Claude CARLAH

Remerciements

Je tiens à remercier, tout d'abord, Catherine Douillard. Merci de ton suivi pendant ces années de thèse et de ton aide précieuse à la rédaction de ce rapport. Merci également à Claude Berrou, Michel Jézéquel et Sylvie Kerouédan.

Je tiens à saluer tous les membres du Département Electronique de l'ENST Bretagne. Merci de votre accueil et de votre particulière *buena onda*. Merci à Patricia Le Mignon, Bernard L'hostis, Gérard Graton, Patrick Adde, Pascale Ferry, Fabrice Seguin, Magali Le Gall et Amer Baghdadi.

Merci à Rémy Moreau et Didier Trichet (Ecole Polytechnique de l'Université de Nantes) pour leur soutien et leur aide pendant l'année 2003-2004.

Merci à mes collègues, thésards ou non, anciens et nouveaux, électroniques ou autres : Selma, Gwen, Mustapha, Laetitia, Horacio, Pablo, Jean-Luc, Charbel, Erwan, Humbertito, Rong, Ali, Eric, Javier, Raül, Richard, Esperanza, Carlos, Tatiana, Alberto, Ana, Daniel, José, Jérôme, Irène, Matthieu, Ilias, Mélody, Vivien...

Merci aux points clés du chemin : Beatriz, Angel, Sandrine, Maria Laura, Valéry, David, Clemens, JC, Cyrille et Gonzalo. Merci aux permanents : Marenita, Akhilesihno, Marcelito, Mena, Noemi, Roemi, RebecaT, Iosmary, Yaset, Nashiely, Mauricio et Cristina. Merci aux brestoises, danseurs ou volleyeuses : Guy, Hicham, Eric, Jean-Michel, Ana, Aude, AnneCat, Aurélie, Mélanie, Ingrid... Merci aux cub-rennais : Pascale, Ernesto et Yunexys. Salut Zaki, Steve, Holger et Yvelisse. Bonjour aux nouveaux : Lan, Eunice et Jerry.

Merci à la plus incroyable des belles petites filles : Noemita. Merci à mes parents de leur soutien inconditionnel, à mon frère, Miguel Angel, et à mes sœurs, Carmen et Pink. Un abrazo a toda mi familia de Madrid, Barcelona, Huelva y La Habana.

*New days are strange, is the world insane ?
Nations dropping bombs; chemical gasses fillin' lungs of little ones
With the ongoin' sufferin' as the youth die young
So ask yourself is the lovin' really gone; so I could ask myself really what is goin' wrong
In this world that we're livin' in people keep on givin' in, makin' wrong decisions...
Not respectin' each other, deny thy brother; a war is goin' on but the reason's undercover
The truth is kept secret, it's swept under the rug; if you never know the truth then you never know love
Where's the love, y'all (I don't know)
Where's the truth, y'all (I don't know)*

*I feel the weight of the world on my shoulder; as I'm getting older, y'all, people gets colder
Most of us only care about money makin'; selfishness got us followin' the wrong direction
Wrong information always shown by the media; negative images is the main criteria
Infecting the young minds faster than bacteria; kids wanna act like what they see in the cinema
Yo', whatever happened to the values of humanity; whatever happened to the fairness in equality
Instead of spreading love we're spreading animosity; lack of understanding leading lives away from unity
That's the reason why sometimes I'm feeling under; that's the reason why sometimes I'm feeling down
There's no wonder why sometimes I'm feeling under; gotta keep my faith alive 'til love is found...*

*But if you only have love to your own race, then you only leave space to discriminate
And to discriminate only generates hate; and when you hate then you're bound to get irate, yeah
Badness is what you demonstrate and that's exactly how anger works and operates...
Take control of your mind and meditate; let your soul gravitate to the love...*

*@ The Black Eyed Peas "Where is the love", Elephunk, A&M Records, June 2003
And... we are not a minority! Hasta siempre, comandante.*

*I believe that harmonies are colors
Everytime I paint, it sharpens my harmony
Yesterday I tried to paint you
But the colors weren't beautiful enough
Your love goes beyond what I can say
Love,
Beyoncé*

@ "Beyoncé Interlude", Dangerously In Love, Sony, June 2003

*Je dédie cette thèse à mes parents,
Carmina et Antonio.*

Table des matières

| | |
|--|----------|
| Remerciements | i |
| Introduction | 1 |
| 1 Quelques notions de communications numériques | 5 |
| 1.1 Chaîne de transmission numérique | 6 |
| 1.2 Quelques notions de codage de canal | 7 |
| 1.2.1 Le codage correcteur d'erreurs | 8 |
| 1.2.2 Les codes convolutifs | 9 |
| 1.2.3 Les codes convolutifs systématiques récurrents | 12 |
| 1.3 Quelques notions sur les signaux pour la transmission numérique sur onde porteuse | 12 |
| 1.3.1 Transmission d'un signal modulé sur canal à bande limitée | 13 |
| 1.3.2 Enveloppe complexe | 13 |
| 1.3.3 Composantes en phase et en quadrature | 13 |
| 1.3.4 Représentation des signaux dans le plan complexe. Notion de constellation | 14 |
| 1.3.5 Énergie d'un signal et énergie d'une constellation | 16 |
| 1.3.6 Réception en présence de bruit et régions de décision | 16 |
| 1.4 Les modulations numériques étudiées | 17 |
| 1.4.1 Définitions | 18 |
| 1.4.2 Modulations par déplacement de phase (MDP, en anglais <i>PSK</i> , <i>Phase Shift Keying</i>) | 19 |
| 1.4.3 Modulation d'amplitude sur deux porteuses en quadrature (MAQ, en anglais <i>QAM</i> , <i>Quadrature Amplitude Modulation</i>) | 20 |
| 1.4.4 Modulation par déplacement d'amplitude et de phase (MDAP, en anglais <i>APSK</i> , <i>Amplitude Phase Shift Keying</i>) | 22 |
| 1.5 Modélisation du canal de transmission | 23 |
| 1.5.1 Canal à bruit blanc additif gaussien | 24 |
| 1.5.2 Les canaux à évanouissements | 25 |

| | | |
|----------|--|-----------|
| 1.5.3 | Notion de diversité | 26 |
| 1.5.4 | Caractérisation statistique du canal de Rayleigh non sélectif en fréquence à évanouissements lents | 27 |
| 1.6 | Conclusion | 28 |
| 2 | Turbocodes et transmissions à grande efficacité spectrale | 29 |
| 2.1 | Les turbocodes | 30 |
| 2.1.1 | Les turbocodes convolutifs | 30 |
| 2.1.2 | Les turbocodes m -binaires | 36 |
| 2.2 | Les modulations codées | 37 |
| 2.2.1 | Distance libre d'une modulation codée | 38 |
| 2.2.2 | Longueur effective et distance produit d'une modulation codée | 39 |
| 2.2.3 | Les Modulations Codées en Treillis (MCT) | 40 |
| 2.2.4 | Les modulations codées entrelacées au niveau bit | 42 |
| 2.3 | Les modulations turbocodées | 43 |
| 2.3.1 | Etat de l'art | 43 |
| 2.3.2 | Principe de l'approche pragmatique | 44 |
| 2.3.3 | Principe des turbo-modulations codées en treillis | 45 |
| 2.3.4 | Comparaison de performances | 47 |
| 2.4 | Conclusion | 49 |
| 3 | Etude de modulations turbocodées pragmatiques à base de codes duo-binaires | 51 |
| 3.1 | Description et performances des turbocodes convolutifs duo-binaires étudiés | 52 |
| 3.1.1 | Structure des turbocodeurs | 52 |
| 3.1.2 | Structure du turbo-décodeur | 55 |
| 3.1.3 | Performances des turbocodes à 8 et 16 états sur canal gaussien | 56 |
| 3.2 | Modulation turbocodée pragmatique à base de codes duo-binaires : cas général | 60 |
| 3.2.1 | Schéma d'émission | 60 |
| 3.2.2 | Schéma de réception | 61 |
| 3.2.3 | Quelques précisions sur les conditions de simulation | 62 |
| 3.3 | Association pragmatique d'un TCC duo-binaire et d'une MDP-8 | 63 |
| 3.3.1 | Construction des symboles de modulation | 63 |
| 3.3.2 | Calcul de l'estimation pondérée des bits s_2 , s_1 et s_0 | 64 |
| 3.3.3 | Comparaison des performances des codes à 8 et 16 états | 64 |
| 3.4 | Association pragmatique d'un TCC duo-binaire et d'une MAQ-16 | 66 |
| 3.4.1 | Codage binaire à signal de la modulation | 66 |
| 3.4.2 | Stratégies de construction des symboles | 66 |
| 3.4.3 | Calcul de l'estimation pondérée des bits s_3 , s_2 , s_1 et s_0 | 67 |

| | | |
|----------|---|------------|
| 3.4.4 | Résultats de simulation | 68 |
| 3.5 | Etude de la construction de symboles dans l'association pragmatique d'un TCC et d'une modulation MAQ-64 | 74 |
| 3.5.1 | Codage binaire à signal de la modulation | 74 |
| 3.5.2 | Stratégies de construction de symboles | 75 |
| 3.5.3 | Calcul des estimations pondérées de bits s_5, s_4, s_3, s_2, s_1 et s_0 | 76 |
| 3.5.4 | Résultats de simulation | 77 |
| 3.6 | Association pragmatique d'un TCC duo-binaire et d'une MDAP-16 pour des communications satellitaires | 83 |
| 3.7 | Conclusion | 84 |
| 4 | Estimation de la performance asymptotique des modulations turbocodées | 87 |
| 4.1 | Performance d'un code correcteur d'erreurs et distance minimale | 88 |
| 4.2 | La méthode de l'impulsion d'erreur | 89 |
| 4.2.1 | Principe de la méthode | 89 |
| 4.2.2 | Algorithme de détermination de la distance minimale | 90 |
| 4.2.3 | Multiplicité de d_{\min} | 90 |
| 4.2.4 | Application pratique de la MIE aux turbocodes | 90 |
| 4.2.5 | Conclusion | 91 |
| 4.3 | Application de la méthode de l'impulsion d'erreur à l'estimation de la performance asymptotique d'une modulation turbocodée pragmatique sur canal gaussien | 92 |
| 4.3.1 | Schéma de transmission | 92 |
| 4.3.2 | Détermination de la performance asymptotique d'une modulation codée : méthode 1 | 92 |
| 4.3.3 | Détermination de la performance asymptotique d'une modulation turbocodée : méthode 2 | 94 |
| 4.3.4 | Comparaison des performances estimées et simulées | 96 |
| 4.4 | Application de la méthode de l'impulsion d'erreur à l'estimation de la performance asymptotique d'une modulation turbocodée pragmatique sur canal à évanouissements | 99 |
| 4.4.1 | Schéma de transmission | 100 |
| 4.4.2 | Notations | 101 |
| 4.4.3 | Bornes de la probabilité d'erreur par paire | 101 |
| 4.4.4 | Probabilité d'événement d'erreur | 102 |
| 4.4.5 | Application de la MIE | 102 |
| 4.4.6 | Comparaison des performances estimées et simulées | 103 |
| 4.5 | Conclusion | 103 |
| | Conclusion | 107 |

| | |
|---|------------|
| Publications | 111 |
| A Les codes convolutifs circulaires | 113 |
| A.1 Le problème de la fermeture du treillis pour les turbocodes | 113 |
| A.2 Les codes convolutifs récurrents circulaires | 114 |
| A.3 Etats de circulation pour les turbocodes utilisés dans notre étude | 115 |
| B Décodage des turbocodes m-binaires | 117 |
| B.1 Notations | 117 |
| B.2 Décodage selon le critère du Maximum A Posteriori (MAP) | 117 |
| B.2.1 Calcul des vraisemblances avant et arrière | 119 |
| B.2.2 Calcul des vraisemblances de branches | 119 |
| B.2.3 Normalisation des vraisemblances avant et arrière | 120 |
| B.3 Décodeur MAP pour le décodage itératif d'un turbocode m -binaire | 120 |
| B.4 L'algorithme simplifié Max-Log-MAP | 122 |
| B.4.1 Calcul des métriques de branches | 123 |
| B.4.2 Calcul des métriques avant et arrière | 123 |
| B.5 Décodeur Max-Log-MAP ou dual-Viterbi pour le décodage itératif d'un tur- bocode m -binaire | 124 |
| B.5.1 Extraction de l'information extrinsèque | 125 |
| C Analyse des probabilités d'erreur binaire dans les modulations étudiées | 127 |
| C.1 Probabilité d'erreur binaire dans une MDP-8 avec <i>mapping</i> de Gray | 127 |
| C.2 Probabilité d'erreur binaire dans une MAQ-16 avec <i>mapping</i> de Gray | 130 |
| C.3 Cas de la MAQ-64 avec <i>mapping</i> de Gray | 131 |
| Bibliographie | 133 |

Introduction

DEPUIS que l'homme est homme, il a eu besoin de communiquer et depuis que Morse a inventé le télégraphe en 1844, les communications à distance ont induit de profondes transformations dans notre société. En effet, communication et développement ont en quelque sorte avancé la main dans la main dans les deux derniers siècles.

Tout au long de la courte histoire des télécommunications, les progrès de ces dernières ont été conditionnés par les avancées dans d'autres domaines tels que la physique, les mathématiques et, plus récemment, l'électronique. Au XIX^{ème} siècle, les progrès en physique ont donné naissance aux télécommunications. Maxwell introduit les lois fondamentales de l'électromagnétisme en 1870, le téléphone étant ensuite inventé par Bell en 1876. En 1888 Hertz publie ses études sur les ondes électromagnétiques ce qui permet à Marconi d'introduire le télégraphe sans fil puis d'établir en 1909 la première liaison hertzienne entre une station à terre et un remorqueur situé à 29 kilomètres de la côte. Au XX^{ème} siècle, Shannon établit en 1948 une théorie mathématique permettant de formaliser et de mieux comprendre les concepts liés à la transmission d'information. Ces travaux donnent naissance à une nouvelle discipline de recherche dans les communications : le codage correcteur d'erreurs. Dans les années qui suivent, de nombreux scientifiques travaillent à la recherche du code optimal. Plusieurs codes sont alors inventés mais, à l'époque, les moyens technologiques limités ne permettaient pas leur mise en œuvre pratique. Depuis l'invention du transistor en 1948 et l'apparition des premiers circuits intégrés en 1958, les constantes avancées en micro-électronique ont permis d'introduire progressivement ces codes dans les chaînes de transmission numérique.

Parmi les avancées importantes dans le domaine du codage du canal, la concaténation de codes simples est à signaler. Cette technique permet d'obtenir des performances équivalentes à celles de codes très complexes avec un décodage nettement plus simple. Concrètement, la concaténation d'un code convolutif et d'un code Reed-Solomon a été adoptée dans le domaine des communications spatiales.

En mai 1991, Berrou et Glavieux introduisent les turbocodes, construits à partir d'une concaténation parallèle de deux codes convolutifs systématiques récurrents. Leurs performances étaient sans précédent dans l'histoire du codage de canal et, depuis, leur utilisation est préconisée dans de plus en plus de standards de télécommunications. Citons par exemple le standard de codage pour la télémétrie (CCSDS) dans les communications spatiales ainsi que les normes DVB-RCS/RCT pour la télévision numérique et UMTS pour la téléphonie mobile.

Le spectre de fréquences est une ressource limitée et, en raison de la coexistence de nombreuses applications radio (téléphonie mobile, réseaux sans fils, diffusion hertzienne, communications par satellite...), chaque système dispose de ressources spectrales très restreintes. Pourtant, dans certaines applications, la gamme de services offerts ne cesse de s'étoffer tandis que le nombre d'utilisateurs ne cesse d'augmenter. L'un des objectifs les plus importants dans la conception de tels systèmes est alors l'exploitation optimale du spectre disponible. Les opérateurs et constructeurs doivent chercher à maximiser l'efficacité spectrale des systèmes,

définie comme le débit utile de transmission par unité de bande occupée.

Ainsi, dans les deux dernières décades, les recherches autour des techniques de modulations codées à grande efficacité spectrale ont bénéficié de progrès importants, permettant de faire face à la demande. Ces techniques deviennent, par conséquent, incontournables pour des canaux limités à la fois en puissance et en bande spectrale. Le canal radio-mobile en est un exemple car limité en spectre à cause des raisons déjà décrites et en puissance, notamment dans les villes où les émetteurs sont installés à faible distance, voire quasiment au contact, de la population. Un autre exemple est le canal de communication par satellite, nécessitant d'un grand débit de transmission et limité en puissance par la taille physique du satellite.

Cette thèse s'inscrit alors dans un contexte d'étude de systèmes de transmission performants vis-à-vis du bruit et efficaces du point de vue de l'occupation du spectre de fréquences. Nous nous sommes ainsi tout naturellement intéressés aux turbocodes et à la problématique de leur association aux modulations à grande efficacité spectrale. Ces associations sont également nommées modulations turbocodées.

Ce mémoire de thèse est constitué de quatre chapitres, organisés comme suit :

Le premier chapitre présente quelques généralités sur les communications numériques. Après avoir décrit la chaîne typique de communication numérique, nous abordons plus en détail le codage de canal et les modulations numériques. Les principes des codes correcteurs d'erreurs et, plus particulièrement, ceux du codage convolutif sont présentés. De même, l'ensemble de concepts liés aux modulations numériques ainsi que les modulations utilisées dans notre étude sont décrits. Finalement nous nous intéressons aux deux modèles de canal de transmission considérés dans cette thèse : le canal à bruit additif gaussien et le canal à évanouissements. Nous présentons une classification de ces derniers et introduisons la notion de diversité.

Le deuxième chapitre introduit les turbocodes et les modulations codées. Dans un premier temps, les principes du turbocodage/décodage sont présentés ainsi que quelques éléments de performance des turbocodes. Une nouvelle famille de turbocodes, à savoir les turbocodes m -binaires, est présentée. Les notions relatives aux modulations codées sont ensuite décrites. Nous introduisons les modulations codées en treillis d'Ungerboeck et les modulations codées entrelacées au niveau bit de Zehavi. Enfin, les modulations turbocodées sont abordées. Nous présentons l'association dite pragmatique, introduite par Le Goff, pour ensuite décrire les techniques de turbo-modulation codée en treillis. Une comparaison en termes de performance et de flexibilité donne les raisons pour lesquelles nous avons retenu les modulations turbocodées pragmatiques dans notre étude.

Le troisième chapitre aborde l'étude de plusieurs associations concrètes d'un turbocode et d'une modulation numérique selon les principes de l'approche pragmatique. Deux turbocodes duo-binaires sont considérés dans l'étude : celui du standard DVB-RCS et une extension de celui-ci à 16 états, dénommée TURBO2000. Les modulations MDP-8, MAQ-16, MAQ-64 et MDAP-16 ont été considérées. Pour les modulations de type MAQ, nous avons étudié et comparé différentes stratégies de construction de symboles.

Le quatrième chapitre est consacré à l'estimation de la performance asymptotique des modulations turbocodées aussi bien sur canal gaussien que sur canal à évanouissements de Rayleigh. La méthode de l'impulsion d'erreur, récemment introduite, permet d'obtenir dans un temps record une estimation de la distance minimale d'un turbocode ainsi que de sa multiplicité, sous certaines hypothèses. L'analyse développée dans ce chapitre permet d'utiliser les informations fournies par cette méthode pour estimer les performances à fort rapport signal

à bruit des modulations turbocodées pragmatiques.

La conclusion finale fait un bilan des résultats obtenus et donne les perspectives de recherche nées du travail effectué dans cette thèse.

CHAPITRE

1

Quelques notions de communications numériques

Ce premier chapitre est consacré à la présentation des différentes fonctions de communications numériques qui ont été étudiées dans le cadre de cette thèse. Dans un premier temps, nous décrivons le rôle des principaux éléments d'une chaîne de transmission numérique. Nous détaillons ensuite les fonctions étudiées au cours de la thèse : le codage de canal et les modulations numériques. Concernant les codes correcteurs d'erreurs, nous présentons plus particulièrement la famille des codes convolutifs, qui sont à la base du schéma de codage étudié. Les différentes modulations considérées sont également décrites en détail. Finalement, nous terminons ce chapitre par une présentation des modèles de transmission examinés dans notre étude : le canal à bruit blanc additif gaussien et les canaux à évanouissements lents non-sélectifs en fréquence.

1.1 Chaîne de transmission numérique

Le schéma de principe d'une chaîne de transmission numérique est présenté sur la figure 1.1. La source d'information peut-être soit *analogique* (*i.e.* signal continu dans le temps à valeurs continues) soit *numérique* (*i.e.* signal discret dans le temps et avec un nombre fini de caractères de sortie). Si le signal est analogique, l'information sera numérisée en échantillonnant le message analogique puis en quantifiant les échantillons obtenus. Dans un système de communications numériques, les messages produits par la source sont transformés en une séquence d'éléments binaires ou *bits*. Idéalement, le message de la source devrait être représenté par un nombre minimal de bits. Si tel n'est pas le cas, on cherche à enlever la redondance du message original afin de réduire la quantité de données à transmettre en conservant toute l'information utile. Ce processus est nommé *codage de source* ou *compression de données*. Pour la suite, la source codée est considérée idéale, c'est-à-dire que les éléments binaires sont indépendants et identiquement distribués sur l'alphabet binaire.

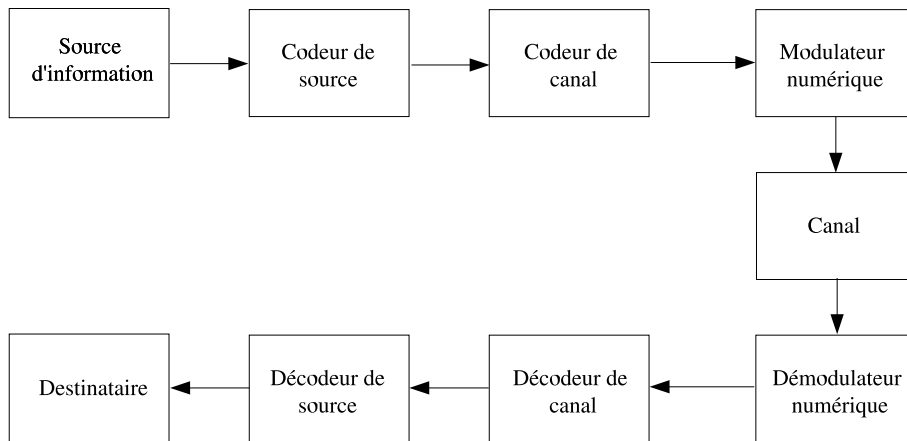


Fig. 1.1 — Modélisation d'une chaîne de transmission numérique

La séquence d'éléments binaires provenant du codeur de source, aussi appelée *séquence d'information*, est appliquée à l'entrée du *codeur de canal*. Le rôle de celui-ci est d'ajouter des éléments binaires de redondance afin que le décodeur correspondant puisse retrouver la séquence d'information émise en dépit des dégradations causées au signal transmis. Ces dégradations sont dues à l'effet du bruit et des interférences existant dans le canal de transmission. Le *codage de canal* permet d'améliorer la qualité de la transmission, au prix de l'augmentation de la complexité du système et de la réduction de son débit.

Pour que le message issu du codeur de canal soit transmis, il est nécessaire de le transformer en un signal électrique. C'est le rôle du *modulateur numérique* qui sert d'*interface* avec le canal de transmission. Le choix du type de signaux dépend des propriétés physiques du milieu de transmission, le signal sera ainsi en général transmis soit en bande de base soit sur fréquence porteuse.

Le *canal de transmission* peut être considéré suivant plusieurs modèles différents. Dans ce document nous considérons que le canal inclut le milieu physique de propagation ainsi que les organes d'émission (transposition de fréquence, filtrage de mise en forme du signal...) et de réception (détection, amplification, filtrage, échantillonnage). Nous considérons également que la détection est cohérente (*i.e.* synchronisation de la phase et récupération de la porteuse parfaites).

Une fois le signal reçu et transformé en une séquence d'information numérique par le *démodulateur numérique*, la séquence est appliquée à l'entrée du *décodeur de canal*. Son rôle est de reconstruire l'information originale en profitant de la connaissance du code utilisé en émission et de la redondance contenue dans les données reçues.

Une mesure de la qualité du système de transmission est le taux d'erreurs binaires, en anglais *Bit Error Rate* (BER). Ce paramètre est défini comme le rapport du nombre de bits erronés reçus au nombre de bits transmis. Il est fonction des types de signaux utilisés pour la transmission de l'information, de la puissance de l'émetteur, des caractéristiques du canal, de la méthode de démodulation, des caractéristiques du code et de l'algorithme de décodage utilisé. Nous utiliserons également le taux d'erreurs de trames, en anglais *Frame Error Rate* (FER), qui est défini comme le rapport du nombre de blocs erronés reçus au nombre de blocs transmis. La qualité de la transmission est d'autant meilleure que ces taux d'erreurs sont faibles. Selon la nature du message et le domaine d'application, un taux d'erreurs à ne pas dépasser est fixé de façon à assurer le destinataire d'une qualité minimale du message reçu.

Le *décodeur de source* effectue l'opération inverse du codeur de source pour reconstituer la séquence d'information originale. Ensuite, si le message est de nature analogique, une conversion numérique-analogique est réalisée. Évidemment, le signal en sortie du décodeur de source est seulement une estimation du signal d'origine en raison des erreurs possibles du décodage de canal et de la distorsion éventuelle introduite par le codeur de source. Les différences entre le signal original et le signal reconstruit sont une mesure de la distorsion introduite par le système de communication numérique.

1.2 Quelques notions de codage de canal

Puisque cette thèse aborde la problématique du codage de canal pour des communications à grande efficacité spectrale, l'opération de codage de canal dans la chaîne de transmission nécessite une description détaillée.

Dans cette section, nous introduisons quelques définitions relatives aux codes correcteurs d'erreurs, puis une description détaillée des codes directement liés au sujet de cette étude, à savoir les codes convolutifs, est donnée.

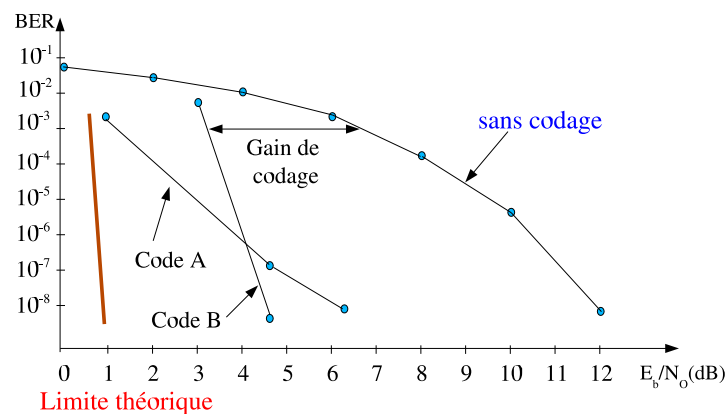


Fig. 1.2 — Représentation graphique du gain de codage et du comportement de deux codes correcteurs d'erreurs type

1.2.1 Le codage correcteur d'erreurs

Le principe d'un *code correcteur d'erreurs* est d'ajouter une certaine quantité de bits de redondance à une séquence de bits d'information afin de pouvoir retrouver en réception la séquence d'information transmise. Nous nous limitons ici au codage linéaire de séquences binaires. Les *codes binaires* ne considèrent que les valeurs de l'alphabet binaire, $B = \{0, 1\}$. D'autre part, un *code* est dit *linéaire* si toute combinaison linéaire de mots de code est encore un mot de code.

La théorie de l'information énoncée par Shannon en 1948 [68] montre qu'il existe des codes correcteurs d'erreurs qui rendent possible la transmission d'information avec une probabilité d'erreur arbitrairement petite. Shannon ne donne toutefois pas de solution pratique pour réaliser ces codes. Dans les années qui suivent, la théorie du codage de canal connaît des avancées importantes. Les codes en bloc sont introduits (par exemple, les codes de Hamming [48] et les codes produits [41]), ainsi que les codes convolutifs [42].

Nous allons introduire ci-après quelques définitions liées aux codes correcteurs d'erreurs avant d'aborder les codes convolutifs.

Gain de codage

Les performances d'un code correcteur d'erreurs sont déterminées par une courbe de taux d'erreurs (BER ou FER) en fonction du rapport signal à bruit de la transmission mesuré en dB. Ce rapport signal à bruit s'exprime comme E_b/N_0 , où E_b est l'énergie transmise par élément binaire d'information utile et N_0 la densité spectrale de puissance monolatérale de bruit gaussien (cf. Section 1.5.1).

Si l'on se place à une certaine valeur de taux d'erreurs, le *gain de codage* est déterminé par l'écart existant entre la courbe sans codage et la courbe avec codage. A titre d'illustration, sur la figure 1.2 nous avons représenté le gain de codage du code B à un BER = 10^{-3} qui est d'environ 3,5 dB.

Distance minimale de Hamming

Le *poids de Hamming* d'un mot de code est le nombre d'éléments binaires non nuls qu'il contient. Par exemple, si l'on considère un code binaire C dont la longueur des mots de code est 8 bits, le poids de Hamming de la séquence $\mathbf{c}_1 = (00101001)$ est égal à 3. Nous adopterons la notation $w_H(\mathbf{c}_1) = 3$.

La *distance de Hamming* entre deux mots de code \mathbf{c}_i et \mathbf{c}_j est définie comme le nombre de bits différents entre \mathbf{c}_i et \mathbf{c}_j . Autrement dit, elle est égale au poids de Hamming du mot de code $\mathbf{c}_i \oplus \mathbf{c}_j$, où \oplus représente la somme *modulo 2*, soit $d_H(\mathbf{c}_i, \mathbf{c}_j) = w_H(\mathbf{c}_i \oplus \mathbf{c}_j)$. Prenant à nouveau le code C comme exemple, la distance de Hamming entre les séquences $\mathbf{c}_1 = (00101001)$ et $\mathbf{c}_2 = (10101000)$ est 2, et on le représente : $d_H(\mathbf{c}_1, \mathbf{c}_2) = 2$.

La *distance minimale de Hamming* d'un code, notée d_{\min} , est définie comme la distance de Hamming minimale entre l'ensemble de ses mots de code. Si le code est linéaire, elle peut également être calculée comme le poids minimal de ses mots non nuls :

$$d_{\min} = \min_{\mathbf{c}_i, \mathbf{c}_j \in C; \mathbf{c}_i \neq \mathbf{c}_j} d_H(\mathbf{c}_i, \mathbf{c}_j) = \min_{\mathbf{c}_i \in C; \mathbf{c}_i \neq \mathbf{0}} w_H(\mathbf{c}_i) \quad (1.1)$$

Convergence et performance asymptotique

Deux paramètres sont à considérer lors de l'analyse des performances d'un code correcteur d'erreurs :

- **Le seuil de convergence** : il s'agit du rapport signal à bruit à partir duquel le système codé devient plus performant que le système de transmission non-codé. Lorsque ce seuil est faible, les performances du système à des niveaux de bruit fort et moyen sont proches de la limite théorique et on peut dire que le code présente une *bonne convergence*.
- **La performance asymptotique** : le comportement à très faible niveau de bruit du système de transmission codé est dicté par la distance minimale de Hamming du code et son rendement. Le gain asymptotique, noté G_a , est donné par l'expression suivante :

$$G_a \approx 10 \log(Rd_{\min}) \quad (1.2)$$

où R est le rendement du code. Une faible valeur de d_{\min} entraîne un fort changement de pente (on parle de *flattening*) dans la courbe de taux d'erreurs dû à un faible gain asymptotique.

Sur la figure 1.2 nous avons présenté les courbes de deux codes dont les comportements sont différents. Le code A présente un bon seuil de convergence mais un faible gain asymptotique. En revanche, le code B présente un bon comportement asymptotique mais un seuil de convergence plus élevé. La recherche d'un bon code dépend du taux d'erreur cible de l'application. En effet, pour les taux d'erreurs moyens ou élevés, il est préférable de privilégier le seuil de convergence au détriment de la distance minimale du code. Pour les faibles taux d'erreurs, mieux vaut privilégier la distance minimale.

1.2.2 Les codes convolutifs

Depuis leur introduction en 1955 par Elias [42], les codes convolutifs ont acquis une importance équivalente à celle des codes en bloc en raison de leur simplicité de mise en œuvre et de leurs bonnes performances de correction.

Le principe du codage convolutif est illustré par la figure 1.3. Chaque bloc de p bits en sortie du codeur dépend non seulement du bloc de q bits présent à son entrée mais aussi des ν blocs présents précédemment. Les codes convolutifs introduisent par conséquent un effet de mémoire d'ordre ν . La grandeur $K = \nu + 1$ est appelée *longueur de contrainte* du code.

Un codeur convolutif est alors constitué d'un registre à ν étages qui mémorise les ν blocs de q bits et d'une logique combinatoire qui calcule les blocs de p bits fournis par le codeur. La quantité $R = q/p$ est le rendement du code. Le code est dit *systématique* si les q bits d'information présents à l'entrée du codeur sont émis (*i.e.* ils se retrouvent explicitement dans le bloc de p bits).

La figure 1.4 montre un exemple de codeur convolutif de rendement $R = 1/2$ et de longueur de contrainte $K = 3$. L'entrée est constituée par des blocs de $q = 1$ bit (code *binnaire*) et la sortie par des blocs de $p = 2$ bits. Les sorties du codeur c_k^1 et c_k^2 sont définies par le produit de convolution entre la suite binaire présente à l'entrée du codeur et la réponse du codeur, définie par ses séquences génératrices g_1 et g_2 :

$$c_k^i = \sum_{j=0}^2 g_{ij} d_{k-j} \quad g_{ij} \in (0, 1)$$

avec $g_1 = [1, 0, 1]$ et $g_2 = [1, 1, 1]$. La notation octale est souvent utilisée pour représenter les séquences génératrices, ainsi $g_1 = 5_{(octal)}$ et $g_2 = 7_{(octal)}$. Un codeur convolutif est alors défini

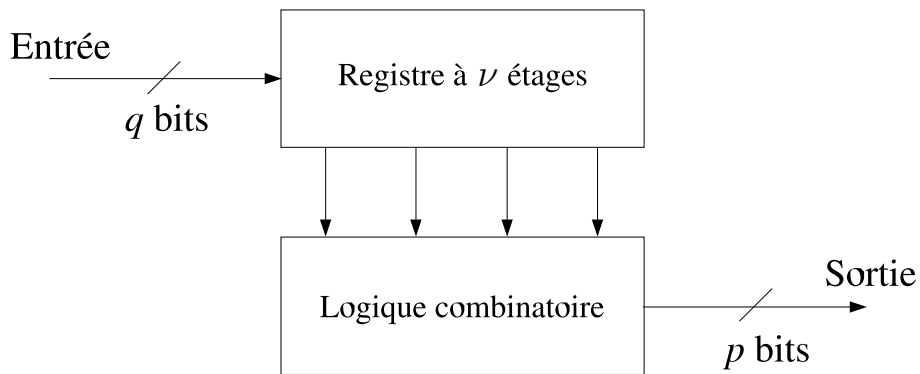


Fig. 1.3 — Principe d'un codeur convolutif classique (non récursif)

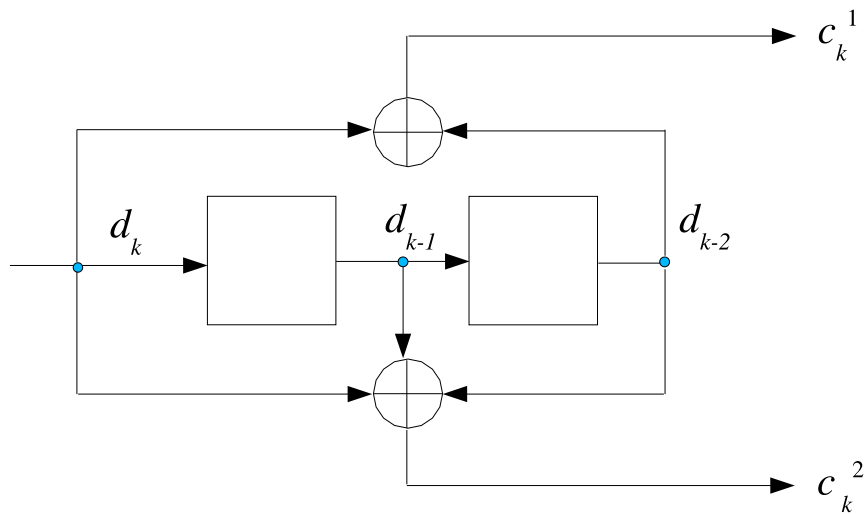


Fig. 1.4 — Exemple de codeur convolutif non systématique ($R=1/2$; $\nu=2$)

à partir de ses polynômes générateurs exprimés en fonction de la variable D (délai). Pour le codeur de la figure 1.4, ces polynômes sont :

$$G_1(D) = g_{10} + g_{11}D + g_{12}D^2 = 1 + D^2$$

$$G_2(D) = g_{20} + g_{21}D + g_{22}D^2 = 1 + D + D^2$$

Représentation des codes convolutifs

Un codeur convolutif peut être représenté sous la forme d'un automate à 2^ν états. L'état de cet automate est déterminé par les ν derniers symboles binaires entrés dans le codeur. Trois types de diagrammes sont couramment utilisés pour représenter le fonctionnement du codeur : le *diagramme d'états*, le *diagramme en arbre* et le *diagramme en treillis*. Nous allons détailler le dernier, qui permet la représentation de toutes les transitions possibles entre les états en fonction du temps.

Le nombre d'états du treillis est donné par 2^ν . L'état du codeur à l'instant i , noté S_i , est la représentation décimale de la mémoire du codeur à l'instant i . Les branches tracées en trait plein (resp. en pointillés) correspondent à la présence d'un bit égal à 1 (resp. 0) à l'entrée du codeur. La séquence codée est fournie par l'ensemble des étiquettes de branche.

Nous utilisons à nouveau le codeur de la figure 1.4 pour donner un exemple de représentation en treillis (figure 1.5). On suppose que l'état initial du codeur est $S_0 = 00$. Après $\nu + 1 = 3$ décalages, le motif du treillis se répète. De chaque noeud partent 2^q branches et en chaque noeud convergent 2^q branches (ici $q = 1$ donc deux branches partent de chaque noeud et deux convergent en chaque noeud).

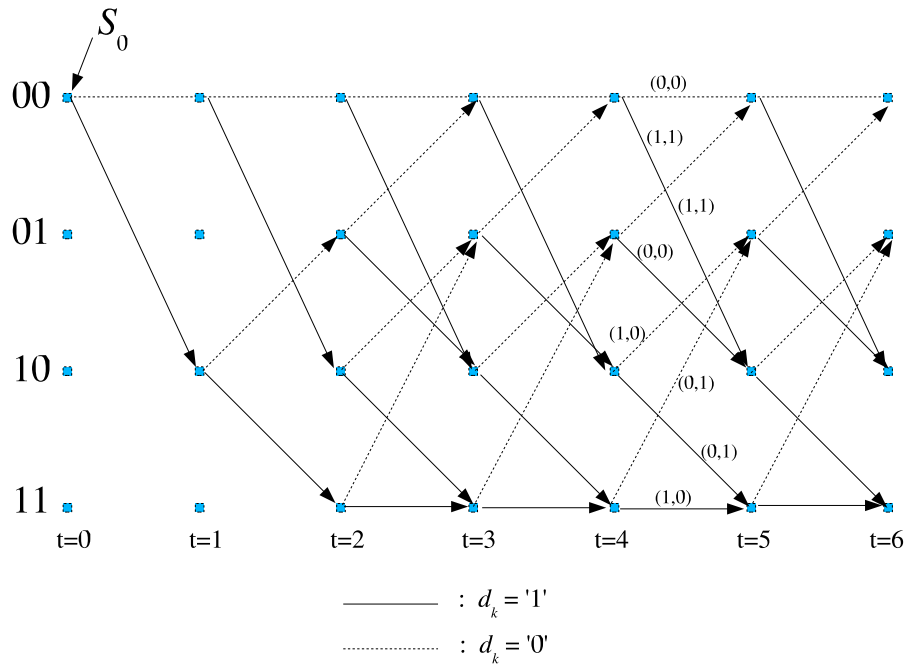


Fig. 1.5 — Représentation du code convolutif de la figure 1.4 sous la forme d'un treillis

Les codes convolutifs poinçonnés

La technique de *poinçonnage* permet d'élaborer des codes de rendements divers à partir d'un code de base par la non transmission de certains bits issus du codeur. Les codes poinçonnés classiques sont, par exemple, obtenus par effacement périodique de certains bits c_k^1 et c_k^2 suivant un *motif de poinçonnage* approprié. La figure 1.6 montre un exemple de poinçonnage effectué sur un codeur convolutif de rendement $1/2$. Si p est la longueur de la séquence émise par le codeur et f est le nombre de bits effacés, le rendement du code poinçonné obtenu est $R = p/(2p - f)$. Le *motif de poinçonnage* indique la position des symboles à effacer. Les codes ainsi construits présentent des performances très proches de celles atteintes avec des codes non poinçonnés de même rendement.

Décodage des codes convolutifs

Le décodage optimal d'un code convolutif, appelé décodage à Vraisemblance Maximale (en anglais, *Maximum-Likelihood decoding*) consiste à trouver dans le treillis la séquence binaire la plus proche de la séquence reçue. Le nombre de séquences possibles étant généralement très important, l'application de cette règle de décodage par une recherche exhaustive est la plupart du temps d'une complexité prohibitive. Des algorithmes à complexité plus réduite ont été développés, celui de Viterbi [87] [43] étant le plus célèbre.

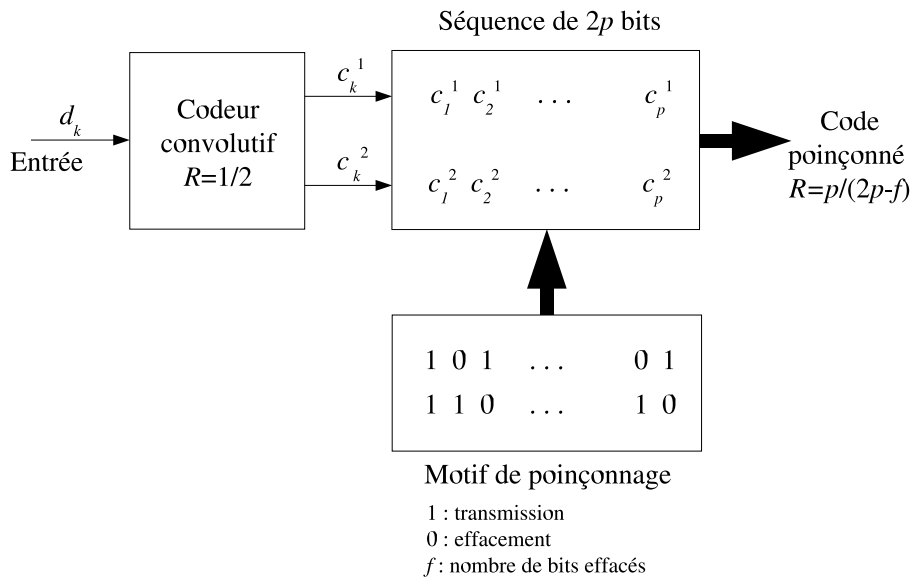


Fig. 1.6 — Poinçonnage d'un code convolutif de rendement $1/2$

1.2.3 Les codes convolutifs systématiques récursifs

Une analyse comparée des courbes de taux d'erreurs de codes Convolutifs Systématiques (CS) et de codes Convolutifs Non Systématiques (CNS) de même ν montre que les codes CNS présentent des probabilités d'erreur plus faibles que les codes CS, excepté pour les très faibles rapport signal à bruit. Dans son travail de thèse, Thitimajshima [83] s'est intéressé à une nouvelle classe de codes : les codes Convolutifs Systématiques Récursifs (CSR). Un code convolutif est dit *récursif* si une ou plusieurs de ses sorties sont réintroduites à l'entrée pour être à nouveau codées. La figure 1.7 présente un exemple de schéma d'un codeur convolutif récursif. Dans cet exemple les sorties du premier et du dernier élément de mémoire sont réinjectées à l'entrée du codeur.

Les codes CSR ont l'avantage de présenter des distances minimales équivalentes aux codes CNS de même ν tout en conservant, à faible rapport signal à bruit, une probabilité d'erreur plus faible. De plus, ils peuvent aisément être poinçonnés afin d'atteindre les rendements de codage souhaités sans que cela n'affecte leurs bonnes propriétés de distance [83].

Dans le cadre de cette thèse, nous nous intéressons aux codes CSR et, plus précisément, à la concaténation parallèle de tels codes, encore appelée *turbocodes*. Les travaux de thèse de Thitimajshima ont montré l'intérêt de l'utilisation des codes CSR pour la construction des turbocodes en raison de leur bon comportement aussi bien à fort qu'à faible rapport signal à bruit. Une présentation des turbocodes est donnée dans le Chapitre 2 de ce document.

1.3 Quelques notions sur les signaux pour la transmission numérique sur onde porteuse

Certains concepts fondamentaux de la théorie de la communication sur canal à bande étroite sont ici présentés. Il s'agit de fixer les notations et de donner les définitions permettant de conduire à la représentation d'une modulation numérique sous forme de constellation.

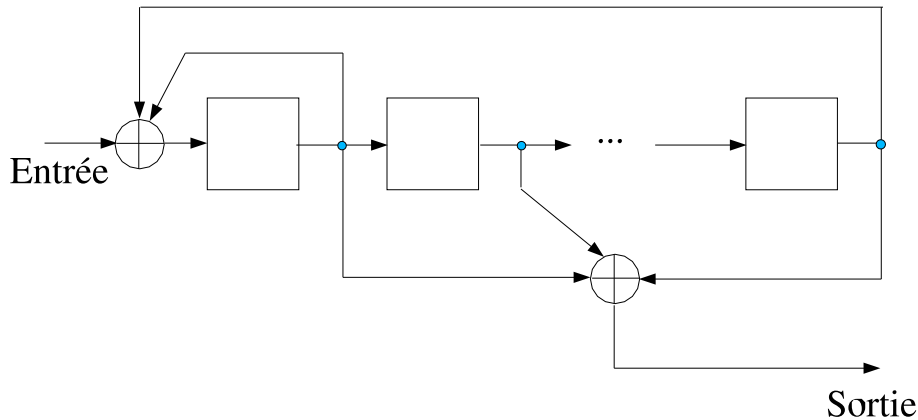


Fig. 1.7 — Exemple de schéma de codeur convolutif récursif

1.3.1 Transmission d'un signal modulé sur canal à bande limitée

Considérons un signal de la forme :

$$s(t) = r(t) \cdot \cos[2\pi f_0 t + \varphi(t)] \quad (1.3)$$

Cette expression représente une porteuse à la fréquence f_0 modulée en amplitude par un signal $r(t)$ et en phase par un signal $\varphi(t)$. Ce signal est dite à *bande étroite* si son spectre est limité à une bande $[-b_e, b_e]$ telle que sa largeur spectrale $B = 2b_e$ est très petite par rapport à f_0 . Dans ce type de signaux, $r(t)$ et $\varphi(t)$ varient beaucoup plus lentement que la sinusoïde $\cos(2\pi f_0 t)$. Le signal $s(t)$ a donc un comportement semblable à celui d'une sinusoïde dont l'amplitude et la phase varient lentement dans le temps. Par exemple, si $f_0 = 100$ MHz et $B = 10$ kHz, en 10 cycles de signal ($\Delta t = 0,1 \mu s \ll 1/B$), l'amplitude et la phase du signal sont pratiquement constantes. La figure 1.8 montre l'allure du spectre d'un signal modulé à bande étroite. Ce type de signaux est assez courant dans les systèmes de communications, d'où l'intérêt d'avoir pour eux une représentation mathématique commode et claire. D'autre part, on supposera par la suite que les filtres en émission et en réception vérifient le critère de Nyquist, garantissant ainsi l'absence d'Interférence Entre Symboles (IES) [45].

1.3.2 Enveloppe complexe

L'enveloppe complexe du signal à bande étroite $s(t)$ est défini comme suit :

$$\bar{s}(t) = r(t) \exp(j\varphi(t)) \quad (1.4)$$

Le signal $\bar{s}(t)$ est en général complexe et n'a pas de réalité physique mais il permet d'avoir une représentation commode du signal réel $s(t)$. $r(t)$ et $\varphi(t)$ contiennent toute l'information de $s(t)$, qui peut être récupéré à tout instant par :

$$s(t) = \text{Re} \{ \bar{s}(t) \exp(j2\pi f_0 t) \} \quad (1.5)$$

1.3.3 Composantes en phase et en quadrature

L'enveloppe complexe $\bar{s}(t)$ peut s'écrire comme la somme de deux composantes :

$$\bar{s}(t) = s_I(t) + js_Q(t) \quad (1.6)$$

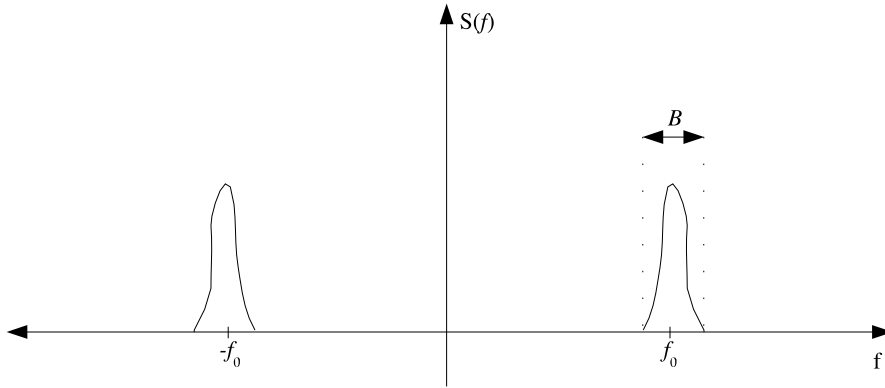


Fig. 1.8 — Spectre d'un signal à bande étroite

avec :

$s_I(t)$: composante en phase avec la porteuse,

$s_Q(t)$: composante en quadrature avec la porteuse.

Leur rapport avec $r(t)$ et $\varphi(t)$ est donné par les relations suivantes :

$$\begin{aligned} s_I(t) &= r(t) \cos \varphi(t) \\ s_Q(t) &= r(t) \sin \varphi(t) \end{aligned}$$

Le signal $s(t)$ peut alors être exprimé en fonction de ses composantes en phase et en quadrature :

$$s(t) = s_I(t) \cos(2\pi f_0 t) - s_Q(t) \sin(2\pi f_0 t) \quad (1.7)$$

1.3.4 Représentation des signaux dans le plan complexe. Notion de constellation

L'interprétation géométrique des signaux est un outil très puissant pour l'analyse de la transmission dans les systèmes numériques. Cela consiste à faire correspondre à chaque signal élémentaire un point dans le plan complexe (ou plan de Fresnel). Les fonctions ψ_x et ψ_y sont les axes de ce plan :

$$\begin{aligned} \psi_x(t) &= \cos(2\pi f_0 t) \\ \psi_y(t) &= -\sin(2\pi f_0 t) \end{aligned}$$

Grâce au concept d'enveloppe complexe, n'importe quel signal appartenant à ce plan peut être exprimé comme une combinaison linéaire des fonctions ψ_x et ψ_y :

$$x(t) = A \cdot \cos(2\pi f_0 t + \varphi) = x_I \cdot \cos(2\pi f_0 t) - x_Q \cdot \sin(2\pi f_0 t) \quad (1.8)$$

où x_I et x_Q sont les coordonnées du signal dans le plan (cf. figure 1.9). Ces coordonnées s'expriment par :

$$\begin{aligned} x_I &= A \cos(\varphi) \\ x_Q &= A \sin(\varphi) \end{aligned}$$

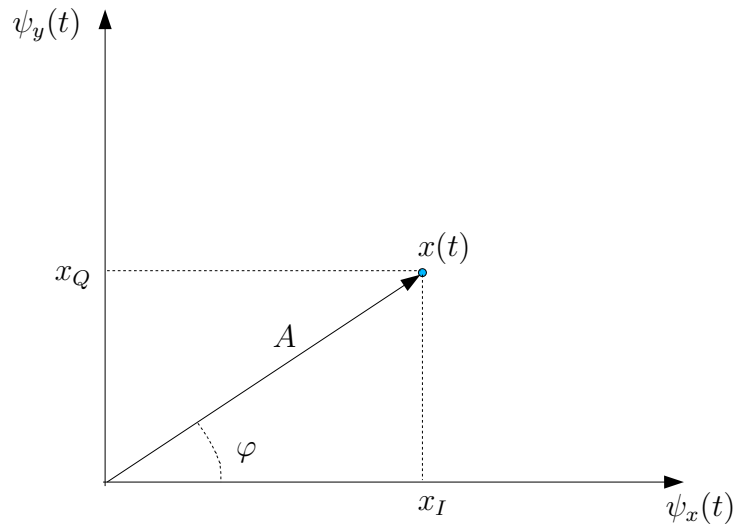


Fig. 1.9 — Représentation d'un signal dans le plan de Fresnel

Dans une transmission numérique, la modulation utilisée pour envoyer l'information utilise un ensemble de M signaux. A chaque système de transmission numérique on associe alors un ensemble de M points nommé *constellation*.

En réception, nous faisons l'hypothèse d'une démodulation cohérente. Un processus de décision est mis en oeuvre afin de retrouver les signaux qui ont été émis. Une certaine probabilité d'erreur est liée à chaque décision. Cette probabilité dépend de la proximité des différents signaux de la modulation : plus les points de la constellation sont éloignés les uns des autres et plus le processus de décision est performant.

En particulier, la *distance euclidienne minimale*, notée d_0 , est définie comme la plus petite des distances entre deux points de la constellation (cf. figure 1.10). Elle est un paramètre déterminant pour la caractérisation d'une modulation. La probabilité d'erreur en réception est directement liée à la valeur de d_0 . Plus d_0 est grande, meilleure est la protection de la modulation vis-à-vis du bruit.

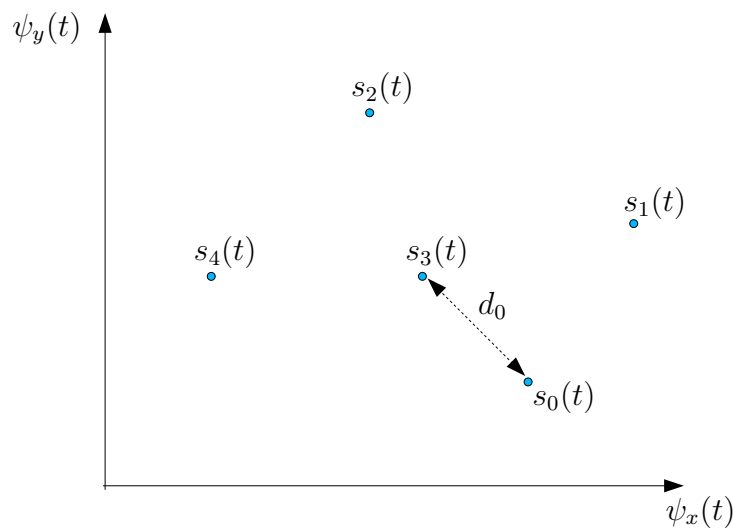


Fig. 1.10 — Représentation de d_0 , distance euclidienne minimale de la constellation

1.3.5 Énergie d'un signal et énergie d'une constellation

L'énergie d'un signal $s(t)$ est définie par :

$$E_s = \int_{-\infty}^{+\infty} |s(t)|^2 dt \quad (1.9)$$

La séquence de signaux modulés envoyée sur le canal peut être exprimée comme suit :

$$S(t) = \sum_k \operatorname{Re}\{A_k h(t - kT) \exp(j2\pi f_0 t + \phi_k)\} \quad (1.10)$$

où T est la durée d'un signal et $h(t)$ est le filtre de mise en forme. Chaque signal $s_k(t)$ est défini par son amplitude A_k et sa phase ϕ_k . Considérant les concepts présentés en Section 1.3.2 et plus précisément l'équation 1.7, on peut écrire :

$$s_k(t) = x_k \cdot \psi_x(t) + y_k \cdot \psi_y(t) \quad (1.11)$$

Le fait de choisir une base orthonormale pour les axes du plan de Fresnel rend plus simple le calcul de l'énergie de la constellation. En effet, si les signaux $\psi_x(t)$ et $\psi_y(t)$ forment une base orthonormale (*i.e.* sont d'énergie unité et orthogonaux entre eux), ils s'expriment comme suit :

$$\begin{aligned} \psi_x(t) &= \sqrt{2/T} h(t - kT) \cos(2\pi f_0 t) \\ \psi_y(t) &= -\sqrt{2/T} h(t - kT) \sin(2\pi f_0 t) \end{aligned}$$

et les coordonnées dans cette base du point (x_k, y_k) qui représente le signal $s_k(t)$ sont :

$$\begin{aligned} x_k &= A_k \sqrt{T/2} \cos \phi_k \\ y_k &= A_k \sqrt{T/2} \sin \phi_k \end{aligned}$$

L'énergie de chaque signal peut alors directement être calculée par :

$$E\{s_k(t)\} = x_k^2 + y_k^2 \quad (1.12)$$

et l'énergie moyenne de la constellation par :

$$E_m = \frac{1}{M} \sum_{i=1}^M \{x_i^2 + y_i^2\} \quad (1.13)$$

où M est le nombre de signaux de la constellation. Dans la suite toutes les représentations de modulations numériques étudiées considèrent une base orthonormale.

1.3.6 Réception en présence de bruit et régions de décision

Le signal reçu à l'entrée du récepteur, appelé *observation*, est la somme du signal émis après modulation et du bruit $n_{\text{total}}(t)$. A priori, ce bruit ne peut pas être représenté dans le plan de Fresnel car il n'appartient pas à l'espace des signaux de la constellation [70]. Néanmoins, ce bruit peut toujours s'exprimer comme :

$$n_{\text{total}}(t) = n(t) + n_{\text{ort}}(t) \quad (1.14)$$

où $n(t)$ est une composante qui appartient à l'espace des signaux de la constellation et $n_{\text{ort}}(t)$ est une composante orthogonale à $n(t)$ (qui ne contient aucune composante appartenant à

l'espace). Il peut être montré [70] que si un récepteur optimal est utilisé, seule la composante $n(t)$ affecte la détection, l'autre composante étant supprimée. $n(t)$ peut donc s'écrire sous la forme :

$$n(t) = n_I \cos(2\pi f_0 t) - n_Q \sin(2\pi f_0 t) \quad (1.15)$$

Les composantes du bruit, n_I et n_Q , sont des variables aléatoires à modéliser selon le type de canal. Ainsi à chaque fois qu'un signal $s_i(t)$ est émis, un signal aléatoire $z(t) = s_i(t) + n(t)$ est reçu, ce signal $z(t)$ pouvant être représenté dans le plan de Fresnel. Il s'avère que si la puissance de bruit est faible, l'ensemble des signaux reçus se concentrent autour des signaux idéaux $s_i(t)$. On observe alors des grappes de points (en anglais, *clusters*) autour des $s_i(t)$ ($0 \leq i \leq M - 1$), comme le montre la figure 1.11. Dans cette figure et dans toute la suite, \mathbf{I} dénote la voie en phase et \mathbf{Q} la voie en quadrature.

En réception, le processus de décision consiste à choisir quel signal a été envoyé connaissant $z(t)$. Le critère le plus évident est de choisir le signal s_i le plus proche dans le plan de Fresnel. La façon la plus efficace d'appliquer ce *critère de proximité* à un signal reçu $z(t)$ est de définir des *régions de décision*. Le plan est ainsi divisé en autant de régions que de signaux $s_i(t)$, chacune de ces régions contenant les points les plus proches de $s_i(t)$. En réception, une simple observation du signal reçu suffira pour décider quel signal a été envoyé. A titre d'exemple, la figure 1.12 montre les régions de décision d'une modulation à trois points alignés ($M = 3$).

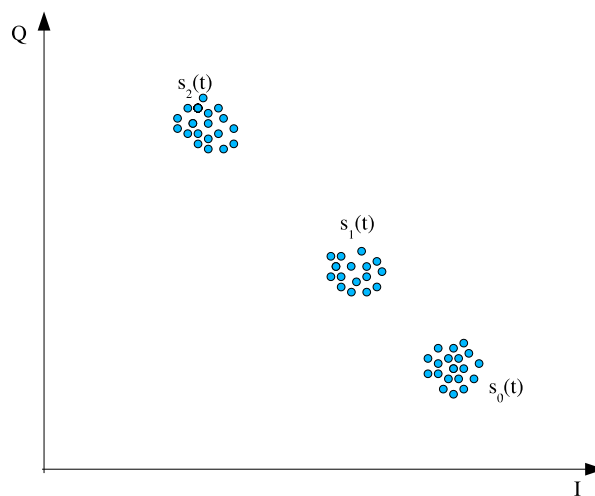


Fig. 1.11 — Représentation des signaux reçus. A faible puissance de bruit ces signaux forment des grappes de points autour des s_i

1.4 Les modulations numériques étudiées

L'opération de modulation consiste à faire varier un paramètre de l'onde porteuse qui est normalement une sinusoïde. Les paramètres modulés peuvent être l'amplitude, la phase ou la fréquence. Dans le cadre de cette thèse nous considérons les modulations par déplacement de phase, les modulations par déplacement d'amplitude sur deux porteuses en quadrature et les modulations par déplacement d'amplitude et de phase.

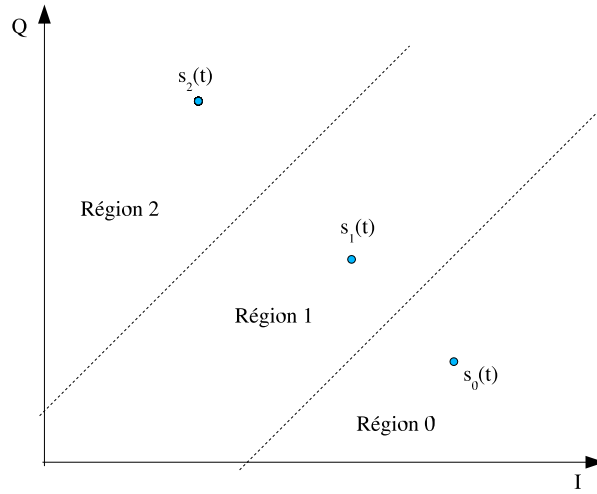


Fig. 1.12 — Exemple de régions de décision pour une modulation à 3 points alignés

1.4.1 Définitions

Dans le schéma de la chaîne de transmission numérique présenté sur la figure 1.1, le codeur de source délivre des éléments binaires (ou *bits*) d_k toutes les T_b secondes. Le *débit binaire* de la source est défini par $D_b = 1/T_b$ (bit/s). Les bits sont groupés en n -tuples ou symboles pour être affectés à des signaux de la modulation. Si la constellation est à M points (*i.e.* la modulation est d'ordre M) : $n = \log_2 M$. La *rapidité de la modulation*, notée D_s , est définie comme le nombre de symboles émis par unité de temps. Elle s'exprime en bauds et vaut :

$$D_s = \frac{1}{T_s} = \frac{D_b}{\log_2 M} \quad (1.16)$$

où T_s est la durée d'émission d'un signal, encore appelé *temps symbole*.

L'*efficacité spectrale* d'un système de communications est le débit utile transmis par unité de bande spectrale occupée, ou autrement dit, le nombre d'éléments binaires d'information transmis par unité de temps et par unité de bande occupée. Si l'on considère une modulation linéaire à M points et qu'un filtre de Nyquist idéal est utilisé pour la transmission, l'efficacité spectrale du système non codé, notée η , vaut :

$$\eta = \log_2 M \text{ (bit/s/Hz)} \quad (1.17)$$

Si un code de rendement R est associé à la modulation, l'efficacité spectrale de l'association est exprimée comme suit :

$$\eta = R \cdot \log_2 M \text{ (bit/s/Hz)} \quad (1.18)$$

En tenant compte du fait que le spectre de fréquences est une ressource très demandée et, en même temps, très limitée, il est crucial de concevoir des systèmes de communications à très grande efficacité spectrale. On tente ainsi de maximiser la quantité d'information transmise en utilisant une largeur de bande fixe.

1.4.2 Modulations par déplacement de phase (MDP, en anglais *PSK*, *Phase Shift Keying*)

Dans ce type de modulation, des signaux de même fréquence et amplitude sont émis avec M phases différentes possibles pour désigner les M symboles. La constellation est constituée de M points déployés sur une circonférence. Il s'agit d'une modulation à enveloppe constante car tous les points de la constellation ont la même amplitude. Cette caractéristique la rend très intéressante pour des canaux non linéaires ou à évanouissements non connus car l'amplitude ne comporte pas d'information.

Une séquence de signaux modulés par déplacement de phase s'exprime sous la forme :

$$S_k(t) = A \sum_k h(t - kT_s) \cos(2\pi f_0 t + \varphi_k + \varphi_0) \quad (1.19)$$

où :

φ_k prend ses valeurs de l'ensemble $(2m + 1)/M$ avec $0 \leq m \leq M - 1$,

φ_0 est une constante,

T_s est le temps symbole,

$h(t)$ représente la fonction créneau (ou porte) :

$$h(t) = \begin{cases} 1 & \text{si } t \in [0, T_s[\\ 0 & \text{sinon} \end{cases} \quad (1.20)$$

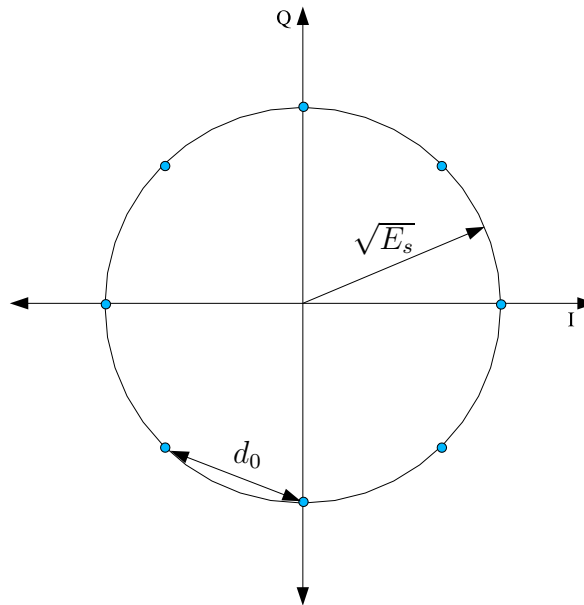


Fig. 1.13 — Constellation de la modulation MDP-8

La figure 1.13 montre la constellation d'une modulation MDP-8. Tous les signaux ont la même énergie E_s , l'énergie moyenne de la modulation étant $E_m = E_s$. Le rayon du cercle où se placent les signaux vaut $\sqrt{E_s}$. La relation entre l'amplitude A et l'énergie E_s est la suivante :

$$A = \sqrt{\frac{2E_s}{T_s}} \quad (1.21)$$

La distance euclidienne minimale, d_0 , s'exprime comme :

$$d_0 = 2\sqrt{E_s} \cos\left(\frac{\pi}{M}\right) \quad (1.22)$$

1.4.3 Modulation d'amplitude sur deux porteuses en quadrature (MAQ, en anglais *QAM, Quadrature Amplitude Modulation*)

Une séquence de signaux modulés en amplitude sur deux porteuses en quadrature s'écrit sous la forme :

$$S(t) = \sum_k a_i h(t - kT_s) \cos(2\pi f_0 t) - \sum_k b_i h(t - kT_s) \sin(2\pi f_0 t) \quad (1.23)$$

Chaque signal dans la séquence est la somme de deux porteuses en quadrature, modulées en amplitude par les signaux $\sum_k a_i h(t - kT_s)$ et $\sum_k b_i h(t - kT_s)$.

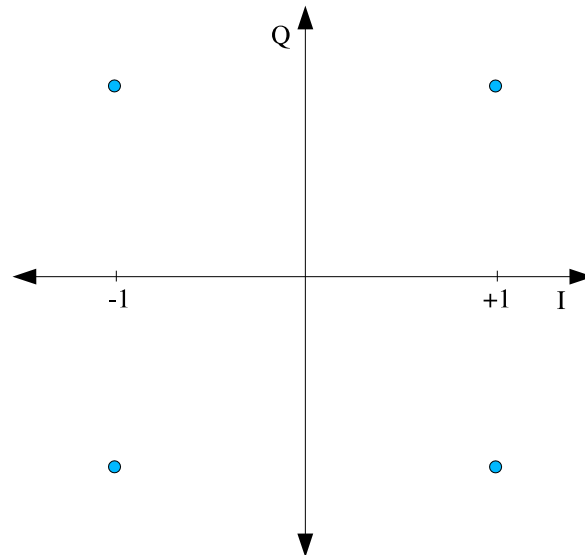


Fig. 1.14 — Constellation de la modulation MAQ-4/MDP-4

Dans notre étude, les modulations MAQ utilisées sont telles que les symboles a_i et b_i sont indépendants et les alphabets où ils prennent leurs valeurs sont identiques. Concrètement, nous considérons les modulations MAQ suivantes : MAQ-4/MDP-4¹, MAQ-16, MAQ-64. Pour ces trois modulations $n = \log_2 M$ est pair et les constellations correspondantes sont carrées. Par contre, le cas de la MAQ-32 est différent car n est impair et la constellation est dite *en croix*. Les figures 1.14 à 1.17 montrent les différentes constellations MAQ- 2^n pour n allant de 2 à 6.

Considérant la représentation sur une base orthonormale, les coordonnées des signaux dans le cas des constellations carrées sont de la forme $(A_i, B_i) \in (\pm 1, \pm 3, \dots, \pm \log_2 M - 1)$ avec :

$$A_i = \sqrt{\frac{T_s}{2}} a_i$$

$$B_i = \sqrt{\frac{T_s}{2}} b_i$$

¹On notera que la constellation de la modulation MAQ-4 est semblable à celle d'une modulation MDP-4

L'énergie moyenne de chaque modulation peut être calculée à partir de l'expression 1.13. Le tableau 1.1 fournit les valeurs d'énergie moyenne et de distance euclidienne minimale pour les différentes modulations MAQ. Ces résultats seront utilisés dans les chapitres suivants.

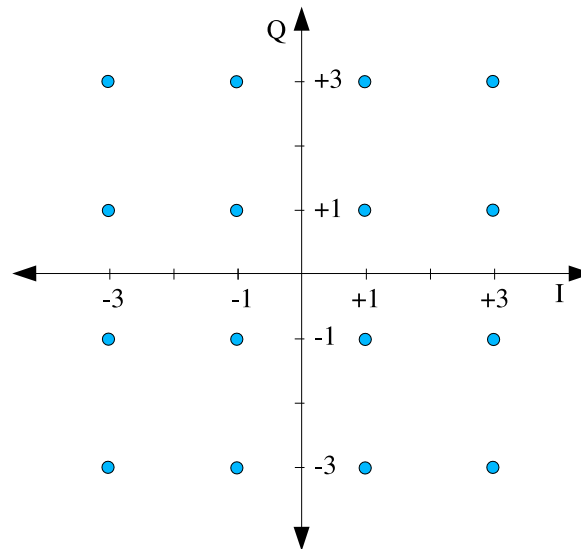


Fig. 1.15 — Constellation de la modulation MAQ-16 (constellation carrée)

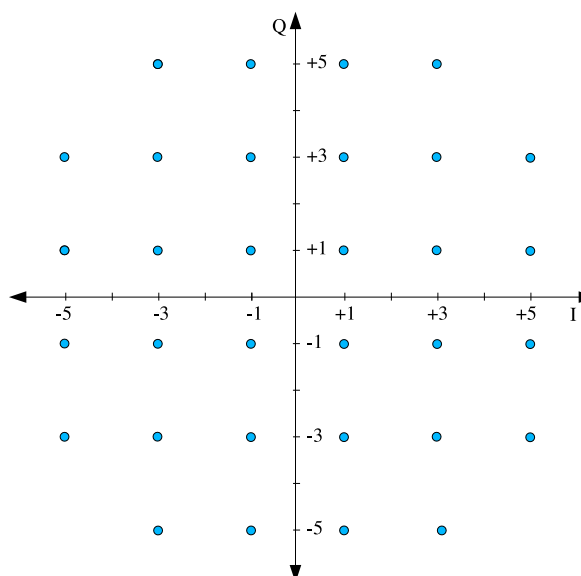


Fig. 1.16 — Constellation de la modulation MAQ-32 (constellation en croix)

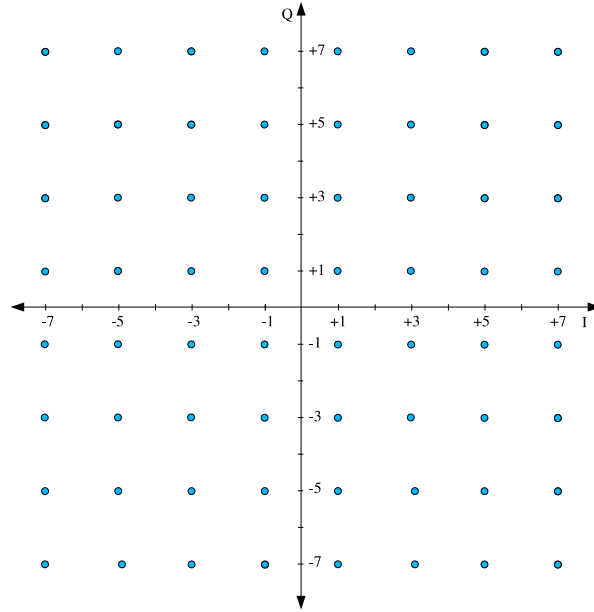


Fig. 1.17 — Constellation de la modulation MAQ-64 (constellation carrée)

| | MAQ-4/MDP-4 | MAQ-16 | MAQ-32 | MAQ-64 |
|-------|---------------|-----------------|-----------------|------------------|
| E_m | 2 | 10 | 20 | 42 |
| d_0 | $\sqrt{2E_m}$ | $\sqrt{2/5E_m}$ | $\sqrt{1/5E_m}$ | $\sqrt{2/21E_m}$ |

Tab. 1.1 — Énergie moyenne et distance euclidienne minimale des différentes modulations MAQ étudiées

1.4.4 Modulation par déplacement d'amplitude et de phase (MDAP, en anglais *APSK*, *Amplitude Phase Shift Keying*)

Dans ce type de modulation, les M symboles sont distribués sur u cercles de rayons R_1, R_2, \dots, R_u , le nombre de symboles sur chaque cercle étant respectivement N_1, N_2, \dots, N_u . Logiquement, $N_1 + N_2 + \dots + N_u = M$. La notation utilisée pour nommer ces modulations est MDAP- $N_1 - N_2 - \dots - N_u$.

Une modulation MDAP peut être vue comme un ensemble de u modulations MDP sur u cercles différents. Les paramètres précisant les rapports entre ces MDPs sont :

$\rho_i = R_{i+1}/R_1$, où R_i est le rayon du cercle i .

φ_i : déplacement de phase relatif entre la constellation MDP du cercle $i + 1$ et celle du cercle 1.

Une séquence de signaux modulés suivant une modulation MDAP s'écrit sous la forme :

$$S(t) = \sum_k h(t - kT) \operatorname{Re} \{ W_k \exp(j2\pi f_0 t) \} \quad (1.24)$$

avec :

$$W_k = \begin{cases} R_1 \exp(j\psi_i^1) & , \psi_i^1 = \frac{2\pi i}{N_1} & , i = 0, \dots, N_1 - 1, \text{ dans le cercle } R_1 \\ R_2 \exp(j\psi_i^2) & , \psi_i^2 = \frac{2\pi i}{N_2} + \varphi_1 & , i = 0, \dots, N_2 - 1, \text{ dans le cercle } R_2 \\ \vdots & \\ R_u \exp(j\psi_i^u) & , \psi_i^u = \frac{2\pi i}{N_u} + \varphi_{u-1} & , i = 0, \dots, N_u - 1, \text{ dans le cercle } R_u \end{cases}$$

Les figures 1.18 et 1.19 montrent deux exemples de modulations MDAP. Dans la littérature, le cas de la MDAP-16 a été le plus étudié [73] [84] [69].

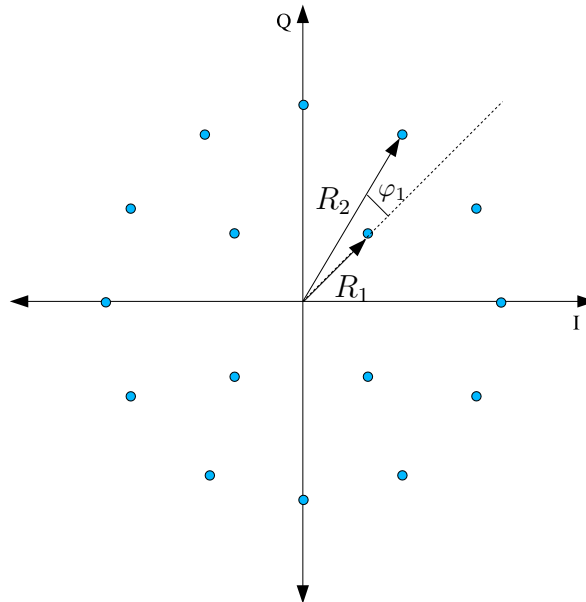


Fig. 1.18 — Constellation de la modulation MDAP-4-12

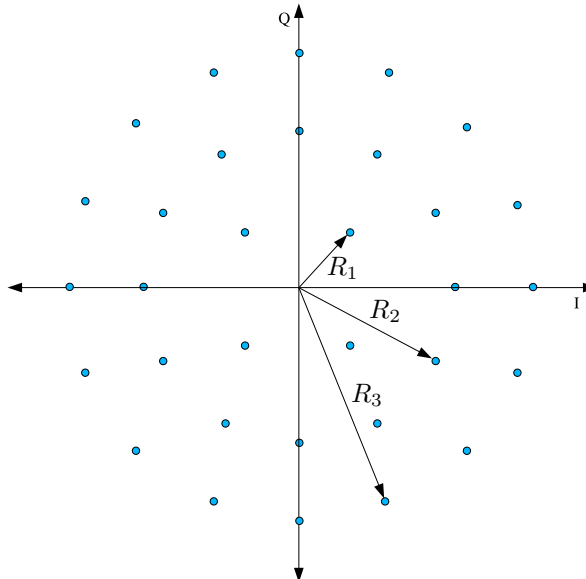


Fig. 1.19 — Constellation de la modulation MDAP-4-12-16

1.5 Modélisation du canal de transmission

Cette section a pour objet de décrire les différents types de canaux considérés dans le cadre de cette thèse. Dans un premier temps, nous présentons les principes physiques à l'origine du modèle de canal à bruit blanc additif gaussien ainsi que le calcul de la variance de bruit

selon la modulation utilisée pour la transmission de données. Dans un deuxième temps, les canaux à trajets multiples sont présentés. Une classification de ce type de canaux est donnée ainsi qu'une description détaillée du canal à évanouissements considéré dans notre étude.

1.5.1 Canal à bruit blanc additif gaussien

La canal à Bruit Blanc Additif Gaussien (BBAG ou AWGN, *Additive White Gaussian Noise*) est le modèle le plus fréquemment utilisé pour caractériser un canal de transmission. Dans ce modèle, le signal reçu $r(t)$ est la somme du signal transmis $s(t)$ et d'un bruit gaussien $n(t)$:

$$r(t) = s(t) + n(t) \quad (1.25)$$

Cette modélisation de canal a une grande importance théorique et pratique. Elle constitue en effet un modèle très précis pour certains types de canaux de transmission, pour les communications satellites et spatiales en particulier. Le bruit additif gaussien est la modélisation d'un bruit dont les sources sont internes ou externes. Les *sources de bruit internes* sont générées à l'intérieur des composants électroniques des équipements du système. Dans la majorité des cas, c'est la source la plus significative et la plus facile à caractériser. Le bruit thermique est la plus importante d'entre elles. Les *sources de bruit externes* sont générées à l'extérieur du système, causées par les intempéries, le bruit industriel, etc. Elles sont difficiles à modéliser.

Calcul de la variance du bruit gaussien en fonction du rapport signal à bruit

Le bruit additif gaussien est caractérisé par un processus aléatoire gaussien centré de densité spectrale de puissance bilatérale égale à $N_0/2$. La variance du bruit est alors : $\sigma^2 = N_0/2$ et la fonction de densité de probabilité est donnée par :

$$p_A(a) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{a^2}{2\sigma^2}\right) \quad (1.26)$$

Si l'on considère un système caractérisé par les paramètres suivants :

E_m : énergie moyenne par symbole,

E_b : énergie moyenne par bit,

R_c : rendement du codeur,

n : nombre de bits par symbole,

on peut écrire :

$$E_m = nR_cE_b \quad (1.27)$$

d'où :

$$\frac{E_b}{N_0} = \frac{E_m}{2R_cn\sigma^2} \quad (1.28)$$

L'égalité 1.28 mène à l'expression de la variance du bruit en fonction du rapport signal à bruit suivante :

$$\sigma^2 = \frac{1}{R_c} \xi \left(\frac{E_b}{N_0} \right)^{-1} \quad (1.29)$$

où la variable ξ est définie comme :

$$\xi = \frac{E_m}{2n} \quad (1.30)$$

Les différentes valeurs de ξ pour les modulations considérées dans notre étude sont données dans le tableau 1.2. Dans le cas de la modulation MDAP-4-12, la valeur de ξ présentée correspond aux valeurs de rayons $R_1 = 1$ et $R_2 = 2.7$ (cf. Section 3.6). Pour le calcul de ces valeurs à partir de l'équation 1.30, on a considéré le tableau 1.1 et la relation $n = \log_2 M$.

| | | | | | | |
|-------|------------|-------|--------|--------|--------|-----------|
| | MAQ-4/QPSK | MDP-8 | MAQ-16 | MAQ-32 | MAQ-64 | MDAP-4-12 |
| ξ | 1/2 | 1/2 | 5/4 | 2 | 7/2 | 0.7147 |

Tab. 1.2 — Valeurs du coefficient ξ pour les différentes modulations étudiées

1.5.2 Les canaux à évanouissements

Les systèmes de communications radio-mobiles utilisent des antennes d'émission et de réception peu directives car le but est d'assurer la couverture totale d'une zone géographique. En conséquence, le signal émis se propage dans plusieurs directions et parvient au récepteur en empruntant des chemins différents. Le signal reçu résulte alors de la superposition de plusieurs répliques du signal émis, caractérisées par des instants d'arrivée et des atténuations différents.

Les canaux à évanouissements sont également nommés canaux à trajets multiples. Les évanouissements liés aux trajets multiples peuvent être classés en évanouissements à grande échelle et évanouissements à petite échelle. Les *évanouissements à grande échelle*, qui peuvent aussi être qualifiés d'évanouissements à long terme, se manifestent par l'atténuation de la puissance moyenne du signal reçu lorsque récepteur et émetteur sont à grande distance. Ces évanouissements comprennent aussi les variations dues à des phénomènes météorologiques. Les *évanouissements à petite échelle* ou à court terme comprennent les variations de l'amplitude et de la phase de très courte durée (de l'ordre d'une demi longueur d'onde). Ils sont dus à l'*étalement temporel* du signal reçu et à l'*effet Doppler*. Ces deux phénomènes sont décrits ci-après.

L'étalement temporel

Lors d'une transmission à trajets multiples, le temps T_m séparant l'arrivée de la première réplique du signal émis et l'arrivée de la dernière réplique du même signal est appelé *étalement temporel maximal*. De même, la *bande de cohérence* du canal, notée B_c , est définie comme la gamme de fréquences sur laquelle les amplitudes des différentes composantes fréquentielles subissent des atténuations semblables.

Il est alors possible de procéder à une première classification des canaux à évanouissements par rapport à la largeur de bande du signal émis : canaux *sélectifs* et canaux *non sélectifs* en fréquence. Si l'on note B la bande de fréquences occupée par le signal, on dit que le canal est non sélectif en fréquence (en anglais, *flat fading channel*) si B est très inférieure à B_c . Dans ce cas, toutes les composantes fréquentielles du signal subissent des atténuations semblables [8]. Dans le cas contraire, c'est-à-dire canal sélectif en fréquence, le système présente de l'Interférence Entre Symboles (IES). C'est pourquoi, en pratique, on cherche à rendre $B \ll B_c$ afin d'éviter l'IES.

L'effet Doppler

Ce phénomène a lieu lors du déplacement de l'émetteur par rapport au récepteur ou lors du déplacement des obstacles entre émetteur et récepteur. La fréquence porteuse de l'onde émise est alors modulée. L'excursion de fréquence maximale est égale à $2f_{d_{max}}$, où $f_{d_{max}}$ est la fréquence Doppler. Cette fréquence $f_{d_{max}}$ est à la fois fonction de la vitesse de déplacement du mobile v et de la longueur d'onde du signal émis λ :

$$f_{d_{max}} = \frac{v}{\lambda} \quad (1.31)$$

La fréquence instantanée de l'onde sur chacun des trajets est soumise à une variation de valeur $f_{d_{max}} \cos(\theta_l)$, où θ_l est l'angle d'arrivée du trajet. Des fluctuations de l'amplitude et de la phase du signal reçu en découlent. Pour caractériser cette non stationnarité, on définit le *temps de cohérence* T_0 comme étant la durée pendant laquelle le canal ne varie pas. $f_{d_{max}}$ et $1/T_0$ sont du même ordre de grandeur.

Une deuxième classification des canaux à évanouissements devient alors possible : canaux à *évanouissements lents* et canaux à *évanouissements rapides*. Le canal est considéré à évanouissements lents (en anglais, *slow fading channel*) lorsque le temps symbole est très inférieure à T_0 , le signal ne variant pas ou alors très peu sur la durée d'un symbole (T_s). Dans le cas contraire on parle d'évanouissements rapides.

Une troisième classification est également possible du point de vue de l'existence ou non d'un trajet direct dans la transmission. On distingue alors les canaux de Rice, lorsqu'il existe un trajet direct, et le canaux de Rayleigh, lorsque ce trajet n'existe pas.

Dans le cadre de notre étude nous considérons les canaux de Rayleigh non sélectifs en fréquence à évanouissements lents. Nous allons définir ci-après la notion de diversité puis fournir une caractérisation statistique du type de canal considéré.

1.5.3 Notion de diversité

Lors d'une transmission sur un canal à évanouissements, le signal émis est soumis à des atténuations importantes. La conséquence immédiate est la dégradation du taux d'erreurs binaires qui s'avère parfois catastrophique. Le principe des techniques de diversité est d'émettre plusieurs répliques du signal d'information sur des canaux indépendants. Si p est la probabilité qu'un signal soit atténué au dessous du seuil de fonctionnement du système, la probabilité que L répliques indépendantes du signal soient atténuées au dessous de ce seuil est p^L . Le paramètre L est appelé *facteur de diversité* du système. Il existe plusieurs façons différentes de faire parvenir au récepteur L répliques du signal indépendamment atténuées, ce sont les *techniques de diversité*. Nous pouvons en particulier citer :

La diversité temporelle : elle consiste à transmettre le signal sur L intervalles temporels séparés d'au moins T_0 (temps de cohérence du canal).

La diversité fréquentielle : elle revient à transmettre le signal sur L fréquences porteuses dont l'écartement fréquentiel est supérieur ou égal à B_c (bande de cohérence du canal).

La diversité spatiale : il s'agit de disposer plusieurs antennes en réception suffisamment éloignées pour assurer une décorrélation des atténuations (normalement une distance d'au moins dix fois la longueur d'onde du canal est nécessaire entre deux antennes afin

d'obtenir des signaux atténués indépendamment). Le traitement en réception sélectionne la meilleure sortie d'antenne ou effectue une combinaison pondérée de toutes les sorties.

1.5.4 Caractérisation statistique du canal de Rayleigh non sélectif en fréquence à évanouissements lents

Nous considérons la transmission d'une sinusoïde $s(t) = \cos(2\pi f_0 t)$ dans un canal à trajets multiples. Chaque trajet introduit un délai temporel et un facteur d'atténuation au signal transmis. Pour le trajet i , on note $\tau_i(t)$ le délai temporel et $\alpha_i(t)$ le facteur d'atténuation. Le signal reçu (sans tenir compte du bruit additif gaussien) est obtenu en faisant la somme des différents trajets :

$$r(t) = \sum_i \alpha_i(t) \cos(2\pi f_0 t - \theta_i(t)) = a_I(t) \cos(2\pi f_0 t) + a_Q(t) \sin(2\pi f_0 t) \quad (1.32)$$

avec :

$$\begin{aligned} \theta_i(t) &= 2\pi f_0 \tau_i(t) \\ a_I(t) &= \sum_i \alpha_i(t) \cos(\theta_i(t)) \\ a_Q(t) &= \sum_i \alpha_i(t) \sin(\theta_i(t)) \end{aligned}$$

Afin d'analyser les propriétés statistiques de $a_I(t)$ et $a_Q(t)$, on suppose que les échantillons du processus aléatoire $\alpha_i(t)$ sont indépendants et identiquement distribués. On suppose également que les échantillons du processus aléatoire $\theta_i(t)$ sont des variables aléatoires uniformes sur $[0, 2\pi]$. Chaque échantillon est indépendant des autres et, en même temps, de chaque $\alpha_i(t)$. Sous ces conditions, on peut montrer que les composantes en phase et en quadrature $a_I(t)$ et $a_Q(t)$ sont deux processus aléatoires gaussiens indépendants [92]. Ces processus gaussiens ont une moyenne nulle et l'enveloppe d'évanouissement, qui est définie comme :

$$a = \sqrt{a_I^2 + a_Q^2}, \quad (1.33)$$

suit une distribution de Rayleigh avec une fonction de densité de probabilité :

$$p_A(a) = \frac{a}{\sigma_a^2} e^{-\frac{a^2}{2\sigma_a^2}} \quad (1.34)$$

où σ_a^2 est la variance des variables aléatoires gaussiennes a_I et a_Q .

Dans le cas du canal de Rice, où il existe un trajet direct de la forme $A \cos(2\pi f_c t)$, la somme des différents trajets peut s'écrire :

$$r(t) = (A + a_I(t)) \cos(2\pi f_c t) + a_Q(t) \sin(2\pi f_c t) \quad (1.35)$$

L'enveloppe de l'évanouissement, définie comme :

$$a = \sqrt{(A + a_I)^2 + a_Q^2} \quad (1.36)$$

suit une distribution de Rice avec une fonction de densité de probabilité de la forme :

$$p_A(a) = 2a(1 + K) e^{-(K+a^2(1+K))} I_0(2a\sqrt{K(K+1)}) \quad (1.37)$$

où $I_0(x)$ est la fonction de Bessel de première espèce d'ordre 0 et K est le rapport entre la puissance moyenne du trajet direct et la puissance moyenne des trajets multiples.

1.6 Conclusion

Dans ce chapitre, après avoir succinctement décrit les différents éléments d'une chaîne de transmission numérique, nous avons établi les bases théoriques permettant de décrire l'ensemble des fonctions concernées par notre étude. Nous avons ainsi présenté les notions fondamentales du codage correcteur d'erreurs et des modulations numériques car c'est l'association des ces deux fonctions qui permet de concevoir des systèmes à grande efficacité spectrale présentant de bonnes performances vis-à-vis du bruit. Une description des modèles de canal de transmission utilisés dans l'étude a également été donnée.

Le chapitre suivant a pour but de présenter dans le détail une famille de codes correcteurs d'erreurs, les turbocodes, particulièrement performants pour le codage de canal, ainsi que leur association avec des modulations numériques à grand nombre de points.

Turbocodes et transmissions à grande efficacité spectrale

Nous avons présenté dans le chapitre précédent les principes généraux du codage correcteur d'erreurs et des modulations numériques. Dans ce deuxième chapitre, nous allons introduire les éléments nécessaires à l'étude de schémas performants de transmission à grande efficacité spectrale, faisant à la fois appel à des codes correcteurs d'erreurs performants, les turbocodes, et à des modulations à grand nombre de points.

Nous présentons dans un premier temps les principes du turbocodage et décodage, tels qu'ils ont été présentés à leur introduction en 1993, ainsi que quelques éléments de performance. Dans un second temps, nous nous intéressons à la famille des turbocodes m -binaires, particulièrement attractifs pour les forts rendements de codage et par conséquent pour les transmissions à grande efficacité spectrale.

Les techniques de modulations codées sont ensuite abordées. Nous présentons plus en détail les modulations codées en treillis et les modulations codées entrelacées au niveau bit. La modulation codée en treillis permet d'atteindre des gains de codage importants sans que cela affecte l'efficacité spectrale du système. Depuis son introduction en 1982, de nombreuses études ont été menées sur cette technique : critères de conception, résultats de simulation, estimation de performances... elles sont à l'origine d'une littérature riche et abondante. Quant aux modulations codées entrelacées au niveau bit, elles ont été introduites en 1989 et leur idée originale de conception a permis d'améliorer de manière significative les performances sur les canaux à évanouissements.

Dans sa dernière partie, le chapitre aborde la problématique de l'association d'un turbo-code à une modulation à grand nombre de points (*i.e.* d'ordre supérieur à 4). Les principaux schémas d'association proposés dans la littérature seront décrits et ensuite comparés en termes de performances.

2.1 Les turbocodes

Cette famille de codes a été pour la première fois présentée à la communauté scientifique en 1993 [19]. Les turbocodes convolutifs proposés par Berrou, Glavieux et Thitimajshima présentaient un excellent gain de codage s'approchant des limites établies par Shannon en 1948 [68]. Ces nouveaux codes étaient construits à partir de la concaténation parallèle de deux codes Convolutifs Systématiques Récursifs (CSR) associés à un entrelaceur interne. A l'époque, la concaténation de codes était déjà une technique bien connue et utilisée par exemple dans les communications spatiales où, typiquement, un code convolutif était concaténé en série avec un code Reed-Solomon. Cette technique permettait en effet de concevoir des systèmes à fort pouvoir de correction présentant une complexité de décodage plus réduite que celle dont on aurait besoin pour décoder un seul code présentant les mêmes performances.

Deux innovations importantes ont permis aux turbocodes d'atteindre des performances sans précédent dans l'histoire du codage de canal. La première est le fait de *concaténer en parallèle* deux codeurs CSR, en général identiques par le moyen d'un *entrelaceur*. La seconde est le procédé de *décodage itératif à entrées et sorties pondérées* : deux décodeurs élémentaires séparés par le même entrelaceur s'échangent des informations au fur et à mesure des itérations.

Dès leur introduction, les turbocodes ont suscité l'intérêt d'un grand nombre de scientifiques ce qui a rapidement généré une quantité importante de contributions sur plusieurs de leurs aspects. Barbulescu et Pietrobon [6] se sont intéressés à la conception d'entrelaceurs pour les turbocodes. Robertson *et al* [62] ainsi que Berrou *et al* [14] ont suggéré des méthodes permettant de réduire la complexité du turbo-décodeur. D'autres publications ont permis de mieux comprendre les excellentes performances des turbocodes. On notera celles de Benedetto et Montorsi [11] [12] ainsi que celle de Perez *et al* [58]. Citons également le tutoriel sur turbocodes par Sklar [71]. Ces dernières années, plusieurs ouvrages sur les turbocodes ont été publiés, par exemple [89] et [50]. D'autre part, le concept du décodage itératif a été utilisé pour d'autres types de codes : dans [60], Pyndiah a étendu le principe des turbocodes aux codes produits; dans [56], Benedetto *et al* ont appliqué le procédé itératif de décodage à la concaténation en série de deux codes convolutifs séparés par un entrelaceur.

L'utilisation de turbocodes dans divers systèmes de communications est en pleine croissance. Par exemple, dans le domaine des communications spatiales, le *Consultative Committee for Space Data Systems* (CCSDS) a adopté un turbocode pour un nouveau standard de télémétrie [77]. Dans le domaine des communications radio-mobiles, les turbocodes sont recommandés pour la transmission de données dans l'UMTS [3]. Les standards de transmission vidéo numérique DVB-RCS [78] et DVB-RCT [79] ont également adopté un turbocode.

2.1.1 Les turbocodes convolutifs

Le turbocodeur

Les premiers turbocodes proposés [18] sont constitués d'une concaténation parallèle de deux codeurs CSR par le moyen d'un entrelaceur ¹. Les deux codeurs agissent sur les mêmes bits d'information. Le premier code les bits dans l'ordre naturel (*i.e.* tels qu'ils sont délivrés par la source) et le second dans l'ordre entrelacé (figure 2.1). La matrice génératrice d'un

¹Les turbocodes sont également connus sous la dénomination de *codes convolutifs concaténés en parallèle* [12].

codeur élémentaire CSR peut être représentée de la façon suivante :

$$G(D) = \begin{bmatrix} 1 & \frac{g_2(D)}{g_1(D)} \end{bmatrix} \quad (2.1)$$

où $g_1(D)$ et $g_2(D)$ sont, respectivement, les polynômes de récursivité et de redondance. Le codeur CSR 1 agit directement sur la séquence d'entrée \mathbf{x} de longueur k bits. Ce codeur a deux sorties : la première est la séquence d'entrée \mathbf{x} et la deuxième, \mathbf{y}_1 , est la séquence des bits de parité ou redondance. La séquence d'information entrelacée, notée $\tilde{\mathbf{x}}$, est introduite dans le codeur CSR 2. La séquence de parité issue du codeur CSR 2 est notée \mathbf{y}_2 . La séquence $\tilde{\mathbf{x}}$ n'est pas transmise. Le rendement naturel du turbocodeur de la figure 2.1 est 1/3. Afin d'obtenir des valeurs de rendement supérieures, les séquences \mathbf{y}_1 et \mathbf{y}_2 sont poinçonnées en utilisant des motifs de poinçonnage adaptés aux rendements souhaités.

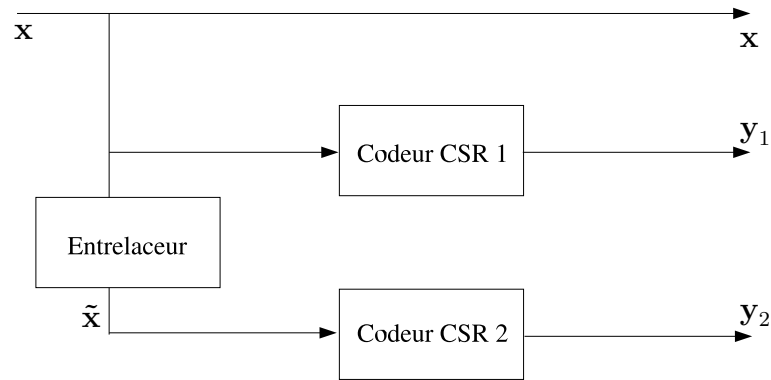


Fig. 2.1 — Schéma de principe d'un turbocodeur

A titre d'exemple, la figure 2.2 montre le turbocode présenté en [19]. Il s'agit d'un turbocode à 16 états, les codeurs élémentaires ayant 4 éléments de mémoire ($\nu = 4$). En notation octale, les polynômes générateurs de chaque codeur CSR binaire sont $g_1 = 37$ et $g_2 = 21$.

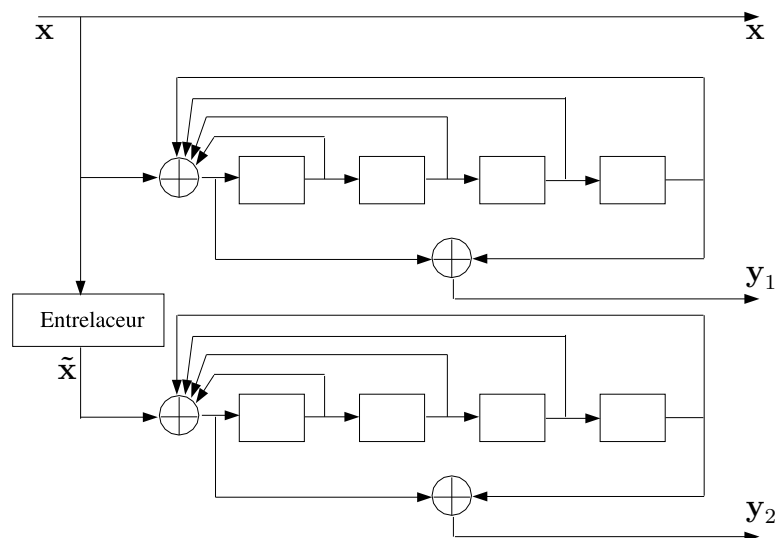


Fig. 2.2 — Exemple de turbocodeur à 16 états de rendement 1/3

L'entrelacement

La fonction d'entrelacement (ou *permutation*) joue un rôle essentiel dans les performances d'un turbocode. D'une part, elle permet de lutter contre l'apparition de paquets d'erreurs en réception pour l'un au moins des codes élémentaires. D'autre part, elle détermine la distribution de poids des mots du code composite et, par conséquent, ses performances asymptotiques à très faible niveau de bruit.

Le point de départ dans la conception d'un entrelacement est souvent la permutation régulière. Elle suppose que le bloc de k bits d'information peut être organisé sous la forme d'un tableau à N_L lignes et N_C colonnes. L'entrelacement consiste alors à y écrire les données ligne par ligne pour ensuite les lire colonne par colonne. La permutation régulière permet d'obtenir une grande distance minimale pour les motifs d'erreurs de poids 2 ou 3 mais pas pour les motifs d'erreurs rectangulaires (poids 4 ou 6) et les combinaisons de ces motifs.

La solution proposée dans [19] consiste à introduire un certain désordre dans la permutation régulière pour *casser* la régularité des motifs composites rectangulaires. De nombreuses études ont été menées sur la recherche d'entrelaceurs à la fois performants et faciles à mettre en œuvre. On notera à titre d'exemple les contributions [4], [51], [32], [6] et [35]. Les entrelaceurs dits *Dithered Relative Prime* [33] sont actuellement parmi les plus performants en terme de comportement asymptotique.

Le turbo-décodeur

Le schéma de principe de décodage d'un turbocode est présenté sur la figure 2.3 [19]. Deux décodeurs élémentaires sont liés par des entrelaceurs dans une structure semblable à celle du turbocodeur. Le décodage des turbocodes est basé sur l'échange itératif d'*informations* dites *extrinsèques* entre les deux décodeurs élémentaires. Ces informations extrinsèques, notées L_{1ex} et L_{2ex} , correspondent aux contributions des décodeurs élémentaires, 1 et 2, au travail de correction. Le turbo-décodage converge d'autant mieux que la corrélation entre les deux processus de décodage élémentaires est faible. Une corrélation minimale serait obtenue avec un entrelacement idéalement aléatoire.

Chaque décodeur a trois entrées : les bits d'information en sortie du canal, les bits de redondance émis par le codeur élémentaire correspondant et les informations extrinsèques provenant de l'autre décodeur élémentaire, relatives aux bits d'information. Chaque décodeur élémentaire fournit une *sortie pondérée* pour chaque bit décodé. Ces sorties pondérées sont typiquement représentées en termes de *Logarithmes de Rapport de Vraisemblance* (LRV). La polarité du LRV détermine la valeur du bit, tandis que son amplitude mesure la probabilité d'une décision correcte. Le LRV en sortie d'un décodeur élémentaire peut être vu comme la somme de deux termes : l'*information intrinsèque*, en provenance du canal de transmission, et l'information extrinsèque. Pour assurer la bonne convergence du processus itératif, seules les informations extrinsèques doivent être échangées entre les décodeurs élémentaires, d'où les opérateurs de soustraction représentés dans la figure 2.3.

Le fonctionnement itératif du turbo-décodeur suit le processus suivant : à la première itération, le décodeur élémentaire 1 ne tient compte que des valeurs \mathbf{x}_{br} et \mathbf{y}_{1br} issues du canal de transmission pour fournir une sortie pondérée qui est introduite après soustraction de \mathbf{x}_{br} en entrée du deuxième décodeur. Ensuite, le deuxième décodeur emploie les informations $\tilde{\mathbf{x}}_{br}$ et \mathbf{y}_{2br} ainsi que l'information extrinsèque provenant du premier décodeur entrelacée, \tilde{L}_{1ex} , pour calculer sa propre sortie pondérée. Pour la deuxième itération, le décodeur élémentaire 1 utilise les valeurs \mathbf{x}_{br} , \mathbf{y}_{1br} et l'information extrinsèque fournie par le deuxième décodeur,

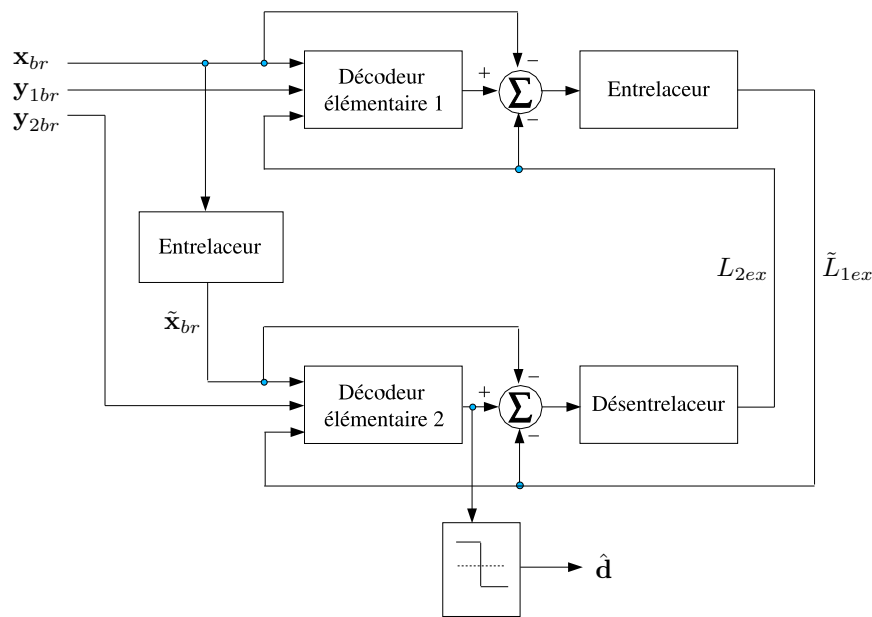


Fig. 2.3 — Schéma de principe d'un turbo-décodateur

L_{2ex} , lors de la première itération. Le cycle se répète ainsi une itération après l'autre, le nombre de bits décodés erronés diminuant au fil des itérations. Cette diminution du taux d'erreurs est très importante dans les premières itérations, puis devient de moins en moins sensible jusqu'à ce qu'il n'y ait plus d'amélioration. Ce phénomène est illustré sur la figure 2.4, où l'on observe les performances du turbocode de la figure 2.2 pour 1, 2, 3, 6 et 18 itérations de décodage, la taille de l'entrelaceur étant de 65536 bits. En pratique, le nombre d'itérations à considérer dépend du compromis performance/simplicité de décodage visé, de l'algorithme de décodage utilisé ainsi que de la taille du bloc.

Pour chaque décodeur élémentaire, deux types différents d'algorithmes de décodage sont couramment considérés :

- L'algorithme de Viterbi à sorties pondérées (en anglais, SOVA : *Soft-Output Viterbi Algorithm*) proposé par Battail [7] et Hagenauer *et al* [46],
- L'algorithme BCJR introduit par Bahl *et al* en 1974 [5], connu également sous le nom de MAP (*Maximum A Posteriori*) car il calcule les probabilités *a posteriori* des bits.

Dans [18], Berrou *et al* utilisent l'algorithme MAP. En effet, cet algorithme rend possible l'échange des informations sur chaque bit décodé car il fournit une probabilité *a posteriori* sur chacun des bits de la séquence décodée. Néanmoins, l'algorithme MAP requiert un grand nombre d'opérations mathématiques dont des calculs d'exponentielles et des multiplications. C'est pourquoi il est considéré comme trop complexe pour être implémenté en pratique dans la plupart de systèmes de communications.

L'algorithme Log-MAP est une transposition de l'algorithme MAP dans le domaine logarithmique. Il est basé sur l'égalité suivante (formule du Jacobien) :

$$\ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|y-x|}) = \max(x, y) + f_c(|y-x|) \quad (2.2)$$

où $f_c(|y-x|)$ est un terme de correction. Les multiplications propres du MAP sont alors calculées comme des additions et les exponentielles disparaissent.

Le Max-Log-MAP est une version simplifiée du MAP qui utilise l'approximation suivante :

$$\ln(e^x + e^y) \simeq \max(x, y) \quad (2.3)$$

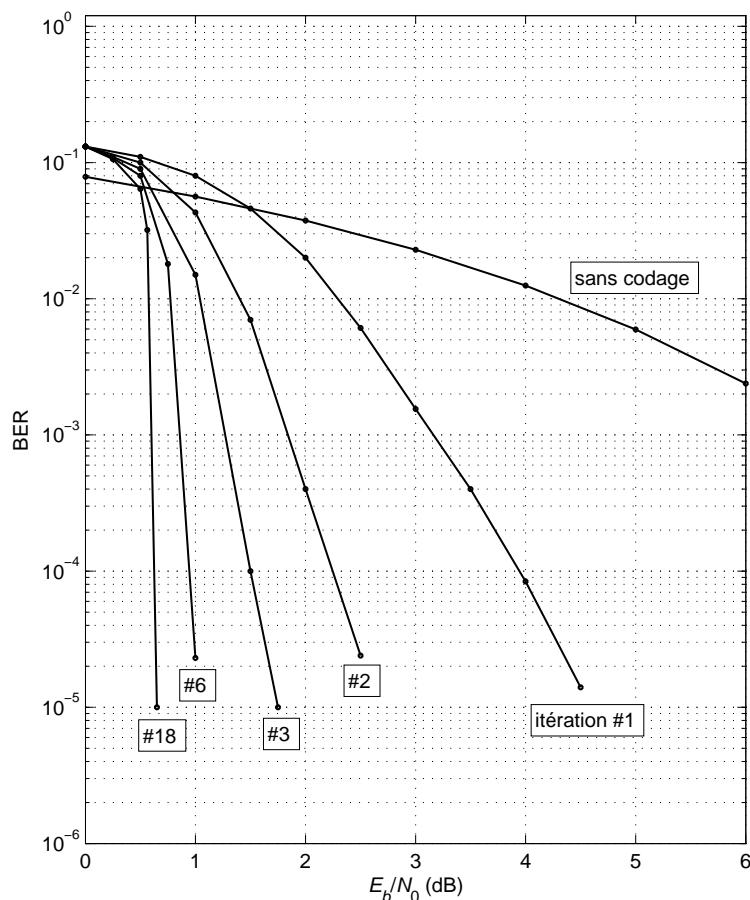


Fig. 2.4 — Performance au fil des itérations du turbocode présenté en [19] (schéma du turbocodeur sur la figure 2.2). Taille de l’entrelaceur : 65536 bits. Rendement $R = 1/2$. Algorithme de décodage MAP.

Il s’agit d’un algorithme sous-optimal [62] mais assez utilisé car il présente un bon compromis simplicité/performance. De plus, il ne requiert pas la connaissance de la variance de bruit σ^2 , ce qui facilite sa mise en œuvre.

Performances des turbocodes

Les performances d’un turbocode sont en grande partie déterminées par l’ensemble de paramètres suivants : les polynômes générateurs et la longueur de contrainte des codeurs élémentaires, la taille de bloc, l’entrelaceur, l’algorithme de décodage élémentaire et le nombre d’itérations de décodage. Le choix de ces paramètres et la variété de critères possibles lors de la conception d’un système turbocodeur/décodeur font que ses performances peuvent se situer à moins de 1 dB de la limite établie par Shannon ou à peine améliorer celles d’un simple codeur convolutif.

Les résultats présentés dans [19] montrent qu’il est en effet possible de s’approcher de 0,5 dB de la capacité d’un canal gaussien à entrées binaires avec un système turbocodeur/décodeur. Cette performance remarquable est obtenue pour une taille d’entrelaceur de 65536 bits et 18 itérations de décodage. Les codeurs élémentaires sont constitués de quatre éléments de mémoire ($\nu = 4$), les polynômes générateurs ayant été choisis égaux à : $g_1 = 37$,

$g_2 = 21$. Un entrelacement de type pseudo-aléatoire est utilisé et l'algorithme de décodage utilisé est le MAP.

La perte (ou le gain) en performance causée par la variation d'un ou plusieurs de ces paramètres a été analysée en détail dans [49]. Cette étude n'étant pas l'objectif de cette thèse, nous retiendrons simplement les principes ou conclusions suivants :

- Le gain en performance qu'apporte un turbocode est d'autant plus important que la taille de bloc est grande. Puisque la complexité du décodeur, à l'exception des mémoires, ne dépend pas de la taille de bloc utilisée, de très bonnes performances peuvent être obtenues lorsque l'on utilise de grandes tailles de bloc. Or, pour beaucoup d'applications une grande taille de bloc n'est pas adaptée en raison du délai qu'elle introduit dans le système. Pour des systèmes turbocodés faisant appel à la transmission de blocs courts le gain de codage potentiel est moins important. En effet, les valeurs de rapports signal à bruit correspondant aux limites théoriques de corrections augmentent lorsque la taille des blocs transmis diminue [37].
- La conception des codeurs élémentaires détermine fortement les performances du turbocode. En général, il faut chercher à maximiser sa distance libre, sans oublier que l'entrelacement fait que cette "maximisation élémentaire" ne se traduit pas nécessairement dans une maximisation de la distance minimale du turbocode. Pour chaque longueur de contrainte les polynômes générateurs les plus performants ont été proposés, à savoir : pour $K = 3$, Hagenauer *et al* [47] et Jung [54] ont utilisé $g_1 = 7$ et $g_2 = 5$; pour $K = 4$, $g_1 = 15$ et $g_2 = 17$ sont proposés dans [49] ; pour $K = 5$, Berrou *et al* [19] ont utilisé $g_1 = 37$, $g_2 = 21$ tandis que Hagenauer *et al* [47] ont proposé $g_1 = 23$, $g_2 = 35$.
- La façon dont la longueur de contrainte, K , des codeurs élémentaires affecte les performances du turbocode dépend du taux d'erreurs considéré : une bonne performance à faible rapport signal à bruit est obtenue avec de petites valeurs de K . En effet, la convergence du processus de décodage itératif est d'autant meilleure que le nombre d'états dans le treillis est faible. En revanche, une grande valeur de K sera nécessaire pour avoir de grandes distances minimales et une bonne performance asymptotique. Par exemple, pour un $BER \leq 10^{-2}$ les meilleures performances sont obtenues avec $K = 3$. Pour de plus faibles taux d'erreurs, $BER \leq 10^{-3}$, le gain en performance augmente avec K , et ce gain est d'autant plus important que le taux d'erreurs est faible. Pour un turbocode constitué des codeurs élémentaires décrits dans le paragraphe précédent, le passage de $K = 3$ à $K = 5$ apporte un gain d'à peine 0,2 dB à un $BER=10^{-3}$, et de plus de 0,3 dB à un $BER=10^{-4}$.
- Dans la plupart des applications, le nombre d'itérations de décodage est compris entre 4 et 12, l'idéal étant de trouver un compromis entre l'amélioration de performances aux dernières itérations et la complexité de décodage pour la taille de bloc utilisée.
- Quant à l'algorithme de décodage à utiliser, on peut affirmer que le MAP et ses versions simplifiées sont plus faciles à implémenter pour des blocs courts que le SOVA. Dans [62], il est mentionné qu'il n'y a pas de dégradation visible entre le MAP et le Log-MAP lorsque le terme correctif de l'équation 2.2 est codé sur 3 bits. D'après l'expérience du Département Electronique de l'ENST Bretagne, on constate que la perte en performance introduite par le Max-Log-MAP est d'environ 0,1-0,2 dB pour des blocs courts et de 0,4 dB pour des blocs longs.

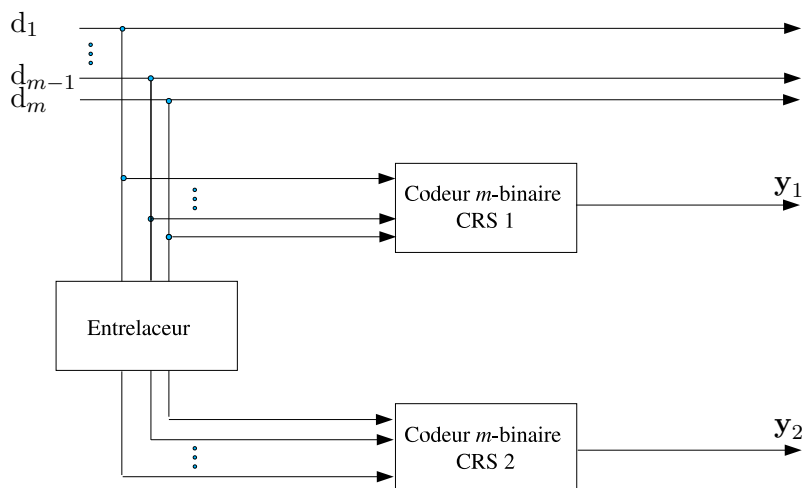


Fig. 2.5 — Turbocodeur m -binaire

2.1.2 Les turbocodes m -binaires

Les turbocodes m -binaires sont construits à partir de codes CSR à m entrées binaires ($m \geq 2$)². Le rendement naturel d'un turbocodeur m -binaire est $m/(m + 2)$ (figure 2.5). Cette nouvelle famille de turbocodes présente un nombre important d'avantages par rapport aux turbocodes binaires classiques [23] :

1. **Possibilité de concevoir un entrelacement plus robuste** : plusieurs bits étant codés simultanément, une permutation à deux niveaux peut être envisagée : entre les groupes de m bits (entrelacement *inter-symbole*) et à l'intérieur du groupe m -binaire (entrelacement *intra-symbole*). L'entrelacement intra-symbole permet d'introduire des non-uniformités, *i.e.* du désordre dans la permutation, sans que cela affecte les bonnes propriétés de l'entrelacement inter-symbole. Le résultat est un entrelacement global plus robuste qui se traduit par une distance minimale plus élevée et une meilleure performance asymptotique.
2. **Diminution de la densité de motifs d'erreurs verrouillés dans le processus de décodage** : ce phénomène a été d'abord observé dans [13] et commenté dans [21]. La figure 2.6 donne une vue symbolique du processus de décodage tel qu'il pourrait apparaître dans une matrice d'entrelacement régulier ligne-colonne. Si le bloc codé est de taille k , la matrice d'entrelacement est carrée avec un côté de longueur \sqrt{k} . Les chemins erronés sont représentés par des lignes. Lorsque ces lignes forment un rectangle, des *motifs d'erreurs verrouillés* apparaissent et le processus d'échange des informations a du mal à converger. L'utilisation de codes m -binaires divise la longueur des chemins erronés par m alors que la taille de la matrice d'entrelacement est réduite par un coefficient $1/\sqrt{m}$. Cette modification du rapport de la longueur des chemins erronés sur la taille d'entrelacement réduit l'occurrence d'apparition de *motifs d'erreurs verrouillés*, entraînant ainsi une meilleure convergence du processus itératif.
3. **Réduction de la perte en performance occasionnée par l'application d'algorithmes sous-optimaux de décodage** : concrètement, il s'avère que la pénalité de 0,2-0,6 dB due à utilisation du Max-Log-MAP au lieu du MAP est divisée par 2 dans le cas d'un turbocode duo-binaire ($m = 2$). Cela peut être dû au fait que, pour une longueur

² $m = 1$ dans un turbocode classique

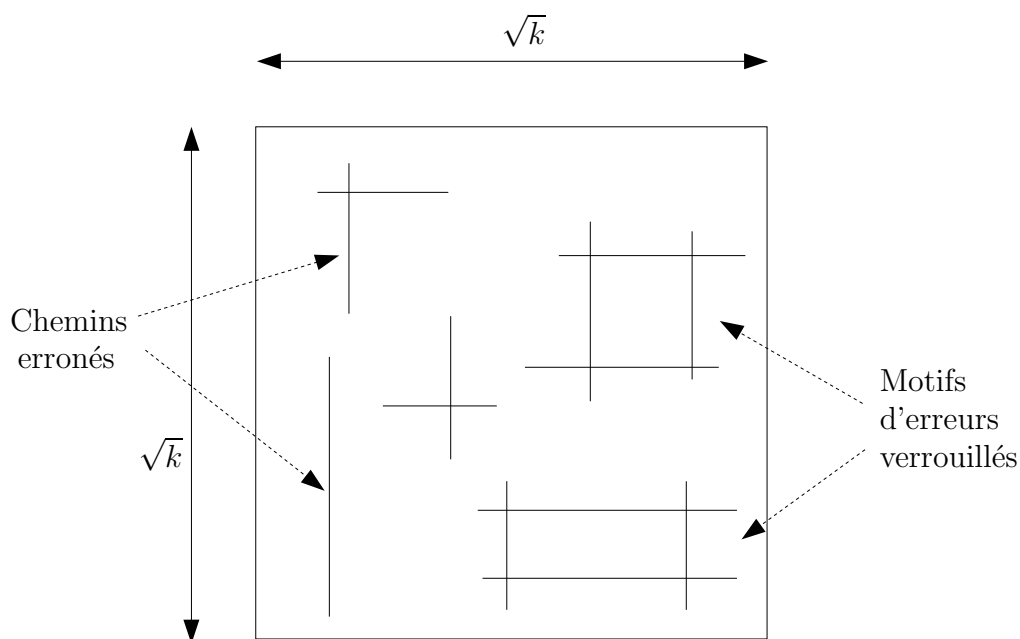


Fig. 2.6 — Chemins erronés et motifs d'erreurs verrouillés dans un processus de turbo-décodage à deux dimensions

de bloc donnée, le nombre de niveaux dans le treillis étant moins grand, l'algorithme de décodage est plus proche du décodage à Vraisemblance Maximale.

4. **Sensibilité aux motifs de poinçonnage réduite :** un turbocodeur binaire a un rendement naturel $R_{nat} = 1/3$. Pour trouver des valeurs de rendements élevées, des bits de redondance sont poinçonnés avant transmission. Dans le cas d'un turbocodeur m -binaire, $R_{nat} = m/(m + 2)$, c'est-à-dire, $R_{nat} \geq 1/2$ pour toute valeur de $m \geq 2$. Par conséquent, moins de bits de parité doivent être enlevés pour atteindre la valeur de rendement désirée. Les codes m -binaires s'avèrent particulièrement intéressants pour des rendements élevés.
5. **Réduction de la latence du circuit d'implémentation :** puisque m bits d'information sont traités parallèlement au codage et au décodage, le débit du système de communications est augmenté d'autant.

Parmi les turbocodeurs m -binaires, les codes duo-binaires ($m = 2$) présentent le meilleur compromis complexité/performance. Les standards de transmission numérique DVB-RCS [78] et DVB-RCT[79] ont adopté un turbocode duo-binaire à 8 états pour le codage correcteur d'erreurs.

2.2 Les modulations codées

Le spectre de fréquences est une ressource à la fois très limitée et très demandée. Sa répartition est sévèrement réglementée par l'Union Internationale des Télécommunications. C'est pourquoi l'un des objectifs le plus important dans la conception d'un système de communications numériques est l'exploitation maximale du spectre disponible. L'efficacité spectrale du système doit ainsi être la plus grande possible afin de pouvoir fournir des services qui demandent toujours des débits plus élevés. Pour que le système présente en même temps de bonnes propriétés de correction d'erreurs, il est alors nécessaire d'unir les bénéfices du codage

à ceux d'une modulation à grand nombre de points.

Traditionnellement, codage et modulation ont été considérés comme deux entités séparées dans les systèmes de communications. En 1982, les travaux présentés par Ungerboeck [85] ont montré qu'une optimisation conjointe du codage et de la modulation était possible : les Modulations Codées en Treillis (MCT) permettaient en effet de définir des schémas avec des gains de codage importants sans avoir à sacrifier en efficacité spectrale, tel que nous allons les décrire en Section 2.2.3.

Bien que les MCTs étaient à l'origine conçues pour le canal gaussien, elles ont ensuite été adaptées pour les canaux à trajets multiples. L'ouvrage de Jamali et Le-Ngoc [53] présente de façon exhaustive les techniques utilisées pour l'application des MCTs aux canaux à évanouissements. En 1989, Zehavi [93] a introduit les modulations codées entrelacées au niveau bit (en anglais, BICM : *Bit-Interleaved Coded Modulation*). Les BICMs associent des codes convolutifs à des entrelaceurs niveau bit indépendants. Cette innovation permet d'augmenter l'ordre de diversité de la modulation codée jusqu'à la valeur de la distance de Hamming du code, les performances des transmissions sur des canaux à évanouissements étant ainsi améliorées de façon très significative. Les BICMs présentent en conséquence des performances meilleures que les MCTs sur les canaux à évanouissements. En revanche, elles restent moins performantes que les MCTs sur le canal gaussien. Une description des BICMs est donnée en Section 2.2.4.

Nous définissons au préalable les paramètres à considérer lors de la conception d'une modulation codée. Ces paramètres déterminent les performances de la modulation codée sur canal gaussien et/ou sur canal à évanouissements.

2.2.1 Distance libre d'une modulation codée

Soient $\mathbf{s}_A = (s_{A1}, s_{A2}, \dots, s_{Al})$ et $\mathbf{s}_B = (s_{B1}, s_{B2}, \dots, s_{Bl})$ deux séquences codées issues d'un bloc codeur/modulateur, où s_i ($i \in 1 \dots l$) représente un symbole codé et l est le nombre total de symboles codés dans une séquence. La distance euclidienne au carré entre \mathbf{s}_A et \mathbf{s}_B s'exprime comme :

$$d_E^2(\mathbf{s}_A, \mathbf{s}_B) = \sum_{i=1}^l |s_{Ai} - s_{Bi}|^2 \quad (2.4)$$

La *distance libre* de la modulation codée est définie comme la plus petite des distances euclidiennes entre deux séquences codées :

$$d_{libre} = \min_{\mathbf{s}_A \neq \mathbf{s}_B} d_E(\mathbf{s}_A, \mathbf{s}_B) \quad (2.5)$$

La règle de décision d'un décodeur à Vraisemblance Maximale sur canal gaussien est basée sur la minimisation de la distance euclidienne entre deux séquences. Autrement dit, le décodeur optimal choisit la séquence codée qui est la plus proche de la séquence reçue en termes de distance euclidienne. La performance de la modulation codée sur canal gaussien est alors essentiellement déterminée par la distance libre de la modulation codée. Il peut être montré [53] qu'une estimation asymptotique du taux d'erreurs de trames (FER) pour de forts rapports signal à bruit est donnée par :

$$\text{FER} \simeq \frac{1}{2} m(d_{libre}) \text{erfc} \left(\frac{d_{libre}}{2\sqrt{N_0}} \right) \quad (2.6)$$

où $m(d_{libre})$ est le nombre moyen de séquences situées à une distance d_{libre} de la séquence transmise.

Par conséquent, les meilleures performances sur canal gaussien sont atteintes lorsque la modulation codée est conçue de façon à ce que d_{libre} soit la plus grande possible. En général, pour une modulation codée, une augmentation de la distance de Hamming du code, d_{min} , n'entraîne pas nécessairement une augmentation de d_{libre} .

2.2.2 Longueur effective et distance produit d'une modulation codée

La distance produit entre deux séquences codées \mathbf{s}_A et \mathbf{s}_B s'exprime comme :

$$d_P(\mathbf{s}_A, \mathbf{s}_B) = \prod_{i=1}^l |s_{Ai} - s_{Bi}| \quad (2.7)$$

La *distance* ou *longueur effective* du code, L , est définie [31] [67] comme la longueur du chemin erroné le plus court dans le treillis exprimée en nombre d'étapes parcourues. La *distance produit* de la modulation codée, $d_P(L)$, est définie comme le produit des distances euclidiennes entre les symboles codés désignés par les branches du treillis entre le chemin correct et le chemin erroné le plus court :

$$d_P(L) = \prod_{i=1}^L |s_i - \hat{s}_i| \quad (2.8)$$

où \mathbf{s}_i est la séquence correspondant au chemin correct et $\hat{\mathbf{s}}_i$ la séquence correspondant au chemin erroné le plus court.

La performance asymptotique de la modulation codée sur des canaux à évanouissements dépend fortement de ces deux paramètres. En effet, pour de forts rapports signal à bruit, la probabilité d'erreurs binaires, P_{eb} , d'une modulation codée pour ce type de canaux peut être approximée par [53] :

$$P_{eb} \simeq \alpha(L, d_P^2(L)) \frac{((1+K)e^{-K})^L}{\left(\frac{1}{4N_0}\right)^L d_P^2(L)} \quad (2.9)$$

avec :

$\alpha(L, d_P^2(L))$: constante dépendant de la distribution de poids du code,

K : paramètre de Rice ($K = 0$ pour le canal de Rayleigh),

L : distance effective du code,

$d_P(L)$: distance produit de la modulation codée.

L'équation 2.9 montre que P_{eb} varie de façon inversement proportionnelle à $(E_s/N_0)^L$, car $d_P(L)$ est un produit de distances entre points et, dans une modulation, ces distances sont directement proportionnelles à $\sqrt{E_s}$ (cf. Section 1.3.5). Autrement dit, P_{eb} varie asymptotiquement comme l'inverse de la puissance L -ème du rapport signal à bruit. Ce phénomène est similaire à celui que l'on trouve avec les techniques de diversité temporelle qui ont été décrites en Section 1.5.3. C'est pourquoi la paramètre L est aussi appelé *ordre de diversité* du code [36].

2.2.3 Les Modulations Codées en Treillis (MCT)

Cette technique a été originellement proposée par Ungerboeck en 1982 [85]. L'idée est d'intégrer un code à une modulation à grande efficacité spectrale. Le nombre de signaux utilisés par le modulateur est augmenté, conduisant ainsi à une expansion de la constellation. Cette expansion fournit la redondance nécessaire pour le codage mais, pour une même énergie moyenne, elle diminue la distance entre les points de la constellation. Cette technique ne présente donc un intérêt que si l'augmentation du taux d'erreurs liée à la réduction de la distance est compensée par le codage.

Considérons par exemple une MDP-4 non codée, où un symbole est constitué de deux bits d'information ($\eta = 2$ bit/s/Hz). Si l'on ajoute un bit de redondance, il est nécessaire d'augmenter l'ordre de la modulation de façon à ce que cette redondance ne diminue pas l'efficacité spectrale du système. Autrement dit, pour réaliser une transmission ayant un efficacité spectrale de $\eta = 2$ bit/s/Hz, on pourra utiliser une modulation MDP-4 sans codage ou une modulation MDP-8 associée à un code de rendement $2/3$.

Si l'on utilise un code convolutif, le fonctionnement du module codeur/modulateur peut être représenté par un diagramme en treillis unique dont le nombre d'états est égal à 2^ν , où ν est la mémoire du code. A chaque branche du treillis on associe un point de la constellation, codage et modulation étant ainsi réalisés de manière conjointe.

Une procédure de démodulation ferme³ suivie d'un décodage ne permet pas une utilisation optimale de l'information reçue. Par conséquent, démodulation et décodage sont également envisagés de manière conjointe, dans un processus unique. Le décodeur opère directement sur les échantillons pondérés issus du canal, l'information étant ainsi exploitée de façon optimale.

Nous allons décrire ci-après les principes de base de construction des MCTs, le lecteur pouvant se référer aux publications d'Ungerboeck [85] [86] ainsi qu'à l'ouvrage [27] par Biglieri *et al* pour un complément d'information sur cette technique.

Construction des MCTs

La figure 2.7 présente la structure générale d'un codeur MCT. Dans ce schéma, à chaque instant n , un bloc de m bits d'information $(a_n^{(1)}, a_n^{(1)}, \dots, a_n^{(m)})$ est introduit dans le codeur. \tilde{m} de ces m bits ($\tilde{m} \leq m$) sont codés par un codeur convolutif de rendement $\tilde{m}/(\tilde{m} + 1)$. Les $m - \tilde{m}$ bits restants ne sont pas codés. Les $(\tilde{m} + 1)$ bits de sortie du codeur convolutif sont utilisés pour sélectionner un des $2^{\tilde{m}+1}$ sous-constellations possibles tandis que les $m - \tilde{m}$ bits non-codés restants sont utilisés pour sélectionner un des $2^{m-\tilde{m}}$ signaux dans la sous-constellation.

Afin de maximiser la distance libre de la MCT, Ungerboeck a proposé les étapes suivantes pour sa conception :

Le partitionnement de la constellation ou *set-partitioning* : il s'agit de diviser la constellation en sous-constellations présentant des distances euclidiennes minimales croissantes.

Les règles de construction du treillis : l'affectation à chaque branche du treillis d'un signal appartenant à la constellation doit, dans la mesure du possible, respecter les trois règles suivantes :

³Autrement dit, une démodulation par régions de décision. Le principe de cette technique a été décrite en Section 1.3.6

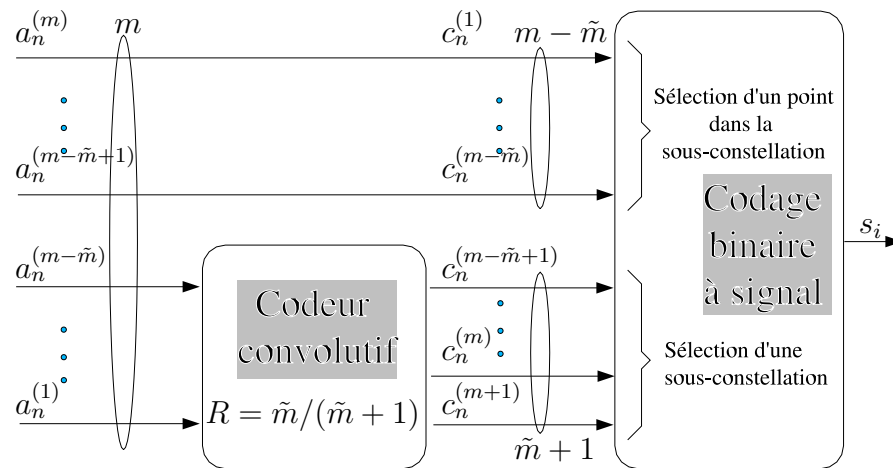


Fig. 2.7 — Structure générale d'un codeur MCT

1. Les M signaux de la constellation A_0 doivent être utilisés avec la même fréquence.
2. Les $2^{m-\tilde{m}}$ branches parallèles, si elles existent, doivent être associées à des signaux appartenant à une même sous-constellation à $2^{m-\tilde{m}}$ points.
3. Les 2^m branches qui quittent un état ou atteignent un état doivent être associées à des signaux appartenant à une même sous-constellation à 2^m points.

La première règle assure au treillis un motif régulier, les règles 2 et 3 garantissent que la distance libre de la MCT est toujours supérieure à la distance euclidienne minimale de la modulation non codée prise comme référence pour le calcul de gain de codage. En effet, la distance euclidienne entre deux chemins qui divergent puis convergent sur une branche (cas de branches parallèles) ou sur plusieurs branches (au moins deux) est toujours supérieure à la distance minimale d'une sous-constellation à 2^m points. Lorsque le nombre d'états du treillis est trop faible, la règle 3 peut ne pas être vérifiée.

Un exemple de partitionnement d'une constellation MDP-8 est donné sur la figure 2.8. La constellation A_0 à $M = 2^{m+1}$ points a une distance euclidienne minimale d_0 . Dans cette constellation, chaque point est associé à un mot binaire de $m + 1$ bits. Le partitionnement se fait sur plusieurs étapes. La première consiste à diviser la constellation A_0 en deux sous-constellations (B_0 et B_1) ayant respectivement 2^m points et la même distance euclidienne minimale d_1 ($d_1 > d_0$). Dans une deuxième étape, chacune des sous-constellations B_0 et B_1 est divisée en deux sous-constellations ayant une distance euclidienne minimale d_2 ($d_2 > d_1 > d_0$). On obtient ainsi quatre sous-constellations : C_{00} , C_{01} , C_{10} et C_{11} .

Quelques exemples des performances des MCTs

Considérons une modulation codée à huit états de phase, $m + 1 = 3$. L'efficacité spectrale du système est donc de $m = 2$ bit/s/Hz et ses performances sont à comparer à celles de la modulation MDP-4. Si on utilise un treillis à 4 états et on respecte les règles de construction d'Ungerboeck, le gain asymptotique de la MCT ainsi obtenue par rapport à une MDP-4 non codée est de 3 dB. Lorsque l'on augmente le nombre d'états du treillis le gain asymptotique devient plus important, par exemple : 3,6 dB pour un treillis à 8 états et 5,2 dB pour un treillis à 128 états. Dans le cas d'une MAQ-16 codée par rapport à une MDP-8 non codée les gains asymptotiques peuvent aller de 3,1 dB pour un treillis à 2 états jusqu'à 7,4 dB pour un treillis à 256 états [86].

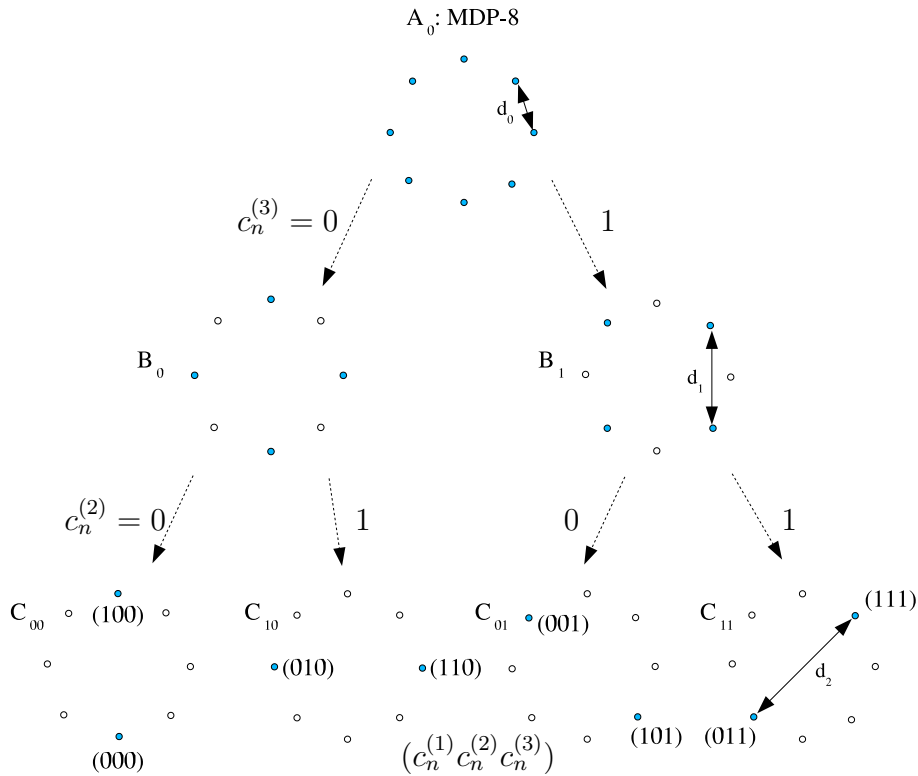


Fig. 2.8 — Partitionnement d'une constellation MDP-8

2.2.4 Les modulations codées entrelacées au niveau bit

Dans [94], Zehavi a introduit la technique de modulation codée entrelacée au niveau bit (BICM : *Bit-Interleaved Coded Modulation*). Elle fut ensuite formalisée par Caire *et al* dans [29]. Cette technique novatrice a pour but d'augmenter l'ordre de diversité des MCTs d'Ungerboeck sur les canaux à évanouissements. L'idée est de rendre l'ordre de diversité aussi grand que la distance minimale de Hamming en entrelaçant les symboles codés au niveau bit. La structure d'un codeur BICM basé sur un codeur convolutif est présentée sur la figure 2.9. Dans un premier temps, les bits d'information sont codés par le codeur convolutif pour ensuite être entrelacés par des entrelaceurs indépendants au niveau bit. Le nombre d'entrelaceurs au niveau bit est égal au nombre de bits compris dans un symbole codé. Dans le cas de la figure 2.9, un symbole se compose de trois bits codés, trois entrelaceurs au niveau bit sont alors utilisés. Cet entrelacement au niveau bit a un objectif double :

1. disperser les paquets d'erreurs dus aux évanouissements corrélés et maximiser l'ordre de diversité du système.
2. décorréliser ou rendre indépendants les bits appartenant à un même symbole.

Les bits entrelacés sont ensuite groupés en symboles (dans le cas de la figure 2.9, des symboles constitués de trois bits) et finalement affectés à un signal de la constellation. Le décodeur BICM montré sur la figure 2.10 implémente le processus inverse de celui du codage BICM. Le bloc démodulateur produit des métriques associées aux bits à partir des signaux reçus. Ces métriques sont ensuite désentrelacées par des désentrelaceurs indépendants afin de reconstruire les mots de code estimés. Finalement, le décodeur convolutif essaye de retrouver la séquence de bits la plus proche de la séquence originale de bits d'information.

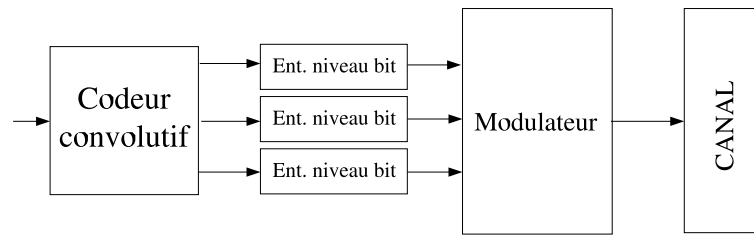


Fig. 2.9 — Schéma de principe du codeur BICM présenté en [94]

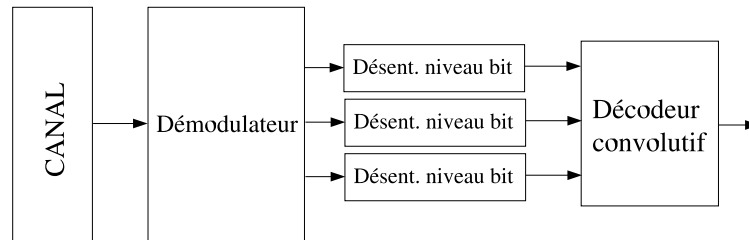


Fig. 2.10 — Schéma de principe du décodeur BICM présenté en [94]

Nous avons vu en Section 2.2.2 que les performances d’une modulation codée sur un canal à évanouissements sont principalement déterminées par le facteur $(E_s/N_0)^L$. Or, il s’avère qu’en raison de la présence des entrelaceurs au niveau bit dans la BICM, l’ordre de diversité, L , coïncide avec d_{\min} . Autrement dit, la distance d_{\min} d’une BICM correspond au nombre de bits différents entre le chemin erroné le plus court et le chemin correct dans le treillis. La BICM conduit alors sur le canal à évanouissements à une probabilité d’erreur plus petite que celle présentée par les MCTs, qui maximisent d_{libre} . Par exemple, le gain apporté par un système BICM (MDP-8, $R = 2/3$) par rapport à un système MCT équivalent est d’environ 3 dB à un BER= 10^{-4} , cet écart devenant plus important à plus faible taux d’erreur [94]. Par contre, les BICMs s’avèrent moins performantes que les MCTs pour des transmissions sur canal gaussien.

2.3 Les modulations turbocodées

Afin de pouvoir profiter des performances remarquables des turbocodes dans des systèmes de communication à bande limitée, nous devons considérer leur association à des modulations à grande efficacité spectrale. L’objectif est ainsi de concevoir des systèmes présentant à la fois un gain de codage important et une grande efficacité spectrale.

2.3.1 Etat de l’art

Nous allons présenter par la suite les principaux schémas d’association turbo-code/modulation qui ont été proposés dans la littérature. Ces techniques peuvent être classifiées comme suit :

- **L’approche pragmatique** introduite par Le Goff [82] en 1994.
- **Les turbo-modulations codées en treillis** : il s’agit d’une extension des MCTs aux turbocodes. Deux différents schémas ont été proposés, correspondant aux travaux de Robertson [63] [64] puis de Benedetto [9] [10].

Les *turbocodes multiniveaux* constituent une autre technique d'association qui a été proposée par Waschmann *et al* dans [90] et appliquée dans [55]. Elle consiste à utiliser des turbocodes comme codeurs élémentaires dans une structure de codage à niveaux multiples [52]. Nous ne décrivons pas cette méthode en détail car elle fait appel à l'utilisation de plusieurs turbocodeurs et décodeurs, présentant une complexité nettement supérieure aux deux techniques précédentes pour des performances similaires.

2.3.2 Principe de l'approche pragmatique

Peu après l'introduction des turbocodes, l'approche pragmatique a été proposée comme première technique d'association d'un turbocode et d'une modulation [82]. Il s'agit d'une technique simple et directe. Son avantage majeur est sa souplesse qui la rend facilement adaptable à différents rendements de codage et ordres de modulation.

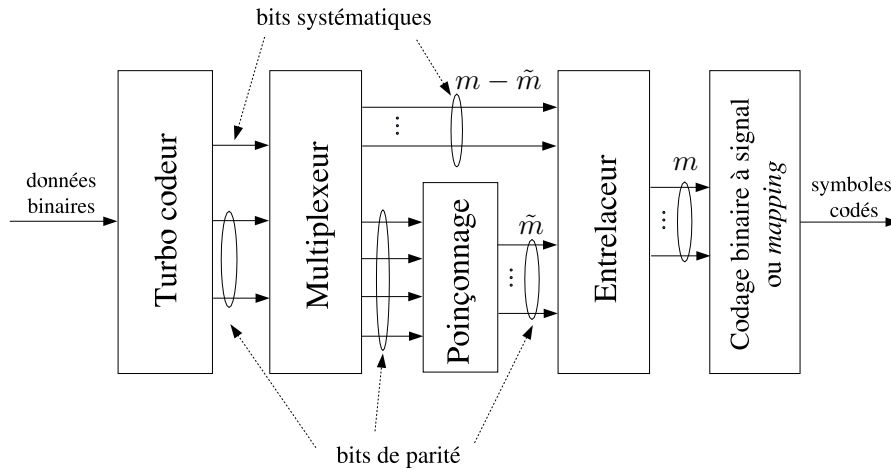


Fig. 2.11 — Schéma de principe d'émission suivant l'approche pragmatique

Le schéma de principe de l'émetteur est représenté sur la figure 2.11. Un seul turbocodeur est nécessaire. Les sorties du codeur sont convenablement poinçonnées et multiplexées afin de récupérer \tilde{m} bits de parité et $m - \tilde{m}$ bits systématiques si la modulation est constituée de $M = 2^m$ signaux. Dans le cas général, m bits codés sélectionnent un des 2^m points de la constellation. Dans le cas particulier d'une MAQ- M à constellation carrée, $m/2$ bits seront affectés à la voie en phase et les autres $m/2$ bits à la voie en quadrature.

Le codage binaire des symboles de la modulation est déterminée par le bloc "Codage binaire à signal", encore appelé *mapping* en anglais. Le *mapping* qui a été proposé pour ce type d'association est un *mapping* de Gray, caractérisé par le fait qu'un seul bit change entre deux symboles adjacents. Le choix de ce type de *mapping* minimise le nombre d'erreurs binaires à l'entrée du turbocodeur ce qui favorise la convergence rapide du processus itératif de décodage. Dans certains constellations, il n'est pas possible de définir un *mapping* de Gray, on utilise alors un *mapping* qui se rapproche le plus proche possible de celui-ci.

En réception (cf. figure 2.12), une estimation pondérée de chacun de bits, \hat{s}_i est calculée à partir du symbole bruité reçu, noté r_t . Cette estimation s'exprime comme le Logarithme de Rapport de Vraisemblances (LRV) normalisé :

$$\hat{s}_i = \frac{\sigma^2}{2} \ln \frac{Pr(s_i = 1 | r_t)}{Pr(s_i = 0 | r_t)} \quad (2.10)$$

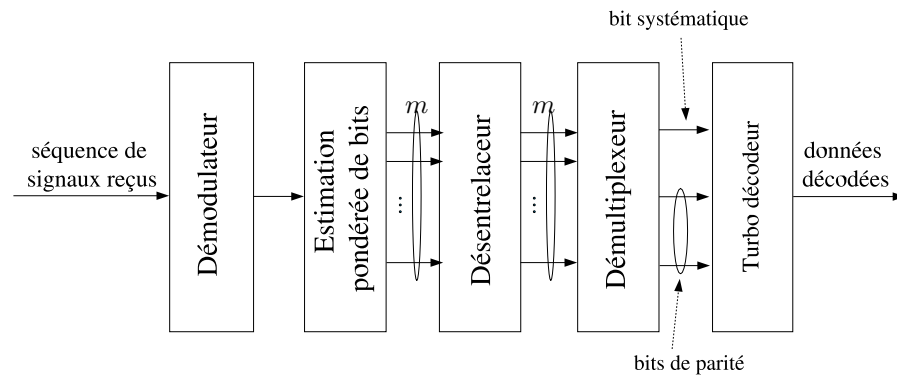


Fig. 2.12 — Schéma de principe d'un récepteur suivant l'approche pragmatique

Dans le cas des modulations MAQs carrées, où les voies en phase et en quadrature peuvent être codées indépendamment (voir exemple de MAQ-16 sur la figure 3.13), les LRVs peuvent être calculés de façon simplifiée si l'on profite du fait que les composantes en phase et en quadrature sont également bruitées de manière indépendante.

Nous donnerons une description plus détaillée des différents blocs constituant les schémas d'émission et de réception dans le Chapitre 3, où nous avons adopté l'approche pragmatique pour les différents cas d'étude.

2.3.3 Principe des turbo-modulations codées en treillis

Les Turbo-Modulations Codées en Treillis (TMCT) associent le principe des MCTs d'Ungerboeck à celui de la concaténation parallèle adopté pour les turbocodes. Cette idée d'association peut être envisagée de plusieurs façons. Deux schémas distincts ont ainsi été proposés, leur différence principale résidant dans la nature des bits poinçonnés. Le premier, présenté par Robertson *et al* [63] [64], propose un poinçonnage des bits de parité uniquement. Le deuxième, proposé par Benedetto *et al* [9] [10], suggère un poinçonnage des bits systématiques.

Schéma de Robertson *et al*

La figure 2.13 présente le schéma de TMCT proposé par Robertson *et al* en 1995 [63]. Le principe du turbocodeur/modulateur consiste à concaténer en parallèle deux codeurs MCTs de rendement $q/(q+1)$, q étant le nombre de bits d'information à l'entrée du codeur. Deux codeurs convolutifs récursifs sont ainsi unis par un entrelaceur de symboles puis suivis d'un codeur binaire à signal. Le message d'information est noté $\mathbf{c} = (c_1, c_2, \dots, c_t, \dots, c_N)$, où c_t est le groupe de q bits d'information produit par la source à l'instant t : $c_t = (c_{t,0}, c_{t,1}, \dots, c_{t,q-1})$. Le codeur transforme la séquence d'entrée en un bloc constitué de N symboles codés : $\mathbf{v} = (v_1, v_2, \dots, v_t, v_N)$ où v_t est un symbole de $(q+1)$ bits : $v_t = (v_{t,0}, v_{t,1}, \dots, v_{t,q})$.

L'entrelaceur permute les symboles d'information, l'ordre des q bits constituant chaque symbole restant inchangé lors de son application en entrée du codeur 2. Il existe la possibilité de ne pas coder l'ensemble des q bits, ce choix dépendant de la constellation, de son *mapping* ainsi que de la valeur du rapport signal à bruit requise [65]. En pratique, cette portion de bits non codés ne serait pas entrelacée.

La séquence codée par le codeur 1 puis modulée par le codeur binaire à signal 1 est notée $\mathbf{x}^u = (x_1^u, \dots, x_t^u, \dots, x_N^u)$ où x_t^u est un symbole dans l'ensemble de signaux de la modulation.

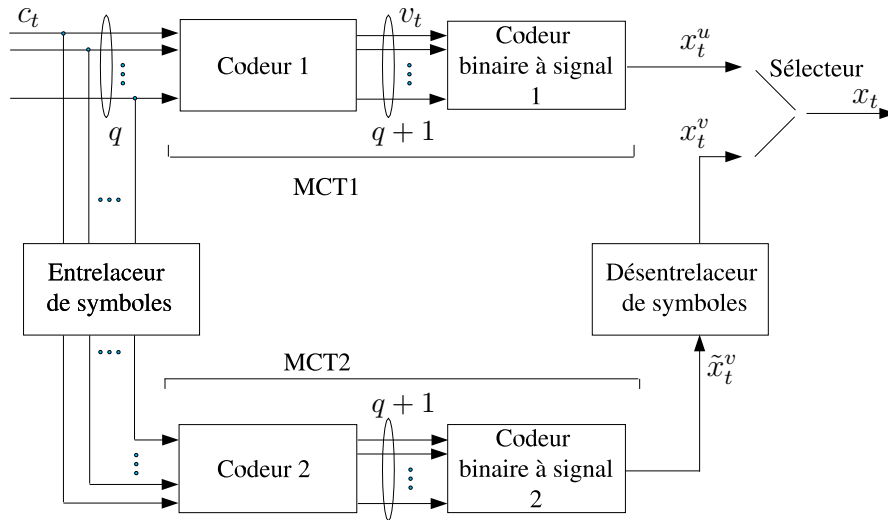


Fig. 2.13 — Schéma de principe d'un turbocodeur/modulateur en treillis avec poinçonnage de bits de parité

La modulation est composée de 2^{q+1} points, donnés par : $x_t^u = x_{t,I}^u + x_{t,Q}^u$ où $x_{t,I}^u$ et $x_{t,Q}^u$ sont, respectivement, les composantes en phase et en quadrature du signal x_t^u . Le *mapping* utilisé dans le bloc “Codeur binaire à signal” est le *Set-Partitioning* d’Ungerboeck [85], présenté en Section 2.2.3.

La séquence de symboles codés issue du codeur binaire à signal 2, notée $\tilde{\mathbf{x}}^v$, est désentrelacée avant l’émission, devenant ainsi la séquence \mathbf{x}^v . On constate que les q bits d’information sont contenus en même temps dans les symboles codés appartenant à \mathbf{x}^u et à \mathbf{x}^v . Afin d’éviter une double transmission d’information, on utilise un sélecteur qui permet de transmettre alternativement les deux sorties. Pour que ce type de schéma atteigne un débit de q bits/sec/Hz, une constellation à 2^{q+1} points doit être utilisée.

Schéma de Benedetto *et al*

Le schéma TMCT proposé par Benedetto *et al* en 1996 [10] présente deux différences principales par rapport au schéma de Robertson. La première est l’utilisation d’entrelaceurs au niveau bit au lieu d’entrelaceurs au niveau symbole. La deuxième est le poinçonnage de bits systématiques et non des bits de parité.

La figure 2.14 montre le schéma général d’une TMCT selon cette approche. Il s’agit de concaténer en parallèle deux codeurs convolutifs récurrents avec poinçonnage des bits systématiques. Le poinçonnage effectué à l’entrée de chaque codeur est conçu d’une façon telle que l’ensemble de symboles de sortie du code concaténé contiennent toute l’information une seule fois. Les codeurs élémentaires sont de rendement $R = q/(q + 1)$ et fournissent un bit de parité pour q bits d’information. Deux bits de parité sont ainsi obtenus et émis, chacun dans son symbole respectif. Pour limiter le nombre de points de la constellation (qui serait *a priori* de 2^{q+1}) à $2^{q/2+1}$, chaque bit d’information n’est utilisé qu’une seule fois dans les blocs “codeur binaire à signal” : les $q/2$ premiers bits sont introduits dans le “Codeur binaire à signal” supérieur et les $q/2$ autres bits dans le codeur inférieur. Néanmoins, les q bits affectent les états des deux codeurs. L’efficacité spectrale du système ainsi conçu est $q/2$ bit/s/Hz.

Le critère d’optimisation proposé pour ce type de modulation turbocodée utilise un nouveau paramètre caractéristique : la *distance euclidienne libre effective*. Elle est définie comme

la distance euclidienne minimale entre deux séquences codées dont les séquences d'entrée ont une distance de Hamming égal à 2. Pour optimiser les performances de ce type de structure, les codeurs élémentaires doivent être conçus de façon à maximiser la distance euclidienne libre effective de la modulation turbocodée pour une modulation et un *mapping* donnés. L'étude présentée dans [10] considère trois types de *mapping* : Gray, *Set-Partitioning* ainsi qu'un nouveau *mapping* dénommé *reordered mapping* par les auteurs.

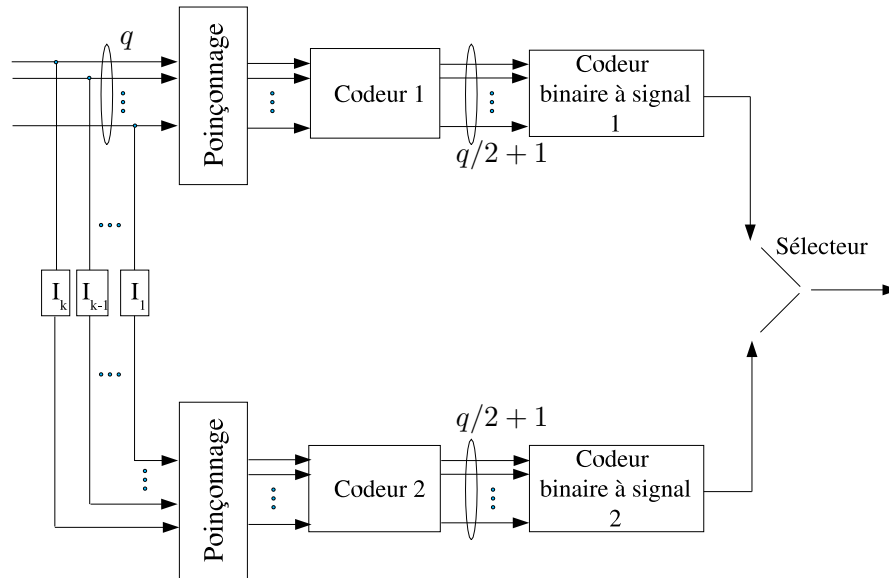


Fig. 2.14 — Schéma de principe d'un turbocodeur/modulateur en treillis selon l'approche de Benedetto *et al*

Nous allons étudier deux exemples pour illustrer ce type de TMCT. Le premier considère une modulation MDP-8. Il est présenté sur la figure 2.15. Dans ce schéma, pour obtenir un rendement de codage de 2/3, on utilise $q = 4$, l'efficacité spectrale du système étant égale à 2 bit/s/Hz.

Un deuxième exemple est donné sur la figure 2.16. Il s'agit d'une turbo MAQ-16 codée en treillis avec poinçonnage des bits systématiques. Le principe est le même que dans l'exemple précédent si ce n'est que dans le cas d'une MAQ- M carrée, il est possible d'affecter les $1 + q/2$ bits issus du codeur supérieur à la voie en phase et les $1 + q/2$ bits issus du codeur inférieur à la voie en quadrature, l'efficacité spectrale résultante étant égale à $q = 2$ bit/s/Hz.

2.3.4 Comparaison de performances

D'après les résultats présentés dans [61], on peut établir une comparaison de performances entre l'approche pragmatique et le schéma TMCT de Robertson. Pour le cas d'une MDP-8 turbocodée et pour une valeur de $BER=10^{-4}$, les performances obtenues avec l'approche pragmatique avec un codeur à 8 états et un algorithme de décodage SOVA sont entre 0,5 et 0,7 dB moins bonnes que celles du schéma turbo MDP-8 codée en treillis équivalent utilisant l'algorithme MAP pour le décodage. Cet écart dépend de la taille de bloc considérée, la valeur maximale de 0,7 dB correspondant à une taille de bloc d'environ 2000 bits.

Les résultats présentés par Benedetto *et al* dans [9] pour une MDP-8 et un rendement de codage 2/3 montrent que la MCT concaténée en parallèle apporte à un $BER=10^{-6}$ un gain de 1 dB par rapport à la courbe obtenue avec l'approche pragmatique pour une même valeur

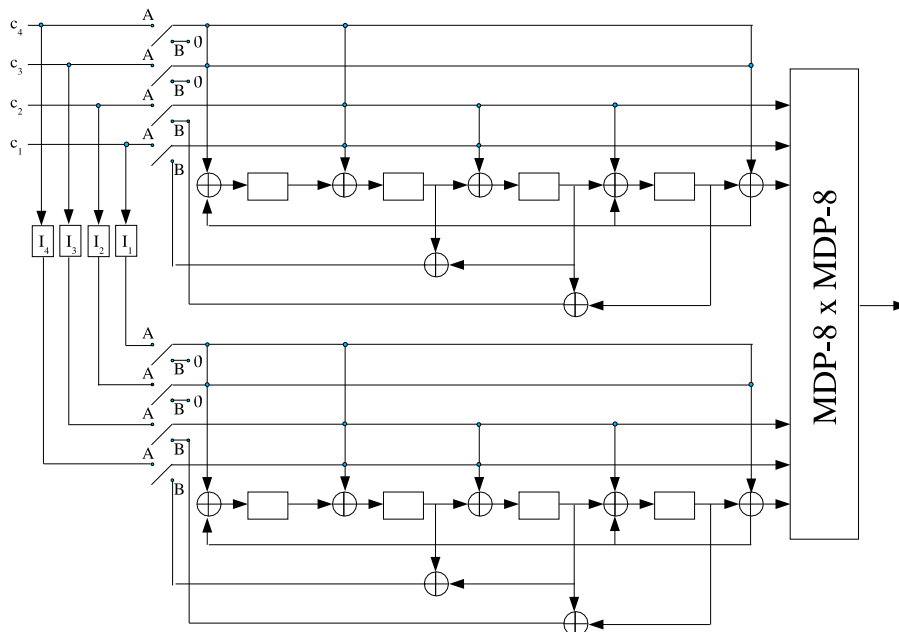


Fig. 2.15 — Exemple de turbo MDP-8 codée en treillis avec poinçonnage de bits systématiques. $q = 4$, $\eta = 2$ bit/s/Hz.

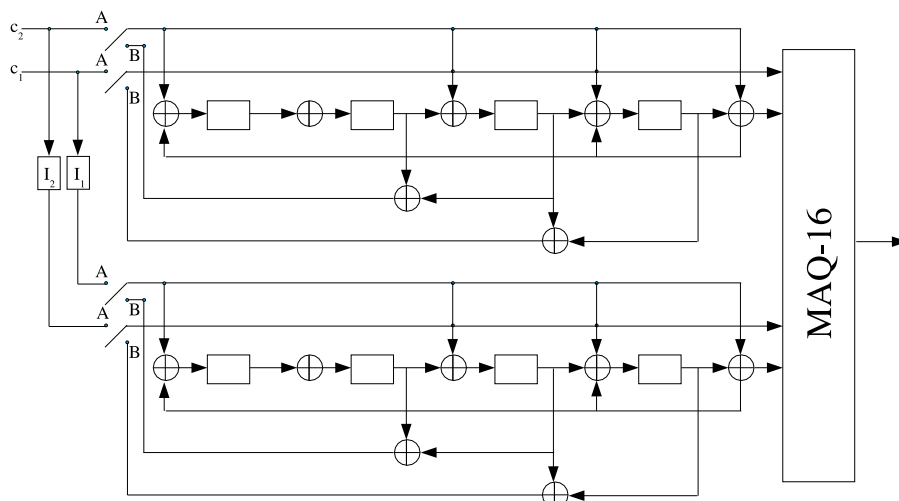


Fig. 2.16 — Turbo MAQ-16 codée en treillis avec poinçonnage de bits systématiques. $q = 2$, $\eta = 2$ bit/s/Hz.

d'efficacité spectrale. On notera que ce résultat est obtenu sous les conditions suivantes : la MCT concaténée en parallèle utilise un décodage MAP, cinq itérations de décodage et quatre entrelaceurs élémentaires de 4096 bits tandis que le schéma pragmatique considéré utilise l'algorithme sous-optimal SOVA et un seul entrelaceur de 4096 bits.

De manière générale on peut affirmer que, en ce qui concerne les travaux présentés de 1994 à 1998, les performances d'une modulation turbocodée suivant le principe de l'approche pragmatique sont de 0,5 dB à 1 dB moins bonnes que celles qui peuvent être obtenues avec un schéma TMCT spécialement conçu pour la même modulation et rendement de codage. Néanmoins, l'approche pragmatique présente une série d'avantages, à savoir : simplicité, adaptation aisée à différentes constellations et divers rendements de codage, compatibilité avec les turbocodes binaires car même décodeur... qui la rendent plus intéressante pour des systèmes demandant une certaine souplesse d'utilisation. D'autre part, la mise en œuvre dans un schéma suivant l'approche pragmatique d'algorithmes de décodage plus performants que ceux originellement utilisés par Le Goff doit permettre de gagner jusqu'à 0,3-0,4 dB en performance et par conséquent de permettre à cette approche simple de concurrencer les méthodes *ad hoc* de TMCT.

2.4 Conclusion

Nous avons d'abord présenté dans ce chapitre les turbocodes, famille de codes correcteurs d'erreurs quasi-optimaux basée sur la concaténation parallèle de codes convolutifs systématiques récurrents, binaires ou m -binaires. Nous avons décrit la structure du codeur, le rôle de l'entrelacement ainsi que le principe du turbo-décodage. Après avoir donné quelques éléments de performances des turbocodes, nous avons expliqué comment les différents critères de conception peuvent affecter ces performances.

En associant ces codes puissants à des modulations d'ordre élevé, des communications de qualité occupant une bande spectrale limitée deviennent réalisables. Les principales techniques de modulations turbocodées proposées dans la littérature ont été présentées. Après avoir comparé et évalué ces différents schémas, nous avons finalement retenu la technique d'association dite pragmatique en raison de sa grande souplesse d'utilisation. D'autre part, les turbocodes m -binaires étant particulièrement attractifs pour des rendements de codage élevés, leur utilisation dans des schémas de modulations turbocodées pragmatiques apparaît prometteur. Il s'agit là du premier volet de mon travail de thèse décrit dans le chapitre suivant.

Etude de modulations turbocodées pragmatiques à base de codes duo-binaires

La première partie de notre étude a consisté à adapter le travail effectué par Le Goff aux turbocodes duo-binaires, en tirant parti des améliorations apportées depuis 1995 aux algorithmes de décodage (remplacement du décodage SOVA par des algorithmes dérivés du MAP). Nous nous sommes, en pratique, intéressées aux modulations MAQ-16, MAQ-64, MDP-8 et MDAP-16.

Dans ce chapitre nous décrivons tout d'abord les structures des codeurs et décodeurs utilisés. Nous présentons ensuite le schéma général d'association pragmatique à partir duquel nous détaillons les caractéristiques propres à chaque cas d'étude. Pour les modulations de type MAQ, nous avons étudié l'influence de la construction des symboles de modulation à partir des bits codés sur les performances des modulations turbocodées considérées. Pour les modulations MDP et MAQ, les tailles de blocs et rendements étudiés sont représentatifs d'applications concrètes dans le secteur des technologies sans fils (par exemple, norme IEEE 802.16, Wireless Metropolitan Area Network).

3.1 Description et performances des turbocodes convolutifs duo-binaires étudiés

Deux TurboCodes Convolutifs (TCC) duo-binaires ont été considérés dans le cadre de notre étude, différant par le nombre d'états des codeurs élémentaires. Les avantages des TCCs duo-binaires ont déjà été présentés en Section 2.1.2. Dans cette Section nous allons présenter les structures de codage et de décodage ainsi que quelques éléments de performances concernant ces codes.

3.1.1 Structure des turbocodeurs

Les figures 3.1 et 3.2 présentent les structures respectives des turbocodeurs à 8 et 16 états étudiés. Le premier codeur a été adopté pour le standard DVB-RCS/RCT [78] [79]. Il est construit autour d'un codeur convolutif systématique récursif (CSR) duo-binaire à 8 états de polynômes $(15, 13)_{octal}$. Le deuxième codeur est une extension du premier, dénommé TURBO2000 et basé sur un codeur CSR duo-binaire à 16 états de polynômes $(23, 35)_{octal}$.

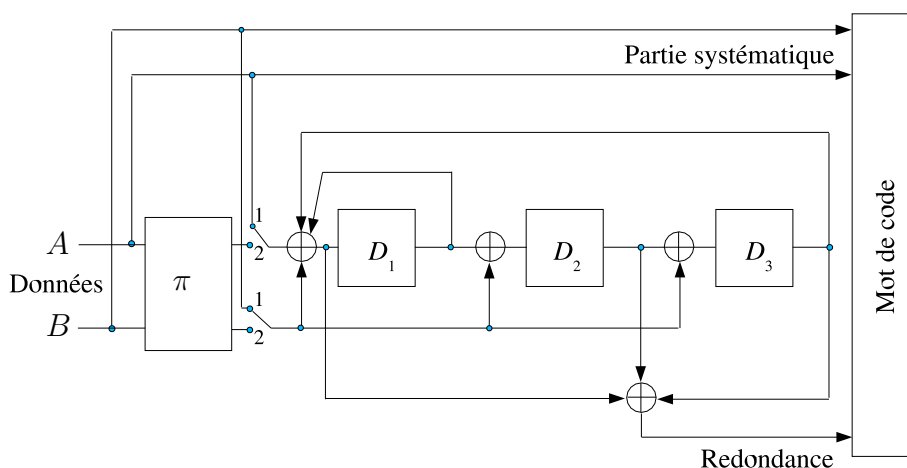


Fig. 3.1 — Codeur duo-binaire à 8 états

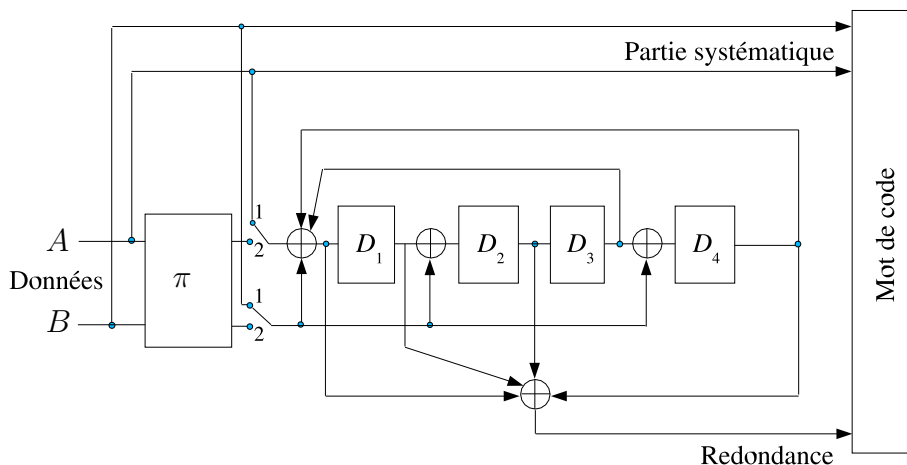


Fig. 3.2 — Codeur duo-binaire à 16 états

Dans les deux structures, les données sont tout d'abord codées dans l'ordre naturel (commutateurs en position 1). Elles sont ensuite codées une seconde fois dans un ordre entrelacé fourni par la permutation temporelle π (commutateur en position 2). Les codes élémentaires font alors appel à deux treillis indépendants, l'un pour l'ordre naturel et l'autre pour l'ordre entrelacé.

L'entrelaceur

L'entrelacement utilisé dans notre étude tire pleinement partie du fait que le codeur est duo-binaire. En effet la permutation est définie à deux niveaux, le premier entre couples (permutation *inter-symbole*) et le seconde à l'intérieur du couple (permutation *intra-symbole*).

Permutation inter-symbole : elle est définie comme suit :

- Pour le codeur à 8 états : on adopte l'équation d'entrelacement des standards DVB-RCS/RCT [78] [79] :

$$i = \pi(j) = (P_0 \cdot j + P(j) + 1) \text{ modulo } N \quad (3.1)$$

avec :

- si $j \text{ modulo } 4 = 0 \Rightarrow P(j) = 0$
- si $j \text{ modulo } 4 = 1 \Rightarrow P(j) = N/2 + P_1$
- si $j \text{ modulo } 4 = 2 \Rightarrow P(j) = P_2$
- si $j \text{ modulo } 4 = 3 \Rightarrow P(j) = N/2 + P_3$
- Pour le codeur à 16 états : on adopte l'équation d'entrelacement suivante :

$$i = \pi(j) = (P_0 \cdot j + P(j) + 1) \text{ modulo } N \quad (3.2)$$

avec :

- si $j \text{ modulo } 4 = 0 \Rightarrow P(j) = 0$
- si $j \text{ modulo } 4 = 1 \Rightarrow P(j) = P_1$
- si $j \text{ modulo } 4 = 2 \Rightarrow P(j) = 4P_0 + P_2$
- si $j \text{ modulo } 4 = 3 \Rightarrow P(j) = 4P_0 + P_3$

où :

- P_0, P_1, P_2 et P_3 sont des paramètres qui sont fonction de la taille du bloc à traiter.
- $i = 0, \dots, N - 1$ est l'adresse de la mémoire où sont rangés les couples dans l'ordre naturel (celui du premier codage).

Permutation intra-symbole : le contenu d'un couple sur deux est inversé. Cette permutation est définie comme suit :

$$\pi_{intra-symbole} = \begin{cases} (A, B) \text{ devient } (B, A) & \text{si } j \text{ modulo } 2 = 0 \\ (A, B) \text{ reste } (A, B) & \text{si } j \text{ modulo } 2 = 1 \end{cases} \quad (3.3)$$

où $j = 0, \dots, N - 1$ est incrémenté à chaque opération de lecture de données lors du second codage et N est le nombre de couples dans un bloc (*i.e.* on utilise des blocs de $k = 2N$ bits de données).

Pour chaque taille de bloc, les paramètres P_0, P_1, P_2 et P_3 sont choisis de façon à ce que les performances soient aussi bonnes pour une large gamme de rendements qu'elles sont pour le rendement naturel, obtenant ainsi un schéma de codage le plus générique possible. Nous avons en effet observé en pratique qu'une permutation conférant un bon comportement asymptotique pour le rendement naturel est en général également performante pour un autre rendement dont il est déduit par poinçonnage. Les valeurs des paramètres P_0, P_1, P_2 et P_3

correspondant aux tailles de bloc considérées dans l'étude sont données sur le tableau 3.1 pour le codeur à 16 états (TURBO2000), le lecteur pouvant se référer à [38] pour le codeur à 8 états.

Les équations définissant la permutation inter-symbole sont basées sur un entrelacement régulier, $i = (P_0 \cdot j + P + 1) \text{ modulo } N$, permettant de casser les motifs d'erreur de poids 2 et 3. De plus, l'introduction d'un certain désordre contrôlé avec $P(j)$ permet également de casser les motifs d'erreurs composites (par exemple rectangulaires).

D'autre part, la transformation par la fonction d'entrelacement inter-symbole des positions paires en positions impaires et *vice versa* garantit une bonne répartition de la redondance vis à vis des couples de données systématiques, si un poinçonnage identique de la redondance est appliqué sur les deux codes (voir paragraphe suivant).

| Taille de bloc (octets) | P_0 | P_1 | P_2 | P_3 |
|-------------------------|-------|-------|-------|-------|
| 54 | 13 | 4 | 4 | 4 |
| 188 | 35 | 6 | 4 | 10 |

Tab. 3.1 — Valeurs des paramètres d'entrelacement inter-symbole dans le codeur TURBO2000 pour les différentes tailles de bloc utilisées

Rendement de codage et poinçonnage

Les codeurs présentés (figures 3.1 et 3.2) fournissent des mots de code constitués de :

- la partie systématique (*i.e.* un couple de données),
- la redondance du codage dans l'ordre naturel (1 bit par couple),
- la redondance du codage dans l'ordre entrelacé (1 bit par couple).

Les rendements considérés dans l'étude sont le rendement naturel du code, $R = 1/2$, ainsi que les rendements $3/5$, $2/3$ et $3/4$. Si Y_1 et Y_2 sont les bits de redondance issus du codage des données dans l'ordre naturel (commutateur en position 1 dans les figures 3.1 et 3.2) et entrelacé (commutateur en position 2), les motifs de poinçonnage pour les rendements retenus sont :

- $R = 3/5$: pour $1 \leq j \leq N$, on garde Y_1 et Y_2 aux places $j = 1$ et $2 \text{ modulo } 3$.
- $R = 2/3$: pour $1 \leq j \leq N$, on garde Y_1 et Y_2 à la place $j = 1 \text{ modulo } 2$.
- $R = 3/4$: pour $1 \leq j \leq N$ on garde Y_1 et Y_2 aux places $j = 1 \text{ modulo } 3$.

Notons que, l'entrelacement transformant les positions paires en impaires, les redondances poinçonnées par un codeur ne sont pas poinçonnées par l'autre, ce qui conduit à un schéma de codage plus robuste.

Codage circulaire

Dans le turbocodeur, les deux opérations de codage font appel à un codage circulaire [15], c'est-à-dire que l'état initial du codeur au début du codage d'une séquence d'information et son état final coïncident. Les avantages du codage circulaire du point de vue du décodage itératif sont décrits en Section 3.1.2. Le principe des codes convolutifs circulaires est présenté en détail dans l'Annexe A.

Le codage de chaque bloc conduit alors à la construction de deux codes circulaires. La construction de tels codes nécessite un pré-traitement dont le rôle est de déterminer l'état de circulation. Ce pré-traitement consiste à :

1. initialiser le codeur dans l'état *tout à zéro*, noté 0.

2. alimenter une première fois le codeur par les données à coder sans produire de redondance.
3. déterminer l'état de circulation en fonction de l'état final à l'aide des tableaux fournis en Annexe A.

Pour obtenir ensuite la séquence codée il faut :

1. initialiser le codeur dans l'état de circulation.
2. alimenter une seconde fois le codeur par les données à coder pour produire la redondance.

3.1.2 Structure du turbo-décodeur

Principe du décodage itératif des turbocodes duo-binaires

La figure 3.3 présente le schéma de principe d'un turbo-décodeur duo-binaire. Il est constitué de 2 décodeurs élémentaires DEC1 et DEC2 associés aux deux opérations de codage.

A_{br} , B_{br} , Y_{1br} et Y_{2br} sont les symboles bruités issus du bloc démodulateur. Les deux premiers correspondent aux bits systématiques et les deux derniers aux bits de redondance. Les valeurs de ces symboles en absence de bruit sont "+1" ou "-1". S'il s'agit d'un code poinçonné, les symboles manquants sont alors remplacés par la valeur neutre "0".

A chaque itération de décodage, chaque décodeur fournit une information extrinsèque Z pour les quatre valeurs possibles de couple de données (A, B) , à savoir $(0, 0)$, $(0, 1)$, $(1, 0)$ et $(1, 1)$. Cette information est utilisée comme information *a priori* à l'entrée de l'autre décodeur. Le décodeur DEC1, qui décode les données dans l'ordre naturel, fournit également à chaque itération les décisions pondérées L relatives aux quatre valeurs de (A, B) , dont sont déduites les estimations binaires de A et B .

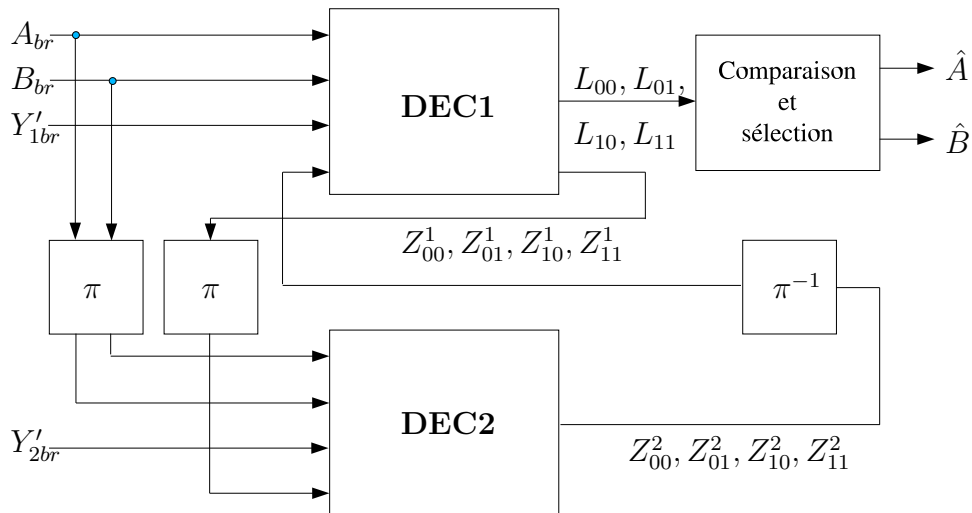


Fig. 3.3 — Principe du turbo-décodage d'un code duo-binaire

En pratique, pour chaque bloc de données codées, le séquençage du décodage mis en œuvre est le suivant :

1. Les informations extrinsèques Z^1 sont mises à zéro.
2. Le décodeur DEC2 effectue le décodage du bloc à l'aide des redondances Y_{2br} et fournit, pour chaque couple de données, une information extrinsèque quaternaire Z^2 .

3. Le décodeur DEC1 effectue ensuite le décodage du bloc à l'aide des redondances Y_{1br} et des informations extrinsèques Z^2 issues de l'étape précédente de décodage et préalablement désentrelacées. Il fournit pour chaque couple de données une information extrinsèque quaternaire Z^1 et une décision pondérée quaternaire L , dont sont déduites les décisions binaires \hat{A} et \hat{B} .
4. Le décodeur DEC2 décode à nouveau le bloc à l'aide des redondances Y_{2br} et des informations extrinsèques Z^1 issues de l'étape précédente de décodage et préalablement entrelacées. Il fournit, pour chaque couple de données une nouvelle information extrinsèque quaternaire Z^2 . Retour à l'étape 3.

Ce processus est itéré autant de fois que nécessaire pour obtenir les performances de correction visées. Une itération est constituée d'un décodage du bloc de données par DEC2 puis par DEC1, ce qui permet de récupérer à la fin d'une itération complète les données dans l'ordre naturel en sortie de DEC1.

L'algorithme de décodage MAP adapté aux turbocodes m -binaires est décrit en Annexe B. La version simplifiée du MAP que nous utilisons dans notre étude, nommée Max-Log-MAP y est également décrite.

Décodage des turbocodes circulaires

Dans le cas de l'application des algorithmes MAP ou Max-Log-MAP, le décodage du code consiste à parcourir le treillis circulaire correspondant dans le sens trigonométrique direct pour le traitement arrière, et dans le sens inverse pour le traitement avant (cf. figure 3.4). L'élaboration des décisions et la construction des informations extrinsèques sont réalisées lors du processus avant pour récupérer les décisions dans l'ordre. Pour chacun des deux traitements, les probabilités ou métriques calculées à la fin de chaque tour de treillis sont utilisées comme valeurs initiales pour le tour suivant. Le nombre de tours effectués autour du cercle est égal au nombre d'itérations requis par le processus de décodage.

L'état de départ des traitements arrière et avant (*i.e.* l'état de circulation calculé lors du codage) est obtenu à l'issue d'une étape de pré-décodage, appelée *prologue* et appliquée sur une profondeur de quelques ν (ν étant la mémoire du code). Aucune information extrinsèque ni aucune décision n'est calculée pendant cette étape.

Deux treillis circulaires de profondeur N sont considérés pour le turbo-décodage, correspondant au décodage des données non entrelacées (code 1) et au décodage des données entrelacées (code 2).

3.1.3 Performances des turbocodes à 8 et 16 états sur canal gaussien

Nous présentons dans un premier temps quelques éléments de performances du turbo-code DVB-RCS/RCT associé à une modulation binaire, par exemple une MDP-2. La figure 3.5 (resp. 3.6) présente ses performances de correction pour une taille de bloc de 53 octets (resp. 188) correspondant à la transmission de blocs ATM (resp. MPEG). Ces résultats correspondent aux mesures obtenues avec le *hardware* TC1000, turbo-décodeur conçu et commercialisé par la société TurboConcept ¹. L'algorithme de décodage est le Max-Log-MAP (cf. Annexe B). Les échantillons reçus sont quantifiés sur quatre bits. Pour les rendements inférieurs à 1/2, une seconde redondance est produite par chaque codeur élémentaire.

¹<http://www.turboconcept.com>

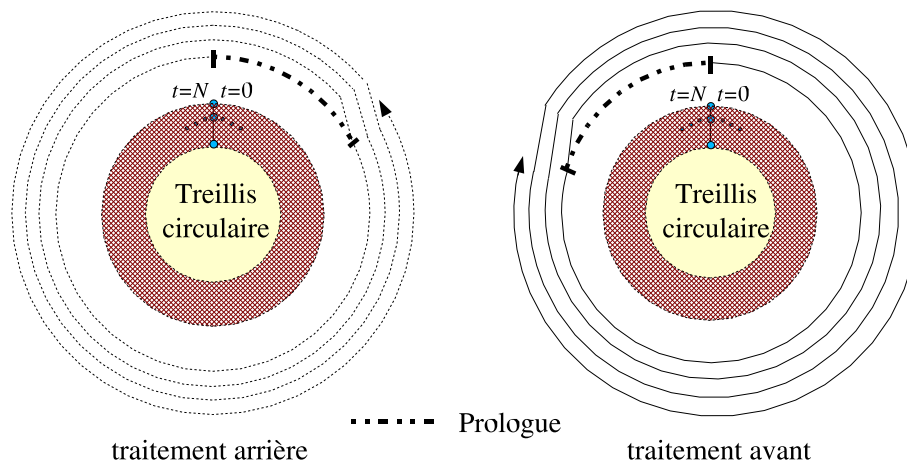


Fig. 3.4 — Traitement d'un code circulaire par l'algorithme MAP ou Max-Log-MAP

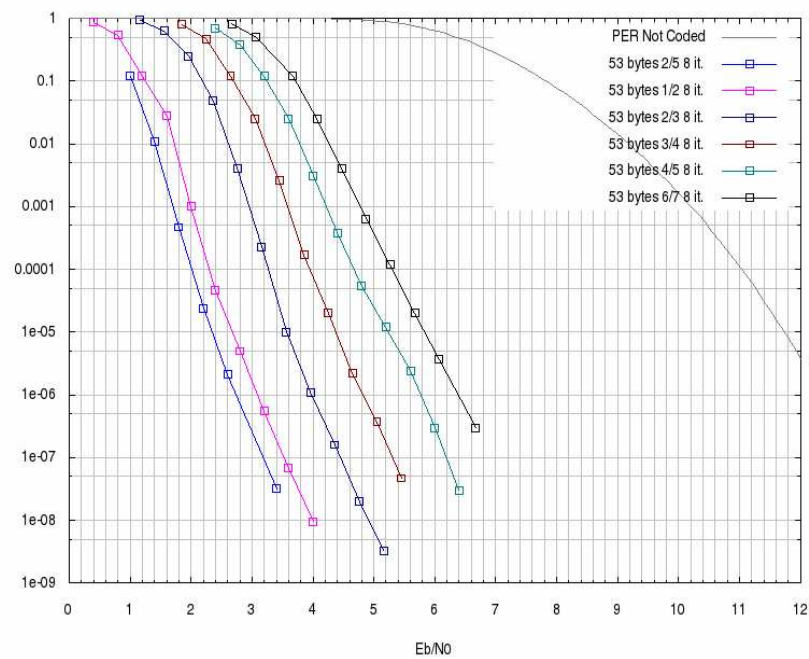


Fig. 3.5 — Taux d'erreurs de trames pour des blocs ATM (53 octets) du turbo code duo-binaire à 8 états avec modulation MDP-2. Mesures *hardware* du décodeur TC1000. 8 itérations de décodage, 4 bits de quantification à l'entrée du décodeur. Algorithme de décodage Max-Log-MAP.

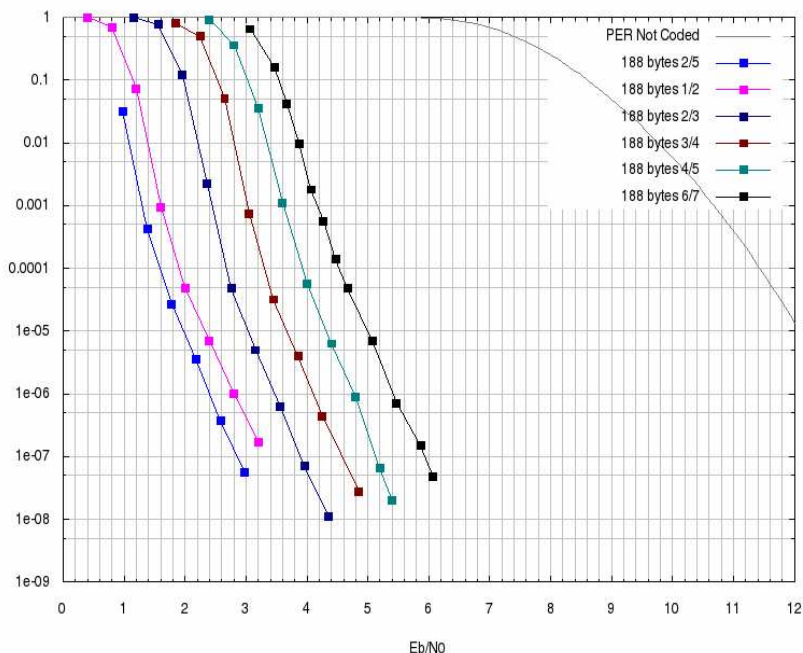


Fig. 3.6 — Taux d'erreurs de trames pour des blocs MPEG (188 octets) du turbocode duo-binaire à 8 états avec modulation MDP-2. Mesures *hardware* du décodeur TC1000. 8 itérations de décodage, 4 bits de quantification à l'entrée du décodeur. Algorithme de décodage Max-Log-MAP.

Le tableau 3.2 présente quelques valeurs de l'écart entre ces courbes de performance et les limites théoriques de correction. Ces limites théoriques ont été obtenues avec l'outil disponible en [2], la taille de bloc étant prise en compte dans leur calcul. Cet ensemble de résultats montre la bonne convergence du turbocode : à un $\text{FER}=10^{-4}$, les résultats sont au plus à 0,9 dB de la limite théorique pour tous les rendements considérés. Une dégradation des pentes des courbes pour des valeurs de $\text{FER} \leq 10^{-5}$ due à une distance minimale limitée fait qu'à un $\text{FER}=10^{-7}$ les résultats se situent de 1,4 à 2 dB des limites théoriques. Notons que la dégradation en performances liée à la quantification des données sur 4 bits est estimée à environ 0,15 dB.

| FER | 10^{-4} | | 10^{-7} | |
|-----------|-----------|-----|-----------|-----|
| | 53 | 188 | 53 | 188 |
| $R = 1/2$ | 0,7 | 0,9 | 1,4 | 2,0 |
| $R = 2/3$ | 0,7 | 0,8 | 1,4 | 1,9 |
| $R = 3/4$ | 0,8 | 0,8 | 1,6 | 1,8 |

Tab. 3.2 — Valeurs en dB de l'écart entre performances effectives (8 itérations de décodage, 4 bits de quantification, algorithme Max-Log-MAP) et limites théoriques pour le turbocode du standard DVB-RCS/RCT

L'un des avantages majeurs de ce turbocode est sa souplesse. En effet, il a été conçu pour être facilement adaptable à différentes valeurs de taille de bloc et de rendement de codage. Pour adapter le codeur à la taille de bloc utilisée, il suffit de changer la valeur des paramètres P_0 , P_1 , P_2 et P_3 dans l'entrelaceur π , la seule contrainte étant que, pour rendre le code circulaire, cette taille ne peut pas être un multiple de 7. Pour adapter le codeur au

rendement requis, il suffit de définir un motif de poinçonnage adapté à ce rendement. On observe sur le tableau 3.2 que l'écart entre les courbes et les limites théoriques varie peu avec le rendement de codage, ceci indiquant que les différents motifs de poinçonnage n'affectent pas les bonnes performances du code original.

La figure 3.7 montre une comparaison de performances entre le turbocode à 8 états et celui à 16 états pour un même rendement de codage $R = 2/3$. Le gain en performance asymptotique apporté par le codeur à 16 états ($K = 5$) par rapport au codeur à 8 états ($K = 4$) est de presque 1 dB à un FER= 10^{-7} .

Finalement, la figure 3.8 présente un ensemble de courbes de simulation correspondant au codeur TURBO2000 avec rendement 3/4 pour deux différentes tailles de bloc. Le tableau 3.3 montre la distance en dB entre ces courbes et les limites théoriques [2] pour deux valeurs représentatives de FER. On remarque l'amélioration des performances asymptotiques par rapport aux résultats obtenus avec le codeur à 8 états : à un FER= 10^{-7} un gain d'environ 1 dB est introduit.

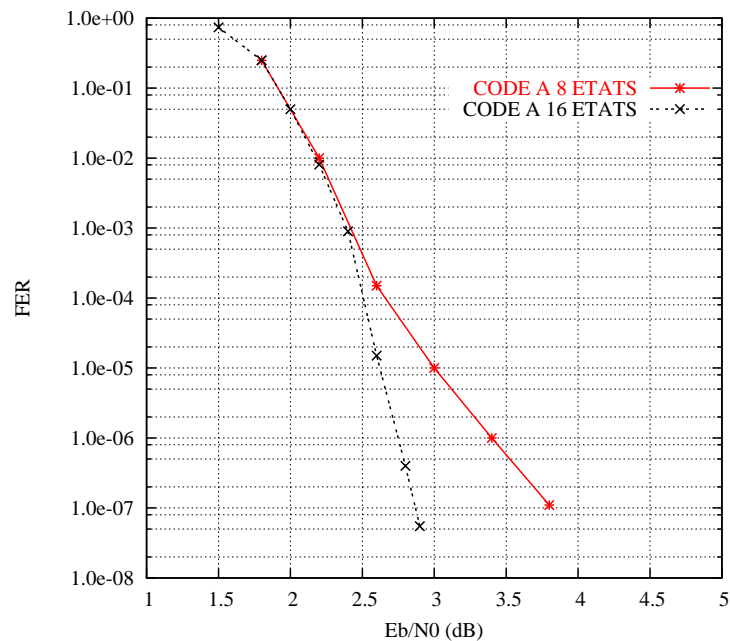


Fig. 3.7 — Résultats de simulation sur canal gaussien. TCC duo-binaire à 8 états et TCC duo-binaire à 16 états, avec modulation MDP-2. Taille de bloc : 188 octets. $R=2/3$. 8 itérations de décodage, 6 bits de quantification, algorithme Max-Log-MAP.

| Taille de bloc (octets) | FER | |
|-------------------------|-----------|-----------|
| | 10^{-4} | 10^{-7} |
| 54 | 0,55 | 0,9 |
| 188 | 0,5 | 0,8 |

Tab. 3.3 — Valeurs en dB de l'écart entre performances effectives (8 itérations de décodage, 6 bits de quantification, algorithme Max-Log-MAP) et limites théoriques pour le turbocode TURBO2000, pour des différentes tailles de bloc. $R = 3/4$

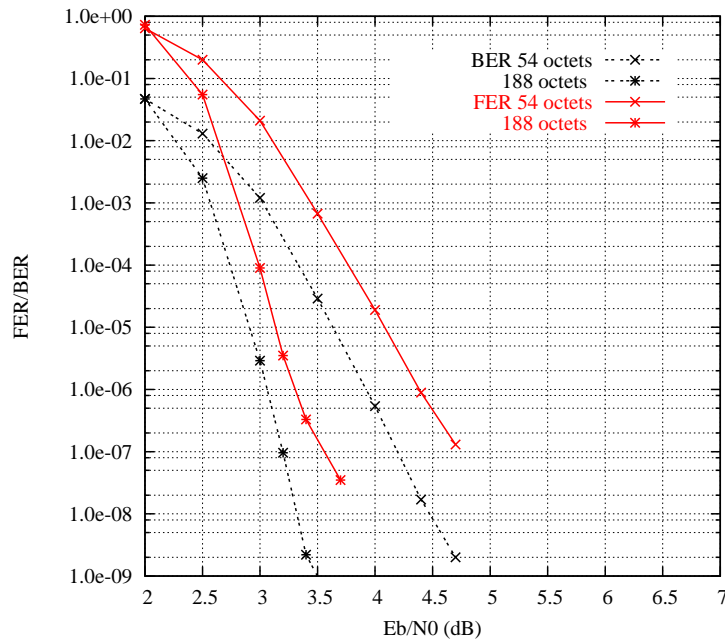


Fig. 3.8 — Résultats de simulation sur canal gaussien. Turbocode convolutif duo-binaire à 16 états. $R=3/4$. 8 itérations de décodage, 6 bits de quantification, algorithme Max-Log-MAP.

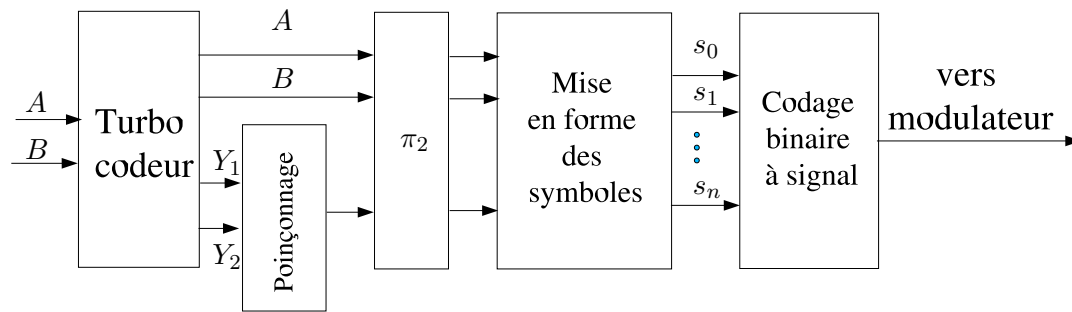
3.2 Modulation turbocodée pragmatique à base de codes duo-binares : cas général

Le point de départ de notre étude est le travail de thèse de Le Goff [80] où l’approche pragmatique, précédemment présentée en Section 2.3.2 a été introduite. Cette approche est à la base des schémas que nous avons adopté, bien que pour obtenir de meilleures performances nous avons introduit le turbocodage duo-binaire. De plus, nous avons étudié de manière plus approfondie l’influence de la distribution des bits codés dans les symboles de modulation.

Nous allons tout d’abord présenter le schéma général de principe de l’association pragmatique d’un turbocode convolutif duo-binaire et d’une modulation à M points, du point de vue de l’émission et de la réception. Ensuite, ces schémas seront particularisés pour chacun des cas étudiés.

3.2.1 Schéma d’émission

La figure 3.2.1 montre le schéma que nous avons adopté dans le cadre de cette approche, où n désigne le nombre de bits dans un symbole de la modulation. Dans ce schéma, les blocs correspondant au turbocodeur et poinçonnage ont déjà été décrits en Section 3.1.1. Le “Codage binaire à signal” correspond à un *mapping* de Gray. D’autre part, les blocs “Entrelaceur π_2 ” et “Mise en forme de symboles” sont détaillés ci-après.



Permutation temporelle π_2

La fonction de permutation temporelle π_2 insérée en sortie du codeur n'est utilisée en pratique que pour les transmissions sur canal à évanouissements de Rayleigh. On suit ainsi le principe de la BICM (cf. Section 2.2.4) en évitant que les bits issus d'un même instant de codage ne soient dans le même symbole émis sur le canal et donc simultanément affectés par un même évanouissement.

En pratique, on applique deux fonctions d'entrelacement régulier distinctes et indépendantes sur les bits systématiques : la première sur les bits A , la seconde sur les bits B . L'équation générique d'entrelacement est la suivante : $i = (P_0 \times j) \text{ modulo } N$ où $i = 0, \dots, N - 1$ est l'adresse de la mémoire où sont rangés les bits dans l'ordre naturel. Pour les bits A , nous avons choisi $P_0 = 11$ et pour les bits B , $P_0 = 23$.

Mise en forme des symboles

Ce bloc constitue la charnière entre le code et la modulation. Il a pour rôle de placer les bits issus du turbocodeur par groupes de n bits (après entrelacement et poinçonnage) pour qu'ils soient ensuite convertis en symboles de modulation dans le bloc "Codage binaire à signal". Le principe de construction des symboles dépend du rendement de code ainsi que de la modulation considérés, ainsi qu'il sera décrit pour chacun des cas d'étude considérés. D'un point de vue général, deux stratégies basiques de construction de symboles peuvent être envisagées :

Schéma A : On affecte en priorité les places les mieux protégées par la modulation aux bits systématiques.

Schéma Y : On affecte en priorité les places les mieux protégées par la modulation aux bits de redondance.

Le choix d'une stratégie ou d'une autre conditionne les performances à faible et à fort rapport signal à bruit ainsi que nous le verrons plus loin.

3.2.2 Schéma de réception

La figure 3.9 donne le schéma de principe du récepteur suivant l'approche pragmatique que nous avons adoptée et mise en œuvre.

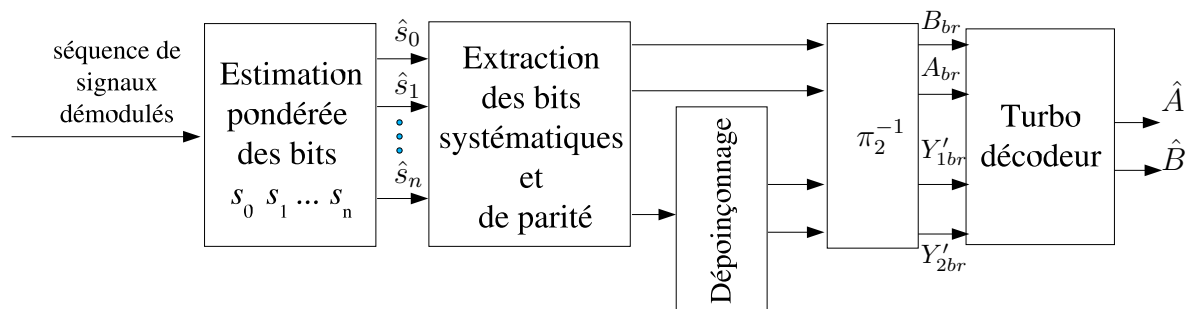


Fig. 3.9 — Schéma général de principe du récepteur dans le cas d'une modulation turbo-codée pragmatique

Estimation pondérée des bits s_0, s_1, \dots, s_n

Les estimations pondérées des bits s_0, s_1, \dots, s_n sont calculées comme les Logarithmes de Rapports de Vraisemblances (LRV) normalisés (cf. équation 2.10). Nous détaillerons pour chacune des modulations considérées ces calculs, qui sont basés sur les distances euclidiennes du point reçu par rapport aux points de la constellation.

Les estimations des bits sont ensuite quantifiées de la façon suivante : si l'on utilise b_q bits de quantification, chaque estimation \hat{s}_i est multipliée par un facteur $V_{max} \times E$ où :

- $V_{max} = 2^{b_q} - 1$,
- E est le facteur d'échelle dont la valeur dépend du rendement du code et du canal de transmission. En pratique, cette valeur est déterminée empiriquement par simulation pour chacun des cas considérés.

Après avoir été quantifiées, les estimations sont écrêtées à $\pm V_{max}$. Dans le cadre de notre étude, les simulations effectuées considèrent b_q égal à 5, voire 6, afin que la quantification n'entraîne pas de dégradation significative des performances.

Extraction des bits systématiques et de parité, permutation π_2^{-1} , dépoinçonnage

Ce sont les opérations inverses de celles des fonctions correspondantes de l'émetteur (mise en forme des symboles, entrelacement et poinçonnage).

Turbo-décodeur

Le même turbo-décodeur est utilisé pour les différentes associations, sa description ayant été donnée en Section 3.1.2.

3.2.3 Quelques précisions sur les conditions de simulation

Lors de simulations nous considérons qu'un résultat fiable de taux d'erreurs trames (resp. binaires) est obtenu lorsque le nombre de blocs (resp. bits) erronés simulés est supérieur à 20 (resp. 100). Ainsi, par exemple, pour valider une valeur de $FER=10^{-5}$, deux millions de blocs doivent être simulés.

Notons également que, tenant compte du schéma considéré (entrées du décodeur quantifiées, algorithme de décodage Max-Log-MAP), l'ensemble de résultats que nous avons obtenu dans cet étude est compatible avec des systèmes réels. Autrement dit, les valeurs de performances simulées que nous allons présenter correspondent à celles d'une implémentation sur

circuit. De plus, aucune estimation sur l'état du canal n'est nécessaire car le décodeur ne requiert pas la connaissance de la variance de bruit.

3.3 Association pragmatique d'un TCC duo-binaire et d'une MDP-8

Nous avons tout d'abord considéré l'association d'un turbocode convolutif duo-binaire et d'une modulation à déplacement de phase à 8 états pour obtenir un système de transmission présentant une efficacité spectrale de 2 bit/s/Hz. Le paramètre n est donc égal à 3 et le rendement de codage vaut $R = 2/3$. La figure C.1 montre le *mapping* de Gray sur la constellation MDP-8.

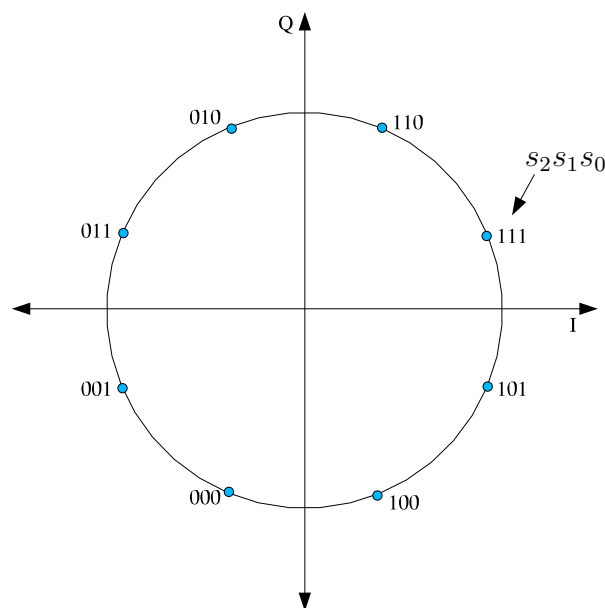


Fig. 3.10 — Constellation et *mapping* de la modulation MDP-8

3.3.1 Construction des symboles de modulation

L'Annexe C de ce rapport présente une analyse de la probabilité d'erreur de chaque position binaire dans la modulation MDP-8 considérée. D'après cette analyse on peut affirmer que la probabilité d'erreur moyenne du bit s_0 est supérieure à la probabilité d'erreur des bits s_2 et s_1 . Autrement dit, le dernier bit de chaque triplet issu du bloc "Codage binaire à signal" est en moyenne moins bien protégé que les deux premiers. Pour le rendement $2/3$, les bits de redondance Y_1 et Y_2 ont été systématiquement placés à la position s_0 , et sont donc moins bien protégés que les données A et B . Il s'agit alors d'un schéma de construction de symboles de type A, comme celui qui était préconisé dans la thèse de Le Goff [80].

Une stratégie particulière de construction des symboles, que nous avons dénommée **TMCT pragmatique** peut être appliquée pour ce cas particulier de rendement de code et d'ordre de modulation. Cette stratégie s'inspire du principe de la TMCT [66], affectant tous les bits issus d'un même instant de codage dans un même symbole de la modulation. Le tableau 3.4 donne la correspondance exacte bit à bit entre les bits issus du bloc "Mise en

forme des symbole” et les symboles de modulation issus du bloc “Codage binaire à signal”, pour la TMCT pragmatique avec schéma A.

| Numéro de symbole ($1 \leq i \leq N/2$) | TMCT pragmatique (schéma A) | |
|--|-----------------------------|-----------|
| | $2i - 1$ | $2i$ |
| s_2 | $A[2i - 1]$ | $A[2i]$ |
| s_1 | $B[2i - 1]$ | $B[2i]$ |
| s_0 | $Y_1[2i - 1]$ | $Y_2[2i]$ |

Tab. 3.4 — Correspondance exacte bit à bit, $R = 2/3$, MDP-8

3.3.2 Calcul de l’estimation pondérée des bits s_2 , s_1 et s_0

En réception, les estimations pondérées des bits s_2 , s_1 et s_0 sont calculées en appliquant l’approximation Max-Log (cf. équation 2.3) à l’expression du LRV correspondant à chaque bit (cf. équation 2.10). Les expression ainsi obtenues sont :

$$\begin{aligned} \hat{s}_2 &= \frac{1}{4} \left[\min(d^2(000), d^2(001), d^2(010), d^2(011)) - \min(d^2(100), d^2(101), d^2(110), d^2(111)) \right] \\ \hat{s}_1 &= \frac{1}{4} \left[\min(d^2(000), d^2(001), d^2(100), d^2(101)) - \min(d^2(010), d^2(011), d^2(110), d^2(111)) \right] \\ \hat{s}_0 &= \frac{1}{4} \left[\min(d^2(000), d^2(010), d^2(100), d^2(110)) - \min(d^2(001), d^2(011), d^2(101), d^2(111)) \right] \end{aligned}$$

Chacune des distances $d^2(s_2s_1s_0)$ représente la distance euclidienne au carré du point reçu par rapport à chaque point de la constellation. Les expression simplifiées ² de ces distances sont :

$$\begin{aligned} \delta(000) &= I \sin \frac{\pi}{8} + Q \cos \frac{\pi}{8} \\ \delta(001) &= I \cos \frac{\pi}{8} + Q \sin \frac{\pi}{8} \\ \delta(010) &= I \sin \frac{\pi}{8} - Q \cos \frac{\pi}{8} \\ \delta(011) &= I \cos \frac{\pi}{8} - Q \sin \frac{\pi}{8} \\ \delta(100) &= -I \sin \frac{\pi}{8} + Q \cos \frac{\pi}{8} \\ \delta(101) &= -I \cos \frac{\pi}{8} + Q \sin \frac{\pi}{8} \\ \delta(110) &= -I \sin \frac{\pi}{8} - Q \cos \frac{\pi}{8} \\ \delta(111) &= -I \cos \frac{\pi}{8} - Q \sin \frac{\pi}{8} \end{aligned}$$

3.3.3 Comparaison des performances des codes à 8 et 16 états

La figure 3.11 donne une comparaison de performances entre les codeurs à 8 et à 16 états associés à la MDP-8, pour une taille de bloc de 188 octets. Les résultats de simulation montrent que, pour un FER= 10^{-7} , le gain de codage apporté par le codeur à 16 états par rapport au codeur à 8 états est de quasiment 1 dB. La valeur de cet écart est la même que celle qui avait été observée dans le cas de la MDP-2 (cf. Section 3.1.3).

La figure 3.12 permet de visualiser les performances du turbocodeur à 16 états associé à la MDP-8 selon la stratégie TMCT pragmatique avec schéma A, pour deux différentes tailles de bloc. Le tableau 3.5 montre l’écart entre les performances de notre système et les limites théoriques pour 54 et 188 octets. Nous avons comparé ces résultats avec ceux obtenus par Le Goff dans son travail de thèse, pour une même longueur de contrainte $K = 5$ et un même rendement $R = 2/3$. Les performances présentées par Le Goff requièrent un rapport signal à

²La simplification appliquée consiste à enlever les termes qui sont communs à toutes les distances.

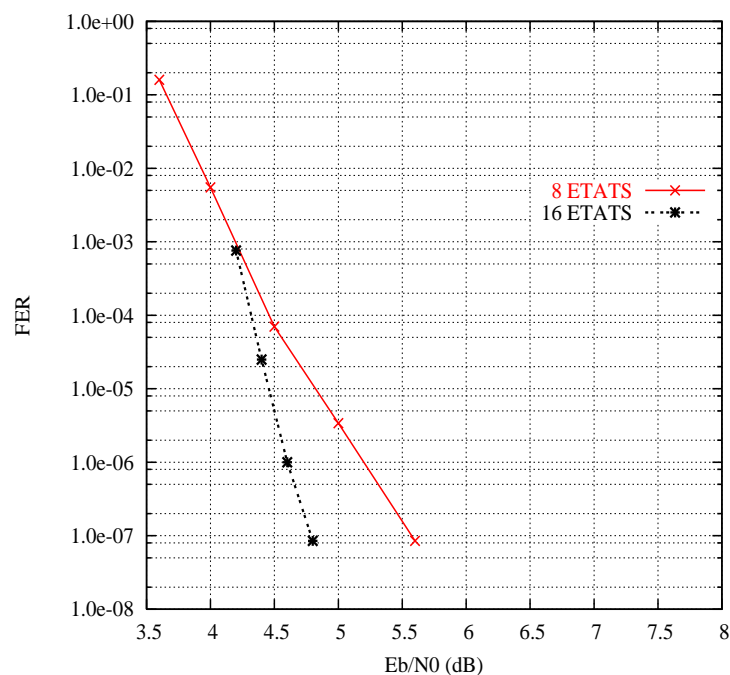


Fig. 3.11 — Comparaison de performances, sur canal gaussien, des turbocodeurs à 8 et à 16 états associés à la MDP-8 selon la stratégie de TMCT Pragmatique avec schéma A. Blocs de 188 octets. Rendement de codage $R = 2/3$. Décodage Max-Log-MAP, entrées du décodeur quantifiées sur 6 bits, 8 itérations de décodage

bruit égal à 4,45 dB pour atteindre un BER = 10^{-4} avec une taille de bloc de 65536 bits. Notre système atteint ce taux d'erreurs avec 4,15 dB de rapport signal à bruit avec des blocs de 1504 bits (*i.e.* 188 octets, cf. figure 3.12). Compte tenu des différentes tailles de bloc, on peut affirmer que nos résultats sont meilleurs que ceux de Le Goff, d'environ 1 dB pour ce taux d'erreurs. Cette amélioration des performances est due à l'utilisation de codes duo-binaires, à l'adoption de l'algorithme Max-Log-MAP³ et à la façon dont les LRVs des bits sont calculés en réception. Le Goff obtient les échantillons des bits à partir des symboles bruités issus du canal faisant appel à une *fonction d'adaptation* simplifiée, tel qu'il est détaillé dans sa thèse [80]. Notre calcul des LRVs (cf. Section 3.3.2) s'avère visiblement plus performant.

| FER | 10^{-4} | 10^{-7} |
|-------------------------|-----------|-----------|
| Taille de bloc (octets) | | |
| 54 | 0,65 | 1,1 |
| 188 | 0,6 | 0,75 |

Tab. 3.5 — Valeur en dB de l'écart entre performances effectives (décodage Max-Log-MAP, entrées du décodeur quantifiées sur 6 bits, 8 itérations de décodage) et limites théoriques du turbocodeur à 16 états associé à la modulation MDP-8

³au lieu du SOVA, algorithme utilisé par Le Goff

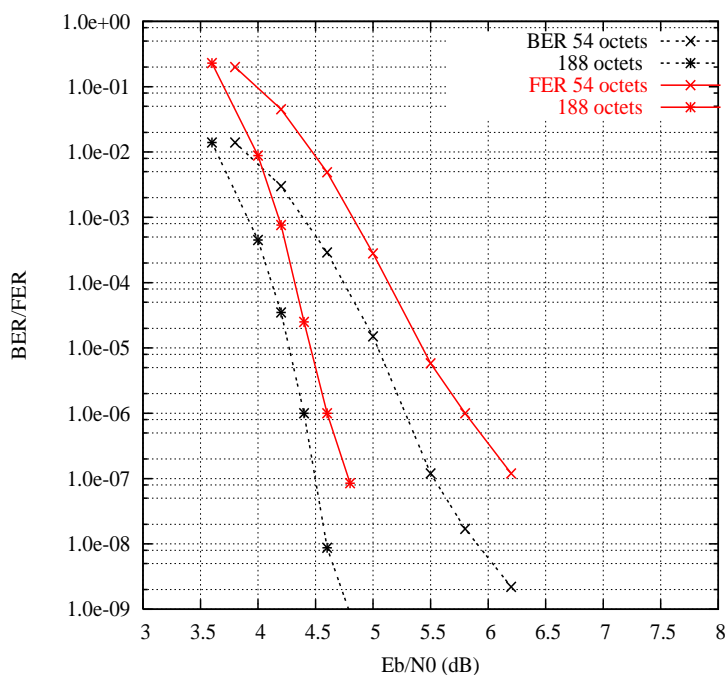


Fig. 3.12 — Comparaison de performances, sur canal gaussien, pour trois différentes tailles de bloc (54 et 188 octets) de l'association pragmatique du turbocodeur à 16 états et la modulation MDP-8. Rendement de codage 2/3. Décodage Max-Log-MAP, entrées du décodeur quantifiées sur 6 bits, 8 itérations de décodage

3.4 Association pragmatique d'un TCC duo-binaire et d'une MAQ-16

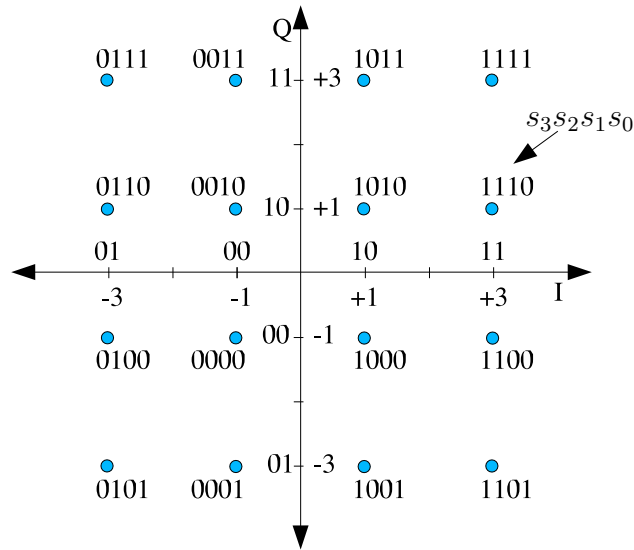
Nous avons ensuite considéré l'étude de la modulation MAQ-16. Pour cette modulation, nous avons en particulier analysé l'impact de la construction des symboles à partir des bits codés sur les performances. Les schémas d'émission et réception correspondent aux schémas des figures 3.2.1 et 3.9 avec $n = 4$. Les rendements de codage étudiés sont $R = 1/2$ et $R = 3/4$.

3.4.1 Codage binaire à signal de la modulation

La constellation et le *mapping* de la MAQ-16 considérée sont montrés dans la figure 3.13. Ce *mapping* correspond à un codage classique des points de la constellation : les voies en phase (I) et en quadrature (Q) sont codées indépendamment et de façon analogue suivant un code de Gray. Cette indépendance des voies I et Q simplifie significativement les calculs lors de la réception.

3.4.2 Stratégies de construction des symboles

Nous présentons dans l'Annexe C l'analyse de la probabilité d'erreur de chaque position binaire dans les symboles pour la modulation MAQ-16 considérée. D'après cette analyse, on peut établir que la probabilité d'erreur moyenne des bits s_2 et s_0 est supérieure à la probabilité d'erreur moyenne des bits s_3 et s_1 . Puisque le codage binaire à signal établit une correspondance entre les bits issus du codage (d'information et de parité) et les bits s_3, s_2, s_1

Fig. 3.13 — Constellation et *mapping* de Gray de la modulation MAQ-16

et s_0 , il est possible de procurer un degré différent de protection aux bits systématiques et aux bits de parité. Nous avons étudié pour les deux rendements considérés l'influence du niveau de protection des bits codés sur les performances en simulant les deux stratégies basiques de construction de symboles : schéma A et schéma Y.

Le principe de construction des symboles dépend du rendement de code considéré, tel qu'il est décrit ci-après :

$R=1/2$: Pour un bloc de N couples de données, le nombre de symboles émis sur le canal est égal à N . La correspondance exacte bit à bit pour le schéma A est donnée sur le tableau 3.6. Pour obtenir le schéma Y on échange les contenus de s_3 et s_2 et les contenus de s_1 et s_0 dans tous les symboles.

$R=3/4$: Pour un bloc de N couples de données, le nombre de symboles émis sur le canal est égal à $2N/3$. Le tableau 3.7 présente la correspondance exacte bit à bit pour le schéma A. Le schéma Y est obtenu en échangeant les contenus de s_3 et s_2 dans les symboles $4i - 2$ et $4i - 1$, et les contenus de s_1 et s_0 dans les symboles $4i - 3$ et $4i - 1$.

| Numéro de symbole ($1 \leq i \leq N$) | i |
|--|----------|
| s_3 | $A[i]$ |
| s_2 | $Y_1[i]$ |
| s_1 | $B[i]$ |
| s_0 | $Y_2[i]$ |

Tab. 3.6 — Correspondance exacte bit à bit pour le schéma A, $R = 1/2$

3.4.3 Calcul de l'estimation pondérée des bits s_3 , s_2 , s_1 et s_0

En réception, les estimations pondérées des bits s_3 , s_2 , s_1 et s_0 sont obtenues en calculant indépendamment sur chaque voie I et Q les distances euclidiennes au carré du point reçu par rapport aux 4 points de la voie considérée $d_I^2(00)$, $d_I^2(01)$, $d_I^2(10)$, $d_I^2(11)$, $d_Q^2(00)$, $d_Q^2(01)$,

| Numéro de symbole ($1 \leq i \leq 2N/3$) | $4i - 3$ | $4i - 2$ | $4i - 1$ | $4i$ |
|---|---------------|---------------|---------------|-------------|
| s_3 | $A[6i - 5]$ | $B[6i - 4]$ | $A[6i - 2]$ | $A[6i - 1]$ |
| s_2 | $B[6i - 5]$ | $Y_2[6i - 5]$ | $Y_1[6i - 3]$ | $B[6i - 1]$ |
| s_1 | $A[6i - 4]$ | $A[6i - 3]$ | $B[6i - 2]$ | $A[6i]$ |
| s_0 | $Y_1[6i - 5]$ | $B[6i - 3]$ | $Y_2[6i - 3]$ | $B[6i]$ |

Tab. 3.7 — Correspondance exacte bit à bit pour le schéma A, $R = 3/4$

$d_Q^2(10)$, $d_Q^2(11)$. Les expressions simplifiées de ces distances sont ⁴ :

$$\begin{aligned}
 \delta_I(00) &= I \\
 \delta_I(01) &= 4 + 3I \\
 \delta_I(10) &= -I \\
 \delta_I(11) &= 4 - 3I \\
 \delta_Q(00) &= Q \\
 \delta_Q(01) &= 4 + 3Q \\
 \delta_Q(10) &= -Q \\
 \delta_Q(11) &= 4 - 3Q
 \end{aligned}$$

où (I, Q) sont les coordonnées du point reçu dans le plan de la constellation.

Ensuite, en appliquant la simplification Max-Log, on peut montrer que les estimations sont alors calculées comme :

$$\begin{aligned}
 \hat{s}_3 &= \frac{1}{4} \left[\min(d_I^2(00), d_I^2(01)) - \min(d_I^2(10), d_I^2(11)) \right] \\
 \hat{s}_2 &= \frac{1}{4} \left[\min(d_I^2(00), d_I^2(10)) - \min(d_I^2(01), d_I^2(11)) \right] \\
 \hat{s}_1 &= \frac{1}{4} \left[\min(d_Q^2(00), d_Q^2(01)) - \min(d_Q^2(10), d_Q^2(11)) \right] \\
 \hat{s}_0 &= \frac{1}{4} \left[\min(d_Q^2(00), d_Q^2(10)) - \min(d_Q^2(01), d_Q^2(11)) \right]
 \end{aligned}$$

3.4.4 Résultats de simulation

Nous considérons pour cette association les deux critères d'évaluation des performances introduits en Section 1.2.1, à savoir : convergence et performance asymptotique. Dans nos simulations, nous supposons la transmission de blocs de 54 octets, les entrées du décodeur sont quantifiées sur 6 bits et on applique 8 itérations de décodage. L'étude a été menée pour les deux types de canaux décrits en Section 1.5 : le canal BBAG (ou canal *gaussien*) et le canal à évanouissements lents de type Rayleigh non sélectif en fréquence (nommé simplement canal *de Rayleigh* pour la suite).

Canal gaussien

Nous considérons dans un premier temps l'association comportant le turbocode à 8 états avec un schéma de construction de symboles de type A pour les deux rendements de codage considérés : $R = 1/2$ et $R = 3/4$ (cf. figure 3.14). Cette association est *a priori* la mieux adaptée pour obtenir une bonne convergence car elle comporte, d'entre les deux codeurs considérés dans l'étude, celui à plus petit nombre d'états et, d'entre les deux schémas de construction de symboles, celui qui protège au mieux les bits systématiques. Nous avons

⁴La simplification appliquée consiste à enlever les termes qui sont communs à toutes les distances.

indiqué sur la figure les écarts entre les courbes de simulations et les limites théoriques (LT) [2] pour deux valeurs représentatives de FER, 10^{-2} et 10^{-4} . Du point de vue de la convergence, les résultats obtenus sont très satisfaisants car on se situe à moins de 1 dB de la limite théorique pour les deux rendements considérés.

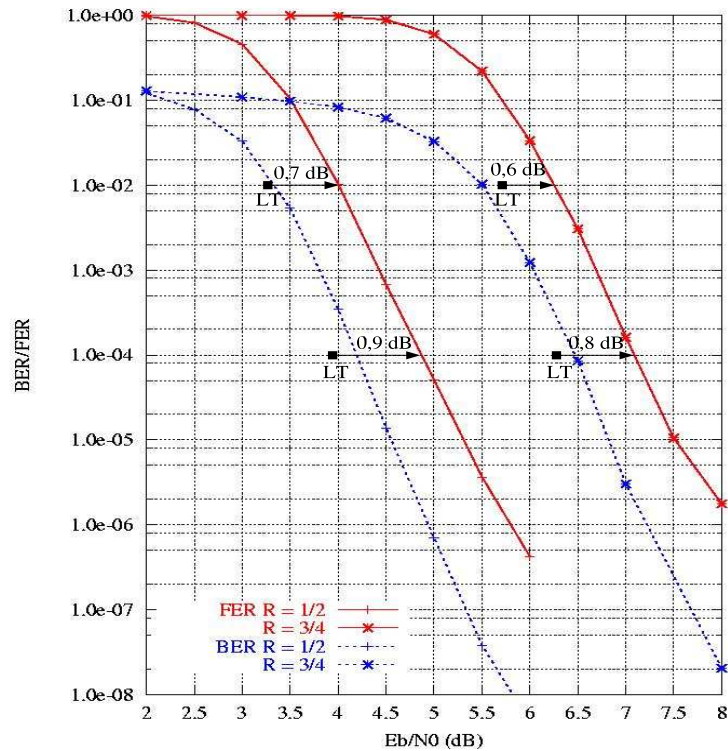


Fig. 3.14 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-16 et du code duo-binaire 8 états du standard DVB-RCT, pour des blocs de 54 octets. Schéma A. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage.

Nous avons ensuite simulé l'association contenant le turbocode 16 états pour les deux types de schémas de construction de symboles (A et Y). La figure 3.15 montre l'ensemble des résultats de simulation. Pour le rendement $R = 1/2$, le schéma A présente de meilleures performances que le schéma Y pour l'ensemble des valeurs E_b/N_0 simulées : à un FER de 10^{-6} , on observe un décalage de plus de 0,3 dB entre les deux courbes. Cet écart a tendance à se réduire lorsque E_b/N_0 croît, les courbes étant probablement amenées à se croiser à un FER voisin de 10^{-8} . Pour le rendement $R = 3/4$, l'écart de performances entre les schémas A et Y est moins important. En effet, à fort rendement du code, le nombre de bits de redondance est faible et les deux schémas A et Y conduisent à des répartitions des bits codés proches. On constate que le schéma Y devient plus performant que le schéma A en dessous d'un FER de 10^{-5} .

Ces conclusions diffèrent en partie de celles établies par Le Goff dans sa thèse à propos de l'influence sur les performances du schéma de construction de symboles. A faible taux d'erreurs, nous avons également constaté qu'un schéma de type A, favorisant la protection des bits systématiques, constitue la solution la plus performante car elle garantit une meilleure convergence du processus itératif. En revanche, à fort rapport signal à bruit, nos conclusions ne coïncident pas avec celles de Le Goff. Il avait en effet conclu, au vu de ses résultats de simulation, qu'à faible niveau de bruit, les performances tendaient à devenir indépendantes

du schéma de construction des symboles. Autrement dit, les courbes de performances correspondant aux différents schémas de construction tendaient à se rejoindre pour les faibles taux d'erreurs. En 1994-95, Le Goff disposait de moyens de calcul ne lui permettant pas de simuler des taux d'erreurs de trames inférieurs à 10^{-5} . La puissance de calcul dont nous disposons actuellement nous permettant d'atteindre des taux d'erreurs de trames de 10^{-7} , nous avons pu observer qu'en fait la stratégie de construction de symboles joue également un rôle au niveau du comportement asymptotique de la modulation turbocodée et que le schéma favorisant la protection des bits de redondance permet d'obtenir le gain asymptotique le plus important.

L'interprétation de ce résultat fait appel à l'analyse des chemins erronés dans le treillis à fort rapport signal à bruit. Nous avons observé que, dans la majorité des cas, les séquences erronées contiennent un nombre plutôt élevé de bits de redondance erronés et un nombre de bits systématiques erronés faible. Autrement dit, les séquences erronées présentent en général un poids d'entrée faible. En particulier, les chemins erronés en question correspondent pour la plupart à des motifs d'erreur rectangulaires [18]. Il en résulte que, du point de vue du comportement asymptotique de la modulation turbocodée, le schéma Y procure les meilleures performances, car il assure une meilleure protection des bits de parité.

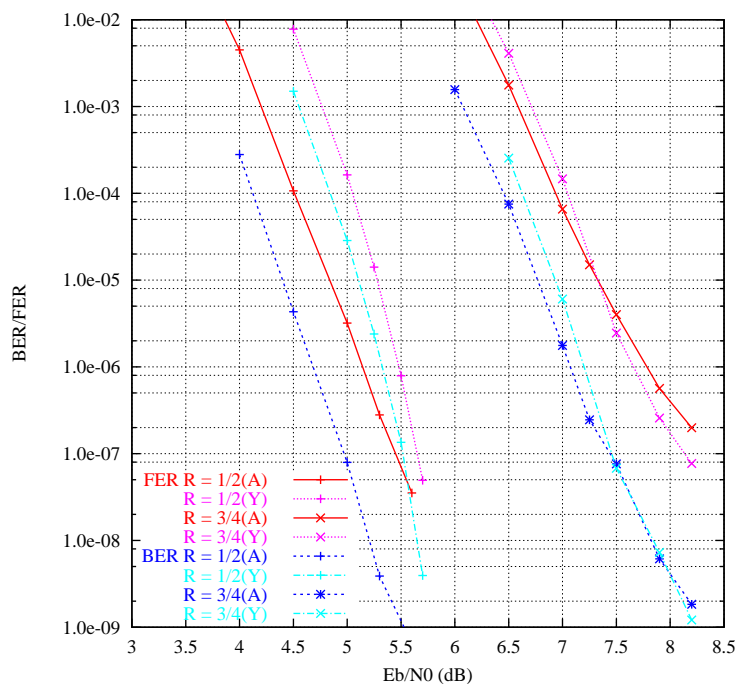


Fig. 3.15 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-16 et du code duo-binaire à 16 états, pour des blocs de 54 octets. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

La figure 3.16 montre une comparaison des performances des codes à 8 et 16 états. Les polynômes générateurs du code 16 états ayant été choisis pour conférer au code le meilleur compromis seuil de convergence/distance minimale, on constate que le code conserve ces propriétés lors de son association pragmatique avec la modulation MAQ-16. Dans le cas d'un schéma A, l'utilisation d'un code à 16 états conduit même à de meilleures performances que le codeur à 8 états à un FER de 10^{-2} pour les deux rendements considérés. Pour les taux d'erreurs inférieurs ou égaux à 10^{-6} , le gain apporté par le passage à un code à 16 états

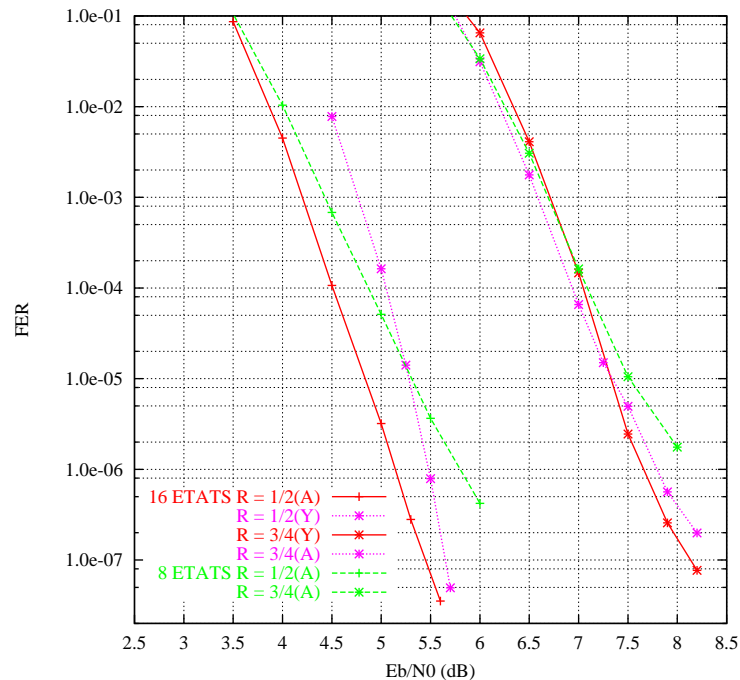


Fig. 3.16 — Comparaison de performances, sur canal gaussien, de l'association pragmatique d'une MAQ-16 avec un code duo-binaire à 8 états et un code duo-binaire à 16 états, pour des blocs de 54 octets. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

associé à un schéma de construction de symboles adéquat est supérieur à 0,5 dB pour les deux valeurs de rendements considérés.

D'après cet ensemble de résultats, on peut affirmer que l'utilisation d'un code 16 états conduit à un schéma globalement plus performant, qui pourrait être retenu aussi bien pour les forts que pour les faibles taux d'erreurs. Néanmoins, pour les taux d'erreurs élevés, le choix d'un code élémentaire à 8 états est une solution performante à faible coût car la complexité du décodeur est divisée par deux par rapport à une solution basée sur un code à 16 états.

Canal de Rayleigh

Nous considérons dans un premier temps l'association de la modulation MAQ-16 avec le turbocode à 8 états et un schéma de type A pour la construction de symboles (cf. figure 3.17). A un BER=10⁻⁴, la courbe de simulation se situe à 2,8 dB (resp. 3,6 dB) de la valeur de la capacité sur canal de Rayleigh pour une rendement de codage $R = 1/2$ (resp. $R = 3/4$). Sur canal gaussien l'écart entre performance obtenue et capacité est de 2 dB (resp. 1,7 dB) pour les mêmes rendements à un FER=10⁻². Nous prenons comme référence, dans cette partie de l'étude, la capacité du canal et non les limites théoriques tenant compte de la taille de bloc et du taux d'erreur visé car, sur canal de Rayleigh, ces limites théoriques n'ont pas encore été calculées. On constate, d'après les valeurs des écarts obtenues, que la modulation turbocodée étudiée n'approche pas la capacité du canal d'aussi près sur canal de Rayleigh que sur canal gaussien. Par conséquent, le schéma d'association pragmatique turbocode/modulation doit probablement être reconsidéré dans le cas des canaux à évanouissements. Cette étude n'a pas été menée dans le cadre de cette thèse.

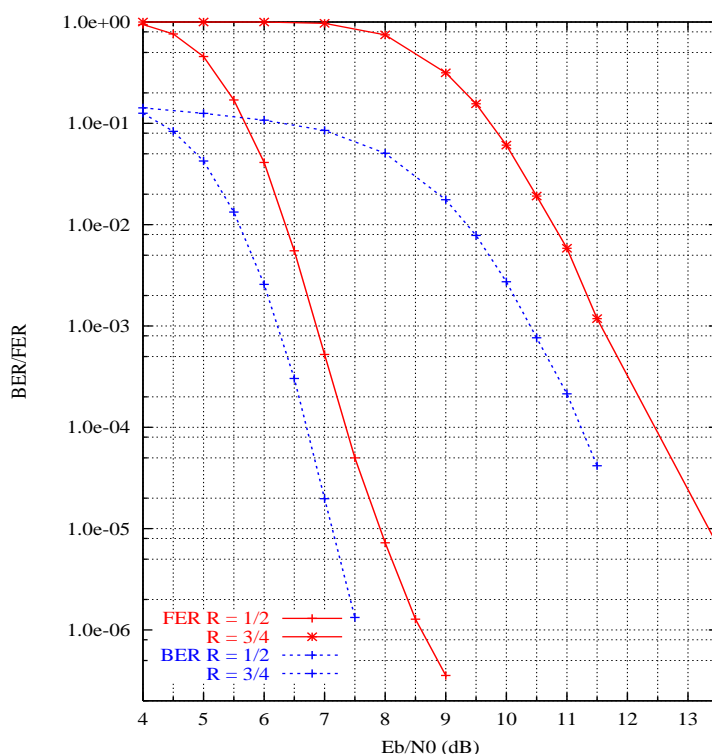


Fig. 3.17 — Performance sur canal de Rayleigh de l'association pragmatique d'une MAQ-16 et du code duo-binaire à 8 états, pour des blocs de 54 octets. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

La figure 3.18 présente les résultats de simulation du code à 16 états pour les deux schémas de construction de symboles définis précédemment. A faible taux d'erreurs, l'écart entre les courbes correspondant aux schémas A et Y est du même ordre de grandeur que l'écart observé sur le canal gaussien (0,7 dB pour $R = 1/2$ et 0,2 pour $R = 3/4$ à un FER de 10^{-2}). En revanche, on constate que cet écart entre les courbes semble se maintenir aux faibles taux d'erreurs. La différence de comportement asymptotique entre les schémas A et Y est beaucoup moins évident que pour le cas gaussien.

En ce qui concerne les courbes correspondant au rendement 3/4, les performances en terme de FER des schémas A et Y deviennent équivalentes en dessous de $2 \cdot 10^{-5}$, mais le schéma A reste le plus performant en terme de BER, avec un gain d'environ 0,3 dB quelle que soit la valeur de E_s/N_0 considérée. Finalement, pour le canal de Rayleigh, la stratégie de construction visant à mieux protéger les bits systématiques est la plus performante pour des taux d'erreurs de trames supérieurs ou égaux à 10^{-7} . Nous ne disposons pas de moyens de simulation permettant de confirmer cette tendance pour les taux d'erreurs inférieurs.

Les performances comparées de la figure 3.19 montrent que l'adoption d'un code élémentaire à 16 états n'entraîne pas non plus de dégradation significative du seuil de convergence du schéma de codage/modulation sur canal de Rayleigh. Pour les taux d'erreurs inférieurs ou égaux à 10^{-6} , le gain apporté par le passage à un code 16 états associé à un schéma A est supérieur ou égal à 0,5 dB. L'utilisation d'un code à 16 états conduit, comme dans le cas du canal gaussien, à un schéma globalement plus performant, qui peut être retenu aussi bien pour les forts que pour les faibles taux d'erreurs, au prix d'une complexité matérielle double.

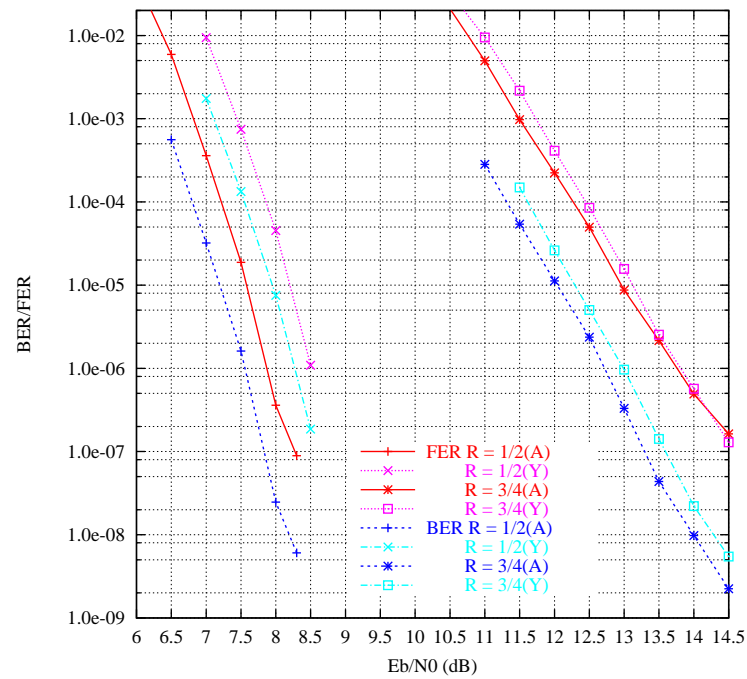


Fig. 3.18 — Performance sur canal de Rayleigh de l'association pragmatique d'une MAQ-16 et du code duo-binaire à 16 états, pour des blocs de 54 octets. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

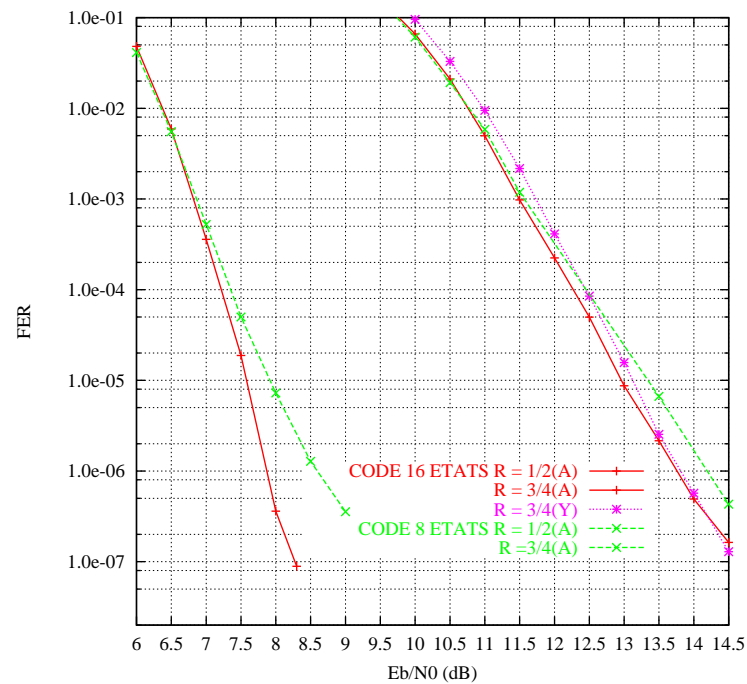


Fig. 3.19 — Comparaison de performances, sur canal de Rayleigh, de l'association pragmatique d'une MAQ-16 avec un code duo-binaire à 8 états/16 états pour des blocs de 54 octets. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

Conclusion

Nous avons considéré l'étude des performances de l'association pragmatique d'un TCC duo-binaire et d'une modulation MAQ-16. L'ensemble des résultats de simulation obtenus aussi bien sur canal gaussien que sur canal de Rayleigh nous a permis de constater les différences entre le comportement de la modulation turbocodée suivant le type de canal considéré. Nous avons ainsi décidé d'étendre cette étude à une modulation d'ordre supérieur pour vérifier si les conclusions obtenues peuvent être généralisées.

3.5 Etude de la construction de symboles dans l'association pragmatique d'un TCC et d'une modulation MAQ-64

Cette section considère le cas de la modulation MAQ-64 turbocodée pragmatique, en portant un intérêt particulier à la problématique de la construction des symboles. Les deux rendements de codage considérés dans cette étude sont $R = 2/3$ et $R = 3/5$. Les schémas d'émission et de réception correspondent à ceux décrits en figures 3.2.1 et 3.9 avec $n = 6$.

3.5.1 Codage binaire à signal de la modulation

La constellation et le *mapping* de la modulation MAQ-64 considérée sont donnés sur la figure 3.20. Cette modulation MAQ-64 est construite sur la base d'une modulation en amplitude à 8 états sur deux porteuses en quadrature. Les voies I et Q sont codées indépendamment et de façon analogue. Sur chaque voie, le codage suit un code Gray.

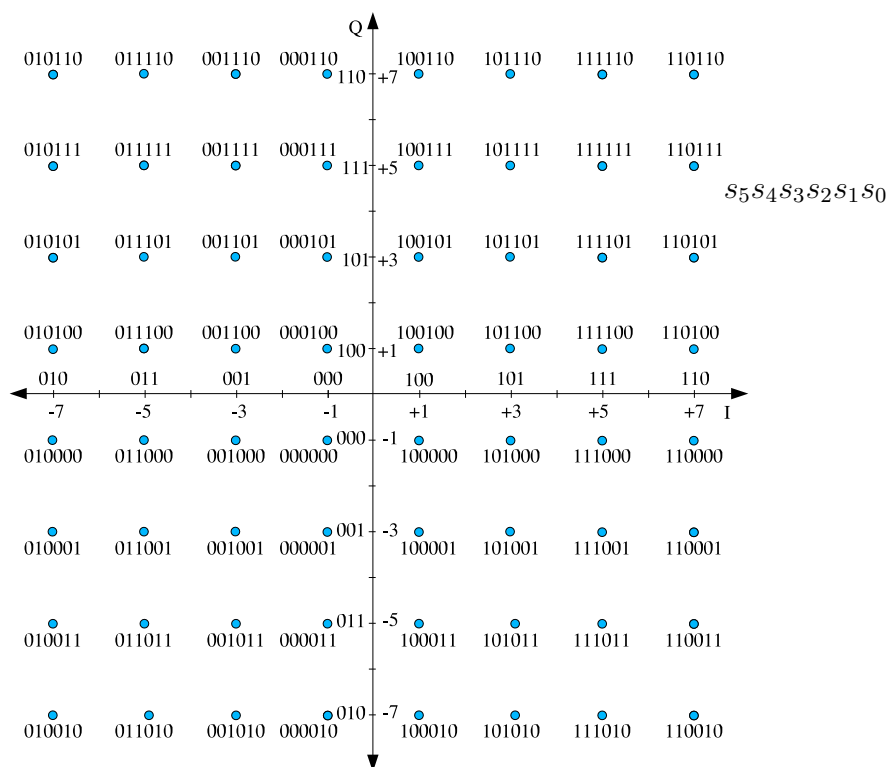


Fig. 3.20 — Constellation et *mapping* de la modulation MAQ-64

3.5.2 Stratégies de construction de symboles

D'après l'analyse présentée dans l'Annexe C, pour cette modulation, il y a trois niveaux différents de protection des bits. Les différentes positions de bits dans les symboles peuvent être classifiées comme suit :

- s_5, s_2 : bits les mieux protégés
- s_4, s_1 : bits moyennement protégés
- s_3, s_0 : bits les moins protégés

Ce classement va nous permettre de définir les différents schémas de construction de symboles que nous allons simuler :

Schéma A : on affecte en priorité les places s_5 et s_2 aux bits systématiques. Cette stratégie vise à favoriser la bonne convergence de la modulation turbocodée.

Schéma Y : on affecte en priorité les places s_5 et s_2 aux bits de redondance. Cette stratégie vise à favoriser le bon comportement asymptotique de la modulation turbocodée.

Schéma Ymix : on affecte en priorité les places s_4 et s_1 aux bits de redondance. Cette stratégie vise à obtenir un comportement intermédiaire entre les schémas A et Y.

Comme dans le cas de la MDP-8 avec rendement de codage $R = 2/3$, il est d'autre part possible d'appliquer le principe de la Turbo-Modulation Codée en Treillis (TMCT) pragmatique, inspiré du principe des TMCT [66] et caractérisé par le fait que tous les bits issus du codeur à un instant donné sont codés dans un même symbole de modulation.

Pour le rendement de codage $R = 2/3$, si N est le nombre de couples de données dans un bloc, le nombre de symboles émis sur le canal est égal à $N/2$. Le tableau 3.8 donne la correspondance exacte bit à bit pour chacun des quatre schémas possibles. Dans le cas de la TMCT pragmatique, \tilde{A} et \tilde{B} représentent les bits A et B dans l'ordre entrelacé. Seuls les bits systématiques dans les places paires du bloc sont transmis dans l'ordre naturel. Les bits qui sont à l'origine dans les places impaires sont transmis dans l'ordre entrelacé. Cette stratégie est possible grâce au fait que l'entrelacement considéré (cf. Section 3.1.1) transforme les positions paires en impaires et *vice versa*. En effet pour que tous les bits systématiques soient transmis il est impératif que l'entrelacement respecte cette caractéristique.

Pour le rendement de codage $R=3/5$, si N est le nombre de couples de données dans un bloc, le nombre de symboles émis sur le canal est égal à $5N/9$. La correspondance exacte bit à bit pour les schémas A, Y et Ymix simulés est donnée dans les tableaux 3.9, 3.10 et 3.11.

| Numéro de symbole ($1 \leq i \leq N/2$) | Schéma A i | Schéma Y i | Schéma Ymix i | TMCT pragmatique i |
|--|-----------------|-----------------|--------------------|-------------------------|
| s_5 | $A[2i - 1]$ | $Y_1[2i - 1]$ | $B[2i - 1]$ | $A[2i]$ |
| s_4 | $B[2i - 1]$ | $B[2i - 1]$ | $Y_1[2i - 1]$ | $B[2i]$ |
| s_3 | $Y_1[2i - 1]$ | $A[2i - 1]$ | $A[2i - 1]$ | $Y_1[2i]$ |
| s_2 | $A[2i]$ | $Y_2[2i - 1]$ | $A[2i]$ | $\tilde{A}[2i]$ |
| s_1 | $B[2i]$ | $A[2i]$ | $Y_2[2i - 1]$ | $\tilde{B}[2i]$ |
| s_0 | $Y_2[2i - 1]$ | $B[2i]$ | $B[2i]$ | $Y_2[2i]$ |

Tab. 3.8 — Correspondance exacte bit à bit pour la construction de symboles dans la modulation MAQ-64 turbocodée, $R = 2/3$

| Numéro de symbole ($1 \leq i \leq 5N/9$) | $5i - 4$ | $5i - 3$ | $5i - 2$ | $5i - 1$ | $5i$ |
|---|---------------|---------------|---------------|---------------|---------------|
| s_5 | $A[9i - 8]$ | $A[9i - 6]$ | $A[9i - 5]$ | $A[9i - 3]$ | $A[9i - 1]$ |
| s_4 | $B[9i - 8]$ | $Y_1[9i - 7]$ | $B[9i - 5]$ | $B[9i - 3]$ | $B[9i - 1]$ |
| s_3 | $Y_1[9i - 8]$ | $Y_2[9i - 7]$ | $Y_2[9i - 4]$ | $Y_2[9i - 2]$ | $Y_1[9i - 1]$ |
| s_2 | $B[9i - 7]$ | $B[9i - 6]$ | $[9i - 4]$ | $B[9i - 2]$ | $B[9i]$ |
| s_1 | $A[9i - 7]$ | $Y_2[9i - 5]$ | $A[9i - 4]$ | $A[9i - 2]$ | $A[9i]$ |
| s_0 | $Y_2[9i - 8]$ | $Y_1[9i - 5]$ | $Y_1[9i - 4]$ | $Y_1[9i - 2]$ | $Y_2[9i - 1]$ |

Tab. 3.9 — Correspondance exacte bit à bit pour la construction de symboles dans la modulation MAQ-64 turbocodée avec $R = 3/5$ et schéma A

| Numéro de symbole ($1 \leq i \leq 5N/9$) | $5i - 4$ | $5i - 3$ | $5i - 2$ | $5i - 1$ | $5i$ |
|---|---------------|---------------|---------------|---------------|---------------|
| s_5 | $Y_1[9i - 8]$ | $Y_2[9i - 7]$ | $Y_2[9i - 4]$ | $Y_2[9i - 2]$ | $Y_1[9i - 1]$ |
| s_4 | $B[9i - 8]$ | $Y_1[9i - 7]$ | $B[9i - 5]$ | $B[9i - 3]$ | $B[9i - 1]$ |
| s_3 | $A[9i - 8]$ | $A[9i - 6]$ | $A[9i - 5]$ | $A[9i - 3]$ | $A[9i - 1]$ |
| s_2 | $Y_2[9i - 8]$ | $Y_1[9i - 5]$ | $Y_1[9i - 4]$ | $Y_1[9i - 2]$ | $Y_2[9i - 1]$ |
| s_1 | $A[9i - 7]$ | $Y_2[9i - 5]$ | $A[9i - 4]$ | $A[9i - 2]$ | $A[9i]$ |
| s_0 | $B[9i - 7]$ | $B[9i - 6]$ | $B[9i - 4]$ | $B[9i - 2]$ | $B[9i]$ |

Tab. 3.10 — Correspondance exacte bit à bit pour la construction de symboles dans la modulation MAQ-64 turbocodée avec $R = 3/5$ et schéma Y

3.5.3 Calcul des estimations pondérées de bits s_5, s_4, s_3, s_2, s_1 et s_0

Les estimations pondérées des bits s_5, s_4, s_3, s_2, s_1 et s_0 sont obtenues en calculant indépendamment sur chaque voie I et Q les distances euclidiennes au carré du point reçu par rapport aux 8 points de la voie considérée $d_I^2(000), d_I^2(001), d_I^2(010), d_I^2(011), d_I^2(100), d_I^2(101), d_I^2(110), d_I^2(111), d_Q^2(000), d_Q^2(001), d_Q^2(010), d_Q^2(011), d_Q^2(100), d_Q^2(101), d_Q^2(110)$,

| Numéro de symbole ($1 \leq i \leq 5N/9$) | $5i - 4$ | $5i - 3$ | $5i - 2$ | $5i - 1$ | $5i$ |
|---|---------------|---------------|---------------|---------------|---------------|
| s_5 | $B[9i - 8]$ | $Y_1[9i - 7]$ | $B[9i - 5]$ | $B[9i - 3]$ | $B[9i - 1]$ |
| s_4 | $Y_1[9i - 8]$ | $Y_2[9i - 7]$ | $Y_2[9i - 4]$ | $Y_2[9i - 2]$ | $Y_1[9i - 1]$ |
| s_3 | $A[9i - 8]$ | $A[9i - 6]$ | $A[9i - 5]$ | $A[9i - 3]$ | $A[9i - 1]$ |
| s_2 | $A[9i - 7]$ | $Y_2[9i - 5]$ | $A[9i - 4]$ | $A[9i - 2]$ | $A[9i]$ |
| s_1 | $Y_2[9i - 8]$ | $Y_1[9i - 5]$ | $Y_1[9i - 4]$ | $Y_1[9i - 2]$ | $Y_2[9i - 1]$ |
| s_0 | $B[9i - 7]$ | $B[9i - 6]$ | $B[9i - 4]$ | $B[9i - 2]$ | $B[9i]$ |

Tab. 3.11 — Correspondance exacte bit à bit pour la construction de symboles dans la modulation MAQ-64 turbocodée avec $R = 3/5$ et schéma Ymix

$d_Q^2(111)$. Les expressions simplifiées de ces distances sont ⁵ :

$$\begin{aligned}
 \delta_I(000) &= I; & \delta_Q(000) &= Q \\
 \delta_I(001) &= 3I + 4; & \delta_Q(001) &= 3Q + 4 \\
 \delta_I(010) &= 7I + 24; & \delta_Q(010) &= 7Q + 24 \\
 \delta_I(011) &= 5I + 12; & \delta_Q(011) &= 5Q + 12 \\
 \delta_I(100) &= -I; & \delta_Q(100) &= -Q \\
 \delta_I(101) &= -3I + 4; & \delta_Q(101) &= -3Q + 4 \\
 \delta_I(110) &= -7I + 24; & \delta_Q(110) &= -7Q + 24 \\
 \delta_I(111) &= -5I + 12; & \delta_Q(111) &= -5Q + 12
 \end{aligned}$$

où (I, Q) sont les coordonnées du point reçu dans le plan de la constellation. En appliquant la simplification Max-Log, on peut montrer que les estimations sont alors calculées comme :

$$\begin{aligned}
 \hat{s}_5 &= \frac{1}{4} \left[\min(d_I^2(000), d_I^2(001), d_I^2(010), d_I^2(011)) - \min(d_I^2(100), d_I^2(101), d_I^2(110), d_I^2(111)) \right] \\
 \hat{s}_4 &= \frac{1}{4} \left[\min(d_I^2(000), d_I^2(001), d_I^2(100), d_I^2(101)) - \min(d_I^2(010), d_I^2(011), d_I^2(110), d_I^2(111)) \right] \\
 \hat{s}_3 &= \frac{1}{4} \left[\min(d_I^2(000), d_I^2(010), d_I^2(100), d_I^2(110)) - \min(d_I^2(001), d_I^2(011), d_I^2(101), d_I^2(111)) \right] \\
 \hat{s}_2 &= \frac{1}{4} \left[\min(d_Q^2(000), d_Q^2(001), d_Q^2(010), d_Q^2(011)) - \min(d_Q^2(100), d_Q^2(101), d_Q^2(110), d_Q^2(111)) \right] \\
 \hat{s}_1 &= \frac{1}{4} \left[\min(d_Q^2(000), d_Q^2(001), d_Q^2(100), d_Q^2(101)) - \min(d_Q^2(010), d_Q^2(011), d_Q^2(110), d_Q^2(111)) \right] \\
 \hat{s}_0 &= \frac{1}{4} \left[\min(d_Q^2(000), d_Q^2(010), d_Q^2(100), d_Q^2(110)) - \min(d_Q^2(001), d_Q^2(011), d_Q^2(101), d_Q^2(111)) \right]
 \end{aligned}$$

3.5.4 Résultats de simulation

Nous avons simulé la modulation MAQ-64 turbocodée pragmatique pour les deux rendements de codage considérés dans l'étude, $R = 2/3$ et $R = 3/5$, sur canal gaussien et sur canal de Rayleigh. Nous avons considéré les mêmes conditions de simulation que dans le cas de la MAQ-16 : transmission de blocs de 54 octets, 8 itérations de décodage et entrées du décodeur quantifiées sur 6 bits.

Sur canal gaussien, $R = 2/3$

La figure 3.21 (resp. figure 3.22) présente les résultats de simulation obtenus avec le codeur à 8 états (resp. 16 états), pour les quatre schémas possibles (cf. tableau 3.8). Du point de vue de la convergence, les schémas A et TMCT pragmatique s'avèrent les plus performants : à un FER=10⁻² leurs performances sont identiques et se situent à 0,6 dB de la limite théorique quel que soit le codeur utilisé ; à un FER=10⁻⁴ la TMCT pragmatique s'avère légèrement plus performante, la courbe se situant à 0,9 dB de la limite théorique.

D'un point de vue général, la TMCT pragmatique présente de meilleures performances que le schéma A classique. En effet, avec cette stratégie de construction de symboles, chaque erreur de transmission perturbant un symbole ne concerne qu'un seul instant de codage pour chaque code élémentaire, ce qui n'est pas le cas avec le schéma A classique où un symbole erroné affecte deux instants de codage.

Du point de vue de la performance asymptotique, on observe que les schémas de type Y et Ymix sont les plus intéressants car il ne présentent pas de changement de pente pour des valeurs de FER supérieures à 10⁻⁷. Nous comparons sur la figure 3.23 les courbes des schémas Y et Ymix pour les deux codeurs utilisés, ce qui nous permet de constater que pour

⁵La simplification appliquée consiste à enlever les termes qui sont communs à toutes les distances.

des applications à taux d'erreur cible très faible le codeur à 16 états avec un schéma Ymix est la solution la plus performante.

On constate d'après l'ensemble des résultats présentés que l'adoption d'un schéma de construction de symboles performant suivant le FER visé par l'application est plus déterminant que le choix d'un ou d'autre codeur. En effet, pour les $FER \leq 10^{-4}$ la meilleure performance est atteinte avec la même valeur de rapport signal à bruit pour les deux codeurs (cf. figures 3.21) et 3.22). A fort rapport signal à bruit, on observe également qu'un $FER = 10^{-7}$ est atteint avec la même valeur de E_b/N_0 pour les deux codeurs (cf. figures 3.23).

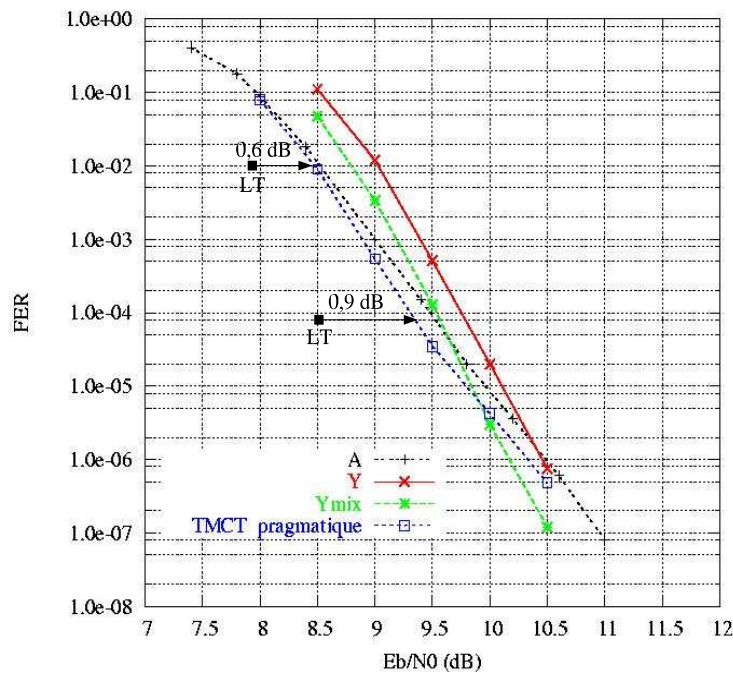


Fig. 3.21 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-64 et du code duo-binaire 8 états du standard DVB-RCT, pour des blocs de 54 octets. Rendement de codage 2/3. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

Sur canal gaussien, $R = 3/5$

Afin d'évaluer la convergence du système pour ce rendement de codage, nous présentons sur la figure 3.24 les résultats de simulation correspondant au schéma A associé aux codes à 8 et à 16 états. A un $FER = 10^{-2}$, les deux associations présentent la même performance qui se situe à 0,9 dB de la limite théorique. Au dessous d'une valeur de $FER = 10^{-3}$, le codeur à 16 états devient plus performant. A un $FER = 10^{-4}$, les résultats de simulation se situent à 1 dB de la limite théorique.

La figure 3.25 présente les résultats de simulation correspondant aux schémas Y et Ymix pour les deux codeurs considérés. On constate que la meilleure performance asymptotique est obtenue avec le codeur à 16 états associé au schéma Ymix.

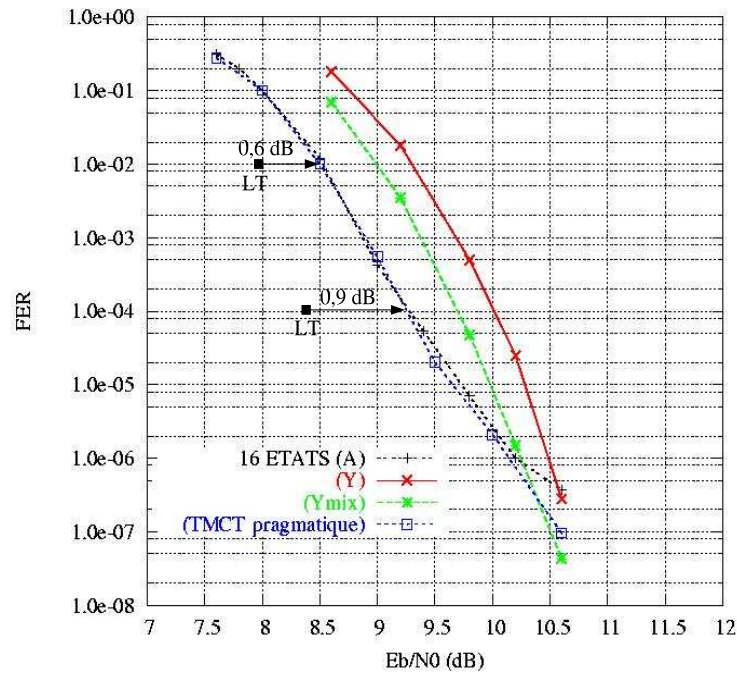


Fig. 3.22 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-64 et du turbocode duo-binaire 16 états, pour des blocs de 54 octets. Rendement de codage 2/3. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

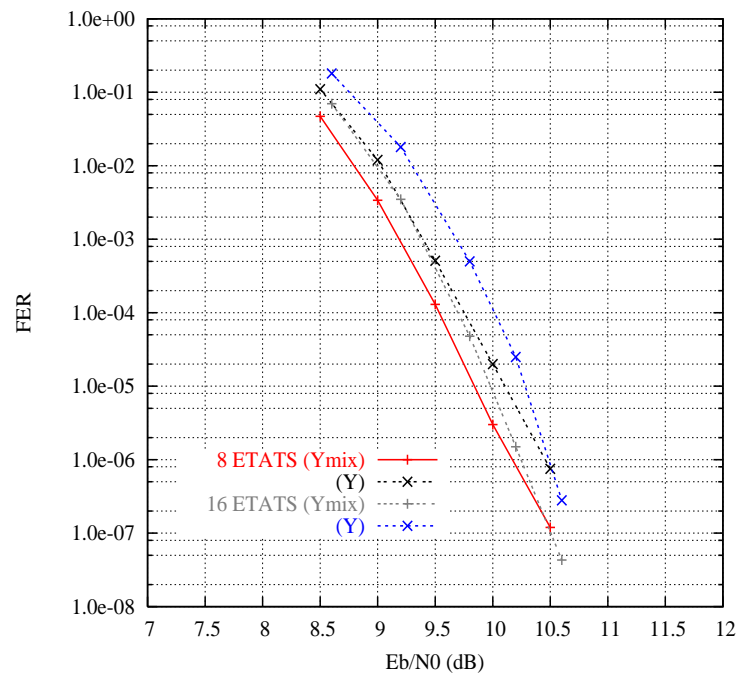


Fig. 3.23 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-64 et du turbocode duo-binaire 8 ou 16 états, pour des blocs de 54 octets. Rendement de codage 2/3. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

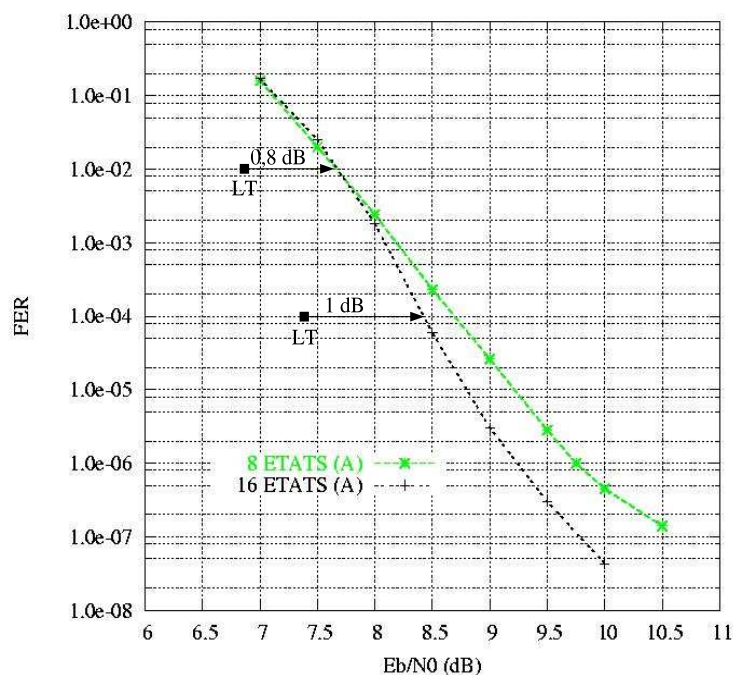


Fig. 3.24 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-64 et du turbo-code duo-binaire à 8 et à 16 états pour des blocs de 54 octets. Rendement de codage 3/5. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

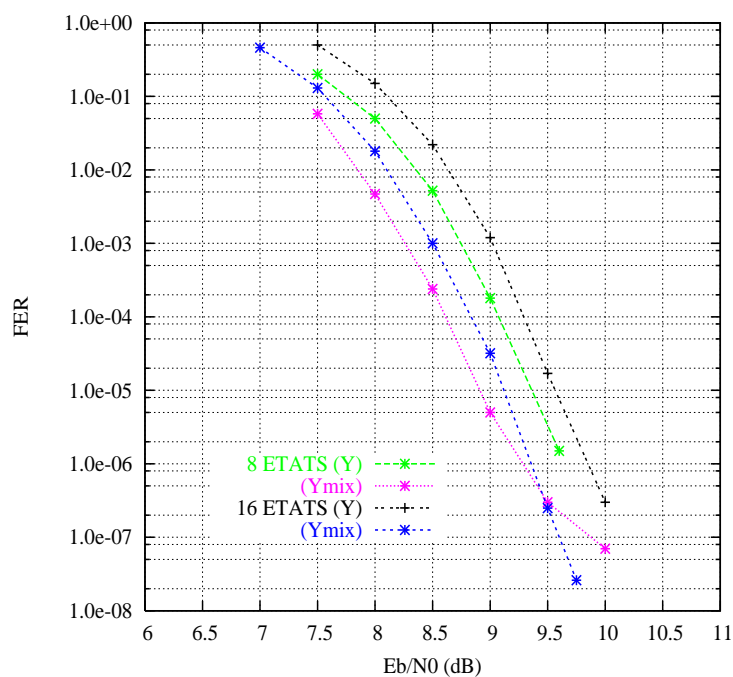


Fig. 3.25 — Performance sur canal gaussien de l'association pragmatique d'une MAQ-64 et du code duo-binaire 8 ou 16 états, pour des blocs de 54 octets. Rendement de codage 3/5. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

Sur canal de Rayleigh, $R = 2/3$

La figure 3.26 présente l'ensemble de résultats de simulation obtenus pour un rendement de codage $R = 2/3$ sur canal de Rayleigh. On observe que, pour les deux codeurs, le schéma de type A présente la meilleure convergence, mais la courbe présente une nette dégradation à partir du $FER=3.10^{-6}$. Le schéma de type Ymix associé au codeur à 16 états présente la meilleure performance à très faible taux d'erreurs.

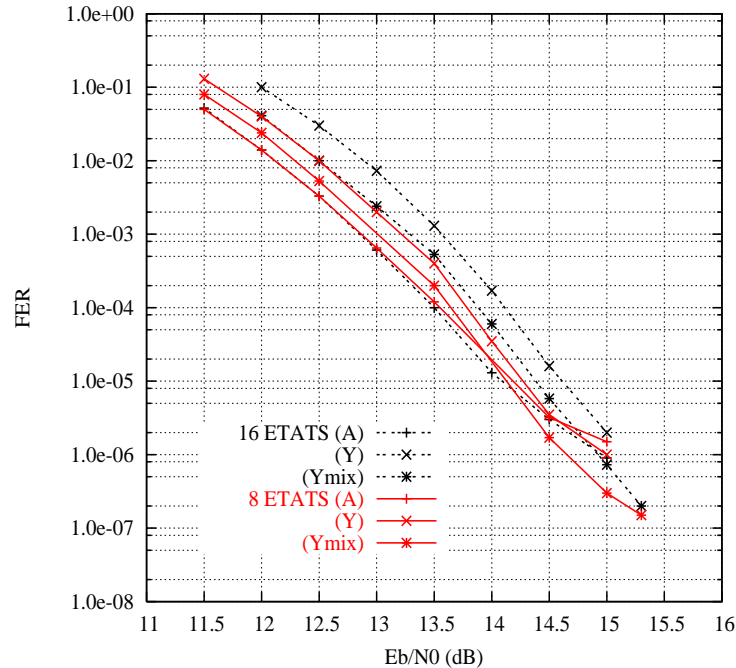


Fig. 3.26 — Performance sur canal de Rayleigh de l'association pragmatique d'une MAQ-64 et du turbocode duo-binaire 8 ou 16 états du standard DVB-RCT, pour des blocs de 54 octets. Rendement de codage 2/3. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

Sur canal de Rayleigh, $R = 3/5$

La figure 3.27 présente l'ensemble de résultats de simulation obtenus pour un rendement de codage $R = 3/5$ sur canal de Rayleigh. La meilleure convergence est obtenue avec le schéma A. Lors de l'association de ce schéma avec le codeur à 16 états, le changement de pente n'apparaît qu'un $FER=10^{-6}$. A très faible taux d'erreur ($FER \leq 10^{-7}$) la solution la plus performante est celle associant le codeur à 16 états et le schéma Ymix.

Conclusion

L'ensemble des résultats de simulation obtenus pour les différents codeurs et schémas de construction de symboles nous a permis d'évaluer plusieurs aspects des performances de la modulation MAQ-64 turbocodée pragmatique à faible et à fort rapport signal à bruit.

Notons dans un premier temps que, au vu des résultats, le code à 16 états conserve le bon compromis seuil de convergence/distance minimale lors de son association pragmatique avec

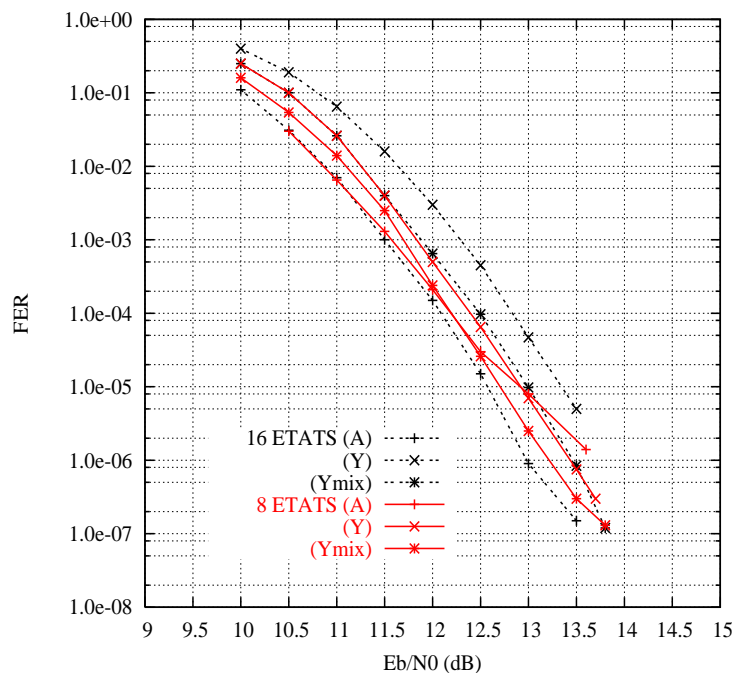


Fig. 3.27 — Performance sur canal de Rayleigh de l'association pragmatique d'une MAQ-64 et du turbocode duo-binaire 16 états, pour des blocs de 54 octets. Rendement de codage 3/5. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage

la modulation MAQ-64, comme il était le cas avec la MAQ-16.

Sur canal gaussien la meilleure convergence est obtenue avec un schéma de construction de symboles favorisant la protection des bits systématiques, que ce soit A ou TTCM pragmatique. Sur canal de Rayleigh, nous avons également observé dans tous les cas d'étude que le schéma A présente la meilleure convergence. On constate ainsi qu'un renforcement de la protection des bits systématiques entraîne une bonne convergence du processus itératif. En effet, dans le processus de décodage, chaque donnée systématique est utilisée en entrée des deux décodeurs. Par conséquent, une erreur sur un bit systématique en sortie du canal provoque une erreur sur l'entrée des deux décodeurs élémentaires alors qu'une redondance erronée n'affecte l'entrée que d'un des décodeurs élémentaires.

Sur canal gaussien, la meilleure performance à faible taux d'erreurs (aux environs d'un $FER=10^{-7}$) est obtenue avec le schéma Ymix, où les bits de redondance ont un niveau moyen de protection vis-à-vis du bruit. On peut néanmoins imaginer qu'à des taux d'erreurs encore plus faibles, le schéma Y devient plus performant que le Ymix, présentant ainsi un meilleur comportement asymptotique.

Sur canal de Rayleigh, le cas du 8 états nous apporte des informations précieuses sur les performances asymptotiques de l'association. Il semble que l'on obtient un comportement semblable à celui sur canal gaussien mais visible à des taux d'erreurs plus faibles (en dessous d'un $FER \leq 10^{-7}$). L'obtention de résultats de simulation fiables à un si faible taux d'erreurs étant très coûteuse en temps et puissance de calcul, une estimation du comportement asymptotique du code par d'autres moyens que la simulation est à considérer.

3.6 Association pragmatique d'un TCC duo-binaire et d'une MDAP-16 pour des communications satellitaires

Nous avons complété les performances présentées précédemment par des résultats obtenus dans le cadre du projet *Modems for High Order Modulations* (MHOMS), financé par l'Agence Spatiale Européenne et portant sur l'étude de la voie de retour ou *return link* de communications satellitaires.

Les modulations de type MAQ-16 ne sont pas bien adaptées aux communications par satellite car très sensibles aux déformations de la modulation causées par les non-linéarités des amplificateurs dans les satellites. Des modulations à amplitude constante, de type MDP, sont ainsi préférées car elles s'avèrent plus robustes vis-à-vis de ces non linéarités. Néanmoins, au delà de 8 points, les modulations MDP affichent des performances médiocres. Aussi pour ce type de communications, des modulations de type MDAP (Modulation à déplacement d'amplitude et de phase) sont envisagées car elles présentent un bon compromis entre performance et sensibilité aux non-linéarités du canal satellite.

Nous avons décrit dans le premier chapitre (Section 1.4.4) les modulations MDAP ainsi que les paramètres qui les caractérisent. La problématique du choix des valeurs optimales pour ces paramètres a été étudiée en [16]. Deux critères d'optimisation sont possibles : optimisation par rapport à la distance euclidienne minimale et optimisation par rapport à la capacité de canal. Nous avons opté pour le deuxième choix. Ainsi pour la MDAP-16, les valeurs des paramètres retenus sont : $N_1 = 4$, $N_2 = 12$, $\varphi_1 = 0$ et $\rho = 2.7$. Le "Codage binaire à signal" adopté correspond à un *mapping* "quasi-Gray" (cf. figure 3.28) qui minimise le taux d'erreurs binaires en sortie du canal pour les paramètres choisis.

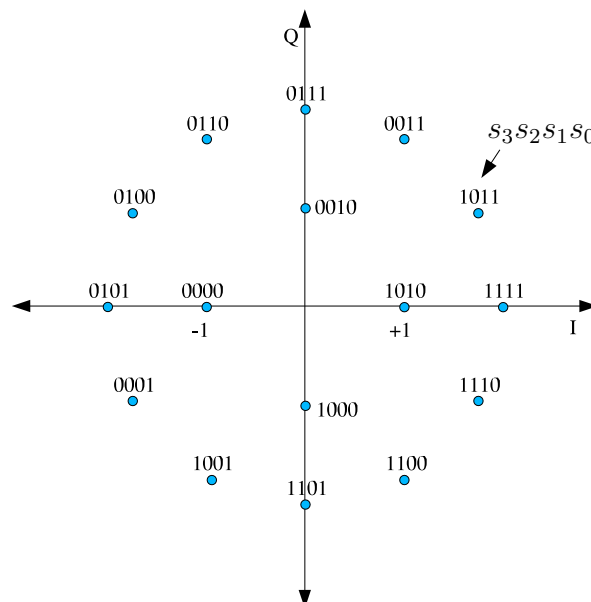


Fig. 3.28 — Constellation et *mapping* de la modulation MDAP-16

Les schémas d'émission et de réception pour ce cas d'association codage/modulation sont identiques à ceux de la MAQ-16, sauf pour le calcul des distances en réception. Les expressions

suivantes sont alors à calculer ⁶ :

$$\begin{aligned}
 \delta(0000) &= 1 + 2I \\
 \delta(0001) &= 1 - 2Q \\
 \delta(0010) &= \rho^2 + 2\rho I \\
 \delta(0011) &= \rho^2 - 2\rho \left(I \cos\left(\frac{\pi}{6}\right) + Q \sin\left(\frac{\pi}{6}\right) \right) \\
 \delta(0100) &= 1 + 2Q \\
 \delta(0101) &= 1 - 2I \\
 \delta(0110) &= \rho^2 + 2\rho \left(I \cos\left(\frac{\pi}{6}\right) + Q \sin\left(\frac{\pi}{6}\right) \right) \\
 \delta(0111) &= \rho^2 - 2\rho I \\
 \delta(1000) &= \rho^2 + 2\rho \left(I \sin\left(\frac{\pi}{6}\right) - Q \cos\left(\frac{\pi}{6}\right) \right) \\
 \delta(1001) &= \rho^2 - 2\rho Q \\
 \delta(1010) &= \rho^2 + 2\rho \left(I \cos\left(\frac{\pi}{6}\right) - Q \sin\left(\frac{\pi}{6}\right) \right) \\
 \delta(1011) &= \rho^2 - 2\rho \left(I \sin\left(\frac{\pi}{6}\right) + Q \cos\left(\frac{\pi}{6}\right) \right) \\
 \delta(1100) &= \rho^2 + 2\rho Q \\
 \delta(1101) &= \rho^2 - 2\rho \left(I \sin\left(\frac{\pi}{6}\right) - Q \cos\left(\frac{\pi}{6}\right) \right) \\
 \delta(1110) &= \rho^2 + 2\rho \left(I \sin\left(\frac{\pi}{6}\right) - Q \cos\left(\frac{\pi}{6}\right) \right) \\
 \delta(1111) &= \rho^2 - 2\rho \left(I \cos\left(\frac{\pi}{6}\right) - Q \sin\left(\frac{\pi}{6}\right) \right)
 \end{aligned}$$

où (I, Q) sont les coordonnées du point reçu dans le plan de la constellation.

Cette modulation offre, tout comme la modulation MAQ-16, deux niveaux différents de protection des bits. Nous n'avons toutefois pas considéré les stratégies de construction des symboles de la même façon que pour la MAQ-16. En effet, la stratégie suivie a consisté à introduire un nouveau paramètre lors de simulations : le pourcentage de bits de redondance à attribuer aux places les mieux protégées par la modulation. Le meilleur compromis entre convergence et performance asymptotique a ainsi été déterminé, évitant tout changement de pente prononcé dans les courbes en deçà de $10^{-6}/10^{-7}$ de FER.

La figure 3.29 montre les résultats ainsi obtenus pour les tailles de bloc considérées dans le projet MHOMS. Seul le cas du canal gaussien a été considéré car il s'agit du modèle de canal le plus représentatif dans ce type de transmissions. On observe que les résultats se situent à 1,2-1,3 dB de la limite théorique pour une valeur de FER= 10^{-6} . Le lecteur peut se référer à [17] pour plus de détails à propos de l'étude menée dans le cadre de ce projet.

3.7 Conclusion

Nous avons présenté dans ce chapitre des schémas de modulations turbocodées pragmatiques associant un turbocode convolutif duo-binaire et une modulation à 8, 16 ou 64 états. La souplesse de ce type de modulation turbocodée nous a permis d'obtenir des résultats de simulation pour plusieurs rendements de codage et tailles de bloc sans apporter de modification au codeur ni au décodeur. Nous avons vérifié à partir des résultats de simulation les bonnes performances des turbocodes pour des systèmes à grande efficacité spectrale. Sur canal gaussien, l'écart entre performance simulée et limite théorique pour des FER $\leq 10^{-4}$ est dans tous les cas inférieur à 1 dB, ce qui montre la bonne convergence de la modulation turbocodée. Quant au comportement asymptotique, il existe dans tous les cas une association codeur/schéma de construction de symboles dont la courbe ne présente pas de changement de pente dans la plage de valeurs de rapport signal à bruit simulées. L'ensemble des résultats

⁶Ce sont des expressions simplifiées, la simplification appliquée consistant à enlever les termes qui sont communs à toutes les distances.

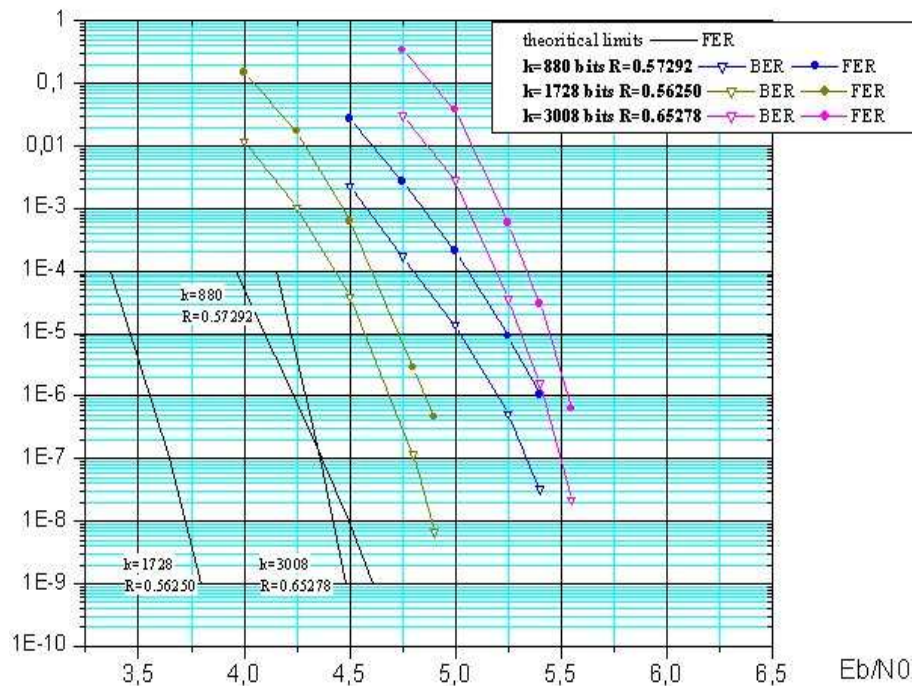


Fig. 3.29 — Performance sur canal gaussien de l'association pragmatique d'une MDAP-16 et d'un turbocode duo-binaire 16 états, pour trois différentes tailles de bloc et rendements de codage 3/5. Décodage Max-Log-MAP

de simulation fournis dans ce chapitre correspond à des performances que l'on peut obtenir en pratique sur des réalisations matérielles. Le fait de travailler avec des conditions idéales de simulation (nombres réels, algorithme de décodage MAP) permettrait de gagner environ 0,2 dB en moyenne sur l'ensemble des performances.

Les nombreuses simulations réalisées nous ont permis de déterminer la stratégie de construction de symboles la plus performante pour les différents cas d'étude. Concrètement nous avons vérifié que la meilleure convergence est obtenue en privilégiant la protection des bits systématiques. En revanche lorsque l'on cherche à avoir de bonnes performances asymptotiques on a intérêt à favoriser la protection de la redondance (schéma Y). Pour un taux d'erreur donné, on peut être amené à utiliser un schéma Y_{mix} ou un schéma permettant de fixer le pourcentage de bits de redondance bien protégés afin de trouver le meilleur compromis convergence/performance asymptotique.

Bien que cette étude nous ait permis de déterminer les stratégies de construction de symboles le plus performantes aussi bien sur canal gaussien que sur canal de Rayleigh, sur ce dernier type de canal l'amélioration des performances de manière globale est à considérer. En effet, une suite possible de notre étude serait l'optimisation des algorithmes de démodulation/décodage pour ce type de canal, en profitant dans une plus grande mesure des informations dont on dispose à propos de l'état du canal.

D'autre part, nous avons constaté que l'obtention par simulation des performances de correction à très faible taux d'erreur (typiquement au dessous d'un $FER=10^{-7}$) est rendue difficile en raison de la puissance de calcul limitée. Nous avons ainsi été amenés à étudier le

problème de l'estimation des performances asymptotiques des modulations turbocodées. Ces travaux sont présentés dans le chapitre suivant.

Estimation de la performance asymptotique des modulations turbocodées

La qualification par simulations des performances à faible bruit des codes correcteurs d'erreurs requiert une puissance de calcul très importante. Ces performances pouvant être estimées lorsque la distance du code est connue, nous nous sommes intéressés aux méthodes permettant d'estimer la distance minimale d'un turbocode. Parmi les travaux menés à ce sujet, nous avons en particulier étudié une méthode récemment introduite : la méthode dite de l'impulsion d'erreur. L'intérêt principal de cette méthode réside dans sa rapidité. Ses principes sont décrits dans la première partie de ce chapitre. Nous abordons ensuite la problématique de son extension aux modulations turbocodées pragmatiques. Nous proposons ainsi une méthode permettant d'estimer les performances asymptotiques de modulations MDP et MAQ turbocodées pour des transmissions sur canal gaussien et sur des canaux à évanouissements de type Rayleigh.

4.1 Performance d'un code correcteur d'erreurs et distance minimale

La borne de l'union permet d'établir une borne supérieure du taux d'erreurs de trames d'un code linéaire sur canal gaussien avec décodage à Vraisemblance Maximale. Elle est donnée par :

$$\text{FER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} m(d) \text{erfc} \sqrt{dR \frac{E_b}{N_0}} \quad (4.1)$$

où :

- $\text{erfc}(x)$ est la fonction d'erreur complémentaire : $\text{erfc}(x) = \frac{2}{\sqrt{x}} \int_x^\infty \exp(-t^2) dt$.
- $m(d)$ est la multiplicité de d , définie comme le nombre de mots de code ayant un poids de Hamming égal à d .

A faible taux d'erreurs, le terme correspondant à d_{\min} domine la valeur de l'expression 4.1, le FER pouvant être approximé par :

$$\text{FER} \simeq \frac{1}{2} m(d_{\min}) \text{erfc} \sqrt{d_{\min} R \frac{E_b}{N_0}} \quad (4.2)$$

L'expression 4.2 montre que la connaissance de la distance minimale du code ainsi que de sa multiplicité rend possible l'estimation analytique des performances d'une transmission codée à faible taux d'erreurs sans nécessiter des simulations coûteuses en temps et en puissance de calcul.

Il existe des méthodes qui permettent de déterminer la distance minimale de codes linéaires en utilisant des propriétés particulières du code, c'est-à-dire sans devoir dresser la liste exhaustive des mots de code. Par exemple, pour les codes en bloc, d_{\min} est le nombre minimal de colonnes de la matrice de parité H linéairement indépendantes ; si la taille de bloc est très grande, son calcul est très complexe. Pour un code convolutif, une analyse du treillis permet d'estimer la distance minimale. Par contre, cette méthode ne peut pas être utilisée dans le cas d'une concaténation de codeurs convolutifs. Il n'est donc pas applicable aux turbocodes. En effet, même lorsque les paramètres d_{\min} et $m(d_{\min})$ des deux codeurs élémentaires sont connus, il est difficile de calculer la valeur de ces paramètres pour le code concaténé parce que la redondance introduite par le second codeur dépend non seulement du message d'origine, mais aussi de la manière dont les données sont entrelacées avant codage. Pour un entrelaceur de taille donnée, il faudrait tenir compte de façon exhaustive de toutes les possibilités d'entrelacer les données. Pour les messages longs, cette méthode devient rapidement trop complexe.

Benedetto *et al* ont introduit dans [11] un modèle d'entrelacement qui rend possible l'évaluation des distances pour tout type de concaténation de codes (parallèle ou série) et pour tout type de codeurs élémentaires (blocs ou convolutifs). Il s'agit d'un modèle d'entrelaceur dit *uniforme* ou *statistique*, dont la caractéristique principale est que l'on peut supposer que toutes les permutations possibles ont la même probabilité. Bien que ce modèle d'entrelacement soit un outil puissant dans l'analyse statistique des performances, il ne fournit que des performances "moyennes" qui peuvent être aisément améliorées à l'aide d'un entrelaceur bien conçu. Par conséquent, il ne serait pas adopté pour des applications réelles et la problématique d'estimation des performances des turbocodes utilisés en pratique reste donc toujours sans solution.

Dans [44], Garello *et al* proposent une méthode permettant de déterminer la distance minimale des turbocodes convolutifs en procédant à une énumération partielle des mots de

codes. Bien que cette méthode présente une complexité réduite par rapport à une énumération exhaustive des mots de code, elle reste toujours coûteuse en temps de calcul.

Récemment, la méthode de l'impulsion d'erreur a été introduite par Berrou *et al* [24] [25]. Il s'agit d'une méthode capable de fournir une estimation du spectre de distances de n'importe quel code linéaire. Elle est alors directement applicable aux turbocodes convolutifs. L'originalité de cette méthode réside dans le fait que, contrairement aux méthodes classiques, elle ne considère pas les propriétés du code mais l'algorithme de décodage qui lui est associé. Elle est basée sur la notion de *réponse à une impulsion d'erreur* du décodeur. Dans le cas d'un décodeur à entrées et sorties pondérées, cette notion a été introduite en [13]. Les principaux avantages de cette méthode sont sa rapidité et son large champ d'application. En particulier, elle peut être appliquée à n'importe quel type de turbocode.

Nous allons d'abord décrire cette méthode pour ensuite aborder l'étude de son extension aux modulations turbocodées.

4.2 La méthode de l'impulsion d'erreur

4.2.1 Principe de la méthode

La figure 4.1 présente le principe de mise en œuvre de la Méthode de l'Impulsion d'Erreur (MIE). Le codeur est alimenté par une séquence d'information de référence, x_{ref} , dont tous les bits ont la valeur binaire "0". L'opération de codage génère des mots de code ne contenant, eux aussi, que des bits "0" car le codeur est linéaire. Ensuite, l'opération de modulation binaire associe un symbole "-1" à chaque "0" émis par le codeur (un symbole "+1" serait associé aux "1" émis par le codeur). C'est ainsi que l'on obtient le mot de code *tout à zéro* après modulation : $x = (-1, -1, \dots, -1)$. Le bloc "Insertion d'une impulsion d'erreur" a pour rôle d'injecter une *impulsion d'erreur*, d'amplitude A_i , à la position i de la séquence codée. La séquence résultante y est égale à x dans toutes les positions, sauf i : $y = (-1, \dots, -1, -1 + A_i, -1, \dots, -1)$. La séquence y est ensuite appliquée en entrée du décodeur. Dans un premier temps, la méthode considère que y est décodée suivant un algorithme de décodage à Vraisemblance Maximale ou ML (*Maximum Likelihood*). Le mot décodé \hat{x} est alors égal à :

$$\hat{x} = \underset{z \in C}{\text{Arg max}} \langle z, y \rangle \quad (4.3)$$

où :

- $\langle z, y \rangle$ est le produit scalaire de z et y ,
- C représente l'ensemble de tous les mots du code considéré.

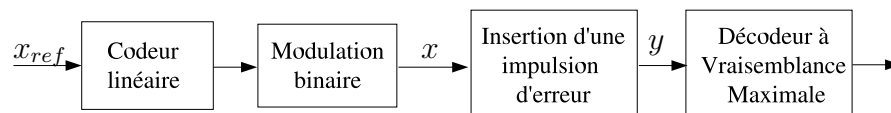


Fig. 4.1 — Schéma mise en œuvre par la Méthode de l'Impulsion d'Erreur (MIE)

On peut également écrire que \hat{x} vérifie :

$$\langle \hat{x}, y \rangle \geq \langle z, y \rangle \quad \forall z \in C \quad (4.4)$$

Le décodeur à Vraisemblance Maximale est capable de récupérer le mot de code émis, et donc de corriger les erreurs, si $|\langle x, x \rangle - \langle x, y \rangle| < d_{\min}$. Si la séquence reçue présente

une impulsion d'erreur à la position i , cette impulsion sera corrigée si $n - (n - A_i) < d_{\min}$, soit $A_i < d_{\min}$. L'application de la méthode consiste alors à ajouter l'amplitude A_i au i -ème symbole, et à incrémenter cette amplitude jusqu'à ce que le décodeur ne soit plus capable de retrouver la séquence *tout à zéro*. Si l'on dénote A_i^* la partie entière de A_i , on peut affirmer que la distance minimale du code est :

$$d_{\min} = \min_i A_i^* \quad (4.5)$$

4.2.2 Algorithme de détermination de la distance minimale

Si l'on suppose que d_{\min} est dans l'intervalle de valeurs $[d_0, d_1]$ où d_0 et d_1 sont deux entiers, d_{\min} peut être déterminée comme suit :

- On fixe $A_{\min} = d_1 + 0.5$
- Pour $i = 1$ jusqu'à n on fait :
 - $A = d_0 - 0.5$;
 - on établit $(\hat{x} = x) = \text{Vrai}$;
 - pendant que $(\hat{x} = x) = \text{Vrai}$ et $(A \leq A_{\min} - 1.0)$, on fait :
 - $A = A + 1.0$
 - $y = x + A \cdot e_i$, avec $e_i = (0, \dots, 0, 1, 0, \dots, 0)$, et le "1" étant dans la position i .
 - décodage ML de y (équation 4.3) $\Rightarrow \hat{x}$ est le mot de code décodé.
 - si $(\hat{x} \neq x)$, $(\hat{x} = x) = \text{Faux}$.
 - $A_{\min} = A$
- d_{\min} est la partie entière de A_{\min} .

Toutes les positions $i = 1, \dots, n$ sont testées. Afin de rendre les calculs plus rapides, A_i^* n'est pas déterminée pour toutes les positions. Si $\hat{x} = x$ avec une impulsion d'erreur A_i plus grande qu'une impulsion d'erreur minimale A_j^* déjà déterminée ($j \leq i$), on passe à la position $i + 1$.

4.2.3 Multiplicité de d_{\min}

La MIE ne fournit pas d'estimation de la multiplicité de d_{\min} . Il est alors nécessaire d'introduire deux hypothèses :

Hypothèse 1 : il existe un seul mot de code de poids A_i^* pour chaque position i .

Hypothèse 2 : toutes les distances A_i^* obtenues pour l'ensemble de positions i ($1 \leq i \leq n$) correspondent à des mots de code différents.

La première hypothèse est optimiste et surévalue la valeur de la multiplicité tandis que la deuxième est pessimiste et la sous-évalue. L'ensemble des expérimentations qui ont été faites ont montré que ces deux phénomènes se compensent dans le cas des turbocodes. Une estimation du FER de la transmission codée peut alors être calculée comme suit [25] :

$$\text{FER} \simeq \frac{1}{2} \sum_{i=1, n} \text{erfc} \left(\sqrt{RA_i^* \frac{E_b}{N_0}} \right) \quad (4.6)$$

4.2.4 Application pratique de la MIE aux turbocodes

Notre but est d'utiliser cette méthode pour estimer la distance minimale d'un turbocode convolutif. Le décodage à Vraisemblance Maximale n'étant pas envisageable pour ce type de

code, nous le remplaçons, dans l'application de la méthode, par un décodage itératif de type turbo. De plus, à cette méthode correspond un modèle de canal à bruit impulsionnel (*i.e.* tout le bruit du canal n'affecte qu'une seule position, le reste de positions n'étant soumises à aucun bruit). Bien que la quasi-optimalité du turbo-décodage ait été prouvée par des simulations avec des motifs d'erreurs réels, il n'est pas certain que le turbo-décodeur minimise la distance entre y et z pour un motif d'erreur de type impulsionnel. Néanmoins, les expérimentations réalisées montrent une bonne concordance entre les distances minimales obtenues en utilisant cette méthode et les performances asymptotiques obtenues par simulation. L'algorithme de décodage à entrées et sorties pondérées que nous avons considéré dans chaque décodeur élémentaire est le Max-Log-MAP. Cet algorithme n'a en effet pas besoin de connaître les paramètres du canal et, par conséquent, il est moins sensible aux changements dans le modèle de bruit.

Une autre considération à prendre en compte est que l'information extrinsèque doit être échangée d'un décodeur élémentaire à l'autre sans aucune modification (pas de gain de boucle, pas d'écrêtage). De même, le nombre d'itérations nécessaires pour une bonne convergence dépend de la valeur de K (longueur de contrainte des codeurs élémentaires). Parfois, des centaines d'itérations sont nécessaires, cela étant dû à la "sévérité" du canal impulsionnel par rapport au canal gaussien classique.

Dans le cas spécifique des turbocodes convolutifs, quelques restrictions sont à considérer dans l'algorithme présenté dans la Section 4.2.2. L'impulsion d'erreur A_i doit seulement être appliquée sur les bits systématiques, pas sur les bits de parité. En effet, un turbocode classique ne corrige pas les redondances et, même s'il les corrigeait, une redondance est liée à un code élémentaire, pas au turbocode complet. De même, le nombre de symboles à tester, noté n_{MIE} , dépend des propriétés de périodicité du code. Dans le cas d'un turbocode, n_{MIE} est le plus petit commun multiple de la période de l'entrelaceur et de la période du motif de poinçonnage, si elles existent. Si la définition de l'entrelacement ne fait apparaître aucun comportement cyclique, n_{MIE} est égal au nombre de bits systématiques dans les séquences codées.

4.2.5 Conclusion

Les résultats présentés dans [24] et [25] montrent des exemples de comparaison entre les résultats de simulation et les performances asymptotiques estimées à partir des valeurs fournies par la MIE pour plusieurs turbocodes standardisés. Les courbes estimées sont à environ 0,2-0,3 dB de celles qui ont été obtenues par simulation, ce qui montre l'efficacité de la méthode. En effet les estimations sont obtenues en quelques secondes de calcul, voire quelques minutes, tandis que le tracé de courbes de simulation demande un temps beaucoup plus grand.

Pour donner au lecteur une idée de l'ordre de grandeur des temps de calcul nécessaires à la simulation d'une courbe de taux d'erreurs, précisons que nous disposons au département Electronique de l'ENST Bretagne de 20 PCs de calcul. Leur utilisation en mono-utilisateur (situation idéale) permet de simuler le turbo-décodage de séquences codées à un débit binaire compris entre 800 kbit/s et 1 Mbit/s. Pour simuler un point à un BER de 10^{-9} , en considérant au moins 100 erreurs binaires simulées, le décodage de 10^{11} bits doit être simulé, ce qui prend environ 30 heures de calcul. Le temps de simulation évoluant linéairement avec la valeur du taux d'erreurs cible, simuler un BER de 10^{-10} nécessiterait environ 300 heures de calcul, soit près de deux semaines.

La MIE est un outil très puissant lors de la conception de codeurs élémentaires et surtout

des entrelaceurs afin d'obtenir des turbocodes convolutifs présentant de bonnes performances à la fois en termes de convergence et de comportement à faibles taux d'erreurs. Concrètement, elle a été souvent utilisée lors de la conception des entrelaceurs présentés dans le Chapitre 3 (cf. Section 3.1.1), pour trouver dans un temps record les valeurs des paramètres P_0 , P_1 , P_2 et P_3 qui maximisent la distance minimale du code pour chacune des tailles de bloc considérées.

4.3 Application de la méthode de l'impulsion d'erreur à l'estimation de la performance asymptotique d'une modulation turbocodée pragmatique sur canal gaussien

Dans la suite de notre étude nous nous sommes intéressés à une extension de la MIE aux modulations turbocodées. Dès le début de l'étude, nous nous sommes aperçus que la MIE ne pourrait pas être directement applicable, en raison de la propriété de linéarité sur laquelle elle est basée. Autrement dit, la MIE repose sur l'utilisation du codage du mot de référence $x_{ref} = (00\dots 0)$ et elle utilise les propriétés de linéarité pour en déduire les performances quel que soit le mot codé. Or, cette propriété n'est pas conservée lorsque l'on utilise des modulations du type MDP- M ou MAQ- M ($M > 4$) avec un codage de Gray. En effet, le niveau de protection du i -ème bit d'un symbole de la constellation ($1 \leq i \leq \log_2 M$) varie suivant le point de la constellation considéré. Par exemple, dans la modulation MDP-8 montrée sur la figure 4.3, on observe que le bit b_2 dans le symbole "101" est mieux protégé que dans "100" car ce symbole "100" est plus proche de la frontière de la région de décision.

Nous avons développé deux méthodes qui permettent d'appliquer les informations fournies par la MIE pour estimer les performances à faible bruit des modulations turbocodées. La première méthode utilise le concept de distance libre de la modulation turbocodée, elle est décrite en Section 4.3.2. La deuxième méthode tient compte des propriétés statistiques de la modulation, conduisant à des estimations plus réalistes. Elle est décrite en Section 4.3.3.

4.3.1 Schéma de transmission

La figure 4.2 montre la chaîne de transmission considérée dans cette partie de l'étude. Le principe de la modulation codée entrelacée au niveau bit (BICM) y est mis en œuvre. Dans ce schéma, aucune hypothèse restrictive n'est imposée sur le turbocode. L'entrelaceur au niveau bit π est supposé idéal (*i.e.* de longueur infinie et totalement aléatoire). Le codeur binaire à signal suit un *mapping* de Gray, caractéristique de l'approche pragmatique. Les modulations à considérer seront du type MDP- M et MAQ- M . En réception, le calcul des LRVs des bits est suivi du désentrelaceur au niveau bit π^{-1} et du turbo-décodeur. Cependant, dans l'étude théorique on suppose que le récepteur est de type à Vraisemblance Maximale, travaillant au niveau des séquences de symboles codés. Nous allons noter $\mathbf{s}_l = (s_1, s_2, \dots, s_l)$ la séquence de signaux en sortie du modulateur et l le nombre de symboles dans la séquence. Chaque signal s_i est un vecteur à deux dimensions dont les coordonnées sont déterminées par la constellation.

4.3.2 Détermination de la performance asymptotique d'une modulation codée : méthode 1

Soit \mathbf{s}_l la séquence codée transmise sur le canal et $\bar{\mathbf{s}}_l$ la séquence choisie par le décodeur à Vraisemblance Maximale. Le probabilité d'erreur par paire $P_2(\mathbf{s}_l, \bar{\mathbf{s}}_l)$ est définie comme la

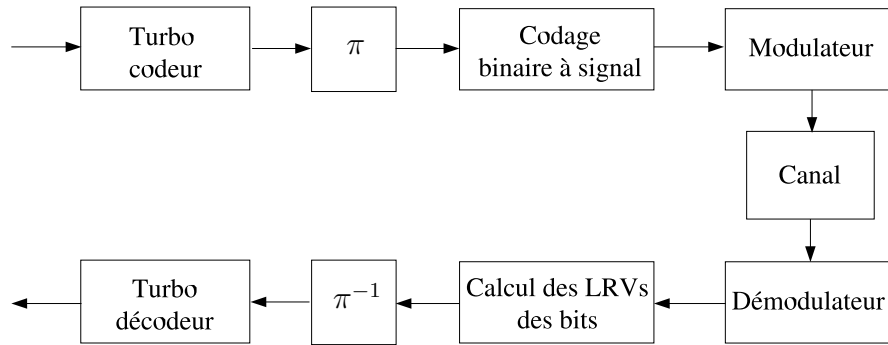


Fig. 4.2 — Schéma de transmission considéré dans l'étude

probabilité que le décodeur choisisse \bar{s}_l au lieu de s_l . Si l'on utilise la borne de l'union, on peut obtenir une borne supérieure de la probabilité d'événement d'erreur en faisant la somme pour toutes les longueurs l , $l = 0, \dots, \infty$ et en considérant toutes les séquences transmises possibles :

$$P_e \leq \sum_{l=1}^{\infty} \sum_{s_l} \sum_{\hat{s}_l \neq s_l} P(s_l) P_2(s_l, \bar{s}_l) \quad (4.7)$$

où $P(s_l)$ est la probabilité *a priori* d'émettre s_l .

Si l'on tient compte du fait que dans le cas d'un canal gaussien avec détection cohérente, la probabilité d'erreur par paire s'exprime sous la forme :

$$P_2(s_l, \bar{s}_l) = \frac{1}{2} \operatorname{erfc} \left(\frac{d_E(s_l, \hat{s}_l)}{2\sqrt{N_0}} \right) \quad (4.8)$$

et qu'à fort rapport signal à bruit la distance libre¹ domine l'équation 4.7, on a tous les éléments pour établir que la performance asymptotique de la modulation turbocodée peut être approximée par :

$$\text{FER} \simeq m(d_{\text{libre}}) \frac{1}{2} \operatorname{erfc} \left(\frac{d_{\text{libre}}}{2\sqrt{N_0}} \right) \quad (4.9)$$

où $m(d_{\text{libre}})$ est la multiplicité de d_{libre} , soit encore le nombre moyen de séquences à une distance d_{libre} de la séquence transmise s_l .

Pour calculer l'estimation donnée par l'équation 4.9, il est impératif de connaître la distance libre de la modulation codée, ainsi que sa multiplicité. Compte tenu de l'entrelacement idéal au niveau bit π (voir schéma dans la figure 4.2), il est possible d'adopter l'hypothèse additionnelle suivante :

Hypothèse 3 : un symbole contient au plus un bit opposé entre les séquences s_l et \bar{s}_l .

Considérant cette hypothèse ainsi que le *mapping* de Gray, on peut écrire :

$$d_{\text{libre}}^2 = d_{\min} d_0^2 \quad (4.10)$$

où :

d_0 : distance euclidienne minimale de la constellation, paramètre caractéristique de la modulation,

¹La distance libre de la modulation codée, notée d_{libre} , est définie comme la plus petite distance euclidienne entre deux séquences appartenant à la modulation codée (cf. Section 2.2.1)

d_{\min} : distance minimale de Hamming du turbocode, fournie par la MIE.

De même, la multiplicité $m(d_{\text{libre}})$ est égale à $m(d_{\min})$, multiplicité de la distance minimale du code, fournie par la MIE lorsque les deux hypothèses décrites en Section 4.2.3 sont adoptées. Soit $\{A_i^*\}$ le spectre de distances obtenu après application de la MIE au turbocodeur, le taux d'erreur asymptotique s'en déduit immédiatement.

Cas d'une MDP- M

Dans le cas d'une modulation MDP- M , $d_0 = 2\sqrt{E_s} \sin\left(\frac{\pi}{M}\right)$, l'équation 4.9 devient alors :

$$\text{FER} \simeq \frac{1}{2} \sum_{i=1}^k \text{erfc} \sqrt{\frac{E_s}{N_0} A_i^* \sin^2\left(\frac{\pi}{M}\right)} \quad (4.11)$$

Le FER asymptotique est ainsi calculable à partir du rapport signal à bruit de la transmission et du spectre de distances fourni par la MIE.

Cas d'une MAQ- M

Nous allons considérer, à titre d'exemple, la MAQ-16. Pour cette modulation $d_0 = \sqrt{\frac{2}{5}E_s}$ (cf. tableau 1.1), l'équation 4.9 s'exprime sous la forme :

$$\text{FER} \simeq \frac{1}{2} \sum_{i=1}^k \text{erfc} \sqrt{\frac{A_i^* E_s}{10N_0}} \quad (4.12)$$

Cette première méthode nécessite donc seulement la connaissance de la relation entre d_0 et E_s pour la constellation considérée, et la seule condition pour l'appliquer est que le *mapping* de la modulation soit de type Gray. Néanmoins, on peut s'attendre à ce que les estimations 4.11 et 4.12 restent assez éloignées des résultats de simulation donc des performances réelles du système en raison du fait qu'elles sont basées sur la connaissance de d_{libre} . Autrement dit, on suppose que tous les bits différents en \mathbf{s}_l et $\bar{\mathbf{s}}_l$ appartiennent à des points adjacents dans la constellation. Prenons par exemple la modulation MDP-8 codée de la figure 4.3. Soit la séquence émise $\mathbf{s}_l = \{\text{"111"}, \text{"111"}, \dots, \text{"111"}\}$, l'estimation de l'équation 4.11 ne considère que les séquences décodées erronées contenant :

- d_{\min} symboles de valeur "110" ou "101" (adjacents à "111" dans la constellation),
- $(l - d_{\min})$ symboles de valeur "111".

Nous présentons ci-après une deuxième méthode qui permet de considérer d'autres séquences décodées erronées susceptibles d'influencer le comportement asymptotique de la modulation codée.

4.3.3 Détermination de la performance asymptotique d'une modulation turbocodée : méthode 2

Dans cette deuxième méthode, le codage binaire de chaque point de la constellation est à considérer. Le développement du raisonnement devient alors dépendant de la modulation considérée au contraire de la méthode précédente. Nous allons l'illustrer avec les cas des modulations MDP-8 et MAQ-16. Elle peut directement être étendue aux autres MDP et MAQ.

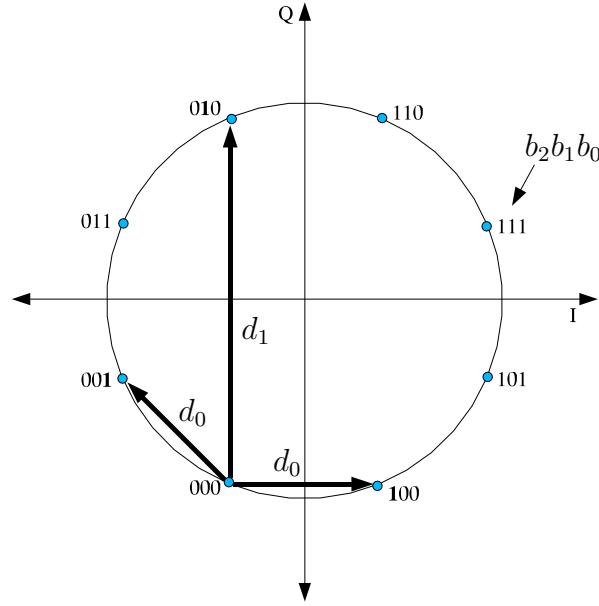


Fig. 4.3 — Représentation des distances d_0 et d_1 pour la modulation MDP-8

Cas de la MDP-8

Pour la modulation MDP-8 représentée sur la figure 4.3, l'analyse peut être divisée en trois étapes :

1ère étape : Soit $d_E(s_i, \bar{s}_i)$ la distance euclidienne entre deux symboles, s_i et \bar{s}_i , différant par un seul bit. La valeur de $d_E(s_i, \bar{s}_i)$ est fonction du bit qui change entre s_i et \bar{s}_i . Sur la figure 4.3, le symbole "000" est pris comme référence pour représenter ces distances :

$$d_E(s_i, \bar{s}_i) = \begin{cases} d_E(\text{"000"}, \text{"001"}) = d_E(\text{"000"}, \text{"100"}) = d_0 = 2\sqrt{E_s} \sin\left(\frac{\pi}{8}\right) \\ d_E(\text{"000"}, \text{"010"}) = d_1 = 2\sqrt{E_s} \sin\left(\frac{3\pi}{8}\right) \end{cases} \quad (4.13)$$

où le bit qui change est représenté **en gras**.

Considérons les séquences codées \mathbf{s}_l et $\bar{\mathbf{s}}_l$ qui, du point de vue du codage, sont à la distance minimale de Hamming d_{\min} (*i.e.* le nombre de bits différents entre elles est égal à d_{\min}). Tenant compte de l'Hypothèse 3, la distance euclidienne au carré entre les séquences \mathbf{s}_l et $\bar{\mathbf{s}}_l$ peut s'écrire :

$$d_E^2(\mathbf{s}_l, \bar{\mathbf{s}}_l) = j d_1^2 + (d_{\min} - j) d_0^2 \quad (4.14)$$

où j est le nombre de bits différents en $\bar{\mathbf{s}}_l$ et \mathbf{s}_l appartenant à des points de la constellation distants de d_1 .

2ème étape : L'analyse détaillée de la probabilité d'erreur d'un bit appartenant à une séquence de symboles modulés par une MDP-8 est présentée dans l'Annexe C. Soient s_i et \bar{s}_i deux symboles appartenant respectivement aux séquences \mathbf{s}_l et $\bar{\mathbf{s}}_l$. A partir des résultats de l'Annexe C, on peut établir :

$$\text{si } s_i \neq \bar{s}_i \Rightarrow \begin{cases} Pr \{d_E(s_i, \bar{s}_i) = d_0\} = \frac{2}{3} \\ Pr \{d_E(s_i, \bar{s}_i) = d_1\} = \frac{1}{3} \end{cases} \quad (4.15)$$

3ème étape : En tenant compte des résultats des deux étapes précédentes, pour un spectre de distances $\{A_i^*\}$ obtenu en appliquant la MIE au turbocode du système, une nouvelle

estimation du taux d'erreurs de trames asymptotique s'écrit :

$$\text{FER} \simeq \sum_{i=1}^k \sum_{j=0}^{A_i^*} \binom{A_i^*}{j} \left(\frac{1}{3}\right)^j \left(\frac{2}{3}\right)^{A_i^*-j} \frac{1}{2} \operatorname{erfc} \left(\frac{\sqrt{j(d_1)^2 + (A_i^* - j)(d_0)^2}}{2\sqrt{N_0}} \right) \quad (4.16)$$

soit encore :

$$\text{FER} \simeq \sum_{i=1}^k \sum_{j=0}^{A_i^*} \binom{A_i^*}{j} \left(\frac{1}{3}\right)^j \left(\frac{2}{3}\right)^{A_i^*-j} \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_s}{N_0} \left\{ j \sin^2 \frac{3\pi}{8} + (A_i^* - j) \sin^2 \frac{\pi}{8} \right\}} \quad (4.17)$$

Cas de la MAQ-16

La figure 4.4 montre la constellation de la MAQ-16 considérée ainsi que les distances à prendre en compte dans l'étude. La procédure d'analyse par étapes appliquée est la même que pour la modulation MDP-8 :

1ère étape : La distance euclidienne entre s_i et \bar{s}_i , symboles différant par un seul bit, dépend des valeurs de s_i et \bar{s}_i , ainsi que du bit qui change entre eux. Si l'on considère par exemple, $s_i = \text{"0111"}$, on peut écrire :

$$d_E(s_i, \bar{s}_i) = \begin{cases} d_E(\text{"0111"}, \text{"0011"}) = d_E(\text{"0111"}, \text{"0110"}) = d_0 \\ d_E(\text{"0111"}, \text{"1111"}) = d_E(\text{"0111"}, \text{"0101"}) = 3d_0 \end{cases} \quad (4.18)$$

La distance euclidienne au carré entre les séquences codées \bar{s}_l et s_l peut s'écrire en fonction de j , nombre de bits différents entre s_l et \bar{s}_l appartenant à des points de la constellation distants de $3d_0$:

$$d_E^2(s_l, \bar{s}_l) = j(3d_0)^2 + (d_{\min} - j)d_0^2 = d_0^2(8j + d_{\min}) \quad (4.19)$$

On notera que dans cette expression, lorsque $j = 0$, $d_E^2(s_l, \bar{s}_l) = d_0^2 d_{\min}$ ce qui correspond à l'expression de d_{libre} (cf. équation 4.10).

2ème étape : D'après les résultats de l'Annexe C on peut d'établir :

$$\text{si } s_i \neq \bar{s}_i \Rightarrow \begin{cases} Pr \{d_E(s_i, \bar{s}_i) = d_0\} = \frac{3}{4} \\ Pr \{d_E(s_i, \bar{s}_i) = 3d_0\} = \frac{1}{4} \end{cases} \quad (4.20)$$

3ème étape : L'estimation statistique du FER asymptotique ainsi obtenue s'exprime alors sous la forme suivante :

$$\text{FER} \simeq \sum_{i=1}^k \sum_{j=0}^{A_i^*} \binom{A_i^*}{j} \left(\frac{1}{4}\right)^j \left(\frac{3}{4}\right)^{A_i^*-j} \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_s}{10N_0} (8j + A_i^*)} \quad (4.21)$$

4.3.4 Comparaison des performances estimées et simulées

Dans un premier temps, nous avons comparé les performances asymptotiques estimées et simulées de l'association pragmatique d'une MDP-8 et des turbocodes présentés dans la Section 3.1 pour une même taille de bloc de 188 octets. L'efficacité spectrale de la transmission est de 2 bit/s/Hz. Sur les figures 4.5 et 4.6 nous avons représenté quatre courbes :

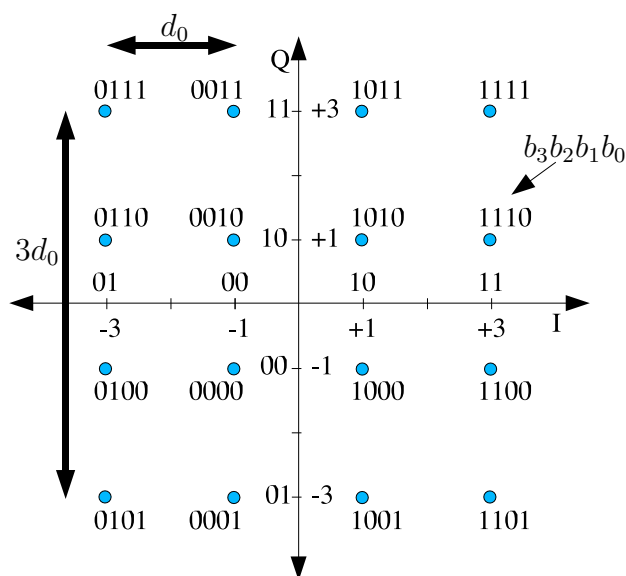


Fig. 4.4 — Constellation et *mapping* de Gray de la modulation MAQ-16

- BICM : résultat de simulation correspondant à la modulation turbocodée entrelacée au niveau bit.
- TMCT pragmatique : résultat de simulation correspondant à un schéma de construction de symboles de type TMCT pragmatique (cf. tableau 3.4).
- Estimation méthode 1 : obtenue à partir de l'expression 4.11 pour $M = 8$.
- Estimation méthode 2 : obtenue à partir de l'expression 4.17.

Pour l'évaluation des expressions 4.11 et 4.17, on ne tient compte, en pratique, que des deux premières valeurs du spectre de distances : d_{\min} , $d_{\min} + 1$ ainsi que de leurs multiplicités $m(d_{\min})$ et $m(d_{\min} + 1)$. Ces valeurs ont été obtenues en appliquant la MIE au turbocode considéré dans chaque association.

Dans le cas du codeur DVB-RCS (cf. figure 4.5) le gain asymptotique est atteint à un $\text{FER}=10^{-6}$ tandis que pour le code 16 états (cf. figure 4.6), il faut descendre en dessous d'une valeur de 10^{-7} de FER pour atteindre le gain asymptotique, la validation de la méthode n'est donc pas évidente parce que l'on est limité par la puissance de calcul des machines.

La courbe correspondant à l'estimation de la méthode 1 donne, comme prévu, une estimation pessimiste du comportement asymptotique. Dans le cas du code 8 états un écart de 1 dB avec les résultats de la simulation BICM peut être observé.

La méthode 2 donne une estimation plus fine, l'écart entre les courbes de simulation BICM et d'estimation étant de seulement 0,4 dB pour le code 8 états.

Nous pouvons, d'autre part, observer que la simulation TMCT présente de meilleures performances que celles de l'estimation de la méthode 2, ce qui était prévisible. Sur la figure 4.5 la courbe 'TTCM pragmatique' croise en effet la courbe 'Estimation méthode 2'. Sur la figure 4.6 on peut imaginer cette tendance sans la visualiser, la puissance limitée de calcul ne permettant pas de simuler dans cette zone à si faible taux d'erreurs dans un temps raisonnable.

Considérons maintenant le cas de la MAQ-16 turbocodée avec un rendement de codage $R = 1/2$ ($\eta = 2$ bit/s/Hz). Nous n'avons traité que le cas 8 états car le code 16 états aurait posé le même problème de puissance de calcul qu'avec la MDP-8. Deux tailles de bloc ont été considérées : 54 octets (figure 4.7) et 188 octets (figure 4.8). Nous avons représenté trois

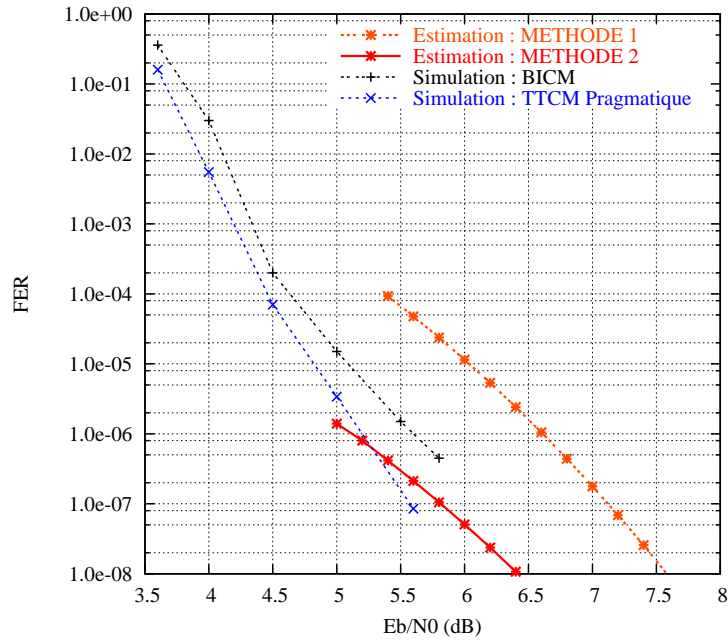


Fig. 4.5 — MDP-8 associée au turbocode 8 états de rendement $R = 2/3$. Courbes de simulation et courbes d'estimation asymptotique du taux d'erreurs de trames. Valeurs fournies par la MIE : $d_{\min} = 13$, $m(d_{\min}) = 0.22$, $d_{\min} + 1 = 14$, $m(d_{\min} + 1) = 0.65$

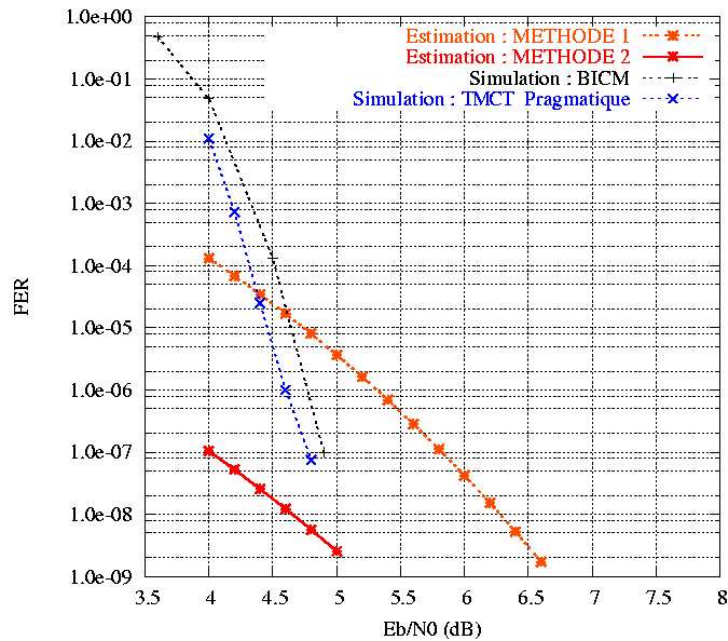


Fig. 4.6 — MDP-8 associée au turbocode 16 états de rendement $R = 2/3$. Courbes de simulation vs. courbes d'estimation asymptotique du taux d'erreurs de trames. Valeurs fournies par la MIE : $d_{\min} = 18$, $m(d_{\min}) = 0.52$, $d_{\min} + 1 = 19$, $m(d_{\min} + 1) = 0.29$

courbes sur ces figures :

- Estimation méthode 1 : obtenue à partir de l'expression 4.12.
- Estimation méthode 2 : obtenue à partir de l'expression 4.21.
- BICM : résultat de simulation correspondant à la modulation turbocodée entrelacée au niveau bit.

On observe que l'estimation asymptotique de la méthode 2 se situe à seulement 0,3 dB de la simulation dans le cas de 188 octets (cf. figure 4.8). Dans le cas de 54 octets on peut observer que la simulation BICM s'avère meilleure que l'estimation par la méthode 2, celle-ci restant néanmoins proche par dessus des résultats de simulation. Ce phénomène met en question la validité de l'Hypothèse 3 pour de petites tailles de bloc : en effet, si la séquence codée n'est pas assez longue il est probable qu'il existe des symboles codés différant de plus d'un bit dans les séquences \mathbf{s}_l et $\bar{\mathbf{s}}_l$. Par conséquent, la distance libre de la modulation codée augmente et les performances s'avèrent meilleures que celles estimées par l'équation 4.21.

Compte tenu des résultats encourageants obtenus dans l'estimation des performances à faible taux d'erreurs sur le canal gaussien, nous avons décidé d'étendre notre étude aux transmissions sur des canaux à évanouissements de type Rayleigh.

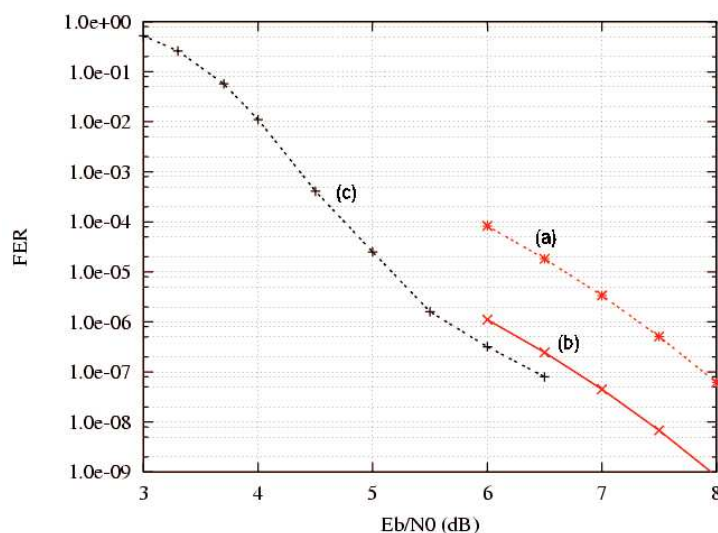


Fig. 4.7 — MAQ-16 associée au turbocode 8 états de rendement $R = 1/2$. Blocs de 54 octets. Valeurs fournies par la MIE : $d_{\min} = 15$, $m(d_{\min}) = 0.5$. (a) : Estimation méthode 1, (b) Estimation méthode 2, (c) Simulation BICM.

4.4 Application de la méthode de l'impulsion d'erreur à l'estimation de la performance asymptotique d'une modulation turbocodée pragmatique sur canal à évanouissements

Le point de départ de l'étude est basé sur les publications concernant les performances des Modulations Codées en Treillis (MCT) sur les canaux à évanouissements. Dans [53], Jamali *et al* présentent la problématique de l'évaluation des bornes de performance des MCTs sur ce type de canaux. L'utilisation de la borne de Chernoff est une solution simple mais peu précise. Dans [72] Slimane *et al* proposent de nouvelles bornes permettant d'approcher les performances réelles des MCTs, leurs résultats étant particulièrement intéressants pour le cas

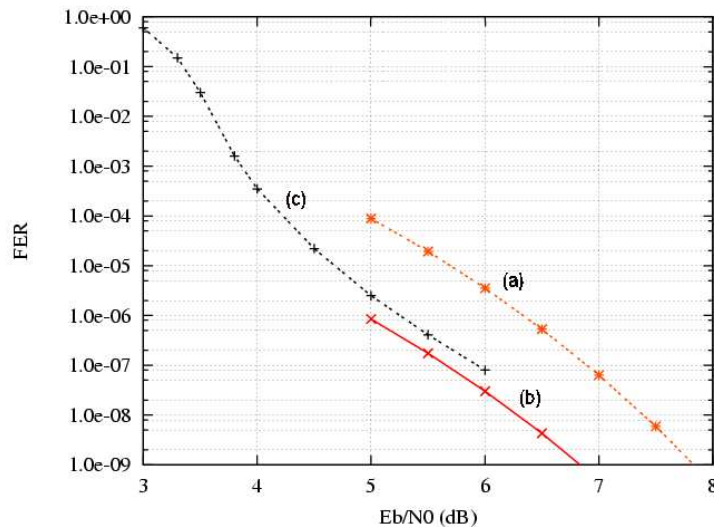


Fig. 4.8 — MAQ-16 associée au turbocode 8 états de rendement $R = 1/2$. Blocs de 188 octets. Valeurs fournies par la MIE : $d_{\min} = 19$, $m(d_{\min}) = 0.125$, $d_{\min} + 1 = 20$, $m(d_{\min} + 1) = 0.375$. (a) Estimation méthode 1, (b) Estimation méthode 2, (c) Simulation BICM.

du canal à évanouissements de Rayleigh avec connaissance parfaite de l'état du canal.

La méthode que nous proposons utilise les résultats de [72] pour estimer la performance asymptotique d'une modulation turbocodée pragmatique sur ce type de canal à évanouissements. Cette démarche a déjà été considérée dans [40], où Duman *et al* présentent des bornes de performances pour des modulations turbocodées sur des canaux à évanouissements, utilisant ces mêmes résultats pour établir des bornes s'avérant particulièrement précises dans le cas du canal de Rayleigh. Les travaux présentés dans [40] sont une extension des études menées par les mêmes auteurs pour le canal gaussien [39], basés sur le concept abstrait d'entrelacement uniforme introduit par Benedetto *et al* dans [12] (cf. Section 4.1). Leurs résultats sont alors conditionnés par l'adoption de cette hypothèse ainsi que par la connaissance de la distribution de poids des mots de code des codeurs élémentaires, une étude spécifique devant être menée pour chaque nouveau schéma considéré.

La méthode que nous avons développée suit le même principe que les méthodes présentées précédemment pour le cas du canal gaussien : utiliser les informations fournies par la MIE à propos du turbocode du système pour estimer la performance de la modulation turbocodée. Elle présente ainsi deux avantages importants par rapport aux travaux de Duman *et al* : elle est valable pour n'importe quel type d'entrelaceur et aucune information n'est requise concernant les codeurs élémentaires.

4.4.1 Schéma de transmission

La schéma de transmission considéré est similaire à celui qui a été utilisé dans le cas du canal gaussien (voir figure 4.2). On adopte un modèle de canal sans mémoire, non sélectif en fréquence, à évanouissements lents de type Rayleigh. On suppose, de plus, que la réception dispose d'une connaissance parfaite de l'état du canal. L'entrelacement au niveau bit (π) du schéma a pour rôle de casser les corrélations dans la séquence d'évanouissements ainsi que d'amener l'ordre de diversité du système à la distance minimale de Hamming du turbocode.

4.4.2 Notations

Dans le schéma de transmission adopté, la séquence de bits d'informations est turbocodée, entrelacée au niveau bit et finalement transformée en séquence de symboles codés par le codeur binaire à signal. Notons $\mathbf{s}_l = (s_1, s_2, \dots, s_l)$ la séquence de signaux en sortie du modulateur et l le nombre de symboles dans la séquence. Chaque signal s_i est un vecteur à deux dimensions dont les coordonnées sont déterminées par la constellation.

Soit $\mathbf{s}_l = (s_1, s_2, \dots, s_l)$ la séquence émise sur le canal et $\mathbf{r}_l = (r_1, r_2, \dots, r_l)$ la séquence reçue en entrée du récepteur. On considère que le décodeur à Vraisemblance Maximale fait une erreur lorsqu'il choisit $\hat{\mathbf{s}}_l = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_l)$ au lieu de \mathbf{s}_l (si $\hat{\mathbf{s}}_l \neq \mathbf{s}_l$).

Le signal reçu à l'instant i s'écrit :

$$r_i = a_i \cdot s_i + n_i \quad (4.22)$$

où a_i est l'amplitude de l'évanouissement et n_i est un échantillon d'un processus de bruit gaussien de moyenne nulle et variance $N_0/2$.

4.4.3 Bornes de la probabilité d'erreur par paire

Puisque l'on adopte un modèle de canal sans mémoire avec connaissance parfaite de son état et que l'on suppose que le décodage est à Vraisemblance Maximale, une borne supérieure de la probabilité d'erreur par paire s'exprime sous la forme [53] :

$$P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \leq \prod_{i=1}^{l_Q} \frac{1}{1 + \frac{1}{4N_0} |s_i - \hat{s}_i|^2} \quad (4.23)$$

où :

- Q est l'ensemble de signaux i pour lesquels $s_i \neq \hat{s}_i$,
- l_Q est le cardinal de Q .

Pour obtenir l'équation 4.23, une borne de Chernoff a été appliquée. Bien que cette borne soit souvent utilisée en raison de sa simplicité, elle n'est pas très précise. En 1995, Slimane *et al* [72] ont développé des nouvelles bornes de la probabilité d'erreur par paire, leurs résultats étant particulièrement intéressants pour le canal de Rayleigh. La caractéristique principale de ces bornes est l'introduction d'un facteur de multiplication, $K(L, 1)$, défini comme :

$$K(L, 1) = \frac{(2L - 1)!!}{2^{L+1} L!} \quad (4.24)$$

où $L = \min(l_Q)$ est la longueur effective du code² et

$$(2L - 1)!! = (2L - 1) \cdot (2L - 3) \cdot \dots \cdot 3 \cdot 1 \quad (4.25)$$

Une approximation précise de la probabilité d'erreur par paire pour de forts rapports signal à bruit est alors donnée par [72] :

$$P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \approx K(L, 1) \prod_{i=1}^L \frac{1}{1 + \frac{1}{4N_0} |s_i - \hat{s}_i|^2} \quad (4.26)$$

²définie comme la longueur du chemin erroné le plus court dans le treillis exprimée en nombre d'étapes parcourues (cf. Section 2.2.2)

4.4.4 Probabilité d'événement d'erreur

La distance produit au carré entre les séquences codées \mathbf{s}_l et $\hat{\mathbf{s}}_l$ est définie par :

$$d_p^2(l_Q) = \prod_{i \in Q} |s_i - \hat{s}_i|^2 \quad (4.27)$$

Une borne supérieure de la probabilité d'événement d'erreur, P_e , peut être obtenue à partir de la borne de l'union [53]. Le terme correspondant à la plus petite l_Q and $d_p^2(l_Q)$ est dominant dans l'expression de P_e pour de forts rapports signal à bruit. On peut ainsi obtenir une approximation de P_e pour de forts rapports signal à bruit :

$$P_e \approx \gamma(L, d_p^2(L)) P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \quad (4.28)$$

où :

- L est la longueur effective de la modulation codée,
- $d_p^2(L)$ est sa distance produit au carré,
- $\gamma(L, d_p^2(L))$ est le nombre moyen de séquences possédant une distance effective L et une distance produit $d_p(L)$.

Néanmoins, cette expression ne peut pas toujours être évaluée dans la mesure où L et/ou $d_p^2(L)$ ne sont pas connus pour la plupart des modulations codées. Nous décrivons par la suite comment la MIE rend possible l'évaluation de l'expression 4.28 pour n'importe quel schéma de modulation turbocodée.

4.4.5 Application de la MIE

Si l'on adopte l'Hypothèse 3 (voir Section 4.3.2) dans cette partie de l'étude, on peut affirmer que la valeur de la longueur effective du code, L , est égale à sa distance minimale de Hamming, d_{\min} . De même, si l'on tient compte du fait que l'expression 4.28 n'est valide que pour les forts rapports signal à bruit, une quatrième hypothèse peut être adoptée :

Hypothèse 4 : $\forall i$, si $s_i \neq \hat{s}_i$, s_i et \hat{s}_i sont des signaux adjacents dans la constellation.

Si d_0 est la distance minimale euclidienne de la constellation, l'Hypothèse 4 peut aussi s'exprimer comme :

$$|s_i - \hat{s}_i| = \begin{cases} 0 & \text{si } s_i = \hat{s}_i \\ d_0 & \text{si } s_i \neq \hat{s}_i \end{cases} \quad (4.29)$$

Si l'on considère cette quatrième Hypothèse, le calcul de la distance produit de la modulation turbocodée devient possible :

$$d_p^2(L) = (d_0^2)^{d_{\min}} \quad (4.30)$$

Finalement, en tenant compte des équations (4.26) et (4.28), l'estimation du FER asymptotique est donnée par :

$$\text{FER} \approx \gamma(d_{\min}, d_p^2(d_{\min})) K(d_{\min}, 1) \left(\frac{1}{1 + \frac{d_0^2}{4N_0}} \right)^{d_{\min}} \quad (4.31)$$

Cette expression peut être évaluée pour n'importe quelle modulation turbocodée en suivant les étapes décrites ci-après :

1ère étape : Application de la MIE au turbocode du système afin de trouver les valeurs de d_{\min} et $\gamma(d_{\min}, d_p^2(d_{\min}))$.

2ème étape : Calcul de $K(d_{\min}, 1)$ (équation 4.24).

3ème étape : Trouver relation entre d_0 et énergie moyenne de la modulation E_m (*i.e.* calcul du coefficient $\omega = \frac{d_0^2}{E_m}$).

4ème étape : Calcul de l'expression suivante pour une valeur de rapport signal à bruit donnée :

$$\text{FER} \approx \gamma(d_{\min}, d_p^2(d_{\min}))K(d_{\min}, 1) \left(\frac{1}{1 + \frac{\omega R_c R_m E_b}{4N_0}} \right)^{d_{\min}} \quad (4.32)$$

où :

- R_c est le rendement du code,
- R_m est le nombre de bits par symbole de modulation.

4.4.6 Comparaison des performances estimées et simulées

Les figures qui suivent montrent les comparaisons des performances estimées avec les performances simulées. On considère la transmission de blocs de 188 octets et les schémas BICM suivants :

- MDP-8 associée au turbocode DVB-RCS, $R_c = 2/3$, sur la figure 4.9,
- MAQ-16 associée au turbocode 16 états, $R_c = 1/2$, sur la figure 4.10.

Les estimations considèrent un modèle de canal sans mémoire et supposent une détection cohérente ainsi qu'un décodage à Vraisemblance Maximale. Les simulations utilisent l'algorithme Max-Log-MAP avec 8 itérations de décodage.

Pour les deux schémas considérés on observe la bonne cohérence des estimations par rapport aux simulations. Dans le cas de la MDP-8 un écart de seulement 0,2 dB sépare les deux courbes pour des valeurs de $\text{FER} \leq 10^{-6}$. Dans le cas de la MAQ-16 l'écart est d'environ 0,5 dB pour un $\text{FER} = 5.10^{-7}$, écart qui semble se réduire pour des valeurs de taux d'erreurs plus faibles.

4.5 Conclusion

Ce chapitre a été entièrement consacré à l'estimation des performances asymptotiques de systèmes de modulations turbocodées. Dans un premier temps, la méthode d'impulsion d'erreur a été présentée en détail. Dans un deuxième temps, nous avons présenté une étude permettant d'utiliser les informations fournies par cette méthode pour évaluer des estimations de performances de modulations turbocodées.

Le résultats obtenus sont très satisfaisants dans la mesure où l'on est capable d'estimer, en quelques secondes et à quelques dixièmes de décibels d'erreur près, des performances qui prendraient des dizaines d'heures de calcul par simulation. L'analyse qui a été effectuée n'a pas de précédent dans la littérature et son avantage majeur réside dans le fait qu'elle est applicable à n'importe quelle modulation turbocodée pragmatique entrelacée au niveau bit car aucune information n'est requise concernant les codeurs élémentaires ou l'entrelaceur. Les résultats obtenus ont été illustrés avec un ensemble de figures comparant les estimations mathématiques avec les résultats de simulation.

Ces travaux ont fait l'objet de plusieurs publications : une première approche de la méthode valable dans le cas de la MDP-8 sur canal gaussien a été présentée dans [22]. Le cas

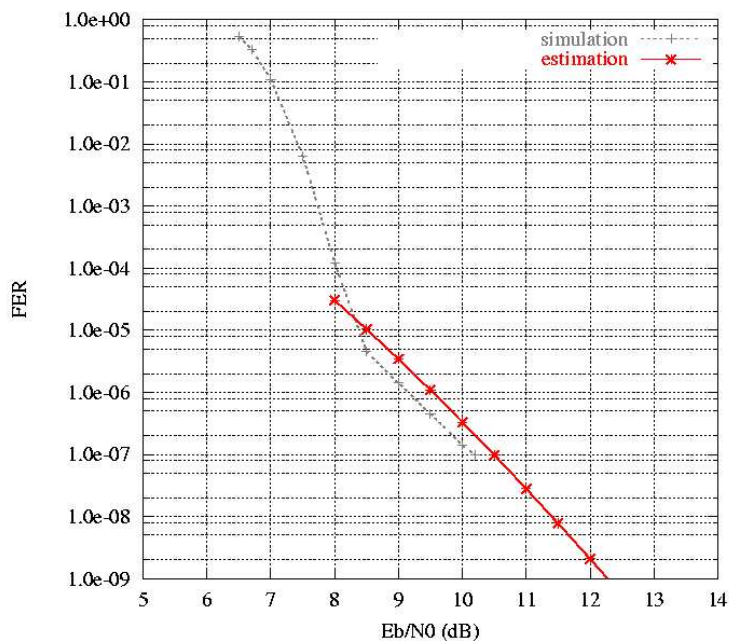


Fig. 4.9 — Estimation de la performance asymptotique et résultats de simulation d'une MDP-8 turbocodée pragmatique sur canal à évanouissements de Rayleigh. Turbocode DVB-RCS, $R = 2/3$. Blocs de 188 octets.

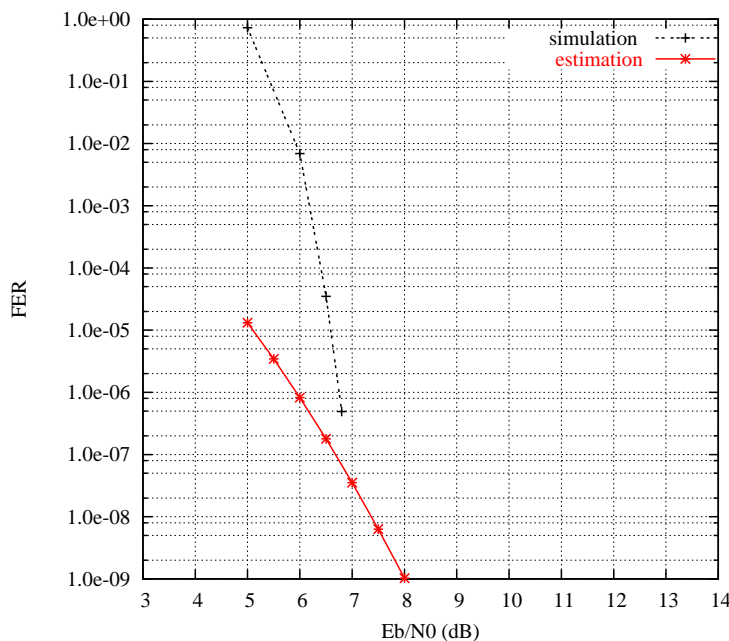


Fig. 4.10 — Estimation de la performance asymptotique et résultats de simulation d'une MAQ-16 turbocodée pragmatique sur canal à évanouissements de Rayleigh. Turbocode à 16 états, $R = 1/2$. Blocs de 188 octets.

de la MAQ-16 turbocodée sur canal gaussien a été présenté dans [76] et celui de la MDP-8 turbocodée sur canal de Rayleigh dans [74]. Finalement, la méthode générale d'estimation sur canal de Rayleigh, valable pour toute modulation classique, a été présentée dans [75].

Une suite possible de ce travail serait de prendre en compte la stratégie de construction des symboles à partir des bits codés dans l'estimation de la performance pour que celle-ci ne soit pas uniquement valable dans le cas d'une structure BICM mais aussi pour les autres schémas de construction de symboles considérés dans la thèse.

Conclusion

CETTE thèse a été consacrée à l'étude de l'association de codes correcteurs d'erreurs très performants, les turbocodes, à des modulations à grand nombre de points. Ce type d'association vise à concevoir des systèmes flexibles à la fois robustes vis-à-vis du bruit et efficaces du point de vue de l'exploitation du spectre occupé.

Nous avons, dans un premier temps, présenté quelques bases concernant les communications numériques, notamment sur le codage de canal et les transmissions numériques sur onde porteuse. Nous avons en particulier décrit le principe du codage convolutif, qui est à la base des turbocodes. Les caractéristiques des différentes modulations numériques retenues dans l'étude (MDP-8, MAQ-16, MAQ-64, MDAP-16) ont été présentées, tout comme les modèles de canal considérés : canal gaussien et canal à évanouissements. Pour ces derniers, un rappel des phénomènes liés aux évanouissements et des paramètres les caractérisant nous a permis de définir le modèle adopté : canal à évanouissements lents de Rayleigh non sélectif en fréquence.

Nous avons ensuite présenté les turbocodes, les aspects suivants étant abordés : structure du turbocodeur convolutif, rôle de l'entrelacement, décodage itératif à entrées et sorties pondérées. Les turbocodes dits duo-binaires ont été retenus dans le cadre de l'étude car ils présentent des performances de correction supérieures à celles des turbocodes classiques (binaires), aussi bien pour les taux d'erreurs élevés que faibles, et ceci pour une complexité de décodage par bit équivalente.

Les techniques de modulations codées étudient l'association des fonctions de codage et de modulation. Parmi celles-ci, le principe et les performances des modulations codées en treillis et des modulations codées entrelacées au niveau bit ont été présentées. Une comparaison de ces deux techniques permet de constater que l'entrelacement au niveau bit apporte des gains de codage importants sur les canaux à évanouissements. En revanche, l'introduction d'un tel entrelacement entraîne une dégradation des performances sur canal gaussien.

Lorsque l'on souhaite introduire un code correcteur d'erreurs puissant tel qu'un turbocode dans une modulation codée, l'approche dite pragmatique constitue la technique la plus simple et offrant la plus grande flexibilité car facilement adaptable à différentes modulations et divers rendements de codage, tout en présentant de bonnes performances de correction. Nous avons ainsi étudié les modulations turbocodées pragmatiques à base de codes duo-binaires pour la transmission de blocs courts.

Deux turbocodes duo-binaires à 8 états et 16 états ont été utilisés. Le premier est celui adopté dans le standard DVB-RCS/RCT. Le deuxième, dénommé TURBO2000, est une extension du premier à 16 états. L'entrelacement y est défini à deux niveaux, intra-symbole et inter-symbole. Le résultat est une permutation globalement très robuste car permettant d'obtenir de bonnes performances aussi bien du point de vue de la convergence que du comportement asymptotique du code. De plus, ces turbocodes font appel à la notion de codage circulaire qui permet de s'affranchir du problème de terminaison du treillis pour le codage et le décodage de données par blocs et simplifie la mise en œuvre du décodage itératif en réception. L'algorithme de décodage élémentaire que nous avons utilisé est l'algorithme sim-

plifié Max-Log-MAP, son adaptation aux turbocodes duo-binaires ayant été détaillée dans l'Annexe B.

Des mesures sur circuit ont permis de constater qu'à un taux d'erreurs de trames de 10^{-4} , les performances du turbocode DVB-RCS/RCT se situent à moins de 0,9 dB de la limite théorique pour les différents rendements considérés. En revanche, les distances minimales limitées de ce turbocode font qu'à faible taux d'erreurs les performances se situent entre 1,4 et 2,0 dB de la limite théorique. D'autre part, les résultats de simulation obtenus pour le codeur TURBO2000 montrent que ce code présente une aussi bonne convergence que celle du code 8 états. De plus, le gain obtenu lors du passage à 16 états de codage est déjà visible à un FER de 10^{-4} , ce code permettant d'approcher les limites théoriques à 0,5 dB. Pour des FER inférieurs à 10^{-7} , le gain en performance asymptotique apporté par le passage de 8 à 16 états est d'au moins 1 dB.

Nous avons adapté le schéma de modulation turbocodée pragmatique proposé par Le Goff dans sa thèse en 1995 aux turbocodes duo-binaires. Nous avons considéré différents types de modulations, avec des nombres de points divers : MDP-8, MAQ-16, MAQ-64 et MDAP-16. Pour les modulations MDP et MAQ, les tailles de blocs et rendements étudiés sont représentatifs d'applications concrètes dans le secteur des technologies sans fils. La modulation MDAP a été étudiée dans le cadre du projet MHOMS (Modems for High Order Modulations), concernant les communications satellitaires à haut débit.

Nous avons présenté les structures générales d'émission et de réception dans le cadre de l'approche pragmatique. Le codage binaire à signal ou *mapping* fait appel à un code de Gray car celui-ci minimise le taux d'erreurs binaires à l'entrée du turbo-décodeur et favorise ainsi la convergence du processus itératif de décodage. Les résultats de simulation obtenus permettent de constater que, du point de vue du seuil de convergence, on se situe dans tous les cas d'étude considérés à moins de 1 dB de la limite théorique. Les résultats de obtenus sont compatibles avec des performances réelles pouvant être mesurées sur un circuit car mettant en œuvre l'algorithme simplifié Max-Log-MAP avec quantification de l'ensemble des données traitées par le décodeur.

D'autre part, une étude sur la mise en forme des symboles a été menée. Les différentes stratégies de construction des symboles à partir des bits issus du turbocode sont déterminées par l'affectation des places les mieux protégées par la modulation aux bits systématiques (schéma dit A) ou aux bits de redondance (schéma dit Y). Pour des valeurs concrètes de rendement de codage et d'ordre de modulation, une stratégie particulière peut être conçue : la TMCT pragmatique. Son principe consiste à ce que les bits appartenant à un même symbole correspondent au même instant de codage.

Afin d'obtenir les meilleures performances pour différents taux d'erreurs cibles, nous avons considéré les différentes combinaisons entre le nombre d'états du codeur et la stratégie de construction des symboles pour pouvoir établir un critère de choix selon que l'on vise à obtenir un faible seuil de convergence ou une bonne performance asymptotique. Cette étude a été particulièrement détaillée pour les modulations MAQ. Nous avons pu conclure que, de manière générale, lorsque l'on cherche à obtenir une bonne convergence, *i.e.* une bonne performance à faible rapport signal à bruit, le schéma à retenir est celui qui favorise la protection des bits systématiques (schéma A). En revanche, lorsque l'on cherche à obtenir un bon comportement asymptotique, on doit favoriser la protection des bits de redondance (schéma Y). Des solutions intermédiaires sont également possibles afin d'obtenir un bon compromis convergence/performance asymptotique. Dans le cas de la MAQ-64, la modulation offre trois niveaux de protection aux bits codés. La stratégie de construction des symboles basée sur

l'association des bits de redondance avec les positions de protection intermédiaires (schéma dit Ymix) offre un bon compromis entre convergence et performance asymptotique, permettant d'obtenir les meilleurs résultats pour des taux d'erreurs de trames situés autour de 10^{-7} . Dans le cas de la modulation MDAP-16, une recherche, par simulations successives, de la valeur minimale du pourcentage de bits de redondance affectés aux positions les mieux protégées permettant d'éviter l'effet de *flattening* aux taux d'erreurs visés ($FER \simeq 10^{-6}/10^{-7}$) nous a permis d'atteindre un bon compromis entre seuil de convergence et performance asymptotique.

Sur canal de Rayleigh, l'opportunité de favoriser la protection des bits de redondance pour les faibles taux d'erreurs n'apparaît pas aussi clairement sur les résultats de simulation que dans le cas gaussien. Pour les taux d'erreurs de trames simulés, jusqu'à 10^{-7} , on ne peut que supposer, au vu des pentes des courbes obtenues, qu'une meilleure protection des bits de redondance conduit à de meilleures performances asymptotiques. Nos moyens de simulation ne nous ont pas permis de le prouver.

La dernière partie du document est enfin consacrée à l'étude de la performance asymptotique des modulations turbocodées pragmatiques par des moyens autres que la simulation. Le point de départ de l'analyse est la Méthode de l'Impulsion d'Erreur (MIE), méthode permettant d'obtenir une estimation de la distance minimale d'un turbocode convolutif ainsi que de sa multiplicité, sous certaines hypothèses. Notre analyse a alors consisté à utiliser les informations fournies par la MIE dans l'estimation des performances asymptotiques des MDP/MAQ turbocodées pragmatiques, ce type d'analyse n'ayant pas d'antécédent dans la littérature.

Sur canal gaussien, deux méthodes ont été proposées. La première consiste à évaluer une expression, obtenue à partir de la borne de l'union, qui approxime le taux d'erreurs asymptotique de la modulation codée. Cette évaluation est possible à partir de la connaissance du spectre de distances du turbocode, fourni par la MIE. Cependant, l'estimation ainsi obtenue reste assez éloignée des résultats de simulation car elle ne tient compte que de la distance libre de la modulation codée, *i.e.* elle suppose que tous les symboles différents dans la séquence émise et la séquence décodée erronée correspondent à des points adjacents dans la constellation. La deuxième méthode permet de considérer d'autres types de séquences erronées susceptibles d'influencer les comportement asymptotique de la modulation codée. Pour cela, une analyse détaillée des caractéristiques statistiques des modulations considérés a été réalisée. L'estimation ainsi obtenue permet de s'approcher jusqu'à 0,3 dB (selon les cas) des résultats de simulation.

Sur canal de Rayleigh, le point de départ de l'étude a été le calcul de bornes proposé par Slimane *et al* pour l'évaluation des performances des modulations codées en treillis sur des canaux à évanouissements. Nous avons montré que, dans le cadre des modulations turbocodées pragmatiques, l'utilisation de ces bornes est possible lorsque l'on considère le principe de l'entrelacement au niveau bit et on travaille à fort rapport signal à bruit. De même, leur évaluation est possible à partir des informations fournies par la MIE. Notre méthode présente ainsi deux avantages importants par rapport aux travaux de Duman *et al* : elle est valable pour n'importe quel type d'entrelacement et aucune information n'est requise concernant les codeurs élémentaires. Nous avons montré avec des exemples concrets que les estimations obtenues dans quelques secondes sont à seulement à 0,2-0,5 dB des résultats de simulation prenant des dizaines d'heures de calcul.

Plusieurs perspectives d'étude sont à considérer à partir du travail présenté dans ce rap-

port de thèse. Dans un premier temps et en tenant compte du fait que l'étude présentée dans le troisième chapitre ne considère que la transmission de blocs courts (54 et 188 octets), l'extension de l'étude à d'autres types d'applications utilisant des blocs plus longs serait à considérer. D'autre part, l'obtention de résultats de simulation pour des modulations turbocodées pragmatiques utilisant des constellations telles que, par exemple, la MAQ-32, MAQ-256, MDAP-32, MDAP-64 permettrait de compléter l'ensemble de résultats déjà obtenus.

D'autre part, dans le cas du canal de Rayleigh, nous avons observé que les modulations turbocodées n'affichent pas la même quasi-optimalité que sur canal gaussien. Une voie naturelle de poursuite de cette étude porterait, par conséquent, sur la compréhension fine de ce comportement quelque peu décevant et sur la recherche de parades qui permettraient aux turbocodes de gagner en performance sur les canaux de Rayleigh et autres canaux non gaussiens.

Enfin, bien que la méthode proposée pour l'estimation des performances asymptotiques des modulations turbocodées conduise à des résultats satisfaisants, elle ne permet toutefois pas de prendre en compte le schéma de construction des symboles à partir des bits codés. Cette problématique serait alors considérée dans une suite possible de l'étude. La méthode pourrait également être appliquée à un éventail plus large de modulations numériques et de canaux à évanouissements, par exemple, le canal de Rice.

Publications

1. C. Berrou, M. Jézéquel, C. Douillard, S. Kerouédan and L. Conde Canencia, “Duo-binary Turbo Codes associated with high-order modulations”, 2nd ESA Workshop on Tracking Telemetry and Command Systems for Space Applications, Noordwijk, The Netherlands, Oct. 2001.
2. C. Berrou, M. Jézéquel, C. Douillard and L. Conde, “Application of the error impulse method in the design of high-order turbo coded modulation”, Proc. Inform. Theory Workshop, ITW’02, pp. 41-44, Bangalore, India, Oct. 2002.
3. L. Conde Canencia, C. Douillard and M. Jézéquel and C. Berrou, “Application of the error impulse method to 16-QAM bit-interleaved turbocoded modulations”, IEE Electronics Letters, March 2003, vol. 39, n. 6, pp. 538-539.
4. L. Conde Canencia and C. Douillard, “Performance estimation of 8-PSK turbocoded modulation over Rayleigh fading channels”, Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC’03, pp. 567-570, Brest, France, Sept. 2003.
5. L. Conde Canencia and C. Douillard, “A new methodology to estimate asymptotic performance of turbocoded modulation over fading channels”, Proc. 2nd Int. Symp. on Image/Video Communications over fixed and mobile networks, ISIVC’04, Brest, France, July 2004.

A

Les codes convolutifs circulaires

L'utilisation d'un code convolutif pour transmettre des blocs n'est pas naturelle. En effet pour décoder un symbole, les algorithmes de décodage (SOVA ou MAP) utilisent l'ensemble des informations reçues du canal, antérieures et postérieures à ce symbole. Les extrémités du bloc ne peuvent donc pas bénéficier à la fois du passé et de l'avenir lors du processus de décodage. Ce problème peut être contourné si le décodeur connaît l'état initial et l'état final du codeur. Il est aisé de forcer l'état initial du codeur par une remise à zéro de l'ensemble des registres. Il est aussi possible soit de transmettre l'état final, soit de le forcer à une valeur connue par exemple en ajoutant des bits de terminaison ou *tail bits* au message initial. Cette technique est connue sous le nom de *fermeture de treillis*.

A.1 Le problème de la fermeture du treillis pour les turbo-codes

Lorsque l'état final du codeur n'est pas connu du décodeur, les informations transmises à la fin du bloc sont alors moins bien protégées car le processus de décodage est moins performant dans cette zone. Ceci peut conduire à une diminution du gain asymptotique. Cette dégradation est fonction de la taille du bloc et peut être compatible avec certaines applications mais inadmissible dans d'autres. La fermeture du treillis est alors à considérer, ce qui est facile pour un code élémentaire. Par contre un turbocode construit sur la base de deux codes élémentaires demande une fermeture simultanée des deux treillis. Les deux techniques suivantes ont été proposées :

1. **Utilisation d'un entrelacement permettant une fermeture automatique du treillis** : il est possible de fermer automatiquement, sans ajout de bits de fermeture, le treillis d'un turbocode à base de codes convolutifs en transformant légèrement le schéma de codage et en utilisant un entrelacement respectant certaines règles de périodicité. Cette solution, décrite dans [20], ne diminue pas l'efficacité spectrale mais impose des contraintes sur l'entrelacement qui rendent difficile la maîtrise du gain de codage et donc des performances à faible taux d'erreurs.
2. **Adoption d'un code circulaire** : un codeur convolutif circulaire garantit que l'état initial et l'état final du treillis sont identiques. Le treillis prend alors la forme d'un cercle ce qui, du point de vue du décodeur, peut être considéré comme un treillis infini [26]. Ce procédé de codage est bien connu sous le nom anglais de *tail-biting* pour les codes non récursifs. Il permet d'allier efficacité spectrale et bonnes performances à forts et faibles taux d'erreurs. Cette technique a été retenue pour les standards DVB-RCS [78]

et DVB-RCT [79]. Elle est présentée ci-après.

Une solution intermédiaire consiste à fermer le treillis d'un seul des codes élémentaires. Cette technique est utilisée dans les standards CCSDS [77] et UMTS [3]. Les bits assurant la fermeture d'un treillis ne sont pas utilisés dans l'autre code élémentaire. Ces bits ne sont donc pas turbocodés ce qui conduit, dans une moindre mesure, aux mêmes inconvénients que la non fermeture du treillis. De plus, la transmission des bits de fermeture engendre une diminution du rendement de codage et donc de l'efficacité spectrale.

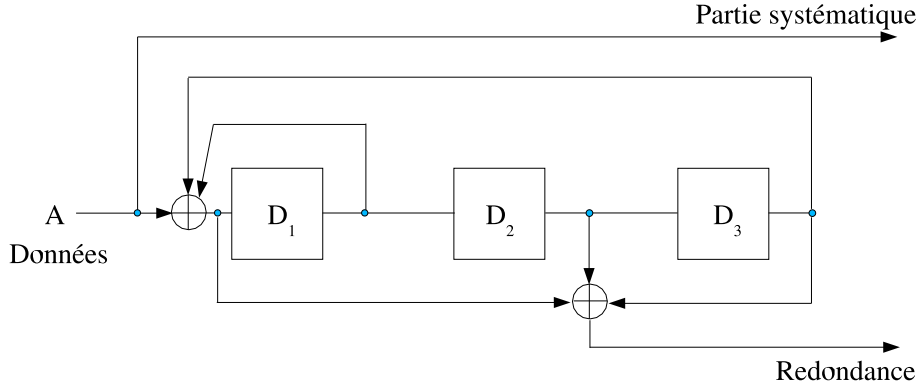


Fig. A.1 — Codeur convolutif récursif avec mémoire de code $\nu = 3$

A.2 Les codes convolutifs récursifs circulaires

Un exemple de codeur convolutif récursif systématique est donné en Figure A.1. A un instant i , l'état \mathbf{S}_i du registre est fonction de l'état précédent \mathbf{S}_{i-1} et du vecteur d'entrée précédent \mathbf{X}_{i-1} . Soit :

$$\mathbf{S}_i = \mathbf{G} \cdot \mathbf{S}_{i-1} + \mathbf{X}_{i-1} \quad (\text{A.1})$$

où \mathbf{G} est la matrice génératrice du code considéré. En prenant l'exemple de la Figure A.1 nous avons :

$$\mathbf{S}_i = \begin{pmatrix} D_1 \\ D_2 \\ D_3 \end{pmatrix} \quad \mathbf{X}_i = \begin{pmatrix} A_i \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 101 \\ 100 \\ 010 \end{pmatrix}$$

A partir de A.1 on peut déduire :

$$\mathbf{S}_1 = \mathbf{G} \cdot \mathbf{S}_0 + \mathbf{X}_0 \quad (\text{A.2})$$

\mathbf{S}_i peut alors être exprimé en fonction de l'état initial et des données fournies au codeur entre les instants 0 et $i - 1$:

$$\mathbf{S}_i = \mathbf{G}^i \mathbf{S}_0 + \sum_{p=1}^i \mathbf{G}^{i-p} \mathbf{X}_{p-1} \quad (\text{A.3})$$

Si k est la longueur de la séquence appliquée à l'entrée du codeur, l'état de circulation \mathbf{S}_c , s'il existe, vérifie $\mathbf{S}_c = \mathbf{S}_0 = \mathbf{S}_k$, soit encore :

$$\mathbf{S}_c = \mathbf{G}^k \mathbf{S}_c + \sum_{p=1}^k \mathbf{G}^{k-p} \mathbf{X}_{p-1} \quad (\text{A.4})$$

ce qui donne :

$$\mathbf{S}_c = \left(\mathbf{I} + \mathbf{G}^k \right)^{-1} \sum_{p=1}^k \mathbf{G}^{k-p} \mathbf{X}_{p-1} \quad (\text{A.5})$$

\mathbf{S}_c existe si et seulement si $(\mathbf{I} + \mathbf{G}^k)$ est inversible. Cette condition n'est pas vérifiée si k est multiple de la période L de la séquence de génération du code récursif car dans ce cas $\mathbf{G}^L = \mathbf{I}$.

Si \mathbf{S}_c existe, le codeur est initialisé à l'état \mathbf{S}_c , il revient à cet état lorsque les k données ont été codées. L'état final et l'état initial peuvent alors être confondus. Le treillis de codage devient un cercle et le code ainsi généré est un code circulaire.

Le calcul de \mathbf{S}_c demande un pré-traitement. Le codeur est tout d'abord initialisé à l'état *tout zéro* puis le message à coder lui est transmis. En utilisant l'équation A.3 l'état final, noté \mathbf{S}_k^0 , peut alors s'écrire sous la forme :

$$\mathbf{S}_k^0 = \sum_{p=1}^k \mathbf{G}^{k-p} \mathbf{X}_{p-1} \quad (\text{A.6})$$

En utilisant A.5 l'état de circulation peut être calculé comme suit :

$$\mathbf{S}_c = \left(\mathbf{I} + \mathbf{G}^k \right)^{-1} \mathbf{S}_k^0 \quad (\text{A.7})$$

En pratique, l'utilisation d'une table permet de déterminer \mathbf{S}_c connaissant \mathbf{S}_k^0 . Le codeur étant initialisé à l'état de circulation le message à coder lui est à nouveau transmis. Cette méthode de fermeture de treillis nécessite un pré-traitement de chaque message avant codage. Ce traitement introduit une latence qui peut être minimisée en utilisant une fréquence de travail plus élevée que le débit binaire des informations à coder. Nous donnons ci-après deux tableaux présentant l'ensemble de valeurs de l'état de circulation pour les deux turbocodeurs duo-binaires étudiés dans le Chapitre 3.

A.3 Etats de circulation pour les turbocodes utilisés dans notre étude

Les tableaux A.1 et A.2 donnent, pour les codes à 8 et à 16 états étudiés en Section 3.1, l'état de circulation en fonction de l'état final de codage à partir de 0, nommé \mathbf{S}_N^0 , et de la taille du bloc, N , mesurée en nombre de couples. Si la taille du bloc est un multiple de 7, pour le code à 8 états, ou 15, pour le code à 16 états, l'état de circulation n'existe pas (voir Section A.2) et on est obligé d'augmenter artificiellement la taille des blocs en ajoutant, par exemple, un octet de données à zéro.

| $N \text{ modulo } 7$ | \mathbf{S}_N^0 | | | | | | | |
|-----------------------|------------------|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0 | 6 | 4 | 2 | 7 | 1 | 3 | 5 |
| 2 | 0 | 3 | 7 | 4 | 5 | 6 | 2 | 1 |
| 3 | 0 | 5 | 3 | 6 | 2 | 7 | 1 | 4 |
| 4 | 0 | 4 | 1 | 5 | 6 | 2 | 7 | 3 |
| 5 | 0 | 2 | 5 | 7 | 1 | 3 | 4 | 6 |
| 6 | 0 | 7 | 6 | 1 | 3 | 4 | 5 | 2 |

Tab. A.1 — Valeurs de l'état de circulation \mathbf{S}_c pour le codeur duo-binaire à 8 états, en fonction de l'état final du codage à partir de 0, \mathbf{S}_N^0 , et du nombre N de couples de données contenus dans chaque bloc

| $N \text{ mod } 15$ | \mathbf{S}_N^0 | | | | | | | | | | | | | | | |
|---------------------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 0 | 14 | 3 | 13 | 7 | 9 | 4 | 10 | 15 | 1 | 12 | 2 | 8 | 6 | 11 | 5 |
| 2 | 0 | 11 | 13 | 6 | 10 | 1 | 7 | 12 | 5 | 14 | 8 | 3 | 15 | 4 | 2 | 9 |
| 3 | 0 | 8 | 9 | 1 | 2 | 10 | 11 | 3 | 4 | 12 | 13 | 5 | 6 | 14 | 15 | 7 |
| 4 | 0 | 3 | 4 | 7 | 8 | 11 | 12 | 15 | 1 | 2 | 5 | 6 | 9 | 10 | 13 | 14 |
| 5 | 0 | 12 | 5 | 9 | 11 | 7 | 14 | 2 | 6 | 10 | 3 | 15 | 13 | 1 | 8 | 4 |
| 6 | 0 | 4 | 12 | 8 | 9 | 13 | 5 | 1 | 2 | 6 | 14 | 10 | 11 | 15 | 7 | 3 |
| 7 | 0 | 6 | 10 | 12 | 5 | 3 | 15 | 9 | 11 | 13 | 1 | 7 | 14 | 8 | 4 | 2 |
| 8 | 0 | 7 | 8 | 15 | 1 | 6 | 9 | 14 | 3 | 4 | 11 | 12 | 2 | 5 | 10 | 13 |
| 9 | 0 | 5 | 14 | 11 | 13 | 8 | 3 | 6 | 10 | 15 | 4 | 1 | 7 | 2 | 9 | 12 |
| 10 | 0 | 13 | 7 | 10 | 15 | 2 | 8 | 5 | 14 | 3 | 9 | 4 | 1 | 12 | 6 | 11 |
| 11 | 0 | 2 | 6 | 4 | 12 | 14 | 10 | 8 | 9 | 11 | 15 | 13 | 5 | 7 | 3 | 1 |
| 12 | 0 | 9 | 11 | 2 | 6 | 15 | 13 | 4 | 12 | 5 | 7 | 14 | 10 | 3 | 1 | 8 |
| 13 | 0 | 10 | 15 | 5 | 14 | 4 | 1 | 11 | 13 | 7 | 2 | 8 | 3 | 9 | 12 | 6 |
| 14 | 0 | 15 | 1 | 14 | 3 | 12 | 2 | 13 | 7 | 8 | 6 | 9 | 4 | 11 | 5 | 10 |

Tab. A.2 — Valeurs de l'état de circulation \mathbf{S}_c pour le codeur duo-binaire à 16 états, en fonction de l'état final du codage à partir de 0, \mathbf{S}_N^0 , et du nombre N de couples de données contenus dans chaque bloc

B

Décodage des turbocodes m -binaires

Cette annexe décrit l'algorithme MAP pour le décodage de codes convolutifs systématiques récurrents m -binaires ainsi que son adaptation pour le décodage itératif. Les codes duo-binaires correspondent au cas particulier $m = 2$.

B.1 Notations

Nous allons utiliser les notations suivantes tout au long de cette annexe :

- Une séquence de données \mathbf{d} est définie par $\mathbf{d} \equiv \mathbf{d}_1^K = (\mathbf{d}_1, \dots, \mathbf{d}_k, \dots, \mathbf{d}_K)$ où \mathbf{d}_k est le vecteur de données m -binaires appliqué à l'entrée du codeur à l'instant k , $\mathbf{d}_k = (d_{k,1}, \dots, d_{k,j}, \dots, d_{k,m})$. La valeur de \mathbf{d}_k pourra également être représentée par la grandeur scalaire entière $i = \sum_{j=1}^m 2^{j-1} \times d_{k,j}$, comprise entre 0 et $2^m - 1$, et l'on écrira alors $\mathbf{d}_k \equiv i$.
- La séquence de sortie du codeur $\mathbf{c} \equiv \mathbf{c}_1^K = (\mathbf{c}_1, \dots, \mathbf{c}_k, \dots, \mathbf{c}_K)$ est constituée de vecteurs de $m + n$ bits $\mathbf{c}_k = (c_{k,1}, \dots, c_{k,j}, \dots, c_{k,m+n})$, où $c_{k,j}$ est un bit systématique si $j \leq m$ et un bit de redondance si $j > m$.
- Dans le cas d'une modulation MDP-2, la séquence modulée $\mathbf{x} \equiv \mathbf{x}_1^K = (\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_K)$ est également constituée de K vecteurs \mathbf{x}_k de taille $m + n$, $\mathbf{x}_k = (x_{k,1}, \dots, x_{k,j}, \dots, x_{k,m+n})$, où $x_{k,j} = 2c_{k,j} - 1$ pour $j = 1, \dots, m + n$.
- La séquence reçue en sortie du canal de transmission est notée $\mathbf{y} \equiv \mathbf{y}_1^K = (\mathbf{y}_1, \dots, \mathbf{y}_k, \dots, \mathbf{y}_K)$, avec $\mathbf{y}_k = (y_{k,1}, \dots, y_{k,j}, \dots, y_{k,m+n})$.
- La suite des états du codeur entre les instants 0 et K est notée $\mathbf{S} = \mathbf{S}_0^K = (S_0, S_1, \dots, S_K)$.

B.2 Décodage selon le critère du Maximum A Posteriori (MAP)

A chaque instant k , les estimations pondérées fournies par le décodeur MAP sont les 2^m probabilités *a posteriori* (en anglais, *A Posteriori Probabilities* ou *APP*) : $Pr(\mathbf{d}_k \equiv i | \mathbf{y})$ pour $i = 0, \dots, 2^m - 1$ (cf. Figure B.1). L'estimation m -binaire correspondante $\hat{\mathbf{d}}_k$ est donnée par la représentation binaire du scalaire i correspondant à la valeur maximale des *APP*.

$$Pr(\mathbf{d}_k \equiv i | \mathbf{y}) = \sum_{(s',s)|d(s',s)\equiv i} Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s' | \mathbf{y}) \quad (\text{B.1})$$

où $(s', s) \mid d(s', s) \equiv i$ désigne l'ensemble des transitions d'état à état $s' \rightarrow s$ dans le treillis causées par $\mathbf{d}_k \equiv i$.

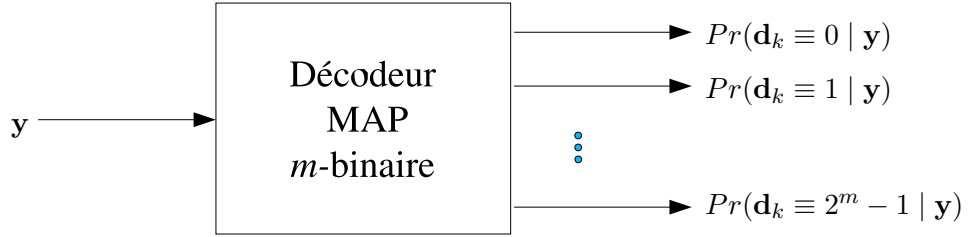


Fig. B.1 — Entrées/sorties d'un décodeur MAP m -binaire à l'instant k

En pratique, on calcule les vraisemblances conjointes $Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s')$. Les *APP* en sont ensuite déduites par :

$$Pr(\mathbf{d}_k \equiv i \mid \mathbf{y}) = \frac{\sum_{(s',s) \mid d(s',s) \equiv i} Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s', \mathbf{y})}{Pr(\mathbf{y})} \quad (\text{B.2})$$

soit :

$$Pr(\mathbf{d}_k \equiv i \mid \mathbf{y}) = \frac{\sum_{(s',s) \mid d(s',s) \equiv i} Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s', \mathbf{y})}{\sum_{(s',s)} Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s', \mathbf{y})} \quad (\text{B.3})$$

Le dénominateur de l'équation B.3 peut être vu comme un simple terme de normalisation commun à toutes les *APP*. En appliquant la séparation classique entre passé et futur, on obtient :

$$Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s', \mathbf{y}) = Pr(\mathbf{y}_{k+1}^K \mid \mathbf{S}_k = s) \cdot Pr(\mathbf{S}_{k-1} = s', \mathbf{y}_1^{k-1}) \cdot Pr(\mathbf{S}_k = s, \mathbf{y}_k \mid \mathbf{S}_{k-1} = s') \quad (\text{B.4})$$

On définit les vraisemblances avant (en anglais *forward*) et arrière (en anglais *backward*) relatives à l'état s , notées respectivement $\alpha_k(s)$ et $\beta_k(s)$, comme suit :

$$\begin{aligned} \alpha_k(s) &= Pr(\mathbf{S}_k = s, \mathbf{y}_1^k) \\ \beta_k(s) &= Pr(\mathbf{y}_{k+1}^K \mid \mathbf{S}_k = s) \end{aligned}$$

De même, la vraisemblance de branche entre les états s' et s , notée $\gamma_k(s', s)$, est définie comme :

$$\gamma_k(s', s) = Pr(\mathbf{S}_k = s, \mathbf{y}_k \mid \mathbf{S}_{k-1} = s') \quad (\text{B.5})$$

On peut montrer que chaque vraisemblance conjointe peut être écrite sous la forme :

$$Pr(\mathbf{S}_k = s, \mathbf{S}_{k-1} = s', \mathbf{y}) = \beta_k(s) \alpha_{k-1}(s') \gamma_k(s, s') \quad (\text{B.6})$$

et que chaque *APP* peut alors être exprimée comme :

$$Pr(\mathbf{d}_k \equiv i \mid \mathbf{y}) = \frac{\sum_{(s',s) \mid d(s',s) \equiv i} \beta_k(s) \alpha_{k-1}(s') \gamma_k(s', s)}{\sum_{(s',s)} \beta_k(s) \alpha_{k-1}(s') \gamma_k(s', s)} \quad (\text{B.7})$$

B.2.1 Calcul des vraisemblances avant et arrière

Les vraisemblances avant et arrière sont calculées de manière récurrente. Pour un code de mémoire ν , la propagation avant conduit à :

$$\alpha_k(s) = \sum_{s'=0}^{2^\nu-1} \alpha_{k-1}(s') \gamma_k(s', s) \quad \text{pour } k = 1, \dots, K$$

L'initialisation de la propagation avant se fait comme suit :

- si l'état de départ du codeur \mathbf{S}_0 est connu $\Rightarrow \alpha_0(\mathbf{S}_0) = 1$ et $\alpha_0(s) = 0$ pour $s \neq \mathbf{S}_0$
- si \mathbf{S}_0 n'est pas connu \Rightarrow tous les $\alpha_0(s)$ sont initialisés à la même valeur.

La propagation arrière conduit à :

$$\beta_k(s) = \sum_{s'=0}^{2^\nu-1} \beta_{k+1}(s') \gamma_{k+1}(s, s') \quad \text{pour } k = K-1, \dots, 0$$

L'initialisation de la propagation arrière se fait comme suit :

- si l'état final du codeur \mathbf{S}_K est connu $\Rightarrow \beta_K(\mathbf{S}_K) = 1$ et $\beta_K(s) = 0$ pour $s \neq \mathbf{S}_K$.
- si \mathbf{S}_K n'est pas connu \Rightarrow tous les $\beta_K(s)$ sont initialisés à la même valeur.

B.2.2 Calcul des vraisemblances de branches

La vraisemblance de branche entre les états s et s' , $\gamma_k(s, s')$ peut être décomposée comme suit :

$$\gamma_k(s, s') = Pr(\mathbf{y}_k | \mathbf{S}_K = s, \mathbf{S}_K = s') Pr(\mathbf{S}_k = s | \mathbf{S}_{k-1} = s') \quad (\text{B.8})$$

soit, si \mathbf{x}_k est le vecteur de symboles modulés correspondant à la transition $s' \rightarrow s$ considérée,

$$\gamma_k(s, s') = Pr(\mathbf{y}_k | \mathbf{x}_k) Pr(\mathbf{S}_k = s | \mathbf{S}_{k-1} = s') \quad (\text{B.9})$$

- Si $s' \rightarrow s$ ne correspond pas à une transition du treillis $\Rightarrow Pr(\mathbf{S}_k = s | \mathbf{S}_{k-1} = s') = 0$ et $\gamma_k(s, s') = 0$.
- Si $s' \rightarrow s$ est une transition existante du treillis \Rightarrow

$$\gamma_k(s, s') = Pr(\mathbf{y}_k | \mathbf{x}_k) Pr^a(\mathbf{d}_k = d(s', s)) \quad (\text{B.10})$$

Le terme $Pr^a(\mathbf{d}_k = d(s', s))$ est la probabilité *a priori* d'émettre le m -tuplet d'information correspondant à la transition $s' \rightarrow s$. Elle est donnée par la statistique de la source. Si les données d'information \mathbf{d}_k sont distribuées de manière équiprobable, $Pr^a(\mathbf{d}_k \equiv i) = 1/2^m$ pour $i = 0, \dots, 2^m - 1$.

$Pr(\mathbf{y}_k | \mathbf{x}_k)$ est la probabilité de transition du canal de transmission considéré. Dans le cas d'un canal gaussien à entrées binaires :

$$\begin{aligned} Pr(\mathbf{y}_k | \mathbf{x}_k) &= \prod_{j=1}^{m+n} Pr(y_{k,j} | x_{k,j}) = \\ &= \prod_{j=1}^{m+n} \left\{ \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_{k,j} - x_{k,j})^2}{2\sigma^2}\right) \right\} = \\ &= \frac{1}{(\sigma\sqrt{2\pi})^{m+n}} \exp\left(-\frac{\sum_{j=1}^{m+n} (y_{k,j} - x_{k,j})^2}{2\sigma^2}\right) \end{aligned} \quad (\text{B.11})$$

$$Pr(\mathbf{y}_k | \mathbf{x}_k) = \frac{1}{(\sigma\sqrt{2\pi})^{m+n}} \cdot \exp\left(-\frac{\sum_{j=1}^{m+n} (y_{k,j}^2 + 1)}{2\sigma^2}\right) \cdot \exp\left(\frac{\sum_{j=1}^{m+n} (y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.12})$$

Dans le cas d'un canal à évanouissements de Rayleigh sans mémoire avec connaissance parfaite de a_k , amplitude de l'évanouissement à l'instant k , cette probabilité est :

$$\begin{aligned} Pr(\mathbf{y}_k | \mathbf{x}_k) &= \prod_{j=1}^{m+n} \left\{ \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_{k,j} - a_k x_{k,j})^2}{2\sigma^2}\right) \right\} = \\ &= \frac{1}{(\sigma\sqrt{2\pi})^{m+n}} \cdot \exp\left(-\frac{\sum_{j=1}^{m+n} (y_{k,j}^2 + a_k^2 x_{k,j}^2)}{2\sigma^2}\right) \cdot \exp\left(\frac{\sum_{j=1}^{m+n} (a_k \cdot y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \end{aligned} \quad (\text{B.13})$$

Chaque *APP* étant un rapport faisant intervenir des vraisemblances de branches au numérateur et au dénominateur (équation B.3), les deux premiers termes de l'expression B.12 qui ne dépendent pas de la transition considérée s'éliminent par division. On peut donc remplacer dans l'expression de $\gamma_k(s, s')$, $Pr(\mathbf{y}_k | \mathbf{x}_k)$ par :

$$Pr'(\mathbf{y}_k | \mathbf{x}_k) = \exp\left(\frac{\sum_{j=1}^{m+n} (y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.14})$$

dans le cas du canal gaussien. D'où :

$$\gamma_k(s, s') = Pr^a(\mathbf{d}_k = d(s, s')) \cdot \exp\left(\frac{\sum_{j=1}^{m+n} (y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.15})$$

Dans le cas du canal de Rayleigh, on peut remplacer dans la même expression $Pr(\mathbf{y}_k | \mathbf{x}_k)$ par :

$$Pr'(\mathbf{y}_k | \mathbf{x}_k) = \exp\left(\frac{\sum_{j=1}^{m+n} (a_k \cdot y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.16})$$

d'où :

$$\gamma_k(s, s') = Pr^a(\mathbf{d}_k = d(s, s')) \cdot \exp\left(\frac{\sum_{j=1}^{m+n} (a_k \cdot y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.17})$$

B.2.3 Normalisation des vraisemblances avant et arrière

En pratique, pour éviter tout problème de précision numérique, on peut normaliser à chaque instant k les vraisemblances avant et arrière sans que cela n'affecte le résultat final (qui est lui-même une valeur normalisée) :

$$\tilde{\alpha}_k(s) = \frac{\alpha_k(s)}{\sum_{s''=0}^{2^\nu-1} \alpha_k(s'')}$$

$$\tilde{\beta}_k(s) = \frac{\beta_k(s)}{\sum_{s''=0}^{2^\nu-1} \beta_k(s'')}$$

B.3 Décodeur MAP pour le décodage itératif d'un turbocode m -binaire

Dans le cas d'un décodage itératif, la brique de base de décodage représentée en Figure B.1 doit être modifiée (cf. Figure B.2) pour :

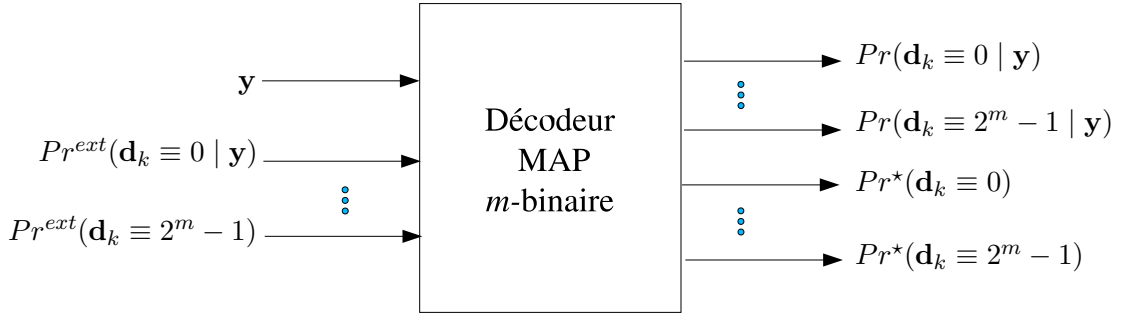


Fig. B.2 — Entrées/sorties d'un décodeur MAP m -binaire à l'instant k , pour le décodage itératif

- prendre en compte les probabilités extrinsèques sur chaque valeur de i , $i = 0, \dots, 2^m - 1$, en provenance de l'autre décodeur.
- produire les probabilités extrinsèques $Pr^*(d_k = i)$, pour $i = 0, \dots, 2^m - 1$.

Les APP $Pr(\mathbf{d}_k \equiv i | \mathbf{y})$, pour $i = 0, \dots, 2^m - 1$, sont identiques à celles produites par le décodeur de la Figure B.1, si ce n'est la probabilité *a priori* modifiée $Pr^\dagger(\mathbf{d}_k \equiv i)$. Cette probabilité a pour expression, à un facteur de normalisation près :

$$Pr^\dagger(\mathbf{d}_k \equiv i) = Pr^a(\mathbf{d}_k \equiv i) \cdot Pr^{ext}(\mathbf{d}_k \equiv i) \quad (\text{B.18})$$

$Pr^{ext}(\mathbf{d}_k \equiv i)$ représente la probabilité extrinsèque relative à $\mathbf{d}_k \equiv i$, fournie par l'autre décodeur élémentaire du décodeur composite. Pour la première demi-itération de décodage, il n'y a pas de probabilité extrinsèque en provenance de l'autre décodeur, $Pr^\dagger(\mathbf{d}_k \equiv i)$ est alors égale à $Pr^a(\mathbf{d}_k \equiv i)$. A chaque itération, le décodeur MAP doit également produire, à chaque instant k , une probabilité extrinsèque associée à chaque valeur possible i de \mathbf{d}_k , $Pr^*(\mathbf{d}_k \equiv i)$. Le calcul de chaque probabilité extrinsèque est obtenue à partir de l'expression suivante :

$$Pr^*(\mathbf{d}_k \equiv i) = \frac{\sum_{(s',s)|d(s',s)\equiv i} \beta_k(s) \alpha_{k-1}(s') \gamma_k^*(s, s')}{\sum_{(s',s)} \beta_k(s) \alpha_{k-1}(s') \gamma_k^*(s, s')} \quad (\text{B.19})$$

Les vraisemblances avant et arrière, $\alpha_k(s)$ et $\beta_k(s)$, peuvent être remplacées par leurs expressions normalisées $\tilde{\alpha}_k(s)$ et $\tilde{\beta}_k(s)$. Les termes sont égaux à :

- Si $s' \rightarrow s$ ne correspond pas à une transition du treillis $\Rightarrow \gamma_k^*(s', s) = 0$.
- Si $s' \rightarrow s$ est une transition existante du treillis $\Rightarrow \gamma_k^*(s', s) = Pr^*(\mathbf{y}_k | \mathbf{x}_k)$, où x_k est le symbole modulé associée à la transition $s' \rightarrow s$ du treillis à l'instant k .

La grandeur $Pr^*(\mathbf{y}_k | \mathbf{x}_k)$ est déduite de la probabilité de transition sur le canal de transmission $Pr(\mathbf{y}_k | \mathbf{x}_k)$, en éliminant les termes faisant intervenir la partie systématique des mots de code.

Dans le cas d'une transmission sur un canal gaussien à entrées binaires, et en partant de l'expression simplifiée $Pr'(\mathbf{y}_k | \mathbf{x}_k)$ de la probabilité de transition, on obtient :

$$Pr^*(\mathbf{y}_k | \mathbf{x}_k) = \exp\left(\frac{\sum_{j=m+1}^{m+n} (y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.20})$$

De la même façon, dans le cas du canal de Rayleigh :

$$Pr^*(\mathbf{y}_k | \mathbf{x}_k) = \exp\left(\frac{\sum_{j=m+1}^{m+n} (a_k \cdot y_{k,j} \cdot x_{k,j})}{\sigma^2}\right) \quad (\text{B.21})$$

B.4 L'algorithme simplifié Max-Log-MAP

L'algorithme MAP requiert un grand nombre d'opérations dont les calculs d'exponentielles et des multiplications. La réécriture de l'algorithme de décodage dans le domaine logarithmique simplifie les traitements. Les estimations pondérées fournies par le décodeur Max-Log-MAP (Figure B.3) sont alors des grandeurs proportionnelles aux logarithmes des APP, dites Log-APP, et notées L :

$$L_k(i) = -\frac{\sigma^2}{2} \ln Pr(\mathbf{d}_k \equiv i \mid \mathbf{y}) \quad i=0, \dots, 2^m - 1 \quad (\text{B.22})$$

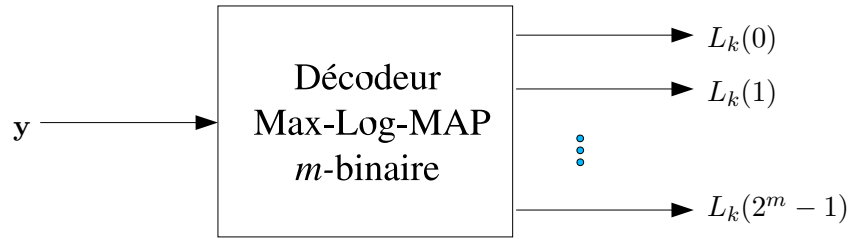


Fig. B.3 — Entrées et sorties d'un décodeur Max-Log-MAP m -binaire à l'instant k

On définit les métriques avant (ou *forward*) et arrière (ou *backward*) relatives au noeud s à l'instant k , $M_k^B(s)$ par :

$$\begin{aligned} M_k^F(s) &= -\sigma^2 \ln \alpha_k(s) \text{ ou indifféremment } M_k^F(s) = -\sigma^2 \ln \tilde{\alpha}_k(s) \\ M_k^B(s) &= -\sigma^2 \ln \beta_k(s) \text{ ou indifféremment } M_k^B(s) = -\sigma^2 \ln \tilde{\beta}_k(s) \end{aligned}$$

La métrique de branche relative à la transition $s' \rightarrow s$ du treillis à l'instant k , $M_k(s', s)$ est égale à :

$$M_k(s', s) = -\sigma^2 \ln \gamma_k(s, s') \quad (\text{B.23})$$

Tenant compte de l'équation B.7, l'expression B.22 donne :

$$L_k(i) = \frac{1}{2} \left[\underbrace{-\sigma^2 \ln \sum_{(s',s) | d(s',s) \equiv i} \beta_k(s) \alpha_{k-1}(s') \gamma_k(s', s)}_{A_k(i)} + \underbrace{\sigma^2 \sum_{(s',s)} \beta_k(s) \alpha_{k-1}(s') \gamma_k(s', s)}_{-B_k} \right] \quad (\text{B.24})$$

Le second terme de l'expression B.24 ne dépend pas de la valeur de i , il est donc commun aux 2^m Log-APP.

En appliquant la simplification Max-Log,

$$\ln(\exp(x) + \exp(y)) \approx \max(x, y) \quad (\text{B.25})$$

on obtient :

$$A_k(i) \approx \min_{(s',s) | d(s',s) \equiv i} (M_k^B(s) + M_{k-1}^B(s') + M_k(s', s)) \quad (\text{B.26})$$

$$B_k \approx \min_{(s',s)} (M_k^B(s) + M_{k-1}^B(s') + M_k(s', s)) \quad (\text{B.27})$$

On remarque qu'avec l'approximation Max-Log, B_k peut s'écrire :

$$B_k = \min_{j=0, \dots, 2^m-1} A_k(j). \quad (\text{B.28})$$

$L_k(i)$ peut donc également être définie comme :

$$L_k(i) = -\frac{\sigma^2}{2} \ln \frac{Pr(\mathbf{d}_k \equiv i | \mathbf{y})}{\max_{j=0, \dots, 2^m-1} Pr(\mathbf{d}_k \equiv j | \mathbf{y})} \quad (\text{B.29})$$

B.4.1 Calcul des métriques de branches

Les métriques de branches sont calculées comme suit :

$$M_k(s', s) = -\sigma^2 \ln \gamma_k(s', s) = -\sigma^2 \ln Pr^a(\mathbf{d}_k = \mathbf{d}(s', s)) - \sigma^2 \ln Pr(\mathbf{y}_k | \mathbf{x}_k) \quad (\text{B.30})$$

où \mathbf{x}_k est le vecteur de symboles modulés correspondant à la transition $s \rightarrow s'$. Si on pose :

$$L_k^a(i) = -\frac{\sigma^2}{2} \ln Pr^a(\mathbf{d}_k \equiv i) \quad (\text{B.31})$$

alors

$$M_k(s', s) = 2L_k^a(d(s', s)) - \sigma^2 \ln Pr(\mathbf{y}_k | \mathbf{x}_k) \quad (\text{B.32})$$

Pour une transmission sur un canal gaussien à entrées binaires :

$$M_k(s', s) = 2L_k^a(d(s', s)) - \sum_{j=1}^{m+n} (y_{k,j} x_{k,j}) \quad (\text{B.33})$$

Pour une transmission sur canal de Rayleigh :

$$M_k(s', s) = 2L_k^a(d(s', s)) - \sum_{j=1}^{m+n} (a_k \cdot y_{k,j} \cdot x_{k,j}) \quad (\text{B.34})$$

B.4.2 Calcul des métriques avant et arrière

Les métriques avant et arrière sont calculées à partir des équations B.2.1 et B.8 en appliquant la simplification Max-Log :

$$M_k^F(s) = \min_{s'=0, \dots, 2^\nu-1} (M_{k-1}^F(s') - M_k(s', s)) \quad (\text{B.35})$$

Dans le cas d'un canal gaussien :

$$M_k^F(s) = \min_{s'=0, \dots, 2^\nu-1} \left(M_{k-1}^F(s') - \sum_{j=1}^{m+n} y_{k,j} \cdot x_{k,j} + 2L_k^a(d(s', s)) \right) \quad (\text{B.36})$$

Dans le cas du canal de Rayleigh :

$$M_k^F(s) = \min_{s'=0, \dots, 2^\nu-1} \left(M_{k-1}^F(s') - \sum_{j=1}^{m+n} a_k \cdot y_{k,j} \cdot x_{k,j} + 2L_k^a d(s', s) \right) \quad (\text{B.37})$$

L'initialisation de la propagation avant se fait comme suit :

- si l'état de départ du codeur \mathbf{S}_0 est connu $\Rightarrow M_0^F(\mathbf{S}_0) = 0$ et $M_0^F(s) = +\infty$ pour $s \neq \mathbf{S}_0$
- si \mathbf{S}_0 n'est pas connu \Rightarrow tous les $M_0^F(s)$ sont initialisés à la même valeur.

De la même façon, les métriques arrière sont calculées comme :

$$M_k^B(s) = \min_{s'=0,\dots,2^\nu-1} (M_{k+1}^B(s') - M_{k+1}(s', s)) \quad (\text{B.38})$$

Dans le cas d'un canal gaussien :

$$M_k^B(s) = \min_{s'=0,\dots,2^\nu-1} \left(M_{k+1}^B(s') - \sum_{j=1}^{m+n} y_{k+1,j} \cdot x_{k+1,j} + 2L_{k+1}^a d(s', s) \right) \quad (\text{B.39})$$

Dans le cas du canal de Rayleigh :

$$M_k^B(s) = \min_{s'=0,\dots,2^\nu-1} \left(M_{k+1}^B(s') - \sum_{j=1}^{m+n} a_{k+1} \cdot y_{k+1,j} \cdot x_{k+1,j} + 2L_{k+1}^a d(s', s) \right) \quad (\text{B.40})$$

L'initialisation de la propagation arrière se fait comme suit :

- si l'état final du codeur \mathbf{S}_K est connu $\Rightarrow M_K^B(\mathbf{S}_K) = 0$ et $M_K^B(s) = +\infty$ pour $s \neq \mathbf{S}_0$
- si \mathbf{S}_K n'est pas connu \Rightarrow tous les $M_K^B(s)$ sont initialisés à la même valeur.

Finalement,

$$A_k(i) = \min_{(s',s)|d(s',s)\equiv i} \left(M_k^B(s) + M_{k-1}^F(s') - \sum_{j=1}^{m+n} y_{k,j} \cdot x_{k,j} \right) + 2L_k^a(i) \quad (\text{B.41})$$

pour un canal gaussien, tandis que pour le canal de Rayleigh :

$$A_k(i) = \min_{(s',s)|d(s',s)\equiv i} \left(M_k^B(s) + M_{k-1}^F(s') - \sum_{j=1}^{m+n} a_k \cdot y_{k,j} \cdot x_{k,j} \right) + 2L_k^a(i) \quad (\text{B.42})$$

Si les données d'information sont uniformément distribuées, toutes les valeurs $2L_k^a(d(s', s))$ sont identiques et peuvent prendre une valeur arbitraire, zéro par exemple. La décision prise par le décodeur est la valeur de i , $i = 0, \dots, 2^m - 1$, qui minimise $A_k(i)$. Notons i_0 cette valeur, on a alors $B_k = A_k(i_0)$ et donc :

$$L_k(i) = \frac{1}{2} (A_k(i) - A_k(i_0)) \text{ pour } i = 0, \dots, 2^m - 1 \quad (\text{B.43})$$

On notera que la présence de coefficient σ^2 dans la définition B.22 de $L_k(i)$ permet de s'affranchir de la connaissance de ce paramètre pour le calcul des métriques et par conséquent pour tout le décodage. C'est un avantage important de la méthode Max-Log-MAP sur la méthode MAP.

B.5 Décodeur Max-Log-MAP ou dual-Viterbi pour le décodage itératif d'un turbocode m -binaire

Dans le cas d'un décodage itératif, la brique de base de décodage représentée en Figure B.3 doit être modifiée (cf. Figure B.4) pour :

- prendre en compte les informations extrinsèques $L_k^{ext}(i)$ sur chaque valeur de $i = 0, \dots, 2^m - 1$, en provenance de l'autre décodeur.
- produire les informations extrinsèques associée $L_k^*(i)$.

Les Log-APP $L_k(i)$, pour $i = 0, \dots, 2^m - 1$, sont identiques à celles produites par le décodeur de la Figure B.3, si ce n'est que les grandeurs $L_k^a(i)$ sont remplacées dans l'équation B.41 par les grandeurs modifiées $L_k^\dagger(i)$:

$$L_k^\dagger(i) = L_k^a(i) + L_k^{ext}(i) \quad (\text{B.44})$$

Pour la première demi-itération de décodage, il n'y a pas d'information extrinsèque en provenance de l'autre décodeur, $L_k^\dagger(i)$ est alors égal à $L_k^a(i)$, pour $i = 0, \dots, 2^m - 1$.

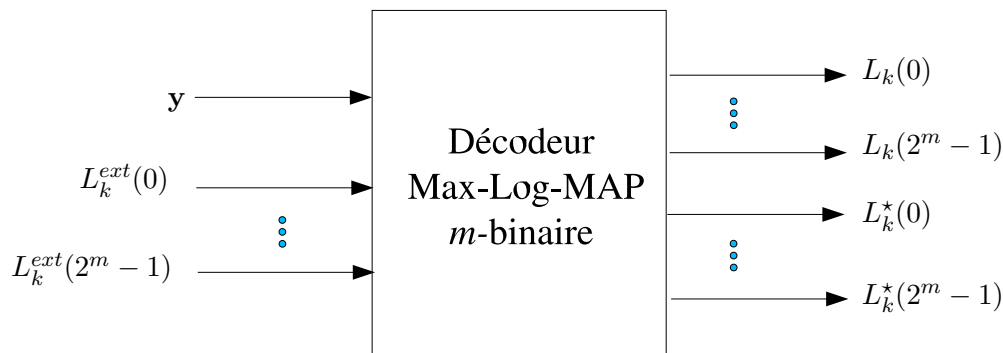


Fig. B.4 — Entrées/sorties d'un décodeur Max-Log-MAP m -binaire à l'instant k , pour le décodage itératif

B.5.1 Extraction de l'information extrinsèque

Pour une valeur de i donnée, les bits systématiques sont les mêmes pour toutes les transitions du treillis telles que $\mathbf{d}_k \equiv i$, donc l'expression B.41 peut être réécrite

$$A_k(i) = \min_{(s',s)|i=d(s',s)} \left(M_k^B(s) + M_{k-1}^F(s') - \sum_{j=m+1}^{m+n} y_{k,j} \cdot x_{k,j} \right) + \sum_{j=1}^m y_{k,j} \cdot x_{k,j} + 2L_k^\dagger(i) \quad (\text{B.45})$$

où x_k est le vecteur de symboles modulés correspondant à $d_k \equiv i$. L'information extrinsèque $L_k^*(i)$ est obtenue en éliminant dans $L_k(i)$ les termes contenant l'information intrinsèque, c'est-à-dire l'information *a priori* modifiée sur d_k , ainsi que l'information systématique :

$$\begin{aligned} L_k^*(i) &= \frac{1}{2} [A_k^*(i) - A_k^*(i_0)] = \\ &= \frac{1}{2} \left[\min_{(s',s)|\mathbf{d}(s',s) \equiv i} \left(M_k^B(s) + M_{k-1}^F(s') - \sum_{j=m+1}^{m+n} y_{k,j} \cdot x_{k,j} \right) \right] - \\ &\quad - \min_{(s',s)|\mathbf{d}(s',s) \equiv i_0} \left[\left(M_k^B(s) + M_{k-1}^F(s') - \sum_{j=m+1}^{m+n} y_{k,j} \cdot x_{k,j} \right) \right] \end{aligned} \quad (\text{B.46})$$

On peut alors écrire :

$$L_k(i) = L_k^* + \frac{1}{2} \sum_{j=1}^{m+1} y_{k,j} [x_{k,j}|_{\mathbf{d}_k \equiv i} - x_{k,j}|_{\mathbf{d}_k \equiv i_0}] + [L_k^\dagger(i) - L_k^\dagger(i_0)] \quad (\text{B.47})$$

Cette expression montre que l'information extrinsèque $L_k^*(i)$ peut, en pratique, être déduite de $L_k(i)$ par simple soustraction. Le facteur $1/2$ dans la définition B.22 de $L_k(i)$ permet d'obtenir une décision pondérée et une information extrinsèque $L_k^*(i)$ à la même échelle que les échantillons bruités $y_{k,j}$.

C

Analyse des probabilités d'erreur binaire dans les modulations étudiées

Nous présentons dans cette Annexe l'analyse de la probabilité d'erreur binaire pour les modulations MDP-8, MAQ-16 et MAQ-64. Cette analyse consiste à calculer la probabilité d'erreur moyenne de chaque position binaire pour chacune des modulations. Les résultats obtenus sont utilisés dans deux parties du rapport : lors du choix de la stratégie de construction des symboles dans le Chapitre 3 et lors de l'estimation des performances dans le Chapitre 4.

C.1 Probabilité d'erreur binaire dans une MDP-8 avec *mapping* de Gray

Soit \mathbf{r} , de coordonnées (r_1, r_2) sur le plan de Fresnel, le signal reçu par le récepteur et $\Theta_r = \tan^{-1} \frac{r_2}{r_1}$ sa phase. Il peut être montré [59] que, dans le cas d'un canal gaussien, la probabilité d'erreur de symbole d'une MDP-8, P_{symb} , s'exprime comme :

$$P_{symb} = 1 - \int_{-\pi/8}^{\pi/8} Pr_{\Theta_r}(\Theta_r) d\Theta_r \quad (C.1)$$

où $Pr_{\Theta_r}(\Theta_r)$ est la fonction de densité de probabilité de Θ_r :

$$Pr_{\Theta_r}(\Theta_r) = \frac{1}{2\pi} \exp\left(-2\frac{E_s}{N_0} \sin^2 \Theta_r\right) \int_0^\infty V \exp\left\{-\left(V - \sqrt{4\frac{E_s}{N_0} \cos \Theta_r}\right)^2 / 2\right\} dV \quad (C.2)$$

avec : $V = \sqrt{r_1^2 + r_2^2}$.

Or, dans notre étude nous ne sommes pas concernés par P_{symb} mais par la probabilité d'erreur de bit, P_{eb} . Cette probabilité dépend de la position occupée par le bit, b_i , dans le symbole $S_j = (b_2 b_1 b_0)$, $j = 0, 1, \dots, 7$, qui est déterminée par le *mapping*. Notre analyse consiste alors à exprimer la probabilité d'erreur de bit pour chaque position binaire. Nous considérons la modulation MDP-8 dont la constellation et le *mapping* sont présentés sur la figure C.1.

Les considérations suivantes sont le point de départ de l'analyse :

1. Notons $I(a, b) = \int_a^b Pr_{\Theta_r}(\Theta_r) d\Theta_r$.

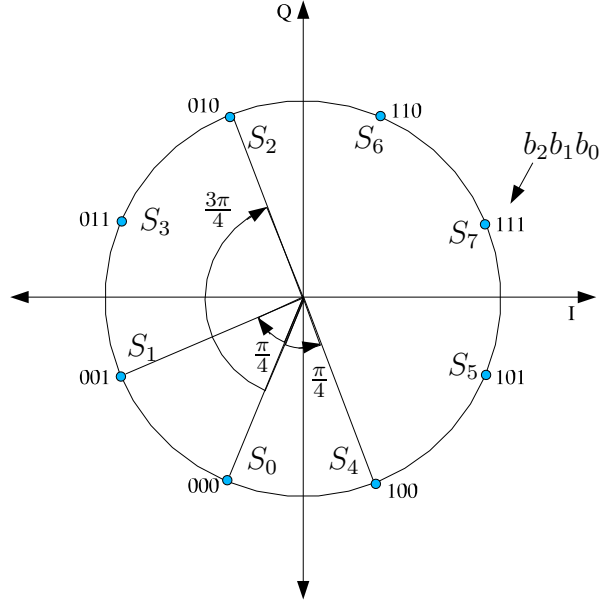


Fig. C.1 — Constellation et *mapping* de la modulation MDP-8

2. La figure C.2 présente l'allure de $Pr_{\Theta_r}(\Theta_r)$ pour différentes valeurs de $\gamma_s = E_s/N_0$. Les courbes correspondant à des valeurs de $\gamma_s : 2 < \gamma_s < 10$ sont celles qui s'adaptent à notre cas d'étude. En tenant compte de l'allure de ces courbes on peut considérer les approximations suivantes :

Approx. 1 : si $|a| > \frac{\pi}{4}$ et $|b| > \frac{\pi}{4} \Rightarrow I(a, b) \simeq I(-\frac{\pi}{4}, \frac{\pi}{4})$

Approx. 2 : si $|a| > \frac{\pi}{4}$ et $|b| \leq \frac{\pi}{4} \Rightarrow I(a, b) \simeq I(-\frac{\pi}{4}, b)$

Approx. 3 : si $|a| \leq \frac{\pi}{4}$ et $|b| > \frac{\pi}{4} \Rightarrow I(a, b) \simeq I(a, \frac{\pi}{4})$

A titre d'exemple, nous allons appliquer en pratique les Approx. 1-3 lors du calcul de la probabilité d'erreur de la position b_2 . Si l'on définit $P_{eb}(b_i|S_j)$ comme la probabilité d'erreur de la position b_i sachant que le symbole S_j a été envoyé, on peut écrire :

- $P_{eb}(b_2|S_1) = 1 - I(-\frac{5\pi}{8}, \frac{3\pi}{8}) \simeq 1 - I(-\frac{\pi}{4}, \frac{\pi}{4})$ (Approx. 1).
- $P_{eb}(b_2|S_0) = 1 - I(-\frac{7\pi}{8}, \frac{\pi}{8}) \simeq 1 - I(-\frac{\pi}{4}, \frac{\pi}{8})$ (Approx. 2).

Graphiquement, cela correspond à considérer l'écart de phase $\Delta\varphi_1$ (resp. $\Delta\varphi_2$) dans le calcul de $P_{eb}(b_2|S_1)$ (resp. $P_{eb}(b_2|S_0)$) au lieu de tout l'écart de phase définissant la région de décision (cf. figure C.3).

Toutes les probabilités $P_{eb}(b_i|S_j)$ ($i = 0, 1, 2; j = 0, 1, \dots, 7$) peuvent être ainsi être déterminées. Afin de simplifier les expressions, nous définissons les probabilités $P_1 = 1 - I(-\frac{\pi}{4}, \frac{\pi}{4})$ et $P_2 = 1 - I(-\frac{\pi}{8}, \frac{\pi}{8}) = 1 - I(-\frac{\pi}{4}, \frac{\pi}{8})$. On obtient :

$$\begin{aligned}
 P_{eb}(b_2|S_j) &= \begin{cases} P_2 & \text{si } j = 0, 2, 4, 6 \\ P_1 & \text{si } j = 1, 3, 5, 7 \end{cases} \\
 P_{eb}(b_1|S_j) &= \begin{cases} P_1 & \text{si } j = 0, 2, 4, 6 \\ P_2 & \text{si } j = 1, 3, 5, 7 \end{cases} \\
 P_{eb}(b_0|S_j) &= P_1 \quad \forall j
 \end{aligned}$$

En tenant compte du fait que $P_2 > P_1$, on peut conclure que les positions binaires b_2 et b_1 sont mieux protégées vis-à-vis du bruit que la position b_0 .

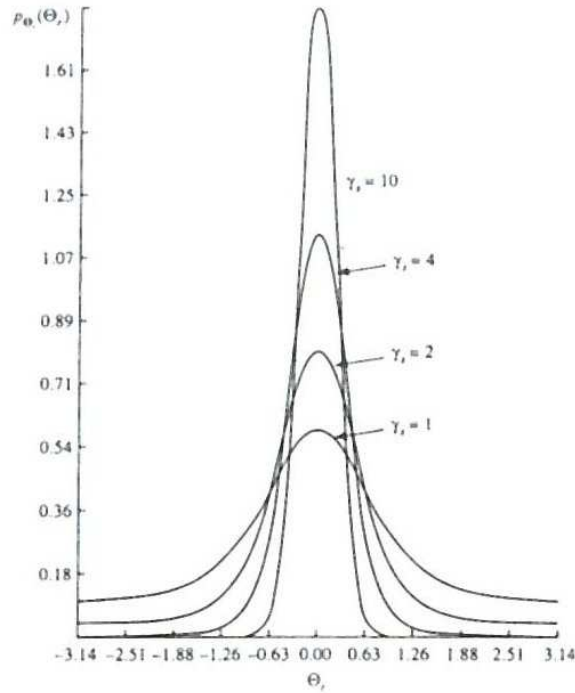


Fig. C.2 — Fonction de densité de probabilité $Pr_{\Theta_r}(\Theta_r)$, pour $\gamma_s = E_s/N_0 = 1, 2, 4, 10$

Nous nous intéressons, d'autre part, à l'écart de phase entre deux symboles différant d'un seul bit, $\Delta\varphi_b$. Nous avons représenté sur la figure C.1 ces écarts de phase pour le symbole S_0 . En observant la modulation de la figure C.1, si l'on prend comme référence n'importe quel symbole, on peut écrire :

- $\Delta\varphi_b = \frac{\pi}{4}$ pour deux bits sur trois (symboles adjacents),
- $\Delta\varphi_b = \frac{3\pi}{4}$ pour un bits sur trois.

Ce sont ces écarts de phase qui déterminent les distances d_0 et d_1 (cf. Figure 4.3) lors de l'analyse présentée dans le Chapitre 4.

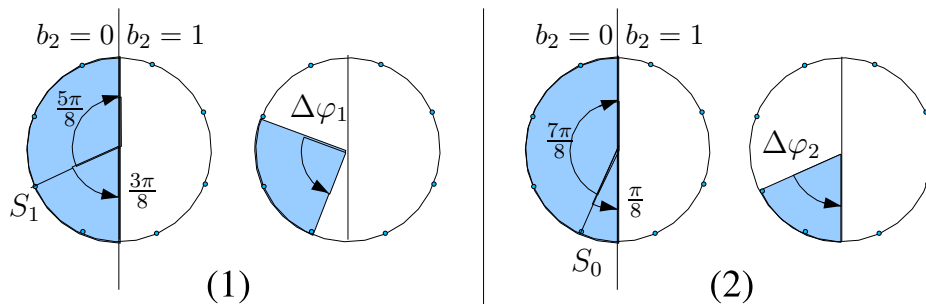


Fig. C.3 — Représentation graphique des approximations décrites en 2.

C.2 Probabilité d'erreur binaire dans une MAQ-16 avec *mapping* de Gray

Considérons la modulation MAQ-16 étudiée dans les Chapitres 3 et 4. Puisque les voies en phase et en quadrature sont codées indépendamment, les résultats obtenus pour une voie sont directement applicables à l'autre voie. Nous considérons la voie en phase représentée sur la figure C.4.

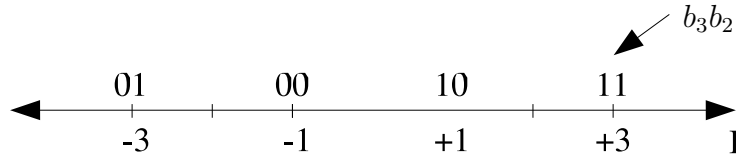


Fig. C.4 — Voie en phase d'une modulation MAQ-16 avec *mapping* de Gray

Le bruit affectant cette voie étant une variable aléatoire gaussienne de moyenne zéro et de variance σ^2 , la probabilité d'erreur binaire dans chaque position ($-3, -1, +1, +3$) est fonction de $Q(x)$, définie comme :

$$Q(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_x^\infty \exp\left\{-\frac{u^2}{2\sigma^2}\right\} du = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \quad (\text{C.3})$$

Soit x_I la coordonnée du symbole émis sur la voie en phase. L'ensemble des probabilités d'erreur à considérer est :

$$\begin{aligned} P_{eb}(b_3|x_I = \pm 3) &= Q\left(\frac{3}{\sigma}\right) \\ P_{eb}(b_3|x_I = \pm 1) &= Q\left(\frac{1}{\sigma}\right) \\ P_{eb}(b_2|x_I = \pm 3) &= Q\left(\frac{1}{\sigma}\right) - Q\left(\frac{5}{\sigma}\right) \\ P_{eb}(b_2|x_I = \pm 1) &= Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{3}{\sigma}\right) \end{aligned}$$

En remarquant que $Q\left(\frac{1}{\sigma}\right) \gg Q\left(\frac{3}{\sigma}\right) \gg Q\left(\frac{5}{\sigma}\right)$, ces probabilités peuvent être approximées comme suit :

$$\begin{aligned} P_{eb}(b_3|x_I = \pm 3) &= Q\left(\frac{3}{\sigma}\right) \\ P_{eb}(b_3|x_I = \pm 1) &= Q\left(\frac{1}{\sigma}\right) \simeq P_{eb}(b_2|x_I = \pm 3) \simeq P_{eb}(b_2|x_I = \pm 1) \end{aligned}$$

On peut alors conclure que dans cette modulation les positions binaires b_3 et b_1 sont mieux protégées que les positions b_2 et b_0 .

On s'intéresse, d'autre part, aux distances entre symboles différant d'un seul bit. Soit $d(b_i|x_I)$ la distance entre deux symboles différant du bit b_i et x_I la coordonnée sur la voie en phase du symbole pris comme référence. L'ensemble de distances à considérer est :

$$\begin{aligned} d(b_i|x_I = -3) &= \begin{cases} 3d_0 & \text{si } i = 3 \\ d_0 & \text{si } i = 2 \end{cases} \\ d(b_i|x_I = -1) &= d_0 \quad i = 2, 3 \\ d(b_i|x_I = +1) &= d_0 \quad i = 2, 3 \\ d(b_i|x_I = +3) &= \begin{cases} 3d_0 & \text{si } i = 3 \\ d_0 & \text{si } i = 2 \end{cases} \end{aligned}$$

En tenant compte du fait que la voie en quadrature est codée de manière identique et indépendamment à la voie en phase, on peut alors conclure que dans 1/4 des cas $3d_0$ est la

distance à considérer, tandis que dans 3/4 des cas la distance à considérer est d_0 . Ce résultat est utilisé en Section 4.3.3.

C.3 Cas de la MAQ-64 avec *mapping* de Gray

Une analyse semblable à celle développée pour la MAQ-16 pourrait être effectuée pour la MAQ-64. Néanmoins, nous ne nous sommes intéressés dans le Chapitre 3 qu'à la protection des différentes positions binaires. Une analyse quantitative des différentes zones de décision pour chacun des bits suffit alors pour déterminer cette classification des bits. Cette analyse est représentée graphiquement sur la figure C.5.

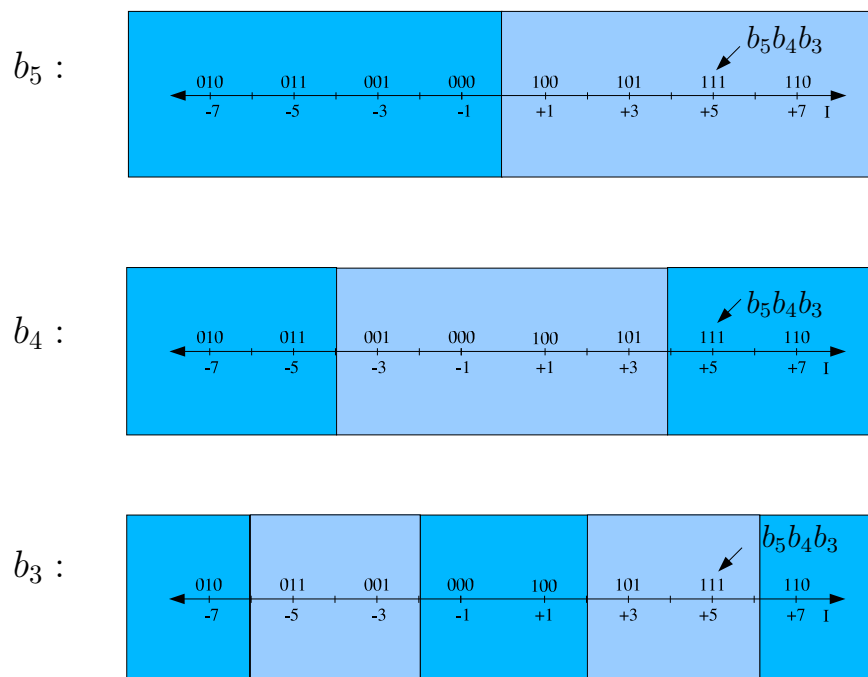


Fig. C.5 — Représentation graphique des régions de décision des bits pour la voie en phase dans une modulation MAQ-64

D'après cette figure et en tenant compte du fait que les régions sont identiques pour la voie en quadrature, on peut classifier les bits comme suit :

- b_5, b_2 : positions au plus haut niveau de protection,
- b_4, b_1 : positions à un niveau moyen de protection,
- b_3, b_0 : positions au plus bas niveau de protection.

Bibliographie

- [1] *TC1000 DVB-RCS turbo decoder data sheet*, dans URL : <http://www.turboconcept.com>.
- [2] *Theoretical performance limit evaluation*, dans URL : <http://www-elec.enst-bretagne.fr/turbo/LIMIT/>, 2002.
- [3] 3RD GENERATION PARTNERSHIP PROJECT, *Technical specification group radio access network; multiplexing and channel coding (FDD); (Release5)*, Rapport technique, 3GPP, Mars 2003.
- [4] J. D. ANDERSEN, *Interleaver design for turbo coding*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 154–156), Brest, France, Sept. 1997.
- [5] L. R. BAHL, J. COCKE, F. JELINEK et J. RAVIV, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inform. Theory, vol. IT-20, pp. 248–287, Mars 1974.
- [6] A. BARBULESCU et S. PIETROBON, *Interleaver design for turbo codes*, IEE Electronics Letters, (pp. 2107–2108), Déc. 1994.
- [7] G. BATAIL, *Pondération des symboles décodés par l'algorithme de Viterbi*, Ann. Télécomm., vol. 42, pp. 31–38, Jan. 1987.
- [8] S. BENEDETTO et E. BIGLIERI, *Principles of digital transmission with wireless applications*, Kluwer Academic/Plenum Publishers, 1999.
- [9] S. BENEDETTO, D. DIVSALAR, G. MONTORSI et F. POLLARA, *Bandwidth efficient parallel concatenated coding schemes*, IEE Electronics Letters, vol. 31, no. 24, Nov. 1995.
- [10] S. BENEDETTO, D. DIVSALAR, G. MONTORSI et F. POLLARA, *Parallel concatenated trellis coded modulation*, dans *Proc. IEEE Int. Conf. Commun., ICC'96*, (pp. 974–978), Juin 1996.
- [11] S. BENEDETTO et G. MONTORSI, *Design of parallel concatenated convolutional codes*, IEEE Trans. Commun., vol. 44, pp. 591–600, Mai 1996.
- [12] S. BENEDETTO et G. MONTORSI, *Unveiling turbo-codes : some results on parallel concatenated convolutional codes*, IEEE Trans. Inform. Theory, vol. 42, no. 2, pp. 409–428, Mars 1996.
- [13] C. BERROU, *Some clinical aspects of turbo codes*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 26–31), Sept. 1997.
- [14] C. BERROU, P. ADDE, E. ANGUI et S. FAUDEIL, *A low complexity soft-output Viterbi decoder architecture*, dans *Proc. IEEE Int. Conf. Commun., ICC'93*, (pp. 737–740), Geneva, Switzerland, Mai 1993.
- [15] C. BERROU, C. DOUILLARD et M. JÉZÉQUEL, *Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes*, Ann. Télécomm., vol. 54, no. 3-4, pp. 166–172, Déc. 1999.

- [16] C. BERROU, C. DOUILLARD, S. KEROUÉDAN, Y. SAOUTER, L. CONDE, M. JÉZÉQUEL, R. CRESPO et J. TOUSCH, *Modem for high-order modulation schemes (MHOMS) final report : Flexible turbo code for low error rates*, Rapport technique, Contract ESA-Alenia Spazio 16593/02/NL/EC, Nov. 2003.
- [17] C. BERROU, R. D. GAUDENZI, C. DOUILLARD, G. GALLINARO, R. GARELLO, D. GIANCRISTOFARO, A. GINESI, M. LUISE, G. MONTORSI, R. NOVELLO et A. VERNUCCI, *High speed modem concepts and demonstrator for adaptative coding and modulation with high order in satellite applications*, dans *Proc. 8th Int. Workshop on Signal Processing for Space Communications , SPSC'03*, Catania, Italy, Sept. 2003.
- [18] C. BERROU et A. GLAVIEUX, *Near optimum error correcting coding and decoding : Turbo-codes*, IEEE Trans. Commun., vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
- [19] C. BERROU, A. GLAVIEUX et P. THITIMAJSHIMA, *Near Shannon limit error-correcting coding and decoding : Turbo-codes*, dans *Proc. IEEE Int. Conf. Commun, ICC'93*, (pp. 1064–1070), Geneva, Switzerland, Mai 1993.
- [20] C. BERROU et M. JÉZÉQUEL, *Frame-oriented convolutional turbo-codes*, IEE Electronics Letters, vol. 32, no. 15, pp. 1362–1364, Juillet 1996.
- [21] C. BERROU et M. JÉZÉQUEL, *Non binary convolutional codes for turbo coding*, IEE Electronics Letters, vol. 35, no. 1, pp. 39–40, Jan. 1999.
- [22] C. BERROU, M. JÉZÉQUEL, C. DOUILLARD et L. CONDE, *Application of the error impulse method in the design of high-order turbo coded modulation*, dans *Proc. Inform. Theory Workshop, ITW'02*, (pp. 41–44), Bangalore, India, Oct. 2002.
- [23] C. BERROU, M. JÉZÉQUEL, C. DOUILLARD et S. KEROUÉDAN, *The advantages of non-binary turbo codes*, dans *Proc. Inform. Theory Workshop, ITW'01*, Cairns, Australia, Sept. 2001.
- [24] C. BERROU et S. VATON, *Computing the minimum distance of linear codes by the error impulse method*, dans *Proc. Int. Symp. on Inform. Theory, ISIT'02*, (p. 5), Lausanne, Switzerland, Juillet 2002.
- [25] C. BERROU, S. VATON, M. JÉZÉQUEL et C. DOUILLARD, *Computing the minimum distance of linear codes by the error impulse method*, dans *Proc. GLOBECOM'02*, Taipei, Taiwan, Nov. 2002.
- [26] C. BETTSTETTER, *Turbo decoding with tail-biting trellises*, Rapport technique, Diplomarbeit, Technischen Universität München, Juillet 1998.
- [27] E. BIGLIERI, D. DIVSALAR, P. J. MCLANE et M. K. SIMON, *Introduction to Trellis-Coded Modulation with Applications*, New York : Macmillan, 1991.
- [28] W. J. BLACKERT et S. G. WILSON, *Turbo trellis coded modulation*, dans *Proc. CISS'96*, Mars 1996.
- [29] G. CAIRE, G. TARICCO et E. BIGLIERI, *Bit-interleaved coded modulation*, IEEE Trans. Inform. Theory, vol. 44, no. 3, pp. 927–946, Mai 1998.
- [30] A. R. CALDERBANK, *Multilevel codes and multistage decoding*, IEEE Trans. Commun., vol. COM-37, pp. 222–229, Mars 1989.
- [31] J. K. CAVERS et P. HO, *Analysis of the error performance of trellis-coded modulations in rayleigh-fading channels*, IEEE Trans. Commun., vol. 40, pp. 74–83, Jan. 1992.
- [32] C. J. CORRADA-BRAVO et I. RUBIO, *Deterministic interleavers for turbo codes with random-like performance and simple implementation*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'03*, (pp. 555–558), Brest, France, Sept. 2003.

- [33] S. CROZIER et P. GUINAND, *High-performance low-memory interleaver banks for turbo-codes*, dans *Proc. IEEE 54th Vehicular Technology Conference*, Atlantic City, NJ, USA, Oct. 2001.
- [34] D. DIVSALAR et F. POLLARA, *Turbo codes for deep-space communications*, Rapport technique, TDA Progress Report 43-120, Fév. 1995.
- [35] D. DIVSALAR et F. POLLARA, *Turbo codes for pcs applications*, dans *Proc. IEEE Int. Conf. Commun., ICC'95*, (pp. 54–59), Seattle, USA, Juin 1995.
- [36] D. DIVSALAR et M. K. SIMON, *The design of trellis coded MPSK for fading channel : Performance criteria*, IEEE Trans. Commun., vol. 36, pp. 1004–1012, Sept. 1988.
- [37] S. DOLINAR, D. DIVSALAR et F. POLLARA, *Code performance as a function of block size*, Rapport technique, TMO progress report 42-133, Jet Propulsion Laboratory, Mai 1998.
- [38] C. DOUILLARD, M. JÉZÉQUEL, C. BERROU, N. BRENGARTH, J. TOUSCH et N. PHAM, *The turbo code standard for DVB-RCS*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'2000*, (pp. 535–538), Brest, France, Sept. 2003.
- [39] T. DUMAN et M. SALEHI, *Performance bounds for turbo-coded modulation systems*, IEEE Trans. Commun., vol. 47, no. 4, pp. 511–521, Avril 1999.
- [40] T. DUMAN et M. SALEHI, *The union bound for turbo-coded modulation systems over fading channels*, IEEE Trans. Commun., vol. 47, no. 10, pp. 1495–1502, Oct. 1999.
- [41] P. ELIAS, *Error-free coding*, I.R.E Trans. Inform. Theory, vol. PGIT-4, pp. 29–37, Sept. 1954.
- [42] P. ELIAS, *Coding for noisy channels*, I.R.E Conv. Rec., vol. 3, pp. 37–46, Sept. 1955.
- [43] G. D. FORNEY, *The Viterbi algorithm*, IEEE Proceedings, vol. 68, pp. 268–278, Mars 1973.
- [44] R. GARELLO, P. PIERLEONI, S. BENEDETTO et G. MONTORSI, *An algorithm to compute the free distance of turbo codes*, dans *Proc. Int. Symp. on Information Theory, ISIT'2000*, (p. 287), Juin 2000.
- [45] A. GLAVIEUX et M. JOINDOT, *Communications Numériques. Introduction*, Masson, 1996.
- [46] J. HAGENAUER et P. HOEHER, *A Viterbi algorithm with soft-decision outputs and its applications*, dans *Proc. IEEE GLOBECOM'89*, (pp. 47.1.1–47.1.7), Dallas, USA, Nov. 1989.
- [47] J. HAGENAUER, E. OFFER et L. PAPKE, *Iterative decoding of binary block and convolutional codes*, IEEE Trans. Inf. Theory, vol. 42, pp. 429–445, Mars 1996.
- [48] R. W. HAMMING, *Error detecting and error correcting codes*, Bell Syst. Tech. J., vol. 29, pp. 147–169, Avril 1950.
- [49] L. HANZO, T. H. LIEW et B. L. YEAP, *Turbo coding, turbo equalisation and space-time coding*, Wiley, 2002.
- [50] C. HEEGARD et S. B. WICKER, *Turbo Coding*, Kluwer Academic Publishers, 1999.
- [51] J. HOKFELT et T. MASENG, *Methodical interleaver desing for turbo codes*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 212–215), Brest, France, Sept. 1997.
- [52] H. IMAI et S. HIRAKAWA, *A new multilevel coding method using error correcting codes*, IEEE Trans. Inform. Theory, vol. ITE-23, pp. 371–377, Mai 1977.

- [53] S. H. JAMALI et T. LE-NGOC, *Coded-modulation techniques for fading channels*, Kluwer Academic Publishers, 1994.
- [54] P. JUNG, *Comparison of turbo-code decoders applied to short frame transmission systems*, IEEE J. Select. Areas Comm., vol. 14, no. 3, pp. 530–537, Avril 1996.
- [55] T. MITTELHOLZER, X. LIN et J. L. MASSEY, *Multilevel turbo coding for m-ary quadrature and amplitude modulation*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 127–34), Brest, France, Sept. 1997.
- [56] G. MONTORSI, S. BENEDETTO, D. DIVSALAR et F. POLLARA, *Serial concatenation of interleaved codes : Performance analysis, design and iterative decoding*, IEEE Trans. Commun., vol. 44, pp. 909–926, Mai 1998.
- [57] H. OGIWARA et F. D. MORRILLO, *Application of turbo codes to trellis-coded modulation*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 200–203), Brest, France, Sept. 1997.
- [58] L. PEREZ, J. SEGHERS et D. COSTELLO, *A distance spectrum interpretation of turbo codes*, IEEE Trans. Inform. Theory, vol. 42, pp. 1698–1709, Nov. 1996.
- [59] J. G. PROAKIS, *Digital Communications*, McGraw Hill International Editions, 1995.
- [60] R. PYNDIAH, A. GLAVIEUX, A. PICART et S. JACQ, *Near optimum decoding of product codes*, dans *Proc. IEEE GLOBECOM'94*, (pp. 339–343), Nov. 1994.
- [61] P. ROBERTSON, *An overview of bandwidth efficient turbo coding schemes*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'97*, (pp. 103–110), Brest, France, Sept. 1997.
- [62] P. ROBERTSON, E. VILLEBRUN et P. HOEHER, *A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain*, dans *Proc. IEEE Int. Conf. on Commun., ICC'95*, (pp. 1009–1013), Seattle, USA, Juin 1995.
- [63] P. ROBERTSON et T. WOERZ, *Coded modulation scheme employing turbo codes*, IEE Electronics Letters, vol. 31, no. 18, Août 1995.
- [64] P. ROBERTSON et T. WOERZ, *A novel bandwidth efficient coding scheme employing turbo codes*, dans *Proc. IEEE Int. Conf. on Commun., ICC'96*, Juin 1996.
- [65] P. ROBERTSON et T. WOERZ, *Extensions of turbo trellis coded modulation to high bandwidth frequencies*, dans *Proc. IEEE Int. Conf. on Commun., ICC'97*, Juin 1997.
- [66] P. ROBERTSON et T. WOERZ, *Bandwidth-efficient turbo trellis-coded modulation using punctured components codes*, IEEE J. Select. Areas Commun., vol. 16, no. 2, Fév. 1998.
- [67] C. SCHLEGEL et D. J. COSTELLO, *Bandwidth efficient coding for fading channels : Code construction and performance analysis*, IEEE J. Select. Areas Commun., vol. SAC-7, pp. 1356–1368, Déc. 1989.
- [68] C. E. SHANNON, *A mathematical theory of communication*, Bell Syst. Tech. J., vol. 27, Juillet et Oct. 1948.
- [69] M. K. SIMON et J. G. SMITH, *Hexagonal multiple phase and amplitude shift-keyed signal sets*, IEEE Trans., vol. COM-21, pp. 1108–1115, Oct. 1973.
- [70] B. SKLAR, *Digital communications : Fundamentals and applications*, Prentice Hall, 1988.
- [71] B. SKLAR, *A primer on turbo code concepts*, IEEE Communications Magazine., (pp. 94–102), Déc. 1997.
- [72] S. SLIMANE et T. LE-NGOC, *Tight bounds on the error probability of coded modulation schemes in Rayleigh fading channels*, IEEE Trans. on Vehicular Technology, vol. 44, no. 1, pp. 121–130, Fév. 1995.

- [73] J. G. SMITH, *Odd bit symmetric QASK*, IEEE Trans., vol. COM-23, pp. 385–389, Mars 1975.
- [74] L. CONDE CANENCIA et C. DOUILLARD, *Performance estimation of 8-PSK turbocoded modulation over Rayleigh fading channels*, dans *Proc. Int. Symp. on Turbo Codes and Related Topics, ISTC'03*, (pp. 567–570), Brest, France, Sept. 2003.
- [75] L. CONDE CANENCIA et C. DOUILLARD, *A new methodology to estimate asymptotic performance of turbocoded modulation over fading channels*, dans *Proc. 2nd Int. Symp. on Image/Video Communications over fixed and mobile networks, ISIVC'04*, Brest, France, Juillet 2004.
- [76] L. CONDE CANENCIA, C. DOUILLARD, M. JÉZÉQUEL et C. BERROU, *Application of the error impulse method to 16-QAM bit-interleaved turbocoded modulations*, IEE Electronics Letters, vol. 39, no. 6, pp. 538–539, Mars 2003.
- [77] CONSULTATIVE COMITEE FOR SPACE DATA SYSTEMS, *Telemetry channel coding*, Rapport technique, Blue Book 1001.0-B-4, 1999.
- [78] ETSI EN 301 790, *Interaction channel for satellite distribution system*, dans V1.2.2, (pp. 21–24), Déc. 2000.
- [79] ETSI EN 301 958, *Interaction channel for digital terrestrial television*, dans V1.1.1, (pp. 28–30), Août 2001.
- [80] S. LE GOFF, *Les Turbo-codes et leur application aux transmissions à forte efficacité spectrale*, Thèse de doctorat, Université de Bretagne Occidentale, 1995.
- [81] S. LE GOFF, *Performance of bit-interleaved turbo-coded modulations on rayleigh fading channels*, IEE Electronics Letters, vol. 36, no. 8, pp. 731–733, 13th Avril 2000.
- [82] S. LE GOFF, A. GLAVIEUX et C. BERROU, *Turbo-codes and high spectral efficiency modulation*, dans *Proceedings of Int. Conf. Commun, ICC'94*, (pp. 645–649), New Orleans, USA, Mai 1994.
- [83] P. THITIMAJSHIMA, *Les codes convolutifs récurrents systématiques et leur application à la concaténation parallèle*, Thèse de doctorat, Université de Bretagne Occidentale, 1993.
- [84] C. M. THOMAS, *Digital amplitude phase keying with m-ary alphabets*, IEEE Trans., vol. COM-22, pp. 168–179, Fév. 1974.
- [85] G. UNGERBOECK, *Channel coding with multilevel/phase signals*, IEEE Trans. Inform. Theory, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.
- [86] G. UNGERBOECK, *Trellis modulation with redundant signal sets. Part II : State of the art*, IEEE Communications Magazine, vol. 25, no. 2, pp. 12–21, Fév. 1987.
- [87] A. J. VITERBI, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inform. Theory, vol. IT-13, pp. 260–269, Avril 1967.
- [88] A. J. VITERBI, *An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes*, IEEE J. Select. Areas Commun., vol. 16, no. 2, pp. 260–264, Fév. 1998.
- [89] B. VUCETIC et J. YUAN, *Turbo Codes. Principles and applications*, Kluwer Academic Publishers, 2000.
- [90] U. WACHSMANN et J. HUBER, *Power and bandwidth efficient digital communication using turbo codes in multilevel codes*, European Trans. on Telecommun., vol. 6, pp. 557–67, Sept.-Oct. 1995.
- [91] T. WOERZ et J. HAGENAUER, *Multistage coding and decoding*, dans *Proc. IEEE GLOBECOM'90*, (pp. 501.1.1–501.1.6), San Diego, USA, Déc. 1990.

- [92] J. M. WOZENCRAFT et I. M. JACOBS, *Principles of Communication Engineering*, New York : John Wiley and Sons, 1965.
- [93] E. ZEHAVI, *8-PSK trellis codes on Rayleigh channel*, dans *IEEE Military Communications Conference, MILCOM'89*, vol. 2, (pp. 536–540), Oct. 1989.
- [94] E. ZEHAVI, *8-PSK trellis codes for a Rayleigh channel*, *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, Mai 1992.