Methodes pour l'Estimation de canal, Egalisation et Codage pour le Traitement Iteratif en présence d'interférences

Alaa Ghaith

To cite this version:

Alaa Ghaith. Methodes pour l'Estimation de canal, Egalisation et Codage pour le Traitement Iteratif en présence d'interférences. Electronique. Télécom ParisTech, 2006. Français. NNT : . pastel-00003232

HAL Id: pastel-00003232
https://pastel.hal.science/pastel-00003232

Submitted on 16 Nov 2010

TELECOM PARIS
école nationale supérieure des télécommunications

École Doctorale d'Informatique, Télécommunications et Électronique de Paris

Thèse

présentée pour obtenir le grade de Docteur
de l'École Nationale Supérieur des Télécommunications

Spécialité: **Electronique et Communications**

Alaa GHAITH

Méthodes pour l'Estimation de Canal, Égalisation et Codage pour le Traitement Itératif

Soutenue le 08 juin 2006 devant le jury composé de

Didier Erasme	Président
Dirk Slock	Rapporteurs
David Declercq	
Ramesh Pyndiah	Examinateur
Joseph Boutros	Directeurs de thèse
Yi Yuan-Wu	

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué de prés ou de loin au bon déroulement de cette thèse, m'ont soutenu, encouragé et aidé au cours de ces années de thèse.

Je souhaite tout d'abord remercier le professeur Didier Erasme de l'ENST Paris, qui m'a honoré en acceptant d'être président du jury. Tous mes remerciements reviennent aux professeurs Dirk Slock de L'Eurécom et David Declercq de l'université Cergy Pontoise, d'avoir accepté d'être rapporteurs, pour le travail de lecture minutieux et la qualité de leurs remarques concernant ce manuscript. Toute ma reconnaissance également au professeur Ramesh Pyndiah de l'ENST Bretagne, pour sa présence dans la composition du jury.

Je remercie France Télécom R&D située à Issy les Moulineaux qui a contribué au financement de ma thèse. Je remercie tous mes anciens collègues de France Télécom et particulièrement Daniel, Anne-gaele, Pierre, Thierry, Jean-Marie, Eric, Gérard, Stéfane, Ahmed, Julien, Sami, Mireille et Amira pour toute la sympathie qu'ils ont offerte pendant mes trois années à France Télécom.

Je remercie tous les collègues et amis du département COMELEC de l'ENST: Daniele, Chantal, Marie, Robert, Jean-Claude, Ghaya. Bien évidemment, je remercie toute l'équipe des thèsards et stagiaires de différents horizons qui ont réussi à me supporter pendant ma thèse et ont partagé leur richesse culturelle. Un grand merci à: Fatma, Ghassan, Sheng, Sylvain, Emilio, Hadi, Sebastien et tous les autres qui me pardonneront de les avoir oubliés. Une pensée toute particulière s'adresse vers mon ami Nicolas.

J'adresse toute ma gratitude à Yi Yuan-Wu, co-responsable industriel, pour sa patience, sa rigueur, son aide précieuse et sa sympathie. Ces années de collaboration avec Yi resteront pour moi des souvenirs inoubliables. Sans elle cette thèse n'aurait jamais été ce qu'elle est aujourd'hui.

Bien sûr, toute ma reconnaissance s'adresse à Joseph Boutros, que je ne peux pas remercier assez pour l'investissement en temps et en confiance dont il a fait preuve dès la première année de ma thèse, l'autonomie qu'il m'a permis d'acquérir très rapidement, pour terminer cette thèse avec le sentiment de devenir expert (du moins sur ce sujet). Intelligence, générosité, passion, pédagogie, expérience et chaleur humaine sont des qualificatifs qui correspond parfaitement à Joseph et qui font de lui un directeur de thèse exceptionnel. Je vous souhaite, Joseph et Yi, beaucoup de bonheur avec votre famille et vos futurs doctorants.

Il est venue le temps de remercier mes amis, mes frères et sœurs et belle-famille. Ce sont eux qui ont souffert le plus de cette thèse et de mes indisponibilités passagères et ont contribué à l'équilibre et au soutien familial qui m'est vital.

Enfin, je dédie cette thèse aux personnes qui m'ont soutenu quotidiennement, partagé les moments de doute mais aussi de joie pendant ces années, et qui mériteraient autant de moi cette distinction prestigieuse de docteur,

À mes parents, à Amal et à Jana.

Acknowledgements

I would like to thank all the persons who had contributed to facilitate the smooth running of this thesis, had supported me, encouraged, and helped in the course of these years.

First, I would like to thank the professor Didier Erasme from ENST Paris, who had accepted to be the president of the jury. All my thanks are also directed to professor Dirk Slock from Eurécom and David Declercq form Cergy Pontoise University, who had accepted to be reporters and for their meticulous reading work and the quality of theirs remarks. All my recognition equally to professor Ramesh Pyndiah from ENST Bretagne, for his presence in the composition of the jury.

I thank France Telecom R&D situed in Issy les Moulineaux who had contributed in the financing of my thesis. I thank all my old colleagues from France Telecom and particularly: Daniel, Anne-gaele, Pierre, Thierry, Jean-Marie, Eric, Gérard, Stéfane, Ahmed, Julien, Sami, Mireille et Amira for all the sympathy that they give during these three years in France Telecom.

I thank all my colleagues and friends in the COMELEC department: Daniele, Chantal, Marie, Robert, Jean-Claude, Ghaya. Of course, I thank all the team of PhD students and trainees who had succeed to support me during the thesis and shared their cultural wealth. Big thanks to: Fatma, Ghassan, Sheng, Sylvain, Emilio, Hadi, Sebastien and all the others who will forgive me to forget them. A particular thought applies to my friend Nicolas.

I would like to apply my gratitude to Yi Yuan-Wu, my industrial co-director, for her patience, her rigor, her precious help and her sympathy. These years of collaboration with Yi will remain for me an unforgettable memory. Without her this thesis will not ever be what it is today.

Of course, all my recognition is applied to Joseph Boutros, which I can not thank enough for his investment in time and in confidence from the first year of my thesis, the independence which he had allowed to me very fast, to finish this thesis with the feeling to be an expert (at least in this subject). Intelligence, generosity, passion, education, experience and human warmth are terms which perfectly correspond to Joseph and which do him an exceptional thesis supervisor. I wish for you, Joseph and Yi, a lot of happiness with your family and your future PhD students.

I also want to thank my friends, my brothers and sisters and my family in-laws. They are the ones who suffered the most from this thesis and have contributed to the balance and the family support which is vital to me.

Finally and most important of all, I would like to dedicate this thesis to the persons who had pushed me to fulfill my ambitions in life and provided me with the healthy environment to achieve success in education. My deepest gratitude goes to the sacrifices they went through to see me become doctor.

To My parents, Amal, and Jana.

Résumé de la thèse

La turbo égalisation est un processus de réception itératif entre l'égalisation et le décodage du canal; les informations souples générées par chaque fonction de réception est améliorée au fur des itérations jusqu'à aboutir le comportement optimal du système.

Dans cette thèse, on propose une nouvelle procédure itérative d'estimation de canal et de détection du symbole dans un schéma de turbo égalisation pour des canaux multi-trajets à évanouissement et des systèmes de transmission codés utilisant des modulations à grande taille M-QAM *($M=2^m$)*. Cette procédure utilise la séquence d'apprentissage et les données décodées, et elle sera appelée la technique de la Séparation de l'Observation (OS). L'estimation de canal utilise la méthode pseudo-inverse. Pour chaque symbole, la détection prend en compte seulement une partie de l'observation qui contient ce symbole et sa longueur correspond à la longueur de la mémoire du canal. Cette partie d'observation est retirée de l'observation totale pour estimer le canal utilisé pour la détection de ce symbole. Ainsi, on a le même nombre d'estimation de canal que de symboles dans un bloque, ce qui rend très complexe cette technique. Cette complexité peut être réduite en utilisant une méthode modifiée de l'estimation pseudo-inverse. Le canal considéré est multi-trajets variable ou non avec le temps. Les techniques de détection utilisées sont l'égalisation linéaire en bloc de forçage à zéro (ZF-BLE) pour l'itération initiale et la méthode de suppression d'interférences (IC) pour les autres itérations. Pour la boucle de retour, les décisions dures ou souples de la sortie de décodeur du canal sont employées.

Ensuite, on étend la technique de la séparation d'observation (OS) à la technique de détection itérative des étiquetages. On considère un système de modulation à bits codés et entrelacés avec un décodage itérative (BICM-ID) sur un canal multi-trajets pour des transmissions codées de 16-QAM. On propose d'étudier l'influence de l'étiquetage sur les performances de la turbo égalisation. Et on analyse l'amélioration apportée par l'utilisation de l'étiquetage optimisé à la turbo égalisation et on va proposer un étiquetage binaire optimisé de la constellation 16-QAM dans l'ordre d'augmenter le gain de codage, qui peut être utilisée pour plusieurs cas pratiques et importants. Il est démontré que un choix prudent de la conception de l'étiquetage différent de l'étiquetage classique de Gray, les performances en probabilité d'erreur par bit sont améliorées.

Pour le codage correcteur d'erreur, on étudie l'algorithme de décodage de la maximisation a posteriori (MAP) pour les codes non binaires sur une extension du corps de Galois *GF(q)*. Cette règle de décodage minimise la probabilité d'erreur par symbole sur des canaux discrets sans mémoire en employant le code dual. Il est montré que cet algorithme possède tous les exigences nécessaires pour le décodage itératif ainsi la sortie du décodeur peut être divisée en trois estimations indépendantes: la valeur souple du canal, le terme a priori et une valeur extrinsèque.

Il représente une forme meilleure de décodage pour les codes LDPC non binaires, qui sont meilleure que les codes LDPC binaires et les codes Reed-Solomon sur le canal gaussien AWGN. Cet algorithme de décodage nous donne un nouveau algorithme de décodage plus rapide et à complexité réduite. L'inconvénient est la complexité qui, dans cette règle de décodage varie inversement proportionnel au rendement du code, faisant la technique particulièrement attractive pour les codes à rendement élevé. On voit que la complexité est réduite en utilisant le code dual et l'algorithme est accéléré en utilisant la transformée d'Hadamard rapide (FHT). On applique cette règle de décodage aux codes LDPC non binaires.

Ensuite, on va proposer des nouveaux codes LDPC adaptatives où on propose d'adapter la taille de la constellation dans le codeur directement avant la construction de la matrice de parité, cela nécessite une connaissance du profile du canal. L'adaptation est conçue d'une manière simple en se basant sur l'évanouissement de chaque sous-canal en supposant que le canal est divisé en plusieurs sous-canaux (sous-porteuses OFDM) et en calculant la capacité de chaque sous-canal et la comparant au rendement du code. On va démontrer que ce code LDPC adaptative peut être facilement appliqué dans un système OFDM et décodé par la même règle conçue auparavant.

Finalement, on va essayer d'améliorer les performances de la turbo égalisation (TE), considérant à l'émetteur une modulation non binaire codée avec le code LDPC non binaire comme code correcteur d'erreur. Le récepteur itératif correspondant combine l'égaliseur à suppression de l'interférence (IC), l'estimation avec la technique de séparation d'observation (OS) et le décodage LDPC non binaire souple. On se focalise sur le développement des codes LDPC pour les canaux avec des entrées non binaires et à mémoire ISI. On est concerné par la conception des codes LDPC turbo égalisés qui donnent la probabilité d'erreur par bit la plus faible possible pour un apport de complexité très faible. On montre que un bon choix de la degré de distribution des codes LDPC non binaires nous conduit à améliorer les performances par rapport aux turbo codes, malgré que celles des codes LDPC binaires avec la même degré de distribution sont plus mauvaise que celles des turbo codes.

Thesis Abstract

Turbo equalization is a receiving process performing iteratively equalization and channel decoding; soft information generated by each receiving function is improved through the iterations until the optimum behavior of the system is reached.

In this thesis, we propose a new iterative channel estimation and symbol detection procedure in a turbo equalization scheme for a multipath fading channel and coded transmission system with a high level modulation M-QAM *($M=2^m$)* by using the training sequence and the decoded data, which is named Observation Separation (OS). The channel estimation corresponds to the pseudo-inverse type. For every symbol, the detection takes into account just the part of the observation, which contains the considered symbol and has a length corresponding to the channel memory length. The previous part of the observation is excluded from the total observation for make the channel estimation of this symbol. Thus there are as many symbol detections as channel estimates. The channel considered is a time variant or invariant multipath. The detection techniques used are the Zero Forcing Block Linear Equalizer (ZF-BLE) for the initial iteration and Interference Cancellation (IC) for the other iterations. We can reduce the complexity by using the modified pseudo-inverse channel estimation. For the feedback loop, the soft or hard decision of the channel decoder output is employed.

Next, we extend the iterative demapping technique to Observation Separation (OS). We consider a bit interleaved coded modulation with iterative decoding (BICM-ID) on a multipath channel for coded transmission system with 16QAM. We propose to study the influence of mapping over the performance of turbo equalization. And we analyze the improvement of mapping optimization for turbo equalization and we proposed an optimized binary mapping of 16-QAM constellation in order to increase coding gain, which can be used for some practically important cases. It is demonstrated that with a carefully designed mapping, different to the classical Gray mapping, bit error rate performance is improved.

For the error correcting codes, we study symbol-by-symbol maximum a posteriori (MAP) decoding algorithms for non binary codes over an extension field *GF(q)*. This decoding rule minimizes the probability of symbol error over a time-discrete memory less channel by employing the dual code. It is shown that these algorithms meet all requirements needed for iterative decoding as the output of the decoder can be split into three independent estimates: soft channel value, a priori term and extrinsic value. It represents a better form of coding for the *q*-ary LDPC codes, which have been shown to outperform binary LDPC codes and Reed–Solomon codes on the AWGN channel and it gives us a new fast and reduced-complexity decoding algorithm. The drawback is the complexity which, in this rule varies inversely with code rate, making the technique particularly attractive for high rate codes. We see that complexity is reduced by using the dual code, and the algorithm is accelerated by using the Fast Hadamard Transform (FHT). We apply this design rule to decoding the non binary LDPC.

Then, we present a new Adaptive LDPC code where we propose to adapt the constellation size directly in the decoder, before the construction of the parity matrix and this necessities the knowledge of the channel profile. The adaptation is design simply based on the fading of each sub-channel by supposing that the channel is divided to several OFDM sub-channels and by computing the capacity of each sub-channel and comparing to the code rate. We will demonstrate that this adaptive LDPC code can be simply applied to the OFDM system and decoded by the same design rule presented beforehand.

Finally, we try to assess the performance of Turbo Equalization (TE), considering at the transmitter a non binary coded modulation with Low Density Parity Check (LDPC) code as error correcting code. The corresponding turbo-receiver combines Interference Cancellation (IC) equalizer, Observation Separation (OS) estimator and a soft non binary LDPC decoder. We focus on developing LDPC codes for channels with non binary inputs and ISI memory. We are concerned with finding LDPC-turbo equalization which produces the lowest possible bit-error rate for a minimum amount of complexity. We show that a good choice of distribution degree of non binary LDPC codes leads us to outperform the performances of turbo codes; this despite the performance of the binary LDPC codes with the same distribution degree is not so good.

Contents

List of Figures

List of Tables

List of Notations

Mathematical notations

$\langle \bullet \rangle_{A,K}$	Arithmetic Means over K
$\bullet^{\dagger}$	Conjugate Transpose operand
$\lVert \bullet \rVert$	Euclidean norm
$\lVert \bullet \rVert_{F}$	Frobenius norm
$\langle \bullet \rangle_{G,K}$	Geometric Means over K
$\lvert \bullet \rvert$	Module
$\bullet^{T}$	Transpose operand
$\mathbb{C}$	Set of Complex
$\mathbb{Z}$	Set of Integer
$\mathbb{R}$	Set of Real
$\bullet^{(l)}$	lth Iteration

Variables notations

C_{AWGN}	Capacity of the AWGN Channel
C_{int}	Capacity of Interference Channel
I_{int}	Mutual Information of Interference Channel
H	Channel Matrix
h_{ij}	Coefficient of Channel Matrix
X	Transmitted Symbol vector

x_i	Transmitted Symbol
R	Received Symbol vector
r_i	Received Symbol
W	AWG Noise vector
$w_i\,/\,\eta_{\mathrm{t}}$	AWG Noise
c_i	Coded bit/symbol
u_i	Information bit/symbol
b_i	Coded and Interleaved Bit
s_i	Coded and Interleaved Symbol
N	Block Length of Transmitted Coded Symbol vector
K	Block Length of Information Symbol vector
N_0	Noise Variance
$\tilde{X}$	Equalized Symbol vector
$\hat{X}$	Estimated Symbol vector
ε_s	Average Symbol Energy
ζ	Error Correcting Code
Π/Π^{-1}	Interleaver and De-Interleaver
$\mathrm{H}_{\mathrm{LDPC}}$	Low Density Check Matrix
R_c	Code Rate

List of Acronyms

AMC	Adaptive Modulation and Coding
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BER	Bit Error Rate
BDFE	Block Decision Feedback Equalizer
BICM	Bit-Interleaved Coded Modulation
BLAST	Bell Labs Layered Space-Time
BLCM	Bit LDPC Coded Modulation
BLE	Block Linear Equalizer
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
BTCIM	Bit Turbo Coded Interleaved Modulation
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CSI	Channel State Information
DAPP	Dual A Posteriori Probability
DDFSE	Delayed Decision Feedback Sequence Estimator
DE	Density Evolution
DFE	Decision Feedback Equalizer
DMT	Discrete Multi-Tone
DS	Direct Sequence
ECC	Error Correction Code
EFD	Effective Free Distance
EM	Expectation Maximization
FEC	Forward Error Correction
FER	Frame Error Rate
FHT	Fast Hadamard Transform

GF	Galois Field
IC	Interference Canceller
ID	Iterative Decoding
ISI	Inter-Symbol Interference
LDPC	Low Density Parity Check
LE	Linear Equalizer
LMS	Least Mean Square
MAI	Multiple Access Interference
MAP	Maximum A Posteriori Probability
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
OMI	Optimized Mutual Information
OS	Observation Separation
PCCC	Parallel Concatenated Codes
PDFD	Parallel Decision Feedback Decoding
PIC	Parallel Interference Cancellation
PSP	Per-Survivor Processing
QAM	Quadrature Amplitude Modulation
QLDPC	*q*-ary LDPC (non binary LDPC)
QPSK	Quadratic Phase Shift Keying
RLS	Recursive Least Square
SCCC	Serially Concatenated Codes
SG	Stochastic Gradient
SISO	Soft-Input Soft-Output
SLCM	Symbol LDPC Coded Modulation
SNR	Signal-to-Noise Ratios
SP	Set Partitioning
SPC	Single Parity Check
TDD	Time-Division Duplex
VA	Viterbi Algorithm
WMF	Whitening Matched Filter
ZF	Zero Forcing

Résumé de la thèse en Français

Chapitre 1: Introduction

Dans les systèmes de télécommunication numériques, l'interférence limite la capacité du récepteur à détecter les symboles discrets envoyés par le transmetteur. Le canal est la partie des systèmes de communication qu'on peut pas le change, pour cela les techniques de détection qui prennent en considération l'interférence sont très importantes.

L'utilisation du principe "Turbo" approche de prés les performances théoriques de Shannon concernant la capacité du canal.

Dans ce chapitre d'introduction, je vais présenter quelques notions de bases et l'organisation de la thèse dans la section I.2.

1.1 Notions

Un des problèmes les plus importants dans la réalisation des systèmes de communication sur un canal sélective en fréquence est la présence de Interférences Entre Symboles (IES). Pour protéger l'intégrité des données à transmettre, on utilise les codes correcteur d'erreurs (CCE), où on ajoute quelques bits de redondance (codage).

Dans les dernières années, le nombre des utilisateurs de systèmes de communication sans fil à augmenter de façon exponentielle. Pour cette raison, augmenter la capacité du système devient de plus en plus un issue cruciale, spécialement pour la prochaine génération des systèmes cellulaires qui ont une capacité limitée par les interférences. La réduction des ces interférences améliore la performance des systèmes. Une approche promettant pour supprimer l'interférence est de travailler sur le traitement de signal.

Les systèmes mobiles ajoute une dimension de plus au problème; les chemins de transmission du signal varient avec le temps. Quand la transmission utilise une séquence de symboles digitales, un canal de transmission à chemin multiple est aussi appelé canal IES. Les symboles arrivent à la récepteur simultanément par chemins différents vont être superposer comme l'image d'un télévision lorsqu'on reçoit deux images en même temps. Le but du concepteur de la récepteur est de combattre le IES, la propagation par des chemins multiples et la mobilité pour que l'intégrité de la message transmit soit préservée.

Il est très important de prendre note que les algorithmes turbos ne sont optimaux. Pourtant l'évidence suggère que ces algorithmes sous optimaux utilisés dans le principe turbo performent proche de l'optimum.

1.2 Aperçu de la thèse

Dans cette thèse, on développe un schéma pratique pour annuler l'effet de l'interférence dans le canal pour que la communication numérique devient plus sûre. On focalise sur le scénario dans lequel l' émetteur ne connaît pas le canal. Dans le chapitre 2, on décrit le model bande de base pour les systèmes de communication numériques et on définie le problème de détection. On présente un petit perspective où on explique comment la détection peut être combinée au codage et décodage du canal dans des scénarios différents.

On présente aussi dans le chapitre 3 quelques schémas classiques de détection qui ont été présentées dans la littérature et quelques livres mais ici elles vont être groupées dans un même chapitre pour faciliter l'introduction de la turbo égalisation faite dans le chapitre 4, qui commence par une explication de la théorie de turbo codage. Après on montre comment la schéma turbo peut être modifiée pour développer le turbo égaliseur.

Le concept nommé la technique de Séparation d'Observation (SO) qui sépare partiellement les observations utilisées pour l'estimation du canal de celles utilisées pour la détection de symboles, est présenter dans le chapitre 5. On applique cette technique SO au système UMTS-TDD et aussi à un système bande étroite avec une modulation binaire BPSK et des modulations haut niveau M-QAM sur un canal IES invariable avec le temps.

Après, on va essayer d'améliorer les performances de notre système en suggérant dans le chapitre 6, la conception d'une constellation optimale dans une schéma BICM-ID avec SO estimation du canal et IC égaliseur.

Dans le chapitre 7, on introduit les codes LDPC et quelques unes de ces propriétés, algorithmes de construction et de décodage. On généralise l'idée originale de décodage "dual APP" (DAPP) à des alphabets non binaires dans le chapitre 8. Par alphabet non binaire on considère que les symboles du code sont pris d'une certaine extension du corps GF($q=2^m$). Dans ce chapitre, on utilise les codes non binaires avec des modulations non binaires (haut niveau). On présente une nouvelle règle de décodage qui est, dans un sens, le dual du décodage Viterbi pour les codes linéaires. Et on explique comment la complexité sera réduite avec notre règle de décodage et on réussit à rendre l'algorithme plus rapide par l'utilisation de transformées d'Hadamard rapides (FHT) des probabilités. Finalement dans ce chapitre, on présente les codes LDPC adaptives adressés aux systèmes sans fil de 3$^{\text{ième}}$ génération et plus comme l'OFDM. L'idée principale de l'utilisation de codage adaptive est de changer le corps de Galois dans des codes LDPC non binaires en adaptant un corps de Galois différent sur chaque fréquence, l'objective est de bénéficier de la totalité du spectre et de prendre avantage de toute la capacité du canal par l'utilisation de la technique de Water-filling en considérant que les caractéristiques du canal sont parfaitement connues et l'émetteur utilise ces estimations pour choisir la constellation appropriée.

Dans le chapitre 9, on focalise sur le développement des codes LDPC pour les canaux avec entrées binaires et mémoire IES. On est concerné par découverte d'un schéma LDPC-Turbo égalisation qui produit le plus petit rapport de bits erronés pour une complexité minimale.

Finalement dans le chapitre 10, on résume les contributions de cette thèse et on discute des directions des futures recherches.

Chapitre 2: Les limites des canaux de communication

Au début de 1940, le consensus général était que lorsqu'on augmente le débit d'information transmises sur un canal de communication, on augmente la probabilité d'erreur. Cependant, dans son papier de 1948 "A M Mathematical Theory of Communication," Shannon a quantifié le débit maximum d'information qui peut être envoyé sans erreur sur un canal de communication [45]. La preuve de Shannon implique qu'on peut avoir un code aléatoire avec une probabilité d'erreur exponentiellement petite pour des longues blocs, associé à une règle de décodage. Malheureusement, ces genres de codes sont extrêmement difficile de décoder en pratique, généralement nécessitant des tableaux et/ou des calculs exponentiellement larges.

Depuis ce temps, les chercheurs ont trouvé des codes structurés qui ont des apports très proche de la limite théorique et qui peuvent être décodé simplement. Il y a plusieurs limites de l'information et dans ce chapitre, on va rappeler le maximum possible de rapports d'information pour différentes scénarios et comment les chercheurs ont essayé d'approcher ces rapports avec des modulations pratiques. En particulier, on s'intéresse aus canaux sans interférences, canaux à interférence et l'émetteur les connaît, et canaux à interférence sans la connaissance de l'émetteur.

Dans ce chapitre, on décrit un model de communication numérique en bande de base et on définit le problème de détection associé à ce model. On présente un perspective où on explique comment la détection est combinée avec le codage et le décodage du canal dans des différentes scénarios de communication pour aboutir à des rapports d'informations transmis proche du maximum théorie.

On présente dans ce chapitre la définition du:

2.1 Canal AWGN et le calcule de la capacité.

2.2 Model du canal à interférences:

2.2.1 Calcule de la capacité avec water filling.

2.2.2 Calcule de la capacité sans water filling.

2.2.3 Comparaison.

Chapitre 3: Schémas de détection classiques

On peut trouver une variété de détecteurs qui peuvent être utilisés dans notre schéma, dans ce chapitre on va discuter quelques un. On commence avec le détecteur à maximum de vraisemblance (MV) qui minimise la probabilité d'erreur et considéré comme optimal. Les autres détecteurs présentés dans ce chapitre sont conçus pour offrir une solution proche de celle de la détection MV avec une complexité plus faible.

Avant de revoir ces détecteurs classiques, il est utile d'établir une méthode avec laquelle on peut comparer les performances des détecteurs. Comme les détecteurs traitent typiquement les données codées et non codées, en focalisant sur les systèmes non codés nous permet d'isoler la qualité du détecteur de la qualité du décodeur.

Dans ce chapitre, on présente et rappelle quelques schémas classiques de détection qui ont été présentées dans la littérature et quelques livres mais ici on les groupe dans un même chapitre pour faciliter l'introduction de la turbo égalisation faite dans le chapitre suivant.

3.1 Détection MV

Le détecteur calcule le vecteur le plus probable basé sur la connaissance du signal reçu R, du canal H et de la distribution du noise W.

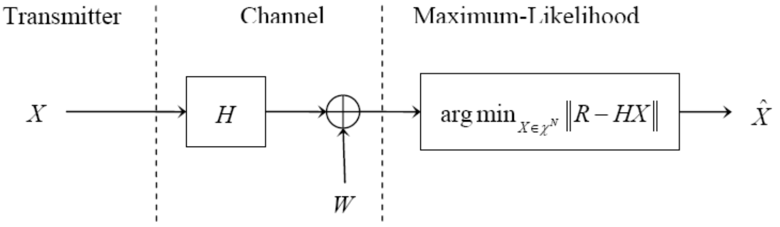


Figure 3.1: Maximum-Likelihood (ML) detection.

3.2 Détection Linéaire

On prend le vecteur reçu R et on le multiplie par une matrice B. On prend la décision sur ce résultat du produit. La matrice B peut être optimisée en utilisant plusieurs critères, mais les deux les plus populaires sont le Zero-Forcing (ZF) et le MMSE.

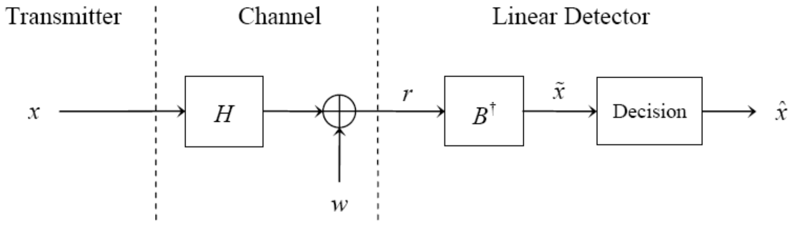


Figure 3.4: Linear Detection.

3.2.1 Whitening Matched Filter

L'estimateur de données conventionnel est un filtre bien défini et assorti qui peut être étendu par un filtre de blanchissement.

Il traite l'IES comme du noise, il est introduit ici car tous les techniques présentées après peuvent être interprété comme un extension du WMF, où le noise est blanchi, et le signal obtenu est envoyé à un filtre.

3.2.2 L'égaliseur linéaire Zero-Forcing

Le ZF-BLE qui minimise la forme quadratique donne une valeur estimée et continue. L'estimation est faite au terme zero forcing car il élimine totalement l'ISI sans respecter le niveau du bruit.

3.2.3 L'égaliseur linéaire Minimum-Mean-Square Error

Le MMSE minimise l'erreur quadratique et donne une valeur estimée, il réduit la dégradation résultante de ZF car la décision prend en compte la corrélation du bruit qui existe dans les variables à estimer.

3.3 Decision-Feedback Detection

Le DFE ajoute au détecteur linéaire une boucle de retour non linéaire, le signal reçu R est multiplié par la matrice B et la décision sera prise sur le résultat après soustraction de la valeur venant de la boucle de retour (c-à-d du résultat précédent)

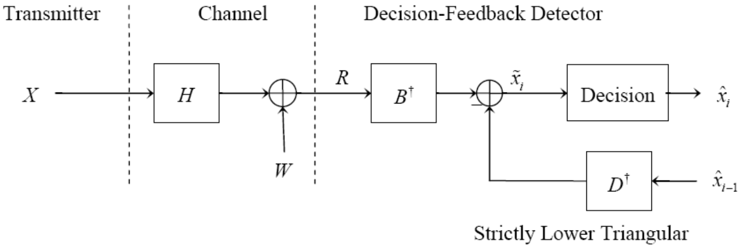


Figure 3.8: Decision Feedback detection.

3.4 Détection Multi-étage

Dans le but de symétriser le problème de pouvoir annuler seulement les symboles futures dans le DFE, le détecteur multi-étage traite le vecteur reçu en bloc d'itérations. Il peut être considéré comme un processeur parallèle, où le DFE est un processeur séquentiel. Et voici quelques exemples:

3.4.1 Annulatif d'Interférence

3.4.2 L'erreur quadratique moyen

3.4.3 Annulatif d'Interférence Adaptatif

Chapitre 4: Turbo Egalisation – Cours

L' introduction de la concaténation des codes correcteurs d'erreur et la théorie de décodage (Turbo Code) dans le champ de communications a permis de travailler avec des débit proche de la limite de Shannon. Ce chapitre commence avec une explication de la théorie du turbo code. Ensuite, on montre comment le schéma turbo peut être modifié pour développer la "turbo égalisation" pour égaliser la sortie du canal IES. Puis les modifications nécessaires pour la turbo égalisation pour l'adapter au canal IES variable avec le temps sont décrites et enfin on discute du problème de la complexité du récepteur.

4.1 Turbo Code

Il entraîne la concaténation de deux convolutional codes séparés par un entrelaceur. Le principe "turbo" parle du décodage itératif du code complexe généré. Chacun des deux codes est décodé séparément, mais avec l'avantage d'avoir quelques informations appris de l'autre code pendant l'itération précédente. Chaque code est décodé au récepteur par un décodeur SISO.

Cette présentation de la théorie turbo code développe une notion qui représente une expansion, combinaison, et clarification des conventions prises de la littérature.

4.1.1 SISO

Quand un bloc d'information codées est transmis sur un canal avec AWGN, l'information du k^{ieme} symbole de la séquence transmise a trois sources indépendant.

1. La valeur reçu du k^{ieme} symbole
2. l'information a priori sur le symbole
3. l'information sur le symbole qui réside dans tous les autres symboles reçus du même bloc.

Le module SISO reçoit l'observation et l'intrinsèque comme entrés et sort l'extrinsèque et la probabilité a posteriori d'information codée qui a été transmise sur le canal.

Le SISO peut être réalisé avec plusieurs algorithmes tant que le codeur peut être représenté dans une treillis.

4.1.2 Configuration Parallèle du Turbo Code

Les codes convolutionels peuvent être concaténés en série ou en parallèle. Initialement on décrit la concaténation parallèle des codes.

4.1.3 Configuration série du Turbo Code

Les codes concaténés en série sont plus simples. Les composants exécutent les mêmes fonctions que dans le cas parallèle seulement l'interconnexion diffère dans le cas série.

4.2 Turbo Egalisation

4.2.1 Turbo Egalisation – Principe

L'égalisation entraîne une compensation pour la réception par rapport à une transmission à chemin multiple. Dans ce paragraphe on décrit la modification faite sur le turbo code pour égaliser les données reçues sur un canal IES.

Le canal IES est modélisé par un nombre de chemins qui arrivent à des instants différentes. Ce model de canal est adapté au treillis et alors le même algorithme employé dans le SISO peut être utilisé pour décoder le canal.

Deux modifications relative au décodeur concaténé en série vont être faite. Premièrement, l'égaliseur SISO qui remplace le premier SISO accepte la sortie du canal et calcule les vraisemblances du canal. Deuxièmement, les sorties extrinsèques sont modifiées. Les valeurs de la sortie supérieure ne sont pas utilisées, la sortie inférieure est cruciale. A l'intérieur de l'égaliseur SISO, les probabilités des symboles sont calculées, et puis l'algorithme du décodage est exécuté comme dans un SISO normal. La décision du turbo égaliseur est faite comme dans le SCCC.

4.2.2 Turbo Egalisation – Métriques de performance

Un des principales buts de n'importe quelle chaîne de transmission est d'avoir un communication sûre avec un SNR minimisé. Un faible SNR est typiquement associé à l'avantage de la faible consommation de la puissance, petite dimension, et meilleure portabilité. Malgré que la turbo égalisation réduit beaucoup le SNR, mais elle augmente la complexité qui consomme beaucoup d'énergie.

Pour représenter la sûreté du schéma de transmission, le rapport des bits erronés (BER) des données reçues est tracé par rapport au SNR. Le SNR est exprimé en terme de Eb/No. La complexité est exprimée en terme de nombre d'additions, de multiplications, maximisation, et l'exigence de mémoire.

4.3 Estimation du Canal

Pour conclure avec l'analyse de la turbo égalisation, le problème de la connaissance du canal doit aussi être soigneusement examiné. On propose ici au moins deux méthodes pour la re-estimation des coefficients du canal. Une partie importante de la dégradation est introduite par le mal fonctionnement de l'estimation du canal qui peut être récupérée par les méthodes de re-estimation, sans une grande augmentation de la capacité.

4.3.1 EM based re-estimation

Jusqu'à maintenant on considère que les coefficients du canal sont connus au récepteur, en pratique, ils sont estimés avec une séquence d'apprentissage insérée dans la séquence transmise. La méthode classique de l'estimation corrélative du canal (pseudo inverse method) cause une dégradation importante de performance par rapport à l'estimation parfaite. Cette lacune peut être réduit par l'utilisation des techniques avancées d'estimation de canal, comme l'algorithme EM, qui est puissant outil qui performe l'estimation des paramètres par la méthode de maximum de vraisemblance.

4.3.2 Re-estimation du canal bootstrap

Au lieu de considérer les symboles estimés après le décodeur IES, les décisions sont prises après le re-entrelacement de la séquence décodés. Comme ça, on profite de la diversité temporelle apportée par l'entrelacement et de l'efficacité du décodage du canal. On va décrire la succession des opérations.

1. Après la re-entrelacement de la sortie doux produite par le décodeur du canal, une décision dure est prise sur chaque bit de chaque symbole de la séquence. Une estimation des symboles utiles est alors disponible.
2. La matrice du système est formée
3. Une solution qui minimise la probabilité d'erreur est bien connue.

4.4 Entrelacement et Des-entrelacement

La fonction d'entrelacement permet la distribution temporelle d'une séquence d'erreur. Utilisé généralement avec les canaux variables avec le temps, l'entrelaceur est une fonction essentielle de la turbo égaliseur même si le canal n'est pas variable avec le temps.

4.5 Constellation Optimisée

On propose ici d'étudier l'influence du mapping sur le performance de la turbo égalisation. On montre que cette technique est intéressante pour un bon choix du mapping. Dans l'ordre d'expliquer le choix technique, on utilise deux différentes approches fournies par Ten Brink and Gorokhov.

4.5.1 The geometrical approach

Gorokhov propose deux critère de conception pour l'optimisation du mapping, basée sur le calcul des distances spécifiques de constellation. Le choix du mapping est alors atteint sans aucun notions sur le point de fonctionnement. Cependant, cette critère de conception donne une préférence pour les performances asymptotiques que pour le traitement itératif durant la convergence qui n'est pas tout le temps désirable.

4.5.2 The mean average mutual information approach

Ten Brink utilise les paramètres de l'information mutuelle, qui permet de distinguer exactement les performances de chaque mapping, cependant, le calcul de ce paramètre demande une simulation intensive et dépends du vue du point de fonctionnement.

Chapitre 5: La technique de la séparation itérative de l'observation

Un des problèmes le plus important dans la sûreté d'une communication de données sur un canal sélectif en fréquence est l'atténuation de l'ISI. Plusieurs systèmes de communications numériques, comme n'importe quel système mobile CDMA rencontre ce problème de transmission sur un canal à chemin multiple où le bruit, l'ISI sont deux des principaux facteurs qui dégrade le performance.

En plus l'augmentation de la capacité des systèmes est devenu une question majeure surtout pour le prochaine génération des systèmes cellulaires qui ont une capacité limitée par l'interférence.

Un approche promettant pour supprimer cet interférence est le travail sur le traitement du signal. Une solution optimale qui inclue estimation du canal, détection de symboles et décodage du canal apparaît exorbitant de point de vue complexité. Plusieurs méthodes conjointes itérative, utilise la décision moue ou dure sont proposées pour le cas des modulations sans et/ou avec étalement du spectre.

Dans ce contexte, on applique le concept nommé Séparation d'Observation qui sépare partiellement les observations utilisées pour l'estimation du canal des observations permettant la détection de symboles.

L'estimation du canal est très simple et consiste principalement d'un traitement classique de bootstrap utilisant la pseudo inverse linéaire. Pour chaque symbole, les techniques de détection prend en considération seulement la partie de l'observation, qui contient le symbole considéré et la longueur correspondante à la longueur de la mémoire du canal.

Les techniques de détection sont le ZF pour l'itération initiale et un annulatif d'interférence classique pour les autres itérations. On a aperçu que si le filtre assorti est utilisé pour l'itération initiale, le résultat restera le même. Une partie importante de la dégradation introduite par les erreurs de l'estimation du canal peut être récupérée par cette méthode de re-estimation, sans augmentation considérable de la complexité.

On applique dans ce chapitre la technique OS au système UMTS-TDD et aussi au système bande étroite avec une modulation binaire BPSK et des modulations haut niveau M-QAM sur un canal IES invariable avec le temps. Pour la comparaison, on va prendre en compte la technique itérative classique parallèle annulatif d'interférence (PIC) et la technique classique de bootstrap déjà appliquée au système GSM. Ainsi un traitement conjoint de détection de symbole et d'estimation du canal est une solution très attirante.

Notre technique SO est réalisée en associant un annulatif d'interférence adaptatif à la détection de symbole avec une estimation de canal pseudo inverse modifiée pour chaque symbole. Pour la boucle de retour, sortie moue et/ou dure du décodeur du canal est considérée. Cependant cette schéma de réalisation sur le concept SO n'est pas unique et la recherche pour des solutions meilleures est ouverte.

Pour une conception réelle comme la 2$^{\text{ième}}$ et 3$^{\text{ième}}$ génération et pour de raison de complexité du décodeur, on propose la boucle de retour dure pour le décodage (Décodage Viterbi).

On a appliqué cette technique au service 12.2 du système uplink UMTS-TDD et aussi au service 12.2 du système QPSK à bande étroite associé et modulation haut niveau M-QAM avec un code convolutionel de rendement 1/3. La longueur de la séquence d'apprentissage et le nombre de chemin estimée du canal sont deux paramètres laissées pour étudier leur effet sur la performance de la procédure SO. Les performances obtenues sont très promettant.

Selon le scénario considéré, les résultats de simulation ont montré proche de 1 dB gain pour la technique SO comparée à la technique classique de l'annulatif d'interférence, ce qui présentent des performances très attirantes.

Chapitre 6: Optimisation de la constellation avec la technique SO

Le décodage itératif est traditionnellement appliqué avec soit une concaténation parallèle soit une concaténation d'au moins deux codes. A l'émetteur, les deux codes sont séparés par un entrelaceur. Au récepteur, le décodage est itéré le long des deux composants de codes pour le système concaténé en parallèle, ou le deux décodeurs pour le système concaténé en série respectivement.

Deux approches principaux pour l'optimisation de la constellation:

Ten Brink utilise les paramètres de l'information mutuelle, qui permettent de distinguer exactement les performance de chaque constellation, cependant, le calcule de ces paramètres demande une simulation intensive et dépend de la vue du point de fonctionnement. Dans ces papiers on trouve une bref description de l'approche pour obtenir une règle de conception basée sur l'information mutuelle. Cela réduit d'une manière significatif le fardeau des simulations exhaustives pour trouver une bonne constellation car l'information mutuelle est très simple à calculer par l'intégration numérique.

Gorokhov approche l'effet de la constellation à travers l'analyse du rendement d'erreur de la démodulation/décodage à maximum de vraisemblance d'une part, et à l'analyse statistique de la métrique de bit demappé dans le première itération de démodulation d'autre part. Ces deux caractéristiques mène Gorokhov pour proposer deux critère de conception pour l'optimisation de la constellation, basées sur le calcule de distances spécifiques de la constellation. La choix de mapping est alors accompli sans n'importe quel notion du point de fonctionnement. Cependant, cette critère de conception donne préférence aux performances asymptotiques qu'au traitement itératif durant la convergence, qui n'est pas toujours désirable.

Dans ces papiers, on trouve une conception de la constellation qui compte sur l'observation commune dont les procédures de décodage itérative approche le comportement des décodeurs optimaux quand SNR augmente, ainsi on obtient la première critère de conception qui optimise la constellation pour assurer de bonne performance du décodage itératif, à des niveaux de SNR relativement hautes. La deuxième critère vise à améliorer la seuil du SNR pour des utilisations pratiques des décodeurs itératifs.

Dans ce chapitre, on suggère la conception d'une constellation optimale dans un schéma BICM-ID tout ça appliqué à un système avec estimation de canal SO et procédure IC. Cela veut dire que on applique le principe turbo dans deux cas: démodulation turbo et détection de symbole turbo. On ne mentionne pas les modulations à treillis codées (TCM) car il n'y a pas de code interne qui peut ajouter de l'information superflue. Le système peut être vu comme deux schémas de décodage itérative en série par lequel le décodeur interne est remplacé par un composant de demapping souple dans la démodulation turbo et un composant de détection de symboles dans la détection turbo. On applique le concept nommé la technique SO, introduite dans le chapitre 5 pour les applications CDMA et système à bande étroite; dans ce chapitre, le symbole peut appartenir à un groupe de modulations haut niveau (16-QAM) et le canal est considéré comme un canal IES invariable avec le temps. Chapitre 5 montre que la technique SO

1. Exhibe des meilleurs performances quand on l'utilise avec une large constellation comparé à IC,
2. Peut approcher et converger aux performances du canal parfaitement connu.

En se basant sur ces résultats proche du canal parfaitement connu, notre objectif principale est d'améliorer notre première itération, qui mène à améliorer les performances quand le canal est connu, par l'optimisation de la constellation. On va montrer comment le demapping itératif réduit le rapport de bits erronés dans ce système décrit précédemment.

On crée une constellation QAM binaire optimisée dans l'ordre d'augmenter le gain de codage, qui assume la connaissance parfaite des informations a priori. Cette expression est nommée la méthode de génie. Cette constellation optimisée peut être utilisée pour plusieurs applications importantes.

Cet algorithme est appliquer au cas de modulation à haut niveau (16-QAM…) et notre constellation optimisée est comparée aux constellations conventionnelles (Gray, Set partionning…) dans les deux cas la technique classique IC et le technique SO, et les résultats des simulations sont présentées.

Le faite de connaître que le codage et la modulation sont isolés par l'entrelaceur de bits, et d'identifier l'impacte du décodage itératif sur le moyen harmonique de la distance euclidienne minimale, on a développé un système turbo égalisation puissant avec une complexité relative.

La partie théorique et le grand nombre de simulations démontre que le moyen harmonique de la distance euclidienne minimale identifiée comme cruciale pour le BICM, peut augmenter fortement avec BICM-ID.

Une nouvelle constellation optimisée sous la critère du moyen harmonique pour la 16-QAM est présentée. Les simulations présentées démontre que même lorsque la constellation a les meilleures performances asymptotiques n'est pas nécessaire qu'elle aura les meilleures performances dans la turbo égalisation; certaines constellations ne sont pas adaptées à la turbo égalisation comme l'Anti Gray. On présente notre constellation optimisée qui semble une bonne compromis et qui améliore la performance de la constellation de gray by 0.5dB sur le SNR élevé mais malheureusement perd prés de 0.5dB sur faible SNR.

Finalement, notre système proposé améliore les performances du système classique aussi lorsque on l'utilise avec ou sans SO et ID. notre système sans la technique SO obtient un gain de 0.5dB, avec SO et sans ID présente un gain de 1 dB, et lorsqu'on utilise les deux techniques proposées SO et ID, on obtient un gain de 1.5dB par rapport au system classique utilisé actuellement.

Chapitre 7: Low-Density Parity-Check Codes

Les codes LDPC ont été inventés par Gallager dans sa thèse. Ils n'ont pas eu l'attention nécessaire pour quelques dizaine d'année jusqu'à récemment lorsque les très réussis turbo codes ont été découvertes. Les codes LDPC ont été redécouverts by Spielman et Mackay. Pour plusieurs canaux et décodeurs itératifs, ils exhibent un phénomène du seuil: lorsque la longueur de code tend vers l'infinité, une petite probabilité de bits erronés aléatoire peut être obtenue si le niveau de bruit est plus petit qu'une certaine seuil. Pour de niveau de bruit au dessus de ce seuil, la probabilité de bits erronés est plus grande d'un constant positif. Gallager a observé ce phénomène pour de canal binaire symétrique lorsqu'il a introduit les codes LDPC réguliers en utilisant une construction explicite de graphes réguliers. Luby a généralisé cette idée aux codes LDPC irréguliers et construits d'une manière aléatoire, il a montré que les codes irréguliers se comporte mieux que ceux réguliers.

Cette observation est généralisée par Richardson et Urbanke pour une grande gamme de canaux à entrée binaire, (canal à effacement, symétrique, Laplace, et AWGN) et pour plusieurs algorithmes de décodage. Richardson démontre le théorème de concentration générale montrant que les performances du décodeur pour des graphes aléatoires convergent à sa valeur attendue tant que la longueur du code augmente. A cause de la difficulté à déterminer ces performances avec un ensemble de codes finis, on a utilisé le comportement attendu à la limite des codes infinis qui peut être déterminé du cycle libre dans le graphe correspondant. Ils ont défini un seuil comme indiqué précédemment pour l'ensemble des codes irréguliers et aléatoires spécifiés par les distributions de degré, et développés un algorithme nommé l'évolution de la densité pour le calcule itératif de la densité du message, permettant la détermination du seuil. Utilisant ce résultat, ils ont construit des codes LDPC qui de manière claire battent le turbo code sur un canal AWGN. Récemment, ce résultat a été amélioré, en suggérant que les codes LDPC force à approcher asymptotiquement la capacité du canal pour les canaux AWGN.

En calculant les seuil et en optimisant les distributions de degré, l'utilisation de l'évolution de la densité est une tâche intensive en calcule pour la plupart des canaux sauf le canal à effacement qui avec l'évolution de la densité devient unidimensionnel et il est possible de faire plus d'analyses et encore de construire des codes qui approchent la capacité. Pour les canaux plus intéressants, compris le canal AWGN, cependant, l'évolution de la densité est très compliqué pour être analyser.

Dans ce chapitre, on introduit les codes LDPC et quelques une de leurs propriétés, méthodes de construction et algorithmes de décodage. Les codes LDPC sont des codes binaires linéaires en bloc et leur nom indique qu'ils ont une matrice de parité qui a une faible densité de 1s. on distingue entre les codes LDPC réguliers et irréguliers où seul les codes irréguliers peuvent approcher la capacité. Pour obtenir un bon codes LDPC il faut utiliser des méthodes de conception bien spéciales. Quelques une de ces méthodes dépendent de l'évolution de la densité. Pour approcher la capacité, des blocs de longueur infinie sont nécessaires. Cependant les résultats des simulation montre que les codes LDPC fonctionnent près de la capacité avec de longueur de bloc modéré (N=10^4 ... 10^5).

Chapitre 8: Décodage des Codes Non Binaires et Adaptatifs

Récemment, beaucoup de recherche ont été dévoué aux codes LDPC binaires et à cause de leur performance proche de la capacité, ils ont surgi comme un candidat potentiel et promettant pour les codes correcteurs d'erreurs dans les systèmes futures sans fils et deviennent un des concurrents sérieux du turbo code. Cependant, une amélioration importante dans les performances peut être aboutie en utilisant les codes non binaires.

En même temps, il y en a eu un intérêt augmentant dans les schémas de décodage à décision souple pour les codes correcteurs d'erreurs. Les différentes algorithmes utilisés pour le décodage des codes LDPC (belief propagation, sum-product…) approchent itérativement la solution à maximum de vraisemblance pour le problème de décodage. Le décodage LDPC est fait par le passage des messages sur les edges entre les nœuds de la graphe bipartie "graphe de Tanner" qui représente la matrice de parité des codes LDPC. La complexité moyenne du processus de décodage est le produit de trois facteurs:

1. Le nombre des opérations par nœud,
2. Le nombre moyen d'itérations, et
3. Le nombre des nœuds actives dans chaque itération.

Un schéma de décodage puissant, comme le turbo code, nécessite l'utilisation des algorithme de décodage qui calcule les probabilités a posteriori basé sur le symbole par symbole. Une contribution au décodage APP est faite par Gallager où le décodage est fait par une module APP sur chaque nœud de parité. Dans la même période, une application du décodage APP aux codes décodables par seuil est faite par Massey. Malheureusement les algorithmes APP proposés ont une grande complexité qui augmente avec le nombre d'états et le nombre de transition dans le treillis du code. De là les algorithmes APP qui réduit la complexité de calcule et/ou les besoins de mémoire sont intéressants pour les applications pratiques. Beaucoup d'auteurs ont observé que le APP peut être calculé sur le treillis du code dual qui simplifie le décodage si le code est de rendement élevé, terminant après quelques transitions du treillis dual par rapport au treillis original.

Ici, on généralise l'idée originale du décodage "dual APP" au alphabets non binaires. Par alphabet non binaire, on considère que les symboles du code sont pris d'une certaine extension du corps $GF(q=2^m)$.

Le premier travail sur les codes q-aire LDPC apparaît avec Mackey et Davey. Dans ce chapitre, on adresse les codes non binaires en conjonction avec une modulation non binaire. On présente une règle de décodage MAP qui est, d'une manière, le décodage dual de l'algorithme de Viterbi pour les codes linéaires. Ce code est aussi approfondi, mais dans un sens que chaque mot du code dual est utilisé dans le processus de décodage. Cela veut dire qu'en pratique la règle de décodage peut être utilisée seulement avec les codes dont leur code dual a un petit nombre of mot de code, i.e. codes à rendement élevé ou codes avec un rendement petit ou moyen mais avec une contrainte de faible longueur. L'application de ces codes non binaires à une modulation orthogonale est d'une importance particulier en pratique. Cette règle joue un rôle majeur dans les cas non itératifs. On décrit encore comment la complexité peut être réduite en utilisant notre règle de décodage et accélère le calcule avec l'utilisation des transformées d'Hadamard rapides de la probabilité.

Finalement, on présente le codage LDPC Adaptatif adressé aux systèmes sans fils de la $3^{ième}$ génération. L'utilisation de la modulation et du codage adaptatives (AMC) est un des clés permettant aux techniques dans les standards de $3^{ième}$ génération à achever une efficacité spectrale très élevée sur des canaux radios. La principale idée de AMC est de changer dynamiquement les schémas de modulation et codage pour les adapter au total efficacité spectrale des conditions de canal. La décision sur la sélection appropriée est faite au récepteur selon les conditions observées du canal avec les informations retournées à l'émetteur dans chaque trame.

Dans ce chapitre, on change seulement le niveau de la modulation durant le codage pour prendre avantage de toute la capacité en utilisant la technique Water Filling, l'ensemble des modulations candidates sont BPSK, 4-QAM, 8-QAM, 16-QAM, 64-QAM et 256-QAM avec la constellation de gray. Où on considère que les caractéristiques du canal sont bien établies et le canal peuvent être sondées pour obtenir une bonne qualité d'estimation du canal, l'émetteur ensuite utilise cette estimation pour choisir la constellation la plus appropriée.

Ici, on ne rentre pas en détail de l'AMC; on va seulement comparer les performances de l'AMC si on adapte le niveau de la modulation dans le modulation ou dans le codage. Dans le premier cas, on considère à l'émetteur un codage LDPC binaire et une modulation où le niveau est adapté au profile du canal, et au récepteur on calcule les probabilités sur chaque bit qui seront utilisées dans le décodage itératif. Dans le second cas, on considère à l'émetteur un codage LDPC adaptatif où le niveau est adapté dans la matrice de codage et une modulation, et au récepteur on calcule les probabilités sur chaque symbole qui seront utilisées dans le décodage LDPC adaptatif. On va voire que la règle de décodage présentée précédemment peut être facilement appliquée au décodage non binaire et au décodage adaptatif et les performances obtenues sont très intéressants.

Les résultats des simulations sur canal AWGN suggèrent que les codes LDPC non binaire (resp. SPC) fonctionnent mieux que les codes LDPC binaires (resp. SPC) spécialement lorsque la modulation est non binaire, et démontre que malgré quand le code turbo fonctionne mieux que le code LDPC binaire, le code q-aire LDPC peut rattraper cette perte et passe devant. Le gain du code q-aire LDPC (resp. q-aire SPC) est de 2 dB (resp. 1 dB) sur un corps de galois GF(256) par rapport au code LDPC (resp. SPC), malgré qu'on n'a pas de gain sur GF(4) mais ce gain augmente avec la taille du corps de galois, i.e. la taille de la modulation. Après on montre que le code q-aire LDPC fonctionne mieux que le turbo code de l'UMTS sur GF(q>4) de 0.6 dB ou 0.8 dB cela dépend de q.

Finalement, on propose pour les systèmes OFDM, des codes LDPC adaptatifs qui peuvent être simplement décodés par la règle de décodage présentée dans ce chapitre. En dépit que les performances des codes LDPC adaptatifs approche à des SNR élevés des codes LDPC binaires, il restera un très bon sujet attirant pour les recherches future.

Chapitre 9: La Technique SO Itérative avec Codes LDPC Non Binaires

Pour les canaux IES, plusieurs d'autres codes ont été considérés pour l'utilisation dans la turbo égalisation, incluant les turbo codes parallèles, les codes convolutionels, les codes de parité et plus récemment les codes LDPC.

On a montré que les codes LDPC sont des codes robustes et ont des performances excellentes sur plusieurs canaux, et lorsque le décodage sur le canal IES, le rapport de bits erronés peut être amélioré davantage par l'utilisation de la turbo égalisation. Les codes LDPC qui approche la capacité du canal AWGN sous le décodage itératif ont été construits par Chung. Depuis ce temps, l'évolution de la densité a été utilisé pour optimiser les codes LDPC pour plusieurs canaux sans mémoire, et les résultats suggèrent, pour chaque canal, que les séquences de codes LDPC décodés itérativement peuvent en effet approcher la capacité du canal. En réalité, la découverte du canal dont la capacité ne peut être approchée par les codes LDPC peut être plus surprenante que la preuve que les codes LDPC décodés itérativement peuvent approcher la capacité de n'importe quel canal symétrique binaire.

Comme on a déjà vu, l'idée de décoder un code transmis sur un canal avec mémoire par l'intermédiaire d'itération a été introduite par Douillard dans le contexte du turbo code. Cet approche peut aussi être généralisé pour les codes LDPC par la construction d'un seul graphe qui représente les contraintes du canal et du code. Cette idée a été examinée pour canal à réponse partielle par Kurkoski, Siegel, et Wolf. Jusqu'à récemment, il était difficile de comparer les performances de la turbo égalisation avec la capacité du canal car la capacité du canal à entré binaire était inconnue. Récemment, une nouvelle méthode a gagné l'acceptation pour l'estimation et un certain nombre d'auteurs ont commencé à désigner les schémas de codage LDPC qui approche les rendements d'information accomplies des ces canaux. Les principaux sujets présentés sont:

1. Théorème de concentration pour les codes Gallager et le décodeur sum-product de passement de message sur les canaux binaires IES.
2. La méthode d'évolution de la densité pour calculer les seuils de la performance zéro erreur sur ces canaux;
3. Théorèmes établies qui les performances asymptotiques de codes Gallager utilisant l'algorithme de sum-product est bornées par le rendement d'information symétrique et la capacité i.i.d.
4. Et le calcule de la borne BCJR, qui est le limite de performance zéro erreur de l'algorithme sum-product si la portion de treillis de l'algorithme est exécutée une seule fois.

Comme dans le cas avec DE pour les canaux binaires, l'évaluation du seuil du code et l'optimisation de ces seuil sont numériquement faites. L'analyse de ce système est plutôt complexe car l'algorithme BCJR est utilisé pour décoder le canal. A cause de la capacité du canal avec mémoire est généralement n'est pas accomplie par le signalement équiprobable, un peut au lieu viser le rendement d'information symétrique du canal (SIR). Le SIR est défini comme le rendement maximal d'information accomplie par le codage aléatoire avec des symboles d'entrée équiprobable. A cause des codes linéaires utilise tous les entrées équiprobables, le SIR est aussi le rendement maximal est directement accomplie avec les codes linéaires.

Dans ce chapitre, la technique de séparation d'observation sur la technique itérative conjointe de l'estimation du canal et de la détection de symboles ont été considérée pour les systèmes 16-QAM codés par le code LDPC non binaire. On a considéré la modulation entrelacé et turbo codé par bit, modulation LDPC codé par bit et modulation LDPC codé par symbole à l'émetteur. Au récepteur, on considère la technique turbo égalisation avec l'estimation SO dans ces trois cas. Et on compare leur résultats. Le performance obtenue est très attractif.

Notre système proposé est réalisé par l'association de codage LDPC non binaire (16-aire) à l'émetteur et la technique SO avec le décodage LDPC non binaire. On a montré que le bon choix de la distribution de degré des codes LDPC non binaire nous mène à améliorer les performances des turbo codes; en dépit des performances des codes LDPC binaires avec le même degré de distribution ne sont pas assez bien. L'inconvénient est la complexité, mais on a présenté dans le chapitre 8 une règle de décodage pour réduire la complexité et accélère l'algorithme.

Selon le scénario considéré, les résultats des simulations montrent un gain près de 0.5 dB pour les codes LDPC non binaires avec la technique SO par rapport aux turbo codes avec la technique SO, et 3.25 dB par rapport aux codes convolutionels avec la technique SO où le BER=10^{-4}; c'est un très bon résultat et une performance attirante.

Chapitre 10: Conclusions et Perspectifs

10.1 Contributions

Dans cette thèse, on a proposé et analysé des stratégies à faible complexité pour combattre l'interférence du canal lorsque l'information sur le canal est indisponible à l'émetteur et estimée au récepteur. De plus, dans le chapitre 8 on a introduit les codes LDPC adaptatifs. Ces stratégies sont centrées sur la technique de la turbo égalisation, qui annule l'interférence du canal des signaux reçus utilisant les décisions celles qui améliore la sûreté avec chaque itération consécutive. Parmi les algorithmes de réception considérés dans le passé, le turbo égaliseur est considéré la plus importante en termes de performances, en plus que l'égaliseur converge avec quelque itérations. Mais il demande une initialisation par un autre algorithme pour atteindre des bonnes performances.

Les propriétés de convergence de l'égaliseur, qui dépend du canal et du SNR au récepteur, ensemble avec l'utilisation du code de canal, détermine les performances de récepteur. On a démontré que la turbo égalisation travaille particulièrement bien lorsqu'elle est utilisée en conjonction avec la technique SO en estimation. Pour approcher une communication sûre au rendement proche de l'information mutuelle de l'interférence du canal, un codage du canal est nécessaire, et une conception convenable du code peut optimiser la transmission et améliore le performance de récepteur si les statistiques du canal sont connues ou no de l'estimation SO.

La technique de séparation d'observation sur le traitement itératif conjoint de l'estimation du canal et de la détection de symboles a été considérée pour les systèmes mobiles CDMA et à bande étroite. Le concept SO correspond à découpler les observations utilisées pour l'estimateur de celles utilisées pour la détection. Les performances obtenues sont très attirantes. Dans cette thèse, notre SO est réalisée par l'association d'un annulatif d'interférence adaptatif pour la détection de symboles avec une estimation de canal pseudo-inverse modifiée pour chaque symbole. Pourtant, cette schéma de réalisation de la concept SO n'est pas unique et la recherche pour une solution meilleure est lancée. On a appliqué cette technique au service 12.2 de l'UMTS-TDD uplink, au service 12.2 du système à bande étroite associé et aussi au systèmes de haut niveau M-QAM.

Après, on a analysé et évalué une nouvelle approche pour la conception du BICM-ID pour les canaux IES. Par connaître que la modulation et le codage sont séparés par un entrelaceur de bit, et identifier l'impacte du décodage itératif sur la moyenne harmonique de la distance euclidienne minimale, on a développé un système de turbo égalisation puissant à complexité relativement faible. On démontre que la moyenne harmonique de la distance euclidienne minimale identifiée comme cruciale pour le BICM, et peut augmenter fortement avec le BICM-ID. Ensuite une nouvelle constellation optimisée sous la critère de la moyenne harmonique pour 16-QAM est présentée. On remarque que même lorsque la constellation a des meilleures performances asymptotiques, ce n'est pas nécessaire que la constellation aura des meilleures performances dans la turbo égalisation; quelques constellations ne sont pas adaptées pour la turbo égalisation comme Anti-Gray.

Ensuite on a essayé d'améliorer les schémas de codage et du décodage et on focalise sur les systèmes récepteurs employant les codes LDPC. Les performances du récepteur sont fortement améliorées, si l'estimation des coefficients du canal et la variance du bruit sont disponibles au récepteur. Cependant pour des bonnes performances du récepteur, on demande quelques itérations de la turbo égalisation et sur le décodeur LDPC.

Après, on a présenté une règle de décodage pour les codes LDPC non binaires. Le codage et décodage LDPC non binaire améliore les performances du code LDPC binaire spécialement quand la modulation est non binaire, et on prouve que même lorsque le turbo code est meilleur des codes LDPC, les codes LDPC non binaires peut rattraper cette perte et mieux. Considérant la complexité, la règle de décodage est pratique pour le haut rendement, applicable aux codes LDPC non binaires et réduit énormément la complexité en utilisant la propriété de dualité et accélère le calcule en utilisant le FHT. La complexité de la règle de décodage pour les codes linéaires est comparable à la complexité de décodeur Viterbi de code dual.

Finalement, la technique SO sur le traitement itératif de l'estimation du canal et de la détection de symbole a été considéré pour les systèmes 16-QAM codés par un code LDPC non binaire. On l'a comparé au modulation codée par un turbo code et au modulation codée par un code LDPC binaire. Au récepteur, on considère la technique de turbo égalisation avec l'estimation SO dans ces trois cas. Notre système proposé est réalisé par l'association du codage LDPC non binaire à l'émetteur et la technique SO avec décodage LDPC non binaire. On a montré que un bon choix de la distribution de degré nous mène à améliorer les performances du turbo codes; en dépit que les performance des codes LDPC non binaires avec la même distribution de degré ne sont pas bonnes. L'inconvénient est la complexité, mais on a présenté dans le chapitre 8 une règle de décodage pour réduire la complexité et accélérer l'algorithme.

On discuter dans cette thèse:

- Dans le chapitre 5, la technique SO pour l'estimation du canal.
- Dans le chapitre 6, la conception d'une constellation optimale pour le schéma turbo égalisation avec la technique SO.
- Les propriétés essentielles des codes LDPC dans le chapitre 7. Dans le chapitre suivant, les codes LDPC non binaire, la règle de décodage non binaire, et les codes LDPC adaptatifs.
- Dans le chapitre 9, application des codes LDPC non binaire sur la schéma de la turbo égalisation avec la technique SO

10.2 Futures Travaux

Bien que dans les chapitres précédents on a vu des possibilités pour plus de recherche, on discute maintenant quelques directions de recherche les plus fructueux inspirées de cette thèse. Il y a beaucoup de questions ouvertes qui peuvent attirer plus de recherche.

Le problème de la détection conjointe demande plus d'études. Au moins, on doit essayer de réduire la complexité de façon que la turbo égalisation soit plus simple à implémenter.

On a besoin de faire l'analyse de la convergence en utilisant l'EXIT chart ou n'importe quelle autre méthode.

Pour les codes LDPC, il y a encore du travail pour déterminer un algorithme généralisé de l'évolution de la densité pour les systèmes non binaires par exemple et peut-être les codes LDPC adaptatifs dans un système turbo égalisation sur un canal IES.

Finalement, tout ce travail dans cette thèse peut être simplement étendu aux systèmes MIMO qui sont prise en compte dans les normes de la future génération sans fil.

Chapter 1
Introduction

In a digital communication system, channel interference limits the ability of the receiver to detect the discrete symbols sent by the transmitter. The channel is a part of the communication system we typically cannot change, so techniques to detect the transmitted symbols that take into account the channel interference are important.

Use of the "turbo" principle to iteratively decode parallel or serial concatenated error correction codes has lead to performance closely approaching Shannon's theoretical channel capacity limit. This principle has also been used to improve the reliability of communications over a multi-path channel.

In this introductory chapter, we present some relevant background material. We present the outline for the rest of the thesis in Section I.2.

1.1 Background

One of the important problems in reliable data communications over frequency selective fading channels is the mitigation of inter-symbol interference (ISI). Many of practical communication systems encounter this problem of data transmission over a multi-path channel where the noise and ISI are two of the main factors that degrade the performance. To protect the integrity of the data to be transmitted, a controlled amount of redundancy is added (encoding) using error correction code (ECC).

In the past few years the number of users of wireless communication systems has been growing exponentially. For this reason increasing system capacity is a critical issue, especially for the next generation cellular systems whose have a capacity limited by interference. Reduction of channel interference (ISI), co-channel interference from own cell (MAI, Multiple Access Interference), or neighboring cells (inter-cell interference) improves the system performance. Promising approach to suppress interference and multi-path channel distortion is the work over signal processing.

Mobile systems add an additional dimension to the problem; the signal transmission pathways are changing with time. When the transmission uses a sequence of digital symbols, a multi-path transmission channel is also referred to as an ISI channel. Symbol that arrive at the receiver simultaneously by different paths are superimposed just like the image on the television when we obtain a double images. The goal of the receiver designer is to combat time-variant ISI due to limited bandwidth, multi-path propagation and movement so that the integrity of the transmitted message is maintained.

Error correction codes have significantly improved the reliability of digital signal transmission and reception. The relatively recent introduction of the "turbo" principal by Berrou and al. [1] has enabled the design of communication systems with channel capacities closely approaching Shannon's theoretical limit (for a bit rate equal to ½, a performance within 0.5 dB of the capacity limit was achieved). The turbo principle involves iteratively decoding a hugely complex code generated by two convolutional encoders separated by an interleaver.

Since their first presentation in 1993 by Berrou and al. [1], a considerable amount of the work has been done on turbo codes, both for improving the original scheme and for better understanding the reasons for their astonishing performances. Moreover, the "turbo principle" has been extended to many fields other than channel coding theory and should now be regarded as a general approach for combining and serially performing in an iterative way two or more tasks in the receiver digital communication chain. Based on the invention of turbo codes [1], in the past few years, a new concept, called "turbo equalization", has emerged as a way of efficiently fighting against channel ISI. The turbo principle has been first applied by Douillard and al. [5] to the field of equalization. The basic idea consists of considering the channel as a time-varying non-recursive non-systematic convolutional code, assuming an outer convolutional channel encoder and a channel interleaver, the reference turbo detection scheme is then formally analogous to a serial concatenation of convolutional codes and the same iterative techniques can be applied to realize joint detection and decoding [7]. Equalization and soft output channel decoding are concatenated and executed in an iterative way.

Performing iterations like in turbo decoding can improve the bit error rate (BER) and the frame error rate (FER) dramatically. In fact simulation results show that all the ISI can be eliminated by such a process and the performance of coded signals over the Gaussian channel can be reached assuming perfect channel estimation and sufficient interleaver depth.

To combat the effects of ISI, linear (linear equalizer, LE), or non-linear processing (decision feedback equalizer, DFE, or detector) of the received symbols can be applied [9], but optimal methods for minimizing the FER or BER are non-linear and based on ML estimation, e.g., the FER-optimizing Viterbi Algorithm [10]. ML estimation turns into maximum a posteriori probability (MAP) estimation, e.g., the BER-optimizing BCJR algorithm [4, 11], in presence of a priori information about the transmitted data.

In the original paper describing the turbo detector [5], Douillard's approach is based on maximum likelihood (ML) sequence estimation for equalization as well as for soft output channel decoding (a min-log-BCJR algorithm was used for symbol detection and a low-complexity SOVA for channel decoding).

More recently, Bauch and al. in [8], optimal symbol-to-symbol BCJR detectors and decoders with a convolutional ECC have been introduced to improve the scheme.

Unfortunately, the complexity of all those ML/MAP or sub-MAP devices depends exponentially on the length of the channel's impulse response (CIR) and the size of modulation, and it might become quickly prohibitive when higher level modulations rather than simple BPSK (or GMSK) and 6-tap (or more) CIR are considered. Consequently, the first challenge is the reduction of the overall computational complexity of the turbo detector. Since the complexity will be dominated by the ISI detector, (all know turbo equalization schemes apply soft output channel decoding to determine the extrinsic information which is the additional information provided by the channel decoder. This extrinsic information is fed as a priori information to the next equalizer's input), an important number of previous studies focus on the design of a low-complexity turbo detector based on a suboptimal soft-input soft-output (SISO) sequence estimator, e.g., SISO-delayed decision feedback sequence estimator (SISO-DDFSE) coupled with a minimum phase pre-filtering, as proposed by Berthet and al. [12]. They analyze and compare several strategies for iteratively decoding trellis-encoded signals over channels with memory, and soft-in/soft-out extensions of reduced complexity trellis search algorithms such as DDFSE or parallel decision feedback decoding (PDFD) algorithms are used instead of conventional BCJR and min-log-BCJR algorithms; the second attractive problem studied in [12] is the combination of the detection and a powerful iteratively decodable codes (turbo codes) and it focus on the serially concatenated codes (SCCC) because he could be more efficient than parallel concatenated codes (PCCC).

Ariyavistakul and Li [13] proposed a joint coding equalization approach, distinct from turbo equalization, working with convolutional coding and a DFE; here, within the DFE, soft information from the DFE forward filter and hard decisions from the decoder using the Viterbi algorithm (VA) are fed back.

Another common technique to decrease the complexity of the MAP equalizer is to reduce the number of states in the trellis; which was applied to turbo equalization in [15].
In order to reduce the computational complexity, the MAP detector can be advantageously replaced by an ISI canceller. This new receiver makes it possible to almost completely overcome ISI over time invariant and/or time varying Rayleigh channels for high spectral efficiency modulation.

Therefore Glavieux [6] is one among the firsts to replace the ML equalizer by a so called ISI canceller. Since this is an adaptive weight FIR filter, its computational complexity depends only in a linear way on the length of the CIR.

Glavieux and al. in [17] propose a low complexity soft-input soft-output M-ary turbo equalizer that allows inter-symbol interference to be reduced drastically. The equalizer is close to an inter-symbol interference canceller (IC). For each stage, the equalizer is updated according to the mean square error (MSE) criterion, and they propose an adaptive algorithms such as stochastic gradient least mean square (SG-LMS) or recursive least square (RLS) which can be used for updating equalizer parameters, these algorithms minimize the MSE criterion.

This idea is enhanced in [18], where the filter coefficients are obtained using the LMS algorithm to match the output of a MAP equalizer, for varying signal-to-noise ratios (SNRs) and feedback information constellations, a linear estimate of the MAP equalizer is stored in a table and used for equalization in the receiver.

Tuchler and al. in [16] introduce new approaches to combining equalization based on linear filtering with decoding, they replace the MAP equalizer with an LE and a DFE, where the filter parameters are updated using the MMSE criterion and derive four different implementations of this general approach.

Wang and Poor [14] proposed a turbo equalization like system part of a multi-user detector for CDMA, this iterative scheme is based in the turbo equalization using an LE to reduce ISI and MAP decoding, the MAP equalizer is thus replaced with an LE, whose filter parameters are updated for every output symbol of the equalizer, and they extend this scheme on any type of channels and in presence of MAI and ISI.

Vila and al. in [19] propose to optimize the interference canceller at each iteration, thanks to the training sequence, so to improve performance in a reduced number of iterations. Lampe and Huber in [21] show that iterative multi-user interference suppression based on adapted MMSE filters combined with serial successive cancellation and single user decoding can reach near optimum performance within a few iteration cycles. Further, they evidence that for sufficiently reliable symbol estimates soft decision feedback can be replaced by hard decision feedback without any performance degradation but with significant savings in complexity.

The approaches in [6, 14, 16-19, 21] address a major shortcoming of the classical turbo equalization scheme, which is the exponentially increasing complexity of the equalizer for channels with a long impulse response or a large signal alphabets.

To overcome the channel symbol detection problem, several iterative multi-user detection and decoding schemes have been proposed, providing close to optimum performance if channel state information (CSI) is available. Unfortunately, a simple concatenation of these multi-user channel symbol detection algorithms with standard estimation schemes results in a considerable performance loss, since channel estimation is still done without exploiting information on multi-user interference. More advanced techniques, also dealing with the problem of channel estimation, use the a priori knowledge with respect to the interference, e.g., its correlation properties, to avoid distortion due to other users in the channel estimation and symbol detection procedure.

The standard approach to equalization with unknown channels is to generate a single channel estimate based on the statistics of the channel. This requires either a training sequence or a delayed decision directed approach. However, a more integrated technique, such as maximum likelihood sequence estimation (MLSE) linked with per-survivor processing (PSP), provides superior performance. More recently, the advent of "turbo processing" [1], has revitalized interest in maximum a posteriori (MAP) equalization in preference to MLSE.

Most previous work on iterative equalization and decoding assumes that the channel impulse response (CIR) is known to the receiver. To conclude with the turbo detection analysis, the problem of channel knowledge also has to be raised and carefully investigated.

The optimum receiver for transmission over unknown fading channels performs joint channel state and data symbol estimation. In [6], a least mean square (LMS) type of channel estimation algorithm is used besides a LE to estimate and track the CIR, where both the estimator and the equalizer incorporate the feedback information from the decoder. Other approaches use this information for estimation and equalization simultaneously, e.g., using a non-linear Kalman filters based on soft statistics. In [22] Tuchler and al. propose a recursive least square (RLS) type of iterative channel estimation algorithm based on soft information from the decoder, which is distinct from the equalization algorithm, and show that this scheme does not always give a better estimate of the channel when the channel is time-invariant. Therefore they devise a simple criterion to decide whether this soft information should be used for estimation. In [24] indeed, it is shown by simulations that MAP and sub-MAP devices are very sensitive to channel estimation. (A degradation of 2.9 dB occurs when assuming mismatched channel estimation by classical technique). In [12] Berthet and al. propose at least two methods for re-estimating the channel coefficients. The first method exploits the EM algorithm [25]. The EM iteration is done after each turbo detector iteration. The second method is even simpler and basically consists of a simple bootstrap process using linear pseudo-inverse. A significant part of the degradation introduced by mismatched channel estimation can be recovered by such re-estimation methods, without substantial complexity increase.

In [23] Linda and al. present an extension to the block processing MAP equalizer for the case where the channel is not assumed to be known, they use an expanded trellis so as to include extra memory for measuring the channel. This allows joint channel estimation and equalization. Expansion of the state space is made possible by the fact that the low-pass nature of the fading effectively introduces correlation into the received signal. The expanded state can then be used to form separate channel estimates for each trellis state, the size of the expanded state is fixed and MMSE techniques are proposed for forming channel estimates. In [20] Lampe propose a practically interesting approach for iterative channel estimation, for a multi-user detection, and single-user decoding based on MAP symbol-by-symbol estimation for DS-CDMA, he shows that near single-user channel phase and amplitude estimation accuracy is achieved for frequency selective fading channels, even in highly loaded systems, this iterative receiver performing pilot symbol aided channel phase and amplitude estimation as well as data and symbol estimation for coded transmission. This estimator consists in simplified minimum mean squared error (MMSE) channel estimation filters for each user and each path.

Strauch and al. in [26] derive a suboptimal estimator with substantially lower complexity, based on feedback soft output from the equalizer to the channel estimator (reduce the error propagation) which will significantly reduce the complexity with only a small loss in performance. In [19], Vila and al. propose to optimize the equalizer structure (an interference canceller) thanks to the training sequence available to estimate the CIR.

It has been shown that iterative demodulation improves dramatically the performance of the receiver, but that improvement depends substantially on the transmitter and specially on the chosen signal labeling. Particularly in recent research, iterative decoding algorithms for spectrally efficient modulation have become a vital field of research in digital communications. Thus it's necessary to show the influence of the mapping on the signal-to-noise ratio (SNR) threshold and on asymptotic performance.

Two principals approach for optimizing the labeling map; Ten Brink [27, 28] and Gorokhov [29]. In [27], Ten Brink and al. propose a design rule based on mutual information to find the 'best' mapping. As turned out in simulations [27, 28] the right choice of the mapping is crucial for a good performance of iterative demapping and decoding. In these papers we find a briefly description of an approach to obtain a design rule based on mutual information. This significantly reduces the burden of performing exhaustive simulations to find good mapping since mutual information is very easy to calculate by numerical integration. Gorokhov, in [29], approach the effect of signal labeling through the error rate analysis of maximum likelihood demodulation/decoding on one hand, and the statistical analysis of de-mapped bit metrics at the first demodulation iteration on the other hand. These two features lead to two different optimization criteria, which are used to design labeling for some practically important cases. In these papers we find a mapping design that relies upon the common observation that iterative decoding procedures approach the behavior of the optimal decoders as SNR grows, thus we obtain the first design criterion that optimize the labeling map to ensure a good performance of the iterative decoding, at a relatively high SNR. The second design criterion aims at an improved SNR threshold of practically used iterative decoders.

Finally, there is a large theoretical work about choice of interleaver, block length, and encoder polynomials. The optimal interleaver is unknown; however, Benedetto and Montorsi [30] have determined an upper error bound for a turbo code using a theoretical interleaver called a uniform interleaver. Furthermore they showed that random interleaver will perform close to the theoretical uniform interleaver. It is important for the interleaver to interleave code symbols and not simply code words. This is especially important when transmitting over a channel with memory. It is advantageous for the block length of the coded information (and interleaver) to be long, but this leads to a proportionally long latency at the receiver.

It is important to take note that the turbo algorithm is not optimal. However, evidence suggests that the sub-optimal algorithms used in turbo principle can perform close to the optimal. Actual maximum likelihood decoding is impossible because of the huge number of states generated by interleaving a large block of data.

1.2 Thesis Outline

In this thesis, we develop practical schemes to mitigate the effects of channel interference to make digital communication more reliable. We focus on the scenario in which the transmitter has no knowledge of the channel interference as is often the case in, for example, wireless communication over radio waves.

In chapter 2, we describe a discrete-time baseband model for digital communication systems and define the associated detection problem. We present a broader perspective where we explain how detection can be combined with channel coding and decoding in different communication scenarios to reliably transmit information at rates close to the theoretical maximum. We also in chapter 3 present and remind some classical detection schemes which have be presented in the literature and some books but grouped in the same chapter to facilitate the introduction of turbo equalization done in chapter 4, which begins with an explanation of turbo coding theory. It then shows how the turbo scheme can be modified to develop a "turbo equalizer" to equalize the output from an inter-symbol interference (ISI) channel. Next the necessary modifications of a turbo equalizer to allow adaptation to a time-varying ISI channel are described and the issue of receiver complexity is discussed.

The concept named Observation Separation (OS) technique which partially separates the observation used for channel estimation from the observation that allows the symbol detection is introduced in chapter 5. We apply this OS technique to the UMTS-TDD system and also to a narrowband system with a binary modulation BPSK and high level modulations M-QAM on a time-invariant ISI channels. For the purpose of comparison, we will take into account the classical iterative parallel interference cancellation (PIC) technique and the classical bootstrap technique already applied to GSM system. Thus joint processing of symbol detection and channel estimation is an attractive solution. Next, we will try to improve the performances of our system by suggesting in chapter 6, a design of optimal mapping in BICM-ID scheme to OS channel estimation and IC procedure system. The system can be regarded as two serially concatenated iterative decoding schemes whereby the inner decoder is replaced by the soft demapping device (also referred to as 'demapper') in the turbo demodulation, and by the detection symbols device in the turbo detection. In this chapter, the symbol can belong to a set of high level modulations (16-QAM) and the channel is considered like a time-invariant ISI channels. Based on the results presented in chapter 5, close to perfect known channel, our main objective is to outperform our first iteration, which leads to improve the performance when the channel is known, by optimization mapping and de-mapping. We design an optimized binary mapping of the QAM constellation, in order to increase coding gain, which assumes perfect a priori information**.** This expression is called *the genie method*. Like if we consider the presence of a genie delivering perfect a priori information. This optimized mapping can be used for some practically important cases.

In chapter 7, we introduce the LDPC codes and some of their properties, constructions and decoding algorithms. LDPC codes are binary linear block codes and, as their name indicates, have a parity check matrix that has only a small number of “1”s per row and per column. Thus, the parity check matrix has a low density of “1”s. We generalize the original idea of “dual APP” (DAPP) decoding to non binary alphabets in chapter 8. By non binary alphabet we mean that code symbols are taken from some extension field $GF(q=p^m)$, mostly binary field $GF(q=2^m)$.In this chapter, we address the non binary codes in conjunction with non binary modulation. We present a symbol-by-symbol *maximum a posteriori* (MAP) decoding rule which is, in a way, the dual of correlation-Viterbi decoding for linear codes. This code is also exhaustive, but in the sense that every word in the dual code is used in the decoding process. This means that in practice this decoding rule can be used only with codes whose dual code has a small number of code words, i.e., high rate codes or low-to-middle rate codes with short constraint lengths. In the same time, we describe how this complexity can be reduced using our decoding rules and make the computation faster with Fast Hadamard Transform (FHT) of the probabilities. Finally in this chapter, we present the Adaptive LDPC codes addressed for 3rd Generation & Beyond wireless systems like orthogonal frequency division multiplexing (OFDM). The use of Adaptive Coding is one of the key enabling techniques in the standards for 3rd-Generation (3G) wireless systems that have been developed to achieve high spectral efficiency. The core idea of Adaptive LDPC codes is to dynamically change the Galois field in a non binary LDPC codes by adapting a different Galois field on each carrier, the objective of adapting is to benefit of the overall spectral efficiency to the channel condition and to take advantage of the whole of the capacity by using the water filling technique. We consider that the characteristics of the channel are well-established and the channel can be probed to obtain a reliable channel quality estimate, the transmitter then uses this estimate to choose the appropriate signaling set. We compare two cases: In the first case, we consider at the transmitter a binary LDPC encoding and a modulator where the level size is adapted to the channel profile, and at the receiver we compute the probabilities over each bit which will be used in the binary LDPC decoder. In the second case, we consider at the transmitter an adapted LDPC encoding where the level size will be adapted in the encoding matrix and a modulator, and at the receiver we compute the probabilities over each symbol and use this in a type of adapting LDPC decoder. We will see that the decoding rule presented in section 8.1 can be easily applied to the non binary decoding and to the adaptive decoding and the performances obtained are very interesting.

In chapter 9, we focus on developing LDPC codes for channels with binary inputs and ISI memory. We are concerned with finding LDPC-turbo equalization which produces the lowest possible bit-error rate for a minimum amount of complexity.

Finally in Chapter 10, we summarize the contributions of this thesis and discuss future research directions.

Chapter 2
Information limits of communication channels

In the early 1940s, the general consensus was that increasing the rate of information transmitted over a communication channel increased the probability of error. However, in his 1948 paper “A Mathematical Theory of Communication,” Shannon quantified the maximum rate of information that can be sent error-free over a communication channel [45]. Shannon’s proof involved using random codes with exponentially small error probability for long block lengths, coupled with a nearest neighbour decoding rule. Unfortunately, such structureless codes are extremely difficult to decode in practice, generally requiring an exponentially large lookup table and/or computation. Ever since then, researchers have sought structured codes whose rates are close to theoretical limits and yet can be decoded simply. There are different information limits for different communication scenarios, and in this section we review the maximum possible information rates for various scenarios and how researchers have tried to achieve those rates with practical modulation schemes. In particular, we look at interference-free channels, interference channels when the transmitter has knowledge of the interference (also known as *channel state information*), and interference channels when the transmitter does not have such knowledge.

In this chapter, we describe a discrete-time baseband model for digital communication systems and define the associated detection problem. We present a broader perspective where we explain how detection can be combined with channel coding and decoding in different communication scenarios to reliably transmit information at rates close to the theoretical maximum.

2.1 AWGN Channel

Additive white Gaussian noise (AWGN) channels can be thought of as interference-free channels. The maximum information rate that can be sent over an AWGN channel or, more commonly, the *capacity* of the AWGN channel in bits per two dimensions is well known to be

$$C_{AWGN} = \log_2\left(1+\frac{\mathcal{E}_s}{N_0}\right). \tag{2.1}$$

Code design and practical decoding techniques for rates close to AWGN channel capacity are well understood today. In low SNR regimes where low-rate codes suffice, turbo codes [1] and low-density parity-check (LDPC) codes [37, 38] approach theoretical limits and are practically decodable with “message-passing” algorithms like those discussed in Chapter 4. In [34], it was demonstrated that an LDPC code can operate at an SNR that is 0.0045 dB away from the minimum SNR necessary for codes of that rate. In high SNR regimes where high-rate codes are required, multilevel codes with multistage decoding [41, 42] can be used to approach capacity.

2.2 Interference Channel Model

In this thesis, we focus on discrete-time baseband channel models, which abstract the channel impairments and hide the specific implementational details of the digital communication system. In doing so, we can talk about different digital communication systems with different kinds of channel interference in one common signal space framework.

Let us now describe the channel model that we use in this thesis. The $N \times 1$ vector X contains the data to be transported over the channel, and is chosen from a finite equiprobable set. Depending on the underlying communication system, the components of X may correspond either to distinct time instants, distinct carrier frequencies, distinct physical locations, etc. The channel interference is modelled as linear interference, which is represented by multiplication of X with a $Q \times N$ matrix H. With channel noise being composed of the superposition of many independent actions, the central limit theorem suggests that we can model the noise as a zero-mean, complex-valued, additive white Gaussian noise (AWGN) vector W with circularly symmetric components of variance N_0. The $Q\times1$ vector R that is obtained at the receiver is thus as illustrated in Fig. 2.1.

$$R = HX + W, \tag{2.2}$$

In this thesis, we are primarily concerned with detection at the receiver of the transmit vector X based on knowledge of R, H, and the statistics of W. The parameters of H can be learned at the receiver via techniques collectively known as *training*, in which H is estimated by sending vectors jointly known to the transmitter and receiver across the channel. If the channel changes with time, then the estimate of H can be updated using the detection decisions.

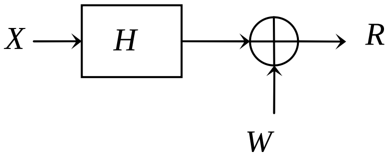


Figure 2.1: The Vector model.

Sometimes it is also useful to periodically perform training in case tracking becomes unsuccessful. In any event, we assume in most of the thesis that H and the statistics of W are explicitly known at the receiver.

Though we focus specifically on applications of the vector detection model (2.2) to digital communication systems, the detection schemes we develop in this thesis are applicable to any scenario in which (2.2) applies. We now complete this section with a few applications in digital communication.

One example of a communication system in which the channel model (2.2) applies is the uplink scenario of a N-user discrete-time synchronous code-division multiple-access (CDMA) system, shown in Fig. 2.2. In this system, the ith user modulates a complex symbol x_i onto a signature sequence $h_i[k]$ of length Q assigned to that user. The modulated signature sequence is sent across the channel, where it encounters channel attenuation by a factor of A_i. The base station receives the superposition of all the users' signals in noise as described by (2.2), where the columns of H are the users' signatures scaled by the corresponding channel attenuation factors; i.e.,

$$\begin{bmatrix} r[0] \\ r[1] \\ \vdots \\ r[Q-1] \end{bmatrix} = \begin{bmatrix} A_0h_0[0] & A_1h_1[0] & \cdots & A_{N-1}h_{N-1}[0] \\ A_0h_0[1] & A_1h_1[1] & \cdots & A_{N-1}h_{N-1}[1] \\ \vdots & \vdots & \ddots & \vdots \\ A_0h_0[Q-1] & A_1h_1[Q-1] & \cdots & A_{N-1}h_{N-1}[Q-1] \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[Q-1] \end{bmatrix} \tag{2.3}$$

Another example in which (2.2) applies is a discrete-time synchronous multiple antenna system with N transmits antennas and Q receives antennas. As shown in Fig. 2.3, each transmit antenna sends a different complex symbol x_i over the channel. For narrowband transmission, the path from the ith transmit antenna to the jth receive antenna is described by a single flat fading coefficient, h_{ji}, the set of which can be assembled into the matrix H. Each receive antenna receives a superposition of signals from all transmit antennas in white noise, so again we encounter the model in (2.2):

$$\begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{Q-1} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0,N-1} \\ h_{10} & h_{11} & \cdots & h_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{Q-1,0} & h_{Q-1,1} & \cdots & h_{Q-1,N-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{Q-1} \end{bmatrix} \tag{2.4}$$

A discrete-time point-to-point channel with inter-symbol interference (ISI), depicted in Fig. 2.4, can also be modelled using (2.2). The transmitted data is a stream of complex symbols x_i, which are corrupted by a convolution with the impulse response of the ISI channel, h_i, and by additive noise, w_i, to produce the received symbols

$$r_i = \sum_k h_{i-k}x_k + w_i. \tag{2.5}$$

If r_i, x_i, and w_i are each arranged in vector format, and $H = [h_1, \ldots, h_P]$ with h_{i-k} being a time-delayed version of h_i arranged in vector format, then we again obtain the model (2.2). An example with an impulse response of length two is

$$\begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N-1} \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_0 & 0 \\ 0 & \cdots & 0 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{bmatrix} \tag{2.6}$$

Note, that the H matrix in this case is square and Toeplitz.

2.2.1 Interference Channels with Water Pouring

In communication systems where the interference matrix H is non-trivial and known at the transmitter, to support the maximum information rate it is necessary to optimally allocate transmit power amongst the various components of X so that the power transmitted over the kth mode of the channel is given by the "water-pouring" formula [43]

$$\varepsilon_{s,k} = \max\left(L - \frac{N_0}{\lambda_k}, 0 \right) \tag{2.7}$$

where L is chosen such that the average transmit energy is

$$\varepsilon_s = \frac{1}{N} \sum_{k=0}^{N-1} \varepsilon_{s,k} \tag{2.8}$$

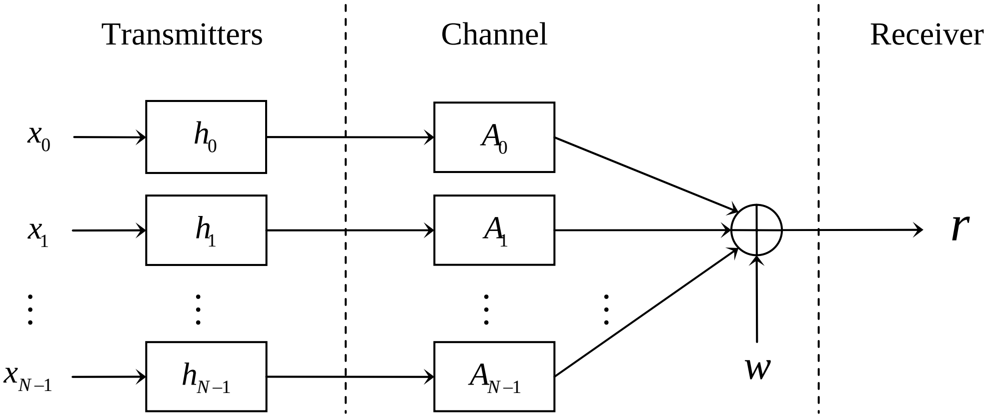


Figure 2.2: Uplink scenario of CDMA system.

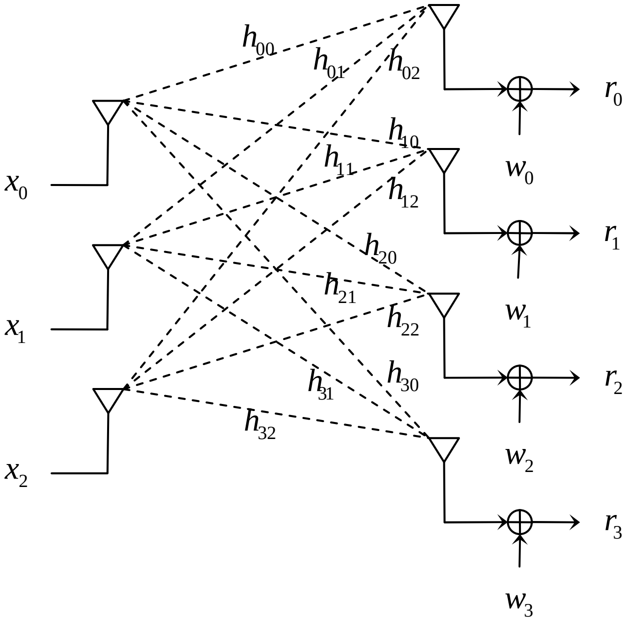


Figure 2.3: Multiple antenna system.

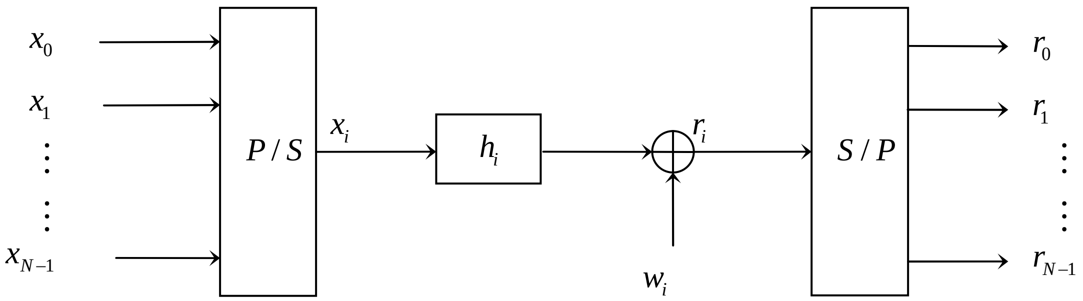


Figure 2.4: ISI channel.

The idea of water pouring is to transmit more power over large modes, and to transmit less power over small modes. In fact, modes that are too small are allocated no transmit power. If *K* is the set of modes allotted transmit power via water pouring, then the capacity of the interference channel in bits per two dimensions is

$$C_{\text{int}} = \frac{K}{N}\log_2\left(\frac{\frac{\varepsilon_s}{N_0}\frac{N}{K}+\left\langle\frac{1}{\lambda_k}\right\rangle_{A,K}}{\left\langle\frac{1}{\lambda_k}\right\rangle_{G,K}}\right) \tag{2.9}$$

Where $\left\langle\frac{1}{\lambda_k}\right\rangle_{A,K}$ and $\left\langle\frac{1}{\lambda_k}\right\rangle_{G,K}$ are, respectively, the arithmetic and geometric means of $\frac{1}{\lambda_k}$ over *K*, given by

$$\left\langle\frac{1}{\lambda_k}\right\rangle_{A,K} = \frac{1}{|K|}\sum_{k\in K}\frac{1}{\lambda_k} \tag{2.10}$$

$$\log\left\langle\frac{1}{\lambda_k}\right\rangle_{G,K} = \frac{1}{|K|}\sum_{k\in K}\log\frac{1}{\lambda_k} \tag{2.11}$$

Techniques are known today to approach the capacity of interference channels as closely as the capacity of interference-free channels, provided the transmitter is aware of the interference. A communication system that incorporates such techniques is shown in Fig. 2.6. At the transmitter, knowledge of channel state information is used in an interference “pre-canceller” that optimally allocates transmit power to the channel modes. The cascade of the transmitter pre-canceller and the channel appears interference-free, so the coding and decoding techniques for AWGN channels can be exploited to approach capacity. There are two main classes of such techniques. One class treats all the modes as belonging to a single channel [33, 40, 47], while the other class partitions the underlying channel into parallel independent sub-channels, over which symbols are transmitted according to the water-pouring power allocation [44].

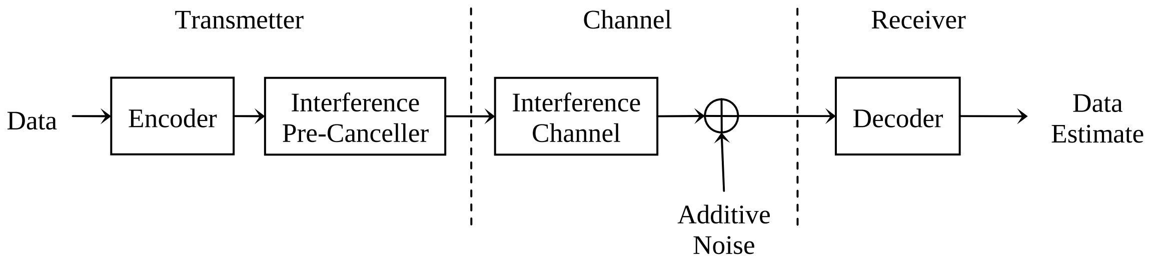


Figure 2.5: A communication system that pre-cancels channel interference.

For ISI channels, an example in the former class is Tomlinson-Harashima precoding, and an example in the latter is discrete multi-tone (DMT).

2.2.2 Interference Channels without Water Pouring

When the transmitter has no channel state information, water pouring cannot be done to achieve the capacity of the interference channel. Rather, the transmit power is evenly distributed across all the modes of the system, leading to a maximum rate of bits per two dimensions.

$$I_{\text{int}} = \frac{1}{N}\sum_{k=0}^{N-1}\log_2\left(1+\frac{\varepsilon_s\lambda_k}{N_0}\right) \tag{2.12}$$

We call this quantity the mutual information of the channel rather than the capacity because the information rate is not optimized at the transmitter via water pouring.

In this scenario, the communication system depicted in Fig. 2.5 can no longer be used, and creating parallel; rather, interference must be dealt with at the receiver. The optimal receiver in a probability of error sense uses a maximum-likelihood (ML) or maximum *a posteriori* (MAP) algorithm that treats the encoder and the interference channel as a single product code and performs joint detection and decoding, shown in Fig. 2.6. Though jointly optimal, the complexity of such a system is usually determined by the product of the complexities of the optimal detector for the corresponding uncoded system and the optimal decoder for the corresponding AWGN channel. Thus, the complexity of such a system is prohibitive.

A classical suboptimal solution is to separate the problems of detection and decoding as shown in Fig. 2.7. The detector should be designed so that the cascade of the interference channel and the detector appears like an AWGN channel, so the complexity of the detector is no different than for uncoded data.

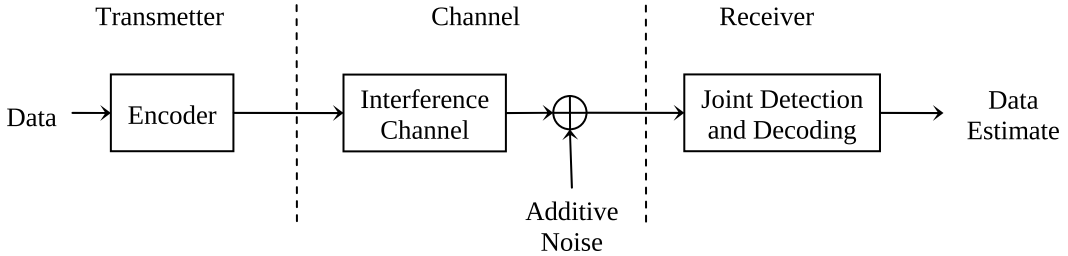


Figure 2.6: Optimal joint detection and decoding.

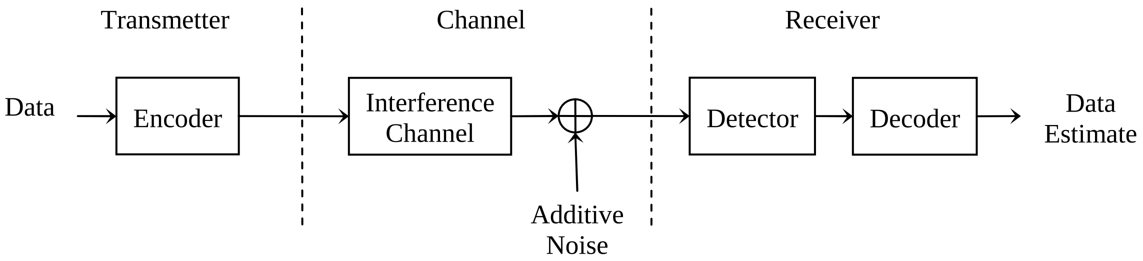


Figure 2.7: Classic suboptimal separation of detection and decoding.

However, it is more difficult to make the cascade of the channel and the detector look like an AWGN channel than it is for the cascade of a transmitter pre-canceller and the channel to look like one, because noise from the channel can be amplified or *enhanced* by the detector. Nevertheless, assuming that the detector is designed well, coding and decoding schemes for AWGN channels can be used, so the complexity of the decoder is the same as if there were no interference in the channel. The result is that the overall complexity is the sum, rather than the product, of the complexities of the individual components.

It is this scenario of the three for which techniques to approach the theoretical limits are least developed. This thesis presents practical schemes with low complexity to approach the mutual information of an interference channel without channel knowledge at the transmitter. Before we review some classical detectors, let us gain some insight by comparing the theoretical information limits of the three communication scenarios presented.

2.2.3 A Comparison of Maximum Information Rates

We now compare the maximum achievable rates of different communication scenarios at the same received SNR, as defined in (2.6). To facilitate the comparison, we assume that $\|H\|_F^2 = \sum_{k=1}^{N} \lambda_k$ is normalized to *N*, the number of columns in *H*, so that the symbol energy ε_s and the noise variance N_0 are also fixed. When the transmit power is equally distributed amongst the modes, the interference channel cannot have a mutual information greater than the corresponding AWGN channel with equivalent SNR. We can see this from the concavity of the log function:

$$I_{\text{int}} = \frac{1}{N}\sum_{k=1}^{N}\log_2\left(1+\frac{\varepsilon_s\lambda_k}{N_0}\right) \le \log_2\left(1+\frac{\varepsilon_s}{N_0}\right) = C_{AWGN} \tag{2.13}$$

"where we have used the normalization of $\|H\|_F^2$, and the rates are per two dimensions. At low SNR, however, we can show that $I_{\text{int}} \approx C_{AWGN}$ using the approximation $\ln(1+\alpha) \approx \alpha$:"

$$I_{\text{int}} = \frac{1}{N}\sum_{k=1}^{N}\log_2\left(1+\frac{\varepsilon_s\lambda_k}{N_0}\right) \approx \frac{1}{N}\sum_{k=1}^{N}\frac{\varepsilon_s\lambda_k}{N_0\ln 2} = \frac{\varepsilon_s}{N_0\ln 2} \approx \log_2\left(1+\frac{\varepsilon_s}{N_0}\right) = C_{AWGN} \tag{2.14}$$

At high SNR, the rate loss from C_{int} to I_{int} due to the absence of water pouring is negligible because the capacity-achieving allotment of transmit power to the various modes is asymptotically equal. The term “high SNR” is relative in this context, because not only does the total transmit power need to be high, but the amount allocated to each mode must be high as well.

Thus, for C_{int} and I_{int} to be approximately equal, a channel with a few small modes requires a higher SNR than a channel with no small modes. At low SNR, however, the capacity C_{int} of an interference channel can substantially exceed both the mutual information without water pouring I_{int} and the capacity of the AWGN channel C_{AWGN} even with the normalization of $\|H\|_F^2$. Intuitively, transmit power is selectively loaded onto channel modes that can support the highest rates.

Figure 2.8 compares C_{int} and I_{int} for the three-tap ISI channel with impulse response to C_{AWGN}.

$$h_i = 0.5\delta_i + 0.707\delta_{i-1} + 0.5\delta_{i-2} \tag{2.15}$$

As noted earlier, the capacity of an ISI channel exceeds that of the corresponding AWGN channel at low SNR, since transmit power can be loaded onto favourable frequencies. If water pouring is not available, then at low SNR the mutual information of the random ISI channel approaches the capacity of the corresponding AWGN channel. At high SNR, the capacity of the ISI channel becomes less than the AWGN channel capacity, and also the effect of water pouring becomes negligible. The asymptotic slopes of all three curves are equal, implying that the penalty of an interference channel is only a fixed rate loss.

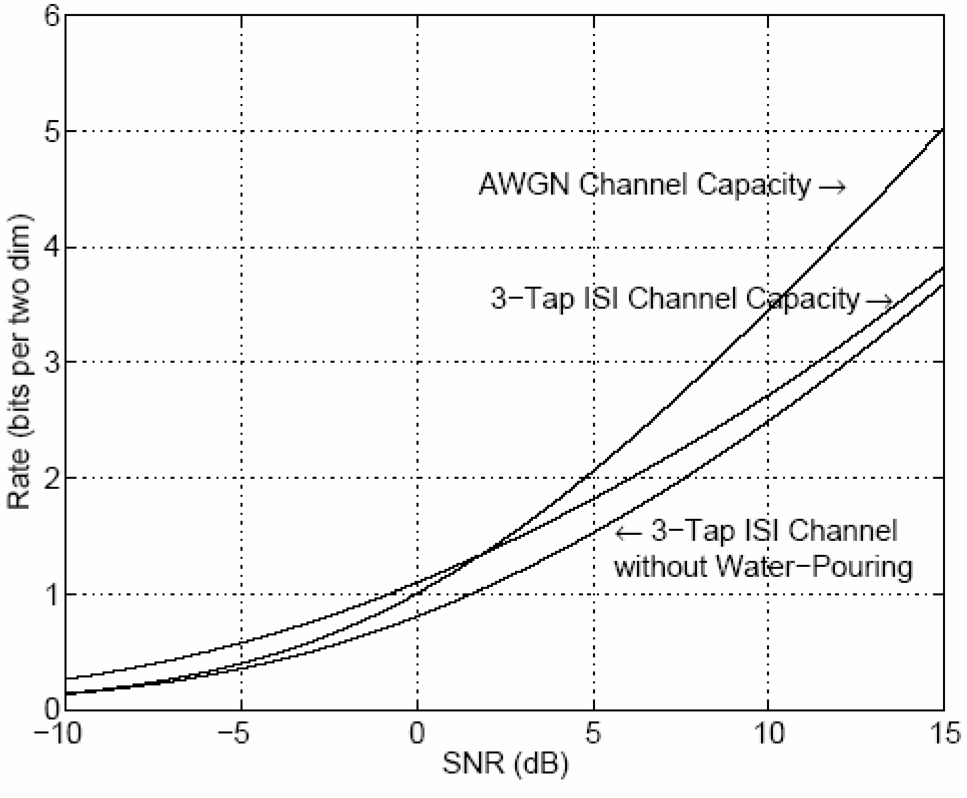


Figure 2.8: Information rates for the three-tap ISI channel of (2.15).

Chapter 3
Classical Detection Schemes

There are a variety of detectors that can be used in the scenario of Fig. 2.7, and in this section we discuss some of the most common. We begin with the maximum-likelihood (ML) detector, which minimizes the probability of vector detection error and can be considered optimal. The rest of the detectors presented in this section are designed to offer an approximate solution to ML detection with lower complexity. While the extent to which these suboptimal detectors trade off accuracy with complexity varies, they all share the idea that the effect of *H* should be explicitly “cancelled” or “undone” so that the decision device can treat the channel as an AWGN channel. In the special case of an uncoded system, the detection problem can be decoupled and processed by a simple symbol-by-symbol decision device.

Before we review these classical detectors, it is useful to establish a way by which we can compare detector performance. Although practical communication systems typically use coded transmission, we will compare detectors based on their performance for uncoded transmission, i.e., each component of *X* contains independent data and is equiprobably chosen from a finite set χ , and so $X \in \chi^N$. Since detectors typically treat coded data as if it were uncoded, focusing on uncoded systems allows us to isolate the quality of the detector from the quality of the decoder.

The bit error rate for a coded system, defined as the probability that a component of $\hat{X}$ is not equal to the corresponding component of $X \in \chi^N$, is the measure that we use in this thesis when specifically talking about detector performance. We use this metric for several reasons. First, the ML detector, which minimizes the probability of vector detection error, has a bit error rate that is almost identical at high signal-to-noise ratio (SNR) to the detector that minimizes bit error probability [10]. Thus asymptotically, the ML detector provides a useful lower bound to the bit error rate of any detector. Second, looking at bit errors is equivalent to computing the Hamming distance between x and $\hat{x}$, which gives a measure of closeness of the detected vector to the transmitted vector. Since the ML vector is closest to *X* on average, we can infer how close the solutions provided by other detectors are to the optimal solution. Third, since suboptimal detectors typically attempt to decouple the detection of coded symbols by using a symbol-by-symbol minimum-distance decision device or *slicer*, using the bit error rate is intuitive. Fourth, codes are often characterized by the number of bit errors in a codeword that can be corrected. In a system that performs detection followed by decoding, a low bit error rate after detection is desirable for good overall performance of the system.

In this chapter, we present and remind some classical detection schemes which have be presented in the literature and some books but here we grouped them in a single chapter to facilitate the introduction of turbo equalization done in the next chapter.

3.1 ML Detection

Since all vectors x are equally likely, the detector that minimizes the probability of vector detection error is the ML detector:

$$\hat{X} = \arg\max_{X\in\chi^N} f\left(R|X\right). \tag{3.1}$$

Since the noise is independent of X, uncorrelated and Gaussian, (3.1) simplifies to the minimum-distance rule depicted in Figure 3.1.

$$\hat{X} = \arg\min_{X\in\chi^N} \left\|R - HX\right\|, \tag{3.2}$$

Thus, the detector computes the most likely vector X based on knowledge of R, H, and the distribution of W. The set of all possible uncoded vectors χ^N can be represented in N-dimensional Euclidean space as the points of a (shifted) orthogonal lattice bounded within an N- dimensional cube, The set of possible vectors is depicted in Fig. 3.2(a) for $N = 2$.

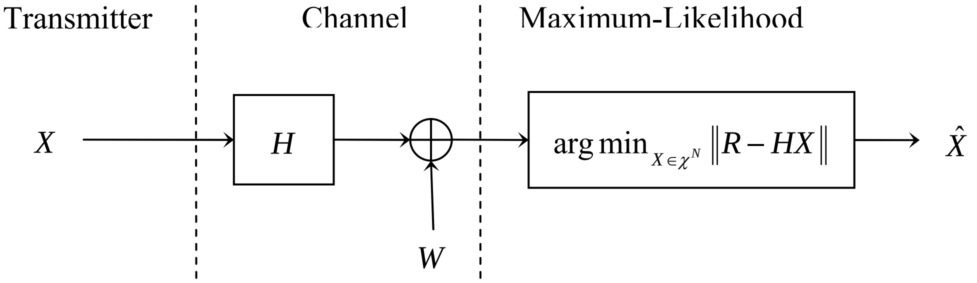


Figure 3.1: Maximum-Likelihood (ML) detection.

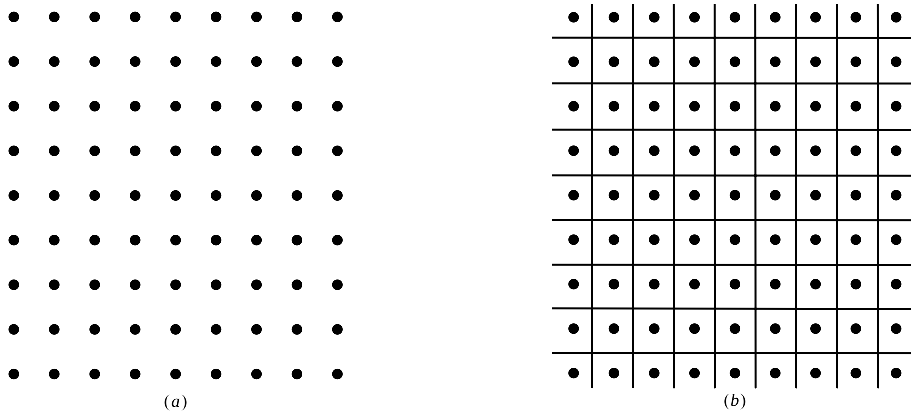


Figure 3.2: (a) Bounded lattice representing the uncoded set of vectors χ^2.
(b) Corresponding decision regions for the AWG channel.

In the special case of an AWGN channel, our model becomes $R = X + W$, and so R is a noise-perturbed version of X. The minimum-distance rule (1.18) simplifies to

$$\hat{X} = \arg\min_{X\in\chi^N} \left\|R - X\right\|. \tag{3.3}$$

Since each component of the uncoded vector *X* affects only the corresponding component of *R*, and since the noise vector is uncorrelated, the ML detector can be decoupled into a set of symbol-by-symbol optimizations; i.e., which can be solved using a symbol-by-symbol minimum-distance decision device or *slicer*.

$$\hat{x}_i = \arg\min_{x_i \in \chi} \left\| r_i - x_i \right\| \quad for\ i = 0,1,\cdots,N-1, \tag{3.4}$$

The decision regions, corresponding to the values of *R* for which each of the possible decisions is made, are depicted in Fig. 3.2(b). The ability to decouple the ML detector into component wise minimizations is indicated by the fact that the boundaries of the decision regions form an orthogonal grid. The minimization for each of the *N* components of *X* requires the comparison of $|\chi|$ differences, so complexity is linear in *N*.

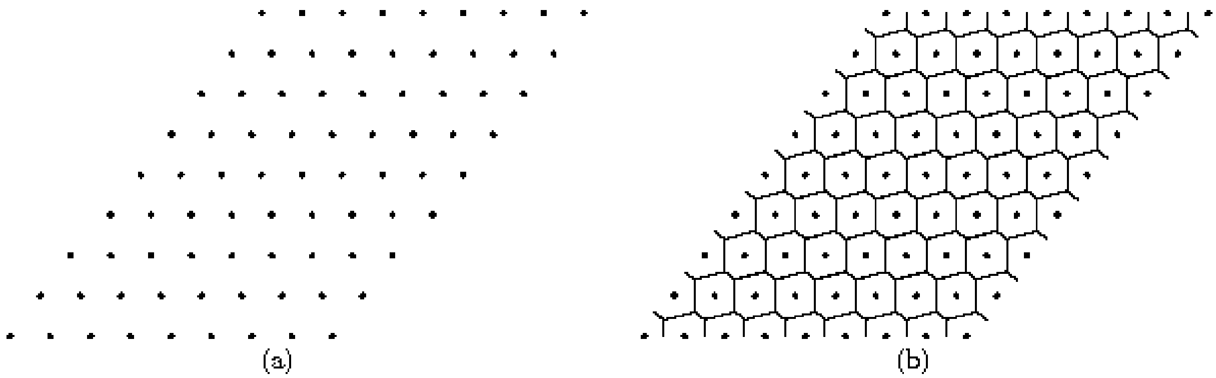


Figure 3.3: (a) Bounded lattice representing all possible vectors *Hx* for an interference channel. (b) Corresponding decision regions

In the general case in which linear interference is present, we have that *R* = *HX*+*W*, and the ML vector detector of (3.2) generally cannot be decomposed into *N* smaller problems. We can see this by first recognizing that the action of **H** on the set of all possible uncoded vectors $X \in \chi^N$ is to map the points of the bounded orthogonal lattice in Fig. 3.2(a) to the points of a bounded lattice with generators along the directions of the columns of **H**, like the bounded lattice in Fig. 3.3(a). The decision regions of (3.2) are now generally polytopes as shown in Fig. 3.3(b), and decoupling of the problem is no longer possible.

The minimization of (3.2) requires the comparison of $|\chi|^N$ differences, so complexity is exponential in *N*. In fact, the least-squares integer program in (3.2) for general **H** matrices has been shown to be nondeterministic polynomial-time hard (NP-hard) [50]. In the ISI channel case, ML detection can be performed using a dynamic programming algorithm known as the Viterbi algorithm [10], which has complexity proportional to $|\chi|^L$ where *L* is length of the channel impulse response. If *L* or $|\chi|$ is large, then ML detection is still complex.

The high complexity of the ML detector has invariably precluded its use in practice, so lower-complexity detectors that provide approximate solutions to (3.2) are used, which we review in the next section.

3.2 Linear Detection

As depicted in Fig. 3.4, linear detectors take the received vector R and pre-multiply it by a matrix $B^{\dagger}$. The resulting product, $\tilde{X}$, is passed to a minimum-distance symbol-by-symbol slicer to produce $\hat{x}$. The matrix B can be optimized using different criteria, but two of the most popular are the zero-forcing (ZF) criterion and the minimum mean-squared error (MMSE) criterion.

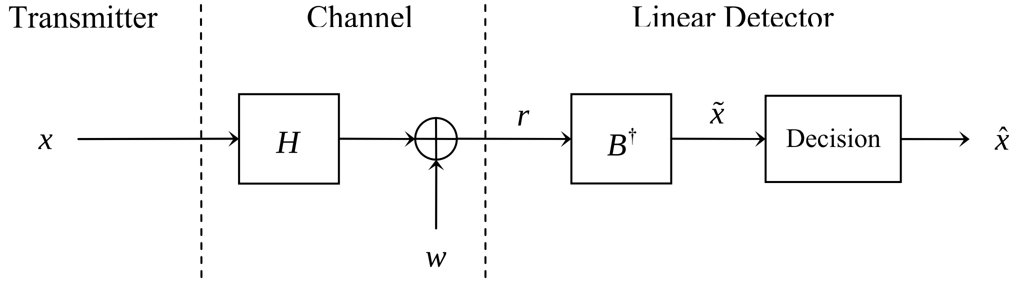


Figure 3.4: Linear Detection.

The ZF criterion, also known as the interference-nulling criterion, chooses B to completely eliminate interference in $\tilde{X}$, while the MMSE criterion chooses B to minimize the variance of $\tilde{X} - X$. The main disadvantage of this low-complexity class of detectors is that they perform quite poorly in a symbol-error rate sense because the matrix $B^{\dagger}$ enhances the variance of the noise components in W and also makes the difference vector $\tilde{X} - X$ correlated. These problems are more severe when the ZF criterion is used, but are present nevertheless for the entire class of detectors.

Shnidman [46] considered the problem of eliminating interference in a multi-user context very early on. The MMSE linear detector for CDMA systems was first described by Xie et al. [51] and Madhow and Honig [43]. Tufts [48] derived the ZF and MMSE linear detectors for the ISI channel case, known in that context as linear *equalizers*.

3.2.1 Whitening Matched Filter

The conventional data estimator consists of a bank of matched filters. In the case of correlated noise, the conventional data estimator is extended by a pre-whitening filter. We will discuss in this section about whitening matched filter (WMF), i.e., a combination of pre-whitening and matched filters followed by symbol rate samplers. Although the WMF treats ISI and MAI as noise, it is introduced here, because all techniques presented in the following can be interpreted as an extension of the WMF, where the noise is de-correlated or pre-whitened, and then the pre-whitened signal is fed into filters matched to the response of the pre-whitening filter to the sequence. The structure of the WMF is shown in Fig. 3.5.

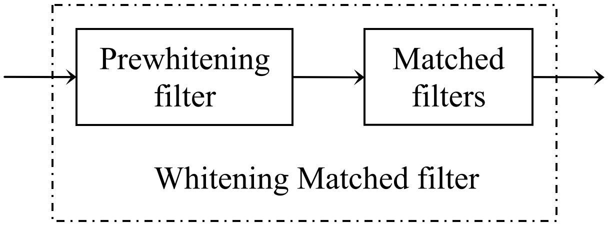


Figure 3.5: Structure of WMF.

Noise whitening followed by matched filtering is only close to the optimum if the ISI and MAI components are negligible, which is the case for large spreading factors and orthogonal signature sequences, this implies low spectral efficiency.

In the following, detection strategies are presented that lead to considerable performance improvements over the WMF and the conventional data estimator.

3.2.2 Zero-Forcing Block Linear Equalizer

The zero-forcing block linear equalizer (ZF-BLE) investigated in past research and minimizing the quadratic form leads to a continuous valued unbiased estimate.

The equalizer leading to the estimate is termed zero forcing since it totally eliminates ISI and MAI, irrespective of the noise level. The ZF-BLE can be interpreted as an extension of the whitening matched filter described in section 3.2.1 by a whitening filter and an ISI and MAI eliminator. If no ISI is present, the ZF-BLE turns into the de-correlating detector. The structure of the ZF-BLE is shown in Fig. 3.6.

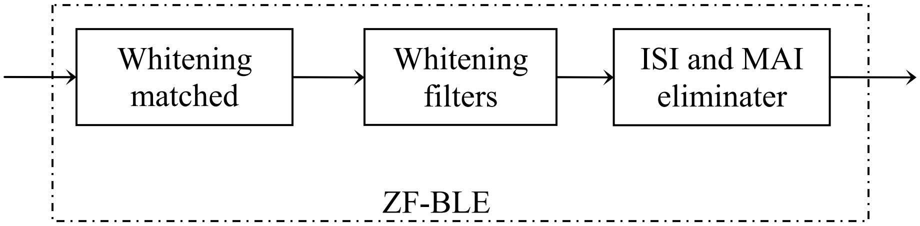


Figure 3.6: Structure of the ZF-BLE.

3.2.3 Minimum-Mean-Square Error Block Linear Equalizer

The minimum mean square error block linear equalizer (MMSE-BLE) minimizes the quadratic error and leads to a continuous valued estimate, the MMSE-BLE can be interpreted as an extension of the ZF-BLE described in section 3.2.2 by a wiener estimator, which reduces the performance degradation of the ZF-BLE resulting from the fact that the decisions do not take into account the noise correlations existing in the decision variables. The term of Wiener estimator observe the ZF-BLE output, it produces an MMSE estimate.

By the factor of Wiener, the desired symbols on the one hand and the ISI, MAI, and noise terms on the other hand, are de-correlated. The structure of the MMSE-BLE is depicted in Fig. 3.7.

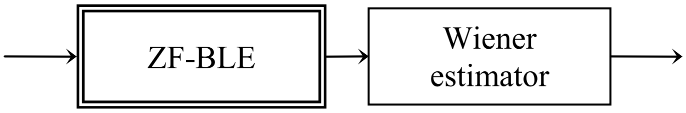


Figure 3.7: Structure of the MMSE-BLE.

For $\sigma^2 \to 0$, the MMSE-BLE approaches the ZF-BLE, and for $\sigma^2 \to \infty$, the MMSE-BLE approaches the WMF.

3.3 Decision-Feedback Detection

The decision-feedback detector builds upon the linear detector by combining it with a nonlinear feedback loop, pictured in Fig. 3.8. As before, the received vector R is pre-multiplied by $B^\dagger$, but rather than making minimum-distance symbol-by-symbol slicer decisions on the entire output vector, decisions are made sequentially, one component at a time. To begin, the first component of $B^\dagger R$, denoted $\tilde{x}_1$, is processed by the slicer to produce the detected symbol $\hat{x}_1$. Assuming that $\hat{x}_1$ is equal to x_1, the feedback loop is used to subtract off the interference caused by x_1 from the remaining components of $B^\dagger R$. The second component of the resulting interference-reduced vector, denoted $\tilde{x}_2$, is then processed by the slicer to produce $\hat{x}_2$. Assuming that $\hat{x}_2$ is a correct decision, the interference caused by x_2 is subtracted from the remaining components of $B^\dagger R$, and the process continues until decisions all the components have been made.

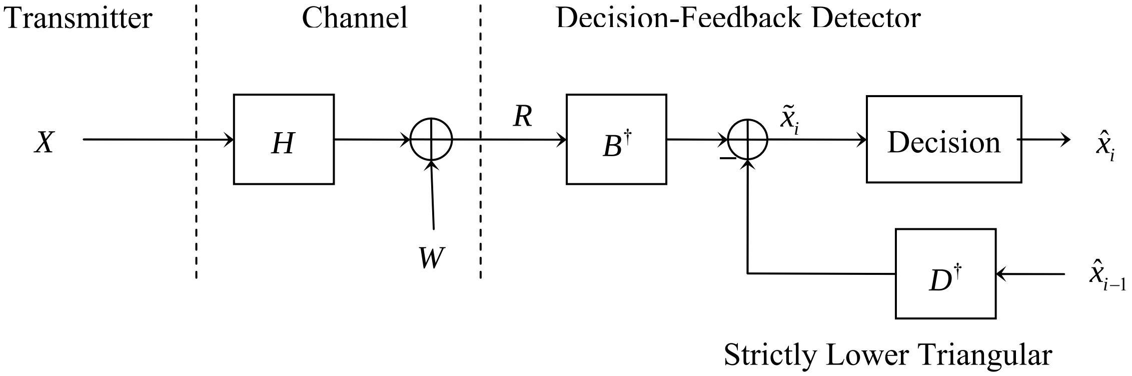


Figure 3.8: Decision Feedback detection.

As with the linear detector, the ZF and MMSE criteria are popular. Though decision-feedback detectors usually perform better than linear detectors, they still have some serious shortcomings. First, noise enhancement is still an issue, though the situation is not as bad as with linear detection. Second, decisions are made sequentially at the slicer and so are used to improve only future decisions, not past ones. Third, the sequential nature of the decision device means that in practice, incorrect decisions can lead to further incorrect decisions, a phenomenon known as *error propagation*. Moreover, since the matrices B and D used in decision-feedback detection are often optimized under the faulty assumption of no error propagation, there may be some mismatch between the desired optimization criterion and the matrices that are used. Fourth, the sequential structure of the decision-feedback detector makes it essentially incompatible for use with ISI channels in conjunction with channel coding (on channels not known at the transmitter, as is the case of interest in this thesis). As a result, use of the decision-feedback equalizer has been largely restricted to uncoded systems.

The idea of feeding back decisions to mitigate the effects of interference for future symbols was first used by Austin [32] in the context of ISI channels. Duel-Hallen [35] introduced the idea to CDMA systems, while Foschini [36] brought the idea to multiple antenna systems via the Bell Labs Layered Space-Time (BLAST) detection algorithm.

3.3.1 Zero-Forcing Block Decision Feedback Equalizer

For the description of the zero-forcing block decision feedback equalizer (ZF-BDFE), the modified received sequence is introduced [31]. Then the decisions are made according to the recursive formula. In the case of coded transmission, it would possible to generate soft inputs for the decoder by using the quantized estimates for decision feedback and the continuous valued estimates before quantization for decoding, possibly together with some channel state information. Subtracting all past decisions from the new estimate, and the estimation can be converging if all past decisions are near to right solution. The structure of the ZF-BDFE is depicted in Fig. 3.9. It is shown that the ZF-BDFE is equivalent to a noise cancelling detector. If continuous valued estimates of u_k, are fed back, which are obtained by omitting threshold detection, the ZF-BDFE turns into the ZF-BLE.

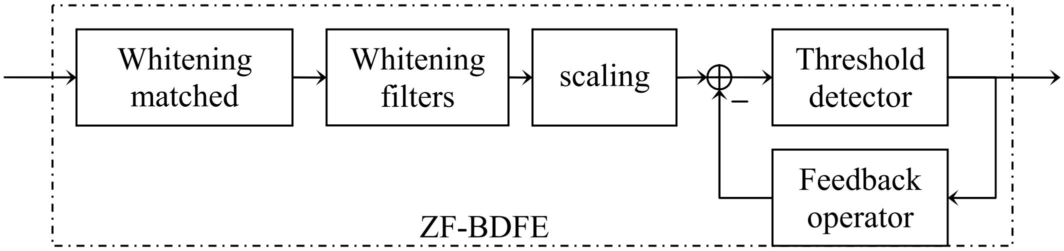


Figure 3.9: Structure of the ZF-BDFE.

3.3.2 Minimum-Mean-Square Error Block Decision Feedback Equalizer

For the description of the minimum mean square error block decision feedback equalizer (MMSE-BDFE), the Cholesky decomposition is defined in [31]. The way of deciding and subtracting past decisions is the same as described in section III.3.1. The structure of the MMSE-BDFE is the same as that of the ZF-BDFE in Fig. 3.9 if the matrix of the whitening filter is changed.

3.4 Multistage Detection

In an attempt to symmetrize the problem of being able to cancel only future symbols in decision-feedback detectors, multistage detectors process the received vector in block iterations. In some sense, multistage detectors can be thought of as parallel processors, whereas decision-feedback detectors are sequential processors. An example of a multistage detector is shown in Fig. 3.10. During the first iteration, the vector R is pre-multiplied by the matched filter $H^{\dagger}$ to produce $\tilde{X}^{(1)}$, which is then sent to the slicer to generate a first set of tentative decisions $\hat{X}^{(1)}$ on *all* the symbols. During the second iteration, the vector R is again pre-multiplied by the matched filter $H^{\dagger}$, but before the result is sent to the slicer, an exact replica of the interference is created and subtracted off assuming $\hat{X}^{(1)}$ is a correct set of decisions. The slicer then takes the resulting vector $\tilde{X}^{(2)}$, and generates a second set of tentative decisions $\hat{X}^{(2)}$. Further sets of tentative decisions $\tilde{X}^{(l)}$ are generated in the same manner, using the tentative decisions of the previous iteration $\tilde{X}^{(l-1)}$ to subtract off interference. After a sufficient number of iterations, the most recent tentative decisions are taken as the final decisions. The two matrices in this example are fixed during each iteration, and are optimized to maximize the signal-to-interference + noise ratio (SINR) at the slicer input assuming correct tentative decisions. In general, the class of multistage detectors includes detectors with the same structure as in Fig. 3.10 but with alternative pairs of matrices that may change with each iteration. The problem with multistage detectors is that the decisions typically do not converge to the optimum ones, and limit cycles and divergence are possible. The reason for the poor performance of the multistage detector in Fig. 3.10 is that, like decision-feedback detectors, the two matrices are optimized assuming correct decisions are made at each iteration. Using this faulty assumption causes the incorrect quantities to be subtracted off as interference, which leads to the propagation of errors from iteration to iteration.

Gersho and Lim [39] developed multistage detectors for the ISI channel case, and Varanasi and Aazhang [49] later introduced them for the CDMA system case.

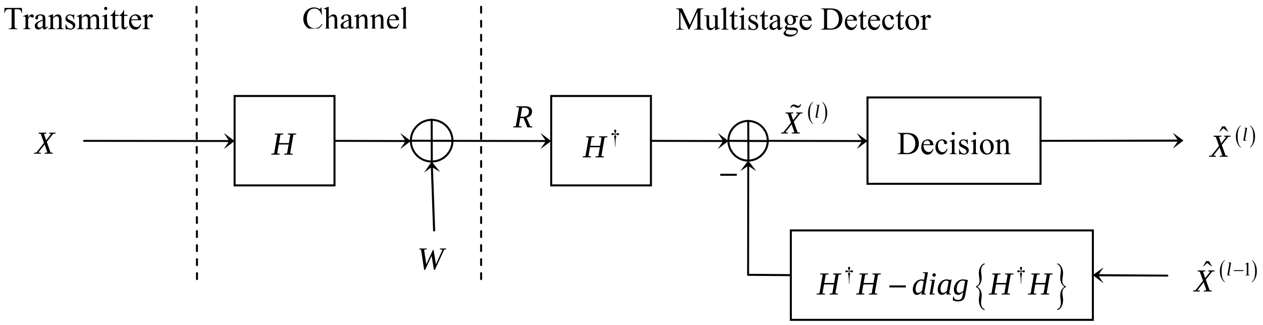


Figure 3.10: A multistage detector.

3.4.1 Interference Canceller

In order to reduce the turbo equalization complexity, the MAP detector can be advantageously replaced by an ISI canceller. This new receiver makes it possible to almost completely overcome ISI over time invariant and/or time varying Rayleigh channels for high spectral efficiency modulations. Fig. 3.11 depicts the interference canceller structure, consisting of two transversal filters fed by the received samples and the data, respectively, estimated from the previous iteration.

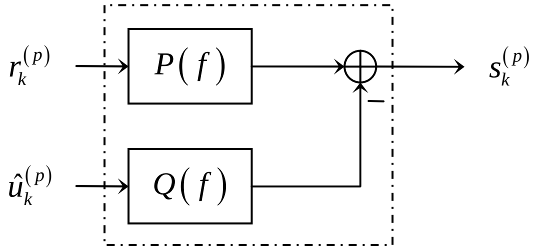


Figure 3.11: Interference Cancellation Structure.

For each iteration, the interference canceller is updated according to a criterion; we will describe here the optimal IC and adaptive IC.

3.4.2 Mean square error

The mean square error (MSE) criterion is defined as MSE = E{|s_k-u_k|^2}, where the superscript p has been dropped for convenience.

In order to determine the optimum IC, it is necessary to assume that the symbols $\left(\hat{d}_k = d_k\right)$ are known. With the constraint for the central coefficient of the filter Q(f) having to be equal to zero (q_o = 0), it can be shown that the IC optimum filters have the following transfer function:

$$P(f)_{optimal} = \beta\frac{H^*(f)}{hh_0} = \sum_{l=0}^{L} p_l e^{-j2\pi flT} \tag{3.5}$$

$$Q(f)_{optimal} = \beta\left(\frac{|H(f)|^2}{hh_0} - 1\right) = \sum_{l=-L}^{L} q_l e^{-j2\pi flT} \tag{3.6}$$

Where H(f) is the transfer function of the channel, hh_o the auto-correlation channel function, β the weighting coefficient, is equal to

$$\beta = \frac{\sigma_u^2 hh_0}{\left(\sigma_u^2 hh_0 + \sigma_w^2\right)} \tag{3.7}$$

And p_l and q_l are the coefficients of filters P(f) and Q(f), respectively.

Thus, the IC output is ISI-free and equal to

$$s_k = \beta\left(u_k + \frac{1}{hh_0}\sum_{l=0}^{L} h_l^* w_{k+l}\right) \tag{3.8}$$

With the output MSE given by

$$MSE_{optimal} = \frac{\sigma_u^2\sigma_w^2}{\left(\sigma_u^2 hh_0 + \sigma_w^2\right)} = 1 - \beta \tag{3.9}$$

And the SNR at the IC output is equal to

$$SNR_{optimal} = \frac{\sigma_u^2 hh_0}{\sigma_w^2} \tag{3.10}$$

It clearly that ISI is completely removed by the IC, without noise enhancement.

Transmitted symbols u_k are generally unknown by the receiver and the IC is sub-optimum because Q(f) is fed in by estimated symbols instead of transmitted symbols. This sub-optimality is taken into account in an adaptive way to adjust the IC coefficients. So, at the first iteration (p = 1), the estimated symbols are equal to zero and the IC approximates the MMSE linear equalizer. After several iterations (p > 1), the probability of the estimated symbols is expected to be right and the adaptive IC is close to the optimum IC defined by the transfer function (3.5) and (3.6).

3.4.3 Adaptive Interference cancellation

Adaptive algorithms such as least mean square (LMS) or recursive least square (RLS) can be used for updating equalizer parameters. These algorithms minimize the MSE. In general, they require an initial or even periodic data sequence (training sequence) known by the receiver to ensure the convergence of the algorithms. Once convergence is established, the algorithms are decision-directed and minimize the estimated MSE given by $M\hat{S}E = E\left\{\left|s_k - \hat{u}_k\right|^2\right\}$ where $\hat{u}_k$ is a tentative decision taken at the equalizer output.

For each iteration, the equalizer structure is depicted in Fig. 3.11. Output channel sequence Y_k and estimated symbols sequence $\hat{U}_k$ provided by the channel decoder output of the previous iteration feed the equalizer (IC). The IC output is given by

$$s_k = P_k^T R_k - Q_k^T \hat{U}_k \tag{3.11}$$

Where $R_k = \left[r_{k+L_1} \cdots r_k \cdots r_{k-L_1}\right]^T$ and $\hat{U}_k = \left[\hat{u}_{k+L_2} \cdots \hat{u}_k \cdots \hat{u}_{k-L_2}\right]^T$ are the received samples vector and the estimated mean values vector, respectively. $P_k = \left[\hat{p}_{-L_1}(k) \cdots \hat{p}_0(k) \cdots \hat{p}_{L_1}(k)\right]$ and $Q_k = \left[\hat{q}_{-L_2}(k) \cdots \hat{q}_0(k) \cdots \hat{q}_{L_2}(k)\right]$ are the equalizer parameters vector corresponding to filters P(f) and Q(f), respectively. L_1 and L_2 are appropriate values greater or equal to L. T denotes the transposition.

For a time invariant channel, the LMS algorithm is used for all iterations (p ≥ 1) and initialized by a training sequence at the beginning of the transmission. Corresponding update equations are given by

$$P_{k+1} = P_k - \mu R_k^*\left(s_k - \hat{u}_k\right) \tag{3.12}$$

$$Q_{k+1} = Q_k + \mu \hat{U}_k^*\left(s_k - \hat{u}_k\right) \tag{3.13}$$

Where μ is an appropriate step size.

For a time varying channel, the RLS algorithm is used for all iterations (p ≥ 1) and data aided by a periodic training sequence. At the first iteration (p = 1), the estimated symbols are unknown and the IC is a purely adaptive transversal filter. For the other iterations (p > 1) and from relations (3.5) and (3.6), the equalizer coefficients p_l and q_l can be calculated with

$$p_l = \frac{\beta h_{l-1}^*}{hh_0} \tag{3.14}$$

$$q_l = \frac{\beta hh_l}{hh_0} \cdot l \neq 0, q_0 = 0 \cdot \tag{3.15}$$

The advantage of this approach is to have a smaller number of taps to be adjusted in order to enable the algorithm to follow channel fluctuation rapidly. However, this approach does not take into account the sub-optimality of IC during the very first iterations.

Furthermore, when a frequency offset exists between the transmitter and the receiver oscillators, the equalizer can integrate a phase-locked loop (PLL).

Chapter 4
Turbo Equalization – Tutorial

The introduction of concatenated error correction codes and decoding theory (turbo codes) in the field of communications has enabled information transmission rates that closely approach Shannon's theoretical limits. This chapter begins with an explanation of turbo coding theory. It then shows how the turbo scheme can be modified to develop a "turbo equalizer" to equalize the output from an inter-symbol interference (ISI) channel. Next the necessary modifications of a turbo equalizer to allow adaptation to a time-varying ISI channel are described and the issue of receiver complexity is discussed.

4.1 Turbo Coding

Turbo coding involves the concatenation of two convolutional encoders separated by an interleaver. The "turbo" principle refers to the iterative decoding of the complex code generated. Each constituent code is decoded separately, but with the benefit of a limited amount of information learned from the other code during previous iterations. This separation enables efficient, near optimal decoding. Each code is decoded at the receiver by a four-port device known as a soft-input soft-output (SISO) decoder. A firm appreciation of the SISO module is the key to understanding how turbo codes are decoded.

This presentation of turbo coding theory develops a detailed notation that represents an expansion, combination, and clarification of conventions taken from the literature. It is based on notations primarily used at two labs, Benedetto et al. at the Jet Propulsion Laboratory [2] and Hagenauer et al. at The Technical University of Munich [3].

4.1.1 SISO

When a block of encoded information is transmitted over a channel with additive white Gaussian noise (AWGN), information about the k^{th} symbol of the transmitted sequence has three independent sources.

1. The received value of the k^{th} symbol.
2. A priori information about the symbol sent in the k^{th} position.
3. Information about the k^{th} symbol that resides in all other received symbols within the block by virtue of the error correction coding done prior to transmission.

These three sources of information are referred to as intrinsic, a priori, and extrinsic information respectively.

The four-port SISO module, illustrate in Fig. 4.1, receives Observation and intrinsic as inputs and returns extrinsic information and the a posteriori probabilities about the encoded information words that were sent through the channel. One input receives a sequence of independent probabilities of transmitted code words based on the channel outputs, named observations and designated "Obs". The other input is the a priori probabilities of information words, "Int". The two outputs are a sequence of probabilities of the information words ("Ext" and "APP") from the corresponding encoder. These probabilities can be generated using the SISO's knowledge of the encoding algorithm. For descriptive purposes these four ports are referred to as the observation for the received words and intrinsic, a posteriori and extrinsic for the uncoded words respectively.

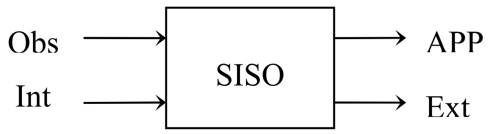


Figure 4.1: Block diagram of a SISO module.

A SISO can be implemented with a variety of algorithms as long as the encoder can be represented as a trellis. The following description of a trellis is established to facilitate explanation of the function of a SISO module. The encoders used in turbo coding are systematic convolutional encoders that accept as input, a sequential block of K information words designated *u*. The information words are comprised of small groups of k_s symbols. The information symbols are drawn from an alphabet of size N_I. The encoder generates a sequence of K parity words p, with p_s parity symbols belonged to an alphabet also of size N_I. Since the codes are systematic the output of the encoder is the sequence of code words, *c*, that are assembled from *u* and *p*. Thus, the kth codeword of c contains $n_s=k_s+p_s$ symbols and can be viewed in three equivalent forms.

$$c_k=\left[u_k\,p_k\right]=\left[u_{k,1}\cdots u_{k,k_s}\,p_{k,1}\cdots p_{k,p_s}\right]=\left[c_{k,1}c_{k,2}\cdots c_{k,n_s}\right] \tag{4.1}$$

The rate of the resulting code is k_s/n_s.

A section of the encoder trellis can be represented by two sets of N states $\{s_0, s_1, s_2, ..., s_{N-1}\}$ interconnected by $N\times N_I^{k_s}$ edges (Fig. 4.2). An edge, e_k, connects the state of the encoder, s_{k-1}, at time k-1 to the state, s_k, at time k.

The following four functions are associated with each edge, e_k, at time k.

1. The starting state $s^S(e_k)$.
2. The ending state $s^E(e_k)$.
3. The input information word $u(e_k)$.
4. The output codeword $c(e_k)$.

If the encoder is in state $s_{k-1}= s^S(e_k)$, an input of $u_k = u(e_k)$ results in a transition to state $s_k = s^E(e_k)$ along edge e_k in association with the generation of the codeword $c_k = c(e_k)$.

The number of states and the edge functions above depend on the characteristics of the particular encoder implemented. Given an initial trellis state, there is a one-to one correspondence between the sequence of edges traversed and each of the three sequences described by the input u, the coded output c, and the sequence of states occupied.

In this implementation, the SISO uses an optimal decoding algorithm that computes the maximum a posteriori probability of an information word at each time k given the entire received sequence of symbols, P{u_k/r}.

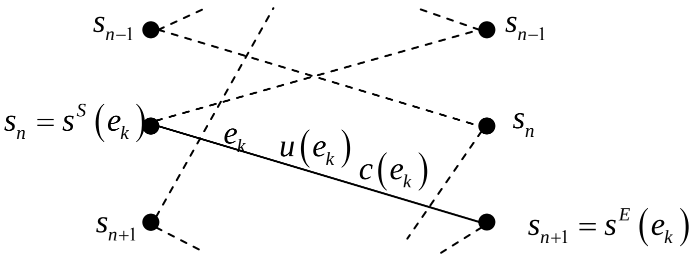


Figure 4.2: Section of encoder trellis.

This trellis decoding algorithm was first described by Bahl et al.[4] in 1974 and is referred to as the BCJR algorithm. Once these probabilities are known, the algorithm selects the word u_k that has the maximum P{u_k/r} as output. The algorithm initially calculates probabilities of transitions along each edge given the matched filter output vector y, i.e. P{e_k/r}. Note that the desired quantity,

$$P\{e_k|r\} = \frac{P\{e_k, r\}}{P\{r\}} \tag{4.2}$$

Since the term P{r} is constant for all edges it does not influence the final decision and may be dropped without any loss of performance. It is convenient to derive P{e_k,r} and assume that the appropriate integral has been done where probabilities of continuous data are expressed.

The encoder is initialized to state zero and forced to return to state zero at the end of the input sequence by calculating and appending a short string of additional input symbols.

It is worthwhile taking advantage of the monotonic nature of the logarithm function and refer to natural logarithm of probabilities. This improves decoder execution speed, as will be shown later, but more importantly for the moment, it leads to improved clarity in illustrating the rest of the SISO algorithm and the interaction of the two SISOs within a turbo decoding scheme.

A SISO module has two inputs, Obs(r_k), the probabilities of the codewords given the observable information received, and Int(u_k) the intrinsic information about the encoder input words. The SISO takes these two sources of information about the encoded sequence, and using its knowledge of the encoder trellis, the SISO uses the BCJR algorithm to generate the quantities, P(u_k,r). The difference between the information generated by the BCJR algorithm and the information provided to the SISO defines the extrinsic information contained within the code.

$$APP(u_k) = \frac{P(u_k, r)}{P(u_k)P(r_k|c_k)} \tag{4.3}$$

APP(u_k) represent information about words u_k, that resides in all of the received sequence excluding the kth word; information that is present by virtue of the trellis constraints imparted by the corresponding encoder. The two quantities Ext(u) and APP(u) are the output of the SISO module.

The description of SISO function has been developed with inputs consisting of a vector of word likelihoods. If each vector element is replaced by a row of symbol likelihoods at the SISO inputs, as can be done when symbol transmissions are independent, extrinsic information for individual symbols can be calculated by the SISO.

4.1.2 Turbo Code in Parallel Concatenated Configuration

In turbo code two convolutional codes can be concatenated in a serial or parallel fashion. The code words are broken into their individual symbols before being transmitted.

Initially we will describe a parallel concatenated convolutional code (PCCC) as illustrated in Fig. 4.3 (encoder) and 4.4 (decoder). The encoders are labeled code 1 and code 2, but the constituent code used may or may not differ. The information words to be transmitted are in vector, u, of length K.

The first encoder generates K parity words p^1 $\left(c_k^1 = \left[u_k\, p_k^1\right]\right)$. Prior to input in the second encoder, u is shuffled by the pseudo-random interleaver, Π, to form u$^\pi$. The interleaver is generated with a random generator, but the interleaver performs an invertible operation. The receiver has the knowledge required to interleave and de-interleave at will. The second encoder operates on u$^\pi$ and produces K parity words, p$^2\left(c_k^2 = \left[u_k^\pi\, p_k^2\right]\right)$.

However, only minimal coding gains arise from sending both copies of the redundant information u and u$^\pi$. Thus the code words transmitted as the vector x, of length K, are formed from c^1 and part of c^2 as follows $x_k = \left[u_k\, p_k^1\, p_k^2\right]$.

The code words x are transmitted one symbol at a time and the channel output r, is presented to the decoder of Fig. 4.4. For clarity of presentation it is helpful to consider operators P(.), Obs(.), Int(.), APP(.), and Ext(.) as casting the probabilities as a matrix when given a vector argument. The matrix size is K by n$_s$ where K is the length of the vector and n$_s$ is the number of symbols in the codewords that comprise the vector argument. Although the rows of Obs(r) are n$_s$ elements in length.

$$Obs\left(r_k\right) = \left[Obs\left(u_{k,1}\right)\cdots Obs\left(u_{k,k_s}\right)Obs\left(p_{k,1}^1\right)\cdots Obs\left(p_{k,k_s}^1\right)Obs\left(p_{k,1}^2\right)\cdots Obs\left(p_{k,k_s}^2\right)\right] \qquad (4.4)$$

$$Obs(r) = \left[Obs\left(u\right)Obs\left(p^1\right)Obs\left(p^2\right)\right] \qquad (4.5)$$

Given knowledge of the interleaver, the appropriate columns of Obs(r) can be selected and manipulated to give correctly ordered symbol probabilities for each SISO. The channel probability inputs to the two SISOs are designated Obs(r^1) and Obs(r^2). For the first SISO decoder $Obs\left(r^1\right) \approx Obs\left(c^1\right) = \left[Obs\left(u\right)Obs\left(p^1\right)\right]$. The second SISO decoder requires interleaved information symbol Probabilities so the probabilities Obs(u) are first interleaved to form Obs(u^π), then $Obs\left(r^2\right) \approx Obs\left(c^2\right) = \left[Obs\left(u^\pi\right)Obs\left(p^2\right)\right]$.

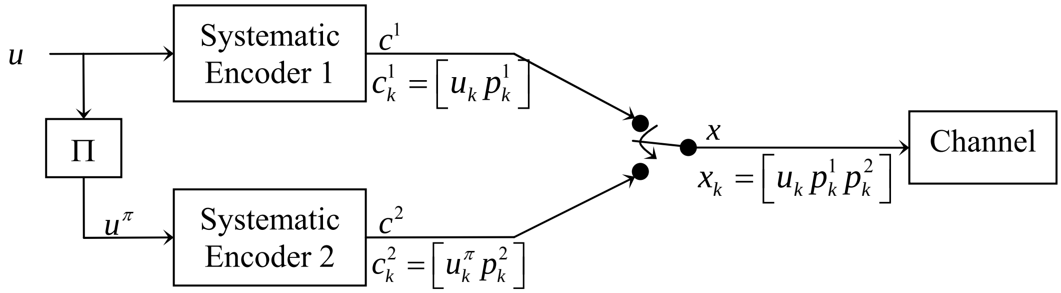

u
Systematic Encoder 1
c^1
$c_k^1 = \left[u_k\, p_k^1\right]$
Π
u^π
Systematic Encoder 2
c^2
$c_k^2 = \left[u_k^\pi\, p_k^2\right]$
x
$x_k = \left[u_k\, p_k^1\, p_k^2\right]$
Channel

Figure 4.3: PCCC Encoder

In Fig. 4.4 Int(u) represents intrinsic information about u. The a priori input to the first SISO is initialized with Int^1(u) = Ext^2(u) = ½. Once these SISO inputs are formed, the BCJR algorithm is executed within SISO 1. The Extrinsic output of the first SISO Ext^1(u) is interleaved and used as input to the second SISO as Int^2(u^π). Ext^1(u) is information that resides in the first code, and is independent of the channel observations and the encoded information in code 2. Therefore it is valid a priori information to give to the second SISO. Alternatively the output of the second SISO Ext^2(u), information encoded by code 2, is independent of encoded information in code 1, so it is valid a priori information, after de-interleaving, to form Int^1(u) before executing SISO 1 a second time. This algorithm can be executed iteratively.

As this process iterates the decoding decisions gradually improve, but the extrinsic information from each SISO slowly lose their independence from one another and rate of improvement in decoding with each iteration soon slows and eventually ceases.

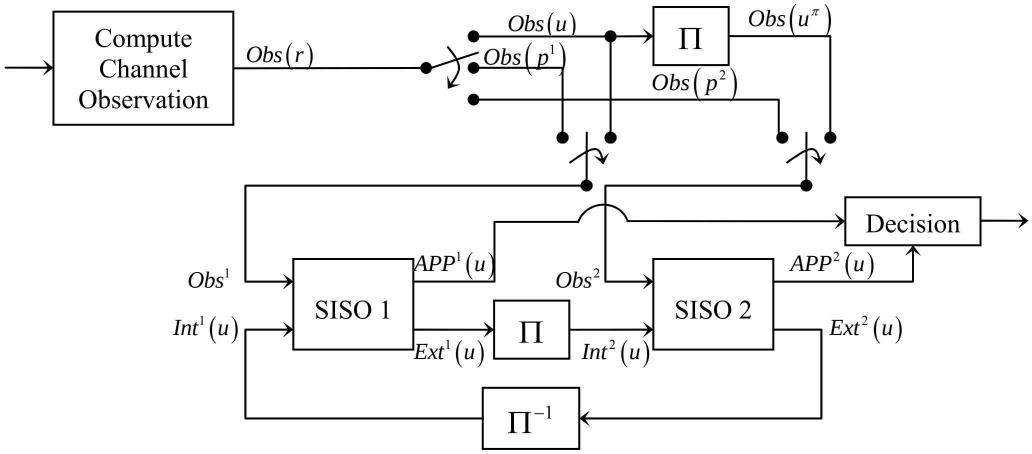


Figure 4.4: PCCC Decoder

4.1.3 Turbo Code in Serial Concatenated Configuration

Serial concatenated convolutional codes (SCCC) are more easily grasped after the discussion of the parallel case above. Fig. 4.5 and 4.6 show block diagram of a serial concatenated encoder and decoder respectively. The components in the figures perform the same functions as in the parallel case although the interconnections differ in the serial case. The first and second constituent codes of the encoder are often referred to as the outer and inner codes, respectively. This terminology arises because it avoids referring to encoders and SISOs by number. The later practice is confusing because the inner code (code 2) is decoded first at the receiver.

In the transmitter, note that the entire codeword of the outer encoder is interleaved to form the input to the inner encoder, $u^2 = c^\pi$, and the output of the inner encoder $c^2 = x$, is the vector used to modulate the signal for transmission. The encoders work exactly as in the parallel case.$\left(c_k^1 = \left[u_k^1 p_k^1\right], c_k^2 = x_k = \left[u_k^2 p_k^2\right], "u^2 = \left(c^1\right)^\pi"\right)$.

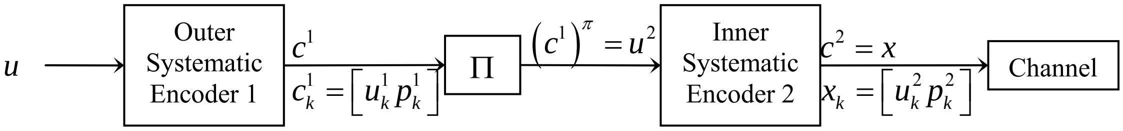

u
Outer Systematic Encoder 1
c^1
$c_k^1 = \left[u_k^1 p_k^1\right]$
Π
$\left(c^1\right)^\pi = u^2$
Inner Systematic Encoder 2
$c^2 = x$
$x_k = \left[u_k^2 p_k^2\right]$
Channel

Figure 4.5: SCCC Encoder

In the serial decoder as in the parallel case, probabilities Obs(r) are formed from the received signal r. For the inner SISO the entire Obs(r) is used as input.$\left(Obs\left(r^2\right) = \left[Obs\left(u^2\right)Obs\left(p^2\right)\right]\right)$.

For Obs(r^1), the observation input to the outer SISO, a component of information comes from the channel and another comes from the inner SISO. The channel component is formed by taking Obs((c^1)$^\pi$) from Obs(r) and de-interleaving to form $\left(Obs\left(c^1\right) = \left[Obs\left(u^1\right)Obs\left(p^1\right)\right]\right)$.

The inner SISO, contributes Ext(u^2) = Int(u^1)$^\pi$) which is de-interleaved to form Int(u^1).

Note that, this differs from the parallel case where the observation SISO input comes entirely from the channel.

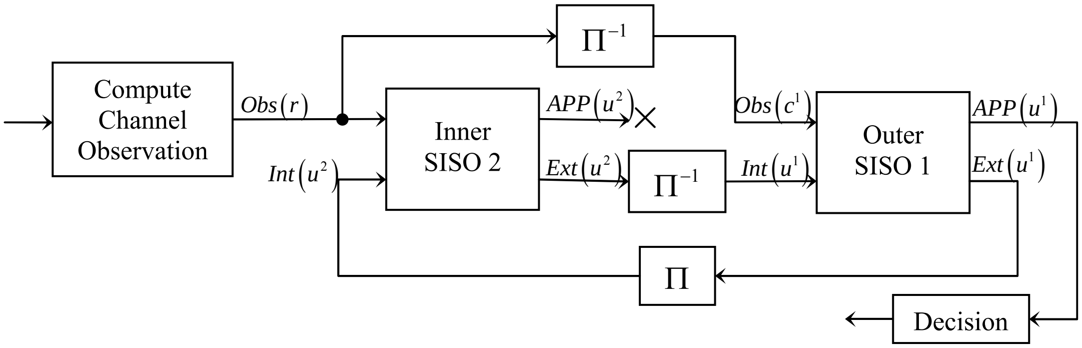


Figure 4.6: SCCC Decoder.

Extrinsic information about the information words from the outer SISO, Ext(u^1), correspond after interleaving to independent information about the words u^2 and is thus suitable as intrinsic inputs to the inner SISO. Iteration around the loop will initially improve overall decoding accuracy, but improvement eventually ceases. Stopping criteria based on cross entropy are similar to the parallel case.

4.2 Turbo Equalization

4.2.1 Turbo equalization - Principle

Equalization involves compensating for receptions over multiple transmission paths. This section describes modification of a turbo code to equalize data received over an ISI channel. Note, we should know that the outer encoder can be a turbo code.

An ISI channel is most commonly modelled by a tap delay line as shown in Fig. 4.7. In the figure, the M channel coefficients, h_m, and Gaussian noise, n_k, are complex. The input u is a real sequence of symbols of alphabet size N_I and the output vector y is complex. Given the input u the ISI channel can be considered to be in one of the states that are characterized by the past M-1 symbols sent into the channel.

This channel model is readily adapted to the trellis model that was introduced in Fig.4.2. The same BCJR algorithm employed in the SISO module can be used to decode the channel. The two key quantities to determine are $P\{u_k\}$, the a priori probability of each input symbol and $P\{r_k/e_k\}$, the probability of receiving the value r_k given the current and past M-1 channels inputs. If it is assumed the channel impulse response is known, at each time k, it is expected that the channel will produce the codeword c_k corrupted by channel noise. Thus $P\{r_k/e_k\}$ can be calculated from the Gaussian distribution of r_k about the symbol associated with the given edge.

$$P\left\{r_k \middle| e_k\right\} = \frac{1}{\sqrt{2\pi\sigma}} \int_{r_k-\Delta r}^{r_k+\Delta r} e^{-\frac{1}{2\sigma^2}\left\| r-c(e_k)\right\|^2} dr = \frac{1}{\sqrt{2\pi\sigma}} \int_{r_k-\Delta r}^{r_k+\Delta r} e^{-\frac{1}{2\sigma^2}\left\| r-\sum_{m=0}^{M-1} h_m u_{k-m}\right\|^2} dr. \qquad (4.6)$$

Given the terms, $P\{u_k\}$and $P\{r_k/e_k\}$, the equalization process needs to be integrated into a turbo coding scheme.

The dotted lines in Fig. 4.7 indicates how the ISI channel model can be broken into two parts, the first behaving as a convolutional encoder and the second part being a simple AWGN channel. Fig. 4.8 shows a block diagram of a serial encoder with the tap delay line functioning as the inner encoder and the Gaussian section of the channel taking on its usual role. Fig. 4.9 shows the turbo equalizer configuration. Two modifications relative to the SCCC decoder of Fig. 4.6 have been made. First, the SISO equalizer that replaces the inner SISO accepts the channel input, *r*, and calculates the channel likelihoods internally. Second the extrinsic outputs are modified. The upper output values are not needed so the output is eliminated. The lower output is critical.

Inside the SISO equalizer, symbol probabilities are calculated, and then the BCJR algorithm is executed as in the normal SISO. The decoding decision for the turbo equalizer is made as in the SCCC.

Note that computation of channel likelihoods and execution of the BCJR can be done independently, so channel observations could be calculated external to the SISO equalizer in a configuration analogous to the SCCC decoder. However, when attempting adaptation to a time varying channel the observation calculations are more intimately related to trellis decoding and the structure given is required.

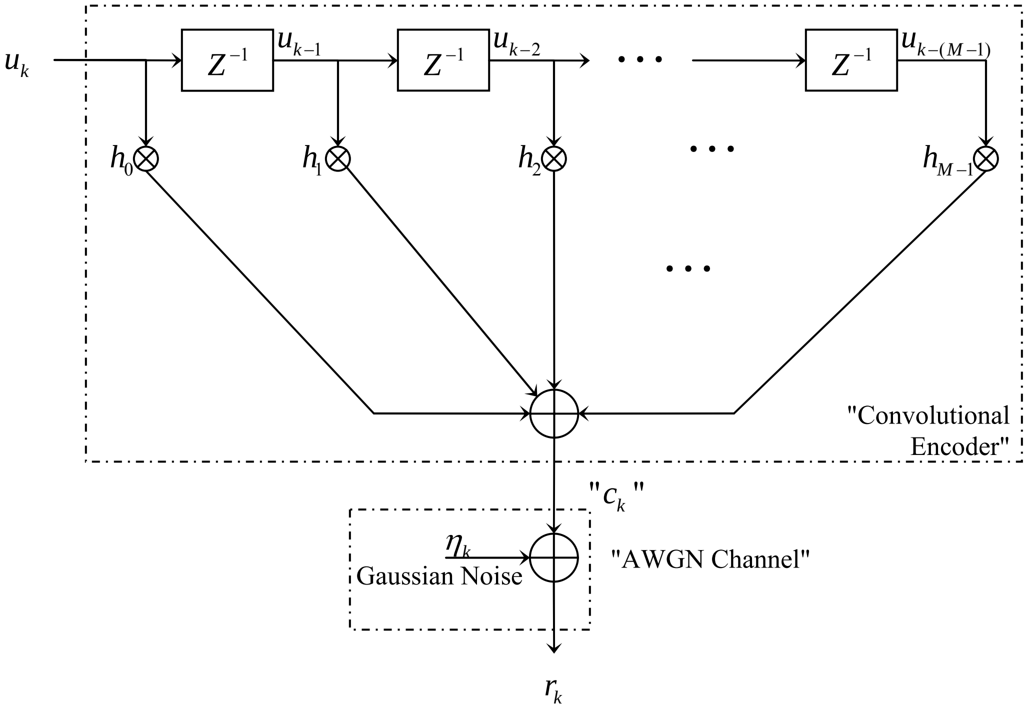


Figure 4.7: ISI Channel Modelled as Tap Delay Line

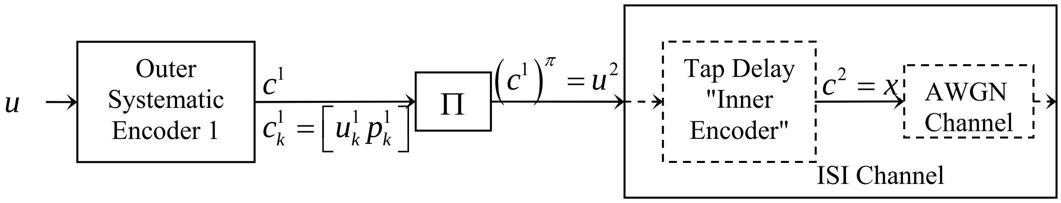


Figure 4.8: ISI channel configured as an inner encoder and AWGN channel

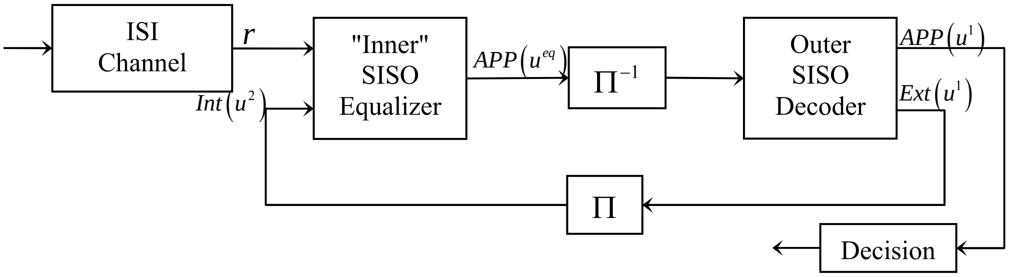


Figure 4.9: Turbo Equalization.

4.2.2 Turbo equalization – Metrics of performance

A major goal of any communication link is to have reliable communication with a minimized signal-to-noise ratio (SNR). A low SNR is typically associated with the benefits of low power consumption, small size, and hence portability. While error correction coding schemes such as turbo codes make possible dramatic reductions in SNR, these gains are acquired at the price of increased complexity at the receiver. The complexity manifests itself by increased power consumption at the receiving end, but more importantly it can introduce a significant delay due to the time required to decode the received signal. In some applications this latency is acceptable. The delay can be minimized by adding hardware decoding circuitry in the receiver.

To represent reliability of a transmission scheme, the bit error rate (BER) of received data is graphed versus SNR. The SNR is expressed in terms of Eb/No, where Eb is the average energy in each transmitted bit of information and No/2 is the intensity of the white noise process active on the channel. The ratio Eb/No is the signal power-to-noise ration where the power of the noise is defined in a bandwidth equivalent to that of the bit rate. This presentation of the reliability is common in the literature because it enables establishment of an SNR threshold for a given BER. Unfortunately this method is very imprecise at estimating very low BERs (especially BERs at SNRs above the SNR threshold) because the error events are so rare and the processing time necessary to encounter a significant number of these rare events is very long. The SNR threshold is defined at a specified BER, but after the last iteration of decoding, the BER versus Eb/No curve is steep. Thus, for a time-invariant ISI channel, the SNR threshold is not very dependent on SNR in the BER = 10^{-5} range.

Complexity is expressed in terms of the number of additions, multiplications, maximization operations, and memory requirements. The rate of growth in the number of these operations when decoding increases exponentially with the memory size of the constituent odes of the turbo code. In the case of a turbo equalizer the memory size grows with the length of the channel impulse response.

4.3 Channel Estimation

To conclude with the turbo equalization analysis, the problem of channel knowledge also should to be raised and carefully investigated. We propose here at least two methods for re-estimating the channel coefficients. A significant part of the degradation introduced by mismatched channel estimation can be recovered by such re-estimation methods, without substantial complexity increase.

4.3.1 EM based re-estimation

Most of the previous work, assume that the channel coefficient vector is known at the receiver, in practice, channel coefficients are estimated by inserting a known training sequence in the transmitted frame. The classical method of correlative channel estimation (pseudo inverse method) causes important performance degradation with respect to the perfect channel estimation. This gap can be reduced by using more advanced channel estimation techniques, such as the expectation-maximization (EM) algorithm, a powerful tool that performs maximum likelihood

(ML) parameter estimation of a doubly stochastic process in an iterative fashion. The Fig. 4.10 represents the structure of the EM channel re-estimation.

The optimum ML solution to the problem of channel estimation $\hat{h}_{ML}$ is obtained by maximizing the log likelihood function of the received vector of samples. In most cases, finding this solution is practically impossible. However, the EM algorithm provides a feasible solution to this optimization problem by iteratively re-estimating the channel coefficients, so that a monotonic increase in the likelihood function is guaranteed. It achieves this monotonic increase by introducing the following auxiliary function:

$$Q\left(h\middle|\hat{h}^{(p)}\right)=E_w\left\{\log p\left(\frac{w}{h}\right)\middle|r_1^{\tau},\hat{h}^{(p)}\right\} \tag{4.7}$$

where $\hat{h}^{(p)}$is the vector of estimated channel coefficients at the pth iteration of turbo detector and w is the so-called complete data that is actually unobservable, but whose knowledge makes the estimation easy. Since the complete data w is unknown, its log-likelihood function is a random variable, and therefore we maximize the conditional mean of this log-likelihood function given the incomplete data (observable) r_1^{τ} and the set of most recent channel estimates $\hat{h}^{(p)}$. The complexity and the rate of convergence of the algorithm are affected by the choice of the complete data w.

The EM re-estimation sequencing is as follows: during each iteration p of the turbo detector, an iteration of the EM algorithm is performed together with the SISO equalization process. This EM iteration itself involves a two step procedure:

1) E-step: Compute $Q\left(h\middle|\hat{h}^{(p)}\right)=E_w\left\{\log p\left(\frac{w}{h}\right)\middle|r_1^{\tau},\hat{h}^{(p)}\right\}$;
2) M-step: Solve $\hat{h}^{(p+1)}=\arg\max_h Q\left(h\middle|\hat{h}^{(p)}\right)$

The new vector of channel coefficients $\hat{h}^{(p+1)}$ is used by the SISO ISI decoder at iteration p+1 of the turbo detector. The expectation operation in the E-step is with respect to the complete data w. the performance of the EM algorithm is very sensitive to the choice of the initial estimate $\hat{h}^{(0)}$. Therefore, we initialize the algorithm by applying the pseudo-inverse method on the training sequence.

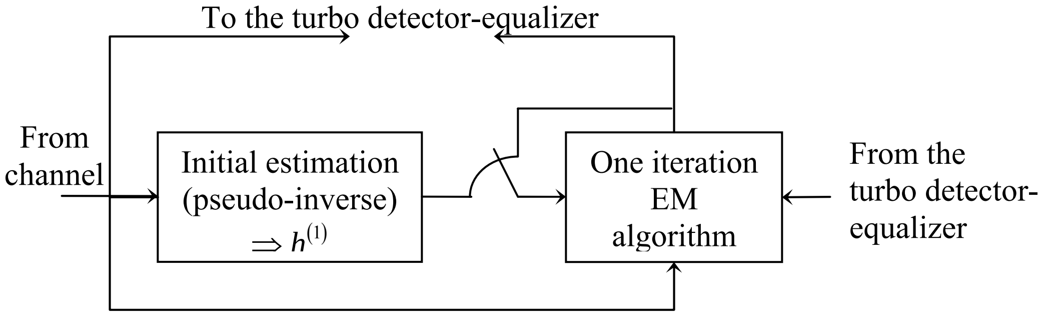

To the turbo detector-equalizer
From channel
Initial estimation (pseudo-inverse) $\Rightarrow h^{(1)}$
One iteration EM algorithm
From the turbo detector-equalizer

Figure 4.10: EM channel re-estimation

4.3.2 Bootstrap channel re-estimation

Instead of considering estimated data symbols after the ISI decoder, however, decisions are taken after re-interleaving of the decoded sequence, like it is shown in Fig. 4.11. Thus, it benefits from time diversity brought by interleaving and from channel decoding efficiency. We now describe the sequencing.

1) After re-interleaving of soft outputs produced by the channel decoder, a hard decision is taken on each bit of each symbol of sequence. An estimate of useful symbols is then available (tail symbols, guard symbols, and training sequence are known a priori).
2) The matrix system is formed.
3) A solution minimizing the error probability is well known.

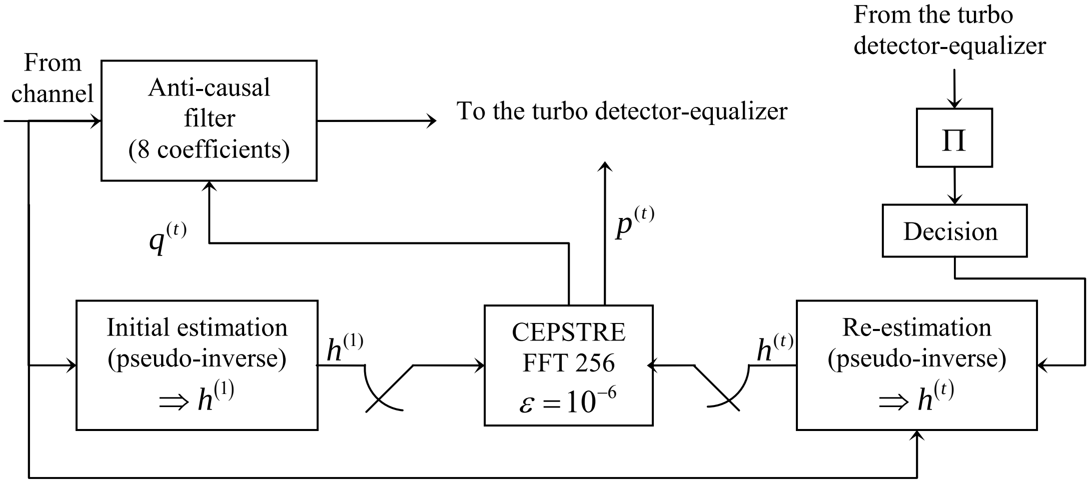

From channel
Anti-causal filter (8 coefficients)
To the turbo detector-equalizer
From the turbo detector-equalizer
Π
Decision
$q^{(t)}$
$p^{(t)}$
Initial estimation (pseudo-inverse)
$\Rightarrow h^{(1)}$
$h^{(1)}$
CEPSTRE
FFT 256
$\varepsilon = 10^{-6}$
$h^{(t)}$
Re-estimation (pseudo-inverse)
$\Rightarrow h^{(t)}$

Figure 4.11: Bootstrap channel re-estimation.

4.4 Interleaving and De-interleaving

The interleaving function allows temporal error sequence distribution to be modified and splits the error series. Used generally with time varying channels, the interleaver is an essential function of the turbo-equalizer even if the channel is time invariant. Over a severe frequency selective channel, the likelihood of the estimated data is weak and the equalizer output presents series of errors with large values which perturbed the channel decoder. Due to this, the interleaving dimension may be sufficient in comparison with the error sequence length but also in comparison with the error value. Some results related to interleaving performance versus interleaver size are given in [17].

The turbo equalizer represented in Fig. 4.9 can interleave symbols or bits according to the position of mapping versus the one of interleaver. The bit-interleaver located between the channel encoder and the mapper is often used and gives excellent performance. Nevertheless, it can be demonstrated that theoretical bounds for high-order modulation give better performance with a symbol-interleaver than a bit-interleaver.

4.5 Optimised labelling maps

We propose here to study the influence of mapping over the performance of turbo equalization. We show this technique interest to a good choice of mapping. In order to explain choice technique, we use two different approaches provided by Ten Brink [27] and Gorokhov [29].

Gorokhov propose two design criteria for optimization labelling, based on the computing of specific distances of constellation. The choice of mapping is then achieved without any notion of functioning point. However, these design criteria give preference to asymptotic performances than that of iterative process during the convergence, which is not always desirable.

Ten Brink use mutual information parameters, which allow distinguishing exactly each mapping performances, however, the computation of this parameter requests an intensive simulation and depends of sight functioning point.

4.5.1 The geometrical approach

Let us consider the optimal decoder. In practice, trellis codes are used as forward error correction (FEC) for noisy fading channels such as (concatenated) convolutional codes. A typical error pattern is characterised by a small number of erroneous coded bits at error rates of potential interest. The number of erroneous coded bits is a small multiple of the free distance of the code; this number is only a small fraction of the total number of coded bits. Due to interleaving, these erroneous coded bits are likely to be assigned to different labels and therefore different symbols. More specifically, the probability of having only one erroneous coded bit per symbol approaches one along with the increase of the data block size. Hence, we are aiming at decreasing the rate of errors such that at most one bit per symbol is corrupted, in order to improve on the overall error rate, for error rates of potential interest.

For a given pair $\{S, \mu\}$of signal set and labelling map μ, define D_e the minimum of the Euclidian distances between the symbols of S whose m-tuple labels differ in exactly one position. Based on the previous observations, Gorokhov suggests the following design criterion.

For given signal set S, select a labelling map μ: $\{0,1\}^m \rightarrow S$ that maximises D_e.
The distance measure D_e will be called the effective free distance of a given pair $\{S, \mu\}$. Note that D_e is lower bounded by the exact free distance D_f of the signalling S where D_f is defined as the minimum of Euclidian distances between the symbols of S.

The efficiency of criterion has been proved theoretically for big data blocks ($N \rightarrow \infty$) when the underlying FEC has a finite trellis complexity. These results apply to most of practically used codes such as the standard convolutional codes. (This analysis, however, rules out random like codes such as turbo- and LDPC codes). We can show that the achievable error rate is close to the optimum when D_e is comparable to the maximum within the aforementioned distance set. In all cases the ratio (D_e^2/D_f^2) reflects the order of performance improvement (in SNR) as compared to the standard labelling maps. It is appropriate to mention that the known maps (e.g., Gray, Set partitioning…) are characterised by $D_e = D_f$.

The second design criterion aims at an improved SNR threshold of the practically used iterative decoders. A typical feature of the iterative decoding is a relatively poor performance up to some SNR threshold. Above this threshold, the error rate of the iterative decoding approaches the performance of optimal decoding. Therefore decreasing the SNR threshold value is highly desirable. This threshold value depends on the starting point of the iterative procedure, i.e., on the distribution of the reliability values provided by de-mapper at the first iteration. The 'worst' reliability values are due to the neighbouring symbols, i.e., the element of S that is D_f apart. Hence the 'average' number of coded bits that suffer from the 'worst case' reliabilities is likely to be proportional to the 'average' number of positions in which the labels, encoding the neighbouring symbols of S, differ.

For a given pair {S, μ} of signal set and labeling map μ, define $\overline{H}$ the average of hamming distances between all m-tuple labels whose respective symbols from S are D_f apart. According to the above discussion, Gorokhov may suggest the following design criterion.

For a given signal set S, select a labelling map μ: $\{0, 1\}^m \rightarrow S$ that minimises $\overline{H}$.
Note that the first criterion, as such, leads to the standard gray labelling map which is characterised by $\overline{H}$ =1. However, the gray labelling also yields $D_e = D_f$ and therefore a poor asymptotic error rate at high SNR. Clearly, the second criterion only makes sense when used along with the first criterion and $D_e > D_f$.

Hence, the design objective is to attain, for a given (preferably maximum possible) $D_e > D_f$, the minimum of $\overline{H}$, with the lower bound $\overline{H} \geq 2$. Labelling maps designed according to these guidelines will be referred to, according to the main criterion as effective free distance (EFD) labelling maps.

Table 4.1 summarize the values of distances D_e and $\overline{H}$ for Gray, Set Partitioning (SP) and EFD mapping and the Fig. 4.12 represent the constellation of 16-QAM EFD mapping proposed by Gorokhov.

Gray mapping had the minimum Hamming distance $\overline{H}$, EFD mapping allow at once to maximize the effective free distance D_e and to minimize the Hamming distance $\overline{H}$, this mapping associated to turbo equalization allow to obtain best performance than the SP mapping. Thus, SP mapping had the lowest free distance, equal to that for Gray mapping, whereas it allow to obtain the best asymptotic performances.

Mapping	Gray	SP	EFD
D_e	D_f	D_f	$\sqrt{5}\, D_f$
$\overline{H}$	1	2.3542	2.1458

Table 4.1: Effective free distance and Hamming distance.

4.5.2 The mean average mutual information approach

As turned out in simulations the right choice of the mapping is crucial for a good performance of iterative de-mapping and decoding. In this section, we will briefly description of an approach to obtain a design rule based on mutual information. This significantly reduces the burden of performing exhaustive simulations to find good mapping since mutual information is very easy to calculate by numerical integration.

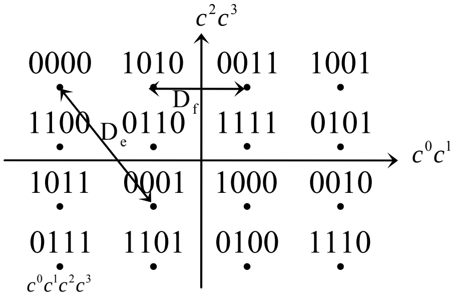


Figure 4.12: Constellation of 16-QAM EFD mapping

We observed that a different random interleaver or a different encoder does not have a significant influence on the BER performance of the iterative decoding scheme, with respect to different mappings. Hence, the choice of the right mapping dominates the achievable BER, and can concentrate on looking solely into the mapping to optimize the composite encoding scheme, consisting of encoder, interleaver and mapper. Naturally the following question arises: 'Which is the best mapping?'

For characterizing a mapping we chose the mean average bit-wise mutual information, conditioned on 'no other bit known', 'one other bit known', …,'all other bits known':

$$\begin{aligned}
I_0 &= \overline{I(X_k;Z)} = \frac{1}{M}\sum_{k=0}^{M-1} I(X_k;Z) \\
I_1 &= \overline{I(X_k;Z|X_l)} = \frac{1}{M(M-1)}\sum_{k=0}^{M-1}\sum_{\substack{l=0\\ l\neq k}}^{M-1} I(X_k;Z|X_l) \\
&\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
I_{M-1} &= \overline{I\left(X_k;Z\middle|\{X_{\forall l,l\neq k}\}\right)} = \frac{1}{M}\sum_{k=0}^{M-1} I\left(X_k;Z\middle|\{X_{\forall l,l\neq k}\}\right)
\end{aligned} \tag{4.8}$$

Note that in this section X = (X_o, …, X_{M-1}) is the input to the AWGN channel, X_k the respective bit of the mapped codeword, and Z the noise affected output of the channel. With the chain rule of mutual information it can be shown that

$$I(X;Z) = \sum_{i=0}^{M-1} I_i \tag{4.9}$$

The symbol wise mutual information I(X;Z) is independent of the applied mapping, and only dependent on the SNR.

Ten Brink interpretation is based on the observation of strong correlation between I_o and the BER. High I_o means low BER (no iteration) and vice versa. However, it can be proven analytically that this relationship is not a strict one – but apparently sufficiently strong for gaining some insight into the mechanisms that make iterative de-mapping and decoding work. The basic idea:

For the first pass through the demapper and decoder, no a priori knowledge is available, and thus the unconditioned bit-wise mutual information I_o is the most important parameter of the mapping. If the iteration works, more and more bits will become known with high confidence, and the remaining 'conditioned' mutual information $I(X:Z) – I_o$ comes into play, until finally, only $I_{all} = I_{M-1}$ (perfect a priori knowledge) limits the achievable BER, if the iteration succeeded in getting that far.

To verify these ideas Ten Brink created a set of random 16-QAM mappings (truly 16-QAM, not 4-PAM mappings anymore) which cover a wide span of different mutual information I_o. He correlated them with the BER results after 7 iterations, and found a strong correlation between BER and mutual information I_o. The surprising result; there is no such thing like a 'best' mapping. Moreover, the best mapping depends on how much iteration one is willing to invest.

For low I_o the iteration does not even get started. The reliability output from the decoder is too poor to be useful for the demapper. Gray mapper exhibits the highest I_o value of all mapping, and thus it minimizes the BER after 'no iteration'. However, the iteration do not help a lot. For Gray mapping I_1, …, I_{all} each are about I_o (only slightly higher the more bits are known). Hence, even with perfect a priori knowledge, the demapper 'sees' an almost unchanged channel $I_{all} \approx I_o$. That is why a priori knowledge cannot help much for Gray mapping.

Keeping in mind that the sum (4.9) is required to add up to a constant value for all mappings, Ten Brink realize that if we sacrifice some mutual information at I_o, we gain some at I_1, …, I_{all}, which would appreciate a priori knowledge much better. And indeed, as we move from Gray mapping towards lower I_o, we notice that the achievable BER goes through several minima, depending on the number of iterations one is willing to spend: for a few number of iterations, we should not shift a lot of mutual information from I_o towards I_1, …, I_{all}, since we would not make use of it anyway due to stopping the iteration before it even can exploit the 'buried capacity' by having gathered enough a priori knowledge. Likewise, if we want to use a lot of iterations, we can shift more mutual information to I_1, …, I_{all}, but need to be aware of leaving sufficient at I_o to get the iteration started at all for the desired SNR region. If I_o gets too small the iterations improve only a little due to declining quality of the decoder soft output values.

Then Ten Brink found three characteristic branches:

1- For low SNR values, the BER is dominated by I_o values of the mapping.
2- Depending on the number of iterations the interleaver size, at a particular SNR value a more or less pronounced 'turbo cliff' introduces the transition from the 'no other bits known' to the 'all other bits known' channel 'capacity' I_{all}.
3- Finally, for higher SNR values, only I_{all} determines the BER performance, independently of the interleaver size.

A careful design is necessary: the 'best' mapping depends on the number of iterations one can afford, the SNR region of interest and on the interleaver size. A pragmatic choice would be to just choose a mapping with 'medium' unconditioned mutual information I_o, to get the iterative improvement started at sufficiently low SNR, and still leave enough conditioned mutual information I_{all} for having an acceptable low BER floor.

Chapter 5
The Iterative Observation Separation Technique

One of the important problems in reliable data communications over frequency selective fading channels is the mitigation of inter-symbol interference (ISI). Many of digital communication systems, like any coded packet oriented narrowband or CDMA mobile systems encounter this problem of data transmission over a multi-path channel where the noise, the inter-symbol interference (ISI) and the multiple access interference (MAI) are two of the main factors that degrade the performance. To protect the integrity of the transmitted data, a controlled amount of redundancy is added (encoding) using error correction code (ECC). In the past few years the number of users of wireless communication systems has been growing exponentially. For this reason increasing system capacity is a critical issue, especially for the next generation cellular systems whose have a capacity limited by interference. Any reduction of channel interference (ISI), and co-channel interference from own cell (MAI, Multiple Access Interference), and neighboring cells (inter-cell interference) improves the system performance. A promising approach to suppress interference and multi-path channel distortion is the work over signal processing.

The relatively recent introduction of the *turbo* principle in 1993 by Berrou and al. has enabled the design of communication systems with channel capacities closely approaching Shannon's theoretical limit. Since this date a considerable amount of the work has been done on *turbo* principle. Based on this invention, in the past few years, a new concept, called *turbo equalization*, has emerged as a way of efficiently fighting against channel ISI. The turbo principle has been first applied by Douillard and al. to the field of equalization. To combat the effects of ISI, linear (linear equalizer, LE), or non-linear processing (decision feedback equalizer, DFE, or detector) of the received symbols can be applied, but optimal methods are non-linear and based on ML estimation. And it turns into maximum a posteriori probability (MAP) estimation in presence of a priori information about the transmitted data.

More recently, Bauch and al. have been introduced optimal symbol-to-symbol BCJR detectors and decoders with a convolutional ECC to improve the scheme. Unfortunately, the complexity of all those ML/MAP or sub-MAP devices depends exponentially on the length of the channel's impulse response (CIR) and the size of modulation and this complexity become quickly prohibitive when higher level modulations rather than simple BPSK (or GMSK) and 6-tap (or more) CIR are considered. In order to reduce the computational complexity, Glavieux is one among the firsts that replaced the ML equalizer by a so called Interference Canceller. Since this is an adaptive weight FIR filter its computational complexity depends only in a linear way on the length of the CIR. The classical approach to equalization with unknown channels is to generate a single channel estimate based on the statistics of the channel. This requires either a known training sequence or a delayed decision directed approach.

Most previous work on iterative equalization and decoding assumes that the channel impulse response (CIR) is perfectly known to the receiver. Practically, the problem of channel knowledge also has to be raised and carefully investigated. Glavieux and al. use a least mean square (LMS) type of channel estimation algorithm besides a LE to estimate and track the CIR; other approaches use this information for estimation and equalization simultaneously. Tuchler and al. propose a recursive least square (RLS) type of iterative channel estimation algorithm based on soft information from the decoder, which is distinct from the equalization algorithm. Due to the additive noise at the receiver side the quality of the channel estimates may be very poor, especially in the case of rapid channel variation. These poor estimates will deteriorate the efficiency of the ISI and MAI reduction techniques. Thus joint processing of symbol detection and channel estimation is an attractive solution.

An optimum solution including channel estimation, symbol detection and channel decoding seems exorbitant in complexity for current communication systems such as GSM and UMTS-CDMA. Many iterative joint methods of symbol detection and channel estimation, using the soft or hard decoded symbols decision feedback, are proposed for the cases of narrowband modulations (no spreading) without MAI and for direct sequence CDMA.

In this context, we apply the concept named Observation Separation (OS) technique which partially separates the observation used for channel estimation from the observation that allows the symbol detection. The channel estimation is very simple and basically consists of a classical bootstrap process using linear pseudo-inverse. For every symbol, the detection takes into account just the part of observation, which contains the considered symbol and has length corresponding to the channel memory length. The detection techniques are the ZF-BLE for the initial iteration and a classical Interference Cancellation (IC) for the other iterations. We have observed that if matched filter is used at initial iteration, the results will be the same. A significant part of the degradation introduced by erroneous channel estimation can be recovered by such re-estimation methods, without substantial complexity increase.

We apply in this chapter the OS technique to the UMTS-TDD system and also to a narrowband system with a binary modulation BPSK and high level modulations M-QAM on a time-invariant ISI channels. For the purpose of comparison, we will take into account the classical iterative parallel interference cancellation (PIC) technique and the classical bootstrap technique already applied to GSM system. Thus joint processing of symbol detection and channel estimation is an attractive solution. Whatever the iterative procedure is, the soft or hard decision of the channel decoder output is used in the feedback loop and so for the symbol decision feedback when IC or PIC is considered.

5.1 The Iterative Observation Separation Technique

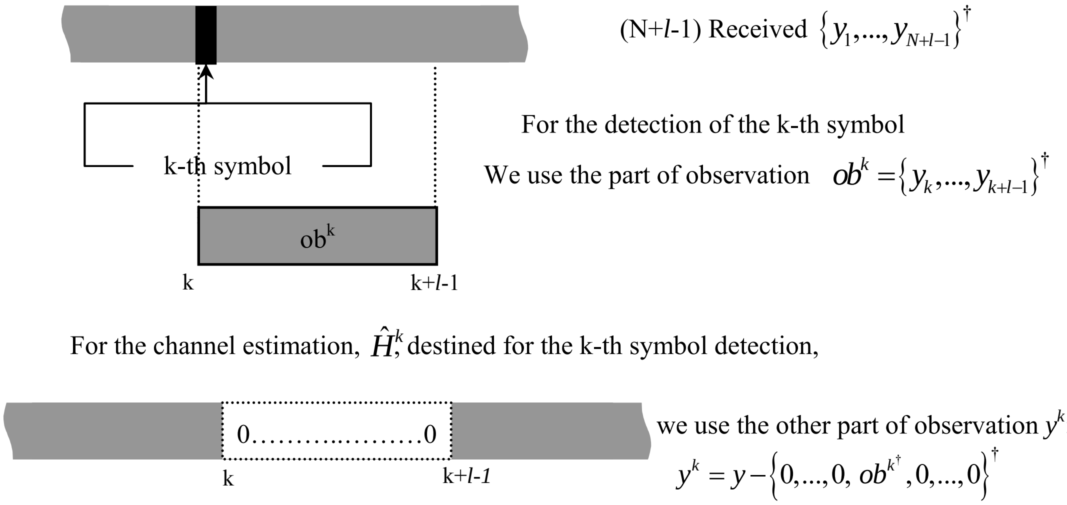
The Observation Separation (OS), presented in Fig. 5.1, is an iterative method which is used in a joint estimation of any channel H of length l and detection of any length N of symbols transmitted $\{x_k\}$.

This technique will be used after the first iteration, where we estimate channel by using the N_t symbols of the training sequence which is a part of $\{x_k\}$, $k=1, \ldots, N$, and any classical channel estimator to obtain $\hat{H}$. After that we can detect the transmitted vector $\{\hat{x}_k\}$ which is returned to be used in the second iteration like the correct transmitted vector, noted as $\{\tilde{x}_k\}$.

Then at the second iteration, for detecting the symbol x_k, the method propose to use a part of the received vector Y, which we will named the observation vector $ob^k = (y_k, y_{k+1}, \ldots, y_{k+l-1})$ and for making the channel estimate $\hat{H}^k$ which is used just for the symbol x_k detection, we use the other part of the received vector $y^k = y - \left\{0,\ldots,0, ob^{k^\dagger}, 0,\ldots,0\right\}^\dagger$.

In the symbol detector, we start by using the interference cancellation with the part of return symbols from the past iteration: $\{\tilde{x}_{n-l+1} \ldots \tilde{x}_{n-1}, \tilde{x}_{n+1} \ldots \tilde{x}_{n+l-1}\}$ and the estimate $\hat{H}^k$. If this part of return symbols is correct and $\hat{H}^k$ is good, we have the following interference samples:

$$s_{k'} = h_{k'-k} x_k + w_{k'} \ , \quad \forall k' \in [k, k+l-1] \tag{5.1}$$



(N+l-1) Received $\left\{y_1,\ldots,y_{N+l-1}\right\}^\dagger$
For the detection of the k-th symbol
k-th symbol
We use the part of observation $ob^k = \left\{y_k,\ldots,y_{k+l-1}\right\}^\dagger$
ob^k
k
k+l-1
For the channel estimation, $\hat{H}^k$, destined for the k-th symbol detection,
0……….……….0
we use the other part of observation y^k:
k
k+l-1
$y^k = y - \left\{0,\ldots,0, ob^{k^\dagger}, 0,\ldots,0\right\}^\dagger$

Figure 5.1: Graphical representation of OS technique.

The new idea introduced in OS requires N-N_t channel estimations. So the receiver complexity may be important. In the following, we propose a simplified version of the Maximum Likelihood (also named pseudo-inverse) channel estimation technique. Recalling that at the n^{th} iteration a classical pseudo-inverse channel estimation using all the observation is defined as

$$\hat{H} = (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y \tag{5.2}$$

Where $\tilde{X}$ denotes the reconstruction of the X matrix with the feedback soft or hard symbols $\{\tilde{x}_k\}$, $k=1 \ldots N$ at $(n\text{-}1)^{\text{th}}$ iteration. The training sequence part in the symbol series $\{\tilde{x}_k\}$ will be unchanged. In our proposition, we introduce:

$$\begin{aligned} \hat{H}^k &= (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y^k \\ &= (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y - (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H \begin{pmatrix} 0 & \text{ob}^k & 0 \end{pmatrix} \end{aligned} \tag{5.3}$$

We obtain

$$\hat{H}^k = \hat{H} - \left(\tilde{X}^H \tilde{X}\right)^{-1} \Delta H^k \ , \quad \begin{pmatrix} \forall k \in [1, N] \\ \text{and k} \notin N_t \end{pmatrix} \tag{5.4}$$

With N_t presents the set of training sequence symbols.

$\left(\tilde{X}^H \tilde{X}\right)^{-1}$ is a square matrix and are calculated once. $\Delta H^k = \tilde{X}^H \ \begin{pmatrix} 0 & ob^k & 0 \end{pmatrix}^T$ contains *l* elements, and is the variable part to be calculated for each symbol. As $\boldsymbol{ob^k}$ has just *l* elements, the correlator is very simple.

So there are the same numbers of operations for channel estimation as the number of symbols in a block. Thus Y^k represents a part of the total observations of *Y* with the exclusion of the part $\boldsymbol{ob^k}$. This can reduce the propagation of error in channel estimation, which saturates the process of convergence of the symbol detector. Thus we obtain the observation separation (OS) in each couple of symbol detector and channel estimator.

5.2 The OS technique in general Narrowband Modulation systems

5.2.1 Conventional BICM Transmitter

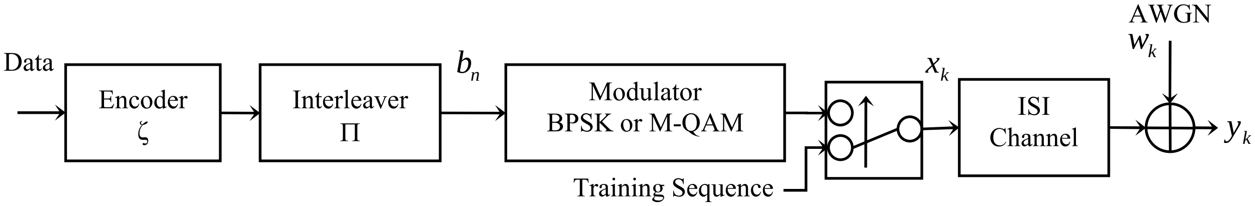


Figure 5.2: BICM Transmitter with an ISI Channel.

Conventional Bit-Interleaved Coded Modulation (BICM) can be modeled as a serial concatenation of a convolutional encoder, random bit interleaver and a modulator as shown in Fig. 5.2. At the transmitter, the information sequence is encoded by a convolutional encoder (ζ) before being bitwise interleaved. The purpose of the bit interleaver (Π) is to break the sequential fading correlation and increase diversity order to the minimum Hamming distance of a code. Next, *m* consecutive bits b_n of the interleaved coded sequence are grouped to form v_k, a channel symbol at the *k*th signaling interval. A modulator maps each v_k to a complex transmitted symbol $x_k = \Omega(v_k)$ chosen from *M*-ary constellation χ where Ω is the labeling map and $M=2^m$.

We assume an ISI channel, the received discrete-time base band signal can be written as

$$y_k = \sum_{i=0}^{l-1} h_i x_{k-i} + w_k, \qquad \forall k \in [1, N+l-1] \tag{5.5}$$

Where x_k are the complex coded symbols in a block with $k=1,\ldots, N$, ($x_k=0$ for $k>N$), *N* is the block length, w_k are white Gaussian noise samples. The *l* complex taps h_i represent the equivalent discrete overall channel impulse response. We consider that we have N_t training sequence symbols located.

5.2.2 Our Proposed Receiver

In the narrowband systems, only the ISI exists. We consider the problem of symbol detection and channel estimation with a coded modulation transmitted over an ISI channel, system presented in Fig. 5.3 and detailed in Fig. 5.4. The channel decoder contains the decoder itself and the re-encoder. Soft or hard feedback is considered.

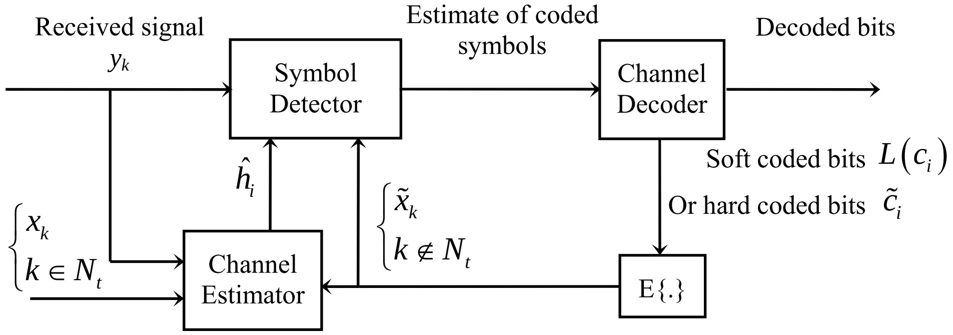

Received signal
y_k
Symbol Detector
Estimate of coded symbols
Channel Decoder
Decoded bits
$\hat{h}_i$
Soft coded bits $L(c_i)$
Or hard coded bits $\tilde{c}_i$
$\begin{cases} \tilde{x}_k \\ k \notin N_t \end{cases}$
$\begin{cases} x_k \\ k \in N_t \end{cases}$
Channel Estimator
E{.}

Figure 5.3: a typical iterative procedure for the channel estimation and symbol detection.

The received signal y_k represented in eq. 5.5, by using vector notation, we can describe the received signal as

$$Y = XH + W \tag{5.6}$$

$$Y = (y_1\ y_2 \dots y_{N+l-1})^T,\ H = (h_0\ h_1\ \dots h_{l-1})^T,\ and\ W = (w_1\ w_2\ \dots w_{N+l-1})^T.$$

With

$$X^T = \begin{pmatrix} x_1\ x_2\ x_3 & \dots & x_N & 0 & \dots\ 0 \\ 0\ x_1\ x_2 & \dots & x_{N-1} & x_N & 0 \\ \dots & & \dots & & \dots \\ 0\ 0\ 0 \dots x_1 \dots x_{N-l+1} & & x_{N-l+2} \dots x_N & & \end{pmatrix}_{l \text{ x } (N+l-1)}$$

In the iterative observation separation technique, at the initial iteration ($n=0$), we use the training sequence which is a part of $\{x_k\}$, $k=1\dots N$, with N_t symbol length to obtain the channel estimate $\hat{H}$.

In the symbol detector, we start by using the interference cancellation with our proposed Observation Separation (OS) technique.

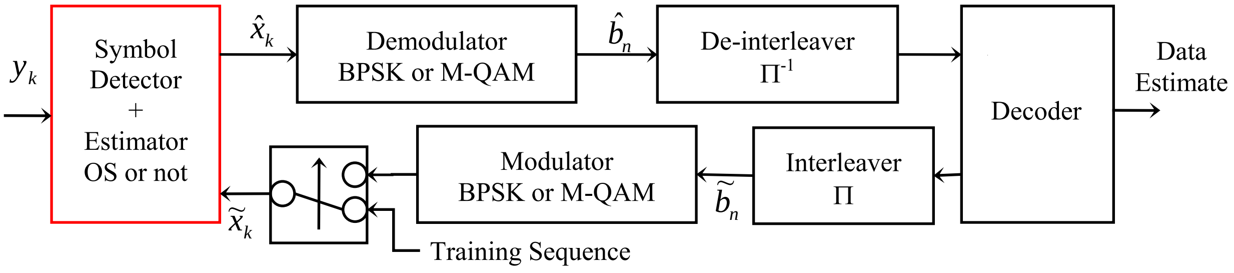


Figure 5.4: The detailed scheme of our Proposed Receiver.

Considering that we have a convolutional encoder, a pseudo-random interleaver and a BPSK or M-QAM modulator at the receiver, after the ISI channel and the AWGN noise we choice to apply a symbol detector and estimator represented in Fig. 5.5, a BPSK or M-QAM demodulator, pseudo-random de-interleaver and a soft or hard decoder follow the detection. And the iterative receiving processes as follow:

Initialization:

1. Make an initial channel estimate based on the known training sequence. The initial channel estimate is based on conventional methods $\hat{H}^{(0)} = \left(X_t^H X_t\right)^{-1} X_t^H Y_t$.
2. Given the channel estimate $\hat{H}^{(0)}$, detect a sequence $\hat{X}^{(0)}$ by ZF-BLE. And then the reliability for detected bits may be calculated by using a demodulator.
3. Decode detected bits after de-interleaving, and return the extrinsic of coded bits to the feedback.

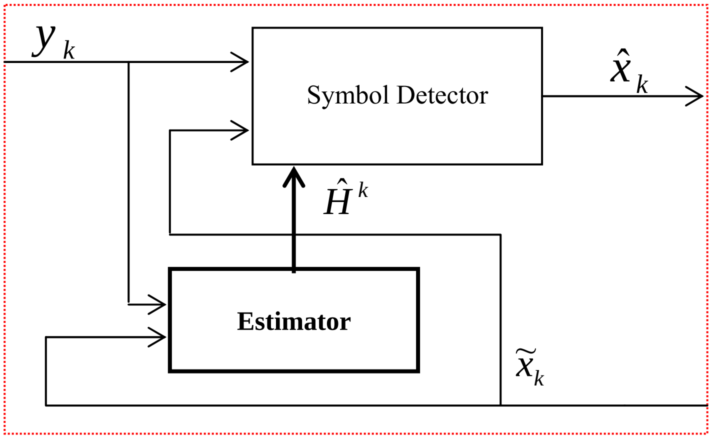


Figure 5.5: Channel estimation and symbols detection.

Iterations: (Classical IC technique)

4. Based on the interleaved extrinsic of coded bits make expectation or mapping operation in the case hard decoder, $\tilde{X}^{(n)}$ for the n^{th} iteration.
5. Update channel estimate using some adaptation rule, e.g. the pseudo-inverse estimator $\hat{H}^{(n+1)}$, based on the returned symbols $\tilde{X}^{(n)}$ and the received symbols Y according to eq. (5.2).
6. Given channel estimate $\hat{H}^{(n+1)}$, update detected sequence $\hat{X}^{(n+1)}$ and its bit wise likelihoods by using the IC, then the same demodulator.
7. De-interleave and decode detected bits…
8. Iterate between steps 4 to 7 as needed.

Iterations: (New OS technique)

The algorithm stays unchanged until the step 4, after that:

5. Based on the returned symbols $\tilde{X}^{(n)}$ and the received symbols Y^k. Update the channel estimate $\left(\hat{H}^k\right)^{n+1}$ for each symbol x_k, (eq. (5.3)).

6. Given channel estimate $\left(\hat{H}^{k}\right)^{n+1}$, update detected symbol $\hat{x}_k^{(n+1)}$ by using the IC, Repeat steps 5 and 6 until detection of all the sequence $\hat{X}^{(n+1)}$.
7. Calculate the bit detected by using demodulator, de-interleaver and decoder on the detected symbols.
8. Iterate between steps 4 to 7 as needed.

5.3 The OS technique in a general CDMA system

Consider the problem of multi-user detection depicted in Fig. 5.6 assuming the presence of ISI and Multi-user Access Interference (MAI).

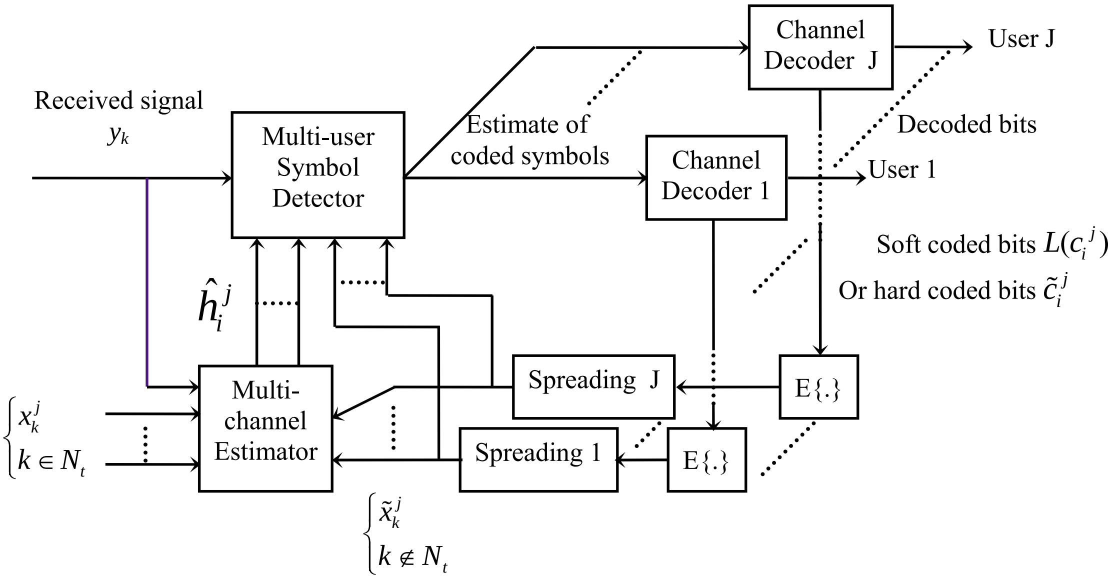

Channel Decoder J
User J
Received signal y_k
Multi-user Symbol Detector
Estimate of coded symbols
Decoded bits
Channel Decoder 1
User 1
Soft coded bits $L(c_i^j)$
Or hard coded bits $\tilde{c}_i^j$
$\hat{h}_i^j$
Multi-channel Estimator
Spreading J
Spreading 1
E{.}
E{.}
$\begin{cases} x_k^j \\ k \in N_t \end{cases}$
$\begin{cases} \tilde{x}_k^j \\ k \notin N_t \end{cases}$

Figure 5.6: a typical iterative procedure for the channel estimation and symbol detection in a CDMA system.

The received signal y_k can be written as

$$y_k = \sum_{j=1}^{J} \sum_{i=0}^{l-1} h_i^{(j)} x_{k-i}^{(j)} + w_k, \qquad \forall k \in [1, N+w-1] \qquad (5.7)$$

where $x_{k-i}^{(j)}$ are the chips of the spread symbol or the chips of the training sequence of user j with j=1, …, J, in a block with k=1, …, N and N corresponding to the number of transmit chips ($x_k^{(j)}$=0 for k > N), w_k are filtered white Gaussian noise samples. The l complex taps $h_i^{(j)}$ represent the equivalent discrete overall channel impulse response (in chip) of user j and are supposed constant in the period of one block. N_t presents the index sub set where N_t training sequence symbols are located.

After multi-user symbol detection, the symbols of each user $\beta_q^{(j)}$ are decoded separately. The channel coding can be applied on more than one block. The channel decoder contains the de-interleaver, the decoder itself, the re-encoder if hard decision is made and the re-interleaver.

When soft feedback is considered, the decoders output the a posteriori log ratio (also called log likelihood ratio) of each coded bit; then the expectation of the coded symbol is calculated. For the hard feedback, the expectation is just a bit to symbol mapping. Then the soft or hard feedback symbols of all the users are re-spread into Q chips separately and fed into the channel estimator and multi-user detector. By using vector notation, the received signal can describe as

$$Y = XH + W \tag{5.8}$$

$$Y = (y_1\ \mathrm{y}_2 \ldots \mathrm{y}_{N+l-1})^T,\ \ and\ W = (w_1\ \mathrm{w}_2\ \ldots\ \mathrm{w}_{N+l-1})^T.$$

$$X = (X^{(1)}\ X^{(2)}\ \ldots X^{(J)})$$

With

$$X^{(j)} = \begin{pmatrix} x_1^{(j)}\, x_2^{(j)}\, x_3^{(j)} & \cdots & x_N^{(j)} & 0 & \ldots\ 0 \\ 0\ \ x_1^{(j)}\, x_2^{(j)} & \cdots & x_{N-1}^{(j)} & x_N^{(j)} & 0 \\ \ldots & & \ldots & & \ldots \\ 0\ \ 0\ \ 0\ \ \cdots x_1^{(j)} & \cdots x_{N-l+1}^{(j)} & x_{N-l+2}^{(j)} & \cdots x_N^{(j)} & \end{pmatrix}^T_{l\ \mathrm{x}\ (N+l-1)}$$

$$H = (h^{(1)T}\ h^{(2)T}\ \ldots h^{(J)T})^T,\quad With\ \ \mathrm{h}^{(j)} = (h_0^{(j)}\ h_1^{(j)}\ \ldots\ h_{l-1}^{(j)})^T$$

In the iterative observation separation technique, at the initial iteration ($n=0$), we use the training sequences which are parts of $\{x_k^{(j)}\}$, $k=1, \ldots, N$, and $j=1, \ldots, J$, with N_t chip length to obtain the channel estimate $\hat{H}$. The multi-user symbol detector delivers the estimate of the series $\{\eta_q, q=1, \ldots, \lfloor N/Q \rfloor$ and $q.Q \notin \boldsymbol{N}_\mathrm{t}\}$ with the symbol set $\eta_q = \{\beta_q^{(j)}, j=1, \ldots, J\}$ which corresponds to the chips $\{\{x_k^{(j)}, x_{k+1}^{(j)}, \ldots, x_{k+Q-1}^{(j)}\}, j=1, \ldots, J\}$ with $k=(q\text{-}1).Q+1$.

For the n^{th} iteration ($n \geq 1$), we propose to use the observation vector $\mathbf{ob^q} = (y_k, y_{k+1}, \ldots, y_{k+Q\text{-}1+l\text{-}1})$ for detecting the q^{th} symbol set η_q and the observation vector Y^q defined by $Y^q = Y - \left(0\ \ \mathrm{ob}^q\ \ 0\right)^{\mathrm{T}}$. For making the channel estimate $\hat{H}^q$ which is used just in the detection of η_q. The vectors $\mathbf{ob}^\mathrm{q}$ and Y^q are mutually independent. Thus we make the observation separation (OS) in each couple of multi-user symbol detection and channel estimation.

In the multi-user symbol detector, there are J parallel interference cancellation (PIC) processes; each process will output one symbol. The j^{th} process starts by using the interference cancellation with the parts of feedback chips at $(n\text{-}1)^{\text{th}}$ iteration: $\{\tilde{x}_{k-l+1}^{(j)}, \tilde{x}_{k-l+2}^{(j)}, \ldots, \tilde{x}_{k-1}^{(j)}, \tilde{x}_{k+Q}^{(j)}, \ldots, \tilde{x}_{k+Q-1+l-1}^{(j)}\}$ and $\{\tilde{x}_{k-l+1}^{(j')}, \tilde{x}_{k-l+2}^{(j')}, \ldots, \tilde{x}_{k+Q-1+l-1}^{(j')}\}$ with $j' \neq j$ and the corresponding $\hat{H}^q$. If all the return symbols are correct and $\hat{H}^q$ is good, in the j^{th} process, we have the following interference free samples:

$$s_{k'}^{(j)} = \sum_{i=0}^{l-1} h_i^{(j)} x_{k'-i}^{(j)} + w_{k'}\ ,\ \ \forall k' \in [k, k+Q+l-2] \tag{5.9}$$

$$and \quad x_{k'}^{(j)} = 0 \quad if \quad k' < k \ or\ k' > k+Q-1$$

Then a matched filter (including the channel response and the de-spreading) is applied.

The new idea introduced here requires $(N\text{-}N_t)/Q$ channel estimations. So the receiver complexity may be important. In the following, we propose a simplified version of the Maximum

Likelihood (also named pseudo-inverse) channel estimation technique. Recalling that at the n^{th} iteration a classical pseudo-inverse channel estimation using all the observation is defined as

$$\hat{H} = (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y \tag{5.10}$$

Where $\tilde{X}$ denotes the reconstruction of the X matrix with the feedback soft or hard chips $\{ \tilde{x}_k^{(j)} \}$, k=1, …, N at $(n\text{-}1)^{\text{th}}$ iteration. The training sequence parts in the symbol series $\{ \tilde{x}_k^{(j)} \}$ will be unchanged. In our proposition, we introduce:

$$\begin{aligned} \hat{H}^q &= (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y^q \\ &= (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H Y - (\tilde{X}^H \tilde{X})^{-1} \tilde{X}^H \begin{pmatrix} 0 & \mathrm{ob}^q & 0 \end{pmatrix} \end{aligned} \tag{5.11}$$

With

$$\hat{H}^q = \hat{H} - \left(\tilde{X}^H \tilde{X} \right)^{-1} \Delta H^q \ , \quad \begin{pmatrix} \forall q \in [1, \lfloor N/Q \rfloor] \\ and\ q.Q \notin N_t \end{pmatrix} \tag{5.12}$$

And

$$\Delta H^q = \tilde{X}^H \ \left(0 \ \ ob^q \ \ 0 \right)^T \tag{5.13}$$

($\boldsymbol{N}_t$ is supposed same for all the users), where $\hat{H}$ and $\left(\tilde{X}^H \tilde{X} \right)^{-1}$ are both calculated only once. ΔH^q (with $J.l$ elements) is the variable part to be calculated for each set of J symbols. As $\mathbf{ob}^q$ has just Q-1+l elements, the correlator represented by (5.13) remains relatively simple. So we reduced the calculation complexity.

5.4 Applications, Simulations and Results

5.4.1 UMTS – TDD Application (Multi-user Application)

UMTS-TDD is a time division duplex CDMA block coded system defined in the standard of the 3-rd generation mobile communication system. We apply the iterative OS technique to the uplink 12.2 kbps service presented in Fig. 5.8.

The radio transmission of UMTS-TDD is a stream of 10 ms frames with 15 time slots per frame, presented in Figure 5.7. A time slot corresponds to the transmission block of our chapter. One slot per frame is reserved for the 12.2 kbps service. This service contains the information data and the control data. A block of 244 bits of information data is applied to a CRC attachment for block error detection, then encoded, interleaved, punctured with 5% rate and finally distributed over two consecutive time slots with 380 bits in each. A rate 1/3 convolutional coder with generator $(557, 663, 711)_{\text{oct}}$ is used. A block of 100 bits of control data is applied also to a CRC attachment, then encoded, interleaved and finally distributed over four consecutive time slots with 90 bits in each. The same convolutional coder as for the information data is used. The resulting multiplexed data in each time slot is interleaved again and combined with 18 signaling bits to form a block of 488 bits and thus a block of 244 QPSK symbols. The useful system parameters are the spreading factor Q=8, the midamble (training sequence) chip length N_t=512, the channel chip length l=57, the guard period equal to 96 chips, the chip rate equal to 3.84 Mcps and the roll-off = 0.22.

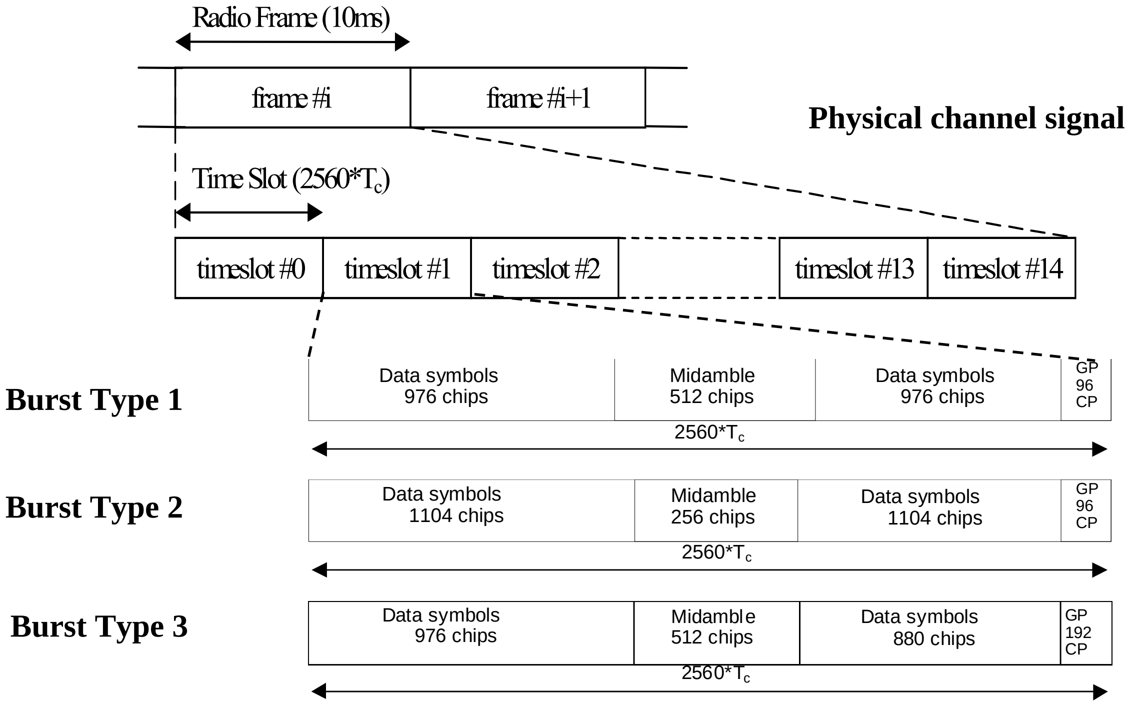


Figure 5.7: The Radio Frame in an UMTS-TDD System.

Simulation Results:

When we consider the ITU "Vehicular A", the channel evolves normally with the mobile speed *v* for four consecutive frames and is made completely independent from four frames to four other frames. We consider also the two paths channel having equal gain, without fading and with the delay between paths equal to two symbols.

The block error rate (BLER or more currently called as FER: frame error rate) and the bit error rate (BER) of the information data are considered. The Viterbi and the log-MAP channel decoders are used for the hard and the soft decision feedbacks respectively. In a performance figure, for the soft feedback, the "Soft" word is always associated with the performance curve; but not for the hard feedback.

At the initial iteration ($n=0$), the pseudo-inverse channel estimation based on the training sequence is employed; and except indication, the matched filter is used for the multi-user detection of OS and PIC techniques. For the classical bootstrap technique named Classic, the multi-user detection technique is always the same for all the iterations. For the n^{th} iteration ($n\geq 1$) of a non-OS technique, the classical pseudo-inverse channel estimation based on the whole return symbols is considered.

For "Vehicular A" channel with v=3 km/h, $J=4$, and $n=0$ (iter0) and 4 (iter4), we depict, in Fig. 5.9, the FER and BER curves versus Eb/N0 of the information bits. We have considered the iterative OS and PIC procedures with hard and soft feedback. We note that soft OS gives the best performance, the hard OS is the second and the gain of soft decision feedback compared to the hard one is small, less than 0.1 dB for OS or 0.3 dB for PIC. For the iterative OS procedure, when *n* changes from 0 to 4, at Eb/N0=10 dB, the FER changes from $1.8\text{x}10^{-1}$ to $9.6\text{x}10^{-3}$; and the BER changes from $4.9\text{x}10^{-2}$ to $2.3\text{x}10^{-3}$.

In Fig. 5.10, we show the FER curves of OS (zf-ble), PIC (zf-ble) and Classic (zf-ble) with *J=4*, *n=0* and 4 and *v*=50 km/h. Note that for a given iterative procedure the text in the parenthesis ( ) corresponds to the detection technique used at the initial iteration. We note that the performance improvement of OS compared to Classic comes first from the use of the PIC structure instead of the ZF-BLE structure and second from the separation of the used observations for the channel estimation and for the symbol detection.

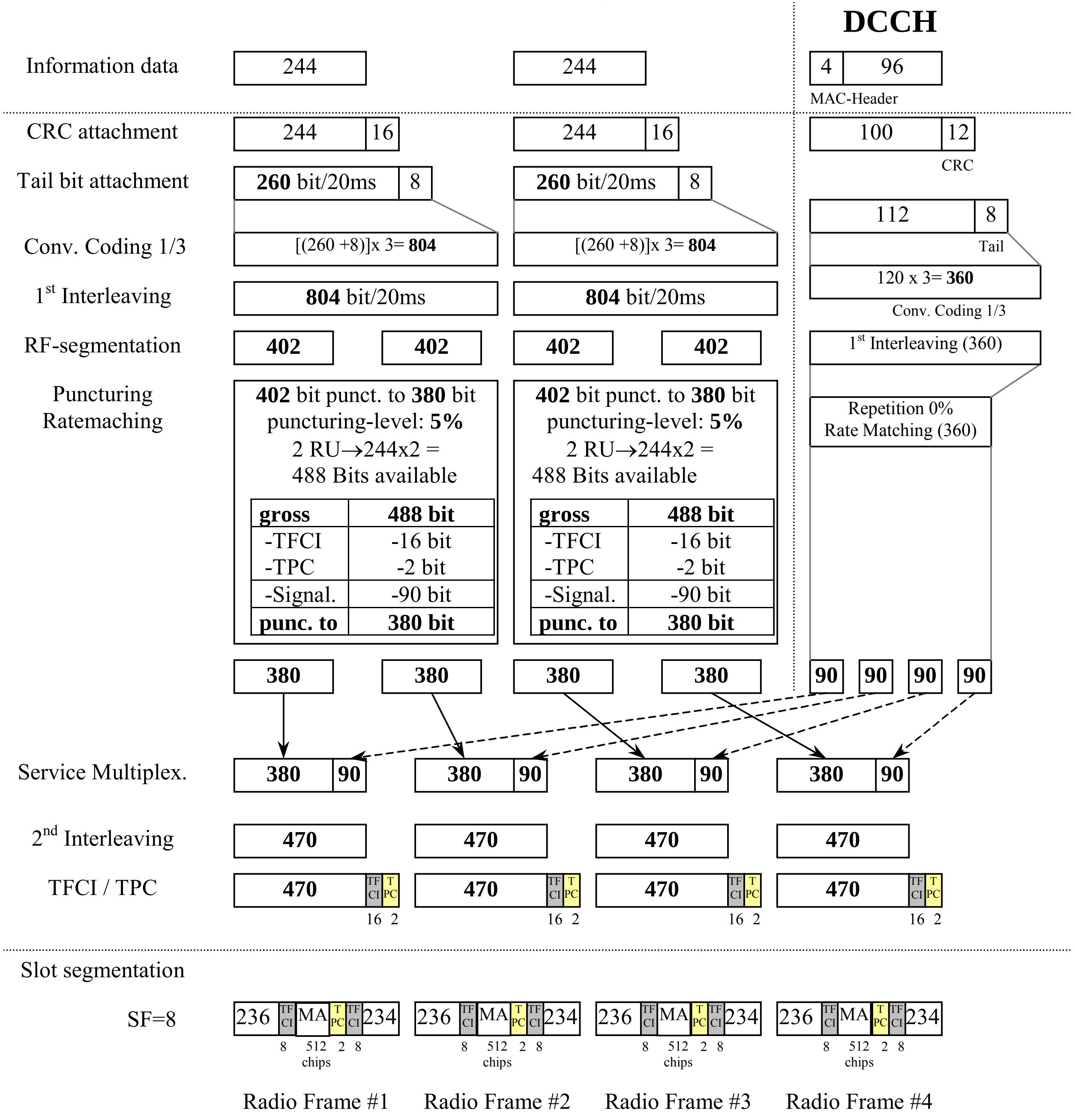


Figure 5.8: Uplink 12.2 kbps service.

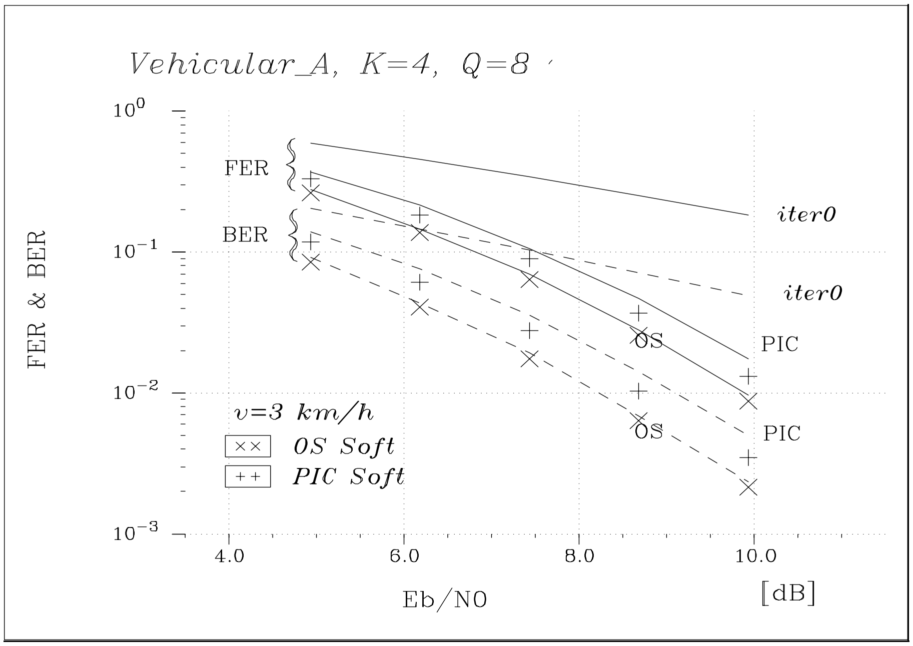


Figure 5.9: Performance of the UMTS-TDD uplink 12.2 kbps service with the soft and hard feedback comparison and *J=4*.

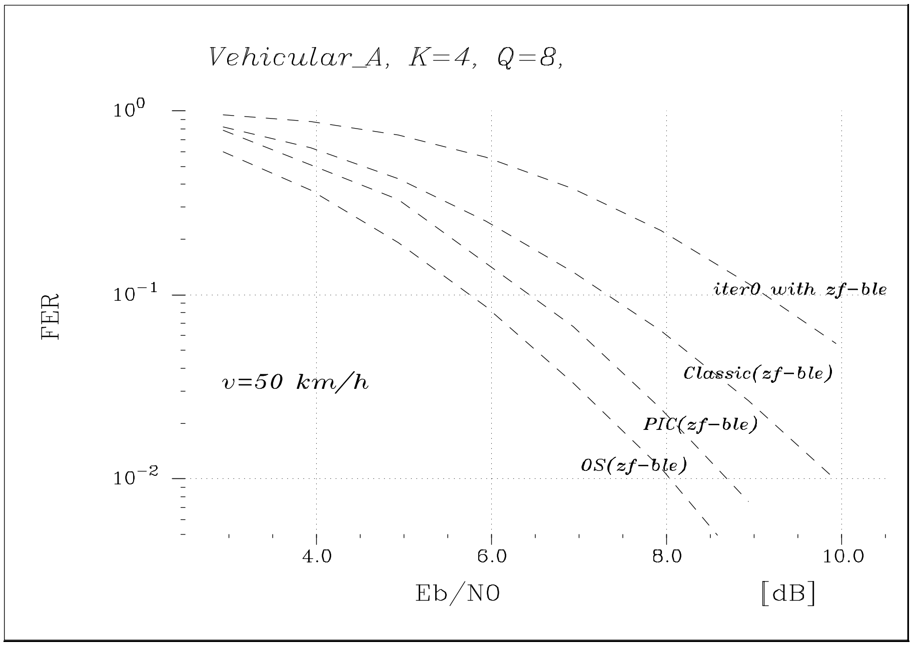


Figure 5.10: FER performance of the OS(zf-ble), PIC(zf-ble) and Classic(zf-ble) for the UMTS-TDD uplink 12.2 kbps service.

In Fig. 5.11, for J=1, Q=8 and two flat paths channel; we show the BER performance of the Classic (mmse-ble), the IC and the OS. For the hard decision feedback, the gain of OS compared to IC at BER=10^{-3} is 0.85 dB.

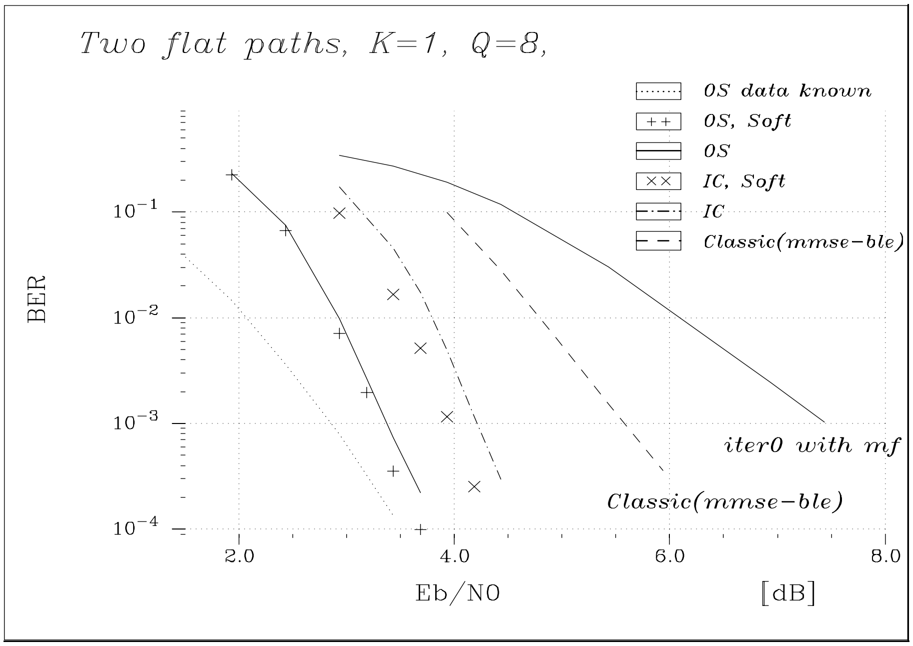


Figure 5.11: BER performance comparison of different techniques for the UMTS-TDD uplink 12.2 kbps service and J=1.

5.4.2 Narrowband Application (Single user Application)

To be able to analyze the narrowband modulation behavior, we make some adaptations on the UMTS-TDD 12.2 kbps service of the previous paragraph. In fact, with the same slot structure and the QPSK modulation, we change the spreading factor to Q=1, the training sequence length $N_t \leq 512/8 = 64$. With channel chip length l=57 in the UMTS-TDD case, the channel symbol length l should be 8; but we set l as parameter for study. The guard period is greater than l.

To make the training sequence, we use the constant amplitude zero auto-correlation (CAZAC) sequences with length of N_{cazac} being multiple of four; the prefix of (N_t - N_{cazac}) bits corresponds to the last (N_t - N_{cazac}) bits of a CAZAC sequence.

And we apply the algorithm described in the previous sections to a system with a high level coded modulation M-QAM.

A block of 632 bits coded with a rate 1/3 convolutional coder with generator $(557, 663, 711)_{\text{oct}}$. The coded bits are interleaved with a pseudo-random interleaver having a block length equal to 1920 bits which are modulated by a M-QAM modulator to (1920/$\log_2$M) symbols with currently used Gray mapping. We add a midamble training sequence with length Nt=N_{cazac}+l-1 BPSK symbols adjustable to estimate l channel taps. N_{cazac} and l leaved as a parameter to study. The result is transmitted over an ISI channel (2 channel taps, Proakis A channel: 11 taps) and recuperated at the receiver with the symbol detector and estimator which was described above, after that the symbols detected are demodulated, and the obtained bits are de-interleaved and decoded with Viterbi decoder when the hard feedback is considered and with MAP decoder in the case of soft feedback. Here some of our simulation results:

Simulation Results:

For the QPSK modulation, the OS and IC BER curves are given for l=8 and 17 in the Fig. 5.12 and for N_{cazac}=56 and 16 in Fig. 5.13. We note that greater the number of possible channel taps is and smaller the training sequence is, more important the gain of OS versus IC can be obtained. In Fig. 5.14, we give the BER and FER curves for hard and soft feedbacks of OS and IC techniques. The OS performs always over IC whatever the feedback type.

For the M-QAM modulation, curves are given for l=8, 18 and 24 and for N_{cazac}=16, 36 and 48. In Fig. 5.15, we present the BER and BLER performances for a 16-QAM modulation over an invariant time channel with length of two; the training sequence contains 36 BPSK symbols to estimate 18 channel taps, to compare the performances with soft feedback versus hard feedback. This figure shows that for the soft feedback like as for the hard feedback, the OS technique outperforms the IC technique in the case of BER and BLER. An important point here is that the hard performances are very close to the soft performances, so only the hard performances are represented in the following results.

We present in Fig. 5.16, the BER and the BLER performances of IC and OS with 16-QAM over a Proakis A channel (11 taps), the same structure of training sequence like the previous results. Furthermore we represent the performances of the same systems without updating the channel estimation with iterations, in other words we estimate one time the channel with training sequence which we called TE procedure. This figure shows that the OS and the IC procedure outperform the TE procedure by 2.5 dB at BER=10^{-4}, in the same time the OS is better than the IC by 0.25 dB. We can conclude here that the OS performs well over long channel.

Fig. 5.17 shows the BER performances of TE, IC and OS with 16-QAM over a two channel taps but the length of training sequence is 48 structured to estimate 8 channel taps, furthermore we represent the BER performances when the channel is perfectly known. We can conclude from this figure the OS outperforms IC by up to 0.25 dB at BER=10^{-5}. The main remark here is the convergence of the OS procedure to system with channel known.

The BER performances of IC and OS with QPSK, 16-QAM and 64-QAM over a two channel taps, the length of training sequence is 48 and can estimate 8 taps are represented in Fig. 5.18. This figure shows that like in the case of QPSK, 16-QAM and the 64-QAM modulation the OS procedure outperforms the IC procedure, and this gain increase with the increasing of the level of modulation. At BER=10^{-4} the gain is 0.2 dB for QPSK, 0.4 for 16-QAM and ~1dB for 64-QAM.

In Fig. 5.19, we present the BER performances of IC and OS with 16-QAM over a two channel taps, we increase the length of training sequence with maintaining a structure which estimate $l=8$ channel taps. This figure shows that in all the cases the OS procedure outperforms the IC procedure and this gain increase with the decreasing of the length of training sequence. At BER=10^{-4}, the gain is less than 0.25 when N_{cazac}=48, it is great than 0.35 when N_{cazac}=36 and it is great than 0.65 when N_{cazac}=16.

Finally, we present in Fig. 5.20, the BER performances of IC and OS with 16-QAM over a two channel taps, we leave the length of training sequence fixed to N_{cazac}=48 but with changing the structure to increase the number of taps estimated. We remark that in all the cases the OS procedure outperforms the IC procedure and this gain increase with the increasing of the number of taps estimated *l*. At BER=10^{-4}, the gain is less than 0.25 when *l*=8, it is equal to 0.5 when *l*=18 and it is great than 0.75 when *l*=24.

5.5 Conclusion

In this Chapter, the observation separation technique on the joint iterative channel estimation and symbol detection technique has been considered for the CDMA, the narrowband QPSK mobile and a high level modulation M-QAM systems. For the return loop, the soft or hard channel decoder output is considered. The OS concept separates the observation used for the estimator from the observation used for the detector. The performance obtained is very attractive.

Our OS is realized by associating an adaptive IC (or PIC when MAI is presented) structure for the symbol detection with a modified pseudo-inverse channel estimation structure for each symbol. Nevertheless this realization schema on the OS concept is not unique and that the research on a better solution is open. For a real conception like the 2nd and 3rd generation and for complexity reason of decoding, we propose the hard decoding feedback (Viterbi decoding).

We have applied this technique to the 12.2 service of the UMTS-TDD uplink system and also to the 12.2 service of the associated narrowband QPSK system and a high level modulation M-QAM with a rate 1/3 convolutional code, a midamble training sequence of length N_t to estimate a *l* channel taps estimation. These two parameters are leaved to study their effect on the performance of the OS procedure. The performance obtained is very promising.

Depending on the scenario considered, simulation results have shown up to 1dB gain for the OS techniques compared to classical IC techniques, this is very attractive performance.

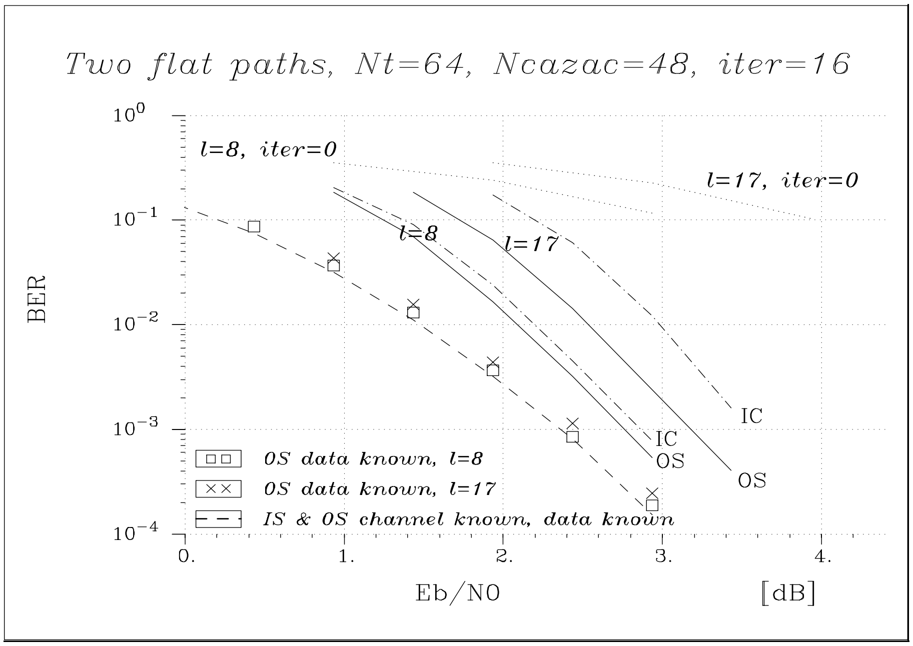


Figure 5.12: BER performance of OS and IC techniques in a QPSK system for different *l*.

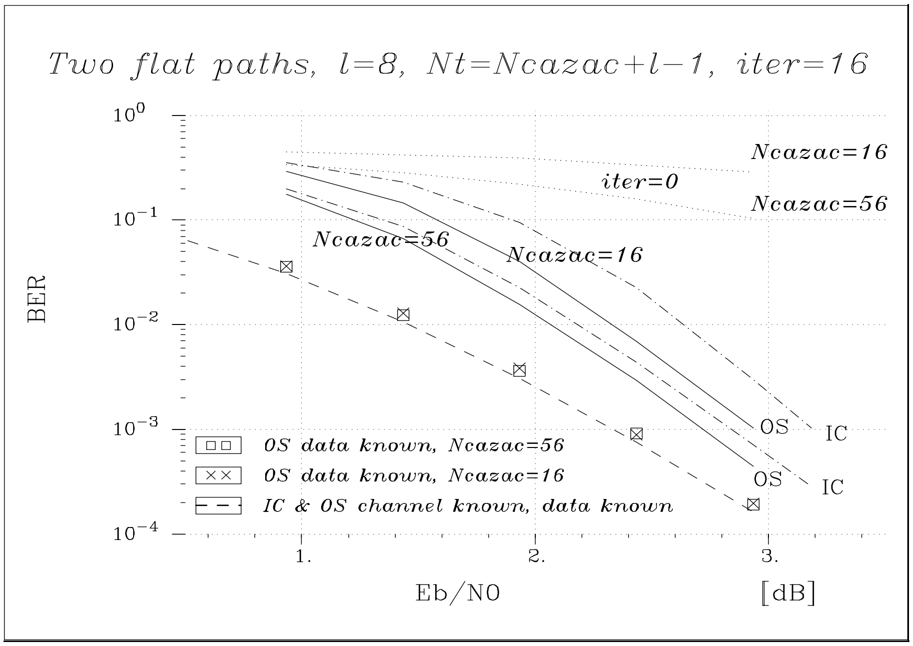


Figure 5.13: BER performance of OS and IC techniques in a QPSK system for different N_{cazac}.

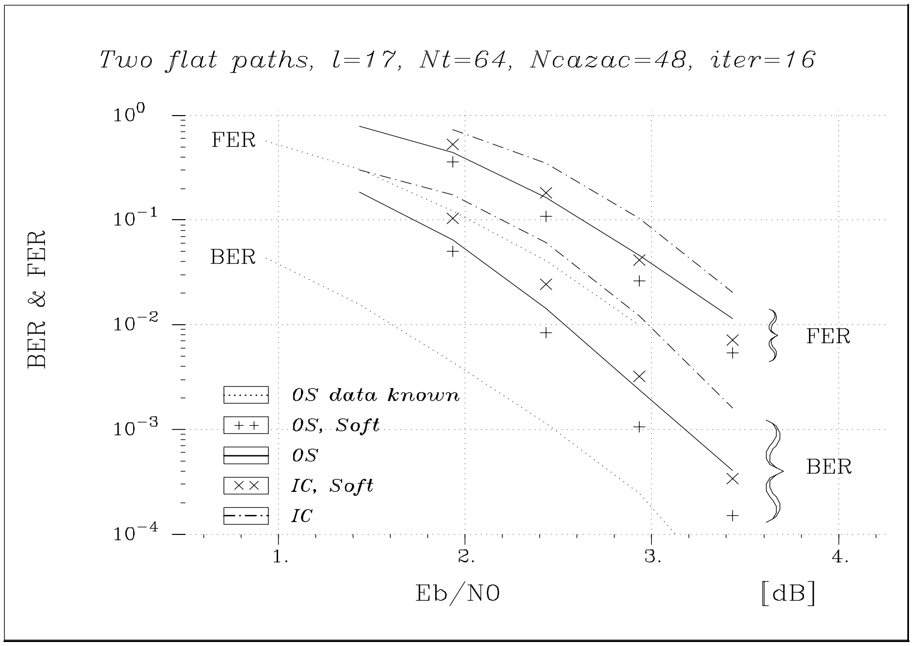


Figure 5.14: FER and BER performance of OS and IC techniques in a QPSK system for soft and hard decision feedback.

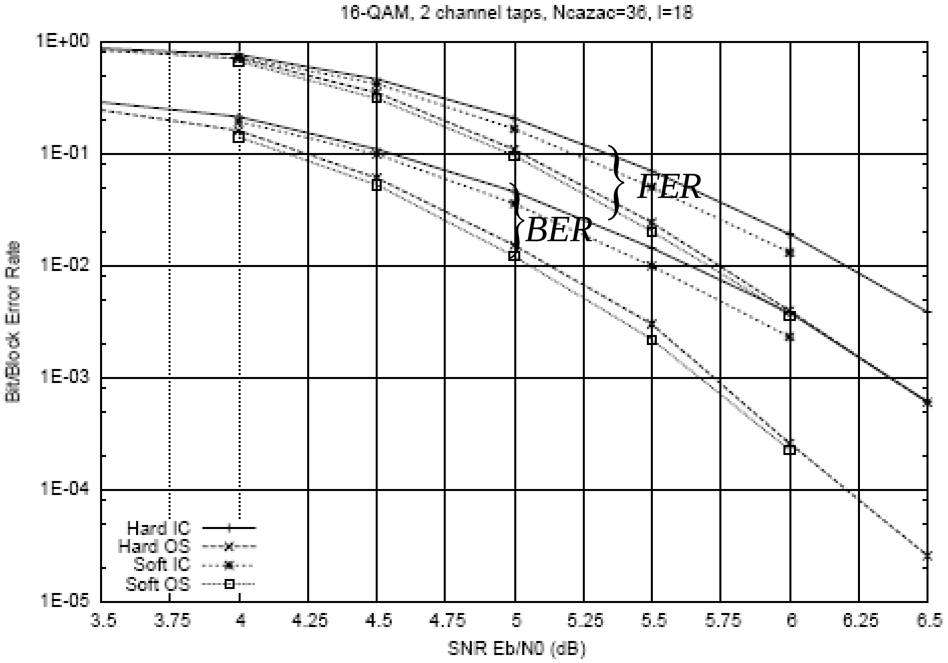


Figure 5.15: FER and BER performance in 16-QAM for soft and hard decision feedback.

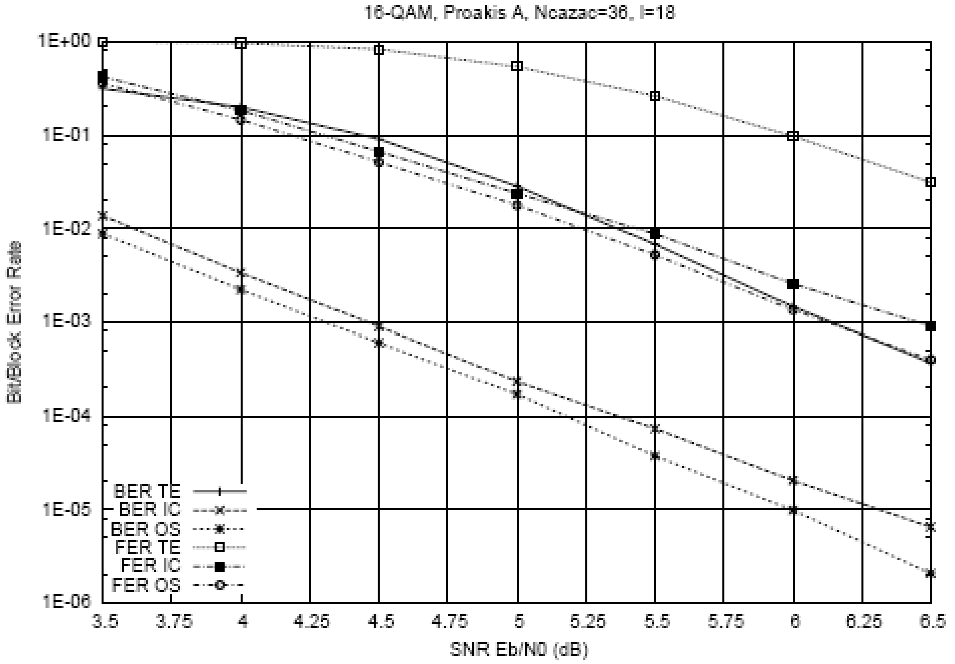


Figure 5.16: FER and BER performance in 16-QAM for Proakis A channel.

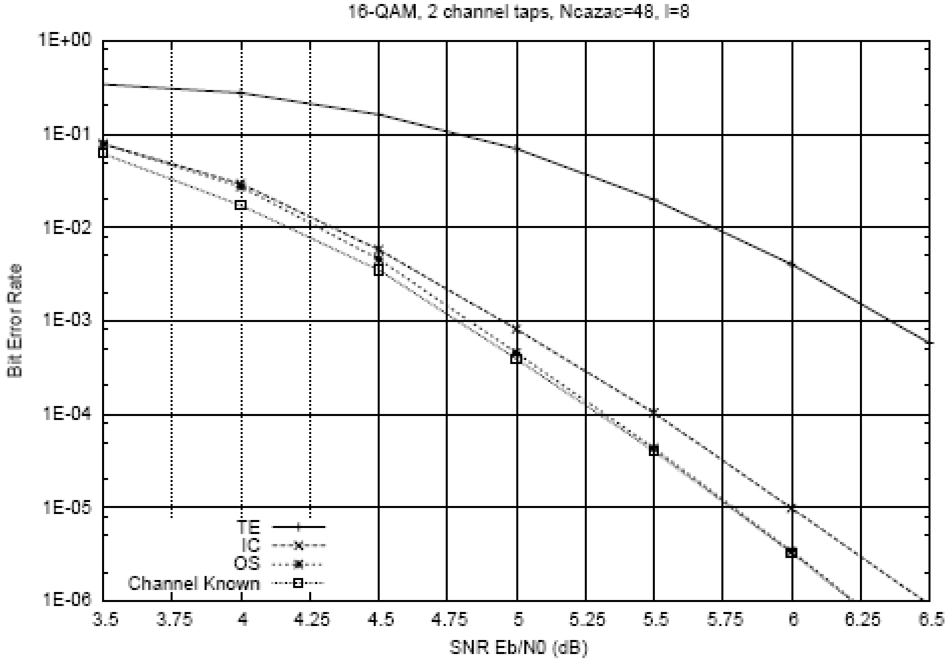


Figure 5.17: BER performance in 16-QAM for 2 channel taps with N_{cazac}=48, l=8.

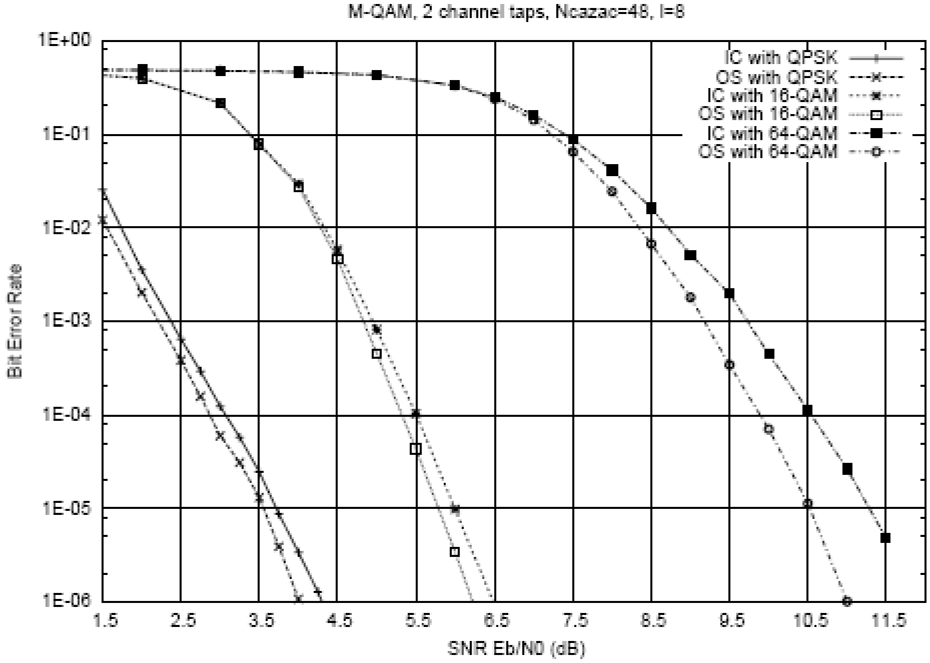


Figure 5.18: BER performance in QPSK, 16-QAM and 64-QAM systems.

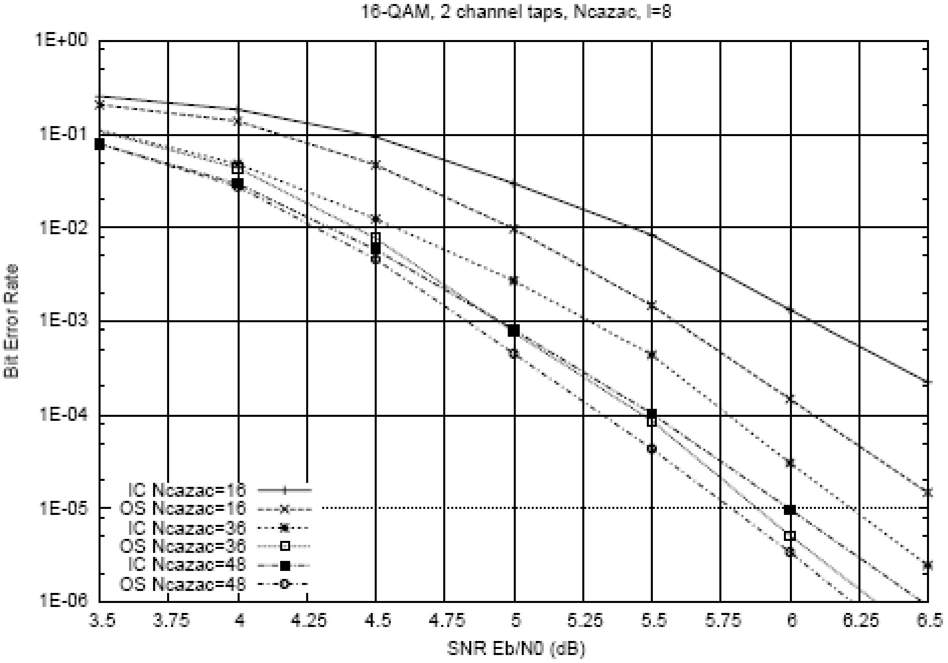


Figure 5.19: BER performance in 16-QAM system with varying N_{cazac}.

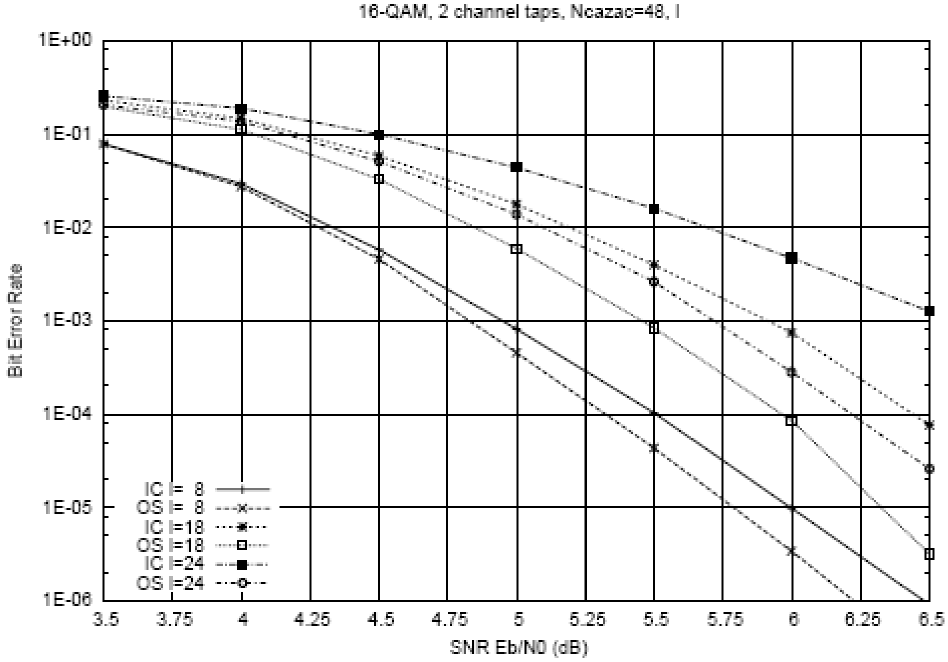


Figure 5.20: BER performance in 16-QAM system with varying *l*.

Chapter 6
Mapping Optimization with Iterative OS Technique

Iterative decoding is traditionally applied with either a parallel or a serial concatenation of at least two codes. At the transmitter the two encoders are separated by an interleaver. At the receiver the decoding is iterated along two component codes for parallel concatenated system, or the inner and the outer decoding path for the serially concatenated system respectively. Parallel concatenated codes applying systematic recursive convolutional codes were first referred to as "turbo codes". As noted in chapter 5, the "turbo principle" can be used not only with traditional concatenated coding schemes, but is more generally applicable to several other algorithms that can be found in modern digital communications, e.g. iterative equalization or multi-user detection.

Two principals approach for optimizing the labeling map; Ten Brink and al. use mutual information parameters, which allow distinguishing exactly each mapping performances, however, the computation of this parameter requests an intensive simulation and depends of sight functioning point. In these papers we find a briefly description of an approach to obtain a design rule based on mutual information. This significantly reduces the burden of performing exhaustive simulations to find good mapping since mutual information is very easy to calculate by numerical integration. Gorokhov approach the effect of signal labeling through the error rate analysis of maximum likelihood demodulation/decoding on one hand, and the statistical analysis of de-mapped bit metrics at the first demodulation iteration on the other hand. These two features lead Gorokhov to propose two design criteria for optimization labeling, based to the computing of specific distances of constellation. The choice of mapping is then achieve without any notion of functioning point. However, these design criteria give preference to asymptotic performances than that of iterative process during the convergence, which is not always desirable. In these papers we find a mapping design that relies upon the common observation that iterative decoding procedures approach the behavior of the optimal decoders as SNR grows, thus we obtain the first design criterion that optimize the labeling map to ensure a good performance of the iterative decoding, at a relatively high SNR. The second design criterion aims at an improved SNR threshold of practically used iterative decoders.

In this chapter, we suggest a design of optimal mapping in BICM-ID scheme all that applied to an OS channel estimation and IC procedure system. This means that we applied the turbo principle in two cases: turbo demodulation and turbo detection symbol. We do not refer to Ungerboeck's trellis coded modulation (TCM) as can be found since there is no "inner encoder" which would add redundant information. The system can be regarded as two serially concatenated iterative decoding schemes whereby the inner decoder is replaced by the soft demapping device (also referred to as 'demapper') in the turbo demodulation, and by the detection symbols device in the turbo detection. We apply the concept named Observation Separation (OS) technique, first introduced in chapter 5 for the CDMA applications and narrowband system, which partially decouples the observation used for channel estimation from the observation that allows the symbol detection; in this chapter, the symbol can belong to a set of high level modulations (16-QAM) and the channel is considered like a time-invariant ISI channels. Chapter 5 show that the OS technique

1. Exhibits better performances when used with large constellation compared to IC,
2. Can approach and converge to the performance when the channel in perfectly known. See Fig. 6.1.

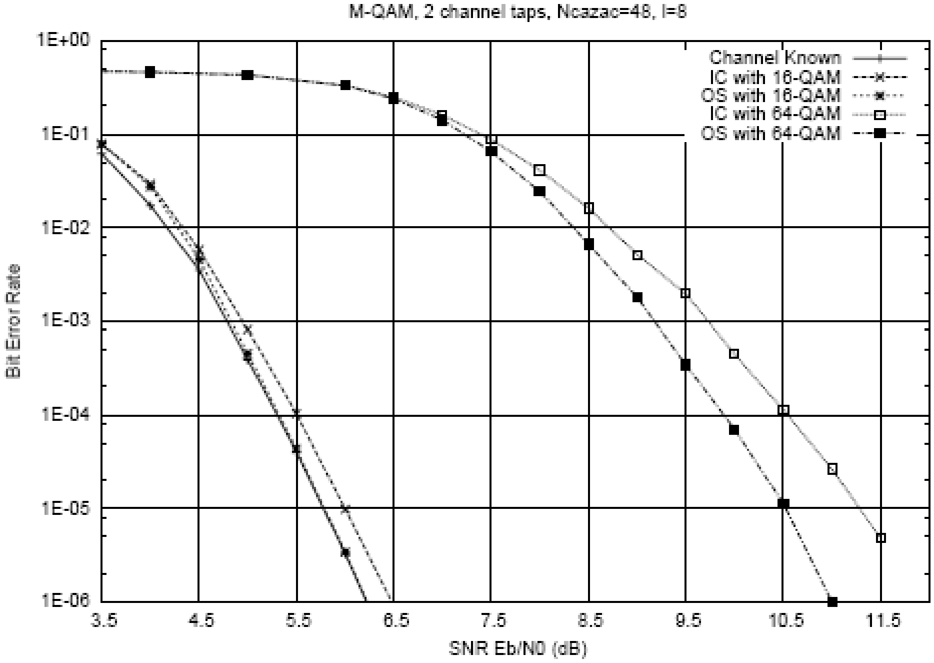


Figure 6.1: Performances of OS technique for 16QAM and 64QAM

Based on these results close to perfect channel, our main objective is to outperform our first iteration, which leads to improve the performance when the channel is known, by optimization mapping and de-mapping. We'll show how iterative demapping reduces the bit error rate in this system described above.

We design an optimized binary mapping of the QAM constellation, in order to increase coding gain, which assumes perfect a priori information. This expression is called *the genie method*. Like if we consider the presence of a genie delivering perfect a priori information. This optimized mapping can be used for some practically important cases.

This algorithm is applied to the case of high level modulation (16-QAM…) and our optimized mapping is compared to conventional mapping (Gray, Set partionning…) in the two cases classical IC techniques and OS techniques, and the results of simulations are presented.

6.1 Iterative decoding of BICM

Let us assume perfect channel state information (CSI) so that h_i are perfectly estimated by the OS concept at the receiver. That's means the output of the IC equalizer and OS estimator $\hat{x}_t$ depends only of one coefficient ρ_t:

$$\hat{x}_t = \rho_t x_t + \eta_t \tag{6.1}$$

Where t corresponding to one channel use, ρ_t is a type of a Rayleigh distributed fading coefficient with E(ρ_t^2) = 1 and η_t is a complex white Gaussian noise sample with the variance of $N_0/2$. For the AWGN channel, $\rho_t = 1$. In our case, $\rho_t =$ cte.

In this section, we review the soft-decision bit metric calculation for iterative decoding (BICM-ID). Our receiver has two main elements as described in Fig. 6.2: an APP QAM-detector that acts as a soft output demodulator, and a SISO decoder for the code ζ. An iterative joint detection and decoding process is based on the exchange of soft values between the SISO QAM-detector and the SISO convolutional decoder. The SISO detector computes the *extrinsic* probabilities $\xi(b_n)$ via a classical sum product expression including the conditional likelihoods $P\left(\hat{x}_t \middle| x_t\right)$ and the *a priori* probabilities $\pi(b_n)$ for the coded bits fed back from the SISO decoder.

The SISO detector computes the *extrinsic* information, which corresponds to the *extrinsic* probability that the j^{th} coded bit equals 1, as given in the following normalized marginalization eq.(6.3).

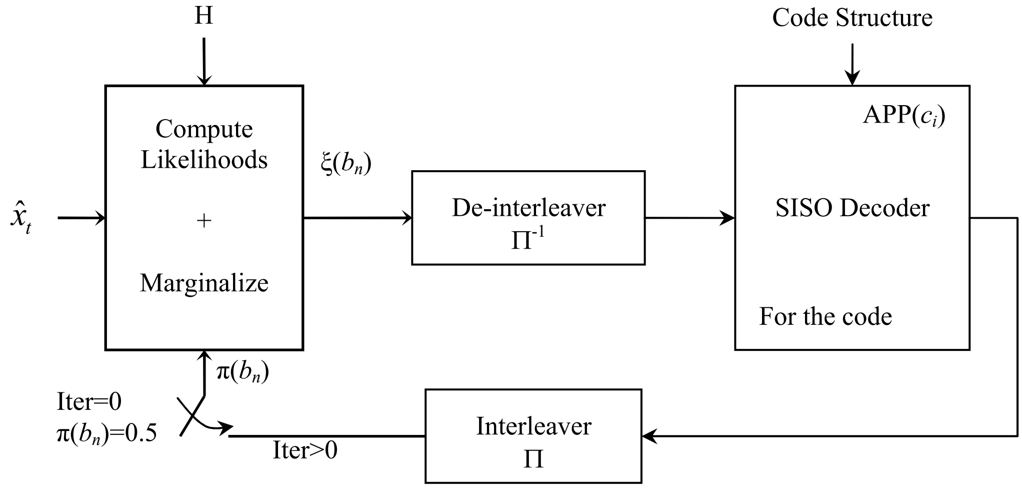


Figure 6.2: Turbo Demodulation. (BICM-ID receiver with soft decision feedback)

$$\begin{aligned}\xi\left(b_n\right) = P\left(b_n = 1 \middle| \hat{x}_t\right) &= \sum_{x_t \in \chi_1^n} P\left(x_t \middle| \hat{x}_t\right)\\ &= \sum_{x_t \in \chi_1^n} P\left(\hat{x}_t \middle| x_t\right).P\left(x_t\right)\end{aligned} \tag{6.2}$$

Where χ_1^n is the subset of χ , whose label has the binary value 1 at the *n*th bit position. The *a priori* probability P(x_t) for the symbol is unavailable on the first pass of demodulation. Therefore, an equally likely assumption is made and π(b_n)=1/2 is used as the *a priori* probability for coded bits input to the SISO detector, which then generates the *extrinsic* probabilities for coded bits.

$$P(x_t)=\prod_{n=1}^{m}\pi(b_n).$$

Using this; the *extrinsic* bit probabilities put out from the SISO detector can be written as

$$\xi(b_n)=\sum_{x_t\in\chi_1^n}P(\hat{x}_t|x_t).\prod_{\substack{r\neq n\\r=1}}^{m}\pi(b_r).$$

And its normalized marginalization:

$$\xi(b_n)=\frac{\sum_{x_t'\in\chi_1^n}\left[\left(e^{-\frac{\|\hat{x}_t-\rho_t x_t'\|^2}{2\sigma^2}}\right)\prod_{r\neq n}\pi(b_r)\right]}{\sum_{x_t\in\chi}\left[\left(e^{-\frac{\|\hat{x}_t-\rho_t x_t\|^2}{2\sigma^2}}\right)\prod_{r\neq n}\pi(b_r)\right]} \tag{6.3}$$

By exploiting the trellis structure of code, the SISO decoder computes the soft values (*a priori, a posteriori* probabilities) for the coded bits using the Forward-Backward algorithm. After the last pass, the final decoded outputs are the hard decisions based on the *a posteriori* bit probabilities APP(c_n).

6.1.1 Analytical Bound for BICM-ID

The union bound of probability of bit error for convolutional codes of rate k/n is given by

$$P_b\leq\frac{1}{k}\sum_{d=d_{\min}}^{\infty}W_I(d)f(d,\Omega,\chi) \tag{6.4}$$

Where $W_I(d)$ is the total input weight of error events at Hamming distance *d* and d_{min} is the minimum Hamming distance of the code. Note that $f(d,\Omega,\chi)$ denotes the pairwise error probability (*PEP*) of BICM and depends only on Hamming distance *d*, a labeling map Ω and a signal constellation χ. Two cases:

a. *Static channel case:*

The union bound of the PEP of BICM can be written in the form

$$f(d,\Omega,\chi)=\phi(\Delta(\hat{x},x))$$

Where $\phi(\Delta(\hat{x},x))=E_h\left[Q\left(\sqrt{\frac{\|\hat{x}-x\|^2\|h\|^2}{4N_0}}\right)\right]$

b. Ergodic channel case:

The Union bound of the PEP of BICM can be written in the form

$$f(d,\Omega,\chi)\le\frac{1}{2\pi j}\int_{\alpha-j\infty}^{\alpha+j\infty}\left[\psi_{ub}(s)\right]^{d}\frac{ds}{s} \tag{6.5}$$

Where $\psi_{ub}(s)=\frac{1}{m2^{m}}\sum_{n=0}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in=\chi_{b}^{n}}\sum_{x\in\chi_{\bar{b}}^{n}}\phi_{\Delta(\hat{x},x)}(s)$

$\bar{b}$ is the complement of bit *b*. Note that $\phi_{\Delta(\hat{x},x)}(s)$ is the Laplace transform of the probability density function of the metric difference $\Delta(\hat{x},x)$ between components $\hat{x}$ and *x*. When Gray labeling is used, irrelevant error events can be expurgated from eq.(6.1) and the *PEP* can be rewritten as

$$f(d,\Omega,\chi)\le\frac{1}{2\pi j}\int_{\alpha-j\infty}^{\alpha+j\infty}\left[\psi_{ex}(s)\right]^{d}\frac{ds}{s} \tag{6.6}$$

Where $\psi_{ex}(s)=\frac{1}{m2^{m}}\sum_{n=0}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in=\chi_{b}^{n}}\phi_{\Delta(\hat{x},x')}(s)$

And $x'=x'(\hat{x})\in\chi_{\bar{b}}^{n}$ denotes the nearest neighbor of $\hat{x}$. However, due to large coding gain introduced by iterative decoding, we are most interested in an *analytical bound for the error free feedback performance* to which the BICM-ID performance converges at very low BER. For convenience, we use the term *error floor* to indicate the error free feedback performance of BICM-ID.

For the analysis of the error floor of BICM-ID, we consider the idealized condition assuming error-free feedback. With perfect knowledge of *some* or *all* values of other bits forming a channel symbol, an *M*-ary constellation can be partitioned into sets of smaller constellations having larger inter-signal Euclidean distance. Here, we choose to feedback the decoding decisions of all other bits; therefore, any *M*-ary signal set is converted to *M/2* binary signal sets. This significantly increases the minimum inter-signal Euclidean distance and also reduces the number of nearest neighbor to one when Gray labeling is not used. Given ideal feedback for each $\hat{x}\in\chi_{b}^{n},\chi_{\bar{b}}^{n}$, contains only one term $x''=x''(\hat{x})$, whose label has the same binary bit values as those $\hat{x}$ of except at the *n*th bit position. Note that $x''(\hat{x})$ is not necessary the same as $x'(\hat{x})$ depending on the labeling map Ω. Therefore, by removing the innermost summation in eq.(6.5), the *PEP* of the error floor of BICM-ID can be written as

$$f(d,\Omega,\chi)\le\frac{1}{2\pi j}\int_{\alpha-j\infty}^{\alpha+j\infty}\left[\psi_{ef}(s)\right]^{d}\frac{ds}{s} \tag{6.7}$$

Where $\psi_{ef}(s)=\frac{1}{m2^{m}}\sum_{n=0}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in=\chi_{b}^{n}}\phi_{\Delta(\hat{x},x'')}(s)$

For Rician fading case:

$$\phi_{\Delta(\hat{x},x)}(s)=\frac{1+K}{(1+K)-s(sN_{0}-1)\|\hat{x}-x\|^{2}}\exp\left\{\frac{Ks(sN_{0}-1)\|\hat{x}-x\|^{2}}{(1+K)-s(sN_{0}-1)\|\hat{x}-x\|^{2}}\right\}$$

Where *K* is the Rice factor. By letting *K*=0, the special case for Rayleigh fading is given as

$$\phi_{\Delta(\hat{x},x)}(s) = \frac{1}{1+s(1-sN_0)\|\hat{x}-x\|^2} \tag{6.8}$$

Then, the *PEP* of the error floor of BICM-ID defined in eq.(6.5) can be numerically evaluated by the Gauss-Chebyshev quadrature method before the bit error probability is calculated using eq.(6.4).

The main difference between (6.6) and (6.7) is that the former remains a valid upper bound only for Gray labeling and becomes an approximation for non-Gray labeling while the latter does not depend on labeling.

6.1.2 Design Optimization

Signal labeling is the crucial part of conventional BICM and our BICM-ID design. It is shown in past literature that Gray labeling yield the best performance for BICM. Using eqs.(6.4), (6.5) and (6.8), the asymptotic performance of BICM in our special case can be approximated by:

$$\log_{10} P_b \cong \frac{-d_2(\zeta)}{10}\left[\left(Rd_h^2(\Omega)\right)_{dB} + \left(\frac{E_b}{N_0}\right)_{dB}\right] + Cte \tag{6.9}$$

Where P_b probability of bit error;

$d_2(\zeta)$ minimum Hamming distance of the code;

R information rate;

$d^2{}_h(\Omega)$ *harmonic mean* of the minimum squared Euclidean distance.

For any *M*-ary constellation with a labeling map Ω, $d^2{}_h(\Omega)$ can be calculated by ($m = \log_2(M)$)

a. Static channel case: $d_h^2(\Omega) = \left(\frac{1}{m2^m}\sum_{n=1}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in\chi_b^n}\exp\left(-\frac{E_s}{4N_0}\|\hat{x}-x'\|^2\right)\right)^{-1}$

b. Ergodic channel case: $d_h^2(\Omega) = \left(\frac{1}{m2^m}\sum_{n=1}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in\chi_b^n}\frac{1}{\|\hat{x}-x'\|^2}\right)^{-1}$ (6.10)

Therefore, it is obvious that the asymptotic BICM performance depends primarily on the minimum Hamming distance of a code $d_2(\zeta)$ and the harmonic mean of the minimum squared Euclidean distance $d^2{}_h(\Omega)$. Specifically, $d_2(\zeta)$ controls the slope of the BER curve while $d^2{}_h(\Omega)$ gives the horizontal offset. Intuitively, the diversity order can further be increased by concatenating the larger signal constellation with a lower rate code; however, it may not provide a lower P_b in the range of interest due to reduction in the minimum inter-signal Euclidean distance among signal constellation points. Note that a labeling map Ω is independent of a convolutional code ζ due to bit interleaving; therefore, it can be separately optimized in our iterative decoding algorithm without altering the code diversity. This is carried out to maximize the harmonic mean of the minimum Euclidean distance as seen with error free feedback.

From eqs.(6.4), (6.7), and (6.8), the asymptotic performance of the error floor of BICM-ID is obtained by:

$$\log_{10} P_b \cong \frac{-d_2(\zeta)}{10}\left[\left(R\widetilde{d}_h^2(\Omega)\right)_{dB} + \left(\frac{E_b}{N_0}\right)_{dB}\right] + Cte \tag{6.11}$$

a. Static channel case: $\tilde{d}_h^2(\Omega) = \left(\frac{1}{m2^m}\sum_{n=1}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in\chi_b^n} \exp\left(-\frac{E_s}{4N_0}\left\|\hat{x}-x''\right\|^2\right)\right)^{-1}$

b. Ergodic channel case: $\tilde{d}_h^2(\Omega) = \left(\frac{1}{m2^m}\sum_{n=1}^{m}\sum_{b=0}^{1}\sum_{\hat{x}\in\chi_b^n} \frac{1}{\left\|\hat{x}-x''\right\|^2}\right)^{-1}$ (6.12)

Note that x'' is the only member in $\chi_{\bar{b}}^n$ as defined in eq. (6.7). Since there is no change in the coding structure or constellation size, all terms on the right hand side of eqs. (6.9) and (6.11) are the same except that $\widetilde{d}_h^2(\Omega)$ substitutes for $d_h^2(\Omega)$. Therefore, from eqs.(6.10) and (6.12), the labeling map *Ω* should be designed such that $\left\|\hat{x}-x''\right\|$ is larger than $\left\|\hat{x}-x'\right\|$ for all $\hat{x}$ (if possible) in order to achieve the iterative decoding gain.

We can calculate the *asymptotic gains* of labeling *Ω*2 with respect to labeling *Ω*1 as follows:

$$Gain_{dB} \approx 10\log_{10}\left(\frac{d_h^2(\Omega_2)}{d_h^2(\Omega_1)}\right) \tag{6.13}$$

$$\widetilde{Gain}_{dB} \approx 10\log_{10}\left(\frac{\widetilde{d}_h^2(\Omega_2)}{\widetilde{d}_h^2(\Omega_1)}\right) \tag{6.14}$$

The asymptotic gain $Gain_{dB}$ in eq.(6.13) computes the gain of labelling *Ω*2 with respect to labelling *Ω*1 in a BICM system or in the first pass of BICM-ID system (before feedback). The asymptotic gain $\widetilde{Gain}_{dB}$ in eq.(6.14) computes the gain of labelling *Ω*2 with respect to labelling *Ω*1 in the other iteration of BICM-ID system (after feedback).

6.2 BICM-ID combined to OS concept

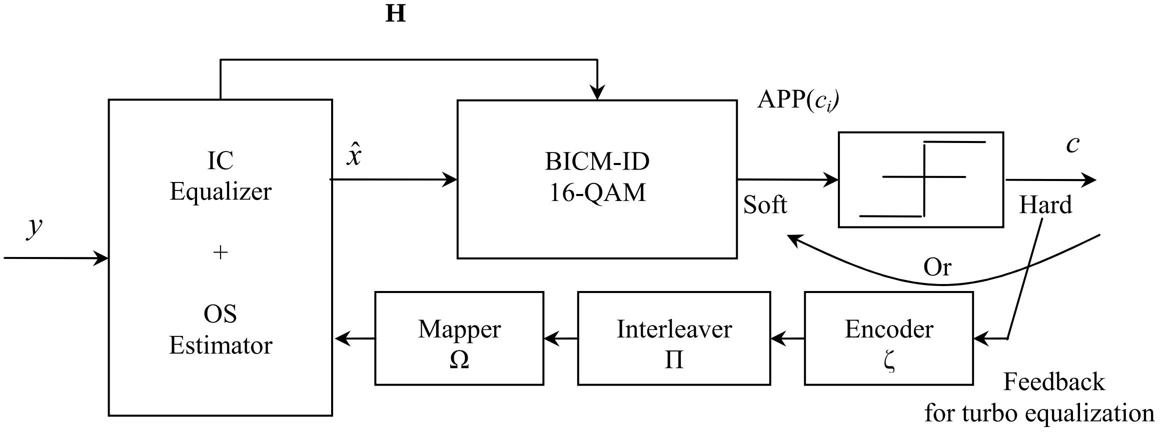


Figure 6.3: BICM-ID combined to OS technique in turbo equalization.

Here, we combined the two techniques explained in chapter 5 and previously in section 6.1, the OS estimation technique for turbo equalization and the BICM-ID technique for turbo demodulation. Then the received signal y is detected by an IC equalizer and an OS estimator to give the signal $\hat{x}$, which is considered like the input of the BICM-ID where it is suffer the first iterative treatment. The *a posteriori* probabilities output of BICM-ID are fed back in their soft or hard values to the equalizer which is the second iterative treatment in our system.
After our explanation on the design criterion and the choice of the optimal mapping in section 6.1.2, we will show here some mapping selected among the others to give a good comparison.
It is good to know that the properties which become the most important in our application are:

- The evolution of the first iteration which mean the moment of triggering.
- The evolution of the last iteration which mean the asymptotic performances.

With these two properties, the Gray mapping has the best performances for the first iteration, in same time it has the worst asymptotic performances. Naturally, the Anti gray mapping has the inverse characteristics. Table 6.1 class the mapping by order from the best to the worst in term of performances in the first iteration and the last iteration. Where OMI and EFD mean Optimal Mutual Information and Effective Free Distance respectively.

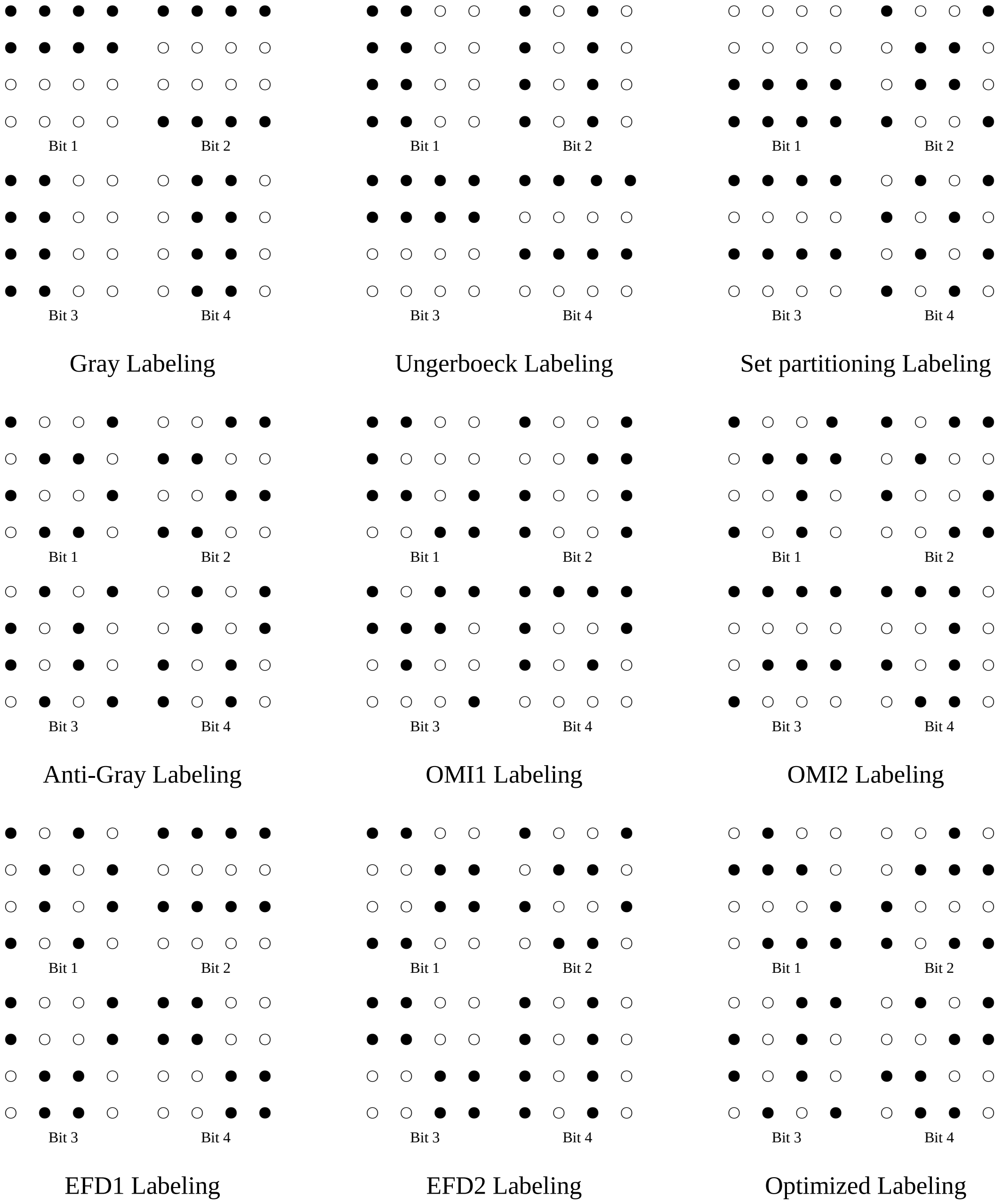
First iteration: Trigger moment	Last iteration: Asymptotic performances
Gray	Anti gray
Optimized	EFD by Gorokhov
OMI by Ten Brink	Optimized
Set partitioning	Set partitioning
EFD by Gorokhov	OMI by Ten Brink
Anti gray	Gray

Table 6.1: Put in order of performances of some mapping.

The performances of the first iteration for EFD and Anti-Gray mapping are very degrade which lead to a very late Trigger moment in the iterative process. These mapping are not adapted for the turbo equalization, in the interested SNR zone, whereas they are agreed very well for the turbo demodulation. Then we will not consider these two mapping in our study. Fig. 6.4 shows the subset partitionning for each for the four bit positions of 16-QAM constellations. The points that have the same color correspond to the decision regions (Black for bit 0, each bit in χ_0^n, White for bit 1, each bit in χ_1^n). It is obvious that all labelling methods have the same minimum Euclidian distance between subsets of χ_1^n and χ_0^n but a different number of nearest neighbors.

Given ideal feedback of all other bits, a 16-QAM constellation is translated to a binary signaling selected from eight possible pairs. Fig. 6.5 illustrates the increase in the minimum Euclidean distance between subsets. Gray labelling is not the preferred choice because most of binary signal sets resulting from ideal feedback have the same inter-signal Euclidean distance as original 16-QAM constellation.

Numerical results from calculating the *harmonic* mean of the minimum Euclidean distance before feedback, d_h^2, and after feedback, $\tilde{d}_h^2$, are shown in Table 6.2 and 6.3. In table 6.2, we present the gain of mapping in relation to gray labeling, and in table 6.3, we call the difference in $\left(\tilde{d}_h^2\right)_{dB}$ and $\left(d_h^2\right)_{dB}$ of conventional BICM with gray labeling as the *offset gain*. This gives a quick comparison between various labeling schemes with iterative decoding and conventional BICM.



Gray Labeling

Ungerboeck Labeling

Set partitioning Labeling

Anti-Gray Labeling

OMI1 Labeling

OMI2 Labeling

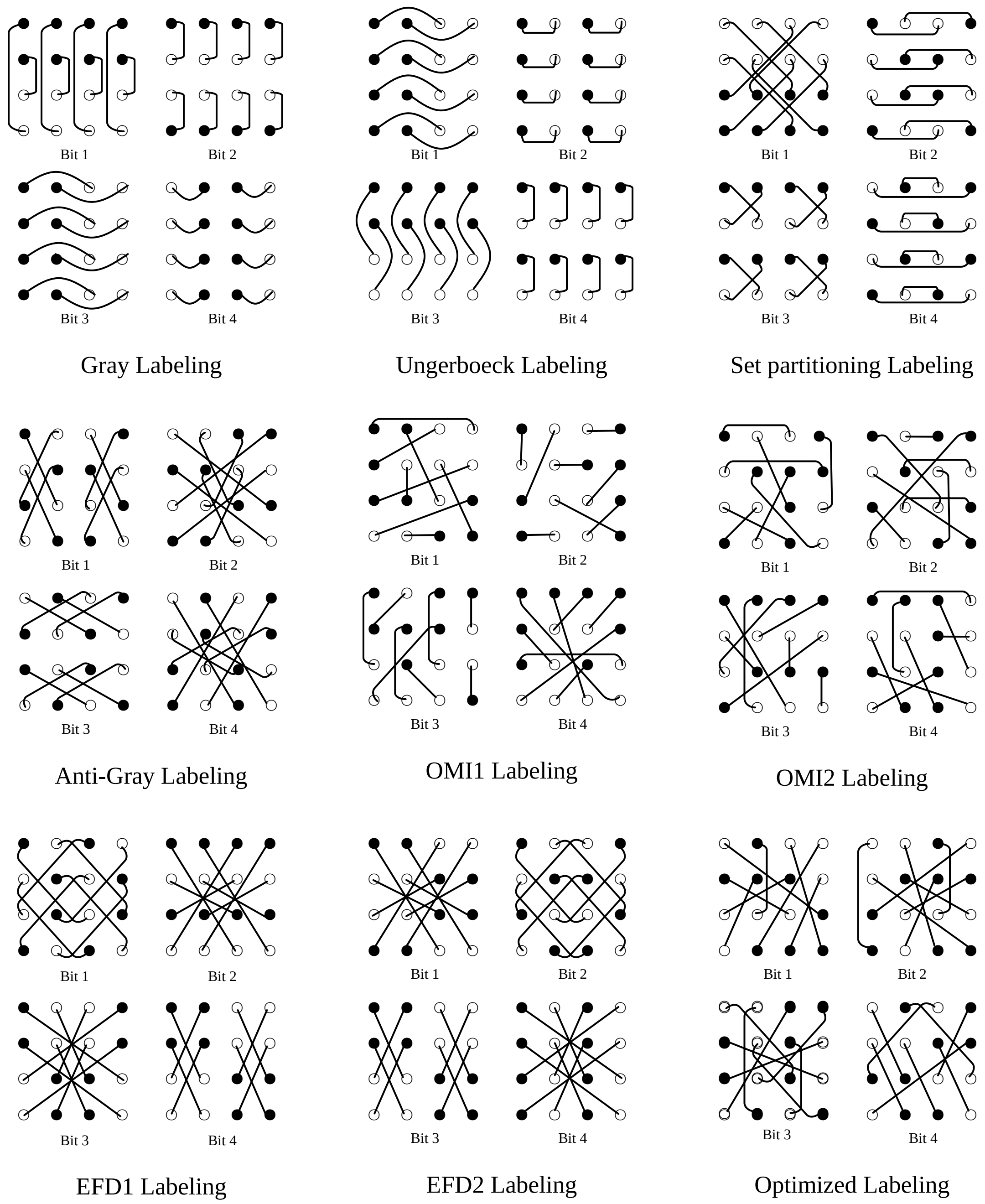
EFD1 Labeling

EFD2 Labeling

Optimized Labeling

Figure 6.4: Subset partitions of 16-QAM for some mapping schemes.

In addition, optimization of $d_h^2(\Omega)$ is done separately from our decoding algorithm; therefore, the *offset gain* is the asymptotic performance improvement regardless of the code structure. It is preferable to have a labeling map that maximizes $\tilde{d}_h^2(\Omega)$ while having sufficiently large original $d_h^2(\Omega)$ such that the feedback decoder can reach its ideal performance within a few passes.



Gray Labeling

Ungerboeck Labeling

Set partitioning Labeling

Anti-Gray Labeling

OMI1 Labeling

OMI2 Labeling

EFD1 Labeling

EFD2 Labeling

Optimized Labeling

Figure 6.5: The minimum Euclidean distance for some mapping schemes.

Consider the impact of signal labeling on the offset gain. For BICM-ID with the same convolutional code, Table 6.3 shows that Gray labeling yields the best performance without feedback due to the large d_h^2 ; however, the performance gain with feedback is very small. Optimized labeling gives the largest $\tilde{d}_h^2$ in ergodic case, but SP labeling gives a $\tilde{d}_h^2$ larger than Optimized in static case, and thus the asymptotic offset gain at the cost of having the poorest first round performance.

Labeling	$\tilde{d}_h^2$	*Gain % Gray (dB)*
Gray	*2.5044*	*0*
SP	*2.8164*	*0.51*
Optimized	*2.9628*	*0.73*

Table 6.2: Distances for ideal a priori information in static channel case at E_b/N_0=4dB.

Labeling	d_h^2	$\tilde{d}_h^2$	*Offset Gain (dB)*
Gray	*0.492*	*0.514*	*0.19*
SP	*0.411*	*1.119*	*3.56*
Optimized	*0.413*	*2.602*	*7.23*

Table 6.3: Distances and Offset Gain for some mapping in ergodic channel case.

6.3 Simulations and Results

In this section, we show simulation results for BICM-ID over an ISI Channel known or estimated by the OS technique. A 16-QAM constellation is used with rate 1/3 convolutional codes. Several labeling maps are used to illustrate the importance of labeling design. We will consider four iterations for the BICM-ID and the turbo equalization with considering hard decision feedback. We also vary the estimated channel length and the training sequence length.

In Fig. 6.6, we show the asymptotic performance of BICM-ID, with different mappings, in a turbo equalization system over a Gaussian, a Rayleigh channel known perfectly.

We remark that a lot of mapping outperform the Gray mapping in his asymptotic performance for turbo equalization, with a gain up to 7dB in case of Rayleigh channel and 0.73dB in case of Gaussian channel for our optimized mapping.

We show, in Fig. 6.7 the performance of BICM-ID, always in turbo equalization system and channel known perfectly. We compare these performances to the BICM with Gray mapping. It is clear that despite his best asymptotic performance the Anti Gray mapping is not adapted to the turbo equalization and it is not able to approach the BICM performances. And this figure certifies that the Gray mapping has the best performances for a weak SNR but for high SNR the other mappings outperform the gray mapping. It seems that the optimized mapping is a good compromise for weak and high SNR.

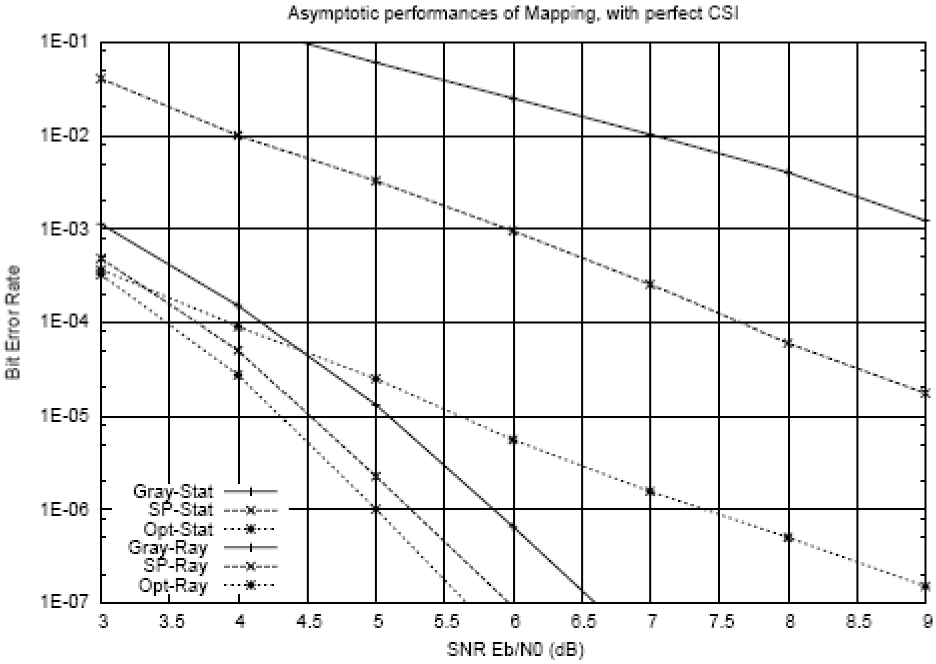


Figure 6.6: Asymptotic performances of considered mapping in our applications.

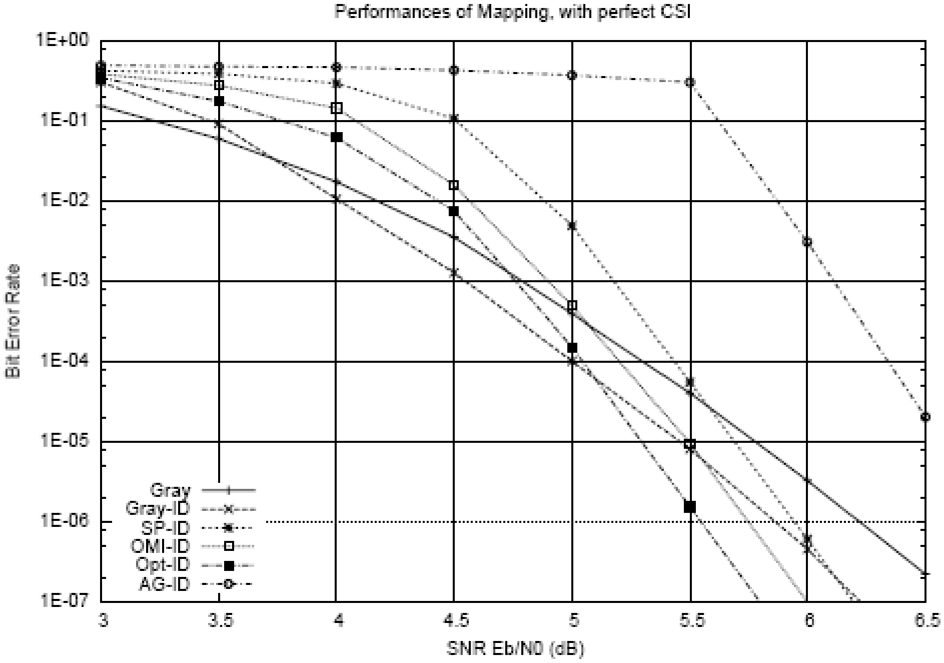


Figure 6.7: Performances of considered mapping in turbo equalization.

Fig. 6.8 and 6.9 show the effect of the use of BICM-ID and OS technique in the same turbo equalization system, and we will vary the estimated channel length *l* and the training sequence length *N*.

We remark from these two figures that use of BICM-ID and OS technique leads to same behavior when the channel is perfectly known. The optimized mapping loses up to 0.5dB at weak SNR but we get back this loss at high SNR contrary to other mappings.

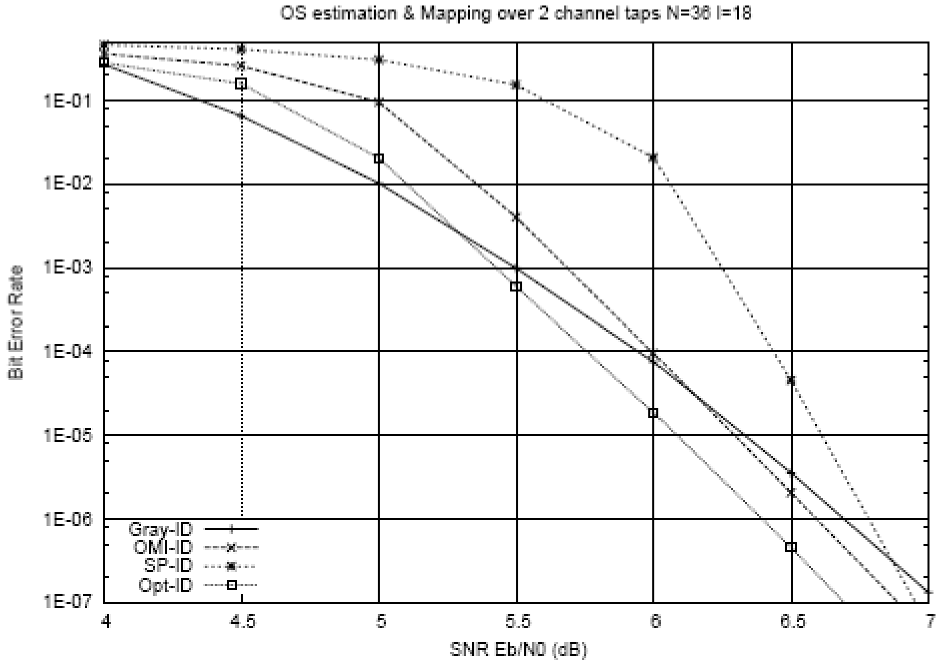


Figure 6.8: Performance of considered mapping over 2 path channel and with OS estimation technique with *N*=36 and *l*=18.

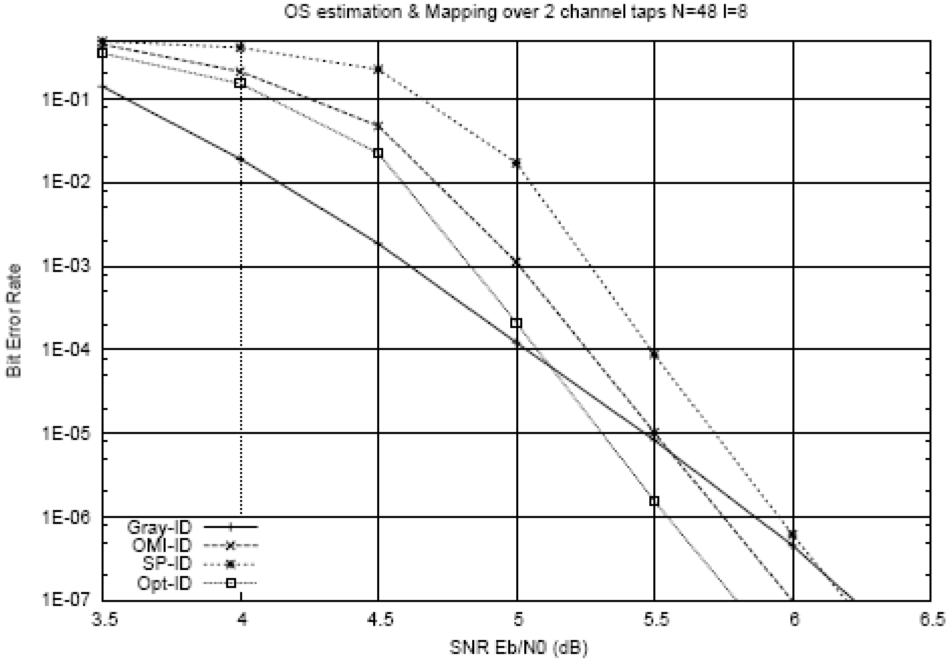


Figure 6.9: Performance of considered mapping over 2 path channel and with OS estimation technique with *N*=48 and *l*=8.

Finally, we represent in Fig. 6.10 a comparison between the classical technique (BICM without ID and IC equalization without OS estimation in a turbo equalization system) and our system applied (BICM-ID and IC equalization with OS estimation in a turbo equalization system).

It is clear from the Fig. 6.10 that with or without ID the OS technique outperform the classical technique by 1dB. With or without OS the BICM-ID outperforms BICM by 0.5dB. And our system proposed outperforms the classical system by up to 1.5dB.

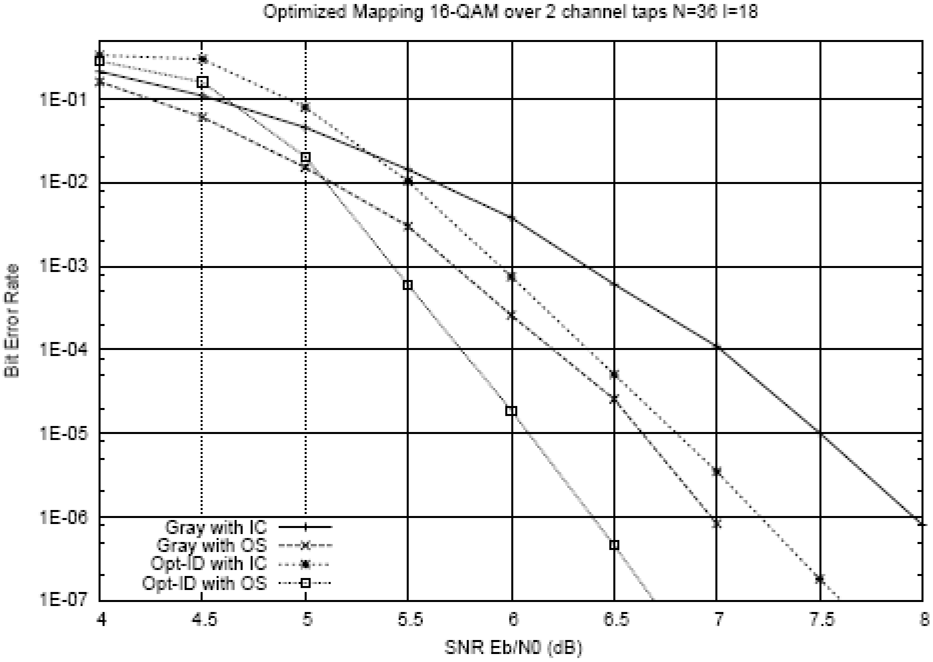


Figure 6.10: Performances of BICM-ID with OS vs BICM without OS.

6.4 Conclusion

In this chapter, we have analyzed and evaluated an approach to the design of BICM-ID for ISI channels. By recognizing that the coding and modulation are isolated by the bit interleaver, and identifying the impact of iterative decoding on the harmonic mean of the minimum Euclidean distance, we have developed a powerful turbo equalization system yet relatively complex.

The theoretical part and the extensive set of simulation results prove that the harmonic mean of the minimum Euclidean distance identified as crucial for BICM, can be greatly increased with BICM-ID and the error free feedback assumption.

A new labeling map optimized under the harmonic mean criterion for 16-QAM is presented. The simulations presented prove that even when the mapping has the best asymptotic performances is not necessary that this mapping will have the best performance in turbo equalization; some mapping is not adapted to turbo equalization like Anti Gray. We presented our optimized labeling map which seem a good compromise in weak SNR and high SNR and which outperforms the Gray mapping by 0.5dB at high SNR but unfortunately loss up to 0.5dB at weak SNR.

Finally, our system proposed outperforms the classical system (IC procedure without OS and ID techniques) also when we use this system with or without OS and ID. Our system without OS technique makes a gain up to 0.5dB, with OS without ID presents a gain up to 1.0dB and when we use the two proposed technique OS and ID we obtain a gain up to 1.5dB in relation to the classical system used actually.

Chapter 7
Low-Density Parity-Check Codes

LDPC codes were invented by Gallager in his thesis [37, 38]. LDPC codes did not get much attention for many decades until recently when highly successful turbo codes were discovered [1]. LDPC codes were then rediscovered by Spielman et al. [63] and MacKay et al. [59]. For many channels and iterative decoders of interest, low-density parity-check (LDPC) codes exhibit a threshold phenomenon [61]: as the block length tends to infinity, an arbitrarily small bit error probability can be achieved if the noise level is smaller than a certain threshold. For a noise level above this threshold, on the other hand, the probability of bit error is larger than a positive constant. Gallager first observed this phenomenon for the binary symmetric channel (BSC) when he introduced regular LDPC codes [37, 38] using an explicit construction of regular graphs. Luby et al. generalized this idea to randomly constructed irregular LDPC codes, showed that irregular codes perform better than regular ones, and also showed that the threshold phenomenon occurs for these codes [57].

In [61], this observation was further generalized by Richardson and Urbanke to a large range of binary-input channels, including binary erasure, binary symmetric, Laplace, and AWGN channels, and to various decoding algorithms including belief propagation (sum-product algorithm), which are collectively called message-passing algorithms. Richardson et al. proved a general concentration theorem showing that the decoder performance on random graphs converges to its expected value as the length of the code increases, generalizing the result of Luby et al. [60]. Since it is difficult to determine the expected performance for an ensemble of finite size, they used the expected behaviour in the limit of infinitely long codes, which can be determined from the corresponding cycle-free graph. They defined the threshold as indicated above for a random ensemble of irregular codes specified by degree distributions, and developed an algorithm called density evolution for iteratively calculating message densities, enabling the determination of thresholds.

Using this result, they constructed LDPC codes that clearly beat the powerful turbo codes [1] on AWGN channels. Recently, this was improved in [34], suggesting that LDPC codes might approach the channel capacity of the AWGN channel asymptotically.

Calculating thresholds and optimizing degree distributions using density evolution is a computationally intensive task for most channels other than BECs. In BECs, density evolution becomes one-dimensional, and it is possible to do more analysis and even to construct capacity achieving codes [58]. For more interesting channels, including AWGN channels, however, density evolution is too complicated to be analyzed.

In this chapter, we introduce the LDPC codes and some of their properties, constructions and decoding algorithms. LDPC codes are binary linear block codes and, as their name indicates, have a parity check matrix that has only a small number of 1s per row and per column. Thus, the parity check matrix has a low density of 1s. We distinguish between regular and irregular LDPC codes where only irregular LDPC codes can approach capacity. To obtain good LDPC codes, special design methods exist. Some methods rely on density evolution [60]. To achieve, capacity infinite block lengths are necessary. However, simulation results show that LDPC codes perform near capacity with moderate block length (***N*** = $10^4 \ldots 10^5$).

7.1 Regular and Irregular LDPC-Codes

A ($\boldsymbol{d_v}$; $\boldsymbol{d_c}$) **regular LDPC code** is a binary linear (***N, K***) block code that has a parity check matrix H with a fixed and small number $\boldsymbol{d_v}$ ones in each column and a fixed and small number $\boldsymbol{d_c}$ ones in each row. We denote $\boldsymbol{d_v}$ the **variable node degree** and $\boldsymbol{d_c}$ the **check node degree**

An **irregular LDPC code** is a binary linear (***N, K***) block code that has a parity check matrix H with a small number of ones in each column and a small number of ones in each row. The variable node degree $\boldsymbol{d_v}$ and the check node degree $\boldsymbol{d_c}$ is not constant over rows and/or columns.

Sometimes $\boldsymbol{d_v}$ and $\boldsymbol{d_c}$ are called *column weight* and *row weight* because they denote the number of ones per column and row in the parity check matrix H. The reason for the additional notation of $\boldsymbol{d_v}$ as variable node degree and $\boldsymbol{d_c}$ as check node degree originates from the representation of LDPC codes by factor graphs which consist of variable nodes and check nodes.
Let ***E*** be the total number of ones in the parity check matrix. With $\boldsymbol{E = Nd_v = Ld_c}$, we get the *design code rate* of regular LDPC codes as

$$R_c = \frac{K}{N} = 1 - \frac{L}{N} = 1 - \frac{d_v}{d_c} \tag{7.1}$$

The actual code rate can be higher than the design code rate if the parity check matrix contains redundant rows. We will present the design code rate of irregular LDPC codes.

7.2 Factor Graphs

Factor Graphs [56] have been introduced to visualize complicated global functions which can be factorized into products of "local" functions. In particular, the sum-product algorithm used to decode LDPC codes computes various marginal functions derived from a global function.

A **factor graph** is a bipartite graph that expresses the structure of the factorization. A factor graph has a **variable node** $\boldsymbol{v_j}$ for each variable, a **check node** $\boldsymbol{c_i}$ for each local function, and an edge from variable node $\boldsymbol{v_j}$ to check node $\boldsymbol{c_i}$ if and only if $\boldsymbol{v_j}$ is an argument of $\boldsymbol{c_i}$.

In Fig. 7.1 we show the factor graph for the LDPC code with the parity check matrix given in (7.7).

According to the definition of the parity check matrix H_{LDPC}, each row i with $1 \le i \le (N-K)$ of H_{LDPC} represents a parity check equation on certain variables v_j with $1 \le j \le N$. A variable v_j is involved in the ith parity check equation if the entry $H_{i,j}$ equals one.

We represent this check structure by a factor graph: The factor graph of an LDPC code consists of ***N*** variable nodes on the left side and ***L = N - K*** check nodes on the right side which are eventually connected through edges. Each check node c_i represents a parity check equation which is determined by row *i* of the parity check matrix H_{LDPC}. We represent this channel dependency through the connections coming from the left of the variable nodes. The connection from variable node v_j to check node c_i is determined by the entry $H_{i,j} = 1$ in the parity check matrix. Because these connections of variable nodes and check nodes are shuffled, we call this connection unit the *edge interleaver.*

Using factor graphs, the terms *variable node degree* d_v and *check node degree* d_c become obvious. The variable node degree d_{v_j} of a certain variable v_j node is the number of edges that are connected to this node. Similarly, the check node degree d_{c_i} is the number of edges that are connected to a given check node c_i.

From the definition of a regular LDPC code, we can define a regular factor graph if the corresponding LDPC code is regular.

7.2.1 Cycles and Girth of a Factor Graph

We give only the essential definitions of some of the properties of (factor) graphs. For more comprehensive information on graphs, we refer to [70].

A **cycle** in a (factor) graph is a closed loop where each vertex is only used once and the staring vertex is identical with the ending vertex. The length ***K*** of the cycle is the number of edges (or vertices).

The minimum length of a cycle contained in a (factor) graph is called ***girth***. A ***tree*** is a graph without cycles.

Cycles and girth of factor graphs are important when decoding LDPC codes.

7.2.2 Degree Distribution

We characterize a class of regular LDPC codes by one variable node degree and one check node degree. Similarly, we associate a *degree distribution* with a class of irregular LDPC codes. Specifying degree distributions, from edge perspective, λ_j denotes the fraction of *edges* connected to all variable nodes that have a certain degree $d_v = j$. Similarly ρ_j denotes the fraction of *edges* connected to all check nodes that have a certain degree $d_c = j$.

Let $d_{v\max}$ be the highest variable node degree and $d_{c\max}$ the highest check node degree.

Then the total number ***E*** of edges (or equivalently the total number of ones in the parity check matrix) equals

$$E = \frac{1}{N}\sum_{i=1}^{d_{v\max}} \frac{\lambda_i}{i} = \frac{1}{L}\sum_{j=1}^{d_{c\max}} \frac{\rho_j}{j} \tag{7.2}$$

Similarly, as for regular LDPC codes we define the *average variable node degree* $\bar{d}_v$ and the *average check node degree* $\bar{d}_c$ for irregular LDPC codes as

$$\bar{d}_v = \frac{E}{N};\ \bar{d}_c = \frac{E}{L} \tag{7.3}$$

Where ***N*** is the number of variable nodes and ***L = N - K*** the number of check nodes.

The *design code rate* for irregular codes results as

$$R_c = 1 - \frac{\bar{d}_v}{\bar{d}_c} \tag{7.4}$$

Then we can specify the degree distribution as polynomials from edge perspective

$$\begin{aligned} \lambda(x) &\triangleq \sum_{i=1}^{d_{v\max}} \lambda_i x^{i-1} \\ \rho(x) &\triangleq \sum_{j=1}^{d_{c\max}} \rho_j x^{j-1} \end{aligned} \tag{7.5}$$

The *design code rate* for irregular codes becomes $R_c = 1 - \frac{\int_0^1 \rho(x)\,dx}{\int_0^1 \lambda(x)\,dx} = 1 - \frac{\sum_j \frac{\rho_j}{j}}{\sum_i \frac{\lambda_i}{i}}$ (7.6)

In this thesis, we use the factor graph representation of LDPC codes as shown in Fig. 7.1, which is a graph representation of the following (***L* x *N)*** parity check matrix H$_{\text{LDPC}}$ (also known as Tanner graph or bipartite graph):

$$H_{LDPC} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ & \vdots & & & \vdots & & & \vdots \\ 0 & 1 & 0 & 1 & 1 & 0 & \cdots & 0 \end{pmatrix} \tag{7.7}$$

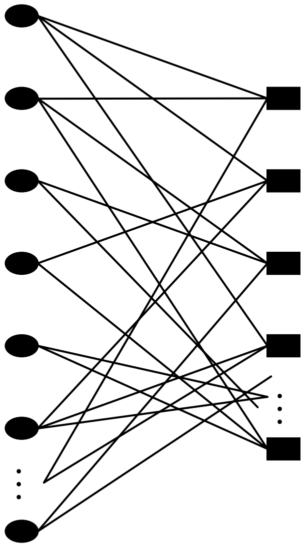


Figure 7.1: Factor graph (Tanner graph) of a regular or irregular LDPC code.

Analysis of Tanner graphs tells us much about properties such as girth, expansion, degree and diameter. These properties affect how *good* the code is and also how well it is suited to the decoding process. The smaller the girth the worse the code performs under sum-product decoding. The diameter of the graph is the maximum, over all pairs of nodes, of the shortest path between them. This number gives a measure of how quickly information is propagated throughout the graph during decoding. It is desirable that the diameter is small. The degree of a node is defined as the number of edges that are connected to it. The decoder complexity is proportional to the degree of the nodes. There is much interest in finding the optimal degree distribution in the graph. A graph is termed a good expander if every set of nodes has a large number of neighbours. Codes from graphs that are good expanders can correct more errors.

7.3 Construction of a Parity Check Matrix for a Degree Distribution

According to the degree distribution, we can construct the corresponding parity check matrix H. This means that we have to place the ones in the parity check matrix so that the degree distribution for both rows (check node degree $\boldsymbol{d_c}$) and columns (variable node degree $\boldsymbol{d_v}$) is fulfilled.

Gallager [37], Mackay & Neal [59] construct it randomly: The constraints typically imposed on the pseudo-randomly constructed parity-check matrices of binary LDPC codes are that the parity-check matrix be regular (or nearly so), and that the code be free of short cycles, especially cycles of length 4. However, the difficulties in constructing code without 4-cycles are compounded as the order of the field increases. Which push Luby & al. [57], Richardson & Urbanke [61] to work in some optimization techniques for the construction a capacity approaching LDPC code, and Lucas & al [71] to use the algebraic construction. An interesting outcome of the research into binary algebraic LDPC codes has been the recognition of the key role played by rank deficient parity-check matrices which can play a significant role in improving the performance of *q*-ary LDPC codes.

All parity check matrices in this thesis have been constructed randomly. Moreover, cycles and small stopping sets should be avoided.

7.4 Decoding Algorithms

Decoding algorithms for LDPC codes are called message passing algorithms, and are iterative algorithms. The reason for their name is that at each round of the algorithms intermediate messages are passed from variable nodes to check nodes, and from check nodes back to variable nodes.

One very important aspect of messages passing is its running time. Since the algorithm traverses the edges in the graph, and the graph is sparse, the number of edges traversed is small. Moreover, if the algorithm runs for a constant number of times, then each edge is traversed a constant number of times, and the algorithm uses a number of operations that is linear in the number of message nodes!

Another important note about messages passing is that the algorithm itself is entirely independent of the channel used, though the messages passed during the algorithm are completely dependent on the channel.

We will describe two generic decoding algorithms for code realizations based on Tanner graphs. The structure of the algorithms matches the graphs directly. It will be convenient to think of these algorithms as parallel processing algorithms, where each variable and each check is assigned its own processor and the communication between them reflects the Tanner graph.
The algorithms come in two versions: the sum-product algorithm and the min-sum algorithm. The ideas behind them are not essentially new; rather, the algorithms are generalizations of well-known algorithms such as the Viterbi algorithm [67] and other trellis-based algorithms. Another important special case is Gallager's algorithm for decoding low-density parity-check codes [37]. A relatively general formulation of the algorithms was also given by Tanner [64].

The overall structure of the algorithms, and the context in which they apply, is illustrated in Fig. 7.2. As shown, the algorithms do not make decisions; instead they compute a set of *final results* upon which a decision can be made. The channel output enters the algorithms as a set of *observations.*

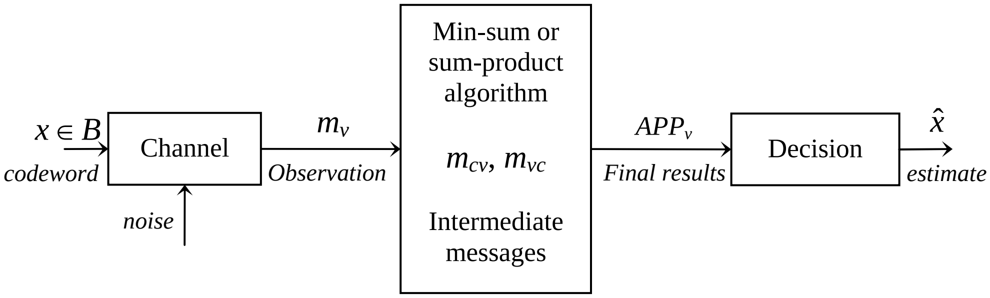


Figure 7.2: Typical decoding application of the min-sum or sum-product algorithm.

Formally, there is one observation for each variable v, denoted by m_v; similarly, there is one final a posteriori probability for each variable v, denoted by APP_v.

During the computation, the algorithms maintain a set of intermediate messages: for each pair *(v,c)* of adjacent variable and check set, there is one check-to-variable message m_{cv}, and one variable-to-check message m_{vc}. These messages are best thought of as having a direction on the Tanner graph. For instance, we will often call $m_{c_i v_j}$ the "contribution" from the check c_i to the variable v_j.

7.4.1 The Sum-Product Algorithm

Maximum-likelihood decoding usually becomes exponentially difficult for some graph codes including LDPC codes as the block length becomes large. Sum-product decoding, also known as belief propagation, can be viewed as applying Bayes rule locally and iteratively to calculate approximate marginal a posteriori probabilities for these codes. The sum-product algorithm is also practical, since the decoding complexity per iteration is linear in block length. If a graph has no cycles, then it can be easily proved that the sum-product algorithm computes marginal posterior probabilities exactly.

However, in many situations of interest including LDPC code graphs as in Fig. 7.1, we have graphs with cycles. In such cases, we may still want to run the sum-product algorithm ignoring cycles and hope for an answer that closely approximates the correct posterior probabilities. Despite a lack of theoretical understanding, the huge success of turbo codes and low-density parity-check codes has ignited further research in this area.

The behaviour of the sum-product algorithm on graphs with a single cycle is relatively well understood [52, 54 and 65]. In this case, the sum-product algorithm converges to a unique stationary point. If all variables are binary-valued, then the component wise maximum likelihood estimates produced by running the sum-product algorithm are correct. However, it is hard to generalize this result to a graph with more than one cycle.

Another approach to understand the sum-product algorithm on a graph with many cycles is to assume that all variables are jointly Gaussian [55, 66 and 62]. In this case, the analysis of the sum-product algorithm can be simplified since a Gaussian distribution is characterized by its mean and variance.

Richardson et al. [61, 60] demonstrated that the average asymptotic behaviour of a sum-product decoder for LDPC codes is numerically computable by using an algorithm called density evolution. They also showed that for many interesting channels, including additive white Gaussian noise (AWGN) channels, one can calculate a threshold value for the ensemble of randomly constructed LDPC codes which determines the boundary of the error-free region asymptotically, as the block length tends to infinity.

We first describe how density evolution works for ($\boldsymbol{d_v}$***;*** $\boldsymbol{d_c}$)-regular binary LDPC codes, where $\boldsymbol{d_v}$ denotes the number of neighbours of a variable node and $\boldsymbol{d_c}$ denotes the number of neighbours of a check node. Under the sum-product algorithm, variable and check nodes exchange messages iteratively.

A check node gets messages from its $\boldsymbol{d_c}$ neighbours, processes the messages, and sends the resulting messages back to its neighbours. Similarly, a variable node receives messages from its $\boldsymbol{d_v}$ neighbours and also from its corresponding channel output, processes the messages, and sends messages back to its neighbours.

Each output message of a variable or a check node is a function of all incoming messages to the node except for the incoming message on the edge where the output message will be sent out. This restriction is essential for the sum-product algorithm to produce correct marginal a posteriori probabilities for cycle-free graphs. This two-step procedure is repeated many times.

Let $m_{vc}^{(l)}$ be the message passed from variable node v to check node c at the l^{th} round of the algorithm. Similarly, define $m_{cv}^{(l)}$. At round 0, $m_{vc}^{(l)}$ is the log-likelihood of the variable node v conditioned on its observed value, which is independent of c. We denote this value by m_v

We will describe the sum-product algorithm together with density evolution in more detail.

Fig. 7.3 and 7.4 show message flows through a variable and a check node, respectively, where we use normal realizations [53] of variable and check nodes, which become a repetition and parity check, respectively. Under sum-product decoding, constraints become computation nodes and states become communication links between constraint nodes.

The sum-product algorithm consists of the following three steps:

- *Initialization*. The observations m_v are initialized as appropriate (using, e.g., channel information and/or some known a priori distribution). The intermediate messages m_{vc} and m_{cv} are set to one.
- *Iteration*. The intermediate messages m_{vc} and m_{cv} are updated a suitable number of times as follows. The variable-to-check message m_{vc} is computed as the sum of the variable's observation and all contributions coming into v except the one from c: (Fig. 7.3)

$$m_{vc}^{(l)} = m_v + \sum\nolimits_{c'\in C_v/\{c\}} m_{c'v}^{(l-1)} \quad if \quad l \geq 1, \tag{7.8}$$

  The check-to-variable message m_{cv} is obtained by: (Fig.7.4)

$$\tanh \frac{m_{cv}^{(l)}}{2} = \prod\nolimits_{v'\in V_c/\{v\}} \tanh \frac{m_{v'c}^{(l)}}{2} \tag{7.9}$$

  Where C_v is the set of check nodes incident to variable node v, and V_c is the set of message nodes incident to check node c

- *Termination*. The a posteriori probabilities APP_v are computed as the product of the variable's observation and all contributions coming into v, i.e.,

$$APP_v = m_v + \sum\nolimits_{c'\in C_v} m_{c'v}^{(l)} \tag{7.10}$$

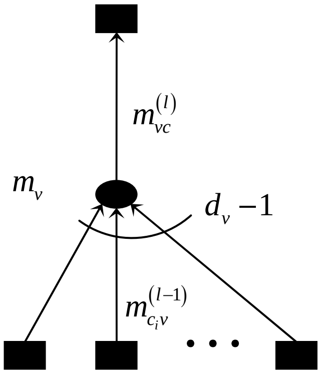


Figure 7.3: Message flow through a variable node.

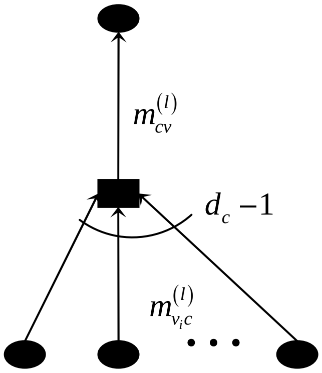


Figure 7.4: Message flow through a check node.

7.4.2 The Min-Sum Algorithm

In this section, we analyze density evolution for the min-sum algorithm. Unfortunately, it turns out that the max-sum algorithm does not preserve the "symmetry condition." Therefore, it is not guaranteed that the probability of error is non-increasing.

On a cycle-free graph, decoding based on the min-sum algorithm produces a probability of error for each bit that is always greater than or equal to that of the sum-product algorithm, because the sum-product algorithm produces the minimum probability of error.

The min-sum algorithm is a straightforward generalization of the Viterbi algorithm [67]. A well-known decoding algorithm for generalized concatenated codes [68] is also related, as is threshold decoding [69]. Before going into the general description, we encourage the reader to go through the example in Appendix A, where the decoding of a (7, 4, 2) binary linear code using the min-sum algorithm is performed in detail.

The algorithm consists of the following three steps:

- *Initialization*. The observations m_v are initialized as appropriate (using, e.g., channel information). The intermediate messages m_{cv} and m_{vc} are set to one.
- *Iteration*. The intermediate messages m_{cv} and m_{vc} are alternatively updated a suitable number of times as follows. The variable-to-check message m_{vc} is computed as the sum of the variable's observation and all contributions coming into v except the one from c:

  $$m_{vc}^{(l)} = m_v + \sum\nolimits_{c' \in C_v / \{c\}} m_{c'v}^{(l-1)} \quad if \quad l \geq 1, \tag{7.11}$$

  The check-to-variable message m_{cv} is obtained by examining all locally valid configurations on the variable v, for each summing all contributions coming into c except the one from v. The minimum over these sums is taken as the message m_{cv}:

  $$m_{cv}^{(l)} = \min \left\lfloor \sum\nolimits_{v' \in V_c / \{v\}} m_{v'c}^{(l)} \right\rfloor \tag{7.12}$$

  Where C_v is the set of check nodes incident to variable node v, and V_c is the set of message nodes incident to check node c
- *Termination*. The a posteriori probabilities APP_v are computed as the product of the variable's observation and all contributions coming into v, i.e.,

  $$APP_v = m_v + \sum\nolimits_{c' \in C_v} m_{c'v}^{(l)} \tag{7.13}$$

For Tanner graphs that contain cycles, there is no general result for a posteriori probabilities, or for the decoding performance.

Despite some shortcomings, there are some applications where the min-sum algorithm is preferable. The min-sum algorithm does not require the estimation of the channel noise and is simpler to execute than the sum-product algorithm. The min-sum algorithm can then be used as a simplified decoding algorithm that approximates the sum-product algorithm.

Chapter 8
Decoding of Non Binary and Adaptive LDPC Codes

Binary LDPC codes were invented by Gallager more than four decades ago and appear to have been largely forgotten until their rediscovery by MacKay and Neal in 1996. In recent years, a lot of research has been dedicated to Binary LDPC codes and due to their near-capacity error correction performance; they have emerged as a promising candidate technology for forward error correction (FEC) in future wireless systems and became serious competitors to Turbo Codes.

At the same time, there has been a growing interest in "soft decision" decoding schemes for error-correcting codes. Powerful turbo codes have been introduced which achieve low bit-error rates at low signal-to-noise ratio. They are of interest in wide range of telecommunications applications.

Gallager only considered *regular and binary* codes, i.e., codes with a fixed number of non-zero binary elements in the rows and columns of the LDPC matrix H_{LDPC}, respectively. However, significant improvements in bit error performance can be achieved by using irregular and non binary codes.

The different algorithms used for LDPC decoding (belief propagation, sum-product …) iteratively approximate the maximum likelihood solution of the decoding problem.

LDPC decoding is done by passing messages on the edges between the nodes of the bipartite "Tanner graph" that represents the parity check matrix of the LDPC code Fig. 7.1. The average complexity of the decoding process is hence the product of three factors:

1. The number of operations per node,
2. The average number of iterations, and
3. The number of active nodes in each iteration.

Powerful decoding schemes, like turbo codes, necessitate the utilization of decoding algorithms that compute *a posteriori* probabilities (APPs) on a symbol-by-symbol basis. An early contribution to APP decoding was made by Gallager where the decoding is performed by APP modules over each check node. At the same period, an application of APP decoding to the framework of threshold decodable codes was made by Massey. Unfortunately, the proposed APP algorithm has a high complexity which increases with both the number of states and the number of transitions in the trellis of the code. Hence, APP algorithms that reduce the computational complexity and/or storage requirements are interest for practical applications. Several authors have observed that APPs can be computed on a trellis of the dual code which simplifies decoding if the code is high rate, resulting in fewer transitions of the dual trellis in comparison with the original (primal) trellis.

Here, we generalize the original idea of "dual APP" (DAPP) decoding to non binary alphabets. By non binary alphabet we mean that code symbols are taken from some extension field *$GF(q=p^m)$*, mostly binary extension field *$GF(q=2^m)$*.

The first work on *q*-ary LDPC codes appeared with Mackay and Davey. In this chapter, we address the non binary codes in conjunction with non binary modulation. We present a symbol-by-symbol *maximum a posteriori* (MAP) decoding rule which is, in a way, the dual of correlation-Viterbi decoding for linear codes. This code is also exhaustive, but in the sense that every word in the dual code is used in the decoding process. This means that in practice this decoding rule can be used only with codes whose dual code has a small number of code words, i.e., high rate codes or low-to-middle rate codes with short constraint lengths. The application of such non binary codes to orthogonal modulation is of particular practical importance. The decision rule does also play in the non iterative case a central role, and the crucial *extrinsic* information is also passed on as an *a priori* information to subsequent decoding iterations. And we describe how this complexity can be reduced using our decoding rules and make the computation faster with Fast Hadamard Transform (FHT) of the probabilities.

Finally, we present the Adaptive LDPC Coding addressed for 3rd Generation (3G) wireless systems. The use of Adaptive Modulation and Coding (AMC) is one of the key enabling techniques in the standards for 3rd-Generation (3G) wireless systems that have been developed to achieve high spectral efficiency on fading channels [74] – [77]. The core idea of AMC is to dynamically change the Modulation and Coding Scheme (MCS) in subsequent frames with the objective of adapting the overall spectral efficiency to the channel condition. The decision about selecting the appropriate MCS is performed at the receiver side according to the observed channel condition with the information fed back to the transmitter in each frame. In this chapter, we dynamically change only the modulation level during the encoding to take advantage of the whole of the capacity by using the water filling technique (see Appendix B), the set of candidate modulation are BPSK (binary data transmitted), 4-QAM, 8-QAM, 16-QAM, 64-QAM and 256-QAM with two dimensional Gray mapping. Where we consider that the characteristics of the channel are well-established and the channel can be probed to obtain a reliable channel quality estimate, the transmitter then uses this estimate to choose the appropriate signaling set.

Here, we don't enter in details of AMC; we will only compare the performances of an AMC if we dynamically adapt the modulation level in the modulator or in the encoding. In the first case, we consider at the transmitter a binary LDPC encoding and a modulator where the level size adapted to the channel profile, and at the receiver we compute the probabilities over each bit which will be used in the binary LDPC decoder. In the second case, we consider at the transmitter an adapted LDPC encoding where the level size will be adapted in the encoding matrix and a modulator, and at the receiver we compute the probabilities over each symbol and use this in a type of adapting LDPC decoder.

We will see that the decoding rule presented in section 8.1 can be easily applied to the non binary decoding and to the adaptive decoding and the performances obtained are very interesting.

8.1 The Decoding Rule

For convenience, we present the decoding rule for linear block codes. The extension to convolutional codes is simple.

Let $c = (c_0, c_1, ..., c_{n-1})$ denote any codeword of (n, k) linear block code ζ over $GF(p^m)$ and $c_j' = (c'_{j0}, c'_{j1}, ..., c'_{j,n-1})$ the j^th^ codeword of the $(n, n-k)$ dual code ζ'. A codeword c is transmitted over a time-discrete memory less channel with output alphabet B. The received word is denoted by $y = (y_0, y_1, \ldots, y_{n-1}), y_j \in B$.

The decoding problem is: given y, compute an estimate $\hat{c}_i$, of the transmitted code symbol c_i, that the probability that $\hat{c}_i$ equals c_i is maximized.

For the demonstration, readers are referred to Appendix C and D for more detailed list of definitions, lemma and theorems; we'll use the following <u>*notations*</u>:

- $q=p^m$; p is a prime number.
- $w = \exp[2\pi\sqrt{-1}/p]$ (primitive complex p^{th} root of unity);
- $\chi_\beta(\gamma) = w^{\tau(\beta.\gamma)}$ β and $\gamma \in GF(p^m)$; χ and $\underline{\chi}$ defined in Appendix D.
- $\delta_{ij} = 1$ *if* $i = j$ and zero otherwise;
- $Pr(x)$ is the probability of x; and $Pr(x|y)$ is the probability of x given y.
- Unless otherwise stated, the elements of $GF(q)$ are taken to be the integers $0, 1, ..., q-1$, and all arithmetic operations are performed in the field of complex numbers.

<u>Decoding Rule</u>: Set $\hat{c}_i = s$, where $s \in GF(p^m)$ maximizes the expression (demonstrate in the *proof*)

$$A_i(s) = \sum_{\beta\in GF(q)} \chi_\beta(-s) \sum_{j=1}^{q^{n-k}} \left[\prod_{l=0}^{n-1} \sum_{\gamma\in GF(q)} \chi_{(c'_{jl}-\beta\delta_{il})}(-\gamma)\Pr(y_l|\gamma) \right] \tag{8.1}$$

<u>*Theorem*</u>*:* Decoding rule (8.1) maximizes the probability that $\hat{c}_i$ equals c_i.

<u>*Proof*</u>*:* We must show that choosing s to maximize $A_i(s)$ is equivalent to maximizing the probability that c_i equals s given the received word y. We do this directly by showing that $Pr(c_i=s|y) = \lambda A_i(s)$, where λ is a positive constant which is independent of s.

We first note that

$$\Pr(c_i = s|y) = \sum_{c\in C, c_i=s} \Pr(c|y) = \sum_{c\in C, c_i=s} \Pr(y|c)[\Pr(c)/\Pr(y)] \tag{8.2}$$

Since the code words of ζ are equiprobable, $Pr(c) = p^{-mk}$ and (8.2) becomes

$$\Pr(c_i = s|y) = [p^{-mk}/\Pr(y)].\sum_{c\in C}\Pr(y|c)\delta_{0,(c.e_i-s)} \tag{8.3}$$

Where $e_i = (\delta_{i0}, \delta_{i1} \ldots \delta_{i,(n-1)})$ is the vector with one in the i^th^ position and zero elsewhere. In terms of their finite Fourier transforms,

$$\delta_{0,(c.e_i-s)} = q^{-1} \sum_{\beta \in GF(q)} \chi_\beta (c.e_i - s) \tag{8.4}$$

$$\Pr(y|c) = q^{-n} \sum_{u \in GF(q)^n} F(y,u) \underline{\chi}_u (c) \tag{8.5}$$

Where $q=p^m$, and

$$F(y,u) = \sum_{v \in GF(q)^n} \Pr(y|v) \underline{\chi}_u (-v) \tag{8.6}$$

$u = (u_0, u_1, ..., u_{n-1})$ and $v = (v_0, v_1, ..., v_{n-1})$ are any elements of V_n, the vector space of all n-tuples over $GF(p^m)$.

Substituting (8.4) and (8.5) in (8.3) yields to

$$\begin{aligned} \Pr(c_i = s|y) &= [p^{-m(n+k+1)} / \Pr(y)].\sum_{c \in C} \left[\sum_{u \in GF(q)^n} F(y,u) \underline{\chi}_u (c) \right] \left[\sum_{\beta \in GF(q)} \chi_\beta (c.e_i - s) \right] \\ &= [p^{-m(n+k+1)} / \Pr(y)]. \sum_{\beta \in GF(q)} \chi_\beta (-s) \sum_{u \in GF(q)^n} F(y,u) \left[\sum_{c \in C} \underline{\chi}_{(u+\beta.e_i)} (c) \right] \end{aligned} \tag{8.7}$$

By using the duality properties of group characters, we know that:

$$\sum_{c \in C} \underline{\chi}_v (c) \underset{v \in GF(q)^n}{=} \begin{cases} q^k = p^{m.k} & \mathit{If} \;\; v \in C^\perp \\ 0 & \mathit{elsewhere} \end{cases} \tag{8.8}$$

Applying (8.8) to (8.7) gives

$$\Pr(c_i = s|y) = [p^{-m(n+1)} / \Pr(y)]. \sum_{\beta \in GF(q)} \chi_\beta (-s) \sum_{c' \in C^\perp} F(y, c' - \beta.e_i) \tag{8.9}$$

Since the channel is memory less, we may write (8.6) as

$$F(y,u) = \sum_{v \in GF(q)^n} \prod_{l=0}^{n-1} \Pr(y_l|v_l) \chi_{u_l} (-v_l) = \prod_{l=0}^{n-1} \sum_{\gamma \in GF(q)} \Pr(y_l|\gamma) \chi_{u_l} (-\gamma) \tag{8.10}$$

Substituting (8.10) in (8.9) yields to

$$\Pr(c_i = s|y) = [p^{-m(n+1)} / \Pr(y)]. \sum_{\beta \in GF(q)} \chi_\beta (-s) \sum_{j=1}^{q^{n-k}} \left[\prod_{l=0}^{n-1} \sum_{\gamma \in GF(q)} \chi_{(c'_{jl} - \beta\delta_{il})} (-\gamma) \Pr(y_l|\gamma) \right] \tag{8.11}$$

$$\Pr(c_i = s|y) = [p^{-m(n+1)} / \Pr(y)].A_i(s) \qquad \square$$

As one might expect, the decoding rule takes a comparatively simple form;

In the binary case: set $\hat{c}_i = 0$, if $A_i(0) > A_i(1)$ and $\hat{c}_i = 1$ otherwise;

In the non binary case: set $\hat{c}_i = s$, if $\max\{A_i(0), A_i(1), ..., A_i(s), ..., A_i(q-1)\} = A_i(s)$.

The computation complexity of the algorithm described above can be reduced and the algorithm can be accelerate in the special case were we have a SPC code (its dual code is a repetition code). The idea is to use a Fast Hadamard Transform (FHT) in the proposed decoding rule and make it faster. And we obtain if *p=2*:

$$A_i(s) \cong \Pr\left(y_i \middle| s\right) * FHT\left\{\prod_{l=0,l\neq i}^{n-1} FHT\left[\Pr\left(y_l \middle| \gamma\right)\right]\right\} \tag{8.12}$$

Since the function $Pr(y_l/\gamma)$ is defined on *GF(q)*, FHT$[Pr(y_l/\gamma)]$ is a q-point FHT instead of q^k-point FHT, on account of the use of dual code. In the first layer, the FHT computes the sum and difference of the probabilities of two field elements differing from each other by only one bit location.

8.1.1 Decoding Rule applied to Single Parity Check Code (SPC)

We will take the single parity check "*SPC(6, 5)*" code over *GF(4)* which is a linear block code with a single parity check symbol.

Let $u = (u_0, u_1, ..., u_4)$ be the message to be encoded. The single parity check symbol is given by $u_5 = v/a_5$ and $v = a_0*u_0+ a_1*u_1+ ... + a_4*u_4$, where a_i elements of *GF(4)*, $a = [a_0, a_1 ... a_5]$ is the parity check matrix of the code *SPC(6,5)* and the sum *(+)* and product *(*)* are done in *GF(4).*

SPC codes are often used for simple error detection. Here we have chosen the SPC codes because their dual codes are repetition codes, i.e., the *SPC(6, 5)* and the repetition code *(6,* 1*)* are dual codes to each other. Furthermore, the repetition codes have a small number of code words.

We will illustrate the decoding rule for the received symbol y_0. The other received symbols $y_1, ..., y_5$ may be decoded simply by permuting the received word y in this equation.

$$A_0(s=i) \cong \Pr\left(y_0 \middle| i\right) * \underset{4}{FHT}\left\{\prod_{l=1,\gamma\in GF(4)}^{5} \underset{4}{FHT}\left[\Pr\left(y_l \middle| \gamma\right)\right]\right\}$$

Where $\underset{4}{FHT}$ is a 4-point FHT.

$\hat{s}_0 = i$, if max$\{A_0(0), A_0(1),A_0(2), A_0(3)\}=A_0(s=i)$, in the case of non iterative decoding.

8.1.2 Simulation Results for SPC(6, 5)

For the non binary SPC codes, the performance of the decoding rule are presented basis on a symbol-error-rate. Some simulation results for the white Gaussian channel suggest that the non binary encoding and decoding outperforms the binary equivalent especially when the modulation is not binary.

Fig. 8.1 and 8.2 illustrate this by showing up to 1 dB gain in the comparison between non binary *SPC(6,5)* and his binary equivalent when the modulation is q-QAM (q=4, 16, 64, 256). The improvement increases with the size of the modulation.

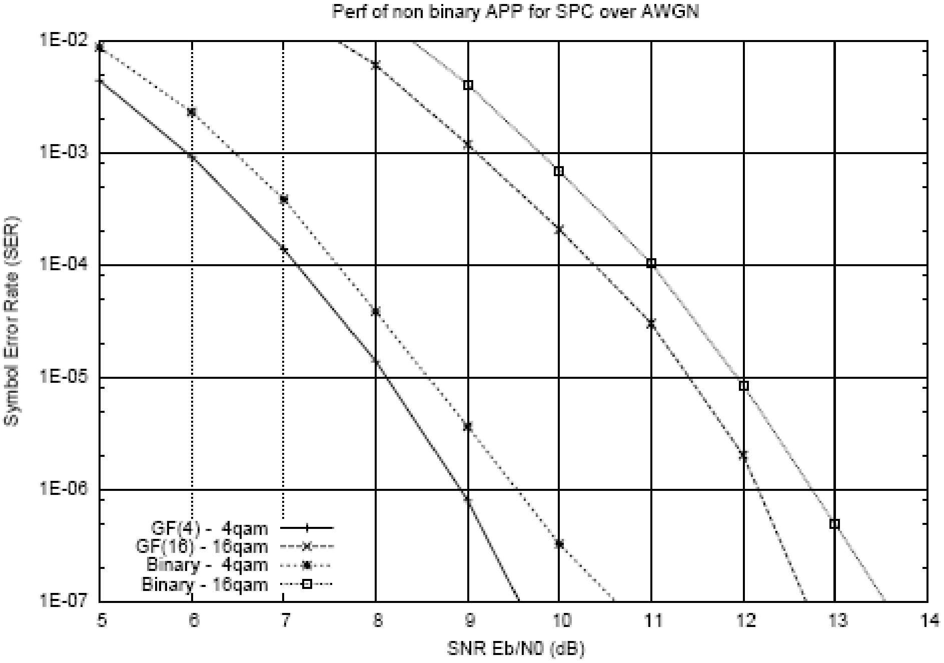


Figure 8.1: Comparison of *SPC(6, 5)* over *GF(4)* (resp. *GF(16)*) with *4-QAM* (resp. *16-QAM*) and its binary equivalent.

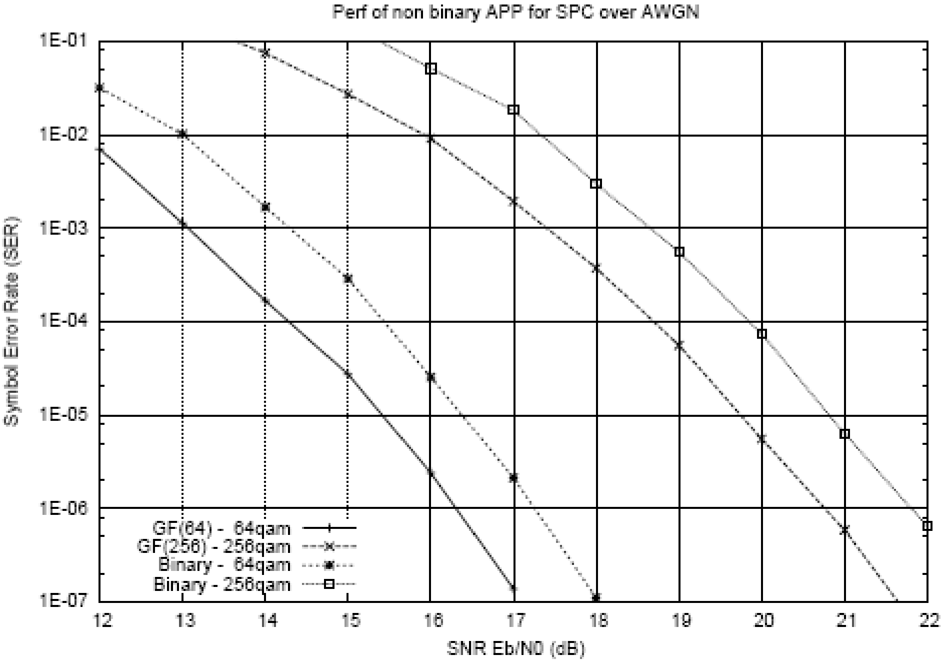


Figure 8.2: Comparison of *SPC(6, 5)* over *GF(64)* (resp. *GF(256)*) with *64-QAM* (resp. *256-QAM*) and its binary equivalent.

8.2 Non Binary LDPC Codes

While a substantial, and rapidly expanding, literature now exists on binary LDPC code construction and decoding, comparatively little is known about generalizations to non-binary, in which codeword symbols are selected from finite fields GF(q)*,* $q = 2^m$*,* $m \geq 2$ [72, 73]. In some of the earliest work on such q-ary LDPC codes, Davey and MacKay demonstrated that LDPC codes defined over non binary fields can substantially outperform binary LDPC codes over the binary symmetric channel (BSC) and additive white Gaussian noise (AWGN) channel [72, 73].

Davey showed that there is an optimum column weight which decreases as the order of the field increases, concluding that the best results could be generated by choosing the highest order field that is feasible and then selecting an appropriate mean column weight [73]. As is the case for most binary LDPC code constructions, once the optimal weight distribution has been determined, the q-ary codes are constructed randomly.

For very long codes this is not a problem as good codes are easily constructed randomly, and convergence to an ensemble average in the long codeword limit has been established [61]. However for short codes there is typically a significant performance gap between the best and worst codes of a particular ensemble and, especially for higher rate codes, good codes can be difficult to construct.

Mackay proved that LDPC codes can approach arbitrarily close to the Shannon limit if we choose sufficiently high column weight of H_{LDPC} *and then choose a sufficiently long block length.* However, as we increase the column weight, the number of cycles in the associated bipartite graph increases drastically which degrade the performance of our iterative decoding algorithm. By moving to *GF(q)* we manage to increase the mean column weight of the equivalent binary parity check matrix H_{LDPC} while retaining the same bipartite graph on which we perform the decoding, see Fig. 8.3. The drawback is that the decoding complexity is increased.

Another way of viewing the difference between binary and q-ary codes is that we increase the state space of each node in the decoding graph by decoding over *GF(q)*, which allows us to track correlations in the true posterior distribution that are not detectable by the binary algorithm. *Increasing the field order q for LDPC codes is comparable to increasing the memory of convolutional codes.*

It is worth pointing out the importance of the choice of non-zero elements in a parity check matrix defined over *GF(q)*. For example, if we choose them all to be ones then the graph of the equivalent binary code splits into m disjoint subgraphs. This split occurs because all the non-zero m x m blocks in the equivalent binary parity check matrix are identity matrices. In this case the decoding is equivalent to that of the binary algorithm applied to each subgraph.

Non-binary LDPC codes provide a new error control technology of combining the non-binary codes and the soft iterative decoding algorithm. In these LDPC codes over $GF(q=2^m)$, each code symbol contains m bits.

In principle, q-ary LDPC codes can be generated from Binary LDPC codes. By substituting each element one in the ***LxN*** parity check matrix H_{LDPC} for a Binary LDPC code with a nonzero element randomly chosen from *GF(q)*, a q-ary LDPC ***LxN*** parity check matrix H_{qLDPC} is obtained. In this way the density and girth properties of the randomly generated binary matrix, H_{LDPC}, are retained in the *q*-ary code. It is clearly that the *GF(q)* elements replacing the ones in H_{LDPC} cannot be all the same, otherwise the resultant *q*-ary LDPC code is simply composed of *m*-disjointed (also interleaved) Binary LDPC codes.

$$\begin{pmatrix} 1 & 1 & 0 \\ 3 & 0 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 \\ 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \end{pmatrix}$$

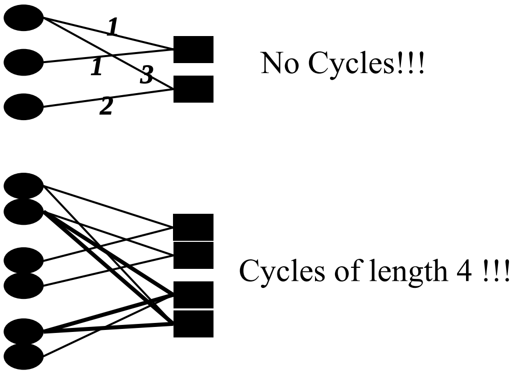


Figure 8.3: Comparison of corresponding graph structure (right) of a fragments parity check matrices over *GF(4)* and his binary equivalent (left). Note the presence of a short cycle in the graph for the binary code.

Conceptually, any Binary LDPC code (random or algebraic, regular or irregular) parity check matrix H_{LDPC} can be used to generate a *q*-ary LDPC code parity check matrix H_{qLDPC}.

The encoding of non-binary LDPC codes is similar to the encoding of binary LDPC codes, where the matrix multiplication is performed over the finite field *GF(q)*.

We can also use the same bipartite graph but allowing the variable nodes to take values from *GF(q)* and allowing the check nodes to impose constraints more complex than binary parity checks.

In the same time, any decoding method for Binary LDPC codes can be extended to *q*-ary LDPC by using the proper field operations. However, the efficient implementation of the belief propagation (BP) algorithm for Binary LDPC codes cannot be done for *q*-ary LDPC codes. This fact increases the decoding complexity of *q*-ary LDPC codes.

8.2.1 Encoding with *q*-ary LDPC matrix

Initially H_{qLDPC} is not in systematic form and is therefore reduced to the form

$$H_{qLDPC} = \left[P | I_L \right] \tag{8.13}$$

Using Gaussian elimination where I_L is the *L*x*L* identity matrix and *P* has dimensions *L*x*K* where $K = N - L$. All arithmetic operations are implemented over the finite field *GF(q)*. The *K*x*N* generator matrix can now be expressed as

$$G_{qLDPC} = \left[I_K | P' \right] \tag{8.14}$$

The encoding of non-binary LDPC codes is similar to the encoding of binary LDPC codes, where the matrix multiplication is performed over the finite field *GF(q)*.

8.2.2 Decoding Rule Apply to *q*-ary LDPC

In the case of *q*-ary LDPC, a lot of decoding algorithms was represented in the literature, where the structure of the algorithms matches the graphs directly. It will be convenient to think of these algorithms as parallel processing algorithms, where each site and each check is assigned its own processor and the communication between them reflects the Tanner graph. All these algorithms can be described by a local message-passing algorithm on the graph, and are iterative algorithms.

During the first half-iteration, each variable node v_i sends its *observations* m_v and its *a priori probabilities* m_{vc} given the input from all adjacent check notes $c_{j'}$, $j'\neq j$ to check node c_j.

During the second half-iteration, each check node c_j sends its *extrinsic probabilities* m_{cv} taking into account *observations* and *a priori probabilities* of all other adjacent variable nodes $v_{i'}$, $i'\neq i$, to variable node v_i.

At the end of the iteration, we compute *a posteriori probability* APP that the variable i is in state a by:

$$APP\left(y_i = a\right) \propto m_v^a . m_{cv}^a . m_{vc}^a \tag{8.15}$$

And we make the decision on the received symbols like this: $\hat{s}_i = a$, if max$\{APP(y_i=0), APP(y_i=1), \ldots, APP(y_i=a), \ldots, APP(y_i=q\text{-}1)\} = APP(y_i=a)$.

At variable nodes:

- The *observations* m_v^a given by the channel remain the same during all iterations.
- The *a priori probabilities* are updated with iterations:
    - ➢ For the first iteration, we fixed all m_{vc}^a to $1/q$
    - ➢ For the other iterations, m_{vc}^a are computed from m_{cv}^a

$$m_{vc}^a = \prod_{c'=0, c'\neq c}^{dv-1} m_{c'v}^a \quad \text{where it has } d_v \text{ adjacent check nodes.}$$

- Finally, the *APP probabilities* are computed by eqs (8.15).

At check nodes: (considered like a SPC)

The *extrinsic probabilities* m_{cv}^a are computed from:

$$m_{cv}^a = \underset{q}{FHT}\left\{ \prod_{v'=0, v'\neq v, b\in GF(q)}^{dc-1} \underset{q}{FHT}\left[m_v^b m_{v'c}^b \right] \right\} \tag{8.16}$$

Where dc is the number of its adjacent variable nodes.

We consider every check node in the tanner graph like a SPC code then its dual code is a repetition code which reduces the computation complexity of the algorithm described above. This algorithm is accelerated in relation to other decoding algorithm in the *q*-ary LDPC codes by using a Fast Hadamard Transform (FHT) in the proposed decoding rule.

The complexity of the decoding algorithm for *q*-ary LDPC codes scales as q^2. This complexity is reduced to $q\log q$ using our decoding rules especially with FHT of the probabilities.

8.2.3 Simulations Result

Davey in his thesis suggests that carefully constructed non-binary LDPC codes are likely to outperform their binary counterparts.

For evaluation, we have applied the decoding rule to:

- We choose three LDPC codes of rate R=1/3 over fields *GF(2), GF(4)* and *GF(8)* over binary Gaussian channel. Note that these codes have a block length of 18000 bits. For comparison we include the performance of a rate 1/3 turbo code with the same block length.
- *q*-ary LDPC codes (*q*=4, 8, 16, 64 and 256) and compared these to their binary equivalent LDPC with *q*-QAM modulation, we take the same rate 1/2 for regular (3, 6) codes and the same block length *N=4000* in term of bits, we include the performance of turbo code.

The results in Fig. 8.4 show that LDPC codes over higher order fields (*GF(4)* and *GF(8)*) can significantly outperform binary LDPC codes of similar block length (N=18000 bits). At 10^{-4}, an improvement of 0.35 dB is shown for the rate R=1/3 code moving from binary to *GF(8)* construction, halving the distance to the 16-state turbo code performance over a Binary Gaussian Channel.

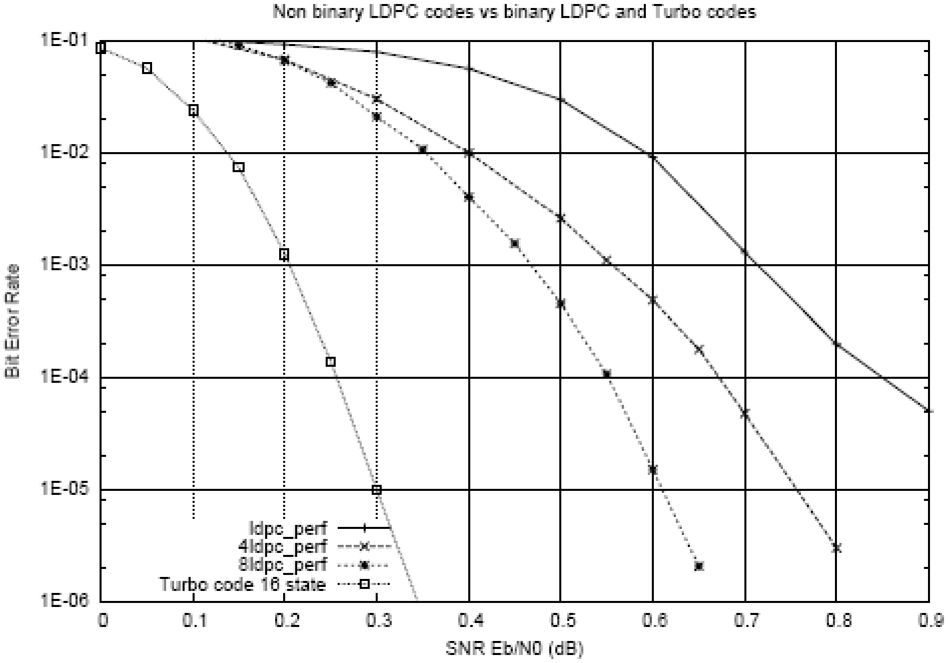


Figure 8.4: Comparison of LDPC codes over a Binary Gaussian Channel over GF(2) "Binary", GF(4) and GF(8).

Next, we will take an AWGN channel, the LDPC codes like the *q*-ary LDPC codes are rate 1/2 regular (3, 6) codes, and the UMTS turbo code used is rate 1/2. We compare *q*-ary LDPC codes of block length N symbols over *GF(q)* , i.e., length 1000 symbols over *GF(16)*, with binary LDPC codes of length 4000 bits. Modulation used is *q*-QAM, i.e., with a code over *GF(16) 16*-QAM modulation is used.

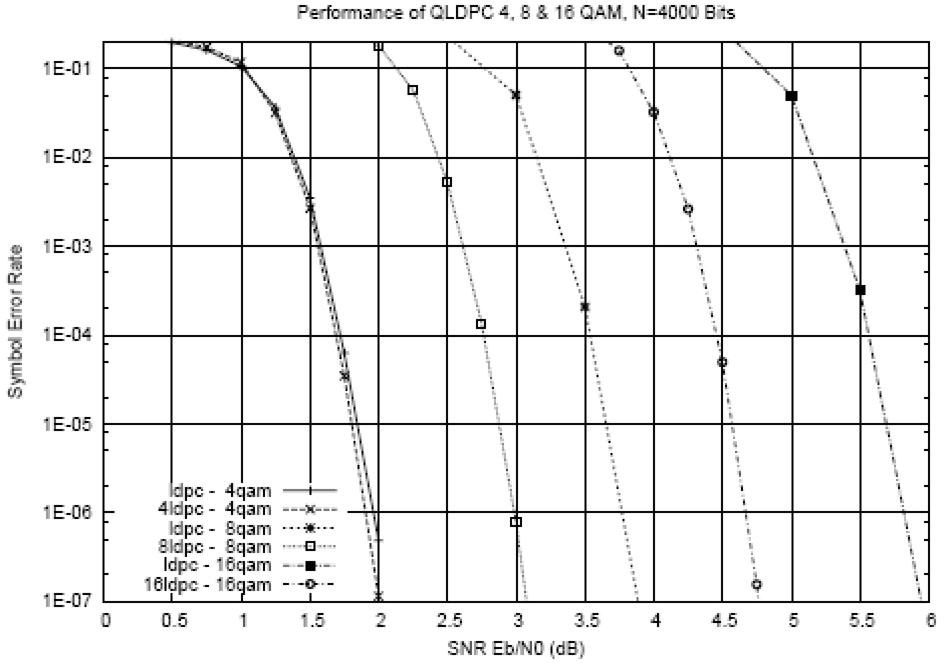


Figure 8.5: Performances of *q*-ary LDPC vs LDPC in 4, 8 and 16QAM modulation over AWGN channel.

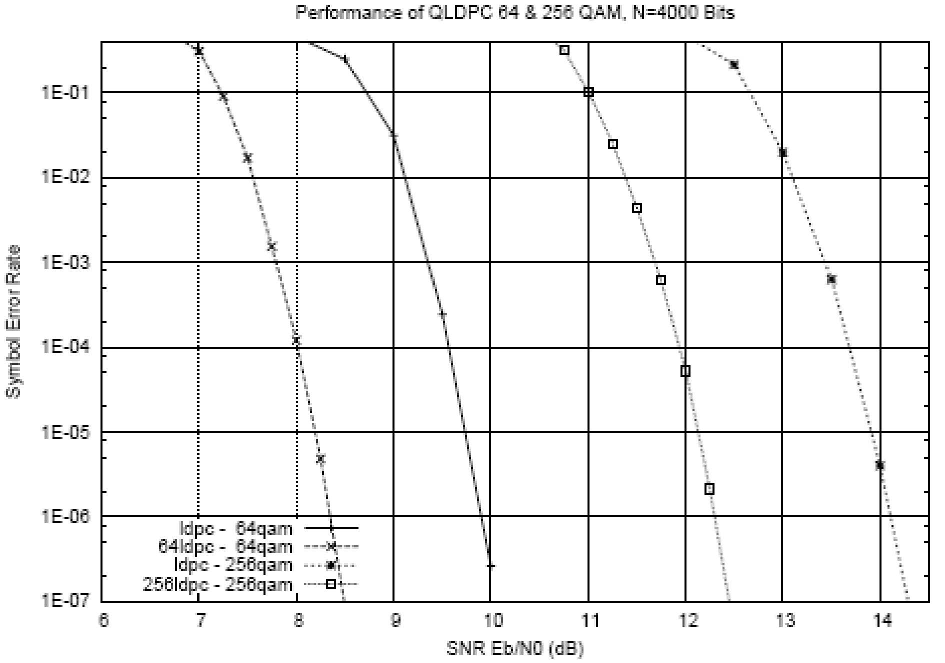


Figure 8.6: Performances of QLDPC vs LDPC in 64 and 256QAM modulation over AWGN channel.

Fig. 8.5 and 8.6 compare the performance of QLDPC codes over *GF(4), GF(8), GF(16), GF(64),* and *GF(256)* with LDPC codes. We note that QLDPC codes outperform LDPC codes practically in all the cases and when *q* grows the improvement increase highly. For *q=4,* we practically don't obtain any gain, for *q=8,* we gain more than 0.75dB at 10^{-4}. This gain increases with q, up to 2 dB for *q=256*.

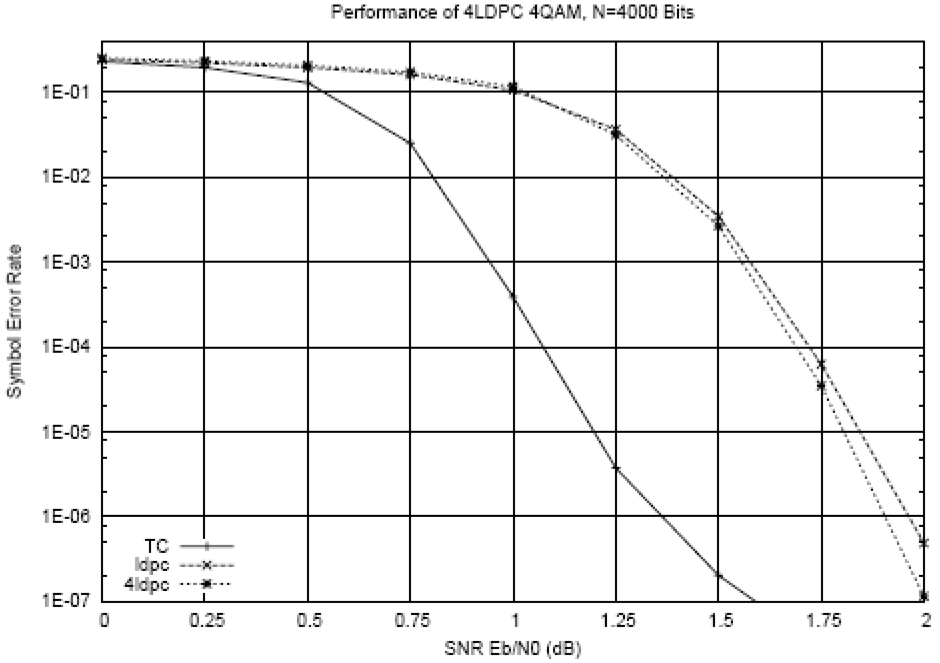


Figure 8.7: Performances of *4*-ary LDPC vs Turbo Code in 4QAM modulation over AWGN channel.

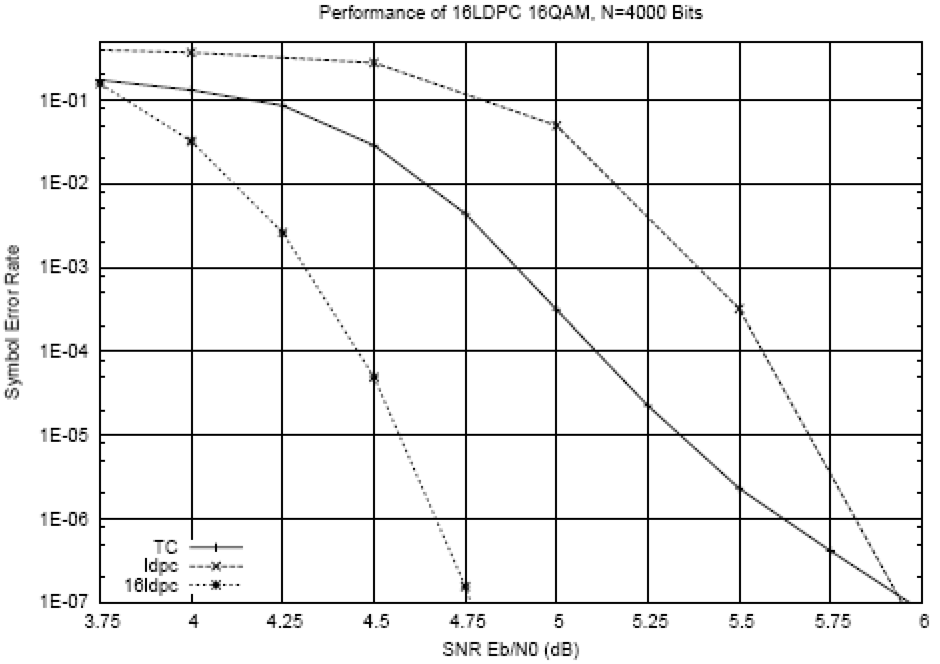


Figure 8.8: Performances of 16-ary LDPC vs Turbo Code in 16QAM modulation over AWGN channel.

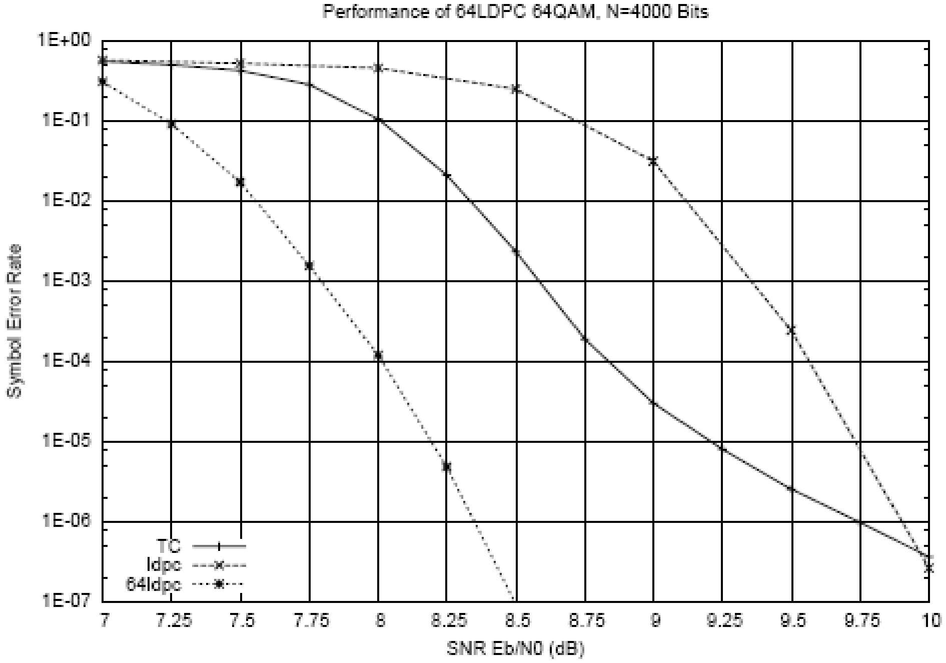


Figure 8.9: Performances of 64-ary LDPC vs Turbo Code in 64QAM modulation over AWGN channel.

Fig. 8.7 shows that over *GF(4),* the turbo code outperforms the QLDPC code by up to 0.6 dB at 10^{-4}, which mean that we have to try to change the characteristics of QLDPC or used an irregular *q*-ary LDPC. In return, in Fig. 8.8 (resp. 8.9), the *q*-ary LDPC codes over *GF(16)* (resp. *GF(64)*) are better than the turbo codes by up to 0.6 dB (resp. by 0.8 dB), despite that the binary LDPC code is worth than the turbo code.

8.3 Adaptive LDPC Codes

High speed wireless data transmission requires robust and spectrally efficient communication techniques for flat-fading channels. When the channel can be estimated and this estimate sent back to the transmitter, the transmission scheme can be adapted relative to the channel characteristics.

Most modulation and coding techniques do not adapt to fading conditions. These non adaptive methods require a fixed link margin to maintain acceptable performance when the channel quality is poor. Thus, these systems are effectively designed for the worst case channel conditions, resulting in insufficient utilization of the full channel capacity. Adapting to the signal fading allows the channel to be used more efficiently since power and rate (only the power in our case) can be allocated to take advantage of favorable channel conditions.

Many AMC techniques have been presented in the literature. In the following, we provide a brief description of some of these papers that are more relevant to this section. Readers are referred to [78] for a more detailed list of references on this topic.

Adaptive transmission, which requires accurate channel estimates at the receiver and a reliable feedback path between the receiver and transmitter, was first proposed in the late 1960's [80]. Interest in these techniques was short lived, perhaps due to hardware constraints, lack of good channel estimation techniques, and/or systems focusing on point-to-point radio links without transmitter feedback. The fact that these issues are less constraining in current systems, coupled with the growing demand for spectrally efficient communication, has revived interest in adaptive modulation methods. The basic idea behind adaptive transmission is to maintain a constant by varying the transmitted power level [80], symbol transmission rate [81], constellation size [82]–[84], coding rate/scheme [85], or any combination of these parameters [86]–[88]. Thus, without sacrificing bit-error rate (BER), these schemes provide high average spectral efficiency by transmitting at high speeds under favorable channel conditions, and reducing throughput as the channel degrades. The performance of these schemes is further improved by combining them with space diversity [89], [93]. Adaptive techniques are also used for high-speed modems [90], satellite links [91]. Our approach is novel relative to all of these adaptive techniques in that we optimize *both* the transmission rate and power to maximize spectral efficiency, while satisfying average power and BER constraints.

In [79] and [92], various rate and power adaptation schemes are investigated. The power adaptation policy found is essentially a water-filling formula in time. In [92], a variable-power variable-rate modulation scheme using M-ary Quadrature Amplitude Modulation (M-QAM) is proposed. The presented results show that the proposed technique provides a 5-10 dB gain over variable-rate fixed-power modulation using channel inversion and truncated channel inversion (where the received power is maintained constant), and up to 20 dB gain over the non adaptive modulation. In [79], the optimal adaptive transmission scheme which achieves the Shannon capacity of a fading channel was derived. In this work, we develop practical variable-rate variable-power MQAM modulation techniques for fading channels inspired by the capacity results in [79]. In [93], the channel capacity of various adaptive transmission techniques is examined. The performance of these techniques employed with space diversity is also investigated. It is shown that the spectral efficiency for a fading channel can be improved by adaptive transmission techniques in conjunction with space diversity. It is also found that when the transmission rate is varied continuously according to the channel condition, varying the transmit power at the same time has minimal impact. In [94], the adaptation technique from [79] and [92] is modified to take into account the effect of constrained peak power. Simulation results show that with a reasonable peak power constraint, there is a small loss in spectral efficiency as compared to the unconstrained case. In [95], an AMC scheme is proposed based on the variable-power variable-rate technique from [79] and [92]. This technique superimposes a trellis code on top of the uncoded modulation. Simulation results show that with a simple four-state trellis code, an effective coding gain of 3 dB can be realized.

8.3.1 Adaptive Modulation and Coding

A baseband system model for the problem is shown in Fig. 8.10. The independent and identically distributed (IID) sequence $\left(b_i\right)$ is the stream of information bits to be transmitted over the channel, and $\left(\hat{b}_i\right)$is the corresponding stream of information bit estimates at the output of the receiver. Each information bit is assumed to be equally likely to be 0 or 1. The fading is modeled as a complex multiplier $h(t)$, thus implying a frequency non-selective channel appropriate for a narrowband wireless channel or a single subchannel of a multicarrier system. The channel is modeled as a Rayleigh fading channel, which is appropriate for narrowband mobile systems or indoor systems without a line-of-sight component. If there is a line of-sight component, the fading in indoor systems is Rician, but it will be clear from the characterization of the effects of the channel variation that extensions to the Rician case are trivial. Furthermore, it can be shown that when there is uncertainty about the value of the Rician factor, the Rayleigh assumption will lead to signaling for the worst case. It is believed that this assumption will be true in any system where the use of adaptive coding is considered; however, as in the consideration of Rician fading, the modifications when this assumption is altered are conceptually minimal. Coherent reception with perfect carrier phase estimation and perfect fading value estimation *at the receiver* is assumed throughout this work.

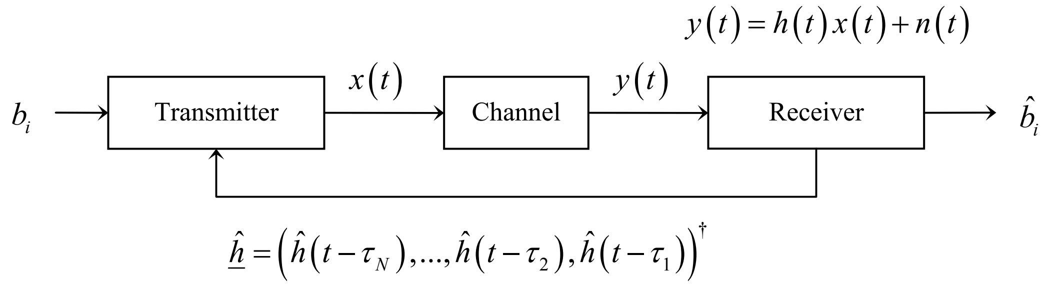

$y(t)=h(t)x(t)+n(t)$
b_i
Transmitter
$x(t)$
Channel
$y(t)$
Receiver
$\hat{b}_i$
$\underline{\hat{h}}=\left(\hat{h}(t-\tau_N),\ldots,\hat{h}(t-\tau_2),\hat{h}(t-\tau_1)\right)^{\dagger}$

Figure 8.10: The baseband Adaptive system model.

The key difference between the system model of Fig. 8.10 and that of a standard communication system is the availability at the transmitter of the vector $\underline{\hat{h}}$ of channel fading estimates, where $\tau_{i+1} > \tau_i$, $\forall i$. The availability of true channel estimates as opposed to estimates of a filtered version of the channel is based on the assumption that the channel varies slowly enough to be assumed constant over the duration of a symbol (and thus estimation) period. These fading estimates can be obtained via literal feedback of measured fading values from the receiver or can be estimated using any signal sent from the current receiver to the current transmitter: a pilot signal at the end of initial handshaking, packet acknowledgment signals, or data sent from the current receiver during the previous slot in a time-division duplex (TDD) system.

The detailed system model is illustrated in Fig. 8.11. We assume that an estimate $\hat{h}(t)$ of the channel power gain $h(t)$ at time *t* is available to the receiver after an estimation time delay of τ_i and that this same estimate is available to the transmitter after a combined estimation and feedback path delay. We assume that the feedback path does not introduce any errors, which can be assured by increasing its delay time and using an ARQ transmission protocol. The availability of channel information at the transmitter allows it to adapt its transmission scheme relative to the channel variation. The rate of channel variation will dictate how often the transmitter must adapt its rate and/or power.

For a specific example of such an adaptive system, consider the transmission of data from a mobile terminal to the access point in a wireless local area network (WLAN) that employs a multicarrier strategy. The system protocol generally involves handshaking that ends with the access point sending the mobile terminal a message that informs the mobile terminal that it can begin to send data. The mobile terminal can use this message sent from the access point to measure the current fading on each subcarrier and to prepare its transmitted signal. The mobile terminal then codes the (now outdated) channel measurements for each subcarrier with a non adaptive code, interleaves the resulting coded bits across subcarriers to obtain frequency diversity, and sends the resulting bits as a prelude to the actual coded data.

The access point decodes the channel measurements for each subcarrier, determines the sizes of the signal sets employed on each subcarrier at each delay by running an algorithm on the channel measurements identical to that being employed at the mobile terminal, and then decodes the data.

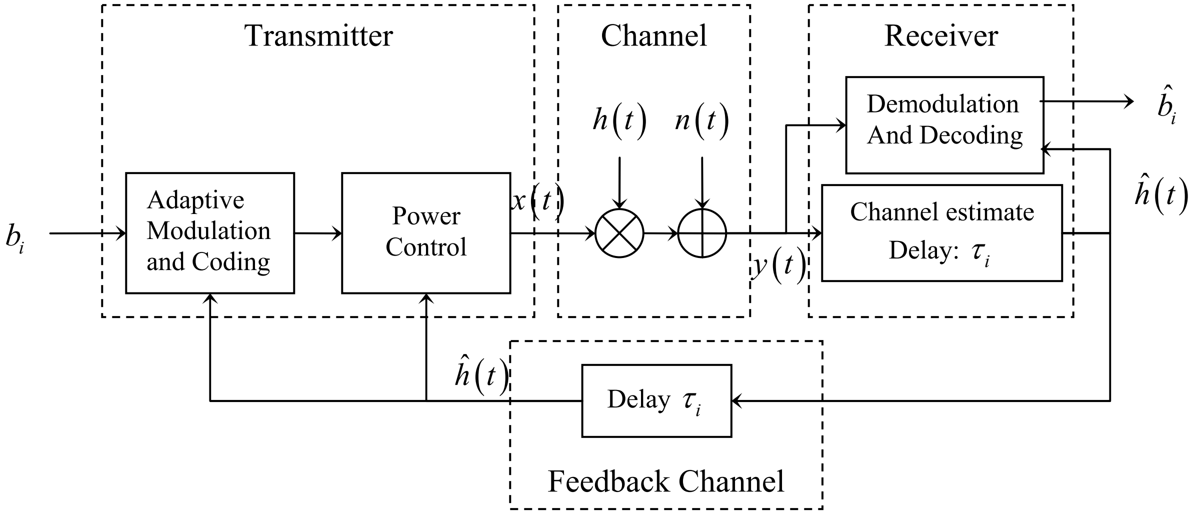


Figure 8.11: Detailed system model.

8.3.2 Adapting constellation in the encoder

Until this section, we spooked about adapting constellation set in the modulator. Here, we will propose to adapt the constellation set in the encoder. The new adaptive modulation and coding is illustrated in Fig. 8.12 by the dotted line. This proposition can be done with any block linear code, in our case; we choose an LDPC code named *adaptive LDPC codes*. Since, we adapt the constellation set before the construction of matrix H_{LDPC}, this solution is much more important in the case of static channel (Non varying channel).

A. Design Rules

Assume for illustrative purposes that the set of candidate signal sets are BPSK (binary data transmitted or 2-QAM), 4-QAM, 8-QAM, 16-QAM, 64-QAM and 256-QAM with two-dimensional Gray mapping. Here, we will explain a simple design rule, we can find more complex in the literature and we present another design rule in Appendix E.

This design rule is based on the value of the capacity, we will considered the channel divided to N_c OFDM subcarriers (Here 256 subcarriers) where we have to compute the channel frequency response of each subcarrier

$$H(f)=\sum_{i=0}^{l-1}h_i.e^{-2\pi f\Delta_i} \tag{8.17}$$

Where l is the length of channel or the number of taps, Δ_i is the physical delay of the channel. f depends on N_c and we consider that it varies from –0.5 to 0.5 with a step equal to $1/N_c$.

Next, we compute the complex square modulus $\|H(f)\|^2$ the value of the capacity over each subcarrier on a certain SNR

$$C_h=\log_2\left(1+\|H(f)\|^2.10^{SNR}\right) \tag{8.18}$$

And we compare it to the coding rate multiplied to the number of bit in a modulation, i.e. in the case of 4-QAM, the capacity is compared to 2*coding rate, for 8-QAM, the capacity is compared to 3*coding rate and so on.

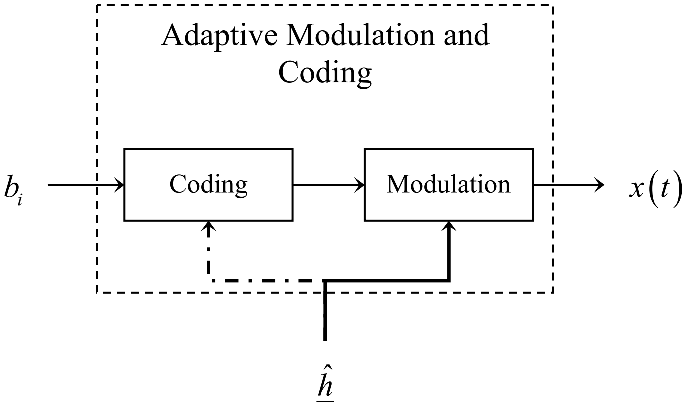


Figure 8.12: Diagram block of AMC, Solid line correspond to adapting in the modulator and Dotted line to adapting in the encoder (Proposed).

To give an example we choose the 12-tap urban area channel from GSM specifications, the number of taps is 12 and it is represented by table 8.1 where the physical delay of each tap in multiple of 0.1 microseconds and the power of each tap in dB.

Tap index	1	2	3	4	5	6	7	8	9	10	11	12
Delays	0	1	3	5	8	11	13	17	23	31	32	50
Powers	-4	-3	0	-2.6	-3	-5	-7	-5	-6.5	-8.6	-11	-10

Table 8.1: Urban area channel characteristics

The table 8.2 represents the number of subcarriers for each modulation at SNR=7dB and rate=0.5 computed by applying the simple design rule explains below.

Coding Rate=0.5	N_2 "2-QAM"	N_4 "4-QAM"	N_8 "8-QAM"	N_{16} "16-QAM"	N_{64} "64-QAM"	N_{256} "256-QAM"
SNR=7dB	35	25	35	80	50	24

Table 8.2: Number of subcarriers for each modulation at SNR=7dB and Rate=0.5.

These numbers are considered in the construction of the adaptive LDPC where they correspond to the number N_{2^i} of columns, where i is number of bits in the modulation.

B. Matrix H_{LDPC} Construction

In the case of adaptive LDPC codes, the construction of the matrix H_{LDPC} is one of the problems to take care. Here, we will enumerate and explain some of construction conditions and rules. An example of adaptive LDPC matrix is presented in Fig 8.13.

- The subchannels are distributed on the columns of the matrix, which mean that the nonzero elements on each column belong to the same Galois Field. From the point of view of "tanner graph", each variable node belongs to one of the selected Galois field. The length of code *N* will be divided to several set of N_{2^i} columns, where *i* is the number of bits in the modulation.

- In the "tanner graph", on each check node the computing will be done in a Galois field, which means that we can't mix any two Galois fields. It is necessary that the Galois fields mixed in each check node respect the following condition: the largest Galois field is an extension of the other. From the point of view of the matrix H_{LDPC}, the nonzero elements on each row belong to the larger Galois field of this row. For example, *GF(2), GF(4), GF(16)* and *GF(256)* together. Or *GF(2), GF(8)* and *GF(64)* together.

$$H_{LDPC} = \begin{array}{cccccccc|l}
GF(2) & GF(4) & GF(8) & \cdots & GF(16) & \cdots & GF(64) & GF(256) & \\
\hline
0 & 3 & 0 & & 13 & & 0 & 123 & GF(4), GF(16), GF(256) \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \\
0 & 0 & 5 & \cdots & 0 & \cdots & 33 & 0 & GF(8), GF(64) \\
1 & 1 & 0 & & 11 & & 0 & 99 & GF(2), GF(4), GF(16), GF(256) \\
\vdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \\
1 & 2 & 0 & \cdots & 9 & \cdots & 0 & 53 & \\
1 & 0 & 3 & & 0 & & 25 & 0 & GF(2), GF(8), GF(64)
\end{array}$$

Figure 8.13: An example of adaptive LDPC matrix.

State Variable Nodes

For these conditions and to avoid the case where we obtain several subgraphes disconnected between them, we will define some state variable nodes which are binary variable nodes used to connect the subgraphes and not transmitted over the channel, in a way that is look like a puncturing. At the receiver, the observations of these state variable nodes are considered equal to 0.5. The number of state variable nodes depends on the distribution degree *(d_v, d_c)*.

Next, we use the integer linear programming to compute the number of check nodes belong to each Galois field, by forcing some constraints to the linear programming.

For the example of urban area channel defined in table 8.1, with rate 0.5 and distribution degree *(3, 6),* the number of state variable nodes is equal to 36. And the following table presents the number of check nodes belong to each Galois field L_{2^i} computing by integer linear programming.

L_2 "GF(2)"	L_4 "GF(4)"	L_8 "GF(8)"	L_{16} "GF(16)"	L_{64} "GF(64)"	L_{256} "GF(256)"
0	16	24	54	34	14

Table 8.3: Number of check nodes for each Galois field at SNR=7dB, Rate=0.5 and *(3, 6)*.

C. Adaptive LDPC Encoding

The Encoding will be done with the binary equivalent matrix H_{bLDPC} of the original adaptive LDPC matrix H_{LDPC}, so first we have to calculate H_{bLDPC}: *(to remind, we consider 6 subchannels and the selected constellation sets are 2-QAM, 4-QAM, 8-QAM, 16-QAM, 64-QAM and 256-QAM)*

If N is the number of columns of the original LDPC matrix, K is the size of the adaptive LDPC code, and the number of adaptive matrix rows $L=N\text{-}K$. The number of columns of the equivalent binary matrix is:

$$N_b = \sum_{\substack{i=1 \\ i\neq 5,7}}^{8} i \times N_{2^i} \tag{8.19}$$

Where N_{2^i} is the number of columns which contain the Galois field $GF(2^i)$, i is the number of bits in the $GF(2^i)$ symbol, the $GF(2^i)$ means that we have selected the constellation set 2^i*-QAM*.
The number of binary equivalent matrix rows is:

$$L_b = \sum_{\substack{i=1, \\ i\neq 5,7}}^{8} i \times L_{2^i} \tag{8.20}$$

L_{2^i} is the number of rows where the larger Galois field is $GF(2^i)$.
The size of the binary equivalent LDPC code is:

$$K_b = N_b - L_b \tag{8.21}$$

The rate of the Adaptive LDPC code is:

$$R_b = \frac{K_b}{N_b} \neq \frac{K}{N} \tag{8.22}$$

The binary equivalent matrix H_{bLDPC} is obtain with replacing the nonzero elements of $GF(2^i)$ in the original adaptive matrix H_{LDPC} by a matrix $(i \text{ x } i)$ and filling the rest by zeros.

The matrix H_{LDPC} is not in systematic form and is therefore reduced to the form

$$H_{LDPC} = \left[P \middle| I_{L_b} \right] \tag{8.23}$$

Using Gaussian elimination where I_{L_b} is the L_bxL_b identity matrix and P has dimensions L_bxK_b. The K_bxN_b generator matrix can now be expressed as

$$G_{LDPC} = \left[I_{K_b} \middle| P' \right] \tag{8.24}$$

The encoding reaches to the encoding of simple binary LDPC codes.

D. Adaptive LDPC Decoding

For the adaptive LDPC decoding, we use the same algorithm used in the non binary LDPC decoding case (see section 8.2.2). We don't have to change anything, but we have to make attention that we work with several Galois fields.

E. Toy Example

To test the adaptive LDPC, the matrix or the graph construction and above all, the use of state variable nodes in this construction, we take an example which we will call it a Toy example.

In this Toy example, we will compare three codes represented in Fig. 8.14: first a little 4-ary code $\zeta_4(N_4=3, L_4=2, K_4=1)$, second a little 8-ary code $\zeta_8(N_8=3, L_8=2, K_8=1)$, and finally a little adaptive code ζ_{Adap} which links the previous two codes, seeing that GF(4) and GF(8) are incompatible we decide to use $N_2=2$ state variable nodes which are linked to both 4-ary check nodes and 8-ary check nodes and linked between them by $L_2=1$ binary check node, like it is presented in Fig. 8.14.

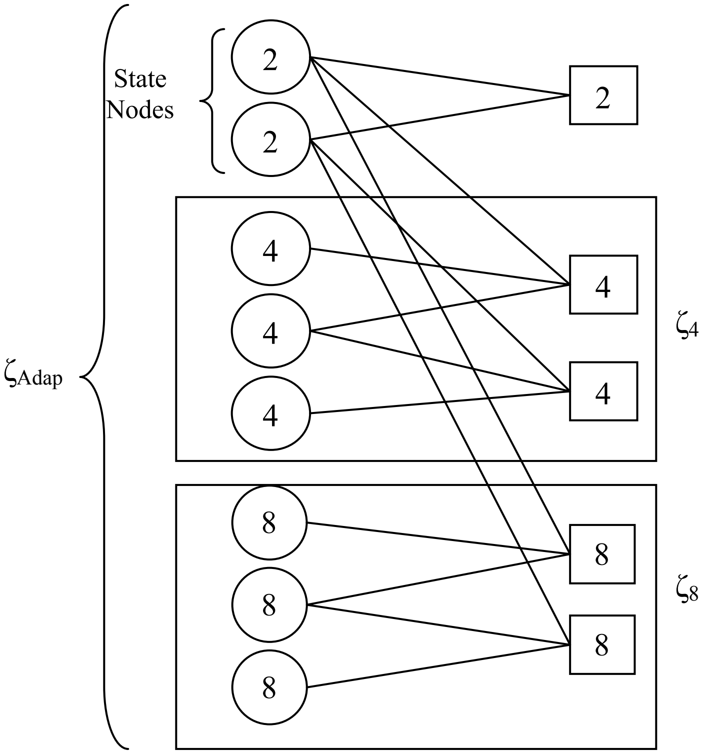


Figure 8.14: The Tanner graph of the Toy Example.

The rate of ζ_4 is equal to the rate of ζ_8 equal to 1/3. For the adaptive code ζ_{Adap}:

$$N_b = N_2 + 2 * N_4 + 3 * N_8 = 2 + 6 + 9 = 17$$

$$L_b = L_2 + 2 * L_4 + 3 * L_8 = 1 + 4 + 6 = 11$$

$$K_b = N_b - L_b = 17 - 11 = 6$$

$$R_c = \frac{K_b}{N_b} = \frac{6}{17} = 0.353 \Rightarrow R_{eff} = \frac{K_b}{N_b - N_2} = \frac{6}{15} = 0.4$$

$$H_4 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix}, \qquad H_8 = \begin{pmatrix} 1 & 3 & 0 \\ 0 & 5 & 6 \end{pmatrix}, \qquad H_{Adap} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 5 & 6 \end{pmatrix}.$$

To compare those three codes, we simulate them over an AWGN channel and we show in Fig. 8.15 the BER performances of those codes where we easily remark that the adaptive code improve the performances of the two non binary codes, ζ_8 makes up to 1 dB at BER=10^{-4} in relation to ζ_4 by linking those two non binary codes in a one adaptive code ζ_{Adap}, we obtain a important gain equal to more than 0.25 dB at BER=10^{-4} in relation to ζ_8.

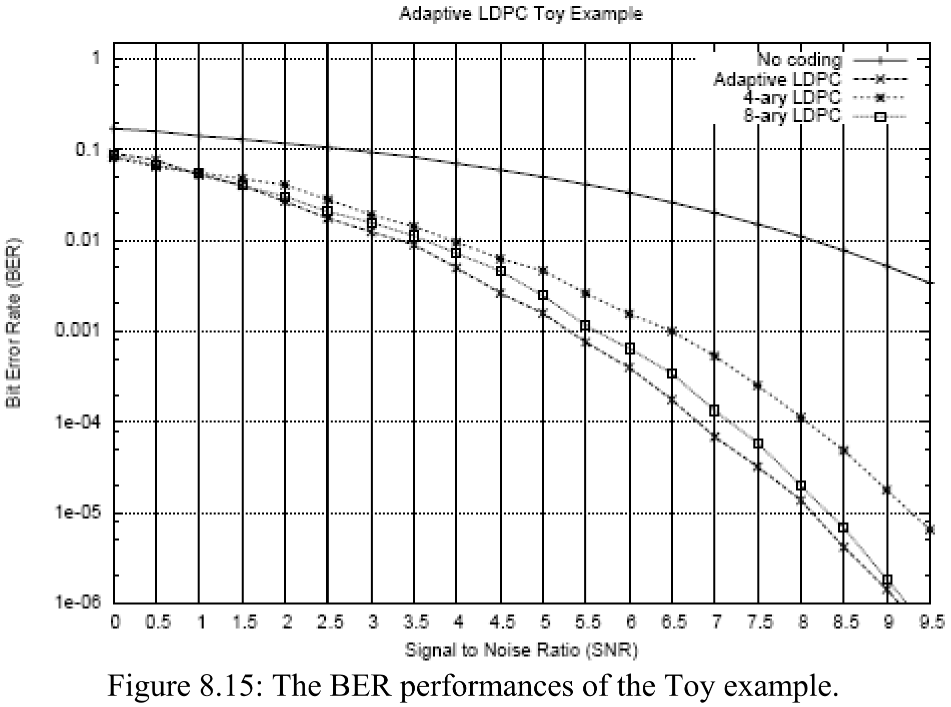


Figure 8.15: The BER performances of the Toy example.

F. Simulation Results

Fig. 8.16 and 8.17 show the BER end FER performances of an adaptive LDPC code by considering an OFDM system with N=300 subcarriers which are divided to N_2=75 and 5 equal number $N_{2^i} = 45$ where i =2, 3, 4, 6 and 8 and N_{2^i} is the number of subchannels which allow a GF(2^i) symbol. The LDPC code is adapted to have N_2=75 bit nodes and $N_{2^i} = 45$ variable nodes from each remaining Galois field. For the moment we consider that we don't have a state variable node which means that all variables are transmitted over the channel.

For this adaptive LDPC code, the number L_2=0 of binary check nodes and the numbers L_{2^i} of 2^i-ary check nodes are equal to 30 for i =2, 3, 4, 6 and 8 computing with integer linear programming, this imply that the characteristics of this adaptive LDPC are: $N_b = 1110$, $L_b = 690$ and $R_c \simeq 0.4$.

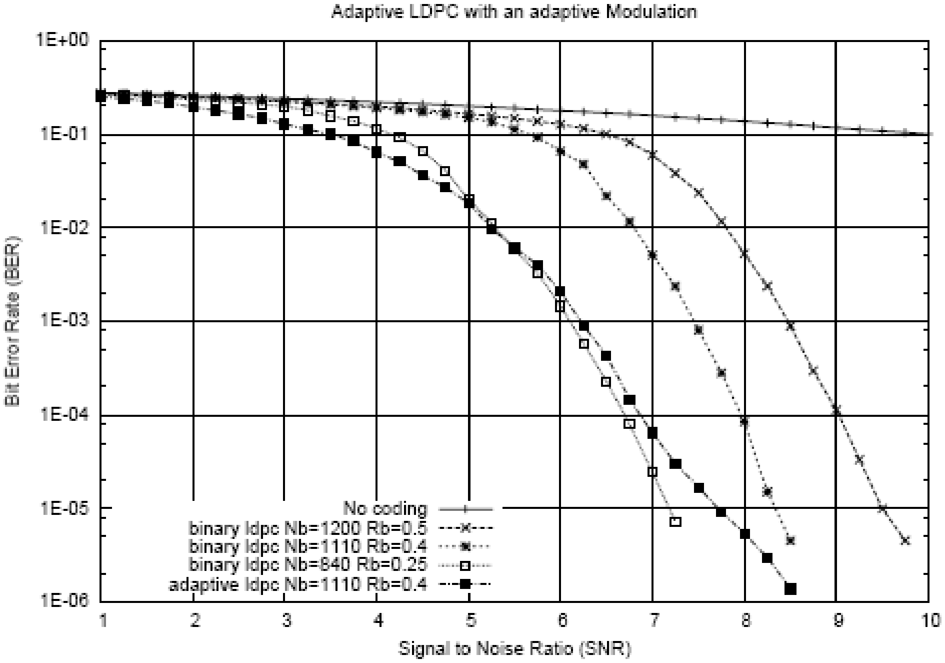


Figure 8.16: The BER performances of adaptive LDPC code over an OFDM channel with 300 subcarriers.

We compare this adaptive LDPC code with the systems where we use a binary LDPC code and an adaptive modulation. The binary LDPC codes considered have coding rate equal to 0.5, 0.4 and 0.25. We remark by Fig. 8.16 and 8.17 that the adaptive LDPC code improves the performance by up to 1dB at BER=10^{-4} in relation to binary LDPC code with rate 0.4 and also approaches closely to one with rate 0.25. The gain when we compare the FER performances.

The performances of the adaptive LDPC codes approach at high SNR to those of binary LDPC code with rate 0.4. We think that the reason is, at high SNR, the hamming distance of the binary variables dominates the others.

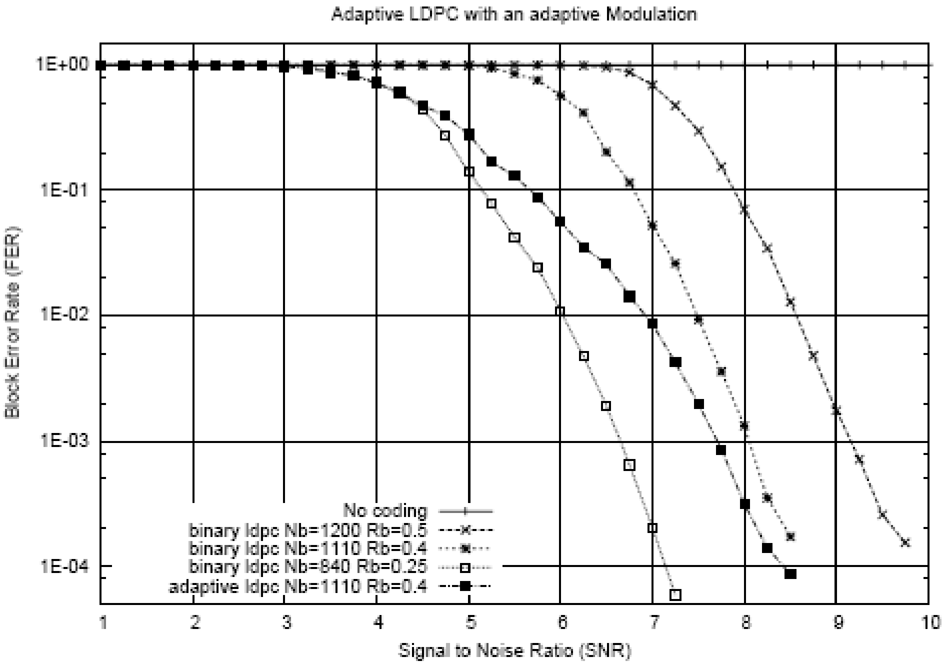


Figure 8.17: The FER performances of adaptive LDPC code over an OFDM channel with 300 subcarriers.

8.4 Conclusion

We have presented a symbol-by-symbol decoding rule for non binary codes, and the performance obtained basis on symbol-error-rate. Simulation results for the white Gaussian channel suggest that the non binary LDPC (resp. SPC) encoding and decoding outperforms the binary LDPC (resp. SPC) especially when the modulation is not binary, and prove that even when the turbo code outperforms the binary LDPC, the q-ary LDPC codes can recover this loss and go beyond. The q-ary LDPC (resp. q-ary SPC) with this decoding rule outperforms the LDPC (resp. SPC) code by more than 2 dB (resp. 1dB) over *GF(256)*, despite that they can't outperform LDPC (resp. SPC) over *GF(4),* and this gain increases with the length of the Galois Field, i.e. the size of the modulation.

Next, we have shown that the q-ary LDPC code outperforms the UMTS turbo code over GF($q>4$) by 0.6 dB or 0.8 dB depending on q.

Considering the complexity, the decoding rule is practical for high rate, applicable to non binary LDPC codes and reduces enormously the complexity by using the duality properties and accelerates the computation buy using the FHT. The complexity of the decoding rule for an (n, k) linear code is comparable to the complexity of a Viterbi decoder for the (n, n-k) dual code.

Finally, we proposed for OFDM system, adaptive LDPC codes which can be easily decoded by the decoding rule presented in this chapter. Despite that the performances of the adaptive LDPC codes approach at high SNR in some cases to binary LDPC, it remains a very attractive subject to the future research.

Chapter 9
Iterative OS Technique with Non Binary LDPC Codes

For Inter Symbol Interference (ISI) channels, various outer codes have been considered for use in turbo equalization, including parallel turbo codes [96] [97], convolutional codes (also known as serial turbo equalization) [98], parity check codes [99] and most recently LDPC codes [100] [101].

Low Density Parity Check (LDPC) codes are strong codes which have excellent performance on a variety of channels, and when decoding on the ISI channel, bit-error rates can be improved further by using turbo equalization.

Sequences of LDPC codes that achieve the capacity of the binary-input additive white Gaussian noise (AWGN) channel under iterative decoding were constructed by Chung in [34]. Since then, density evolution (DE) [60] has been used to optimize LDPC codes for a variety of memoryless channels (e.g., [102]), and the results suggest, for each channel, that sequences of iteratively decoded LDPC codes can indeed achieve the channel capacity. In fact, the discovery of a channel whose capacity cannot be approached by LDPC codes would be more surprising than a proof that iteratively decoded LDPC codes can achieve the capacity of any binary-input symmetric channel (BISC).

Like we have seemed, the idea of decoding a code transmitted over a channel with memory via iteration was first introduced by Douillard in the context of turbo codes and is known as *turbo equalization*. This approach can also be generalized to LDPC codes by constructing one large graph which represents the constraints of both the channel and the code. This idea was investigated for partial-response channels by Kurkoski, Siegel, and Wolf in [104].

Until recently, it was difficult to compare the performance of turbo equalization with channel capacity because the binary-input capacity of the channel was unknown. Recently, a new method has gained acceptance for estimating the achievable information rates of finite state channels (FSCs), and a number of authors have begun designing LDPC based coding schemes which approach the achievable information rates of these channels [103][105]. The main topics presented in [103] are:

1. Concentration theorems for Gallager codes and the sum–product message-passing decoder over binary ISI channels;
2. A density evolution method for computing the thresholds of “zero-error” performance over these channels;

3. Theorems establishing that the asymptotic performance of Gallager codes using the sum–product algorithm is upper-bounded by the symmetric information rate and the i.i.d. capacity;
4. And the computation of the BCJR-once bound, which is the limit of "zero-error" performance of the sum–product algorithms if the trellis portion of the algorithm is executed only once.

As is the case with DE for general BISCs, the evaluation of code thresholds and the optimization of these thresholds are done numerically. For FSCs, the analysis of this system is quite complex because the BCJR algorithm is used to decode the channel.

Since the capacity of a channel with memory is generally not achievable via equiprobable signaling, one can instead aim for the symmetric information rate (SIR) of the channel. The SIR is defined as the maximum information rate achievable via random coding with equiprobable input symbols. Since linear codes use all inputs equiprobably, the SIR is also the maximum rate directly achievable with linear codes.

In this chapter, we focus on developing LDPC codes for channels with non binary inputs and ISI memory. We are concerned with finding LDPC-turbo equalization which produces the lowest possible bit-error rate for a minimum amount of complexity.

9.1 The OS Technique with LDPC Codes

9.1.1 Bit or Symbol LDPC Coded Modulation Transmitter

Bit (resp. Symbol) LDPC Coded Modulation BLCM (resp. SLCM) can be modeled as a serial concatenation of a binary (resp. non binary) LDPC encoder, and a 16-QAM modulator as shown in Fig. 9.1.

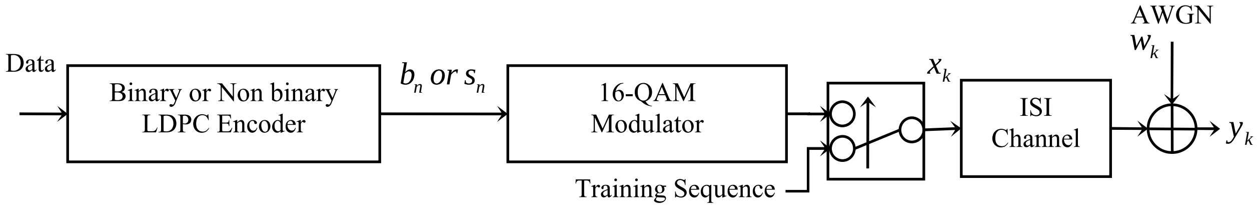


Figure 9.1: Bit/Symbol LDPC Coded Modulation Transmitter with ISI channel.

For error correction we use binary or non binary LDPC codes [3] whose code graph chosen uniformly at random from the ensemble of regular graphs. The ISI coefficients are assumed to be real.

At the transmitter, we will consider three cases:

1. Bit Turbo Coded Interleaved Modulation (BTCIM) where the information bits are encoded by a Turbo encoder (ζ) before being bitwise interleaved. Next, *4* consecutive bits b_n of the interleaved coded sequence are grouped to form a channel symbol and a 16-QAM modulator maps this channel symbol to a complex transmitted symbol x_k.

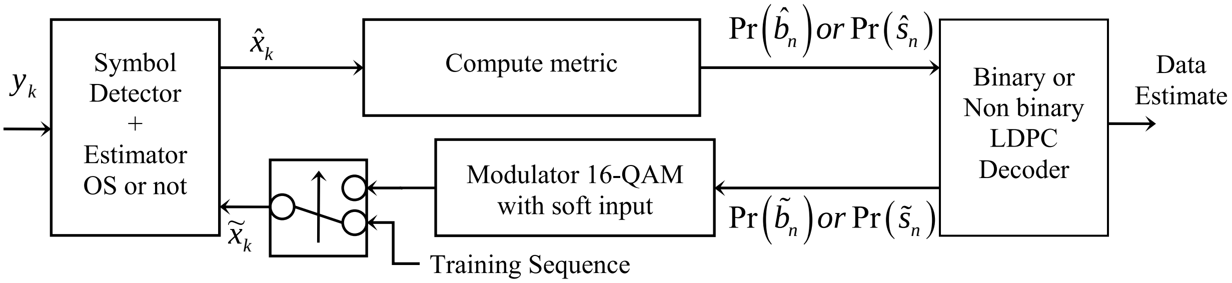
2. Bit LDPC Coded Modulation (BLCM) where the information bits are encoded by a binary LDPC encoder (ζ), here the bitwise interleaver is in the structure of the graph of binary LDPC codes. Next, we group and maps like in the case 1.
3. Symbol LDPC Coded Modulation (SLCM) where the information bits are first grouped by *4* to form a channel symbol which will be encoded by a non binary LDPC encoder (ζ) by consequence a GF(16) LDPC encoder, here the interleaver is symbolwise and included in the structure of the graph of non binary LDPC codes. Next, a 16-QAM modulator maps the coded symbol to a complex transmitted symbol x_k.

We assume an ISI channel, the received discrete-time base band signal can be written as eq. (5.5)

$$y_k = \sum_{i=0}^{l-1} h_i x_{k-i} + w_k, \qquad \forall k \in [1, N+l-1] \tag{9.1}$$

Where x_k are the complex coded symbols in a block with $k=1,\ldots, N$, ($x_k=0$ for $k>N$), N is the block length, w_k are white Gaussian noise samples. The l complex taps h_i represent the equivalent discrete overall channel impulse response. We consider that we have N_t training sequence symbols located.

9.1.2 Joint Iterative LDPC Decoding and detection with OS technique



y_k
Symbol Detector + Estimator OS or not
$\hat{x}_k$
Compute metric
$\Pr(\hat{b}_n) or \Pr(\hat{s}_n)$
Binary or Non binary LDPC Decoder
Data Estimate
Modulator 16-QAM with soft input
$\Pr(\tilde{b}_n) or \Pr(\tilde{s}_n)$
$\tilde{x}_k$
Training Sequence

Figure 9.2: Turbo Equalization with LDPC Decoder and OS Technique.

Joint equalization and decoding schemes are described for Inter Symbol Interference (ISI) channels in chapter 4. Equalization is performed using Interference Cancellation (IC) criterion and estimation uses the OS technique explained in chapter 5. Binary and non binary LDPC decoders are represented in details in chapter 7 and 8.

In the system represented in Fig. 9.2, we consider a fairly standard for the joint iterative decoding of an LDPC code and a channel ISI with memory. The turbo equalizer consists of an IC symbol detector and a binary or non binary LDPC decoder. The first half of decoding iteration entails running the symbol detector on using the feedback symbols from the LDPC decoder returned in a hard or soft form. The second half of decoding iteration corresponds to executing LDPC iterations using internal edge messages from the previous iteration and the observation of the symbol detector output.

In other words, the equalizer sends the symbol detected to compute metric block which compute the observation on each bit (in the case 1 and 2) and on each symbol (in the third case). Then these observations are sent to the turbo decoder (case 1), to the binary LDPC decoder (case

2), or to the non binary LDPC decoder (case 3) which uses it as *a priori* information and performs a fixed number of sum-product message-passing iterations before passing its extrinsic information to the 16-QAM modulator. This modulator calculates a hard 16-QAM symbol from a soft input and passing them to equalizer. This process is continued until the receiver converges or a maximum number of iterations is exhausted.

9.2 The Effect of the Distribution Degree

9.2.1 Description of simulation experiments

Having constructed a non binary parity check matrix, we test its performance by numerical simulation. We simulate a non variant two taps ISI channel and turbo equalization with OS technique as described in chapter 5, and we examine the success of detection and decoding several million blocks. Each block of 250 *q*-ary symbols is transmitted per channel use, and the likelihood of each corresponding noise symbol depends on equalization outputs.

The performance is gauged by plotting the bit error rate against the signal to noise ratio. In all the experiments presented in this chapter, we allow the non binary LDPC decoding algorithm to run for a maximum of 100 iterations divided into 4 turbo equalization iterations, in each iteration of turbo equalization we perform 25 iterations of non binary LDPC decoding algorithm before announcing a decoding failure. The block length of coding is about 1000 *q*-ary symbols which are transmitted into 4 channel blocks.

In our initial investigations of non binary LDPC codes, we constructed parity check matrices defined over field *GF(*16*)* with a fixed column and row symbol weight to obtain a coding rate 1/3 to compare these to simulation results with convolutional code presented in chapter 5 and simulation results with UMTS turbo code of rate 1/3, the fixed row symbol weight don't exceed 6 to not increase the complexity of non binary decoding algorithm.

9.2.2 Surprising and confusing results

The results shown in Fig. 9.3 are, at first sight, rather strange. It show results for codes with a fixed column weight of 4 or 2 despite that a binary LDPC code with a fixed column weight of 2 is the worst LDPC code, and a fixed row weight of 6 or 3 defined over fields *GF(*16*)* for rate 1/3. We see that LDPC codes in the binary case or the non binary case with row weight 6 and column weight 4 are far above 2.5 dB at BER=10^{-4} from the turbo code performance, despite that the non binary LDPC code improves the performance in relation to binary LDPC code by up to 0.5 dB at BER=10^{-4}. In comparison with convolutional code used in chapter 5 and 6, the LDPC codes improve the performance in high Signal to Noise Ratio (SNR) but it remains unsatisfied.

In the case of LDPC codes with row weight 4 and column weight 2, it is clear that the binary LDPC code is the worst code, but a little hope appears with the non binary LDPC code because it presents very good performances where we obtain a gain up to 0.5 dB at BER=10^{-4}, until SNR=3 dB where the slope changes its tangent and continues its decline but very slowly, until it joins the slope of binary LDPC code performances.

What is going on? To answer this we must examine the effect of the distribution degree (column and/or row weight) on our system model.

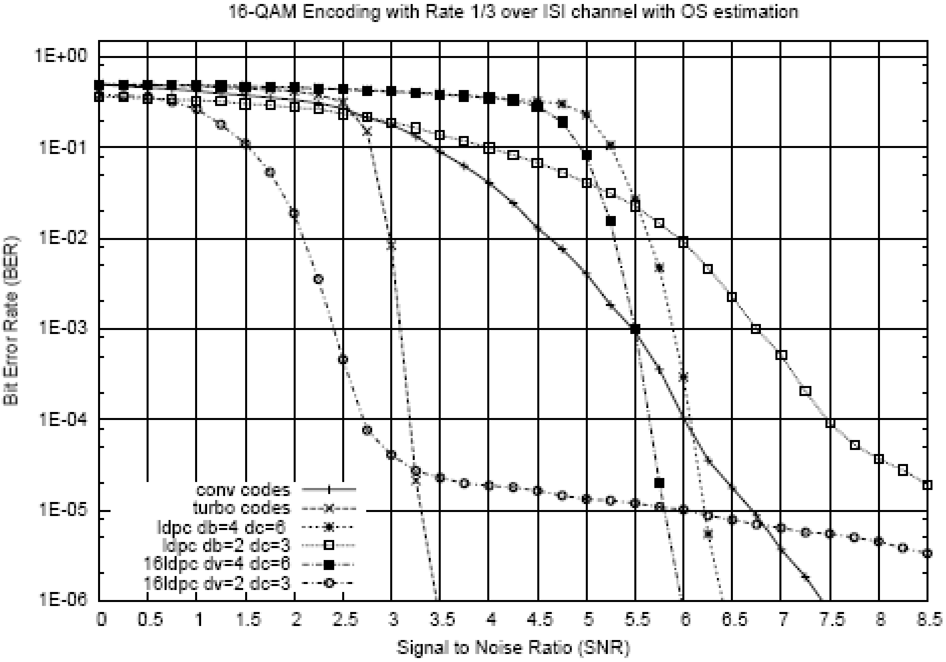


Figure 9.3: Surprising and confusing results on LDPC codes with OS Estimation.

9.2.3 Finding better LDPC codes over *GF(*16*)*

We present here, the problem of good choice of column and row weight. We can fix the row weights which imposes one constraint on the possible column weights. This can define the space in which valid column profiles λ are found. *We define a function that assigns a performance scores to each possible column profile and then extremes this function.* Although the method is simple, its helps us find parameters for excellent non binary and binary codes.

$$\begin{gathered}\sum_{i=1}^{d_b}\lambda_i=1,\quad 0\le\lambda_i\le1,\quad\forall i\\ \sum_{i=1}^{d_c}\rho_i=1,\quad 0\le\rho_i\le1,\quad\forall i\end{gathered}\tag{9.2}$$

Given a row profile, the column profile is further constrained by the rate R of the code:

$$\sum_{i=1}^{d_b}i\lambda_i=(1-R)\sum_{j=1}^{d_c}j\rho_j\tag{9.3}$$

Eqs. (9.3) simply reflects the fact that the number of non-zero entries in the columns must sum to the number of non-zero entries in the rows.

We require a function to assess the quality of a given column or row profile and then must minimize this function over $(d_b-2)or(d_c-2)$ -dimensional manifold. We have tried two numerical functions: one based on empirical decoding trials, the other based on Monte Carlo simulation of decoding. We found the empirical function to be the more accurate of the two, given sufficient iterations. The Monte Carlo simulations suffered from the fact that finite block length effects seem to be more pronounced with the addition of high weight columns or row. Thus, the agreement between Monte Carlo simulations and finite codes was not as close as in the case of regular codes.

Empirical function

In this approach it first constructs a code with the desired parameters. It uses the average number of iterations required for decoding to estimate the usefulness of the code. The number of iterations required varies from block to block, so our estimate of the function is necessarily fairly noisy.

For the average iteration count to be meaningful, it is necessary that all blocks are decoded. For this reason we use a variant of the decoding algorithm in which we artificially increase the signal to noise level as decoding progresses. This ensures that all blocks can eventually be decoded. We keep the noise vector and noise amplitude for each bit fixed, but increases the signal amplitude after each iteration.

Monte Carlo function

We can also score parameters using Monte Carlo methods to simulate an infinite code and calculating the average bit entropy of the tentative decoding after a fixed number of iterations of the decoding algorithm.

Care is required when simulating irregular codes. When we choose a noise node our column profile tells us with what probability that node has *i* neighbors. However, when we add a check node, the degree of that node must be chosen according to the probability that an edge chosen at random is connected to a check node of a given degree. This is not the same quantity as that given by the row profile, just as the fraction of passengers who find themselves on crowded buses is greater than the fraction of buses that are crowded.

In finite irregular codes the presence of high weight columns introduces many cycles of relatively short length and consequently the Monte Carlo simulations agree less well with the empirical results than for regular constructions.

Finding good codes

Recent work by Richardson, Shokrollahi and Urbanke offers an analytical approach to the design of LDPC codes based on density evolution. For given code parameters, the method allows the expected fraction of incorrectly determined nodes to be calculated as a function of the iteration count.

Although the method assumes a loop-free graph, they proved that, with probability that approaches 1 exponentially fast in the block length, the decoder will not deviate from loop-free behavior by more than ε. For a given row/column profile, the supremum over all noise levels for which the fraction of incorrectly determined nodes approaches zero is the *"cutoff noise level"* for that profile. This can be used as a function for empirical searches for good profiles.

The problem of minimizing a general n-dimensional non linear function is hard. Several methods exist that are guaranteed to find a local minimum and simulated annealing may be used to help in escaping shallow minima. Our problem is made more acute because not only do we lack any gradient information, but also our evaluations of the function itself are noisy, producing many false local minima.

We use the relatively straightforward downhill simplex method. We start by choosing $(d_b - 1)$ points in the $(d_b - 2)$-dimensional manifold in which we want to search. These define a simplex. We ensure that the simplex encloses a non-zero volume by finding a basis for our manifold and placing the vertices in the basis vector directions relative to an interior point.

We evaluate the function at each vertex, and move the vertex with the worst score. Usually we reflect it through the centre of the opposite face which means the average of other vertices, stretching whenever possible to increase the step size. If this results in a more worst we instead shrink the point towards the opposite face. If this also fails we shrink all vertices towards the best point. Repeating these steps we converge to a local minimum of the function. To combat the effect of noise in the estimation of the function, a new search can be started from the final point of the previous search.

9.3 Simulations Results

We conclude from section 9.2.2 that the column weight d_b should be between 2 and 4. We will try to find the best LDPC code included in this interval. In previous we will call fixed column weight when the column weight is fix, and non-integer mean column weight when the column weight is not constant for all column but keeping the column weight as uniform as possible. For example, a matrix of fixed column weight 3 would have λ=(0, 0, 1) whereas a matrix of mean column weight 2.5 might have λ=(0, 0.5, 0.5). Similarly, we define the fixed row weight and the non-integer mean row weight.

To check the effectiveness of our system, the use of non binary LDPC codes over ISI channel with OS estimation, we proceed in two times.

9.3.1 LDPC codes with fixed column weight

We leave the column weight fix between 2 and 4 which means that d_b=3, and we try to find a non-integer mean row weight without changing the rate of the code. We will use a row weight with mean equal to 4.5.

These can be compared to the results find with turbo codes and convolutional codes. Fig. 9.4 shows that the performances of LDPC codes, in binary case as in non binary case, approach the performance of turbo code and the loss is decreased to 2 dB for the binary case and to 0.75 dB for the non binary case at BER=10^{-4}, which mean that the gain obtained by using the non binary LDPC codes in relation to binary LDPC codes increase to 1.25 dB at BER=10^{-4}. The performances of LDPC codes are now satisfied in comparison with convolutional codes.

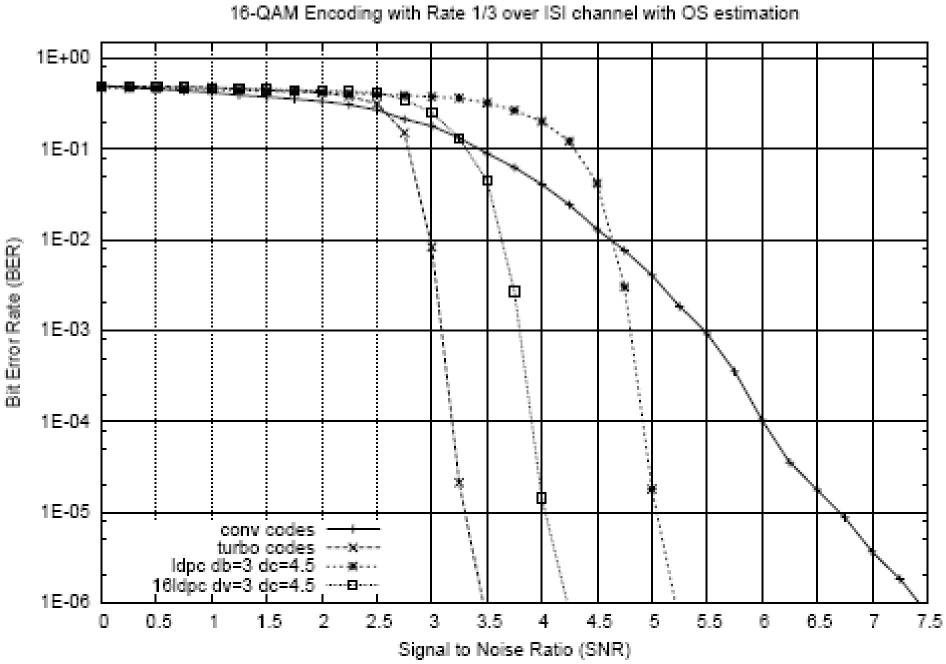


Figure 9.4: The performances of LDPC codes with fixed column weight.

9.3.2 LDPC codes with non-integer column and row weight

We find that the use of non-integer mean row weight improve the performances of LDPC codes in our proposed system, we will try to perform a completely irregular LDPC codes with non-integer column and row weight, with keeping the column and row weight as uniform as possible.

We present results for rate 1/3 codes. The best code we have found has a mean column weight of 2.4 and a mean row weight of 3.6. The column and row weight are almost uniform. The column profile is λ=(0, 0.6, 0.4) and row profile is ρ=(0, 0, 0.7, 0, 0.3).

Fig. 9.5 shows the performance of the best non binary LDPC codes of rate 1/3 found for system over ISI channel with OS estimation. We compare this irregular non binary LDPC code with turbo code of rate 1/3 for the same system, and we find that we obtain a gain of 0.5 dB at BER=10^{-4} in relation to turbo code performance which represents already a gain of 2.75 dB in relation to the convolutional codes, despite the binary LDPC code with the same column and row profile is not as good as possible.

To gain intuition into these results, consider again the graph based view of the decoding algorithm. Ideally a variable node should be connected to many check nodes, to gain most information about its state. Conversely, a check node prefers few parents, so that it can provide more confident estimates of each parent's state.

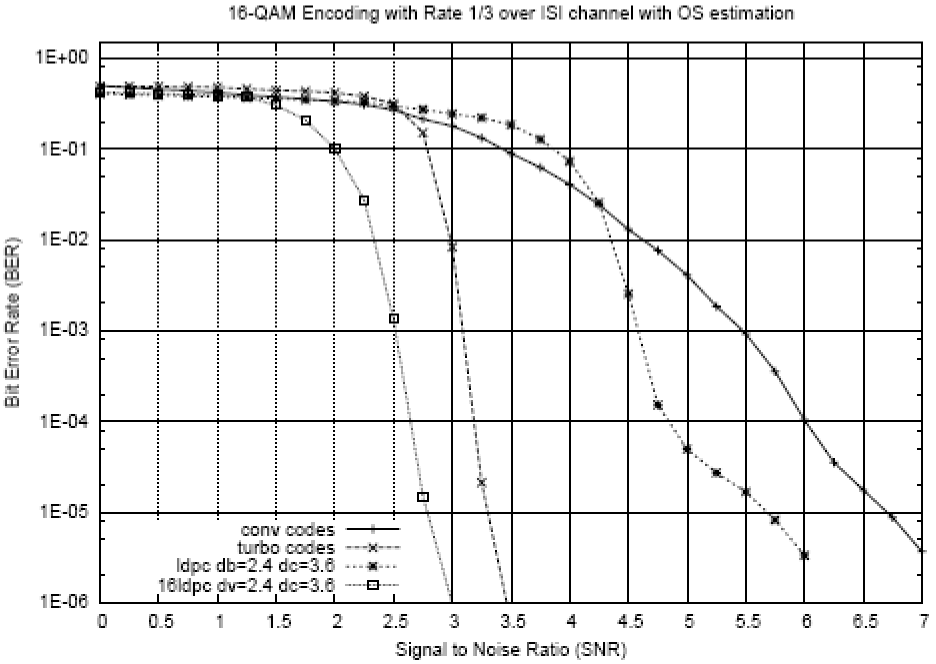


Figure 9.5: The performance of LDPC codes with non-integer column and row weight.

9.4 Conclusion

In this Chapter, the observation separation technique on the joint iterative channel estimation and symbol detection technique has been considered for a non binary LDPC coded 16-QAM systems. We have considered Bit Turbo Coded Interleaved Modulation, Bit LDPC Coded Modulation, and Symbol LDPC Coded Modulation at the transmitter. At the receiver, we considered the turbo equalization technique with OS estimation in these three cases. And we compare their results. The performance obtained is very attractive.

Our system proposed is realized by associating the non binary (16-ary) LDPC encoding at the transmitter and the OS technique with non binary LDPC decoding. We have showed that a good choice of distribution degree of non binary LDPC codes leads us to outperform the performances of turbo codes; this despite the performance of the binary LDPC codes with the same distribution degree is not so good. The drawback is the complexity, but we have presented in chapter 8 a decoding rule to reduce the complexity and make faster the algorithm.

Depending on the scenario considered, simulation results have shown up to a gain of 0.5 dB for the non binary LDPC codes with OS technique compared to the turbo codes with OS technique, and 3.25 dB in relation to convolutional codes with OS technique at BER=10^{-4}; this is very attractive performance.

Chapter 10
Conclusions and Perspectives

10.1 Contributions

In this thesis, we have proposed and analyzed low-complexity strategies to fight interference channel when channel state information is unavailable at the transmitter and estimated at the receiver. Furthermore, in the chapter 8 we have introduced the adaptive LDPC codes. These strategies are centered on the turbo equalization technique, which cancels channel interference from received signals using tentative decisions whose reliability improves with each successive iteration. Of the receiver algorithm considered in the past, turbo equalizer can be considered the most interesting one in terms of performance, in addition to which the equalizer converges with fewer iterations. But it requires initialization by another algorithm to reach good performance.

The convergence properties of the equalizer, which depend on the channel and the receiver SNR, together with the utilized channel code, determine the receiver performance. We have demonstrated that the turbo equalization works particularly well when used in conjunction with the OS technique in the estimation. To achieve reliable communication at rates up to the mutual information of an interference channel, channel coding is required, and a suitable code design can optimize the transmission and improve the receiver performance if the channel statistics are known or not thanks to OS estimation.

The observation separation technique on the joint iterative channel estimation and symbol detection technique has been considered for the CDMA and the narrowband mobile systems. The OS concept corresponds to decouple the observation used for the estimator from the observation used for the detector. The performance obtained is very attractive. In this thesis, our OS is realized by associating an adaptive IC (or PIC when MAI is presented) structure for the symbol detection with a modified pseudo-inverse channel estimation structure for each symbol. Nevertheless this realization scheme on the OS concept is not unique and research on a better solution is open. We have applied this technique to the 12.2 service of the UMTS-TDD uplink system, to the 12.2 service of the associated narrowband system and also to a high level modulation M-QAM systems.

Next, we have analyzed and evaluated a novel approach to the design of BICM-ID for ISI channels. By recognizing that the coding and modulation are isolated by the bit interleaver, and identifying the impact of iterative decoding on the harmonic mean of the minimum Euclidean distance, we have developed a powerful turbo equalization system yet relatively complex. We prove that the harmonic mean of the minimum Euclidean distance identified as crucial for BICM, can be greatly increased with BICM-ID and the error free feedback assumption. Then a new labeling map optimized under the harmonic mean criterion for 16-QAM is presented. We remark that even when the mapping has the best asymptotic performances, it is not necessary that this mapping will have the best performance in turbo equalization; some mapping is not adapted to turbo equalization like Anti Gray.

Then we have tried to improve coding and decoding schemes and we have focused on receiver systems employing LDPC codes. The performance of the receiver is greatly improved, if the estimates of channel transmission coefficients and the noise variances are available at the receiver. However for good performance of the receiver we require some iteration on the turbo equalization and on the LDPC decoder.

After, we have presented a symbol-by-symbol decoding rule for non binary LDPC codes. The non binary LDPC encoding and decoding outperforms the binary LDPC equivalent especially when the modulation is not binary, and we prove that even when the turbo code outperforms the binary LDPC, the non binary LDPC can recover this loss and go beyond. Considering the complexity, the decoding rule is practical for high rate, applicable to non binary LDPC codes and reduces enormously the complexity by using the duality properties and accelerates the computation by using the FHT. The complexity of the decoding rule for an (n, k) linear code is comparable to the complexity of a Viterbi decoder for the (n, n-k) dual code.

Finally, the observation separation technique on the joint iterative channel estimation and symbol detection technique has been considered for a non binary LDPC coded 16-QAM systems. We have compared it to Bit Turbo Coded Interleaved Modulation, Bit LDPC Coded Modulation. At the receiver, we considered the turbo equalization technique with OS estimation in these three cases. Our proposed system is realized by associating the non binary (16-ary) LDPC encoding at the transmitter and the OS technique with non binary LDPC decoding. We have showed that a good choice of distribution degree of non binary LDPC codes leads us to outperform the performances of turbo codes; this despite the performance of the binary LDPC codes with the same distribution degree is not so good. The drawback is the complexity, but we have presented in chapter 8 a decoding rule to reduce the complexity and make faster the algorithm.

We discussed in this thesis:

- In chapter 5, we have introduced the OS technique which partially separates the observation used for channel estimation from the observation that allows the symbol detection. We apply it to the turbo equalization receivers on a time-invariant ISI channels.
- In chapter 6, we suggest a design of optimal mapping in BICM-ID scheme to OS channel estimation and IC procedure system. We design an optimized binary mapping of the QAM constellation, in order to increase coding gain, which assumes perfect a priori information**.**
- We have discussed the essential properties of LDPC codes in Chapter 7. In Chapter 8, we have derived the non binary LDPC codes, adaptive LDPC codes, and the potential integration with orthogonal frequency division multiplexing (OFDM). A based system with adaptive LDPC codes presented is also another option.
- In chapter 9, we have applied the knowledge gained in Chapters 7 and 8 to the code design and optimization of non binary LDPC codes for the turbo equalization with OS estimation over ISI channel.

10.2 Future Works

Although in earlier chapters we have touched upon possibilities for more research, we now discuss some of the potentially more fruitful research directions inspired by this thesis. There are many interesting open questions that may attract further research.

The problem of joint detection and decoding warrants further study. At least, we have to try to reduce the complexity so that the turbo equalization been more simply implemented.

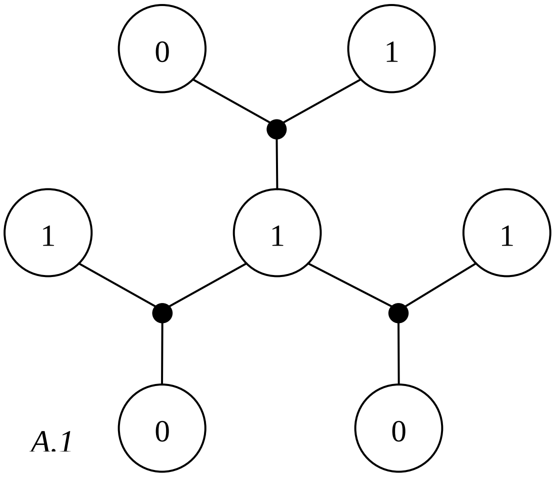
We have to do a convergence analysis by using the EXIT chart or any another method, and it is unclear whether the optimal preamble placement to minimize the error in the channel estimation which help us to accelerate the convergence. This needs to be analyzed further.

For the LDPC codes, there is much work to be done in generalizing the DE algorithm to more general modulated channels in a non binary system, and perhaps to apply the adaptive LDPC codes in a turbo equalization system over an ISI channel.

Finally, all the work in this thesis can be simply extended to MIMO systems which are taken an interest in the future wireless generation norm.

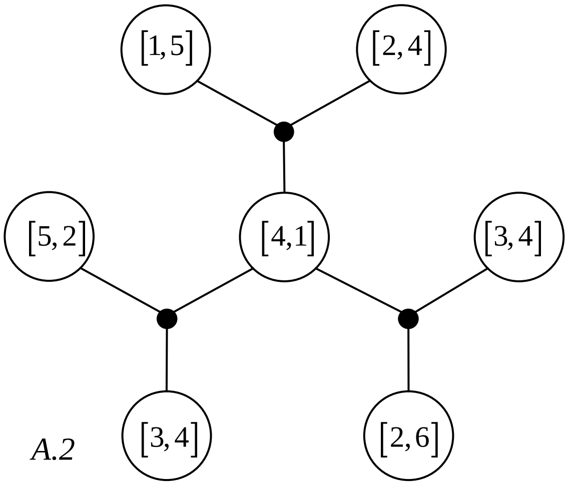
Appendix A
Min-Sum Algorithm Apply to linear (7, 4, 2) code

The min-sum algorithm applied to a binary linear (7, 4, 2) code, whose Tanner graph is shown in A.1. The Likelihood observations are indicated in A.2. A.3–A.7 illustrate the computation of the intermediate messages and final results for a few of the variables. In A.8, the final results of all variables are shown.



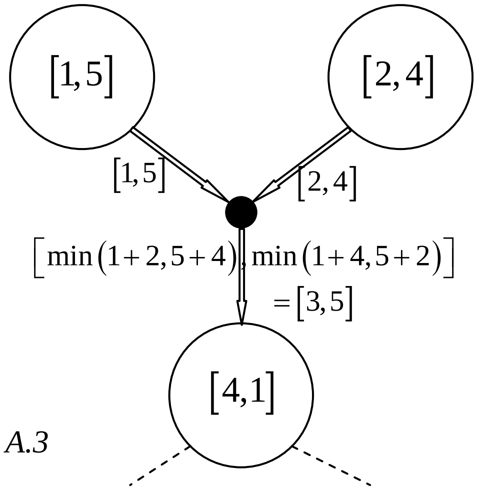
A.1

The Tanner graph and a codeword. The circles (sites) correspond to the codeword components and the small dots (checks) to the parity-check equations, i.e., the three variables connected to any check are required to have even



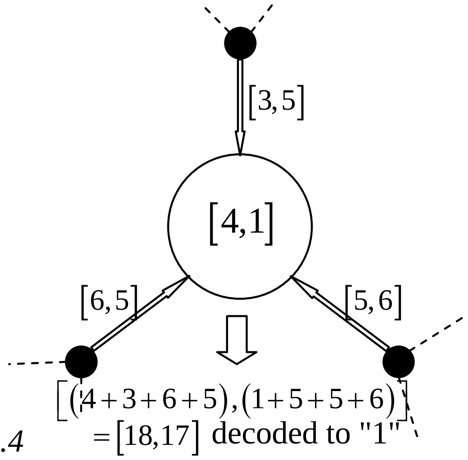
A.2

The channel output after transmitting a random codeword. The numbers in the variables are the **local costs** (log-likelihoods) for assigning “0” or “1” to that site. The decoding problem is to find a codeword with the smallest **global** cost, defined as the sum of the local costs in the variables.



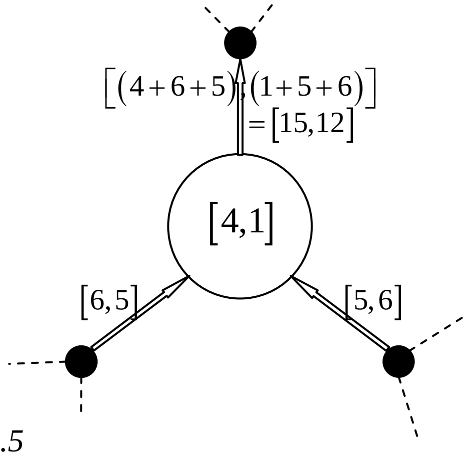
A.3

The check-to-variable functions from the upper check to the middle site. For each possible value in the middle variable, the check finds the smallest possible cost contribution from the two topmost variables. e.g., for a “0” in the middle variables, the patterns “00” and “11” are examined.



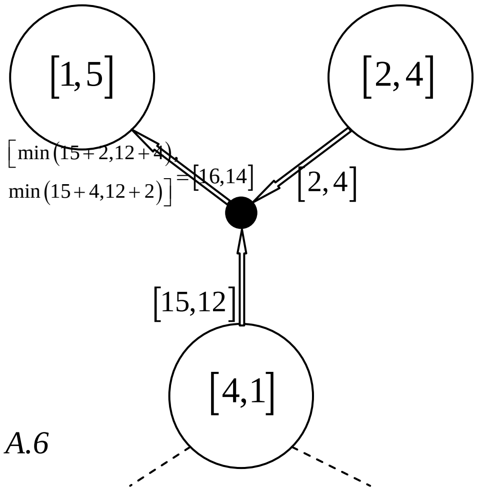
A.4

Final decision of the middle variable. (The two lower checks have computed their contributions to the middle variable in the same way as was done in c.) The global cost of “0” and “1” in the middle variable is then just the sum of the local costs and the three incoming cost contributions.



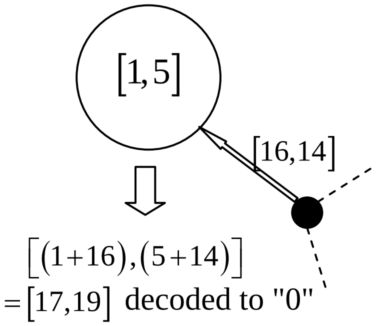
A.5

The variable-to-check functions from the middle variable to the upper check is the smallest possible cost in the lower five variables that results from assigning “0” or “1” to the middle variable; it is computed by adding the middle variable’s local costs to the sum of the contributions from the two lower checks.



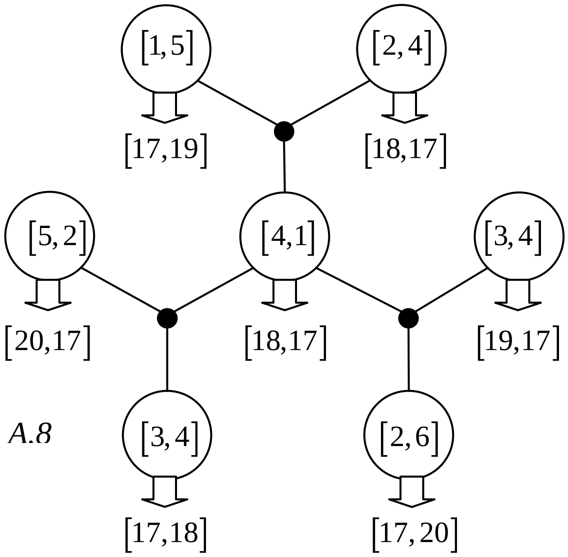
A.6

The top-left variable receives the smallest cost contributions from the rest of the graph that results from assigning “0” or“1”to that variable.



A.7

The final function of the upper-left variable is the sum of its local costs and the cost contributions from its only check. The resulting costs are the smallest possible global costs that result from assigning a “0” or “1” to the upper-left variable.



A.8

The rest of the Tanner graph is processed in the same way, resulting in final costs as shown. The resulting optimal codeword turns out to be the one shown in a).

Appendix B
Capacity Evaluation with Waterfilling

It is assumed that the channel input is real and Gaussian distributed. The channel total bandwidth 1/*T* is decomposed into *N* equally spaced subbands of width 1/*T*$_i$,

$$T_i = \frac{T}{N} \tag{B.1}$$

Where, *T* is the symbol period in seconds, T_i is the symbol period for the *i*th subband, *N* is the number of subbands and *i*, *i*=1…*N* is the subband index.

The baseband power and the bandpass symbol energy are related by $P = 2\,E_s / T$. The above relation is also valid for a subband, i.e. $P_i = 2E_{si}/T_i$. The power conservation gives

$$P = \sum_{i=1}^{N} P_i \qquad \Rightarrow \qquad \sum_{i=1}^{N} E_{si} = N \times E_s \tag{B.2}$$

Where, *P* is the average transmit total power, P_i is the transmit power in the *i*th subband, E_s is the average total energy per signal and E_{si} is the energy per symbol in the *i*th subband.

The noise power in a subband is written as

$$N_i = N_0 \times \left(1/T_i\right) \times \left(1\big/\left|H\left(i\right)\right|^2\right) \tag{B.3}$$

Where, N_i is the noise power in the *i*th subband, N_0 is the noise power spectral density, $H\left(i\right)$ is the amplitude channel response in the *i*th subband and $\left|H\left(i\right)\right|^2$ is the power channel response in the *i*th subband.

The above relation comes from a product of the bandwidth by the power spectral density. The noise power is divided by the channel power response since it is equivalent to multiplying the transmit energy E_{si} by $\left|H\left(i\right)\right|^2$. In other words, the channel with attenuation $\left|H\left(i\right)\right|$ and noise power N_0/T_i is replaced by an equivalent channel with attenuation 1 and noise power N_i.

Before writing the capacity formula, let us note that

$$\frac{P_i}{N_i} = \frac{2E_{si}/T_i}{\left(N_0/T_i\right)\big/\left|H\left(i\right)\right|^2} = \frac{2E_{si}}{N_0} \times \left|H\left(i\right)\right|^2 \tag{B.4}$$

The partial capacity associated to the *i*th subband is

$$C_i = \frac{1}{2}\log_2\left(1+\frac{P_i}{N_i}\right) = \frac{1}{2}\log_2\left(1+\frac{2E_{si}}{N_0}\times\left|H(i)\right|^2\right) \tag{B.5}$$

The total capacity in bits per dimension, while keeping in mind that *T* is constant, is the average of the partial capacities,

$$C = \frac{\sum_{i=1}^{N} C_i T_i}{T} = \frac{\sum_{i=1}^{N} C_i}{N} \tag{B.6}$$

Once the capacity is computed, the energy per bit is derived by $E_b = E_s/C$. Thus, it is easy to make E_s/N_0 vary from -50 dB to 20 dB, compute the capacity *C* (in bits/dim) for each SNR value and then derive E_b/N_0.

Now, let us maximize the total capacity under the constraint that E_s has a fixed valued. Define $F\left(\left\{E_{si}\right\}\right) = C + \lambda E_s$. We get

$$F\left(\left\{E_{si}\right\}\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log_2\left(1+\frac{2E_{si}}{N_0}\times\left|H(i)\right|^2\right) + \lambda\left(\sum_{i=1}^{N}E_{si}\right)$$

Derive with respect to E_{si},

$$\frac{dF}{dE_{si}} = \frac{1}{2N}\times\frac{1}{\log(2)}\times\frac{2\left|H(i)\right|^2/N_0}{1+2E_{si}\left|H(i)\right|^2/N_0} + \lambda = 0$$

From the above equality, we find that

$$\frac{2E_{si}}{N_0} + \frac{1}{\left|H(i)\right|^2} = \theta = water\,level$$

which is the same as

$$\frac{2E_{si}}{N_0} = \left(\theta - \frac{1}{\left|H(i)\right|^2}\right)^+$$

By using the relation $\sum_{i=1}^{N} E_{si} = NE_s$, we find the important equation giving the value of the optimal water level,

$$\sum_{i=1}^{N}\left(\theta - \frac{1}{\left|H(i)\right|^2}\right)^+ = 2N\times\frac{E_s}{N_0}$$

The optimal capacity maximized by this strategy is

$$C = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log_2\left(\theta\left|H(i)\right|^2\right).$$

Appendix C
Weight Distribution of a Block Code

Performances of an Error Corrector Code (ECC) over a soft or a hard output channel depend on the weight distribution. For example, the number A_d of codeword situated at a Hamming distance $d=d_{Hmin}$ from transmitted word assign directly the error probably. The error probability increases if the number of neighbours increases. Thus, asymptotically ($E_b/N_0 >> 1$), performances are stroked to the minimal distance and the number of neighbours. On the other hand, for medium and low Signal to Noise Ratio (SNR), code words situated beyond d_{Hmin} also affect the error probability.

For reflect the computing of performances on information bits at the input of the encoder, we need a relation between the weight of the output (i.e. weight of code words) and the one of the input. We will see that it is very difficult to obtain, except for small codes. Thus, in the case of block codes, computing the error probability per codeword is immediate but the error probability per information bit will be very approximate.

In this appendix, we will describe the polynomial weight enumerator of a block code. We present the MacWilliams identity which link the weight distribution of the code and the one of its dual code. We indicate finally, that distributions of Hamming distance around code words are identical when the block code is linear.

C.1 The polynomial weight enumerator

The Hamming weight indicates if the symbol is null or not. For precision, it is necessary to know the value of the symbol when it is no null (q-1 possible values). The two definitions below describe the polynomial Hamming weight enumerator and the polynomial complete weight enumerator.

Definition 1 *(Hamming Weight Enumerator)*

Suppose $\zeta(n, k, d)$ a linear code defined over $GF(q)$. We call polynomial Hamming weight enumerator, or polynomial enumerator the following:

$$A(x) = \sum_{i=0}^{n} A_i x^i$$

Where A_i is the number of code words with weight i. The sum of these coefficients is equal to the size of code, $\sum_{i=0}^{n} A_i = q^k$.

The power of variable *x* in *A(x)* indicates the Hamming weight in the code words (number of non null symbols) and the coefficient gives their number. To make *A(x)* homogeneous, it is possible to introduce another variable for the number of null symbols. The polynomial enumerator become homogeneous of *n* degree, $A\left(x_0, x_1\right) = \sum_{i=0}^{n} A_i x_0^{n-i} x_1^i$.

For example, the polynomial enumerator of the Hamming code (7, 4, 3) is written $A\left(x\right) = x^7 + 7x^4 + 7x^3 + 1$, where the coefficients are equals to $A_0 = 1$, $A_1 = A_2 = 0$, $A_3 = A_{dHmin} = 7$, $A_4 = 7$, $A_5 = A_6 = 0$, $A_7 = 1$. We remark the discontinuity and the symmetry of the distribution.

Definition 2 *(Complete Weight Enumerator)*

Suppose *ζ(n, k, d)* a linear code defined over *GF(q)*. We call complete polynomial weight enumerator:

$$A\left(x_0, x_1, ..., x_{q-1}\right) = \sum_i A\left(i\right) x_0^{i_0} x_1^{i_1} ... x_{q-1}^{i_{q-1}}$$

Where $A(i)$, $i = (i_0, i_1, ..., i_{q-1})$ is the number of code words having i_0 null symbols, i_1 symbols equal to 1, i_2 symbols equal to 2, etc … the sum of coefficients is equal to the size of code, $\sum_i A_i = q^k$. The sum of power is equal to the code length, $i_0 + i_1 + ... + i_{q-1} = n$.

The complete polynomial weight is a polynomial with *q* variables. The power i_j of variable x_j indicates the number of symbols equal to $j, j \in GF(q)$. The coefficient $A(i)$, $i = (i_0, ..., i_j, ..., i_{q-1})$ give the number of code words having the configuration of vector *i*. We remark that, the definition 2 describe a homogeneous polynomial of *n* degree.

The polynomial Hamming weight enumerator can be obtain from the polynomial complete weight enumerator by forcing $x_1 = x_2 = ... = x_{q-1} = x$, $x_0 = 1$ and replacing the sum of integer $i_0 + i_1 + ... + i_{q-1}$ by the only index *i*. When the code is binary, the complete polynomial weight can be written under the form $A\left(x_0, x_1\right) = \sum_i A\left(i_0, i_1\right) x_0^{i_0} x_1^{i_1} = \sum_i A_i x_0^{n-i} x_1^i$. It is equivalent to the polynomial Hamming weight enumerator. Thus, the complete weight enumeration is interesting only in the case of non binary codes ($q > 2$)

Definition 3 *(Input Redundancy Weight Enumerator)*

Suppose *ζ(n, k)* a binary linear code. We call polynomial input redundancy weight enumerator:

$$A\left(x, y\right) = \sum_{i,j} A\left(i, j\right) x^i y^j$$

Where *A(i,j)* is the number of code words having the weight of information bits *i* and the weight of parity bits *j*. The sum of these coefficient is equal to the size of code, $\sum_{i,j} A\left(i, j\right) = 2^k$.

Replacing *y* by *x* and the sum *i + j* by *i,* we obtain the polynomial weight enumerator *A(x)* from *A(x, y)*.

It is very difficult to compute the polynomial input redundancy weight enumerator for any code. The only method available is to make the list of detailed weight of the code.

On other hand, computing the complete polynomial enumerator is sometimes facilitating by the application of MacWilliams identity describe below.

Definition 4 *(Golay Code)*

The Golay code (23, 12, 7) is a cyclic binary linear code defined by the generator polynomial:

$$g(x) = x^{11} + x^{10} + x^{6} + x^{5} + x^{4} + x^{2} + 1$$

The Golay code (23, 12, 7) is a perfect code, i.e. the Hamming born verify $1 + C_{23}^{1} + C_{23}^{2} + C_{23}^{3} = 2^{11}$.

Definition 5 *(Extensive Golay Code)*

The extensive Golay code (24, 12, 8) is a non cyclic linear binary code construct by addition one parity bit to the Golay code (23, 12, 7):

$$b_{24} = \sum_{i=1}^{23} b_i \quad \left(sum\ in\ GF\left(2\right)\right)$$

This code is self dual, i.e. $\zeta^{\perp} = \zeta$.

Definition 6 *(Primitive Reed Solomon Code)*

The primitive Reed Solomon code is a cyclic linear code defined over the field of $GF(2^m)$ symbols, with length n = $2^m - 1$, and correction capacity t, generate by:

$$g(x) = (x - \alpha)(x - \alpha^{2})\ldots(x - \alpha^{2t})$$

Where α is a primitive element of $GF(2^m)$.

The Reed Solomon code is a MDS code because it verify $n - k = 2t$ or $d_{Hmin} = n - k + 1$.
The Hamming weight distribution of the primitive Reed Solomon code is given by:

$$A_i = nC_n^i \sum_{j=0}^{i-2t-1} (-1)^j C_{i-1}^j (n+1)^{i-2t-1-j}$$

C.2 MacWilliams Identity

Suppose $\zeta(n, k)$ a linear code defined over $GF(q)$ and $\zeta^{\perp}(n, n-k)$ its dual code. MacWilliams Identity allow to compute the complete polynomial enumerator $A(x_0, x_1, ..., x_{q-1})$ (resp. Hamming enumerator $A(x)$) of ζ in function of two polynomial $B(x_0, x_1, ..., x_{q-1})$ (resp. $B(x)$) of the dual code $\zeta^{\perp}$.

Theorem 1: Suppose $\zeta(n, k)$ a binary linear code. Then the code contains 2^k words of even weight, or it contains 2^{k-1} words of even weight and 2^{k-1} words of odd weight.

Proof: if all the code words have an even weight $Card(\zeta) = 2^k \Rightarrow$ The first part of the theorem is easy. Suppose that the code has at least one code word c of odd weight. Λ is the set of even weight code words and Ω is the set of odd weight code words, $\zeta = \Lambda \cup \Omega$ and $c \in \Omega$. $\forall x \in \Lambda + c$, $x \in \Omega$, then $\Lambda + c \subset \Omega$. But $\forall x \in \Omega,\ x + c \in \Lambda$ because $x + c$ has an even weight.

$x + c \in \Lambda \Rightarrow x \in \Lambda + c$. Then $\Omega \subset \Lambda + c$. We prove that $\Lambda + c = \Omega$. We deduce that $Card\left(\Lambda\right) =$ $Card\left(\Omega\right) = Card\left(\zeta\right)/2 = 2^{k-1}$.

□

The theorem 1 indicates that every binary linear code have only even weight code words, if not then exactly the half of the code words have an even weight and the other half have an odd weight.

Definition 7 *(Hadamard Matrix)*

The Hadamard Matrix of order *n* is a matrix **H** type *n* x *n* constitute from +1 and -1, and verifying $\mathbf{HH}^{\dagger} = n\mathbf{I}$.

The *n* lines (or the *n* columns) of a Hadamard matrix are orthogonal, i.e. the matrix is a rotation matrix. The *n* lines form *n* functions of *n* samples known under the name of Walsh functions.

Theorem 2: *(Construction of Hadamard matrix)*

(1) Sylvester Construction: if $\mathbf{H}_n$ is a Hadamard matrix of order *n,* then we can construct the Hadamard matrix of order *2n*:

$$\mathbf{H}_{2n} = \begin{bmatrix} \mathbf{H}_n & \mathbf{H}_n \\ \mathbf{H}_n & -\mathbf{H}_n \end{bmatrix}$$

(2) Suppose that $n = 2^m$. We note $u_1, u_2, \ldots, u_n$ the *n* elements of $GF(2)^m$ (all the binary words of size *m*). The binary element obtain by the scalar product of two binary words is defined by $u_i.u_j = \sum_{r=1}^{m} u_{ir}u_{jr} \in GF\left(2\right)$. Then, the matrix defined by $\mathbf{H} = \left[h_{ij}\right]$, where $h_{ij} = \left(-1\right)^{u_i.u_j}$ is a Hadamard matrix of *n* order.

Proof: *(1)* It is easy because $\mathbf{H}_{2n}$ satisfy the relation: $\mathbf{H}_{2\mathrm{n}}\mathbf{H}_{2\mathrm{n}}^{\dagger} = 2n\mathbf{I}$

(2) Computing the scalar product between two lines *i* and *k* of **H**.

$$h_i.h_k = \sum_{j=1}^{n} h_{ij}h_{kj} = \sum_{j=1}^{n} \left(-1\right)^{u_i.u_j} \left(-1\right)^{u_k.u_j} = \sum_{j=1}^{n} \left(-1\right)^{\left(u_i+u_k\right).u_j}$$

If $i = k$ the scalar product is equal to *n* because $u_i + u_i = 0$. If $i \neq k$ then there is a line *r* where $u_i + u_k = u_r$. Now if the weight of u_r is $w \leq m$, $u_r.u_j$ produce the weight of u_j limited by *w*. Thus, when *j* change from 1 to $n = 2^m$, we obtain 2^{m-w} time the distribution weight of the universal code $GF(2)^w$, where the half of words have even weight and the other have odd weight. The scalar product of these two lines h_i and h_k is null.

□

Definition 8 *(Hadamard Transformer)*

(1) Suppose $X = (x_1, x_2, \ldots, x_n)$ a vector $\in \mathbb{R}^n$. The Hadamard Transformer *(Walsh or Fourier)* of the real vector *X* is defined by:

$$Y = \mathbf{H}X$$

(2) Suppose *f* an application defined in $GF(2)^n \rightarrow \Bbbk$. Then, the Hadamard Transformer of the application *f''* defined by:

$$f''\left(u\right) = \sum_{v \in GF\left(2\right)^m} \left(-1\right)^{u.v} f\left(v\right), \qquad \forall u \in GF\left(2\right)^n$$

Lemma 1: Suppose $\zeta(n, k)$ a binary linear code. We consider the application f: $GF(2)^n \rightarrow \mathbb{k}$, then

$$\sum_{u\in\zeta^{\perp}} f(u) = \frac{1}{2^k}\sum_{u\in\zeta} f''(u)$$

Proof:

$$\sum_{u\in\zeta} f''(u) = \sum_{u\in\zeta}\sum_{v\in GF(2)^n}(-1)^{u.v} f(v) = \sum_{v\in GF(2)^n} f(v)\sum_{u\in\zeta}(-1)^{u.v}$$

If $v \in \zeta^{\perp}$ the sum is equal to $Card(\zeta) = 2^k$ because $u.v = 0$.
If $v \notin \zeta^{\perp}$ from **theorem 1**, $u.v$ take equally 0 and 1 values. In this case, the sum is null.
Those mean: $\sum_{u\in\zeta} f''(u) = 2^k \sum_{u\in\zeta^{\perp}} f(u)$

□

Theorem 3: *(MacWilliams Identity, Binary Code)*

Suppose $\zeta(n, k)$ a binary linear code and $\zeta^{\perp}$ the dual code. Then the polynomial weight enumerator $A(x)$ and $B(x)$ of ζ and $\zeta^{\perp}$:

$$B(x) = \frac{1}{2^k}(1+x)^n A\left(\frac{1-x}{1+x}\right)$$

Proof: we apply **Lemma 1** with $f(u) = x^{w(u)}$ where $w(u)$ is the Hamming weight of u. We obtain

$$f''(u) = \sum_{v\in GF(2)^n}(-1)^{u.v} x^{w(u)}$$

Where $u = (u_1, ..., u_n)$ and $v = (v_1, ..., v_n)$. Then

$$f''(u) = \sum_{v\in GF(2)^n}(-1)^{u_1v_1+...+u_nv_n}\prod_{i=1}^{n} x^{v_i} = \sum_{v_1=0}^{1}\sum_{v_2=0}^{1}...\sum_{v_n=0}^{1}\prod_{i=1}^{n}(-1)^{u_iv_i} x^{v_i}$$

We know that

$$\sum_{2^m}\prod_{m} a1_{i_1} a2_{i_2}...am_{i_m} = \prod_{m}\sum_{2}(aj_0 + aj_1)$$

Where the index $i_1, ..., i_m$ are binary, we can write

$$f''(u) = \prod_{i=1}^{n}\sum_{w=0}^{1}(-1)^{u_iw} x^{w}$$

If $u_i = 0$, the sum is equal to $1 + x$. If $u_i = 1$, the sum is equal to $1 - x$. Thus we obtain

$$f''(u) = (1+x)^{n-w(u)}(1-x)^{w(u)}$$

The equation of **Lemma 1** became

$$\sum_{u\in\zeta^{\perp}} x^{w(u)} = \frac{1}{2^k}\sum_{u\in\zeta}(1+x)^{n-w(u)}(1-x)^{w(u)}$$

□

We remind that $\forall\, \beta \in GF(q = p^m)$ it can be write under the form $\beta = \beta_0 + \beta_1\alpha + ... + \beta_{m-1}\alpha^{m-1}$ or $\beta = (\beta_0, \beta_1, ..., \beta_{m-1})$. The element α is the primitive element of

GF(q) and the coefficients $\beta_i \in GF(p)$, i.e. $0 \le \beta_i \le p-1$. Suppose w the complex number $w = e^{(2j\pi)/p}$, the p^{th} root of unity.

Definition 9 *(GF(q) characters)*

$\forall \beta = (\beta_0, \beta_1, ..., \beta_{m-1}) \in GF(q)$**,** we define the application $X_\beta : GF(q) \to \mathbb{C}$ given by:

$$X_\beta(\gamma) = w^{\beta_0\gamma_0 + ... + \beta_{m-1}\gamma_{m-1}}$$

Where $\gamma = (\gamma_0, ..., \gamma_{m-1}) \in GF(q)$.

The application X_β is named character of *GF(q)*.

The characters of *GF(q)* verify the below proprieties:

- $\forall \beta,\ \gamma \in GF(q),\ \ X_\beta(\gamma) = X_\gamma(\beta).$
- $\forall \beta, \gamma, \gamma' \in GF(q),\ \ X_\beta(\gamma + \gamma') = X_\beta(\gamma) X_\beta(\gamma').$
- $\forall \beta, \beta', \gamma \in GF(q),\ \ X_{\beta+\beta'}(\gamma) = X_\beta(\gamma) X_{\beta'}(\gamma).$

Lemma 2: $\forall \beta \in GF(q) - \{0\}$, we have

$$\sum_{\gamma \in GF(q)} X_\beta(\gamma) = 0$$

Proof: The sum is equal to

$$\sum_{\gamma \in GF(q)} w^{\beta_0\gamma_0 + ... + \beta_{m-1}\gamma_{m-1}} = \prod_{j=0}^{m-1} \left(\sum_{\gamma_j = 0}^{p-1} w^{\beta_j \gamma_j} \right)$$

Seeing that β is non null, there is r where $\beta_r \ne 0$. Then

$$\sum_{\gamma_r = 0}^{p-1} w^{\beta_r \gamma_r} = \sum_{k=0}^{p-1} w^k = \frac{1 - w^p}{1 - w} = 0$$

□

Lemma 3: We consider the Hadamard transformer f'' of the application f defined by *GF(q)*^{n}

$$f''(u) = \sum_{v \in GF(q)^n} X_u(v) f(v)$$

Suppose ζ*(n, k)* a linear code defined in *GF(q)*, then

$$\sum_{u \in \zeta^\perp} f(u) = \frac{1}{q^k} \sum_{u \in \zeta} f''(u)$$

Proof: the proof is identical with the one of **Lemma 1** and use the **Lemma 2**.

□

Appendix D
New Character in Non binary Case

Notations: $q = p^m$, p is a prime number. Usually *p=2;*

$\omega = e^{j\frac{2\pi}{p}}$ primitive complex pth root of unity;

Definition 1: A new characters $\chi_\beta(\gamma)$ and $\underline{\chi}_u(v)$:

$\forall\beta = (\beta_0, \beta_1, ..., \beta_{m-1}) \in GF(q)$ we define an application $\chi_\beta(\gamma)$:

$$GF(q) \to \mathbb{C} : \ \chi_\beta(\gamma) = \omega^{\tau(\beta.\gamma)}$$

Where $\gamma = (\gamma_0, ..., \gamma_{m-1}) \in GF(q)$ and $\tau(\beta) = \beta_0$.

$\forall u = (u_0, u_1, ..., u_{n-1}), v = (v_0, v_1, ..., v_{n-1}) \in GF(q)^n$ we define an application $\underline{\chi}_u(v)$:

$$GF(q)^n \to \mathbb{C} : \ \underline{\chi}_u(v) = \omega^{\tau(u.v)}$$

Where $\tau(u) = u_0$.

Properties:

τ : $\forall\beta, \beta'$ where $\beta = (\beta_0, \beta_1, ..., \beta_{m-1}) \in GF(q)$ and $\beta' = (\beta'_0, \beta'_1, ..., \beta'_{m-1}) \in GF(q)$; we have:

- $\tau(\beta + \beta') = (\beta + \beta')_0 = \beta_0 + \beta'_0 = \tau(\beta) + \tau(\beta')$
- $\tau(\beta.\beta') = (\beta.\beta')_0 = \beta_0.\beta'_0 = \tau(\beta).\tau(\beta')$

$\forall u = (u_0, u_1, ..., u_{n-1}), v = (v_0, v_1, ..., v_{n-1}) \in GF(q)^n$ where $u_0, ..., u_{n-1}, v_0, ..., v_{n-1} \in GF(q)$;

$$\left\{\begin{aligned} u.v &= \left(u_0\alpha^0 + u_1\alpha^1 + ... + u_{n-1}\alpha^{n-1}\right).\left(v_0\alpha^0 + v_1\alpha^1 + ... + v_{n-1}\alpha^{n-1}\right) \\ &= \left(\left(u_0.v_0 + u_1.v_{n-1} + u_2.v_{n-2} + ...\right)\alpha^0 + \left(u_0.v_1 + u_2.v_{n-1} + ...\right)\alpha^1 + ... + \left(u_0.v_{n-1} + ...\right)\alpha^{n-1}\right) \end{aligned}\right\}$$

We have:

- $\tau(u+v) = (u+v)_0 = (u_0 + v_0, u_1 + v_1, ..., u_{n-1} + v_{n-1})_0 = u_0 + v_0 = \tau(u) + \tau(v)$
- $\tau(u.v) = (u.v)_0 = ((u_0.v_0 + ...), (u_0.v_1 + ...), ..., (u_0.v_{n-1} + ...))_0$
  $= (u_0.v_0 + ...) \neq (u_0.v_0) = \tau(u).\tau(v)$

$\chi_\beta(\gamma)$: $\forall \beta, \beta', \gamma, \gamma' \in GF(q)$ we have:

- $\chi_\beta(\gamma) = \chi_\gamma(\beta)$ ***proof*** $\left\{ \chi_\beta(\gamma) = \omega^{\tau(\beta.\gamma)} = \omega^{\tau(\gamma.\beta)} = \chi_\gamma(\beta) \right\}$ □
- $\chi_\beta(\gamma + \gamma') = \chi_\beta(\gamma)\chi_\beta(\gamma')$

proof $\left\{ \chi_\beta(\gamma + \gamma') = \omega^{\tau(\beta.(\gamma+\gamma'))} = \omega^{\tau(\beta.\gamma+\beta.\gamma')} = \omega^{\tau(\beta.\gamma)}\omega^{\tau(\beta.\gamma')} = \chi_\beta(\gamma)\chi_\beta(\gamma') \right\}$ □

- $\chi_{(\beta+\beta')}(\gamma) = \chi_\beta(\gamma)\chi_{\beta'}(\gamma)$

proof $\left\{ \chi_{(\beta+\beta')}(\gamma) = \omega^{\tau((\beta+\beta').\gamma)} = \omega^{\tau(\beta.\gamma+\beta'.\gamma)} = \omega^{\tau(\beta.\gamma)}\omega^{\tau(\beta'.\gamma)} = \chi_\beta(\gamma)\chi_{\beta'}(\gamma) \right\}$ □

Lemma 1: $\forall \beta \in GF(q) - \{0\} \Rightarrow \sum_{\gamma \in GF(q)} \chi_\beta(\gamma) = 0$

Proof: $\sum_{\gamma \in GF(q)} \chi_\beta(\gamma) = \sum_{\gamma \in GF(q)} \omega^{\tau(\beta.\gamma)} = \sum_{k=0}^{q-1} \omega^k = \frac{1-\omega^q}{1-\omega} = 0.$

□

Lemma 2: Considering the Hadamard Transform f'' of an application f defined in GF(q)^n.

$$f''(u) = \sum_{v \in GF(q)^n} \underline{\chi}_u(v) f(v)$$

And ζ(n, k) a linear code defined in GF(q), then:

$$\sum_{u \in C^\perp} f(u) = \frac{1}{q^k} \sum_{u \in C} f''(u)$$

Proof: $\sum_{u \in C} f''(u) = \sum_{u \in C} \sum_{v \in GF(q)^n} \underline{\chi}_u(v) f(v) = \sum_{v \in GF(q)^n} f(v) \sum_{u \in C} \underline{\chi}_u(v)$.

If $v \in \zeta^\perp \Rightarrow u.v = 0 \Rightarrow \tau(u.v) = 0 \Rightarrow \omega^{\tau(u.v)} = 1 \Rightarrow \sum_{u \in C} \underline{\chi}_u(v) = q^k$

If $v \notin \zeta^\perp \Rightarrow u.v \neq 0 \Rightarrow \sum_{u \in C} \underline{\chi}_u(v) = 0$ using the **Lemma 1**.

Those mean: $\sum_{u \in C} f''(u) = q^k \sum_{u \in C^\perp} f(u)$

□

Appendix E
A Design Rules for Adapted Modulation

E.1 Characterization of the channel variation

Adaptive signaling employs (perhaps imperfect) knowledge at the transmitter of the current fading values to select a signal set from which to draw the current symbol. Consider the choice of the signal set for the i^{th} symbol. Recalling the assumption that analysis can assume the channel fading is constant over the duration of a single symbol, let $h_i = \left|h\left(iT_s\right)\right|$ be the amplitude of the fading that multiplies the i^{th} transmitted symbol. Assume temporarily that $R_h\left(\tau\right)$ is known at the transmitter, and let N be the number of outdated estimates employed. Define the N by N autocorrelation matrix $\Sigma_{\hat{h}}$ of the real or imaginary component of the vector of outdated channel estimates as

$$\Sigma_{\hat{h}} = E\left[\hat{h}_R \hat{h}_R^{\dagger}\right] = E\left[\hat{h}_I \hat{h}_I^{\dagger}\right] \tag{E.1}$$

and the length N column correlation vector ρ as

$$\rho = E\left[\hat{h}_R h_R\left(iT_s\right)\right] = E\left[\hat{h}_I h_I\left(iT_s\right)\right] \tag{E.2}$$

Since the focus here is on characterizing the effects of the channel variation independent of the channel estimation algorithm employed, assume that the outdated fading estimates are made perfectly; that is, $\hat{h}\left(iT_s - \tau_j\right) = h\left(iT_s - \tau_j\right),\ j = 1,...,N$. Since $h_R\left(iT_s\right)$ and the variables in $\hat{h}_R$ are jointly Gaussian, $h_R\left(iT_s\right)$ is Gaussian when conditioned on $\hat{h}_R = \hbar_R$ with mean $\rho^{\dagger}\Sigma_{\hat{h}}^{-1}\hbar_R$ and variance $\sigma^2 = R_h\left(0\right) - \rho^{\dagger}\Sigma_{\hat{h}}^{-1}\rho$. Likewise, $h_I\left(iT_s\right)$ is Gaussian when conditioned on $\hat{h}_I = \hbar_I$ with mean $\rho^{\dagger}\Sigma_{\hat{h}}^{-1}\hbar_I$ and variance $\sigma^2 = R_h\left(0\right) - \rho^{\dagger}\Sigma_{\hat{h}}^{-1}\rho$. Thus, $h_i = \left|h\left(iT_s\right)\right|$ is Rician when conditioned on $\hat{h}$ with conditional probability density function given by

$$\Pr_{h_i|\hat{h}}\left(h_i \middle| \hbar\right) = \frac{h_i}{\sigma^2} e^{-\left(h_i^2 + s^2/2\sigma^2\right)} I_0\left(\frac{h_i s}{\sigma^2}\right), \quad h_i \geq 0 \tag{E.3}$$

Where $I_0\left(.\right)$ is the zero$^{\text{th}}$-order modified Bessel function and the non centrality parameter is given by $s^2 = \left(\rho^{\dagger}\Sigma_{\hat{h}}^{-1}\hbar_R\right)^2 + \left(\rho^{\dagger}\Sigma_{\hat{h}}^{-1}\hbar_I\right)^2$. Note that σ^2 is the mean squared error of a minimum mean squared error (MMSE) predictor of the in-phase or quadrature fading value of interest, and s^2 is the sum of the squares of the MMSE prediction of $h_R(iT_s)$ and the MMSE prediction of $h_I(iT_s)$.

For interpretation of Eq. E.3, consider the N=1 case. Defining $\rho = R_h(\tau_1)$, and normalizing such that $E\left[\left(h_R(iT_s)\right)^2\right] = E\left[\left(h_I(iT_s)\right)^2\right] = 1$, one obtains $s^2 = |\hbar|^2\rho^2$ and $\sigma^2 = 1-\rho^2$. Define the specular-to-diffuse component (Rician) factor of (E.3) by $K = \left(s^2/2\sigma^2\right) = \left(|\hbar|^2\rho^2/2\left(1-\rho^2\right)\right)$. For a fixed correlation ρ, the effective channel conditioned on a single outdated fading estimate varies from a low-SNR nearly Rayleigh fading channel to a high-SNR strongly Rician fading channel, with the type of fading dependent on the value of the outdated estimate. A similar interpretation holds for $N > 1$.

E.2 Adaptive Coded QAM

In this section, strongly robust adaptive coded modulation will be designed. Only the strongly robust $N = 1$ case will be considered, which requires performance to be guaranteed for all $\rho = [\rho_{\min}, 1]$, where $\rho_{\min}$ is the minimum value of $R_h(\tau_1)$.

A. Design Rules

Temporarily, no energy adaptation is employed; in other words, the average energy of the signal constellation employed will be constant across symbols. Although the standard water-pouring energy allocation method can be employed. The proposed energy adaptation scheme will be presented in Section E.2-B.

Let P_b be the target bit error probability for the system, which operates at the average received signal-to-noise ratio E_s / N_0, where E_s is the average received energy per QAM symbol. Specification of the adaptive transmitter requires finding $\tilde{M}(v), \forall v$ where $\tilde{M}(v)$ is the number of signals in the QAM signal set employed when $\left|\hat{h}(iT_s - \tau_1)\right| = v$. If $\tilde{M}(v)$ is chosen such that P_b is maintained for each v

$$\tilde{M}(v) = \max\left\{M : \sup_{\rho_{\min}\le\rho\le 1} \tilde{P}_M\left(\frac{E_s}{N_0}, v, \rho\right) \le P_b\right\} \tag{E.4}$$

Where $\tilde{P}_M\left((E_s/N_0), v, \rho\right)$ is defined as the bit error probability of the M-QAM signal set at average received SNR E_s/N_0 when $R_h(\tau_1) = \rho$ and $\left|\hat{h}(iT_s - \tau_1)\right| = v$. Assume that maximum likelihood symbol detection given the current channel fading amplitude is employed on the samples of the matched filter output at the receiver. Conditioned on the value h_i of the current fading amplitude, the probability of bit error for M-QAM (which is upper bounded by the symbol error probability), $M > 2$, is upper bounded as

$$\begin{aligned} P_M\left(h_i^2 \frac{E_s}{N_0}\right) &\le 4Q\left(\sqrt{\frac{3}{2(M-1)}\frac{E_s}{N_0}h_i^2}\right) \\ &\le 2\exp\left(-\frac{3}{4(M-1)}\frac{E_s}{N_0}h_i^2\right) \end{aligned} \tag{E.5}$$

Where $Q(x) \triangleq \left(1/\sqrt{2\pi}\right)\int_x^\infty e^{-\left(u^2/2\right)}du$. The fact that the average received energy is twice the average transmitted energy has been recalled, and the second inequality is obtained by employing the bound $Q(\alpha) \le \frac{1}{2}e^{-\left(\alpha^2/2\right)}$. Results indicate that employing (E.5) for the calculation of $\tilde{P}_M\left(\left(E_s/N_0\right), v, \rho\right)$ results in very conservative designs, as do the results from employing the generally tighter bound $Q(\alpha) \le \left(1/\sqrt{2\pi\alpha}\right)e^{-\left(\alpha^2/2\right)}$ to obtain the analog of (E.5). Thus, the tight approximation will be employed for all M for much of the design work for coded systems.

$$P_M\left(h_i^2 \frac{E_s}{N_0}\right) \approx 0.2\exp\left(-\frac{3}{4(M-1)}\frac{E_s}{N_0}h_i^2\right) \tag{E.6}$$

Note that (8.22) is not an upper bound in all cases considered as it will be implicitly employed for all h_i and applied to M = 2. Using (E.6) yields

$$\begin{aligned} \tilde{P}_M\left(\frac{E_s}{N_0}, v, \rho\right) &= E\left[P_M\left(h_i^2\frac{E_s}{N_0}\right) \middle| \left|\hat{h}\left(iT_s - \tau_1\right)\right| = v\right] \\ &\approx \begin{cases} \dfrac{0.2\exp\left[-\dfrac{v^2\rho^2}{2\left(1-\rho^2\right)}\left(1 - \dfrac{1}{1+\dfrac{3}{2}\dfrac{E_s}{N_0}\dfrac{\left(1-\rho^2\right)}{(M-1)}}\right)\right]}{1+\dfrac{3}{2}\dfrac{E_s}{N_0}\dfrac{\left(1-\rho^2\right)}{(M-1)}} & if\ \rho < 1 \\ 0.2\exp\left(-\dfrac{3}{4}\dfrac{E_s}{N_0}\dfrac{v^2}{(M-1)}\right) & if\ \rho = 1 \end{cases} \end{aligned} \tag{E.7}$$

Where the approximation is obtained by substituting (E.3) and (E.6) into the equality and evaluating the expectation over h_i.

From (E.4), (E.7) must be evaluated at its supremum on $\rho \in \left[\rho_{\min}, 1\right]$. Since the right side of (E.7) is a continuous function on this closed interval, it achieves its maximum on this interval at a point which will be denoted $\rho*$. The following solution is found by standard calculus techniques. Let

$$\tilde{\rho} = \begin{cases} 0 & v \geq \sqrt{2} \\ \sqrt{\left(1 + \dfrac{2\left(M-1\right)}{3}\dfrac{N_0}{E_s}\right)\dfrac{\left(2-v^2\right)}{2}} & 0 \leq v \leq \sqrt{2} \end{cases}$$

The worst case autocorrelation is then given by

$$\rho^* = \begin{cases} \rho_{\min} & \tilde{\rho} \leq \rho_{\min} \\ \tilde{\rho} & \rho_{\min} < \tilde{\rho} < 1 \\ 1 & 1 \leq \tilde{\rho} \end{cases} \tag{E.8}$$

The signal set is specified using (E.7) and (E.8) in $\tilde{M}\left(v\right) = \max\left\{M : \tilde{P}_M\left(\left(E_s/N_0\right), v, \rho^*\right) \leq P_b\right\}$. Note that $\tilde{M}\left(v\right)$ is not decreasing in *v*. Thus, the adaptive scheme can be specified by the values v_m, *m* = 2, 4, 8, 16, 64 and 256, where v_m is defined as the threshold such that $v \geq v_m$, for *m*-QAM can be employed. For a fixed ρ, (E.7) can be solved explicitly to find v_m, but since ρ^* depends on *v*, the resulting equation is not useful in general if strong robustness is desired. However, in many situations, particularly for small P_b, it can be shown analytically that one need only consider $\rho = \rho_{\min}$, and thus the thresholds can be found explicitly.

B. Energy Adaptation

The rate of the method of the previous section is limited by the following reason: for all *v* such that $v_m < v < v_{2m}$, the estimate is more favorable than that required to use *m*-QAM but not favorable enough to use (*2m*)-QAM. A solution to this problem is to employ energy adaptation. First, a signal set is chosen according to the previous section with no energy adaptation. Then (E.7) and (E.8) are used to decide the minimum energy required to maintain P_b given the channel estimate *v*, thus employing a method similar to the power-pruning. This minimum energy is employed for the current symbol and the remainder of the energy is saved for the next symbol.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshina, "Near Shannon limit error correcting coding and decoding: Turbo codes," in IEEE int. conf. on Commun., pp. 1064-1070, 1993.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output maximum a posteriori (map) module to decode parallel and serial concatenated codes," TDA Progress Report 42-127, JPL, November 1996.

[3] J. Hagenaur, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE trans. on Information Theory, vol. IT-42, pp. 429-445, March 1996.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE trans. on Information Theory, pp. 284-287, March 1974.

[5] C. Douillard and al. "Iterative correction of inter-symbol interference: Turbo equalization," European Trans. on Commun., vol. 6, pp. 507-511, Sept.-Oct. 1995.

[6] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," Proc. Int. Symposium on Turbo codes & related topics, Brest, France, pp. 96-102, Sept. 1997.

[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performances, analysis, design and iterative decoding," TDA Progress Rep. 42-126, Aug.1996.

[8] G. Bauch, H. Khorram and J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," in Proc. EPMCC`97, Bonn, Germany, pp. 307-312, Sept.1997.

[9] J. Proakis, Digital Communications, 4th Ed. Singapore: McGraw-Hill, 2000.

[10] G. Forney, "Maximum-Likelihood sequence estimation of digital sequences in the presence of inter-symbol interference," IEEE Trans. on Information Theory, vol. 18, pp. 363-378, May 1972.

[11] Y. Li, B. Vucetic, Y. Sato, "Optimum soft-output detection for channels with inter-symbol interference," IEEE Trans. on Information Theory, vol. 41, pp. 704-713, May 1995.

[12] A.O. Berthet, B.S. Unal, and R. Visoz, "Iterative decoding of convolutionally encoded signals over multi-path Rayleigh fading channels," IEEE journal on selected areas in communications, vol. 19, No. 9, Sept. 2001.

[13] S. Ariyavisitakul and Y. Li, "Joint coding and decision feedback equalization for broadband wireless channels," IEEE J. select Areas Commun., vol. 16, pp. 1670-1678, Dec. 1998.

[14] X. Wang and H.V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," IEEE Trans. Commun., vol.47, pp. 1046-1061, July 1999.

[15] A.O. Berthet, R. Visoz, and P. Tortolier, "Sub-optimal Turbo detection for coded 8-PSK signals over ISI channels with application to EDGE advanced mobile system," in Proc. 11th Int. Symp. Personal, Indoor, and Mobile Radio Commun., vol. 1, pp. 151-157, Sept. 2000.

[16] M. Tuchler, R. Koetter, and A.C. Singer, "turbo equalization: Principles and new results," IEEE Trans. on Commun., vol. 50, pp. 754-767, May 2002.

[17] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization: Adaptive equalization and channel decoding jointly optimized," IEEE journal on selected areas in Commun., vol. 19, No 9, pp. 1744-1752, Sept. 2001.

[18] D. Raphaeli and A. Saguy, "Linear equalizers for turbo equalization: new optimizing criterion for determining the equalizer taps," in Proc. 2nd Int. Symp. Turbo codes, Brest, France, pp. 371-374, Sept. 2000.

[19] P. Vila and al., "Improved interference cancellation for turbo equalization," Acoustics, Speech, and Signal Processing ICASSP '00. Proceedings. IEEE International Conference, vol. 1, pp. 416 -419, 2000.

[20] A. Lampe, "Iterative multi-user detection with integrated channel estimation for coded DS-CDMA," IEEE Trans. on Commun., vol. 50, No. 8, pp. 1217-1223, Aug. 2002.

[21] A. Lampe, and J.B. Huber, "Iterative interference cancellation for DS-CDMA systems with high system Loads using reliability dependent feedback," IEEE Trans. on vehicular technology, vol. 51, No. 3, pp. 445 – 452, May 2002.

[22] M. Tuchler, R. Otnes, and A. Schmidbauer, "Performance of soft iterative channel estimation in turbo equalization," ICC 2002. IEEE International Conf. on Commun., vol. 3, pp. 1858-1862, 2002.

[23] L.M. Davis, I.B. Collings, and P. Hoecher, "Joint MAP equalization and channel estimation for frequency selective and frequency flat fast fading channels," IEEE Trans. on Commun., vol. 49, No. 12, pp. 2106-2113, Dec. 2001.

[24] G. Bauch, and V. Franz, "A comparison of soft-in soft-out algorithms for turbo detection," in Proc. ICT, vol. 2, Portos Carras, Greece, pp. 259-263, June 1998.

[25] C.N. Georghiades, and J.C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," IEEE Trans. on Commun., vol. 45, No. 3, pp. 300-308, March 1997.

[26] M. Sandell, C. Luschi, P. Strauch, and R. Yan, " Iterative channel estimation using soft decision feedback," Global Telecommun. Conf., GLOBECOM 98. The Bridge to Global Integration. IEEE, vol. 6, pp. 3728 -3733, 1998.

[27] S.T. Brink, J. Speidel, and R.H. Yan, "iterative demapping and decoding for multi-level modulation," Global Telecom. Conf., GLOBECOM 98, The Bridge to Global Integration. IEEE, vol. 1, pp. 579 - 584, 1998.

[28] S. ten Brink, "Iterative decoding for multi-code CDMA," Vehicular Technology Conf., IEEE 49th, vol. 3, pp. 1876-1880, July 1999.

[29] A. Gorokhov, and M. van Dijk, "Optimized labeling Maps for bit-interleaved transmission with turbo demodulation," Vehicular Technology Conf., VTC 2001 Spring. IEEE VTS 53rd, vol. 2, pp. 1459-1463, 2001.

[30] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," IEEE Trans. on Information Theory, vol. 42, pp. 409-429, March 1996.

[31] A. Klein, G. Kaleh, and P.W. Baier, "Zero forcing and minimum mean square error equalization for multi-user detection in code-division multiple access channels," IEEE trans. on vehicular technology, vol. 45, No. 2, pp. 276-287, May 1996.

[32] M. E. Austin, “Decision-feedback equalization for digital communication over dispersive channels,” M.I.T. Res. Lab Electron., Tech. Rep. 461, Aug. 1967.

[33] G. Caire and S. Shamai, “On the achievable throughput of a multiantenna Gaussian broad-cast channel,” *IEEE Trans.Inform.The ory*, vol. 49, pp. 1691–1706, July 2003.

[34] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Commun. Letters*, vol. 5, pp. 58–60, Feb. 2001.

[35] A. Duel-Hallen, “Decorrelating decision-feedback multiuser detector for synchronous CDMA,” *IEEE Trans.Commun.* , vol. 41, pp. 285–290, Feb. 1993.

[36] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs Tech.J.* , pp. 41–59, Autumn 1996.

[37] R. G. Gallager, “Low-density parity-check codes,” IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.

[38] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[39] A. Gersho and T. L. Lim, “Adaptive cancellation of intersymbol interference for data transmi-ssion,” *Bell Syst.T ech.J.* , vol. 60, pp. 1997–2021, Nov. 1981.

[40] H. Harashima and H. Miyakawa, “A method of code conversion for a digital communication channel with intersymbol interference,” *IEEE Trans.Commun.* , vol. COM-20, pp. 774–780, Aug. 1972.

[41] H. Imai and S. Hirakawa, “A new multilevel coding method using error correcting codes,” *IEEE Trans.Inform.The ory*, vol. 23, pp. 371–377, May 1977.

[42] J. Leech and N. J. A. Sloane, “Sphere packing and error-correcting codes,” *Canad.J.Math.*, vol. 23, pp. 718–745, 1971.

[43] U. Madhow and M. L. Honig, “MMSE interference suppression for direct-sequence spread spectrum CDMA,” *IEEE Trans.Commun.* , vol. 42, pp. 3178–3188, Dec. 1994.

[44] A. Ruiz, J. M. Cioffi, and S. Kasturia, “Discrete multiple tone modulation with coset coding for the spectrally shaped channel,” *IEEE Trans.Commun.* vol. 40, pp. 1012–1029, June 1992.

[45] C. E. Shannon, “A mathemathical theory of communication,” *Bell Syst.T ech.J.* ,vol. 27, pp. 379–423 and pp. 623–656, July and Oct. 1948.

[46] D. Shnidman, “A generalized Nyquist criterion and an optimum linear receiver for a pulse modulation system,” *Bell System Tech.J.* , vol. 46, pp. 2163–2177, Nov. 1967.

[47] M. Tomlinson, “New automatic equalizer employing modulo arithmetic,” *Electron. Lett.*, vol. 7, pp. 138–139, Mar. 1971.

[48] D. W. Tufts, “Nyquist’s problem—The joint optimization of transmitter and receiver in pulse amplitude modulation,” *Proc.IEEE* , vol. 53, pp. 248–259, Mar. 1965.

[49] M. K. Varanasi and B. Aazhang, “Near-optimum detection in synchronous codedivision multiple access systems,” *IEEE Trans.Commun.* , vol. 39, pp. 725–736, May 1991.

[50] S. Verdu, “Computational complexity of optimal multiuser detection,” *Algorithmica*, vol. 4, no. 3, pp. 303–312, 1989.

[51] Z. Xie, R. Short, and C. Rushforth, “A family of suboptimum detectors for coherent multiuser communications,” *IEEE J.Sele cted Areas Commun.*, vol. 8, pp. 683–690, May 1990.

[52] S. M. Aji, G. B. Horn, and R. J. McEliece. On the convergence of iterative decoding on graphs with a single cycle. In Proc. Int. Symp. Inform. Theory, page 276, Cambridge, MA, August 1998.

[53] G. D. Forney, Jr. Codes on graphs: Normal realizations. IEEE Trans. Inform. Theory.

[54] G. D. Forney, Jr., F. Kschischang, and B. Marcus. Iterative decoding of tail-biting trellisses. In Proc. Inform. Theory Workshop, pages 11-12, San Diego, February 1998.

[55] B. Frey. Turbo factor analysis. submitted to Neural Computation.

[56] F.R. Kschischang, B. Frey, and H.-A. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 1998.

[57] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, pages 249-258, 1998.

[58] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. In Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, pages 150-159, 1997.

[59] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low-density parity-check codes. Electronics Letters, 32:1645-1646, August 1996.

[60] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Transactions on Information Theory*, 47(2):619– 637, February 2001.

[61] T. J. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, February 2001.

[62] P. Rusmevichientong and B. Van Roy. An analysis of turbo decoding with Gaussian priors. In Advances in Neural Information Processing Systems, 1999.

[63] M. Sipser and D. A. Spielman. Expander codes. IEEE Trans. Inform. Theory, 42(6):1710-1722, November 1996.

[64] R. M. Tanner, "A recursive approach to low complexity codes", *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[65] Y. Weiss. Correctness of local probability propagation in graphical models with loops. to appear in Neural Computation, 2000.

[66] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In Advances in Neural Information Processing Systems, 1999.

[67] G. D. Forney, Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 268–278, March 1973.

[68] V. A. Zinoviev and V. V. Zyablov, "Decoding of non-linear generalized concatenated codes", *Probl. Peredach. Inform.*, vol. 14, pp. 46–52, 1978.

[69] J. L. Massey, *Threshold decoding*, MIT Press, Cambridge, MA, 1963.

[70] R. Diestel. *Graph Theory*. Springer-Verlag New York, 2000.

[71] Lucas, R., M.P.C. Fossorier, Yu Kou, and Shu Lin. "Iterative decoding of one step majority logic deductible codes based on belief propagation." Communications, IEEE Transactions on 48:931–937. June 2000.

[72] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(*q*)," *IEEE Commun. Letters*, vol. 2, no. 6, pp. 165–167, June 1998.

[73] M. C Davey, *Error-correction using low-density parity-check codes*, Ph.D. thesis, University of Cambridge, Cambridge, 1999.

[74] Motorola and Nokia, "Proposal for standardization of very high rate mixed voice-data traffic capabilities, based on extending and enhancing 1X systems," *3GPP2, S00-200003210-020*, March, 2000.

[75] Motorola and Nokia, "1XTREME," *3GGP2, S00-20000321-019*, March, 2000.

[76] Motorola, Nokia, Philips, TI and Altera, "Joint 1XTREME Proposal for 1XEV-DV," *3GPP2, C50-20010611-008a*, June, 2001.

[77] Motorola, Nokia, Texas Instruments, Altera, and Philips Semiconductors, "1XTREME Physical Specification for Integrated Data and Voice Services in cdma2000 Spread Spectrum Systems," *3GPP2, C50-200010611-013R1,* June, 2001.

[78] L. Hanzo, C. H. Wong, M. Yee, *Adaptive Wireless Transceivers: Turbo-Coded, Turbo-Equalized and Space-Time Coded TDMA, CDMA, and OFDM Systems.* (John Wiley, 2002)

[79] A. J. Goldsmith and P. P. Varaiya, "Capacity of Fading Channels with Channel Side Information," *IEEE Trans. Inform. Theory, vol. 43*, pp. 1986-1992, Nov. 1997

[80] J. F. Hayes, "Adaptive feedback communications," *IEEE Trans. Commun.Technol.*, vol. COM-16, pp. 29–34, Feb. 1968.

[81] J. K. Cavers, "Variable-rate transmission for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. COM-20, pp. 15–22, Feb. 1972.

[82] S. Otsuki, S. Sampei, and N. Morinaga, "Square-QAM adaptive modulation/TDMA/TDD systems using modulation level estimation with Walsh function," *Electron. Lett.*, vol. 31, pp. 169–171, Feb. 1995.

[83] W. T. Webb and R. Steele, "Variable rate QAM for mobile radio," *IEEE Trans. Commun.*, vol. 43, pp. 2223–2230, July 1995.

[84] Y. Kamio, S. Sampei, H. Sasaoka, and N. Morinaga, "Performance of modulation-level-controlled adaptive-modulation under limited transmission delay time for land mobile communications," in *Proc. IEEE VTC'95*, July 1995, pp. 221–225.

[85] B. Vucetic, "An adaptive coding scheme for time-varying channels," *IEEE Trans. Commun.*, vol. 39, pp. 653–663, May 1991.

[86] S. M. Alamouti and S. Kallel, "Adaptive trellis-coded multiple-phasedshift keying for Rayleigh fading channels," *IEEE Trans. Commun.*, vol.42, pp. 2305–2314, June 1994.

[87] T. Ue, S. Sampei, and N. Morinaga, "Symbol rate and modulation level controlled adaptive modulation/TDMA/TDD for personal communication systems," in *Proc. IEEE VTC'95*, July 1995, pp. 306–310.

[88] H. Matsuoka, S. Sampei, N. Morinaga, and Y. Kamio, "Symbol rate and modulation level controlled adaptive modulation/TDMA/TDD for personal communication systems," in *Proc. IEEE VTC'96*, Apr. 1996, pp. 487–491.

[89] S. Sampei, N. Morinaga, and Y. Kamio, "Adaptive modulation/TDMA with a BDDFE for 2 Mbit/s multi-media wireless communication systems," in *Proc. IEEE VTC'95*, July 1995, pp. 311–315.

[90] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, vol. 28, pp. 5–14, May 1990.

[91] M. Filip and E. Vilar, "Optimum utilization of the channel capacity of a satellite link in the presence of amplitude scintillations and rain attenuation," *IEEE Trans. Commun.*, vol. 38, pp. 1958–1965, Nov. 1990.

[92] A. J. Goldsmith and S. G. Chua, "Variable-Rate Variable-Power MQAM for Fading Channels," *IEEE Trans. Commun., vol. 45,* pp. 1218-1230, Oct. 1997.

[93] M-S. Alouini and A. J. Goldsmith, "Capacity of Rayleigh Fading Channels Under Different Adaptive Transmission and Diversity-Combining Techniques", *IEEE Trans. Veh. Technol., vol. 48,* pp. 1165-1181, July 1999.

[94] W. Choi, K. Cheong, and J. Cioffi, "Adaptive Modulation with Limited Peak Power for Fading Channels", *IEEE Proc.,* pp. 2568-2571, March. 2000.

[95] A. J. Goldsmith and S. G. Chua, "Adaptive Coded Modulation for Fading Channels," *IEEE Trans. Commun., vol. 45,* pp. 595-602, May 1998

[96] W. E. Ryan, L. L. McPheters, and S. W. McLaughlin, "Combined turbo coding and turbo equalization for PR4-equalized Lorentzian channels," in *Proceedings of the Conference on Information Sciences and Systems*, March 1998.

[97] W. E. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *Proceedings IEEE International Conference on Communications*, (Atlanta, GA, USA), pp. 947–951, IEEE, June 1998.

[98] T. Souvignier, A. Friedman, M. ¨Oberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo codes for PR4: Parallel versus serial concatenation," in *Proceedings IEEE International Conference on Communications*, (Vancouver, BC, Canada), pp. 1638–1642, IEEE, June 1999.

[99] J. Li, K. R. Narayanan, E. Kurtas, and C. N. Georghiades, "On the performance of high-rate TPC/SPC codes and LDPC codes over partial response channels," *IEEE Transactions on Communications*, vol. 50, pp. 723–734, May 2002.

[100] J. L. Fan, A. Friedmann, E. Kurtas, and S. W. McLaughlin, "Low density parity check codes for magnetic recording," in *Proceedings 37th Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL, USA), pp. 1314–1323, September 1999.

[101] H. Song, R. M. Todd, and J. R. Cruz, "Low density parity check codes for magnetic recording channels," *IEEE Transactions on Magnetics*, vol. 36, no. 5, pp. 2183–2186, 2000.

[102] J. Hou, P. H. Siegel, and L. B. Milstein. The performance analysis of low density paritycheck codes on Rayleigh fading channels. In *Proc. 38th Annual Allerton Conf. on Commun., Control, and Comp.*, volume 1, pages 266–275, Monticello, IL, USA, Oct. 2000.

[103] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1636-1652, July 2003.

[104] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf. Joint message-passing decoding of LDPC codes and partial-response channels. *IEEE Trans. Inform. Theory*, 48(6):1410–1422, June 2002.

[105] H. D. Pfister, J. B. Soriaga, and P. H. Siegel. On the achievable information rates of finite state ISI channels. In *Proc. IEEE Global Telecom. Conf.*, pages 2992–2996, San Antonio, Texas, USA, Nov. 2001.

Publications

[106] Y. Yuan-Wu, A. Ghaith and J. J. Boutros. An Iterative Channel Estimation and Interference Cancellation Procedure with Partial Decoupling of Observation for Coded Transmission. *IEEE VTC Spring 2004*, Milan, Italy. May 2004.

[107] A. Ghaith and Y. Yuan-Wu. Iterative OS Channel Estimation and IC Procedure for high level Coded modulation M-QAM. *IEEE VTC Fall 2004*, Los Angeles, United States Of America. September 2004.

[108] A. Ghaith, Y. Yuan-Wu and J. J. Boutros. Analysis and Performance Evaluation for Mapping Optimization in BICM-ID with OS Estimation and IC Equalization. *IEEE WiMob 2005*, Montreal, Canada. August 2005.

[109] A.Ghaith and Y. Yuan-Wu. OS Estimation and IC Equalization in BICM-ID with Mapping Optimization. *IEE Beyond 3G 2005*, London, United Kingdom. November 2005.

[110] A. Ghaith, J. J. Boutros and Y. Yuan-Wu. Fast and Reduced-Complexity Decoding Rule for *q*-ary LDPC Codes by Using the Duality Properties. *IEEE WiMob 2006*, Montreal, Canada. June 2006.

[111] J. J. Boutros, A. Ghaith and Y. Yuan-Wu. Non-binary adaptive low-density parity-check codes for frequency selective channels: code construction and iterative decoding. *IEEE Information Theory Workshop 2006*, Chengdu, China. October 2006.