INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° Attribué par la bibliothèque

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Signal, Image, Parole, Télécom

préparée au laboratoire **AST Grenoble Lab**

dans le cadre de l'Ecole Doctorale

Electronique, Electrotechnique, Automatique, Télécommunications, Signal

présentée et soutenue publiquement

par

Frédéric LEHMANN

le 10 décembre 2002

Titre :

Les Systèmes de Décodage Itératif et leurs Applications aux Modems Filaires et Non-filaires

---

Directeur de thèse : Joël LIENARD

JURY

M.	Philippe GODLEWSKI	Président
M.	Claude BERROU	Rapporteur
M.	Giuseppe CAIRE	Rapporteur
M.	Joël LIENARD	Directeur de thèse
M.	Denis MESTDAGH	Encadrant
M.	Ljupco KOCAREV	Examinateur

Remerciements

Je remercie en tout premier lieu mon directeur de thèse, le professeur Joël Lienard, chercheur au LIS, pour avoir encadré ce travail. Je remercie aussi les différents laboratoires chez France Telecom, STMicroelectronics et UCSD dans lesquels une partie de cette thèse s'est déroulée. Merci à STMicroelectronics, en particulier à Denis Mestdagh ainsi qu'à l'ANRT qui ont apporté le support financier nécéssaire au déroulement de cette thèse.

J'exprime ma gratitude envers Philippe Godlewski qui me fait l'honneur de présider le jury. Je remercie également Claude Berrou, professeur à l'ENST de Bretagne et Giuseppe Caire, professeur à l'institut Eurecom d'avoir rapporté de manière détaillée et constructive ce travail. Merci également à Ljupco Kocarev pour avoir examiné ce travail.

Je remercie enfin tous ceux qui ont fortement contribué à cette thèse par leurs conseils et de nombreuses discussions. Je pense en particulier à Jean-Marc Brossier, Thierry Lenez et Olivier Isson chez STMicroelectronics, ainsi qu'à Henry Pfister à UCSD. Un grand merci enfin à tous ceux avec lesquels j'ai eu le privilège de travailler sur les phénomènes non-linéaires lors de mon séjour à UCSD : le professeur Ljupco Kocarev, Gian Mario Maggio, Zarko Tasev et Bartolo Scanavino.

Introduction

De manière générale, le décodage itératif consiste à décoder un code par étapes successives à l'aide de plusieurs décodeurs à faible coût au lieu de décoder le code avec un seul décodeur complexe. Récemment, il a été démontré que les systèmes de décodage itératif peuvent fonctionner à des rendements très proches de la limite de Shannon, imposée par le théorème du codage sur canaux bruités [2], avec toutefois une complexité raisonnable. En particulier, les *codes LDPC* ("low-density parity-check") irréguliers et les *turbo codes* sont des candidats prometteurs pour de futures applications.

Les codes LDPC ont été introduits en premier par Gallager [3] en 1962 et redécouverts par MacKay *et al.* [8] en 1996. L'innovation cruciale des codes LDPC étant l'introduction des algorithmes de décodage itératif. Plus tard, l'apparition des turbo codes [16] en 1993 a montré que des performances proches de la capacité peuvent être atteintes en décodant des codes constitutifs avec un décodeur itératif de type SISO ("soft-input/soft-output").

Au cours des dix dernières années, la recherche dans les systèmes de décodage itératif s'est de plus en plus spécialisée. Dans cette thèse, nous allons présenter des contributions originales dans les domaines suivants.

Afin de mieux comprendre la structure des codes eux-mêmes et leurs performances avec un décodeur à vraisemblance maximale, les propriétés en terme de distance des codes doivent être étudiées [19].

Récemment, plusieurs techniques ont été proposées dans la littérature pour analyser le décodage itératif. La première méthode consiste à décrire des algorithmes de décodage itératif en tant que systèmes dynamiques non-linéaires [45]. On montre ainsi que toute une gamme de phénomènes tels que les points fixes, les orbites périodiques, les bifurcations [46] et le chaos [49] se produisent. Une seconde famille de méthodes consiste à suivre l'évolution de la densité de probabilité de l'information échangée dans le décodeur, ces densités étant souvent approchées par des gaussiennes [53, 54].

Finalement, l'avantage pratique du décodage itératif réside dans le fait que des codes puissants peuvent maintenant être décodés avec une complexité raisonnable. C'est pourquoi une grande part de la recherche est focalisée sur l'évaluation

des performances du décodage itératif pour de futurs systèmes de communication filaires et non-filaires.

L'objectif principal de cette thèse est d'exposer le lien entre les différents moyens existant à ce jour pour appréhender le fonctionnement du décodage itératif, à savoir :

1. la théorie des systèmes dynamiques non-linéaires
2. les techniques classiques du codage canal qui se rapportent à l'étude de performance d'un code à partir d'un polynôme connu sous le nom de fonction énumératrice de poids.

Plan de la thèse

La **première partie** de la thèse (Ch. 1 à 2) donne un résumé de l'état de l'art des systèmes de décodage itératif. Dans le Ch. 1, nous rappelons brièvement les concepts de base du décodage itératif à travers un exemple simple. Le Ch. 2 présente un aperçu des principaux codes qui utilisent à ce jour le décodage itératif, à savoir les codes LDPC et les codes concaténés. La construction du code, les propriétés en terme de distance ainsi que les algorithmes de décodage sont donnés dans chaque cas. Dans les parties suivantes, les contributions originales de la thèse sont développées.

La **seconde partie** de la thèse (Ch. 3 à 6) traite de l'analyse des systèmes de décodage itératif et de la structure sous-jacente des codes. Le Ch. 3 est centré sur la description des algorithmes de décodage itératif en tant que systèmes dynamiques non-linéaires. En traçant l'évolution de l'entropie moyenne et en fixant les rapports du bruit nous obtenons un système dynamique simplifié à une dimension et à un paramètre, puis nous procédons à une étude expérimentale des trajectoires du décodage itératif. Nous nous intéresserons en particulier à la dynamique non-linéaire du décodage itératif des codes LDPC et des codes produits. Dans les Ch. 4 et 5, nous proposons un modèle simplifié basé sur les densités de probabilité gaussiennes pour analyser le décodage itératif. Plus précisemment, nous calculons une fonction à 1-D dont les itérées représentent directement la probabilité d'erreur pour le canal à bruit gaussien additif (AWGN) et le canal à évanouissements de Rayleigh. Ces modèles simples permettent une analyse qualitative de la dynamique non-linéaire de l'algorithme de décodage. En particulier, nous verrons comment expliquer et calculer le seuil de bruit et comment caractériser la dynamique du décodage itératif à rapport signal sur bruit élevé en fonction des paramètres du code. Finalement, dans le Ch. 6, les propriétés relatives aux distances des codes LDPC sont étudiées. Pour l'ensemble des codes LDPC irréguliers, un lien entre la dynamique du décodage itératif et les propriétés

en terme de distance est établi.

La **troisième partie** (Ch. 7 à 8) est consacrée à l'évaluation des performances des systèmes de décodage itératif pour les modems filaires et non-filaires. Dans le Ch. 7 on considère l'effet du codage canal sur les algorithmes de chargement pour des systèmes VDSL utilisant la modulation BICM ("bit interleaved coded modulation"). On donne ensuite une comparaison des performances des codes Reed-Solomon et des codes BCH concaténés en série. Le Ch. 8 montre le bénéfice du décodage itératif pour les systèmes de transmission ATM non-filaires utilisant des code produits optimisés.

Acronymes and symboles

Acronymes

- ARQ : “Automatic Repeat Request”
- ATM : “Asynchronous Transfer Mode”
- AWGN : “Additive White Gaussian Noise”
- TEB : Taux d’Erreur Binaire
- BCH : Bose Chaudhuri Hocquenghem (code)
- BCJR : Bahl Cocke Jelinek Raviv (algorithme)
- BICM : “Bit Interleaved Coded Modulation”
- BPSK : “Binary Phase Shift Keying”
- CC : Codes Concaténés
- CWEF : “Conditional Weight Enumerating Function”
- DMT : “Discrete Multitone” (modulation)
- FDD : “Frequency division duplexing”
- FEXT : “Far-end crosstalk”
- i.i.d. : Indépendamment et identiquement distribué
- IRWEF : “Input Redundancy Weight Enumerating Function”
- IOWEF : “Input Output Weight Enumerating Function”
- LDPC : “Low Density Parity-Check” (code)
- MAP : Maximum A Posteriori
- PC : Codes Produits ou “Product Codes”
- PCC : Codes Concaténés en Parallèle ou “Parallel Concatenated Codes”
- MAQ : Modulation d’Amplitude en Quadrature
- RS : Reed-Solomon (code)
- SCC : Codes Concaténés en Série ou “Serially Concatenated Codes”
- SISO : “Soft-In Soft-Out” (algorithme)
- SOVA : “Soft Output Viterbi Algorithm”
- RSB : Rapport Signal sur Bruit
- VDSL : “Very high bit rate Digital Subscriber Line”
- WEF : “Weight Enumerating Function”

Symboles

- $(.)^T$: opérateur transposition
- d_{min} : distance minimum
- d_c : degré à droite maximum dans le graphe bipartite d'un code LDPC irrégulier
- d_H : distance de Hamming
- d_v : degré à gauche maximum dans le graphe bipartite d'un code LDPC irrégulier
- $C(n, k, d_{min})$: code en bloc de longueur n, de nombre de bits informatifs k et de distance minimum d_{min}
- E_b/N_0 : rapport énergie par bit sur densité spectrale de puissance du bruit
- $\mathcal{F}(.)$: opérateur transformée de Fourier
- λ : polynôme de distribution des degrés à gauche d'un code LDPC irrégulier
- $P(.)$: probabilité
- P_b : probabilité d'erreur binaire
- ρ : polynôme de distribution des degrés à droite d'un code LDPC irrégulier
- Q : fonction queue de distribution gaussienne
- σ : écart type du bruit du canal
- σ^* : seuil de bruit

Table des matières

Table des figures

Liste des tableaux

Première partie

Systèmes de Décodage Itératif

Chapitre 1

Concepts de Base du Décodage Itératif

Dans un système de communication numérique sur un canal bruité, un encodeur est souvent utilisé à l'émetteur avant l'étape de modulation, afin de corriger les erreurs de transmission au récepteur. Lorsque la structure du code s'y prête, le processus de décodage peut être exécuté en plusieurs étapes ou itérations simples, d'où le nom de décodage itératif. Dans ce chapitre, les principes du décodage itératif sont présentés à l'aide d'un exemple de code simple, qui illustre comment le fait d'itérer les sorties souples des décodeurs peut améliorer les performances en terme de taux d'erreur binaire. Le matériel présenté içi est largement repris de l'article didactique [1].

1.1 Rapports de Vraisemblance Logarithmiques

Nous considérons des codes concaténés formés de plusieurs codes constitutifs séparés par un entrelaceur. Les codes concaténés sont non seulement des codes puissants, mais leur structure permet de plus une faible complexité de décodage grâce au décodage itératif. Le décodage itératif consiste à décoder alternativement les codes constitutifs et à passer de l'information entre les décodeurs constitutifs. Afin d'exploiter au mieux l'information produite par chaque décodeur constitutif, il a été établi que passer des décisions souples plutôt que des décisions fermes peut conduire à d'excellentes performances. En 1993, Berrou, Glavieux et Thitimajshima [16] introduisirent les turbo codes. En utilisant un décodage itératif à décisions souples, ils présentèrent un code de rendement 1/2 atteignant une probabilité d'erreur binaire de 10^{-5} avec un E_b/N_0 de seulement 0.7 dB.

Les fondations mathématiques du décodage à décisions souples reposent sur le théorème de Bayes. Considérons une transmission non-codée utilisant la modu-

lation BPSK ("binary phase shift keying") sur le canal AWGN ("additive white Gaussian noise"). Un bit d'information $d = 0$ est transmis comme $x = +1$ et un bit d'information $d = 1$ est transmis comme $x = -1$. L'observation à la sortie du canal bruité est donnée par $y = x + n$, où n représente l'échantillon de bruit ayant une distribution gaussienne centrée d'écart-type σ. Soit $P(d = 0)$ (resp. $P(d = 1)$) la probabilité *a priori* que le digit émis soit un 0 (resp. un 1). Les probabilités *a posteriori* ou vraisemblances sont calculées à partir du théorème de Bayes' comme suit

$$\begin{aligned} P(d = 0|y) &= \frac{P(y|d = 0)P(d = 0)}{P(y)} \\ P(d = 1|y) &= \frac{P(y|d = 1)P(d = 1)}{P(y)} \end{aligned} \tag{1.1.1}$$

et peuvent être considérées comme un raffinement de la connaissance a priori sur la valeur du digit transmis fourni par l'observation du canal. La décision ferme optimale $\hat{d}$ au sens du *maximum a posteriori* (MAP) est alors la suivante

$$\hat{d} = \begin{cases} 0 & \text{si } P(d = 0|y) > P(d = 1|y) \\ 1 & \text{sinon.} \end{cases}$$

La décision souple est définie par le rapport de vraisemblance logarithmique

$$L(d|y) = \ln \frac{P(d = 0|y)}{P(d = 1|y)}.$$

En utilisant (1.1.1) la décision souple peut être décomposée en

$$L(d|y) = L_c(y) + L_a(d),$$

où $L_c(y) = \ln \frac{P(y|d=0)}{P(y|d=1)}$ est le rapport de vraisemblance logarithmique du canal et $L_a(d) = \ln \frac{P(d=0)}{P(d=1)}$ est le rapport de vraisemblance logarithmique *a priori*. Le critère de décision au sens du MAP peut alors se s'écrire

$$\hat{d} = \begin{cases} 0 & \text{if } L(d|y) > 0 \\ 1 & \text{sinon.} \end{cases}$$

Il s'ensuit que le signe de la décision souple détermine la décision ferme et que la valeur absolue de la décision souple détermine la fiabilité de cette décision. En particulier, si l'on suppose le bruit gaussien, le rapport de vraisemblance

logarithmique du canal a pour expression

$$L_c(y) = \ln \frac{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-1)^2}{2\sigma^2}}}{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y+1)^2}{2\sigma^2}}} = \frac{2}{\sigma^2} y.$$

Jusqu'à présent nous avons considéré un système de transmission non-codé. Pour un système de transmission codé, nous montrerons dans le Ch. 2 que si l'information a priori est distribuée de manière indépendante, la sortie souple d'un décodeur peut s'écrire sous la forme

$$L(d) = L_c(y) + L_a(d) + L_e(d), \tag{1.1.2}$$

où $L_e(d)$ est le rapport de vraisemblance logarithmique *extrinsèque* représentant la connaissance acquise grâce au processus de décodage.

1.2 Principes du Décodage Itératif

Considérons le code à deux dimensions illustré par la Fig. 1.1. Les données d sont réparties dans un tableau à k_1 lignes et k_2 colonnes. Un code horizontal génère le bloc de parité noté p_h et un code vertical génère le bloc de parité noté p_v. Chaque mot de code horizontal (resp. vertical) est de longueur n_2 (resp. n_1) et correspond à k_2 (resp. k_1) bits informatifs.

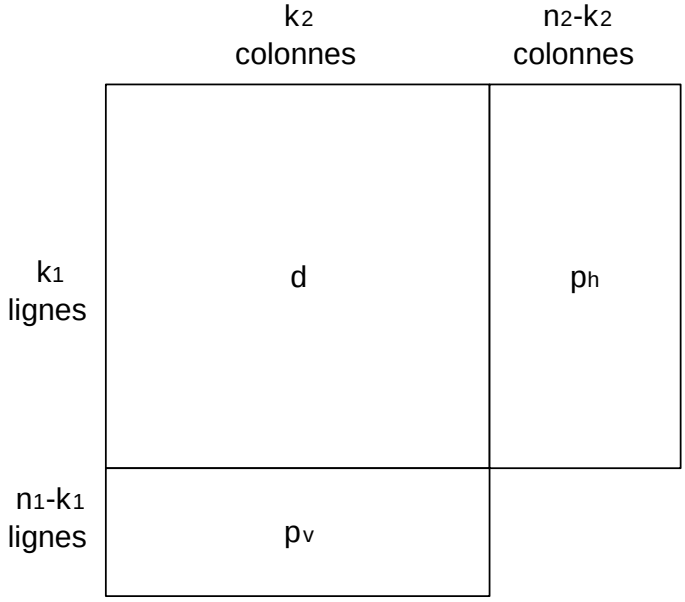

FIG. 1.1 – Code à deux dimensions.

Soit $L_{eh}(d)$ (resp. $L_{ev}(d)$) les rapports de vraisemblance logarithmiques *extrinsèques* obtenus à partir d'un décodage MAP horizontal (resp. vertical). Nous

supposerons que les données sont équiprobables, donc $L_a(d) = 0$ avant qu'un décodage n'aie eu lieu. L'algorithme de décodage itératif de ce code se déroule de la manière suivante :

1. Initialiser l'information a priori à $L_a(d) = 0$.
2. Décoder horizontalement et produire l'information extrinsèque comme suit

$$L_{eh}(d) = L(d) - L_c(y) - L_a(d).$$

3. Choisir $L_a(d) = L_{eh}(d)$.
4. Décoder verticalement et produire l'information extrinsèque comme suit

$$L_{ev}(d) = L(d) - L_c(y) - L_a(d).$$

5. Choisir $L_a(d) = L_{ev}(d)$.
6. Si suffisamment d'itérations se sont produites pour conduire à une décision ferme fiable, aller à 7 ; sinon retourner à 2.
7. Prendre des décisions fermes sur les données à partir du signe de $L(d) = L_c(y) + L_{eh}(d) + L_{ev}(d)$.

Notons que les décodeurs constitutifs échangent seulement le terme extrinsèque parce qu'il correspond à la nouvelle information acquise durant l'étape de décodage courante. Il est commode d'interpréter l'information extrinsèque comme de la diversité qui améliore les décisions souples à chaque étape de décodage. De plus, (1.1.2) est valide uniquement lorsque l'information a priori est distribuée de manière indépendante. Cette hypothèse est approximativement vérifiée d'une étape de décodage à la suivante grâce à l'entrelaceur intégré dans le code. En effet, l'information extrinsèque produite par le décodeur horizontal doit être désentrelacée avant d'être consommée par le décodeur vertical (et vice versa). Il s'ensuit que l'information extrinsèque devient à peu près décorrélée à l'entrée des décodeurs constitutifs.

1.3 Exemple : Code de Parité à Deux Dimensions

Considérons une équation de parité de la forme

$$c = a + b \mod 2.$$

Soient $L_c(y_a)$ et $L_c(y_b)$ les rapports de vraisemblance logarithmiques du canal correspondant à a et b, respectivement. Soient $L_a(a)$ et $L_a(b)$ les rapports de vraisemblance logarithmiques a priori correspondant à a et b, respectivement. On montrera dans le Ch. 2 que

$$\tanh\left(\frac{L_e(c)}{2}\right) = \tanh\left(\frac{L_c(y_a) + L_a(a)}{2}\right) \tanh\left(\frac{L_c(y_b) + L_a(b)}{2}\right), \qquad (1.3.3)$$

où $L_e(c)$ représente le rapport de vraisemblance logarithmique correspondant à c.

Nous considérons un code à deux dimensions où les codes constitutifs horizontal et vertical sont simplement des codes de parité ; et nous choisissons $k_1 = k_2 = 2$. Un exemple de mot de code avec une réalisation des rapports de vraisemblance logarithmiques du canal est donné par la Fig. 1.2. Notons que les décisions fermes basées sur les seules observations du canal sont erronées pour d_2 et d_3.

Mot de code

$d_1 = 1$	$d_2 = 0$	$p_{12} = 1$
$d_3 = 0$	$d_4 = 1$	$p_{34} = 1$
$p_{13} = 1$	$p_{24} = 1$	

Rapports de vraisemblance logarithmiques du canal

$L_c(y_1)$=-1.5	$L_c(y_2)$=-0.1	$L_c(y_{12})$=-2.5
$L_c(y_3)$=-0.2	$L_c(y_4)$=-0.3	$L_c(y_{34})$=-2.0
$L_c(y_{13})$=-6.0	$L_c(y_{24})$=-1.0	

FIG. 1.2 – Exemple de mot de code.

Nous appliquons maintenant la procédure de décodage itératif décrite dans la Sec. 1.2, en utilisant (1.3.3) afin de calculer les rapports de vraisemblance logarithmiques pour les lignes et les colonnes. Les Tab. 1.1 à 1.8 montrent l'évolution de l'information extrinsèque et des sorties pondérées mises à jour pour deux itérations. Les rapports de vraisemblance logarithmiques sont calculés avec une précision de 10^{-1}. Observons que les décisions fermes après le premier décodage

horizontal sont déjà correctes, cependant les fiabilités associées sont très faibles pour d_3 et d_4. Pour les étapes de décodage ultérieures, les décisions fermes restent inchangées tandis que la confiance augmente.

TAB. 1.1 – $L_{eh}(d)$ après le premier décodage horizontal.

0.1	1.4
0.3	0.2

TAB. 1.2 – Sorties souples mises à jour après le premier décodage horizontal.

-1.4	1.3
0.1	-0.1

TAB. 1.3 – $L_{ev}(d)$ après le premier décodage vertical.

-0.1	0.1
1.4	-0.8

TAB. 1.4 – Sorties souples mises à jour après le premier décodage vertical.

-1.5	1.4
1.5	-0.9

TAB. 1.5 – $L_{eh}(d)$ après le second décodage horizontal.

0.0	1.5
1.0	-1.1

TAB. 1.6 – Sorties souples mises à jour après le second décodage horizontal.

-1.6	1.5
2.2	-2.2

TAB. 1.7 – $L_{ev}(d)$ après le second décodage vertical.

-0.8	0.8
1.5	-0.8

TAB. 1.8 – Sorties souples mises à jour après le second décodage vertical.

-2.3	2.2
2.3	-2.2

Chapitre 2

Ensembles de Codes et Décodage Itératif

2.1 Capacité d'un Canal et Codage

Un système général de communication numérique est représenté par la Fig. 2.1. Nous faisons l'hypothèse standard que l'émetteur est constitué d'une source transmettant des messages de digits binaires i.i.d. (indépendamment et identiquement distribués) suivie d'un encodeur ajoutant des bits de redondance soigneusement choisis au message. Le mot de code binaire résultant est alors transmis sur un canal bruité. Ainsi qu'il a été montré par C.E. Shannon [2], la probabilité d'erreur devient arbitrairement faible lorsque la taille du bloc tend vers l'infini, seulement si le rendement de la transmission est inférieur à la *capacité* du canal. Au récepteur, un décodeur tente de trouver le message le plus vraisemblable sachant les observations du canal. L'application du critère du *maximum de vraisemblance* est optimal, mais le calcul de la vraisemblance pour chaque mot de code a une forte complexité pour des tailles de mot de code élevées. Cependant, des stratégies sous-optimales, telles que le décodage itératif, peuvent diminuer la complexité de manière significative tout en conservant de bonnes performances.

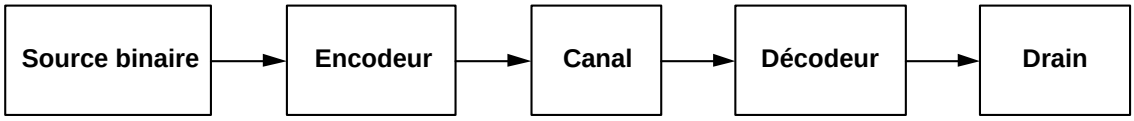

FIG. 2.1 – Modèle de communication numérique.

2.2 Codes LDPC

2.2.1 Construction des Codes

Les codes LDPC ("low-density parity-check") ont été introduits par Gallager [3, 4] en tant qu'une classe de codes avec une probabilité d'erreur évanescente lorsque la taille du bloc est infinie et une complexité de décodage raisonnable. La construction de code initialement proposée par Gallager est l'ensemble des codes LDPC *réguliers* de longueur n et défini par une matrice de parité $\mathbf{H}$ de dimension $(m \times n)$, où m représente le nombre de bits de parité. Pour les codes LDPC réguliers, la matrice $\mathbf{H}$ contient d_c (resp. d_v) uns dans chaque ligne (resp. colonne). Typiquement, d_c et d_v sont des entiers petits de telle sorte que $\mathbf{H}$ soit clairsemée. Cet ensemble est généralement appelé l'ensemble des (n, d_v, d_c) codes LDPC réguliers. Les positions des uns sont choisies au hasard et la séquence binaire $\mathbf{c} = (c_1, \ldots, c_n)$ de longueur n est un mot de code si et seulement si $\mathbf{Hc}^T = \mathbf{0}$. Un code LDPC peut aussi être représenté par son graphe bipartite associé [5, 6] où chaque bit dans le mot de code est représenté par un noeud de variable ou "à gauche" et chaque équation de parité est représenté par un noeud de parité ou "à droite" ; et une connexion relie un noeud de variable à un noeud de parité si et seulement si le noeud de parité intervient dans l'équation de parité correspondant au noeud de parité. Le nombre de connexions reliées à un noeud est appelé le degré du noeud.

Récemment, cette construction a été étendue à l'ensemble des codes LDPC irréguliers [12]. Le principe consiste à introduire des degrés de liberté dans le graphe en attribuant des degrés variables aux noeuds à gauche et à droite. Le graphe bipartite d'un code LDPC particulier est illustré par la Fig. 2.2. Supposons que d_v (resp. d_c) soit le degré maximum des noeuds de variable (resp. parité), alors le polynôme de distribution des degrés $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ (resp. $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$) spécifie la distribution des degrés des noeuds de variable (resp. parité). Par définition λ_i (resp. ρ_i) représente la fraction des connexions dans le graphe reliées à des noeuds de variable (resp. parité) de degré i. L'ensemble des (n, λ, ρ) codes LDPC irréguliers correspond aux graphes avec n noeuds de variable, des polynômes de distribution des degrés donnés par λ et ρ et des connexions choisies au hasard. Dans [7], on montre que le rendement du code est alors $r = 1 - \int_0^1 \rho(x)dx / \int_0^1 \lambda(x)dx$. En particulier, si $\lambda(x) = x^{d_v - 1}$ et $\rho(x) = x^{d_c - 1}$, le code est regulier et il n'existe aucun paramètre à optimiser. Cependant, pour les codes LDPC irréguliers, des degrés de liberté sont obtenus en permettant aux polynômes λ et ρ d'avoir plus d'une entrée non nulle. Ces polynômes peuvent alors être optimisés afin d'obtenir des performances proches de la limite de Shannon.

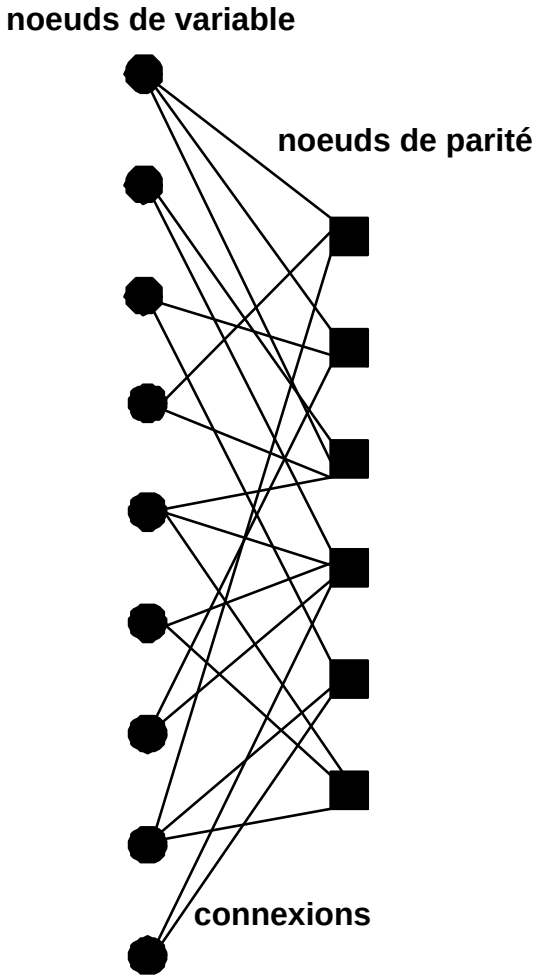

FIG. 2.2 – Graphe bipartite d'un code LDPC irrégulier.

2.2.2 Propriétés en Terme de Distance des Codes LPDC Réguliers

Nous introduisons tout d'abord la notion de *spectre de distance*. Soit $N(l)$ le nombre de mots de code de poids l, le *spectre de distance* est défini par $\{N(l), l \geq 0\}$. Nous définissons aussi la fonction énumératrice de poids ou "weight enumerating function"(WEF) donnée par

$$N(Z) = \sum_{l \geq 0} N(l) Z^l,$$

où Z est une variable formelle. Cet outil est utile pour calculer des bornes relatives à la probabilité d'erreur des mots de code P_e pour le canal gaussien à entrée binaire [24]

$$P_e \leq N(Z)|_{e^{-rE_b/N_0}},$$

où r représente le rendement de codage et E_b/N_0 est le rapport énergie par bit sur densité spectrale de puissance du bruit.

Le spectre de distance moyen de l'ensemble des codes LDPC réguliers, calculé par Gallager [4], sert à montrer que les codes LDPC réguliers sont asymptotiquement bons et à évaluer des bornes relatives à la probabilité d'erreur pour un décodage à vraisemblance maximale.

La construction de la matrice de parité $\mathbf{H}$ proposée par Gallager est la suivante. Soit H_1 la matrice de parité couvrant les n noeuds de variable avec n/d_c equations de parité non-recouvrantes contenant d_c uns consécutifs. L'exemple suivant va clarifier la construction.

Exemple 2.2.1. *Supposons que $n = 20$ et $d_c = 4$, il s'ensuit que,*

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

L'ensemble des (n, d_v, d_c) codes LDPC réguliers est défini comme les codes dont la matrice de parité est de la forme

$$\mathbf{H} = \begin{bmatrix} \pi_1(H_1) \\ \pi_2(H_1) \\ \vdots \\ \pi_{d_v}(H_1) \end{bmatrix}, \tag{2.2.1}$$

où $\pi_i(.)$ est definie comme une permutation aléatoire des colonnes de H_1, $i = 1, \ldots, d_v$. Par construction, pour tous les codes appartenant à cet ensemble, le nombre de noeuds de variable correspondant au graphe bipartite est n et le degré des noeuds de variable (resp. parité) est d_v (resp. d_c). Notons que sans perte de généralité $\pi_1(.)$ peut être choisie comme étant la permutation identité.

Etant donné la construction de l'ensemble décrite plus haut, soit $N_1(l)$ le nombre de séquences de poids l et de longueur n qui satisfont toutes les équations de parité spécifiées par H_1. L'énumérateur de poids d'un code de parité de longueur d_c est [4]

$$A_{d_c}(Z) = \frac{(1+Z)^{d_c} + (1-Z)^{d_c}}{2}.$$

Nous définissons $g_{d_c}(s) = A_{d_c}(Z = e^s)$ et $\mu_{d_c}(s) = \ln g_{d_c}(s)$. Par construction, une séquence de poids l qui satisfait H_1 est la concaténation de n/d_c séquences non-recouvrantes satisfaisant une équation de parité de longueur d_c. L'énumérateur

de poids de tout sous-code dont la matrice de parité est une permutation des colonnes de H_1 est par conséquent donné par

$$\sum_{l=0}^{n} N_1(l)e^{sl} = g_{d_c}(s)^{n/d_c}, \tag{2.2.2}$$

où nous avons utilisé le changement de variable $Z = e^s$. La partie gauche de (2.2.2) peut être bornée inférieurement par $N_1(l)e^{sl}$ quels que soient s et l, de sorte que

$$N_1(l) \leq \exp\left(\frac{n}{d_c}\mu_{d_c}(s) - sl\right). \tag{2.2.3}$$

L'exposant dans (2.2.3) est minimal lorsque,

$$l = \frac{n}{d_c}\mu'_{d_c}(s). \tag{2.2.4}$$

Soit $P(l)$ la probabilité qu'une séquence donnée de poids l et de longueur n satisfasse toutes les équations de parité dans $\mathbf{H}$, i.e. les équations dans toutes les matrices $\pi_i(H_1)$, $i = 1, \ldots, d_v$, alors

$$P(l) = \left(\frac{N_1(l)}{\binom{n}{l}}\right)^{d_v} = \binom{n}{l}^{-d_v} N_1(l)^{d_v}. \tag{2.2.5}$$

Le nombre moyen de mots de code de poids l dans l'ensemble des (n, d_v, d_c) codes LDPC réguliers est donné par [4]

$$N(l) = \binom{n}{l}P(l) = \binom{n}{l}^{-d_v+1} N_1(l)^{d_v} \leq C(\delta, n)\exp(-nB(\delta, d_v, d_c)), \tag{2.2.6}$$

avec

$$\begin{aligned}
C(\delta, n) &= [2\pi n\delta(1-\delta)]^{\frac{d_v-1}{2}} \exp\left(\frac{d_v - 1}{12n\delta(1-\delta)}\right) \\
B(\delta, d_v, d_c) &= (d_v - 1)H(\delta) - \frac{d_v}{d_c}\mu_{d_c}(s) + d_v s\delta \\
\delta &= \frac{\mu'_{d_c}(s)}{d_c} \\
H(\delta) &= -\delta\ln\delta - (1-\delta)\ln(1-\delta).
\end{aligned}$$

On peut montrer que pour $d_v > 2$, $B(\delta, d_v, d_c) = 0$ a une solution unique δ_0 pour $0 < \delta < 1/2$. De plus, $B(\delta, d_v, d_c) > 0$ pour $0 < \delta < \delta_0$ et $B(\delta, d_v, d_c) < 0$ pour $\delta_0 < \delta < 1/2$.

Soit d_{min} la distance minimum d'un code dans l'ensemble des (n, d_v, d_c) codes

LDPC réguliers, alors [4]

$$P(d_{min} \leq n\delta) \leq \sum_{l=1}^{n\delta} \binom{n}{l} P(l).$$

En utilisant (2.2.6), la borne supérieure suivante est obtenue

$$P(d_{min} \leq n\delta) \leq \sum_{l=1}^{n\delta} C(\delta = l/n, n) \exp\left[-nB(\delta = l/n, d_v, d_c)\right]. \tag{2.2.7}$$

Pour de grandes valeurs de n, le comportement de la somme dans (2.2.7) est déterminé par le signe de $B(\delta, d_v, d_c)$, avec saut en $\delta = \delta_0$. Cela signifie que la plupart des codes dans l'ensemble ont une distance minimum proche de ou supérieure à $\delta_0 n$, c'est pourquoi δ_0 est appelé le *rapport de distance minimum typique*. A rapport signal sur bruit élevé, le plancher d'erreur sur le canal à bruit gaussien additif (AWGN) à entrée binaire peut être exprimé comme le terme dominant de la borne par réunion relative à la probabilité d'erreur binaire des mots de code

$$P_e \approx N(d_{min}) e^{-r d_{min} E_b/N_0} \leq C(\delta_0, n) e^{-nr\delta_0 E_b/N_0}, \tag{2.2.8}$$

où r est le rendement de codage et E_b/N_0 represente le rapport énergie par bit sur densité spectrale de puissance du bruit. Donc le plancher d'erreur de presque tous les codes dans l'ensemble peut être rendu arbitrairement faible pourvu que n soit suffisamment grand.

De plus, lorsque d_c tend vers l'infini, δ_0 tend vers la borne de Gilbert-Varshamov qui est le rapport de distance minimum typique de l'ensemble équiprobable des codes [4].

Exemple 2.2.2. *Le Tab. 2.1 donne des valeurs numériques de δ_0 pour des codes LDPC réguliers de rendement* 1/2*. Lorsque d_c devient grand δ_0 est proche de* 0.11*, la borne de Gilbert-Varshamov.*

TAB. 2.1 – Rapport de distance minimum typique de codes LDPC réguliers de rendement 1/2.

d_v	d_c	δ_0
5	10	0.0843
6	12	0.0956
8	16	0.1052
15	30	0.1099

2.2.3 Décodage Itératif des Codes LDPC

Sauf mention expresse du contraire, l'algorithme de décodage itératif et son analyse seront présentés uniquement pour les (n, d_v, d_c) codes LDPC réguliers. La généralisation aux codes LDPC irréguliers est immédiate et peut être trouvée dans [7]. Puisque le décodage au sens du maximum de vraisemblance est optimal mais irréalisable en pratique, Gallager introduisit un algorithme de décodage itératif sous-optimal avec un nombre constant d'opérations par bit pour les codes LDPC [3]. Ce décodage à faible complexité a été revisité récemment par McKay [8], Kschischang *et al* [9] et Richardson et Urbanke [6] dans le contexte de la théorie des graphes. Nous définissons un *cycle* de longueur L dans un graphe bipartite (voir Fig. 2.2) comme un chemin reliant un noeud à lui même par l'intermédiaire de L connexions.

Le décodage itératif des codes LDPC consiste en un algorithme à passage de messages mettant à jour itérativement le rapport de vraisemblance des noeuds dans le graphe bipartite. Le principe général de l'algorithme est le suivant : à chaque itération, tous les messages sont supposés statistiquement indépendants. Cette hypothèse est raisonnable parce que la matrice de parité est clairsemée. C'est pourquoi, ainsi qu'il a été expliqué dans [9], les dépendances dues à l'existence de cycles peuvent être ignorées sans dégradation significative des performances de décodage.

Le message envoyé par un noeud sur une connection e est le rapport de vraisemblance de ce noeud, sachant les rapports de vraisemblance provenant de toutes les connexions incidentes, excepté e. Cette restriction est introduite pour éviter de fortes dépendances entre les messages échangés d'une itération à la suivante. La Fig. 2.3 a) illustre le message v envoyé par un noeud de variable à un noeud de parité, et est donné par

$$v = u_0 + \sum_{i=1}^{d_v - 1} u_i, \tag{2.2.9}$$

où u_0 est le rapport de vraisemblance en sortie du canal correspondant à ce noeud de variable. La Fig. 2.3 b) illustre le message u envoyé par un noeud de parité à un noeud de variable, et est donné par la "règle de la tanh" [6]

$$\tanh\left(\frac{u}{2}\right) = \prod_{i=1}^{d_c - 1} \tanh\left(\frac{v_i}{2}\right) \tag{2.2.10}$$

Une démonstration précise de la "règle de la tanh" sera présentée dans la Sec. 2.3.3. Les équations (2.2.9) et (2.2.10) constituent une itération de décodage et chaque

noeud de variable est initialisé par le rapport de vraisemblance en sortie du canal u_0, puisqu'à la première itération chaque noeud de parité est de manière équiprobable un 0 ou un 1.

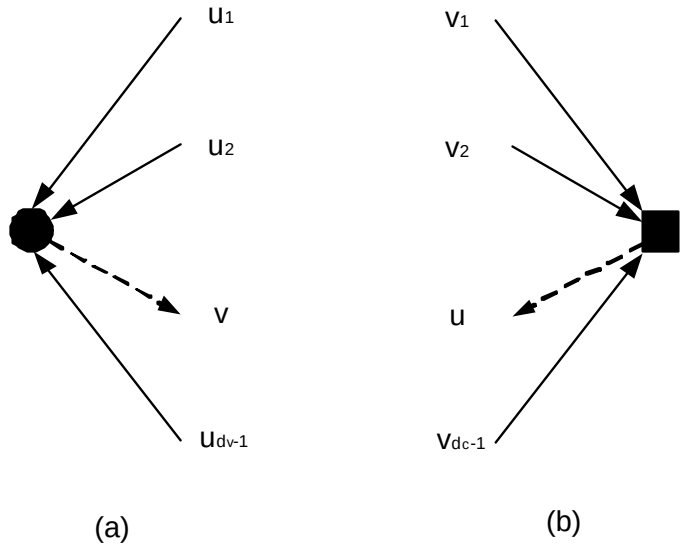

FIG. 2.3 – Passage de message a) d'un noeud de variable à un noeud de parité b) d'un noeud de parité à un noeud de variable.

2.2.4 Analyse du décodage Itératif des Codes LPDC

Une analyse exacte de cette méthode de décodage est connue sous le nom d'"évolution de densité" ou "density evolution" [6, 7]. Le raisonnement sous-jacent à l'idée d'évolution de densité est que pour des codes de grande longueur, un "théorème de concentration" ou "concentration theorem" [6] garantit que la performance d'un graphe particulier choisi au hasard peut être assimilée à la performance du graphe sans cycle correspondant, i.e. les messages échangés à chaque itération sont des variables aléatoires i.i.d. Soient $p_{u_0}(x)$, $p_v(x)$ et $p_u(x)$ la densité de probabilité des messages u_0, v et u, respectivement. Ces densitées existent puisquent les messages des noeuds de variable et de parité sont identiquement distribués pour une itération donnée l. Soit $\mathcal{F}(.)$ l'opérateur transformée de Fourier, il découle immédiatement de (2.2.9) que

$$p_v(x) = \mathcal{F}^{-1}\left[\mathcal{F}(p_{u_0}(x))\mathcal{F}(p_u(x))^{d_v-1}\right]. \tag{2.2.11}$$

De même, en se servant de (2.2.10), $p_u(x)$ peut être obtenu sous forme analytique en fonction $p_v(x)$ [6]. La définition de l'évolution de densité est le calcul des densités de probabilité des messages des noeuds de variable et de parité pour les itérations successives.

L'évolution de densité est utile pour montrer que les codes LPDC présentent un effet de seuil intéressant. Par exemple, considérons le canal gaussien à entrée binaire avec un écart-type du bruit σ et soit $P_e^l(\sigma)$ la fraction des messages issus des noeuds de variable conduisant à des décisions binaires erronées. On peut démontrer l'existence d'un seuil $\sigma^* = \sup\left\{\sigma > 0 : \lim_{l\to+\infty} P_e^l(\sigma) = 0\right\}$ [6]. Le Tab. 2.2. donne la valeur des seuils pour quelques codes LDPC réguliers de rendement $1/2$; la valeur de σ correspondant à la capacité est $\sigma_{opt} = 0.979$. De plus, l'évolution de densité appliquée aux codes LDPC irréguliers permet

TAB. 2.2 – Seuils de codes LDPC réguliers de rendement 1/2.

d_v	d_c	σ^*
3	6	0.88
4	8	0.83
5	10	0.79

d'optimiser les polynômes de distribution des degrés λ et ρ de manière à obtenir une valeur du seuil σ^* proche de la capacité [7, 10]. Des résultats similaires pour le canal binaire symétrique (CBS) ont été présentés dans [11].

2.3 Codes Concatenatés (CC)

2.3.1 Construction des Codes

Les codes concaténés ont été introduits en premier par Elias [13] et Forney [14] en tant qu'une classe de codes puissants avec un pouvoir de correction élevé. Une complexité de décodage faible était atteinte grâce à un algorithme de décodage séquentiel des codes constitutifs à entrées/sorties fermes. L'introduction du turbo-décodage, qui consiste en un décodage itératif des codes constitutifs à entrées/sorties souples ou "soft-input/soft-output" (SISO), suivi d'un échange d'information *extrinsèque* [15]-[17], a montré par la suite que des performances de décodage proches de la limite de Shannon peuvent être atteintes à l'aide de CC.

Par simplicité, nous nous restreindrons à des CC faisant intervenir des codes en bloc systématiques séparés par un entrelaceur. En particulier, si des codes convolutifs sont utilisés, nous supposerons que leur treillis est *terminé*, ce qui signifie que l'encodeur est ramené à l'état tout-zéro. Soient k, I et R le nombre de bits informatifs, la taille de l'entrelaceur et le rendement de codage du CC, respectivement.

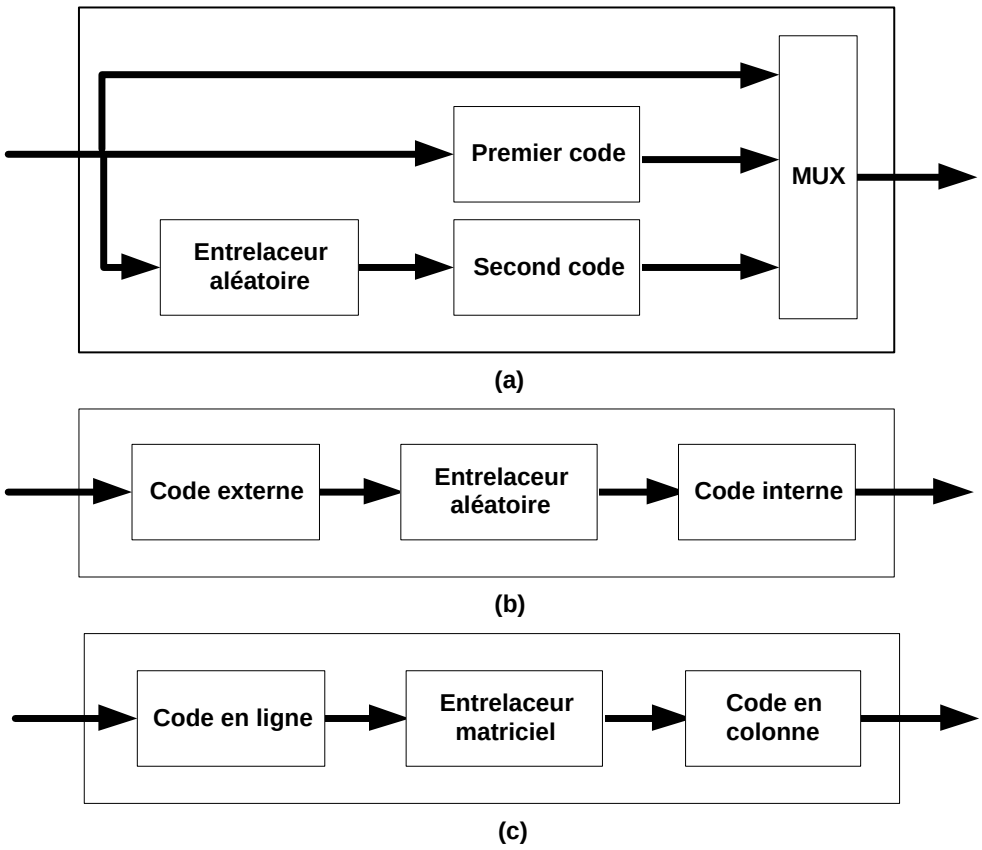

FIG. 2.4 – Codes concatenés : (a) PCC, (b) SCC, (c) PC.

Les codes concaténés en parallèle ou "parallel concatenated codes" (PCC) [19] sont décrits par la Fig. 2.4(a). Les k bits informatifs sont permutés par un entrelaceur aléatoire, ensuite les bits entrelacés et non-entrelacés sont fournis au premier et au second encodeur, respectivement. Le mot de code final est formé en multiplexant les bits informatifs avec les bits redondants émanant des encodeurs. Soient R_1 et R_2 le rendement des codes constitutifs d'un PCC. En négligeant l'effet de la terminaison du treillis si un code convolutif est utilisé, on obtient

$$R = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} - 1}$$
$$I = k.$$

Les codes concaténés en série ou "serially concatenated codes" (SCC) [21] sont décrits par la Fig. 2.4(b). Un code externe ou "outer code" ajoute de la redondance aux k bits informatifs et le mot de code résultant est permuté par un entrelaceur aléatoire. Le code interne ou "inner code" utilise la sortie de l'entrelaceur comme bits d'information et ajoute sa propre redondance. Le mot de code final est formé par la sortie de l'encodeur interne. Soient R_i et R_o le rendement des codes interne et externe, respectivement. En négligeant l'effet de

la terminaison du treillis si un code convolutif est utilisé, on obtient

$$R = R_i \times R_o$$
$$I = \frac{k}{R_o}.$$

Les codes produits ou "product codes" (PC) [13] sont décrits par la Fig. 2.4(c). Le code en ligne (resp. en colonne) est un code en bloc de type (n_R, k_R) (resp. (n_C, k_C)). les k bits informatifs sont placés dans un tableau de k_C lignes and k_R colonnes. Puis, les n_C lignes (resp. n_R colonnes) sont encodées en utilisant le code en ligne (resp. en colonne). Soient R_R et R_C le rendement du code en ligne et en colonne d'un PC, respectivement ; nous obtenons

$$R = R_R \times R_C$$
$$I = n_R \times n_C.$$

Notons qu'il n'y a pas de différence entre SCC et PC, à part le fait que l'entrelaceur aléatoire est remplacé par un entrelaceur matriciel.

2.3.2 Propriétés en Terme de Distance des Codes Concatenatés

Comme pour les codes LDPC, la distribution moyenne des distances est utile pour comprendre les performances asymptotiques des codes concaténés.

Par simplicité, nous faisons l'hypothèse standard qu'un PCC est formé de deux code constitutifs identiques, dont la "fonction énumératrice des poids d'entrée/redondance" ou "*input-redundancy weight enumerating function*" (IRWEF) [19] est

$$A(W, H) = \sum_w \sum_h A_{w,h} W^w H^h = \sum_w W^w A_w(H),$$

où $A_{w,h}$ représente le nombre de mots de code avec un poids informatif w ainsi qu'un poids redondant h, et $A_w(H)$ est la "fonction énumératrice de poids conditionnelle" ou "*conditional weight enumerating function*" (CWEF) correspondant aux séquences informatives de poids w. Les CWEF et IRWEF moyens d'un PCC sont obtenus comme suit [19]

$$A_w^{PCC}(H) = \frac{A_w(H)^2}{\binom{I}{w}}, \quad w = 0, \ldots, I$$
$$A^{PCC}(W, H) = \sum_w W^w A_w^{PCC}(H).$$

En appliquant la borne par réunion, la probabilité d'erreur binaire P_b avec décodage à vraisemblance maximale sur un canal gaussien à entrée binaire peut être majorée par [19]

$$P_b \leq \frac{W}{I} \frac{\partial A^{PCC}(W,H)}{\partial W}\bigg|_{W=H=e^{-REb/N_0}}.$$

Pour mieux comprendre l'influence des paramètres I et Eb/N_0, cette expression peut être réécrite comme suit

$$P_b \leq \sum_d C_d I^{\alpha(d)} e^{-dREb/N_0}, \tag{2.3.12}$$

où d représente le poids des mots de code, C_d est une constante et $\alpha(d)$ est l'exposant de la taille de l'entrelaceur. Donc la contribution de chaque mot de code de poids d dans la somme décroît exponentiellement avec Eb/N_0 et son comportement en fonction de la taille de l'entrelaceur dépend du signe et de la valeur absolue de l'exposant $\alpha(d)$. Le terme dominant, à rapport signal sur bruit modéré et pour une taille d'entrelaceur élevée, correspond à la valeur de d pour laquelle $\alpha(d)$ est maximum. En particulier, pour des PCC utilisant des codes convolutifs récursifs systématiques ou "recursive systematic convolutional codes" (RSC), il a été démontré que le terme dominant est [20]

$$P_b \leq \beta I^{-1} e^{-\frac{RE_b}{N_0}\left(\sum_{j=1}^{2} d_{j,2}^p + 2\right)},$$

où $d_{j,2}^p$ est le plus petit poids des parités en sortie du $j$ème code constitutif provenant de séquences informatives de poids 2, et β est une constante. Par conséquent, $d_{eff} = \sum_{j=1}^{2} d_{j,2}^p + 2$ est appelé la *distance minimum effective* et I^{-1} le *gain d'entrelacement*.

Le même raisonement s'applique aux SCC. Définissons la "fonction énumératrice de poids d'entrée/sortie" ou "*input-output weight enumerating function*" (IOWEF) des codes interne et externe par

$$A^o(W,L) = \sum_w \sum_l A_{w,l}^o W^w L^l$$
$$A^i(L,D) = \sum_l \sum_d A_{l,d}^i L^l D^d.$$

$A_{w,l}^o$ représente le nombre de mots de code de poids informatif w et de poids total l dans le code externe, et $A_{l,d}^i$ représente le nombre de mots de code de poids informatif l et de poids total d dans le code interne. L'IOWEF moyen

$A^{SCC}(W, D)$ d'un SCC est donné par [21]

$$A_{w,d}^{SCC} = \sum_{l=0}^{I} \frac{A_{w,l}^{o} \times A_{l,d}^{i}}{\binom{I}{l}}$$
$$A^{SCC}(W, D) = \sum_{w} \sum_{d} A_{w,d}^{SCC} W^{w} D^{d},$$

où $A_{w,d}^{SCC}$ est le nombre moyen de mots de code de poids informatif w et de poids total d dans le SCC. En appliquant la borne par réunion, la probabilité d'erreur binaire P_b avec décodage à vraisemblance maximale sur le canal gaussien à entrée binaire peut être majorée par [21]

$$P_b \leq \frac{1}{k} \frac{\partial A^{SCC}(W, D)}{\partial W}\bigg|_{W=1, D=e^{-REb/N_0}}.$$

Si les codes constitutifs sont de type RSC, cette expression à la même forme que (2.3.12). Soient d^o et d_{eff}^i la distance minimum du code externe et le poids minimum des mots de code générés par des séquences informatives de poids 2 dans le code interne. Si d^o est pair (resp. impair) le gain d'entrelacement vaut $I^{-\frac{d^o}{2}}$ (resp. $I^{-\frac{d^o+1}{2}}$) et la distance minimum effective vaut $\frac{d^o d_{eff}^i}{2}$ (resp. $\frac{(d^o-3)d_{eff}^i}{2} + h_m^{(3)}$, où $h_m^{(3)}$ est le poids minimum des mots de code générés par des séquences informatives de poids 3 dans le code interne) [21].

Il découle de la discusion précédente qu'une optimisation du gain d'entrelacement et de la distance minimum effective en choisissant des codes constitutifs convenables, constitue un critère approprié pour concevoir des PCC et des SCC. Des gains de performance supplémentaires peuvent être obtenus en concevant soigneusement l'entrelaceur [22, 23].

Pour un PC, la distance minimum est le produit des distances minimum des codes constitutifs [13]. Par conséquent, l'optimisation d'un PC se réduit au choix de codes constitutifs avec une distance minimum élevée. Par construction, l'entrelaceur est un entrelaceur matriciel et par conséquent ne peut pas être optimisé.

2.3.3 Décodage Itératif des Codes Concatenés

Soit $\mathbf{u} = (u_1, \ldots, u_k)$ une séquence informative binaire. Comme nous nous restreignons à deux codes constitutifs, nous pouvons noter $\mathbf{c}_1 = (c_{1,1}, \ldots, c_{1,n_1})$ et $\mathbf{c}_2 = (c_{2,1}, \ldots, c_{2,n_2})$ les mots de code résultant de l'encodage de $\mathbf{u}$ par les codes constitutifs, sans se soucier du fait de savoir si le CC est un PCC, un SCC ou un PC. Sans perte de généralité, posons que $\mathbf{c}_1$ correspond au premier code, au

code externe et au code ligne pour un PCC, un SCC et un PC, respectivement. Alors, $\mathbf{c}_2$ correspond au second code, au code interne et au code en colonne pour un PCC, un SCC et un PC, respectivement. Définissons la sortie d'un canal bruité sans mémoire correspondant à $\mathbf{c}_1$ et $\mathbf{c}_2$ par $\mathbf{y}_1 = (y_{1,1}, \ldots, y_{1,n_1})$ et $\mathbf{y}_2 = (y_{2,1}, \ldots, y_{2,n_2})$, respectivement. Si le canal délivre des sorties binaires (resp. analogiques), nous dirons que $\mathbf{y}_1$ et $\mathbf{y}_2$ constitutent des observations dures (resp. souples) en provenance du canal. Sauf mention expresse du contraire, nous supposerons que des observations souples en provenance du canal sont disponibles. Le décodeur optimal calcule des rapports de vraisemblance logarithmiques au sens du *maximum a posteriori* (MAP) comme suit [24]

$$L_i = \ln \frac{P(u_i = 0|\mathbf{y}_1, \mathbf{y}_2)}{P(u_i = 1|\mathbf{y}_1, \mathbf{y}_2)} = \ln \frac{\sum_{\mathbf{u}:u_i=0} P(\mathbf{u}|\mathbf{y}_1, \mathbf{y}_2)}{\sum_{\mathbf{u}:u_i=1} P(\mathbf{u}|\mathbf{y}_1, \mathbf{y}_2)}, \quad i = 1, \ldots, k.$$

Il est clair que ce décodeur admet des entrées souples en provenance du canal et produit des sorties souples dont le signe et la valeur absolue déterminent respectivement la décision ferme et la fiabilité de cette décision. Les décodeurs vérifiant cette propriété sont appelés des décodeurs à entrée/sortie souple ou "soft-input/soft-output (SISO) decoders". En appliquant le théorème de Bayes, on obtient

$$L_i = \ln \frac{\sum_{\mathbf{u}:u_i=0} P(\mathbf{y}_1, \mathbf{y}_2|\mathbf{u})P(\mathbf{u})}{\sum_{\mathbf{u}:u_i=1} P(\mathbf{y}_1, \mathbf{y}_2|\mathbf{u})P(\mathbf{u})}, \quad i = 1, \ldots, k.$$

Malheureusement, le décodeur optimal est trop complexe en pratique. Une alternative sous-optimale efficace, connue sous le nom de *turbo* décodage ou décodage *itératif*, a été introduite par Berrou [16] et consiste en un décodage séquentiel de type SISO des codes constitutifs.

Tout d'abord, considérons un décodage MAP de $\mathbf{c}_1$, sachant les observations du canal $\mathbf{y}_1$ et l'information *a priori* $P(\mathbf{c}_1)$, et appelons SISO1 le décodeur SISO correspondant. SISO1 calcule le vecteur de rapports de vraisemblance logarithmiques $\mathbf{L}_1 = (L_{1,1}, \ldots, L_{1,n_1})$, avec

$$L_{1,i} = \ln \frac{P(c_{1,i} = 0|\mathbf{y}_1)}{P(c_{1,i} = 1|\mathbf{y}_1)} = \ln \frac{\sum_{\mathbf{c}_1:c_{1,i}=0} P(\mathbf{y}_1|\mathbf{c}_1)P(\mathbf{c}_1)}{\sum_{\mathbf{c}_1:c_{1,i}=1} P(\mathbf{y}_1|\mathbf{c}_1)P(\mathbf{c}_1)}, \quad i = 1, \ldots, n_1.$$

Etant donné que les observations du canal sont i.i.d.

$$P(\mathbf{y}_1|\mathbf{c}_1) = \prod_{j=1}^{n_1} P(y_{1,j}|c_{1,j})$$

$$P(\mathbf{y}_2|\mathbf{c}_2) = \prod_{j=1}^{n_2} P(y_{2,j}|c_{2,j}).$$

Supposons que l'information *a priori* disponible soit aussi i.i.d., alors

$$\begin{aligned} P(\mathbf{c}_1) &= \prod_j P(c_{1,j}) \\ P(\mathbf{c}_2) &= \prod_j P(c_{2,j}). \end{aligned} \tag{2.3.13}$$

Comme nous allons le voir, selon le type de CC considéré (PCC, SCC ou PC), le nombre de termes dans les produits apparaissant dans l'Eq. (2.3.13) n'est pas le même. Il s'ensuit immédiatement que

$$L_{1,i} = \ln \frac{\sum_{\mathbf{c}_1:c_{1,i}=0} \prod_{j=1}^{n_1} P(y_{1,j}|c_{1,j}) \prod_j P(c_{1,j})}{\sum_{\mathbf{c}_1:c_{1,i}=1} \prod_{j=1}^{n_1} P(y_{1,j}|c_{1,j}) \prod_j P(c_{1,j})}, \quad i = 1, \ldots, n_1.$$

Définissons $\mathbf{Z}_1 = (Z_{1,1}, \ldots, Z_{1,n_1})$, $\mathbf{A}_1 = (A_{1,1}, \ldots, A_{1,n_1})$ et $\mathbf{E}_1 = (E_{1,1}, \ldots, E_{1,n_1})$ comme le vecteur d'information du canal, *a priori* et *extrinsèque*, respectivement, avec

$$\begin{aligned} Z_{1,i} &= \ln \frac{P(y_{1,i}|c_{1,i}=0)}{P(y_{1,i}|c_{1,i}=1)} \\ A_{1,i} &= \ln \frac{P(c_{1,i}=0)}{P(c_{1,i}=1)} \\ E_{1,i} &= \ln \frac{\sum_{\mathbf{c}_1:c_{1,i}=0} \prod_{j\neq i} P(y_{1,j}|c_{1,j}) \prod_{j\neq i} P(c_{1,j})}{\sum_{\mathbf{c}_1:c_{1,i}=1} \prod_{j\neq i} P(y_{1,j}|c_{1,j}) \prod_{j\neq i} P(c_{1,j})}, \quad i = 1, \ldots, n_1. \end{aligned}$$

Les rapports de vraisemblance logarithmiques *a posteriori* correspondant au décodage de $\mathbf{c}_1$ peuvent être réécrits comme suit

$$L_{1,i} = Z_{1,i} + A_{1,i} + E_{1,i}, \quad i = 1, \ldots, n_1.$$

De même, pour l'autre décodeur appelé SISO2, definissons $\mathbf{Z}_2 = (Z_{2,1}, \ldots, Z_{2,n_2})$, $\mathbf{A}_2 = (A_{2,1}, \ldots, A_{2,n_2})$ et $\mathbf{E}_2 = (E_{2,1}, \ldots, E_{2,n_2})$ comme le vecteur d'information du canal, *a priori* et *extrinsèque*, respectivement, avec

$$\begin{aligned} Z_{2,i} &= \ln \frac{P(y_{2,i}|c_{2,i}=0)}{P(y_{2,i}|c_{2,i}=1)} \\ A_{2,i} &= \ln \frac{P(c_{2,i}=0)}{P(c_{2,i}=1)} \\ E_{2,i} &= \ln \frac{\sum_{\mathbf{c}_2:c_{2,i}=0} \prod_{j\neq i} P(y_{2,j}|c_{2,j}) \prod_{j\neq i} P(c_{2,j})}{\sum_{\mathbf{c}_2:c_{2,i}=1} \prod_{j\neq i} P(y_{2,j}|c_{2,j}) \prod_{j\neq i} P(c_{2,j})}, \quad i = 1, \ldots, n_2. \end{aligned}$$

Les rapports de vraisemblance logarithmiques *a posteriori* correspondant au

décodage de $\mathbf{c}_2$ peuvent être réécrits comme suit

$$L_{2,i} = Z_{2,i} + A_{2,i} + E_{2,i}, \quad i = 1, \ldots, n_2.$$

Le décodeur itératif complet est décrit par la Fig. 2.5. Un entrelaceur $\mathbf{P}$ et un désentrelaceur $\mathbf{P}^{-1}$ sont nécéssaires pour réordonner les rapports de vraisemblance logarithmiques extrinsèques après chaque décodage constitutif. Le

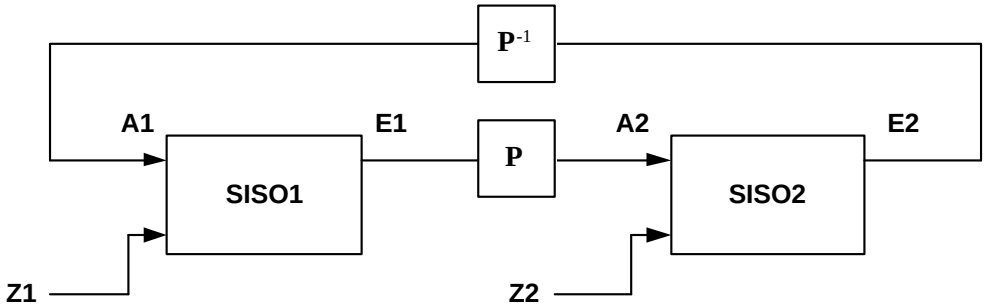

FIG. 2.5 – Schéma bloc du décodage itératif d'un CC.

décodage itératif d'un CC consiste en un algorithme à passage de messages, mettant à jour de manière itérative les rapports de vraisemblance logarithmiques des bits de chaque code constitutif. Les messages reçus par un décodeur en provenance de l'autre décodeur sont utilisés en tant qu'information *a priori*. Le principe général de l'algorithme est le suivant : à chaque itération, les messages reçus par un décodeur sont supposés statistiquement indépendants. Cette hypothèse est raisonnable lorsque la taille de l'entrelaceur est grande. Alors, les dépendances dues à la corrélation résiduelle entre les messages peuvent être ignorées sans dégradation catastrophique des performances de décodage. Donc l'Eq. (2.3.13) est justifiée. Ce décodage est réalisé itérativement en utilisant séquentiellement les décodeurs SISO1 et SISO2. SISO1 utilise les observations en provenance du canal $\mathbf{Z}_1$ et l'information *a priori* $\mathbf{A}_1$ sous la forme de rapports de vraisemblance logarithmiques, afin de générer les rapports de vraisemblance logarithmiques bit à bit *a posteriori* $\mathbf{L}_1$. L'information *extrinsèque* est alors définie par $\mathbf{E}_1 = \mathbf{L}_1 - \mathbf{Z}_1 - \mathbf{A}_1$. Après entrelacement, $\mathbf{E}_1$ est utilisé en tant qu'information *a priori* $\mathbf{A}_2$ en conjonction avec $\mathbf{Z}_2$ par SISO2 afin de générer les rapports de vraisemblance logarithmiques bit à bit *a posteriori* $\mathbf{L}_2$. L'information *extrinsèque* est alors définie par $\mathbf{E}_2 = \mathbf{L}_2 - \mathbf{Z}_2 - \mathbf{A}_2$ et est uilisé en tant qu'information *a priori* par SISO1 après désentrelacement. Les rapports de vraisemblance logarithmiques *a priori* sont initialisés à zéro puisque tant qu'aucun décodage n'a eu lieu, chaque bit est *a priori* de manière équiprobable un 0 ou un 1. Notons que les messages échangés dans le décodeur contiennent seulement le terme *extrinsèque*. Cette restriction est introduite afin d'éviter de fortes dépendances entre les messages

échangés d'une itération à l'autre.

Il n'existe que des différences minimes entre le décodage itératif d'un PCC, d'un SCC et d'un PC. Ainsi que nous l'avons déjà mentionné, ces différences influencent seulement le terme *a priori*. Pour un PCC, les codes constitutifs n'ont que les bits informatifs en commun, c'est pourquoi les décodeurs ne peuvent utiliser d'information *a priori* que pour ces bits. Dans le cas d'un SCC, l'encodeur interne encode à la fois les bits d'information et les bits redondants générés par le code externe. C'est pourquoi de l'information *a priori* est disponible à la fois pour les bits informatifs et les bits redondants à l'entrée du SISO externe, tandis que de l'information *a priori* n'est disponible que pour les bits informatifs à l'entrée du SISO interne [21]. De même, pour un PC, puisque par construction toutes les lignes et colonnes sont des mots de code, de l'information *a priori* est disponible à la fois pour les bits informatifs et les bits redondants à l'entrée des SISO en ligne et en colonne [18].

Exemple 2.3.1. *En guise d'exemple, nous présentons le calcul dû à Gallager, de l'information* extrinsèque *pour un code de parité de longueur* d_c.

Lemme 2.3.2. *Nous commençons par un lemme de Gallager [3]. Considérons une séquence de* n *digits binaires indépendants dans laquelle le* i*ème digit est un* 0 *avec une probabilité* p_i^0 *et un* 1 *avec une probabilité* p_i^1. *Alors, la probabilité qu'un nombre pair de digits soient à* 1 *vaut*

$$\frac{1+\prod_{i=1}^{n}(p_i^0-p_i^1)}{2},$$

et la probabilité qu'un nombre impair de digits soient à 1 *vaut*

$$\frac{1-\prod_{i=1}^{n}(p_i^0-p_i^1)}{2}.$$

Démonstration. Considérons la fonction

$$f(t)=\prod_{i=1}^{n}(p_i^0+p_i^1 t).$$

Observons que le coefficient de t^j dans le dévloppement en série de $f(t)$ est la probabilité qu'il y ait j uns. La fonction $f(-t)$ est identique, excepté que toutes les puissances impaires de t sont négatives. C'est pourquoi $\frac{f(t)+f(-t)}{2}|_{t=1}$ (resp. $\frac{f(t)-f(-t)}{2}|_{t=1}$) represente la probabilité qu'un nombre pair (resp. impair) de digits soient à 1. □

Lemme 2.3.3. *Maintenant, nous introduisons un autre résultat intéressant dû à Richardson et Urbanke [6]. Si m est le rapport de vraisemblance logarithmique $\ln \frac{p^0}{p^1}$, alors il s'ensuit que $p^0 - p^1 = \tanh\left(\frac{m}{2}\right)$. Réciproquement, si $q = p^0 - p^1$, alors $\ln \frac{p^0}{p^1} = \ln \frac{1+q}{1-q}$.*

La démonstration procède de manipulation algébriques simples.

Soit $\mathbf{c}$ *un mot de code quelconque d'un code de parité de longueur d_c, définissons les probabilités* a priori $p_i^0 = P(c_i = 0|y_i)$ *et* $p_i^1 = P(c_i = 1|y_i)$, *pour* $i = 1, \ldots, d_c$. *Par définition, le rapport de vraisemblance logarithmique* extrinsèque *bit à bit s'écrit*

$$u_i = \ln \frac{\sum_{\mathbf{c}:c_i=0} \prod_{j \neq i} P(c_j|y_j)}{\sum_{\mathbf{c}:c_i=1} \prod_{j \neq i} P(c_j|y_j)}, \quad i = 1, \ldots, d_c.$$

Nous rappelons que dans le contexte des codes LDPC, la notation u_i désigne le message envoyé par un noeud de parité, tandis que dans le contexte des CC, u_i désigne le $i$ème bit d'information. Il est clair que le numérateur est la probabilité qu'un nombre pair de digits soient à un, sachant que $c_i = 0$ et que le dénominateur est la probabilité qu'un nombre impair de digits soient à un, sachant que $c_i = 1$, donc par le lemme 2.3.2, le rapport de vraisemblance logarithmique extrinsèque *est équivalent à,*

$$u_i = \ln \frac{1 + \prod_{j \neq i}(p_j^0 - p_j^1)}{1 - \prod_{j \neq i}(p_j^0 - p_j^1)}, \quad i = 1, \ldots, d_c.$$

En définissant les rapports de vraisemblance logarithmiques a priori *par $v_i = \ln \frac{p_i^0}{p_i^1}$ pour $i = 1, \ldots, d_c$, le lemme 2.3.3 conduit à la dénomée "règle de la tanh" de la Sec. 2.2.3*

$$\tanh\left(\frac{u_i}{2}\right) = \prod_{j \neq i} \tanh\left(\frac{v_j}{2}\right), \quad i = 1, \ldots, d_c.$$

La "règle de la tanh" des codes de parité est le seul exemple bien connu où l'information extrinsèque *admet une expression analytique. En général ce n'est pas le cas et l'information* extrinsèque *doit être calculée numériquement en soustrayant explicitement les rapports de vraisemblance logarithmiques* a priori *et du canal aux rapports de vraisemblance logarithmiques* a posteriori. *La section suivante pésente des algorithmes SISO efficaces pour réaliser ceci.*

2.4 Algorithmes de Décodage SISO pour les CC

L'objectif de cette section est de présenter des algorithmes SISO pratiques calculant des rapports de vraisemblance logarithmiques bit à bit au sens du MAP ou des approximations convenables. Presque tous les CC peuvent être décodés

itérativement en utilisant l'un des algorithmes présentés.

2.4.1 Algorithme BCJR

L'algorithme BCJR, nommé d'après ses auteurs [25], permet un calcul exact de rapports de vraisemblance logarithmiques symbole par symbole au sens du MAP pour une chaîne de Markov discrète. Des applications de l'algorithme BCJR au décodage itératif ont été initialement proposées dans [16] et discutées plus tard dans de nombreuses publications [26]-[35]. Nous allons présenter içi l'approche proposée initialement par Bahl *et al* [25].

Considérons une chaîne de Markov dont les M états sont indexés par $m = 0, 1, \ldots, M-1$. L'état de la chaîne de Markov à l'instant t est désigné par S_t et sa sortie par X_t. Une séquence d'états de l'instant t à t' est désignée par $\mathbf{S}_t^{t'} = S_t, S_{t+1}, \ldots, S_{t'}$ et les symboles de sortie correspondants par $\mathbf{X}_t^{t'} = X_t, X_{t+1}, \ldots, X_{t'}$.

Les probabilités de transition sont données par

$$p_t(m|m') = P(S_t = m|S_{t-1} = m')$$

et les probabilités de sortie par

$$q_t(X|m, m') = P(X_t = X|S_t = m, S_{t-1} = m').$$

La chaîne de Markov débute à l'état initial $S_0 = 0$, produit la séquence de symboles de sortie $\mathbf{X}_1^\tau$ et termine dans l'état final $S_\tau = 0$. La séquence $\mathbf{X}_1^\tau$ est envoyée sur un canal sans mémoire dont la sortie bruitée est $\mathbf{Y}_1^\tau = Y_1, Y_2, \ldots, Y_\tau$. Les probabilités de transition du canal $R(\cdot|\cdot)$ vérifient

$$P(\mathbf{Y}_1^t|\mathbf{X}_1^t) = \prod_{j=1}^{t} R(Y_j|X_j), \quad t = 1, \ldots, \tau.$$

Selon l'application, nous aurons besoin des probabilités *a posteriori* des états

$$P(S_t = m|\mathbf{Y}_1^\tau) = P(S_t = m, \mathbf{Y}_1^\tau)/P(\mathbf{Y}_1^\tau)$$

ou des transitions

$$P(S_{t-1} = m', S_t = m|\mathbf{Y}_1^\tau) = P(S_{t-1} = m', S_t = m, \mathbf{Y}_1^\tau)/P(\mathbf{Y}_1^\tau).$$

Par commodité, nous introduisons les probabilités conjointes

$$\begin{aligned} \lambda_t(m) &= P(S_t = m, \mathbf{Y}_1^\tau) \\ \sigma_t(m, m') &= P(S_{t-1} = m', S_t = m, \mathbf{Y}_1^\tau). \end{aligned}$$

Exemple 2.4.1. *Un code convolutif peut être considéré comme une chaîne de Markov avec $M = 2^\nu$ états, où ν représente la taille du registre à décalage de l'encodeur. Le code convolutif récursif systématique à 4 états dont les polynômes générateurs sont $g_0(D) = 1 + D + D^2$ et $g_1(D) = 1 + D^2$ est illustré par la Fig. 2.6. En représentant $g_0(D)$ en octal par 7 et $g_1(D)$ par 5, ce code est noté $(1, 5/7)$. La représentation en chaîne de Markov de ce code est donnée par le diagramme de transition d'états de la Fig. 2.7, où chaque transition entre les états est annotée par le bit d'information suivi du bit de parité correspondant. Dans la représentation en treillis illustré par la Fig. 2.8, les états sont représentés par des noeuds et les transitions valides par des branches annotées par les sorties correspondantes.*

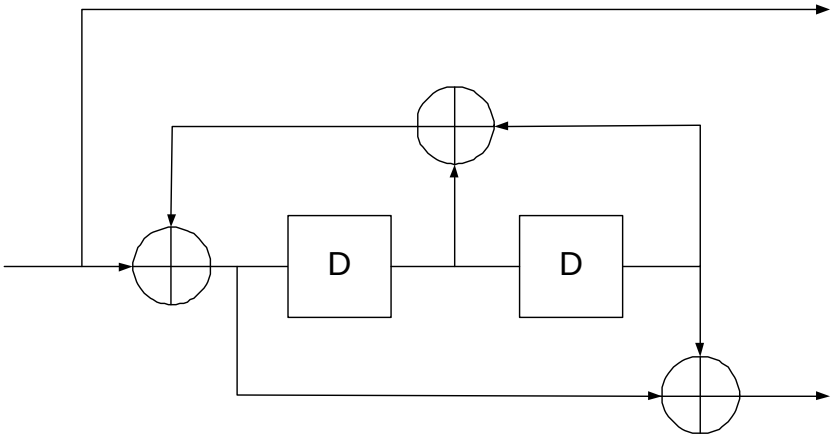

FIG. 2.6 – Schéma bloc du code convolutif $(1, 5/7)$.

Par commodité, nous définissons les probabilités suivantes

$$\begin{aligned} \alpha_t(m) &= P(S_t = m, \mathbf{Y}_1^t) \\ \beta_t(m) &= P(\mathbf{Y}_{t+1}^\tau | S_t = m) \\ \gamma_t(m', m) &= P(S_t = m, Y_t | S_{t-1} = m') \end{aligned}$$

Appliquons le théorème de Bayes

$$\begin{aligned} \lambda_t(m) &= \alpha_t(m)\beta_t(m) \\ \sigma_t(m', m) &= \alpha_{t-1}(m')\gamma_t(m', m)\beta_t(m). \end{aligned}$$

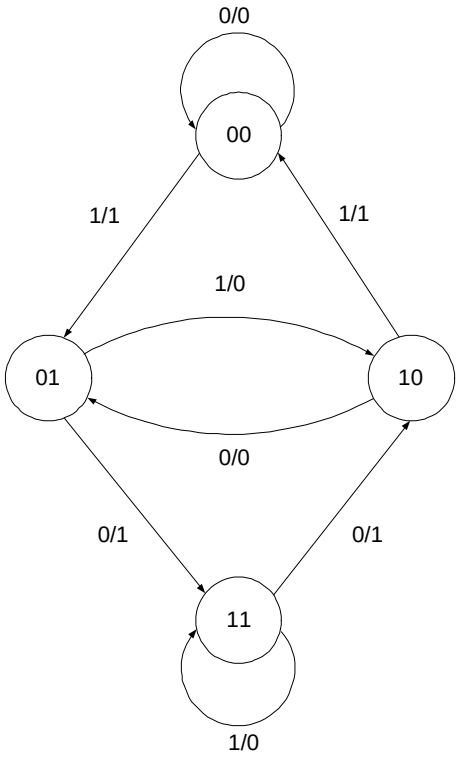

FIG. 2.7 – Diagramme d'état du code convolutif $(1, 5/7)$.

Une détermination récursive de $\alpha_t(m)$ et $\beta_t(m)$ est obtenue en exploitant la représentation en treillis [25]. Pour $t = 1, 2, \ldots, \tau$

$$\alpha_t(m) = \sum_{m'=0}^{M-1} \alpha_{t-1}(m')\gamma_t(m', m),$$

avec l'initialisation

$$\begin{aligned} \alpha_0(0) &= 1 \\ \alpha_0(m) &= 0, \quad m \neq 0. \end{aligned}$$

De même, pour $t = 1, 2, \ldots, \tau - 1$

$$\beta_t(m) = \sum_{m'=0}^{M-1} \beta_{t+1}(m')\gamma_{t+1}(m, m'),$$

avec la condition aux limites

$$\begin{aligned} \beta_\tau(0) &= 1 \\ \beta_\tau(m) &= 0, \quad m \neq 0. \end{aligned}$$

Ensuite, par le théorème de Bayes, la métrique de branche $\gamma_t(m', m)$ est de

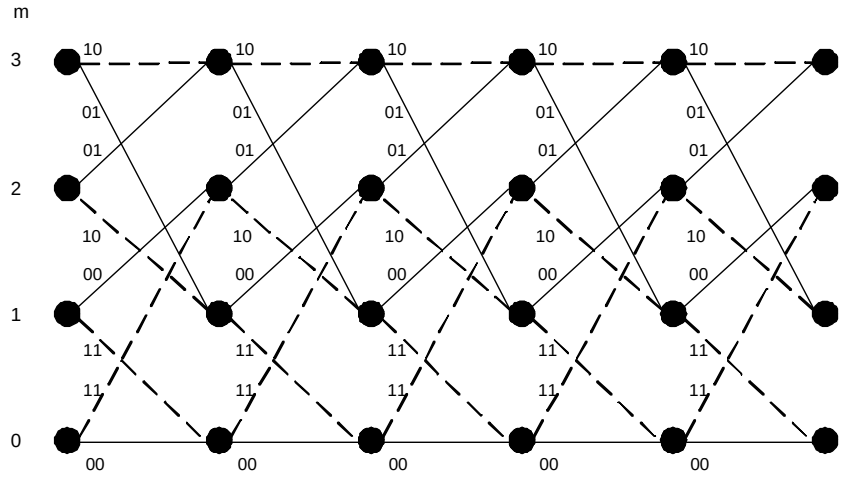

FIG. 2.8 – Représentation en treillis du code convolutif $(1, 5/7)$.

la forme

$$\begin{aligned}\gamma_t(m', m) &= \sum_X P(S_t = m|S_{t-1} = m')P(Y_t|X)P(X_t = X|S_{t-1} = m', S_t = m) \\ &= \sum_X p_t(m|m')q_t(X|m', m)R(Y_t, X),\end{aligned}$$

où $p_t(m|m')$ représente la probabilité *a priori* d'une transition de l'état m' vers l'état m à l'instant t, $q_t(X|m', m)$ vaut un si la transition est valide et zéro sinon, finalement, $R(Y_t, X)$ est la probabilité de transition du canal.

Il est maintenant évident d'appliquer cet algorithme au calcul du rapport de vraisemblance logarithmique *a posteriori* d'un digit x_t présent à l'instant t. Soient $B_t^{(0)}$ et $B_t^{(1)}$ l'ensemble des branches dans le treillis à l'instant t correspondant à $x_t = 0$ et $x_t = 1$, respectivement. Le rapport de vraisemblance logarithmique désiré est obtenu en calculant des probabilités marginales sur toutes les transitions

$$\ln \frac{P(x_t = 0|\mathbf{Y}_1^\tau)}{P(x_t = 1|\mathbf{Y}_1^\tau)} = \ln \frac{\sum_{(m,m')\in B_t^{(0)}} \sigma_t(m', m)}{\sum_{(m,m')\in B_t^{(1)}} \sigma_t(m', m)}.$$

Notons que l'algorithme BCJR peut aussi être appliqué au décodage SISO des codes en bloc. Ceci est dû au fait que tout code en bloc admet une représentation en treillis [36]-[38]. Cependant, la complexité de cette approche est en général trop grande pour des applications pratiques.

Nous pouvons maintenant résumer les opérations d'un décodeur de type BCJR

1. Précalculer les métriques de branche $\gamma_t(m', m)$ à partir des observations *a priori* et du canal.
2. Calcul récursif ascendant des métriques de branche $\alpha_t(m)$.

3. Calcul récursif descendant des métriques de branche $\beta_t(m)$.
4. Calcul des rapports de vraisemblance logarithmiques *a posteriori* bit à bit par le calcul de probabilités marginales sur toutes les transitions dans la représentation en treillis du code.

2.4.2 Algorithme SOVA

L'algorithme de Viterbi à sorties pondérées ou "soft-output Viterbi algorithm" (SOVA) calcule de bonnes approximations des rapports de vraisemblance logarithmiques bit à bit lorsqu'une représentation en treillis efficace d'un code est disponible. L'idée générale consiste à modifier l'algorithme de Viterbi classique [39] en associant à chaque décision ferme la fiabilité de cette décision [40]-[43]. Comparé à l'algorithme BCJR, l'algorithme SOVA produit des sorties pondérées avec une complexité plus faible et une dégradation acceptable des performances en terme de taux d'erreur binaire. Nous utiliserons les mêmes hypothèses et notations que dans la Sec. 2.4.1.

Chaque chemin dans le treillis, representé par une suite d'états $\mathbf{S}_1^\tau$ débutant en $S_0 = 0$ et finissant en $S_\tau = 0$, correspond à un unique mot de code $\mathbf{X}_1^\tau$. Puisque le canal est sans mémoire, il est utile de considérer pour chaque transition dans le treillis à l'instant t

- la métrique de branche $\gamma_t(m', m) = \ln P(S_t = m, Y_t | S_{t-1} = m')$
- la métrique d'état cumulée $M_t(m) = \max_{\mathbf{X}_1^t} \ln P(\mathbf{X}_1^t | \mathbf{Y}_1^t)$, où la maximisation porte sur tous les chemins dans le treillis finissant à l'état m à l'instant t.

Par rapport à l'algorithme BCJR, la métrique de branche est la même et la métrique d'état cumulée est similaire à $\alpha_t(m)$. Cependant, tous les chemins dans le treillis finissant à l'état m à l'instant t sont pris en considération pour le calcul de $\alpha_t(m)$, tandis que seulement le chemin à vraisemblance maximale est retenu dans le calcul de $M_t(m)$.

L'idée de base de l'algorithme de Viterbi est de réaliser le calcul de $M_t(m)$ de manière récursive en éliminant à chaque noeud du treillis le chemin incident ayant la log-vraisemblance la plus faible

$$M_t(m) = \max_{m'} \{M_{t-1}(m') + \gamma_t(m', m)\},$$

avec l'initialisation

$$\begin{aligned} M_0(0) &= 0 \\ M_0(m) &= -\infty, \quad m \neq 0. \end{aligned}$$

La méthode découle simplement du fait que les log-vraisemblances sont additives sur un canal sans mémoire. Chaque noeud dans le treillis retient aussi laquelle des branches incidentes correspond au chemin survivant. Si chaque état a deux branches incidentes, comme c'est le cas pour les codes convolutifs binaires, le SOVA stocke en sus la fiabilité de la décision prise en chaque noeud du treillis

$$\Delta_t(m) = \max_{m'}\{M_{t-1}(m') + \gamma_t(m', m)\} - \min_{m'}\{M_{t-1}(m') + \gamma_t(m', m)\} \geq 0.$$

Une fois que cette procédure est executée pour tous les indices temporels $t = 0, \ldots, \tau$, l'algorithme de Viterbi classique trouve le mot de code à vraisemblable maximale $\hat{\mathbf{X}}_1^\tau$ en remontant la séquence des états les plus vraisemblables $\hat{\mathbf{S}}_1^\tau$ en commençant à $S_\tau = 0$.

En définissant la métrique d'un mot de code $\mathbf{X}_1^\tau$ par

$$M(\mathbf{X}_1^\tau) = \ln P(\mathbf{X}_1^\tau | \mathbf{Y}_1^\tau),$$

nous avons

$$M(\hat{\mathbf{X}}_1^\tau) = M_\tau(0).$$

Le SOVA complète les opérations réalisées par l'algorithme de Viterbi classique en calculant des sorties pondérées de la manière suivante. En supposant que nous devons calculer le rapport de vraisemblance logarithmique d'un digit x_t présent à l'instant t

$$L(x_t|\mathbf{Y}_1^\tau) = \ln \frac{P(x_t = 0|\mathbf{Y}_1^\tau)}{P(x_t = 1|\mathbf{Y}_1^\tau)} = \ln \frac{\sum_{\mathbf{X}_1^\tau : x_t = 0} e^{M(\mathbf{X}_1^\tau)}}{\sum_{\mathbf{X}_1^\tau : x_t = 1} e^{M(\mathbf{X}_1^\tau)}}. \tag{2.4.14}$$

Typiquement, un terme domine la somme au numérateur et au dénominateur de (2.4.14). C'est pourquoi, une bonne approximation est obtenue aux rapports signal sur bruit modérés à élevés en utilisant

$$L(x_t|\mathbf{Y}_1^\tau) \approx \max_{\mathbf{X}_1^\tau : x_t = 0} M(\mathbf{X}_1^\tau) - \max_{\mathbf{X}_1^\tau : x_t = 1} M(\mathbf{X}_1^\tau). \tag{2.4.15}$$

Les maximisations dans (2.4.15) font intervenir le mot de code à vraisemblance maximale et le mot de code concurrent, qui correspond au mot de code de plus grande métrique tel que ce mot de code soit en désaccord avec le mot de code à vraisemblance maximale à propos de la valeur du digit x_t. Il est évident que le mot de code à vraisemblance maximale impose le signe de $L(x_t|\mathbf{Y}_1^\tau)$ dans (2.4.15) et par conséquent la valeur de la décision ferme $\hat{x_t}$. Avec une forte probabilité, le mot de code concurrent est l'un des chemins éliminés qui rejoint en quelque

point le chemin à vraisemblance maximale. Il s'ensuit que

$$L(x_t|\mathbf{Y}_1^\tau) \approx (1-2\hat{x}_t) \min_{\hat{S}_t \in E} \Delta(\hat{S}_t),$$

où E est le sous-ensemble de $\{\hat{S}_t, t = 0, \ldots, \tau\}$ contenant les états en lesquels un chemin vérifiant $x_t \neq \hat{x}_t$ rejoint le chemin à vraisemblance maximale. Cette situation est illustrée par la Fig. 2.9 où le chemin à vraisemblance maximale est le chemin tout-zéro et un chemin concurrent rejoint le chemin à vraisemblance maximale à l'instant t. Le code est le même que celui déjà utilisé dans l'Ex. 2.4.1. Les transitions de poids informatif zéro et un sont représentées par des lignes continues et en pointillé, respectivement. Supposons que nous voulions calculer le rapport de vraisemblance logarithmique du bit informatif à l'instant $t-4$, alors $\hat{S}_t = 0$ fait partie de E et le chemin concurrent est en désaccord en ce qui concerne la valeur de ce bit. Cependant, si nous voulions calculer le rapport de vraisemblance logarithmique du bit informatif à l'instant $t-2$, alors $\hat{S}_t = 0$ ne fait pas partie de E. Dans la pratique, pour savoir si un état fait partie de E ou non, il est nécéssaire de remonter la suite des états correspondant au chemin concurrent.

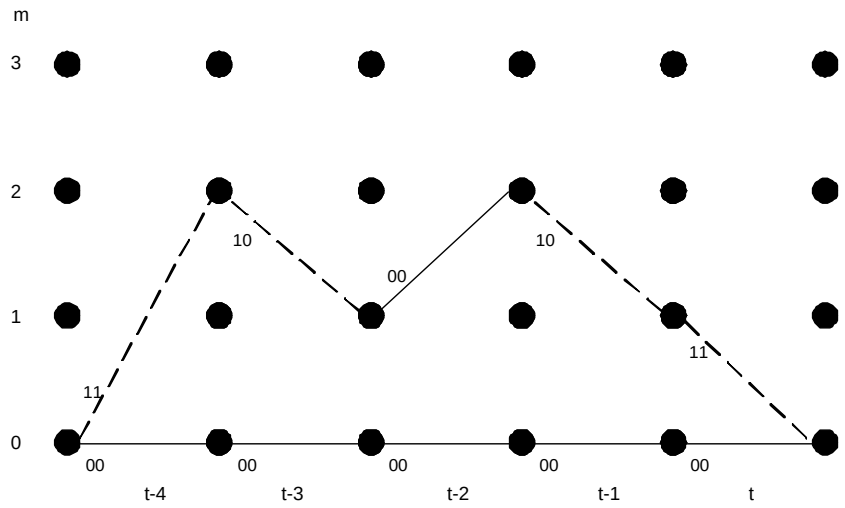

FIG. 2.9 – Exemple de chemin concurrent pour le code convolutif $(1, 5/7)$.

Nous pouvons maintenant résumer brièvement les opérations réalisées par un décodeur de type SOVA :

1. Précalculer les métriques de branche $\gamma_t(m', m)$ à partir des observations *a priori* et du canal.
2. Calcul récursif ascendant des métriques d'état cumulées $M_t(m)$ et des fiabilités $\Delta_t(m)$.
3. Remonter le chemin à vraisemblance maximale.

4. Calcul des rapports de vraisemblance logarithmiques *a posteriori* bit à bit en trouvant le chemin concurrent de plus grande métrique.

2.4.3 Algorithme de Chase à Sorties Pondérées

De même que le SOVA, l'algorithme de Chase à sorties pondérées modifie un algorithme de décodage à sorties dures bien connu dans le but de fournir une approximation des rapports de vraisemblance logarithmiques bit à bit. Cet algorithme convient pour les codes en bloc lorsqu'un décodeur algébrique efficace existe. Soient $\mathbf{c} = (c_1, \ldots, c_n)$ un mot de code d'un code en bloc C de longueur n et $\mathbf{y} = (y_1, \ldots, y_n)$ les observations en provenance du canal. La distance minimum du code est désignée par d_{min}. Définissons la métrique $M(\mathbf{c}) = \ln P(\mathbf{c}|\mathbf{y})$, alors les rapports de vraisemblance logarithmiques bit à bit s'écrivent

$$L(c_i|\mathbf{y}) = \ln \frac{\sum_{\mathbf{c}\in C:c_i=0} e^{M(\mathbf{c})}}{\sum_{\mathbf{c}\in C:c_i=1} e^{M(\mathbf{c})}}, \quad i = 1, \ldots, n. \tag{2.4.16}$$

Typiquement, un terme domine la somme au numérateur et au dénominateur de (2.4.16). C'est pourquoi, une bonne approximation est obtenue aux rapports signal sur bruit modérés à élevés en utilisant

$$L(c_i|\mathbf{y}) \approx \max_{\mathbf{c}\in C:c_i=0} M(\mathbf{c}) - \max_{\mathbf{c}\in C:c_i=1} M(\mathbf{c}), \quad i = 1, \ldots, n. \tag{2.4.17}$$

Les maximisations dans (2.4.17) font intervenir le mot de code à vraisemblance maximale, qui correspond au mot de code ayant la plus grande métrique, ainsi que le mot de code concurrent, qui correspond au mot de code de plus grande métrique tel que ce mot de code soit en désaccord avec le mot de code à vraisemblance maximale à propos de la valeur du digit en $i$ème position. Si aucun mot de code concurrent n'est trouvé

$$L(c_i|\mathbf{y}) \triangleq \beta(1 - 2\hat{c}_i), \tag{2.4.18}$$

où β est un paramètre à optimiser par simulation [18]. Il est évident que le mot de code à vraisemblance maximale impose le signe de $L(c_i|\mathbf{y})$ dans (2.4.17) et par conséquent la valeur de la décision ferme $\hat{c}_i$, for $i = 1, \ldots, n$. Parce qu'une maximisation brutale des métriques prenant en compte tout les mots de code de C est en général trop complexe, cette maximisation est réalisée sur un sous-ensemble de C obtenu à partir de l'algorithme de Chase [44].

Les opérations réalisées par l'algorithme de Chase pour trouver une liste réduite de mots de code sont les suivantes :

1. Décision ferme sur les observations en provenance du canal.
2. Création d'une liste de $2^{\lfloor d_{min}/2 \rfloor}$ séquences de test formées par éffacement

des $\lfloor d_{min}/2 \rfloor$ positions les moins fiables parmi les décisions fermes.

3. Décodage algébrique des séquences de test.

2.5 Conclusions

Dans ce chapitre, nous avons rappelé l'état de l'art en ce qui concerne le décodage itératif.

Pour les deux classes principales de codes pouvant être décodés itérativement, à savoir les codes LDPC et les codes concaténés, nous avons présenté la construction du code, les propriétés en terme de distance ainsi que les algorithmes de décodage itératif associés. En particulier, nous avons vu que le décodage itératif des codes LDPC est basé sur la "règle de la tanh" et que le décodage itératif des codes concaténés repose sur l'algorithme BCJR et ses versions sous-optimales, à savoir l'algorithme SOVA et l'algorithme de Chase à sorties pondérées.

Deuxième partie

Analyse des Systèmes de Décodage Itératif

Chapitre 3

Phénomènes Non-linéaires dans les Systèmes de Décodage Itératif

D'importants travaux expérimentaux ont montré que des performances de décodage proches de la limite de Shannon peuvent être atteintes grâce au décodage itératif [16],[8],[18]. Les simulations ont montré que le comportement de l'algorithme peut être classé en trois régions distinctes :

- Pour de faibles valeurs de RSB, les rapports de vraisemblance logarithmiques extrinsèques, après un certain nombre d'itérations, convergent souvent vers des valeurs correspondant à un grand nombre de décisions fermes incorrectes.
- Pour des valeurs élevées du RSB, les rapports de vraisemblance logarithmiques extrinsèques, après un certain nombre d'itérations, convergent souvent vers des valeurs correspondant à des décisions fermes en majorité correctes. Cependant, la courbe de taux d'erreur binaire correspondante atteint un *plancher* qui décroît lentement avec le RSB.
- La transition entre les régions mentionnées précédemment est appelée la *région de chute* parce qu'après un certain nombre d'itérations, la courbe de taux d'erreur binaire correspondante décroît rapidement avec le RSB.

Jusqu'à un passé récent, il n'existait pas de modèle analytique pour expliquer ce comportement. Dans un article pionnier, Richardson [45] a montré comment décrire le décodage itératif en tant que système dynamique discret avec un grand nombre de dimensions. En étudiant la dynamique non-linéaire des algorithmes de décodage itératif, on montre l'existence de toute une gamme de phénomènes tels que les points fixes, les orbites périodiques, les bifurcations [46] et le chaos [49].

3.1 Concepts de Base des Systèmes Dynamiques

Considérons un fonction non-linéaire $g(\mathbf{x}, \boldsymbol{\mu})$, ayant pour variables $\mathbf{x} \in \mathbb{R}^n$ ainsi que pour paramètres de contrôle $\boldsymbol{\mu} \in \mathbb{R}^p$, et ses itérées $g^i(\mathbf{x}, \boldsymbol{\mu})$, pour $i \geq 0$. Etant donné une condition initiale $\mathbf{x}_0$, la trajectoire associée ou orbite est l'ensemble des points $\{g^i(\mathbf{x}_0, \boldsymbol{\mu}), i \geq 0\}$.

Si toute trajectoire commençant dans $A \subset \mathbb{R}^n$ reste dans A, A est un *ensemble invariant*. De plus, le *bassin d'attraction* d'un ensemble invariant est défini comme l'ensemble ouvert des conditions initiales $\mathbf{x}_0$ telles que les trajectoires commençant en $\mathbf{x}_0$ convergent finalement vers cet ensemble invariant. Nous allons rencontrer trois types d'ensembles invariant : les points fixes, les orbites périodiques et les ensembles invariants chaotiques. Un *point fixe* $\mathbf{x}^*$ est defini par $g(\mathbf{x}^*, \boldsymbol{\mu}) = \mathbf{x}^*$. Soit $\boldsymbol{\eta} = \mathbf{x} - \mathbf{x}^*$ une petite perturbation autour du point fixe, par linéarisation autour de $\mathbf{x}^*$ nous obtenons

$$g(\boldsymbol{\eta} + \mathbf{x}^*, \boldsymbol{\mu}) = g(\mathbf{x}^*, \boldsymbol{\mu}) + \mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})\boldsymbol{\eta} + O(|\boldsymbol{\eta}|^2),$$

où $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})$ est le jacobien $g(\mathbf{x}, \boldsymbol{\mu})$ en $\mathbf{x} = \mathbf{x}^*$. Si $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu}) \neq 0$, le terme $O(|\boldsymbol{\eta}|^2)$ est négligeable et la perturbation devient

$$\boldsymbol{\eta'} \approx \mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})\boldsymbol{\eta}.$$

Les valeurs propres de $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})$ déterminent la stabilité du point fixe $\mathbf{x}^*$, c'est pourquoi $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})$ est aussi appelée la *matrice de stabilité*. Un point fixe $\mathbf{x}^*$ est dit hyperbolique si $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})$ n'a pas de valeur propre sur le cercle unité . Si toutes les valeurs propres de $\mathbf{J}_g(\mathbf{x}^*, \boldsymbol{\mu})$ sont à l'intérieur (resp. à l'extérieur) du cercle unité, le point fixe est un drain (resp. une source). Un point périodique $\mathbf{x}$ de période k vérifie $g^k(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{x}$. La période primaire de $\mathbf{x}$ est la plus petite péride de $\mathbf{x}$. L'ensemble de toutes le itérées d'un point périodique forme une *trajectoire ou orbite périodique*. De même que pour les points fixes, la stabilité d'une trajectoire périodique est obtenue à partir du module des valeurs propres du jacobien de g^k. Finalement, un *ensemble invariant chaotique* correspond à des trajectoires apériodiques sur le long terme et qui dépendent sensiblement des conditions initiales.

Exemple 3.1.1. *Considérons la fonction logistique $g(x, r) = rx(1-x)$ où la variable x et le paramètre r sont des nombre réels [47]. Les Figs. 3.1-3.3 montrent l'évolution des trajectoires $x_n = g^n(x_0, r)$ pour $r = 2.8$, $r = 3.5$ et $r = 3.9$, respectivement. La condition initiale est fixée à $x_0 = 0.1$. Pour $r = 2.8$, la trajectoire converge vers un point fixe, pour $r = 3.5$, la trajectoire converge vers une orbite périodique dont la période primaire vaut 4 et pour $r = 3.9$ la trajectoire*

est chaotique. Pour une fonction de type 1-D, le jacobien se réduit à la dérivée

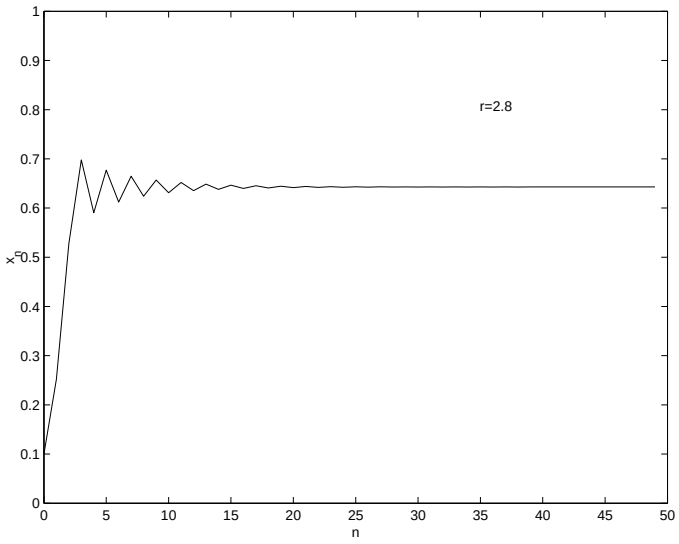

FIG. 3.1 – Itérées de la fonction logistique pour $r = 2.8$ et $x_0 = 0.1$, aboutissant à un point fixe.

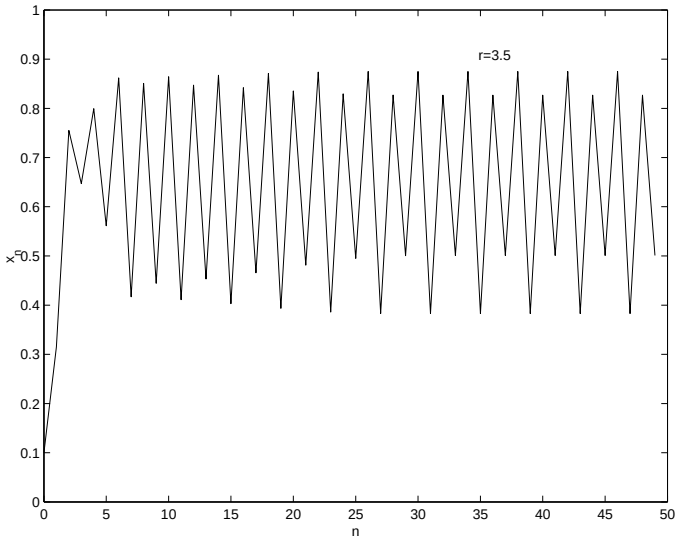

FIG. 3.2 – Itérées de la fonction logistique pour $r = 3.5$ et $x_0 = 0.1$, aboutissant à une orbite périodique de période 4.

de la fonction, $g'(x, r) = r(1 - 2x)$*. En particulier, nous avons* $g'(x, 2.8) < 1$ *si et seulement si* $x > (1 - 1/2.8)/2 \approx 0.32$*. Puisque le point fixe obtenu avec la condition initiale* $x_0 = 0.1$ *lorsque* $r = 2.8$ *est strictement supérieur à* 0.32*, ce point fixe est stable.*

L'exemple précédent à montré dans quelle mesure de faibles variations des paramètres de contrôle d'une fonction peuvent modifier qualitativement les trajectoires. Ce phénomène, connu sous le nom de *bifurcation*, peut se produire de trois manières différentes pour un point fixe [48] :

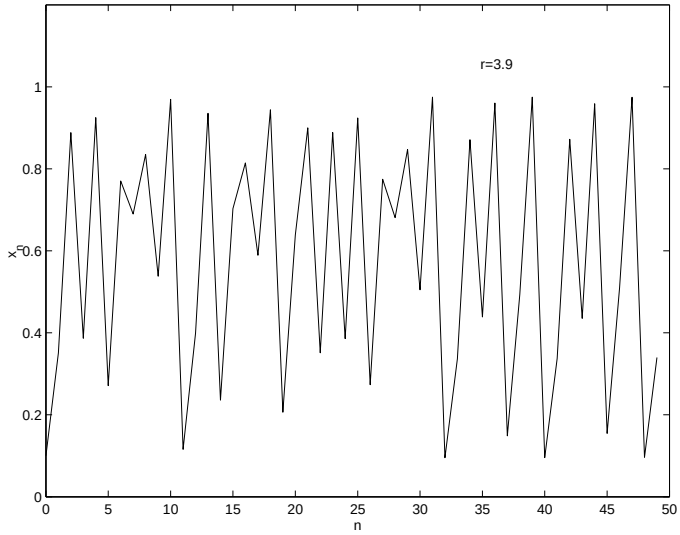

FIG. 3.3 – Itérées de la fonction logistique pour $r = 3.9$ et $x_0 = 0.1$, correspondant à l'existence d'un ensemble invariant chaotique.

- **Bifurcation de Neimark-Sacker :** une paire de valeurs propres complexes conjuguées du jacobien évalué au point fixe, franchissent le cercle unité. Après la bifurcation, le point fixe devient instable et est entouré d'un ensemble invariant fermé stable. Des variations supplémentaires dans les valeurs des paramètres peuvent entraîner une bifurcation de cet ensemble invariant.
- **Bifurcation à doublement de période :** une valeur propre réelle traverse le cercle unité en -1. Après la bifurcation, le point fixe devient instable et une orbite asymptotiquement stable de période 2 apparaît dans son voisinage. Des variations supplémentaires dans les valeurs des paramètres peuvent entraîner une bifurcation de cette orbite de période 2.
- **Bifurcation tangente :** une valeur propre réelle traverse le cercle unité en $+1$. Après la bifurcation, le point fixe disparaît sans donner naissance à un ensemble invariant dans son voisinage.

3.2 Description du Décodage Itératif en tant que Fonction Non-linéaire

Dans cette section, nous allons décrire l'algorithme de turbo décodage en tant que fonction non-linéaire à instants discrets. Par simplicité, nous nous restreignons à la formulation d'origine de Richardson [45] pour une concaténation parallèle de codes convolutifs récursifs systématiques (RSC). Soient $\mathbf{b} = (b_1, \ldots, b_n)$, $\mathbf{c}_1 = (c_{1,1}, \ldots, c_{1,n})$ et $\mathbf{c}_2 = (c_{2,1}, \ldots, c_{2,n})$ la séquence des bits d'information, les

bits de parité générés par le premier encodeur RSC et les bits de parité générés par le second encodeur RSC, respectivement. Puisqu'une permutation de longueur n est utilisée pour entrelacer $\mathbf{b}$ avant de générer $\mathbf{c}_2$, cette permutation fait partie des paramètres de la fonction décrivant l'algorithme de turbo décodage. Les observations en provenance du canal correspondant à $\mathbf{b}$, $\mathbf{c}_1$ et $\mathbf{c}_2$ sont désignées par $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y}_1 = (y_{1,1}, \ldots, y_{1,n})$ et $\mathbf{y}_2 = (y_{2,1}, \ldots, y_{2,n})$, respectivement. En notant $\mathbf{b}_0$ la séquence informative tout-zéro, nous définissont les *log-densités normalisées* suivantes

$$\begin{aligned} P_0(\mathbf{b}) &= \ln p(\mathbf{x}|\mathbf{b}) - \ln p(\mathbf{x}|\mathbf{b}_0) \\ P_1(\mathbf{b}) &= \ln p(\mathbf{y}_1|\mathbf{b}) - \ln p(\mathbf{y}_1|\mathbf{b}_0) \\ P_2(\mathbf{b}) &= \ln p(\mathbf{y}_2|\mathbf{b}) - \ln p(\mathbf{y}_2|\mathbf{b}_0). \end{aligned}$$

Exemple 3.2.1. *Supposons que le canal soit le canal gaussien à entrée binaire où un zéro binaire est transmis en tant que $+1$ et un un binaire est transmis en tant que -1. L'écart-type du bruit gaussien est noté σ and* $\mathrm{wt}(.)$ *désigne le poids de Hamming d'une séquence binaire. Il s'ensuit que*

$$\begin{aligned} p(\mathbf{x}|\mathbf{b}) &= (\sqrt{2\pi\sigma^2})^{-n} \exp\left(-\frac{\sum_{i=1}^{n}(x_i - (1-2b_i))^2}{2\sigma^2}\right) \\ p(\mathbf{x}|\mathbf{b}_0) &= (\sqrt{2\pi\sigma^2})^{-n} \exp\left(-\frac{\sum_{i=1}^{n}(x_i - 1)^2}{2\sigma^2}\right). \end{aligned}$$

Par conséquent

$$P_0(\mathbf{b}) = -\frac{2}{\sigma^2} \sum_{i:b_i \neq 0} x_i.$$

Lorsque la séquence informative est $\mathbf{b}_0$, les séquences $\mathbf{c}_1$ et $\mathbf{c}_2$ correspondantes sont aussi de type tout-zéro, c'est pourquoi

$$\begin{aligned} P_1(\mathbf{b}) &= -\frac{2}{\sigma^2} \sum_{i:c_{1,i} \neq 0} y_{1,i} \\ P_2(\mathbf{b}) &= -\frac{2}{\sigma^2} \sum_{i:c_{2,i} \neq 0} y_{2,i}. \end{aligned}$$

En prenant en compte le fait que le canal est sans mémoire, nous obtenons [46]

$$\begin{aligned} P_0(\mathbf{b}) &= \mathcal{N}\left(-\frac{2}{\sigma^2}\,\mathrm{wt}(\mathbf{b}), \frac{4}{\sigma^2}\,\mathrm{wt}(\mathbf{b})\right) \\ P_1(\mathbf{b}) &= \mathcal{N}\left(-\frac{2}{\sigma^2}\,\mathrm{wt}(\mathbf{c}_1), \frac{4}{\sigma^2}\,\mathrm{wt}(\mathbf{c}_1)\right) \\ P_2(\mathbf{b}) &= \mathcal{N}\left(-\frac{2}{\sigma^2}\,\mathrm{wt}(\mathbf{c}_2), \frac{4}{\sigma^2}\,\mathrm{wt}(\mathbf{c}_2)\right), \end{aligned}$$

où $\mathcal{N}(m, v)$ désigne une densité de probabilité gaussienne de moyenne m et de variance v.

Soit H_i (resp. H_i^c) l'ensemble de toutes les séquences binaires dont la $i$ème position est un un (resp. un zéro). Définissons les vecteurs colonne

$$\begin{aligned}
\mathbf{b}_0 &= (0, \ldots, 0)^T, \\
\mathbf{b}_1 &= (1, 0 \ldots, 0)^T, \\
\mathbf{b}_2 &= (0, 1, 0 \ldots, 0)^T, \ldots \\
\mathbf{b}_n &= (0, 0 \ldots, 0, 1)^T.
\end{aligned}$$

Une densité p est une *densité produit* normalisée si et seulement si

$$p(\mathbf{b}) = \prod_{b_i \neq 0} p(\mathbf{b}_i),$$

où

$$p(\mathbf{b}_i) = \frac{\sum_{\mathbf{b} \in H_i} p(\mathbf{b})}{\sum_{\mathbf{b} \in H_i^c} p(\mathbf{b})}$$

représente le rapport de vraisemblance logarithmique du $i$ème bit. La *log-densité produit* correspondante vaut

$$P(\mathbf{b}) = \sum_{b_i \neq 0} P(\mathbf{b}_i),$$

où

$$P(\mathbf{b}_i) = \ln \frac{\sum_{\mathbf{b} \in H_i} p(\mathbf{b})}{\sum_{\mathbf{b} \in H_i^c} p(\mathbf{b})}$$

représente le rapport de vraisemblance logarithmique du $i$ème bit. On s'aperçoit qu'une densité produit est entièrement définie par la connaissance des probabilités marginales bit à bit. En particulier on voit que les log-densités P_0, P_1 et P_2 de l'Ex. 3.2.1 sont des densités produits.

Etant donné une log-densité normalisée $P = \ln p$, il existe une unique log-densité produit $\pi(P)$ ayant les mêmes marginales bit à bit que P. $\pi(P)$ est entièrement défini par la connaissance de

$$\pi(P)(\mathbf{b}_i) = \ln \frac{\sum_{\mathbf{b} \in H_i} p(\mathbf{b})}{\sum_{\mathbf{b} \in H_i^c} p(\mathbf{b})}, \quad i = 1, \ldots, n$$

Le jacobien de $\pi(P)$ est la matrice $\mathbf{J}_P$ de dimension $(n \times n)$ mesurant la dépendance de la vraisemblance du bit j par rapport au bit i, pour i et j allant de 1 à n. Les

entrées du jacobien sont données par [45]

$$
\begin{aligned}
(\mathbf{J}_P)_{ij} &= \frac{\sum_{\mathbf{b}\in H_i\cap H_j} p(\mathbf{b})}{\sum_{\mathbf{b}\in H_i} p(\mathbf{b})} - \frac{\sum_{\mathbf{b}\in H_i^c\cap H_j} p(\mathbf{b})}{\sum_{\mathbf{b}\in H_i^c} p(\mathbf{b})}, \quad \text{si } i \neq j \\
(\mathbf{J}_P)_{ij} &= 1, \quad \text{si } i = j,
\end{aligned}
\tag{3.2.1}
$$

En particulier si P est une densité produit, $\mathbf{J}_P = \mathbf{I}$, où $\mathbf{I}$ désigne la matrice identité de dimension $(n \times n)$.

Maintenant, définissons les log-densités produits Q_1 et Q_2 correspondant à l'information extrinsèque à la sortie du premier et du second décodeur constitutif, respectivement ; nous pouvons décrire l'algorithme de turbo décodage de la Sec. 2.3.3 par la récursion

$$
\begin{aligned}
Q_1 &\leftarrow \pi(P_0 + P_1 + Q_2) - (P_0 + Q_2) \\
Q_2 &\leftarrow \pi(P_0 + P_2 + Q_1) - (P_0 + Q_1),
\end{aligned}
\tag{3.2.2}
$$

avec pour initialisation $Q_2 = 0$. Dans (3.2.2), la première équation correspond à la mise à jour par le premier décodeur constitutif des rapports de vraisemblance logarithmiques associés aux bits informatifs et jouant le rôle de n variables. Les observations en provenance du canal correspondant aux bits informatifs et aux bits de parité du premier encodeur jouent le rôle de $2n$ paramètres. De même, la seconde équation dans (3.2.2) correspond à la mise à jour par le second décodeur constitutif des rapports de vraisemblance logarithmiques associés aux bits informatifs et jouant le rôle de n variables. Les observations en provenance du canal correspondant aux bits informatifs et aux bits de parité du second encodeur jouent le rôle de $2n$ paramètres. Par conséquent, nous pouvons considérer l'algorithme de turbo décodage comme un système dynamique en *boucle fermée* à n variables and $3n$ paramètres, où le premier et le second décodeur constitutif en cascade jouent le rôle du système en *boucle ouverte*. La matrice de stabilité du système dynamique en boucle fermée est donée par [45]

$$
(\mathbf{J}_{P_0+P_2+Q_1} - \mathbf{I})(\mathbf{J}_{P_0+P_1+Q_2} - \mathbf{I}). \tag{3.2.3}
$$

Nous résumons maintenant les résultats principaux obtenus à partir de cette description :

1. A RSB asymptotiquement faible, l'algorithme de turbo décodage possède avec une grande probabilité un point fixe unique appelé point fixe *indécisif*, qui correspond à de nombreuses décisions erronées sur les bits informatifs [46].
2. A RSB asymptotiquement élevé, l'algorithme de turbo décodage possède

avec une grande probabilité des points fixes appelés points fixes *non-équivoques*, qui correspondent à des décisions correctes sur les bits informatifs. De plus, l'algorithme ne peut converger que vers un seul de ces points fixes [46].

3. La proximité du turbo décodeur au décodeur du maximum de vraisemblance peut être obtenue en calculant des formules approchées de la différence des vraisemblances fournies par ces décodeurs [45].

3.3 Dynamique Non-linéaire du Décodage Itératif

3.3.1 Le Décodage Itératif en tant que Fonction Non-linéaire à un Paramètre

La Sec. 3.2 a montré que l'algorithme de turbo décodage est un système dynamique complexe avec un grand nombre de variables et de paramètres. Afin d'étudier la dynamique du système, il est souhaitable de simplifier le modèle.

Supposons que les échantillons de bruit correspondant aux observations en provenance du canal $\mathbf{x}$, $\mathbf{y}_1$ et $\mathbf{y}_2$ soient représentés sous forme vectorielle par $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{3n})$ et que les $3n-1$ rapports $\nu_1/\nu_2, \nu_2/\nu_3, \ldots, \nu_{3n-1}/\nu_{3n}$ soient fixés. Il s'en suit que la séquence de bruit $\boldsymbol{\nu}$ est entièrement définie par la variance d'échantillons

$$\hat{\sigma}^2 = \frac{1}{3n} \sum_{i=1}^{3n} \nu_i^2,$$

qui constitue une bonne approximation de la variance du bruit du canal σ^2, puisque n est typiquement un entier grand. Une première simplification proposée par Agrawal et Vardy [46] consiste à paramétrer le système avec le seul paramètre $\hat{\sigma}$. Ces auteurs sont ensuite à même de simuler des points fixes indécisifs et non-équivoques pour des valeurs pratiques de RSB. En calculant les valeurs propres de la matrice de stabilité donnée par (3.2.3), ils ont mis en évidence que les points fixes indécisifs sont stables pour des valeurs de RSB en-dessous de la région de chute. En même temps que le RSB augmente, le point fixe indécisif bifurque soit en disparaissant soit en devenant instable et en donnant lieu à un ensemble invariant attracteur dans son voisinage. Si le RSB est encore augmenté, ces ensembles invariants bifurquent à leur tour à la fin des la zone de chute et l'algorithme de turbo décodage converge vers un point fixe non-équivoque.

Tasev *et al.* [49] ont proposé de simplifier davantage l'étude de la dynamique en représentant les itérées de l'entropie moyenne des bits d'information, désignée par E. Définissons $p_i^l(0)$ comme la probabilité que le $i$ème bit d'information soit un 0 binaire après l itérations de décodage. L'entropie moyenne des bits

d'information à l'itération l est donnée par

$$E(l) = -\frac{1}{n}\sum_{i=1}^{n} p_i^l(0)\ln p_i^l(0) + (1-p_i^l(0))\ln(1-p_i^l(0)).$$

$E(l)$ donne une mesure de la fiabilité des décisions pour une séquence informative donnée de longueur n. Une grande valeur de l'entropie moyenne indique une ambiguité de l'algorithme de décodage quant à la valeur des bits informatifs. Une faible valeur de l'entropie moyenne indique l'absence d'ambiguité de l'algorithme de décodage quant à la valeur des bits informatifs. Notons que $E \to 0$ ne signifie pas automatiquement que les décisions fermes soient correctes, mais plutôt que l'algorithme de turbo décodage est très confiant quant à la valeur de ces décisions fermes, sachant les observations en provenance du canal. Cependant, les performances étonnamment bonnes du turbo décodage semblent indiquer que dans la plupart des cas, les décisions fermes sont en fait correctes lorsque $E \to 0$. En traçant les itérées de $E(l)$ pour $l \geq 0$, nous obtenons une représentation en 1-D simple des trajectoires de l'algorithme de turbo décodage dans l'intervalle $[0, 1]$. Grâce à cette méthode, il a été montré que l'algorithme de turbo décodage présente des ensembles invariants chaotiques et des régimes transitoires chaotiques dans la zone de chute [49].

Dans la section suivante, nous allons appliquer cette technique aux codes LDPC et aux codes produits. Nous montrerons que les algorithmes de décodage itératif pour ces codes ont des trajectoires qui sont qualitativement similaires à la dynamique obtenue pour la concaténation parallèle de codes convolutifs.

3.3.2 Dynamique du Décodage Itératif des Codes LDPC

Considérons l'ensemble des (n, λ, ρ) codes LDPC de longueur n, de polynôme de distribution du degré des noeuds de variable $\lambda(x)$ et de polynôme de distribution du degré des noeuds de parité $\rho(x)$. Nous tirons un code au hasard dans l'ensemble et nous étudions quelques trajectoires typiques de l'algorithme de décodage décrit dans la Sec. 2.2.3. Plus précisemment, nous traçons l'évolution du nombre d'erreurs binaires dans le bloc. De plus, nous traçons $E(l+1)$, l'entropie moyenne du bloc à l'itération $l+1$ en fonction de $E(l)$ afin d'obtenir une repésentation à 1-D des trajectoires [47]. Nous supposons que le mot de code tout-zéro est transmis sur le canal gaussien à entrée binaire. Les échantillons de bruit gaussien sont contenus dans le vecteur $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n)$. Comme dans [46], l'analyse des bifurcations est réalisée en fixant les $(n-1)$ rapports $\nu_1/\nu_2, \nu_2/\nu_3, \ldots, \nu_{n-1}/\nu_n$.

C'est pourquoi le système est paramétré par un seul paramètre, à savoir

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} \nu_i^2.$$

Notons R le rendement de codage. Nous définissons le RSB par $1/(2R\hat{\sigma}^2)$, ce qui constitue une bonne approximation de E_b/N_0 lorsque n est grand.

Trajectoires pour les Codes LDPC Réguliers

Nous considérons l'ensemble des $(216, 3, 6)$ codes LDPC réguliers. Nous allons montrer que de manière analogue aux concaténations en parallèle de codes convolutifs, les trajectoires de l'algorithme de décodage des code LDPC présentent des points fixes indécisifs, des points fixes non-équivoques, des orbites périodiques, des ensembles invariants chaotiques et des régimes transitoires chaotiques. Ces différents comportements peuvent être obtenus en considérant quelques réalisations du bruit.

Le premier exemple de trajectoire est donné par la Fig. 3.4. La Fig. 3.4 1) montre un point fixe indécisif pour une faible valeur du RSB. Les itérées du nombre d'erreurs binaires et l'entropie moyenne sont tracées pour les itérations $l \geq 940$ afin de s'affranchir du régime transitoire. Notons que l'entropie moyenne du bloc et le nombre d'erreurs binaires baissent avec le RSB parce que les observations en provenance du canal deviennent moins bruitées. La Fig. 3.4 2) est caractéristique d'une *bifurcation de Neimark-Sacker*. Le point fixe indécisif devient instable pour un RSB = 1.20 dB et est entouré par une petite orbite périodique fermée. Observons que ceci n'affecte pas les décisons fermes parce que les variations dans les rapports de vraisemblance logarithmiques ne sont pas suffisamment élevées pour provoquer des changements de signe. La Fig. 3.4 3) montre qui si le RSB est augmenté davantage à 1.44 dB, l'orbite fermée s'agrandit. Il s'ensuit que le nombre d'erreurs binaires prend aussi un comportement périodique parce que des changements de signe se produisent dans les rapports de vraisemblance logarithmiques. Lorsque le RSB est augmenté à 1.45 dB (voir Fig. 3.4 4)), la trajectoire converge vers un point fixe non-équivoque correspondant à une valeur de E proche de 0 et des décisions binaires correctes. Notons la présence d'un régime transitoire chaotique durant les 63 premières itérations, qui indique l'existence d'un ensemble invariant chaotique non-attracteur près du point fixe. Un comportement similaire est observé pour un RSB = 1.52 dB (voir Fig. 3.4 5)), mais la durée du régime transitoire chaotique est réduite à 21 itérations. Des améliorations supplémentaires du RSB réduisent encore plus la durée du régime transitoire chaotique jusqu'à ce qu'il disparaisse complètement.

L'entropie moyenne décroît alors de façon monotone avec le nombre d'itérations. Celà signifie que la taille du bassin d'attraction du point fixe non-équivoque s'accroît lorsque le RSB augmente.

Le comportement de cette trajectoire peut être résumé par la Fig. 3.5. Notons la bifurcation de Neimark-Sacker à 1.20 dB. A 1.44 dB, le cycle limite se déforme juste avant son éclatement à 1.45 dB. Ce phénomène typique est connu sous le nom d'eclatement de tore dans la littérature.

Une seconde trajectoire typique est illustrée par la Fig. 3.6. La Fig. 3.6 1) montre un point fixe indécisif pour un RSB = 0.59 dB. La Fig. 3.4 2) est caractéristique d'une *bifurcation tangente*. Le point fixe indécisif disparaît pour un RSB = 0.60 dB et la trajectoire converge vers une orbite périodique fermée. Le fait que l'orbite fermée soit tangente à la première bissectrice $E(l+1) = E(l)$ est un reste de la bifurcation tangente. Observons que le nombre d'erreurs binaires correspondant reste périodiquement bloqué à 16 durant 57 itérations consécutives, ce qui correspond au même phénomène. Les Fig. 3.6 3) and 4) montrent un exemple typique de route vers le chaos par éclatement de tore, au cours de laquelle une orbite fermée est graduellement transformée en un attracteur chaotique. Finalement, grâce à la Fig. 3.6 5), on voit que lorsque le RSB est augmenté, soit l'attracteur chaotique perd sa stabilité soit la condition initiale ne se situe plus dans le bassin d'attraction de l'ensemble invariant chaotique. La trajectoire converge alors vers un point fixe non-équivoque après un régime transitoire chaotique.

La trajectoire illustrée par la Fig. 3.7 est similaire à celle de la Fig. 3.6, cependant la dynamique est plus complexe à cause de l'apparition d'une *bifurcation à doublement de période*. Pour un RSB de -0.03 dB, la trajectoire converge vers un attracteur chaotique stable. Pour un SNR = 0.86 dB, la trajectoire converge vers un cycle limite stable de période 2. Ce cycle limite est matérialisé par les deux points de la Fig. 3.7 5)b), alternativement visités par les itérées de l'entropie moyenne $E(l)$. Observons que le nombre d'erreurs binaires oscille aussi pour un RSB = 0.86 dB. Le cycle limite de période 2 finit par bifurquer pour un RSB = 0.87 dB et une fois de plus la trajectoire converge vers un point fixe non-équivoque après un régime transitoire chaotique..

Nous observons qu'en général, les trajectoires atteignent un point fixe non-équivoque pour des valeurs de RSB à plus ou moins 1.5 dB autour du seuil de bruit des $(3, 6)$ codes LDPC réguliers de longueur infinie qui vaut 1.1 dB [6].

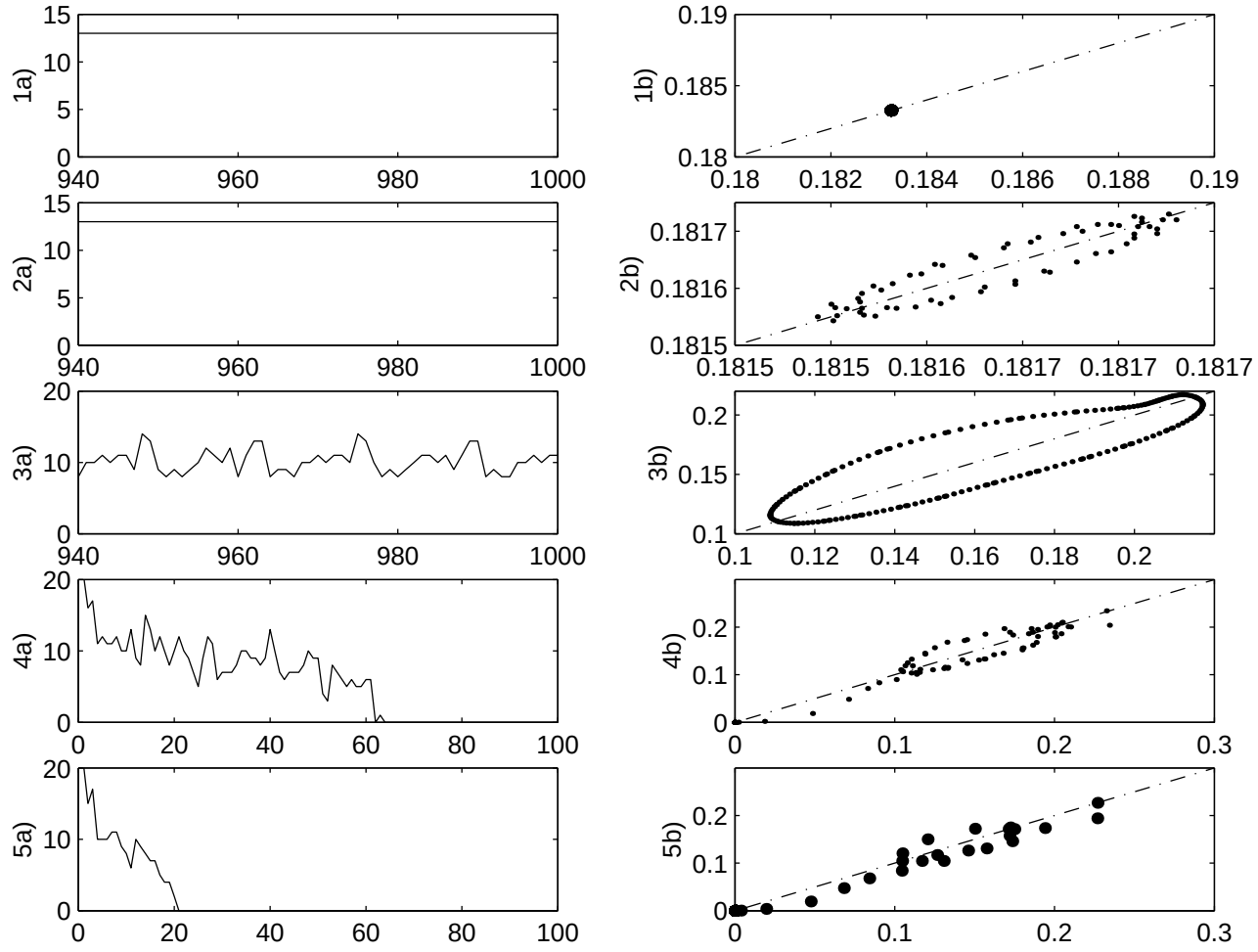

FIG. 3.4 – Trajectoires du décodage itératif d'un $(216, 3, 6)$ code LDPC. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) 1.19 dB, 2) 1.20 dB, 3) 1.44 dB, 4) 1.45 dB, 5) 1.52 dB.

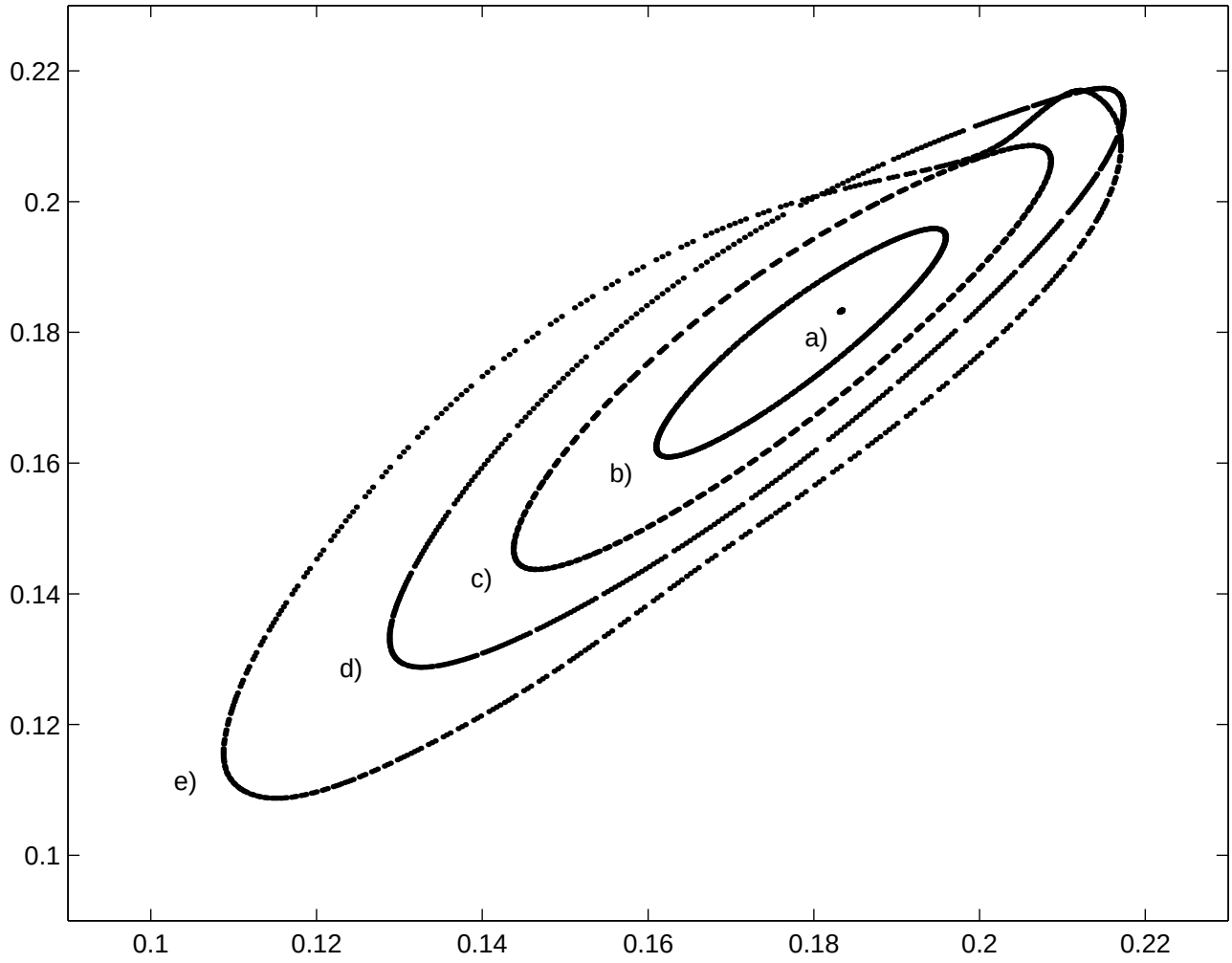

FIG. 3.5 – $E(l+1)$ en fonction de $E(l)$ pour la trajectoire représentée dans la Fig. 3.4. Valeurs du RSB : a) 1.19 dB, b) 1.23 dB, c) 1.27 dB, d) 1.33 dB, e) 1.44 dB.

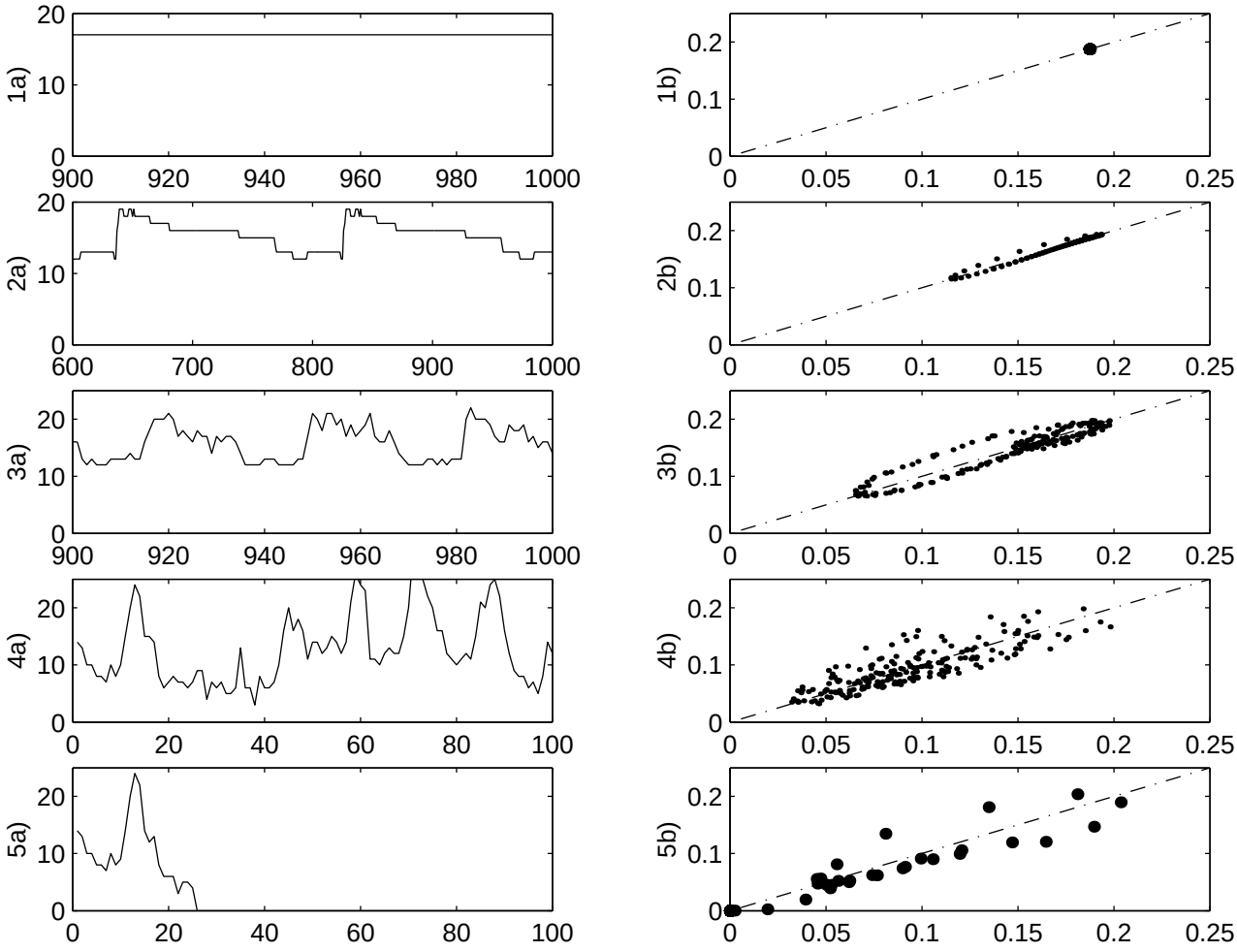

FIG. 3.6 – Trajectoires du décodage itératif d'un $(216, 3, 6)$ code LDPC. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) 0.59 dB, 2) 0.60 dB, 3) 0.76 dB, 4) 1.64 dB, 5) 1.76 dB.

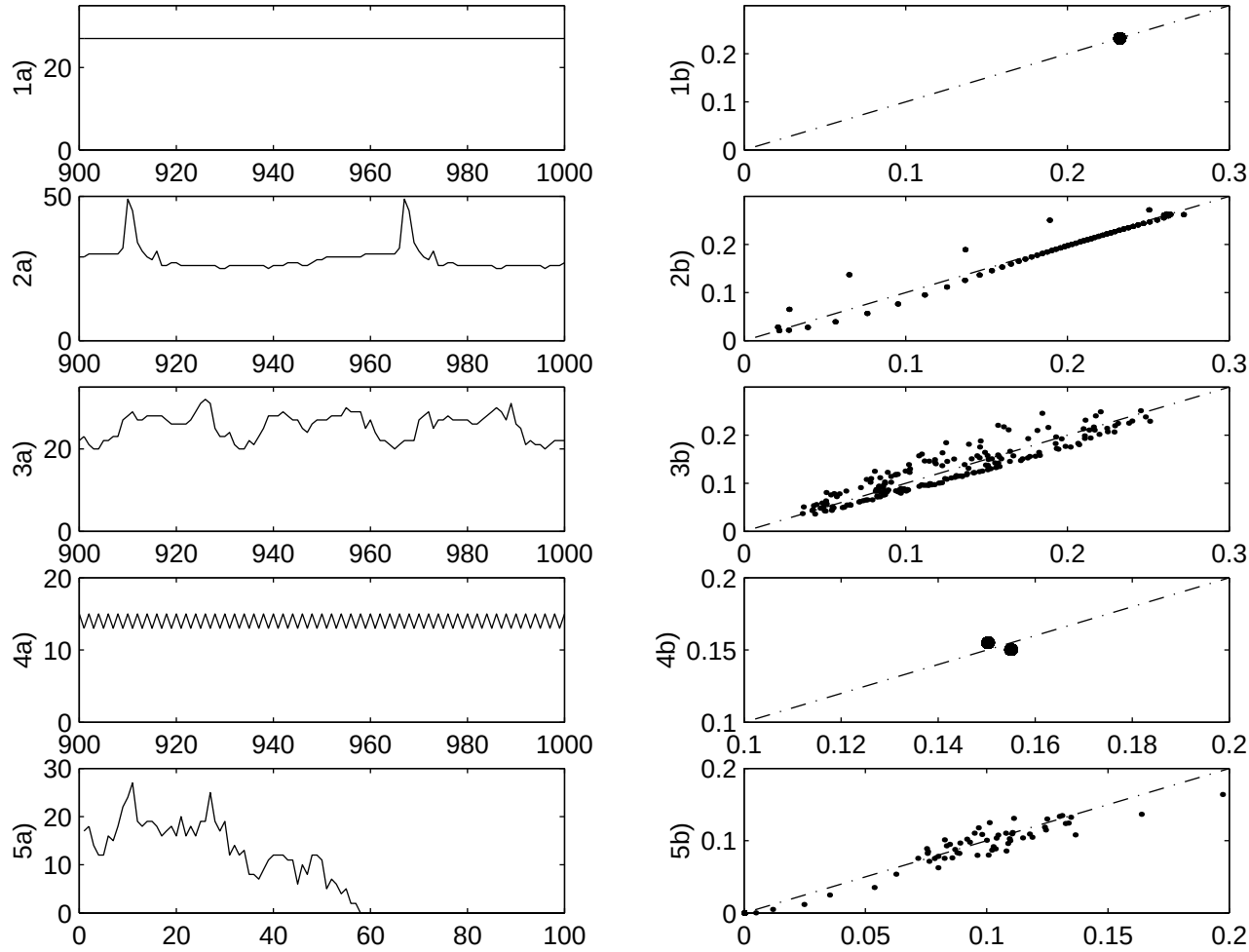

FIG. 3.7 – Trajectoires du décodage itératif d'un $(216, 3, 6)$ code LDPC. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) -1.1 dB, 2) -1.0 dB, 3) -0.03 dB, 4) 0.86 dB, 5) 0.87 dB.

Trajectoires pour les Codes LDPC Irréguliers

Nous considérons l'ensemble des $(2000, \lambda, \rho)$ codes LDPC dont le degré maximal des noeuds de variable est $d_v = 4$ et le degré maximal des noeuds de parité est $d_c = 6$. Les polynômes $\lambda(x) = 0.38354x + 0.04237x^2 + 0.57409x^3$ et $\rho(x) = 0.24123x^4 + 0.75877x^5$ ont été optimisés par évolution de densité et le seuil de bruit correspondant à l'ensemble pour des tailles de bloc infinies vaut $(E_b/N_0)^* = 0.8085$ dB [7].

Nous observons que l'originalité des trajectoires du décodage itératif des codes LDPC irréguliers réside dans le fait qu'au lieu d'un seul point fixe, plusieurs points fixes peuvent apparaître avant qu'un point fixe non-équivoque ne soit atteint. Ceci n'est pas surprenant dans la mesure où l'évolution de densité, qui a été utilisée pour générer les polynômes λ and ρ, exhibe un comportement analogue pour des codes de longueur infinie [7]. La Fig. 3.8 illustre ce phénomène. Les Fig. 3.8 1) et 2) montrent la disparition d'un point fixe indécisif par l'intermédiaire d'une *bifurcation tangente* et la formation d'une orbite périodique fermée. Avec un RSB croissant, une route vers le chaos par éclatement de tore se produit. Soit l'attracteur chaotique perd sa stabilité soit la condition initiale ne se situe plus dans le bassin d'attraction de l'ensemble invariant chaotique pour un SNR = 0.72 dB (voir Fig. 3.8 3) et 4)). La trajectoire converge alors vers un point fixe aux alentours de $E \approx 0.1$. Si le RSB augmente davantage, ce nouveau point fixe descend graduellement vers zéro et la trajectoire correspondante devient tangente à la première bissectrice $E(l+1) = E(l)$ (voir Fig. 3.8 5)). Dans cet exemple, bien qu'un point fixe non-équivoque soit atteint à RSB élevé ($E \to 0$), le nombre d'erreurs binaires atteint un plancher se situant à 6.

3.3.3 Dynamique du Décodage Itératif des Codes Produits

Nous considérons le décodage itératif des codes produits $BCH(32, 26)^2$ et $BCH(64, 51)^2$ sur le canal gaussien à entrée binaire. En employant la même technique que dans la Sec. 3.3.2, nous étudions les trajectoires de l'algorithme de décodage en utilisant comme décodeurs constitutifs ceux décrits dans la Sec. 2.4.3. Nous traçons l'évolution du nombre d'erreurs binaires parmi les bits informatifs. De plus, nous traçons $E(l+1)$, l'entropie moyenne des bits informatif à l'itération $l+1$ en fonction de $E(l)$.

La Fig. 3.9 montre une trajectoire de décodage typique pour le code produit $BCH(32, 26)^2$. A RSB faible, aucun point fixe indécisif n'existe, au contraire la trajectoire est déjà chaotique. Dans la région de chute, pour un RSB = 1.60 dB, les valeurs de l'entropie moyenne atteignent de manière répétée des valeurs

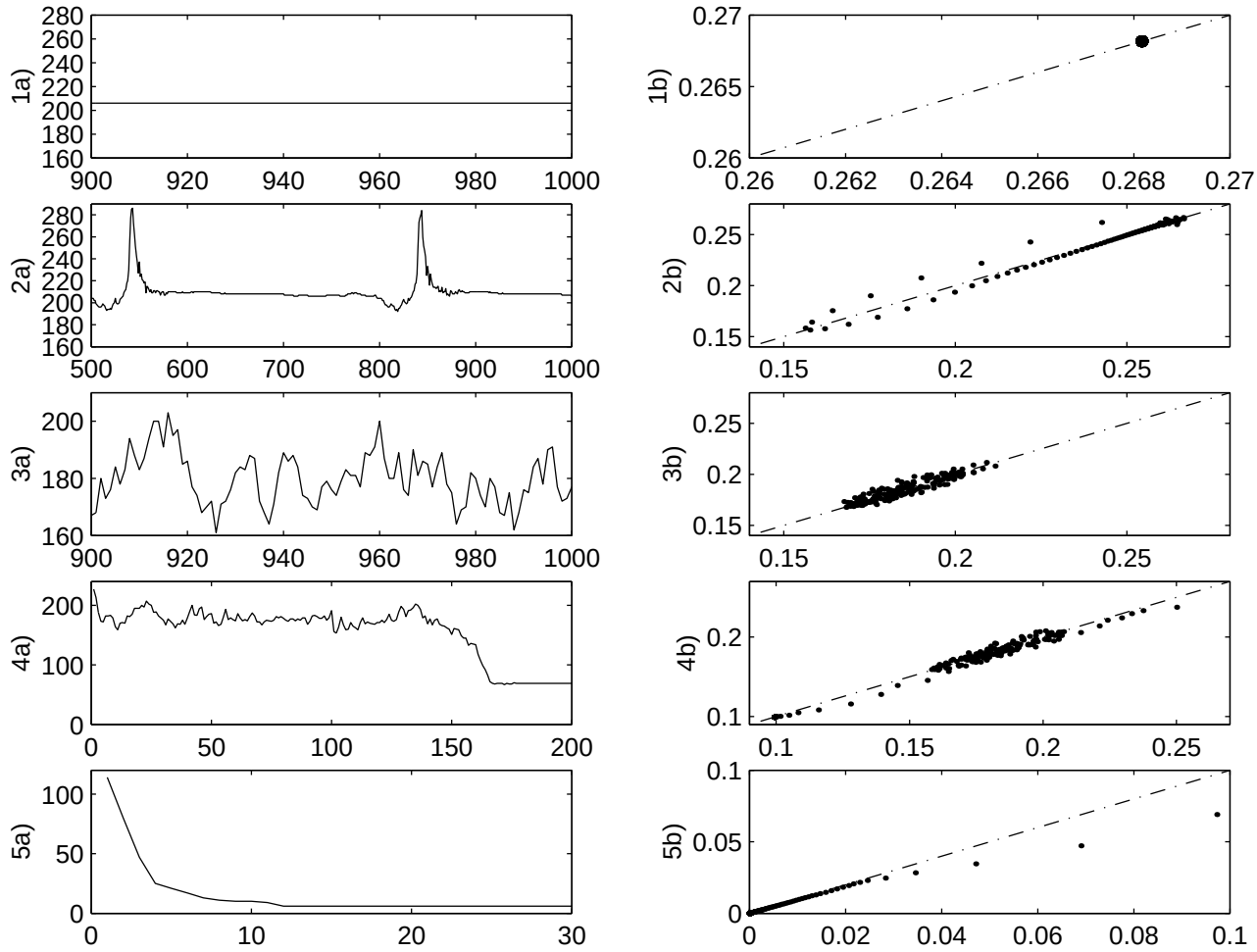

FIG. 3.8 – Trajectoires du décodage itératif d'un $(2000, \lambda, \rho)$ code LDPC irrégulier. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) 0.32 dB, 2) 0.34, dB 3) 0.70 dB, 4) 0.72 dB, 5) 2.48 dB.

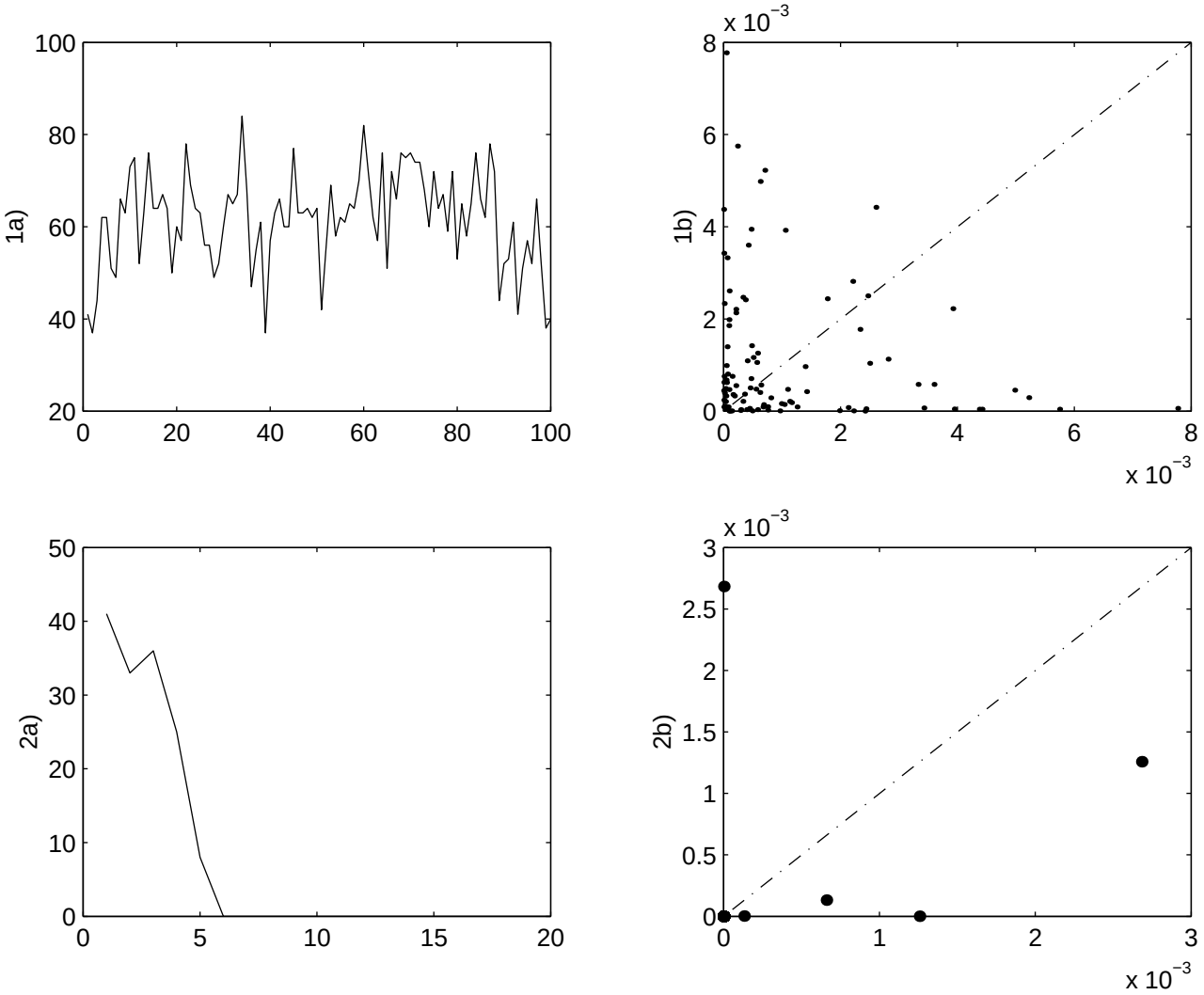

FIG. 3.9 – Trajectoires du décodage itératif du code produit $BCH(32,26)^2$. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) 1.60 dB, 2) 1.61 dB.

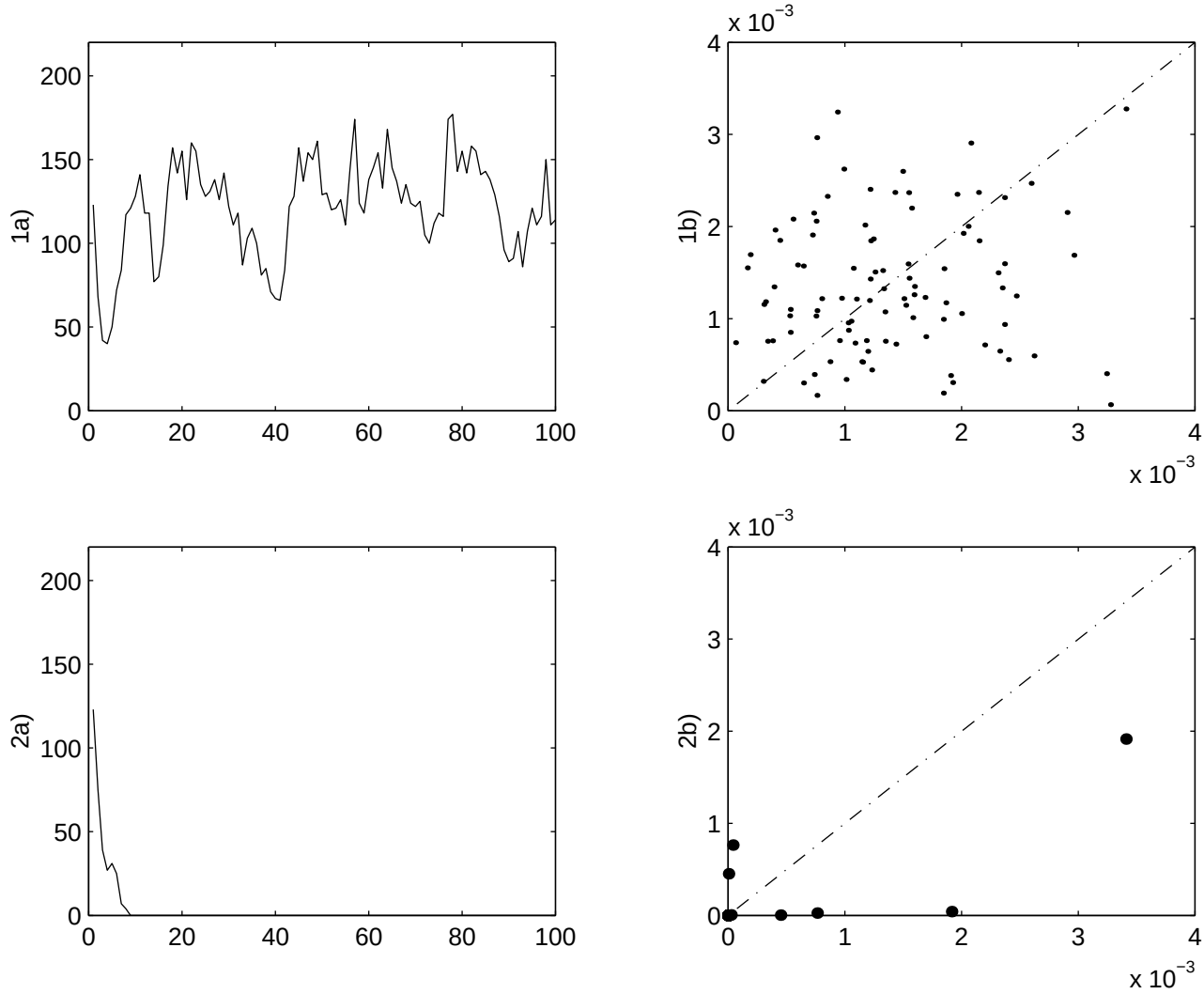

FIG. 3.10 – Trajectoires du décodage itératif du code produit $BCH(64,51)^2$. a) Nombre d'erreurs binaires en fonction de l b) $E(l+1)$ en fonction de $E(l)$. Valeurs du RSB : 1) 2.770 dB, 2) 2.771 dB.

proches de 0 avant de retourner à un comportement chaotique. Finalement, pour un RSB = 1.61 dB, la trajectoire converge après un régime transitoire très court vers un point fixe non-équivoque.

Un comportement similaire est observé dans la Fig. 3.10 pour le code produit $BCH(64, 51)^2$.

3.4 Conclusions

Les systèmes de décodage itératif sont des systèmes dynamiques non-linéaires avec un grand nombre de dimensions et de paramètres. Afin de simplifier l'étude expérimentale de la dynamique, nous utilisons l'évolution de l'entropie moyenne pour obtenir un représentation à 1-D des trajectoires. De plus, en fixant les rapports du bruit, nous obtenons un système paramétré uniquement par la variance d'échantillons du bruit.

De nombreuses simulations montrent qu'en général, les décodeurs itératifs convergent à RSB faible vers un point fixe indécisif correspondant à de nombreuses erreurs. Lorsque le RSB augmente, les points fixes indécisifs subissent une bifurcation de Neimark-Sacker, à doublement de période ou tangente. Si un cycle limite périodique fermé apparaît, il bifurquera aussi si le RSB est augmenté davantage. La trajectoire converge alors soit vers un ensemble invariant chaotique ou directement vers un point fixe non-équivoque correspondant à des décisions en majorité correctes après un régime transitoire chaotique. Soit l'attracteur chaotique perd sa stabilité soit la condition initiale ne se situe plus dans le bassin d'attraction de l'ensemble invariant chaotique lorsque le RSB augmente et la trajectoire trouve un point fixe non-équivoque. De plus, la durée de vie du régime transitoire baisse lorsque le RSB croît. Des variantes de cette description générale incluent l'existence de points fixes multiples dans la région de chute en ce qui concerne le décodage itératif des codes LDPC irréguliers, ainsi que l'absence de point fixe indécisif à RSB faible en ce qui concerne le décodage itératif des codes produits.

Chapitre 4

Analyse du Décodage Itératif des Codes LDPC et des Codes Produits Utilisant L'Approximation Gaussienne

Le chapitre précédent a montré en quoi les algorithmes de décodage itératif peuvent être considérés comme des fonctions non-linéaires complexes et analysés en employant la théorie des systèmes dynamiques non-linéaires. Cependant, le grand nombre de variables et de paramètres complique l'étude des trajectoires. L'objectif du présent chapitre est de contribuer à la compréhension du décodage itératif en construisant un modèle approché unidimensionnel simple.

Nous proposons une nouvelle approche de type évolution de densité afin d'analyser différents systèmes de décodage itératif, tels que les codes LDPC et les codes produits, à partir de densités de probabilité gaussiennes. Plus précisemment, pour ces classes de codes, nous calculons une fonction à 1-D dont les itérées représentent directement la probabilité d'erreur pour le canal à bruit gaussien additif (AWGN) et le canal à évanouissements de Rayleigh. Ces modèles simples permettent une analyse qualitative de la dynamique non-linéaire de l'algorithme de décodage. Comme application, nous calculons les seuils de décodage et nous montrons qu'ils sont en accord avec les résultats de simulation disponibles dans la littérature.

4.1 Introduction à l'Evolution de Densité Utilisant l'Approximation Gaussienne

Récemment, différentes techniques ont été proposées dans la littérature pour analyser le décodage itératif en suivant l'évolution de la densité de probabilité de l'information échangée dans le décodeur. Cette idée a été introduite à l'origine pour les codes LDPC [6, 7] sous le nom d'*évolution de densité*. Pour ces codes particuliers, les densités de probabilité exactes des messages échangés dans le décodeur sont connues parce que l'information extrinsèque admet une forme analytique donnée par la "règle de la tanh", ainsi que nous l'avons expliqué dans la Sec 2.2.4. Cette méthode a ensuite été étendue aux turbo codes [50] en utilisant des techniques Monte-Carlo pour calculer l'histogramme de l'information extrinsèque. Toutefois, l'évolution de densité requiert une évaluation numérique des densités des messages utilisés par le décodeur et est généralement gourmande en temps de calcul. Souvent, la densité de probabilité de l'information extrinsèque est approximée par une gaussienne, soit pour simplifier l'analyse soit lorsqu'une expression analytique de l'information extrinsèque n'est pas disponible. La validité de cette hypothèse a été reconnue en premier par Wiberg [51] et utilisée dans [52] pour réaliser une analyse approchée des codes LDPC. L'approximation gaussienne, en conjonction avec des simulations Monte Carlo, a aussi été proposée pour analyser les performances de l'algorithme de turbo décodage [53, 54]. Les travaux précédents relatifs aux approximations gaussiennes ont reposé sur différents paramètres afin d'obtenir un modèle unidimensionnel, en particulier la moyenne dans [52], le RSB (rapport signal sur bruit) dans [53, 54], l'information mutuelle dans [55, 56] et le TEB (taux d'erreur binaire) [56, 57]. Une autre méthode qui fait correspondre la moyenne et la covariance a été présentée dans [58].

Ici, nous proposons un modèle du décodage itératif des codes LDPC et des codes produits à partir du TEB en utilisant l'approximation gaussienne. Pour les codes LDPC, nous suivons une méthode partiellement similaire à celle suggérée dans [52] afin d'analyser le décodeur à passage de messages. Cependant, notre méthode est basée sur une expression analytique des probabilités d'erreurs. Par probabilité d'erreur, nous entendons içi la probabilité que des noeuds de variable envoyent des messages incorrects. De plus, nous montrons que notre approche conduit à une condition de stabilité en accord avec l'évolution de densité. D'autre part, pour les codes produits, notre point de départ est [18]. Pour cette classe de codes, nous introduisons une nouvelle approche d'évolution de densité basée sur l'évaluation de l'information extrinsèque échangée par les décodeurs constitutifs. Dans les deux cas, malgré la simplicité du modèle, il est possible de prédire les

seuils du décodeur avec une précision acceptable en comparant avec les simulations.

4.2 Modèle du Décodage à Passage de Messages des Codes LDPC

4.2.1 Préliminaires sur les Codes LDPC

Une discussion détaillée des codes LDPC peut être trouvée dans la Sec. 2.2. Nous rappelons brièvement qu'un code LDPC est défini par un graphe bipartite [6] formé de noeuds de variable et de noeuds de parité reliés par des connexions. Supposons que d_v (resp. d_c) soit le degré maximal des noeuds de variable (resp. de parité) ; nous désignons le polynôme de distribution des degrés de variable (resp. de parité) par : $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ (resp. $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$) [7].

Le message v envoyé par un noeud de variable à un noeud de parité sur la connexion e est le rapport de vraisemblance logarithmique de ce noeud de variable, sachant les rapports de vraisemblance logarithmiques u_i des noeuds de parité reçus sur toutes les connexions incidentes, excepté e, et sachant le rapport de vraisemblance logarithmique du canal u_0 [6] :

$$v = u_0 + \sum_i u_i \tag{4.2.1}$$

Le message u envoyé par un noeud de parité à un noeud de variable sur la connexion e est le rapport de vraisemblance logarithmique de ce noeud de parité, sachant les rapports de vraisemblance logarithmiques v_i des noeuds de variable reçus sur toutes les connexions incidentes, excepté e [6] :

$$\tanh\left(\frac{u}{2}\right) = \prod_i \tanh\left(\frac{v_i}{2}\right) \tag{4.2.2}$$

Les équations (4.2.1) et (4.2.2) constitutent une itération de décodage et chaque message des noeuds de variable est initialisé par le rapport de vraisemblance logarithmique du bit correspondant en provenance du canal, u_0 [7, 52].

4.2.2 Approximation Gaussienne des Densités

Nous allons suivre de près l'approche prise par les auteurs de [52]. En se basant sur de nombreuses simulations de l'algorithme d'évolution de densité, il existe suffisamment de données expérimentales pour dire que la densité de probabilité des messages u des noeuds de parité est gaussienne. Ceci est particulièrement

vrai lorsque le polynôme de distribution des degrés à droite $\rho(x)$ est concentré sur un petit nombre de degrés, ceci étant vérifié pour les codes réguliers et aussi pour presque tous les codes irréguliers [52]. De plus, l'analyse est grandement simplifiée si une densité de probabilité, appelons-là f, vérifie la condition de symmétrie : $f(x) = e^x f(-x)$. Il a été montré par Richardson *et al.* [7] que les densités de probabilité de u_0, v and u dans les equations (4.2.1) et (4.2.2) satisfont la condition de symmétrie.

Tout au long de l'analyse, nous nous restreindrons à la modulation à deux états de phase (BPSK) ($0 \to +1$, $1 \to -1$). L'algorithme à passage de messages peut être analysé avec les hypothèses suivantes. Si la longueur du code tend vers l'infini, le théorème de concentration [6] assure que la performance d'un graphe bipartite particulier choisi au hasard peut être assimilée à la performance du graphe sans cycles, *i.e.* les messages reçus par chaque noeud à chaque itération sont des variables aléatoires i.i.d. (identiquement et indépendamment distribuées). Dans la suite, cette hypothèse sera supposée valide. Sans perte de généralité, nous admettrons que le mot de code tout-zéro est transmis, c'est pourquoi la probabilité d'erreur $P_e^l(\sigma)$ à l'iteration l est simplement la probabilité moyenne que les messages des noeuds de variable soient négatifs [7].

Canal AWGN

Nous considérons le canal AWGN et notons σ l'écart-type du bruit. Soit $m_{u_0} = \frac{2}{\sigma^2}$ la moyenne de u_0, et m_u^l ainsi que m_v^l les moyennes de u et v à l'itération l, respectivement. Notre objectif est de trouver une expression de $P_e^{l+1}(\sigma)$ la probabilité d'erreur à l'itération $l+1$ en tant qu'une fonction (non-linéaire) de $P_e^l(\sigma)$, la probabilité d'erreur à l'itération l. Dans ce but, considérons un message u d'un noeud de parité de degré j à l'iteration $l+1$; de part l'équation (4.2.2) :

$$\text{sign}(u) = \sum_i \text{sign}(v_i) \mod 2,$$

où sign(x) vaut 0 si $x > 0$ et 1 sinon. Gallager [3] a montré que la probabilité d'avoir $u < 0$ pour des noeuds de degré j vaut : $\frac{1}{2}\left[1 - \left(1 - 2P_e^l(\sigma)\right)^{(j-1)}\right]$. Maintenant, en moyennant sur tous les degré des noeuds de parité et en prenant en compte que la densité de probabilité de u est approximativement une gaussienne symmétrique, nous obtenons la probabilité d'avoir $u < 0$ comme suit

$$Q\left(\sqrt{\frac{m_u^{l+1}}{2}}\right) = \frac{1}{2}\sum_{j=2}^{d_c} \rho_j \left[1 - \left(1 - 2P_e^l(\sigma)\right)^{(j-1)}\right], \tag{4.2.3}$$

où : $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}} dt$.

De même, considérons l'équation (4.2.1) dans le cas du message v d'un noeud de variable de degré i, alors

$$m_v^{l+1} = m_{u_0} + (i-1)m_u^{l+1},$$

et la densité de probabilité de v est une gaussienne symmétrique puisque v est une somme de variables aléatoires dont la densité de probabilité est gaussienne et symmétrique, c'est pourquoi la probabilité d'avoir $v < 0$ vaut

$$Q\left(\sqrt{\frac{m_v^{l+1}}{2}}\right) = Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\frac{m_u^{l+1}}{2}}\right).$$

Maintenant, en moyennant sur tous les degrés des noeuds de variable possibles, nous obtenons :

$$P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\frac{m_u^{l+1}}{2}}\right). \tag{4.2.4}$$

Ensuite, en combinant les équations (4.2.3) and (4.2.4) et en définissant le polynôme $s(x)$ par

$$s(x) = \frac{1}{2}\sum_{j=2}^{d_c} \rho_j \left[1 - (1-2x)^{(j-1)}\right],$$

nous obtenons l'expression suivante de la probabilité d'erreur à l'itération $l+1$

$$P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s\left(P_e^l(\sigma)\right)\right)\right\}^2}\right). \tag{4.2.5}$$

Autrement dit, l'Eq. (4.2.5) représente une fonction non-linéaire unidimensionnelle de la forme

$$P_e^{l+1}(\sigma) = f\left(P_e^l(\sigma), \sigma\right), \quad l \geq 1 \tag{4.2.6}$$

décrivant la dynamique de l'algorithme à passage de messages en terme de probabilité d'erreur, avec $P_e^0(\sigma) = Q\left(\frac{1}{\sigma}\right)$. La fonction non-linéaire $f(x;\sigma)$ est définie par :

$$\boxed{f(x;\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s\left(x\right)\right)\right\}^2}\right)} \tag{4.2.7}$$

où σ agit comme un paramètre de contrôle.

Nous montrons maintenant que la fonction (4.2.7) admet la même condition de stabilité que celle trouvée en [7] en utilisant l'évolution de densité.

Théorème 4.2.1. *$x = 0$ est un point fixe stable de la fonction (4.2.7) si et seulement si : $\lambda'(0)\rho'(1) < e^{\frac{1}{2\sigma^2}}$.*

Démonstration. A partir de l'equation (4.2.7) nous avons $\lim_{x\to 0} f(x;\sigma) = 0$, donc 0 est un point fixe de la fonction. De plus, ce point fixe est stable si $\lim_{x\to 0} \frac{\partial f}{\partial x}(x,\sigma) < 1$ [47]. Nous montrons dans l'Annexe A que $\lim_{x\to 0} \frac{\partial f}{\partial x}(x,\sigma) = e^{-\frac{1}{2\sigma^2}} \lambda'(0)\rho'(1)$, ce qui achève la preuve. □

Canal à Evanouissements de Rayleigh

Sur le canal à évanouissements de Rayleigh, la densité de probabilité des rapports de vraisemblance du canal vaut [56] :

$$p_{u_0}(u_0) = \frac{\sigma^2}{2\sqrt{1+2\sigma^2}} \exp\left(\frac{u_0 - \sqrt{1+2\sigma^2}|u_0|}{2}\right) \tag{4.2.8}$$

tandis que la probabilité de transition du canal est donnée par

$$p = \int_{-\infty}^{0} p_{u_0}(u_0)du_0 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+2\sigma^2}}\right). \tag{4.2.9}$$

Les messages des noeuds de parité ont une densité de probabilité pouvant être approximée par une gaussienne symmétrique de moyenne m_u^{l+1} à l'itération $(l+1)$. Il s'ensuit qu'à l'itération $(l+1)$, la densité de probabilité du message v d'un noeud de variable de degré i est la convolution de $p_{u_0}(u_0)$ avec une gausienne symmétrique de moyenne $(i-1)m_u^{l+1}$. La probabilité d'avoir $v < 0$ a une expression connue [56] :

$$Q\left(\sqrt{\frac{(i-1)m_u^{l+1}}{2}}\right) - \frac{1}{\sqrt{1+2\sigma^2}} Q\left(\sqrt{(1+2\sigma^2)\frac{(i-1)}{2}m_u^{l+1}}\right) \exp\left(\frac{\sigma^2(i-1)m_u^{l+1}}{2}\right). \tag{4.2.10}$$

En moyennant cette expression sur tous les degrés des noeuds de variable :

$$\begin{aligned} P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i \Bigg\{ & Q\left(\sqrt{\frac{(i-1)m_u^{l+1}}{2}}\right) \\ & - \frac{1}{\sqrt{1+2\sigma^2}} Q\left(\sqrt{(1+2\sigma^2)\frac{(i-1)}{2}m_u^{l+1}}\right) \exp\left(\frac{\sigma^2(i-1)m_u^{l+1}}{2}\right) \Bigg\}. \end{aligned}$$

En rappelant que : $\frac{m_u^{l+1}}{2} = \left[Q^{-1}\left(s(P_e^l(\sigma))\right)\right]^2$, nous pouvons définir la fonction :

$$f(x;\sigma) = \sum_{i=2}^{d_v} \lambda_i \Bigg\{ Q\left(\sqrt{(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2}\right) - \frac{1}{\sqrt{1+2\sigma^2}} Q\left(\sqrt{(1+2\sigma^2)(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2}\right) \exp\left(\sigma^2(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2\right) \Bigg\}. \tag{4.2.11}$$

Les itérées de la probabilité d'erreur des messages des noeuds de variable est alors donnée par

$$\begin{cases} P_e^{l+1}(\sigma) &= f\left(P_e^l(\sigma), \sigma\right) \\ P_e^0(\sigma) &= \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+2\sigma^2}}\right). \end{cases}$$

De même que dans le cas du canal AWGN, 0 est un point fixe de f dont la condition de stabilité (voir Annexe B) est donnée par

$$\lambda'(0)\rho'(0) < 1 + \frac{1}{2\sigma^2} \tag{4.2.12}$$

ce qui coincide avec le résultat trouvé dans [10].

4.3 Modèle du Décodage Itératif des Codes Produits

4.3.1 Préliminaires sur les Codes Produits

Un code produit $C_p = C_1 \bigotimes C_2$ est défini par la concaténation en série de deux codes en bloc $C_1(n_1, k_1, d_1)$ et $C_2(n_2, k_2, d_2)$. Nous supposerons que des codes binaires sont utilisés. Les bits d'information sont placés dans un tableau de k_1 lignes et k_2 colonnes. Les colonnes (resp. les lignes) sont encodées en utilisant C_1 (resp. C_2), comme décrit dans la Fig. 4.1.

Le processus de décodage itératif est décrit par la Fig. 4.2. Le décodage est réalisé itérativement en colonne puis en ligne en utilisant l'algorithme de Chase modifié [18]. Le décodeur colonne utilise des observations en provenance du canal $\mathbf{Z}$ et de l'information *a priori* $\mathbf{A}_c$ sous la forme de rapports de vraisemblance logarithmiques afin de générer des rapports de vraisemblance logarithmiques *a posteriori* bit à bit $\mathbf{L}_c$. L'information *extrinsèque* est alors définie par $\mathbf{E}_c = \mathbf{L}_c - \mathbf{Z} - \mathbf{A}_c$. Après entrelacement matriciel, $\mathbf{E}_c$ est utilisé en tant qu'information *a priori* $\mathbf{A}_r$ en conjonction avec $\mathbf{Z}$ par le décodeur ligne afin de générer des rapports de vraisem-

blance logarithmiques *a posteriori* $\mathbf{L}_r$ pour chaque bit. L'information *extrinsèque* est alors définie par $\mathbf{E}_r = \mathbf{L}_r - \mathbf{Z} - \mathbf{A}_r$ et est utilisée en tant qu'information *a priori* pour les colonnes après entrelacement matriciel.

4.3.2 Analyse du Décodeur Itératif

Supposons que l'algorithme de Chase numéro 2 soit utilisé en tant que décodeur ligne/colonne avec la modification proposée dans [18] pour obtenir des sorties pondérées. Pour une decription détaillée de l'algorithme, voir la Sec. 2.4.3. Soit $C(n, k, d)$ l'un des codes constitutifs ; le pouvoir de correction d'erreurs de ce code vaut $t = \lfloor (d-1)/2 \rfloor$ bits et le nombre de positions les moins fiables utilisé pour générer la liste des mots de code candidats vaut $q = \lfloor d/2 \rfloor$. Supposons aussi que la modulation à deux états de phase (BPSK) soit utilisée ($0 \rightarrow +1$, $1 \rightarrow -1$) et que le mot de code tout-zéro est transmis. Nous modifions la méthode proposée dans [59] pour obtenir une bonne approximation du TEB des codes constitutifs. Supposons que le décodeur admette à son entrée le vec-

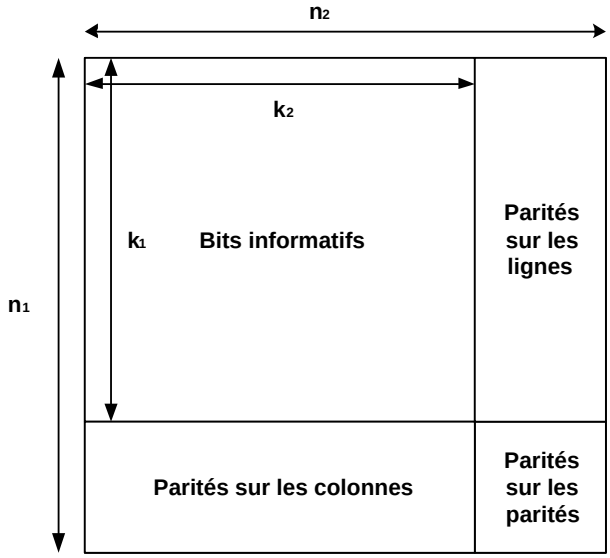

FIG. 4.1 – Code produit $C_p = C_1 \bigotimes C_2$.

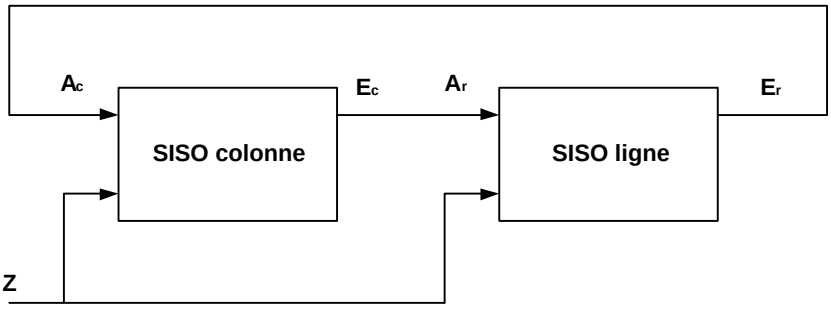

FIG. 4.2 – Schéma bloc du décodeur itératif d'un code produit.

teur des rapports de vraisemblance logarithmiques $\mathbf{r} = (r_1, \ldots, r_n)$ à la place des sorties du canal. Soit $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ le vecteur des fiabilités correpondant avec $\alpha_i = |r_i|$, $i = 1, \ldots, n$. Si i erreurs de transmission se produisent, les valeurs de fiabilité correspondant aux i décisions fermes érronées (resp. aux $n - i$ décisions ferme correctes) peuvent être réordonnées dans l'ordre décroissant : $\beta_1(i) \geq \beta_2(i) \geq \cdots \geq \beta_i(i)$ (resp. $\gamma_1(n-i) \geq \gamma_2(n-i) \geq \cdots \geq \gamma_{n-i}(n-i)$). Une bonne approximation du taux d'erreur des mots de code de l'algorithme de Chase numéro 2 est donné par [59] :

$$P_e = \sum_{i=t+1}^{q+t} \binom{n}{i} p^i (1-p)^{n-i} P\left(\beta_{t+1}(i) \geq \gamma_{n-q-t}(n-i)\right) + \sum_{i=q+t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i},$$

où p représente la probabilité de transition du canal. Avec une légère modification, on obtient une approximation du TEB comme suit

$$P_b = \sum_{i=t+1}^{q+t} \frac{i}{n} \binom{n}{i} p^i (1-p)^{n-i} P\left(\beta_{t+1}(i) \geq \gamma_{n-q-t}(n-i)\right) + \sum_{i=q+t+1}^{n} \frac{i}{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{4.3.13}$$

Supposons que les éléments de $\mathbf{r}$ soient i.i.d. avec une densité de probabilité $f(x)$ et soit $f_\alpha^c(x)$ (resp. $f_\alpha^e(x)$) la densité de probabilité associée à une fiabilité correspondant à une décision ferme correcte (resp. incorrecte), alors

$$p = \int_{-\infty}^{0} f(x)dx \tag{4.3.14}$$

$$f_\alpha^c(x) = \frac{f(x)}{1-p} u(x) \tag{4.3.15}$$

$$f_\alpha^e(x) = \frac{f(-x)}{p} u(x), \tag{4.3.16}$$

où $u(x)$ est la fonction échelon unité.

La méthode de calcul du terme $P\left(\beta_{t+1}(i) \geq \gamma_{n-q-t}(n-i)\right)$ peut être trouvée dans [59] et est rappelée dans l'Annexe C.

4.3.3 Approximation Gaussienne des Densités

Un modèle couramment utilisé de la densité de probabilité de l'information *extrinsèque* est la gaussienne *symmétrique* [52, 54, 57] qui est paramétrée uniquement par sa moyenne m_{E_r} pour les lignes et m_{E_c} pour les colonnes, la variance étant le double de la moyenne. Nous allons utiliser ce modèle dans la suite de cette analyse. De plus, nous supposons que l'information *a priori* est i.i.d., même si cette hypothèse n'est vérifiée dans la pratique que pour des entrelaceurs de

grande taille.

canal AWGN

Sur le canal AWGN, les rapports de vraisemblance logarithmiques du canal $\mathbf{Z}$ ont une distribution gaussienne *symmétrique* de moyenne $m_Z = 2/\sigma^2$, où σ représente l'écart-type du bruit du canal. C'est pourquoi les rapports de vraisemblance logarithmiques à l'entrée du décodeur colonne $\mathbf{Z} + \mathbf{A}_c$ ont aussi une distribution gaussienne *symmétrique* de moyenne $m_Z + m_{E_r}$. Nous obtenons le TEB après décodage des colonnes P_b^c en appliquant la méthode décrite dans la Sec. 4.3.2 avec

$$f(x) = \frac{q\left(\dfrac{x - (m_Z + m_{E_r})}{\sqrt{2\,(m_Z + m_{E_r})}}\right)}{\sqrt{2\,(m_Z + m_{E_r})}}$$

où $q(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$. Puisque $\mathbf{L}_c = \mathbf{Z} + \mathbf{A}_c + \mathbf{E}_c$, la densité de probabilité $p_{L_c}(x)$ de $\mathbf{L}_c$ est encore gaussienne et *symmétrique*. Soit m_{L_c}la moyenne correspondante, il s'ensuit que

$$P_b^c = \int_{-\infty}^{0} p_{L_c}(x)dx = Q\left(\sqrt{\frac{m_{L_c}}{2}}\right),$$

où $Q(x) = \int_x^{+\infty} q(t)dt$. Il en résulte que

$$m_{E_c} = 2\left[Q^{-1}\left(P_b^c\right)\right]^2 - m_Z - m_{E_r}.$$

Une méthode analogue est utilisée pour décrire le décodeur ligne. Les rapports de vraisemblance logarithmiques à l'entrée du décodeur ligne sont $\mathbf{Z} + \mathbf{A}_r$ de moyenne $m_Z + m_{E_c}$. Nous obtenons le TEB après décodage des lignes P_b^r en appliquant la méthode décrite dans la Sec. 4.3.2 avec

$$f(x) = \frac{q\left(\dfrac{x - (m_Z + m_{E_c})}{\sqrt{2\,(m_Z + m_{E_c})}}\right)}{\sqrt{2\,(m_Z + m_{E_c})}}$$

Notons que les rapports de vraisemblance logarithmiques $\mathbf{L}_r = \mathbf{Z} + \mathbf{A}_r + \mathbf{E}_r$ ont une distribution gaussienne *symmétrique* de moyenne m_{L_r}, il s'ensuit que

$$P_b^r = Q\left(\sqrt{\frac{m_{L_r}}{2}}\right),$$

Et finallement,

$$m_{E_r} = 2\left[Q^{-1}\left(P_b^r\right)\right]^2 - m_Z - m_{E_c}.$$

En itérant ce processus avec l'initialisation $m_{E_r} = 0$, nous obtenons une description du décodage itératif des codes produits sur le canal AWGN.

Canal à Evanouissements de Rayleigh

Sur le canal à évanouissements de Rayleigh, la distribution des rapports de vraisemblance logarithmiques du canal $\mathbf{Z}$ vaut [56] :

$$p_Z(z) = \frac{\sigma^2}{2\sqrt{1+2\sigma^2}} \exp\left(\frac{z - \sqrt{1+2\sigma^2}|z|}{2}\right) \tag{4.3.17}$$

C'est pourquoi la distribution des rapports de vraisemblance logarithmiques à l'entrée du décodeur colonne $\mathbf{Z} + \mathbf{A}_c$ est la convolution de $p_Z(z)$ avec une gaussienne *symmétrique* de moyenne m_{E_r} [56] :

$$\begin{aligned} f(x) =& \frac{\sigma^2}{4\sqrt{1+2\sigma^2}} \exp\left(\frac{\sigma^2 m_{E_r}}{2}\right) \\ & \times \left[\exp\left(\frac{1+\sqrt{1+2\sigma^2}}{2} x\right) \operatorname{erfc}\left(\frac{x/\sqrt{m_{E_r}} + \sqrt{m_{E_r}(1+2\sigma^2)}}{2}\right) \right. \\ & \left. + \exp\left(\frac{1-\sqrt{1+2\sigma^2}}{2} x\right) \operatorname{erfc}\left(\frac{-x/\sqrt{m_{E_r}} + \sqrt{m_{E_r}(1+2\sigma^2)}}{2}\right) \right] \end{aligned} \tag{4.3.18}$$

En utilisant $f(x)$, nous obtenons le TEB des colonnes P_b^c en appliquant la méthode décrite dans la Sec. 4.3.2. Puisque $\mathbf{L}_c = \mathbf{Z} + \mathbf{A}_c + \mathbf{E}_c$, la densité de probabilité $p_{L_c}(x)$ des rapports de vraisemblance logarithmiques $\mathbf{L}_c$ est aussi la convolution de $p_Z(z)$ avec une gaussienne *symmétrique* Gaussian de moyenne $m = m_{E_r} + m_{E_c}$. Il s'ensuit que [56] :

$$P_b^c = \int_{-\infty}^{0} p_{L_c}(x)dx = T_\sigma(m), \tag{4.3.19}$$

où

$$T_\sigma(m) = Q\left(\sqrt{\frac{m}{2}}\right) - Q\left(\sqrt{\frac{(1+2\sigma^2)m}{2}}\right) \frac{1}{\sqrt{1+2\sigma^2}} \exp\left(\frac{\sigma^2 m}{2}\right). \tag{4.3.20}$$

Il en résulte que :

$$m_{E_c} = T_\sigma^{-1}\left(P_b^c\right) - m_{E_r} \tag{4.3.21}$$

Le décodeur ligne est décrit par les mêmes équations, mais en remplaçant P_b^c par P_b^r et en inversant les rôles de m_{E_r} et m_{E_c}. En itérant ce processus avec l'initialisation $m_{E_r} = 0$, on obtient une description du décodage itératif des codes

produits sur le canal à évanouissements de Rayleigh.[1]

4.4 Application : Calcul de Seuils

Cette partie du travail est consacrée au calcul du seuil de l'algorithme de décodage itératif. Comme nous allons le montrer, l'existence du seuil s'explique par la dynamique non-linéaire du modèle à 1-D décrivant le système de décodage itératif.

4.4.1 Codes LDPC

Nous étudions les propriétés de convergence de l'algorithme à passage de messages au moyen de la fonction unidimentionnelle (4.2.6) calculée dans la Sec. 4.2.2 pour le canal AWGN.

Théorème 4.4.1. *Définissons le seuil* $\sigma^* = \sup\left\{\sigma > 0 : \lim_{l\to+\infty} P_e^l(\sigma) = 0\right\}$. *Si* $\sigma \leq \sigma^*$, $P_e^l(\sigma)$ *converge vers* 0, *sinon* $P_e^l(\sigma)$ *converge vers une valeur strictement supérieure à* 0.

Démonstration. A partir de l'équation (4.2.7) il est clair que $f(x;\sigma) \geq 0$ et $\frac{\partial f}{\partial x}(x,\sigma) > 0, \forall \sigma > 0$ et $\forall x \in]0, P_e^0(\sigma)]$, c'est pourquoi $P_e^l(\sigma)$ est décroissant et converge vers un point fixe. Il s'ensuit que s'il n'y a pas de point fixe dans $[0, P_e^0(\sigma)]$ autre que 0, $P_e^l(\sigma)$ converge vers 0. Réciproquement, supposons qu'il existe un point fixe $x > 0$ dans $[0, P_e^0(\sigma)]$ alors $P_e^l(\sigma) \geq x, \forall l > 0$ puisque $f(x;\sigma)$ est croissante sur $[x, P_e^0(\sigma)]$. On en déduit que $P_e^l(\sigma)$ converge vers un point fixe strictement supérieur à 0. Maintenant, à partir de l'Eq. (A.0.1) de l'Annexe A on voit aisément que $\frac{\partial f}{\partial \sigma}(x,\sigma) > 0, \forall \sigma > 0$ et $\forall x \in]0, P_e^0(\sigma)]$, donc $\sigma > \sigma^*$ implique $P_e^l(\sigma) > P_e^l(\sigma^*)$, ceci achève la preuve. □

Exemple 4.4.2. *L'exemple suivant illustre l'effet de seuil pour un* $(d_v = 3, d_c = 27)$ *code LDPC régulier de rendement* $\frac{8}{9}$ *sur le canal AWGN. Le seuil trouvé grâce à l'analyse présentée dans la Sec. 4.2.2 vaut* $\sigma^* = 0.496$. *Les Figs. 4.3 à 4.5 montrent la fonction (4.2.7) et les itérées successives de* $P_e^l(\sigma)$, *ainsi que la première bissectrice, pour* σ *inférieur, égal et supérieur à* σ^*, *respectivement. Pour* $\sigma = \sigma^*$, *une bifurcation tangente se produit [47] : deux points fixes, l'un stable (S), l'autre instable (I) apparaissent (voir Fig. 4.5).*

[1] Notons qu'à la première itération, la distribution des rapports de vraisemblance logarithmiques à l'entrée du décodeur ligne est la convolution de $p_Z(z)$ avec un Dirac, puisqu'initialement $m_{E_r} = 0$.

Un comportement similaire peut être observé pour la fonction (4.2.11) sur le canal à évanouissements de Rayleigh.

Avant de procéder à l'évaluation numérique des seuils, nous ajoutons la remarque suivante à propos de l'optimisation des codes LDPC [7], [52].

Remarque 4.4.3. *L'expression de* $\frac{\partial f}{\partial \sigma}(x,\sigma)$ *dans l'équation (A.0.1) est minimale pour tout* $x \in [0, P_e^0(\sigma)]$ *lorsque* $s(x)$ *est minimum pour tout* x *; c'est le cas lorsque le polynôme* $\rho(x)$ *est concentré sur le plus petit degré possible pour un* $\lambda(x)$ *donné. En notant que* $\lim_{\sigma\to 0} f(x;\sigma) = 0$ *et en rappelant que* $f(x;\sigma)$ *est croissante pour tout* $\sigma > 0$ *et* $\forall x \in]0, P_e^0(\sigma)]$*, intuitivement* σ^* *est maximal lorsque la distribution des degrés des noeuds de parité est concentrée. Lors de l'optimisation de codes LDPC irréguliers, une fois que* $\rho(x)$ *est fixé, il est aisé de trouver un* $\lambda(x)$ *de manière à maximiser le seuil. Ceci est en accord avec un fait déjà mentionné dans la Sec. 4.2.2, à savoir que de bons codes LDPC irréguliers ont un polynôme* $\rho(x)$ *concentré et avec le fait que la performance d'un code de parité augmente si le nombre de bits d'information décroît. Des résultats similaires pour le canal binaire symmétrique et le canal binaire à éffacements ont été présentés dans [11] et [12], respectivement.*

Résultats Numériques

Nous terminons cette discussion en comparant les valeurs des seuils σ^* et des rapports $\left(\frac{E_b}{N_0}\right)^*$ correspondants obtenus à l'aide de notre analyse et par évolution de densité. Le Tab. 4.1 donne les seuils pour les codes LDPC réguliers avec $d_v = 3$ de rendement $R = \frac{m}{m+1}$, pour $m = 1, \ldots, 8$. On peut se rendre compte que le modèle proposé dans la Sec. 4.2.2 estime le seuil en E_b/N_0 avec une précision comprise entre 0.1 et 0.3 dB. La précision obtenue par les auteurs de [52] est

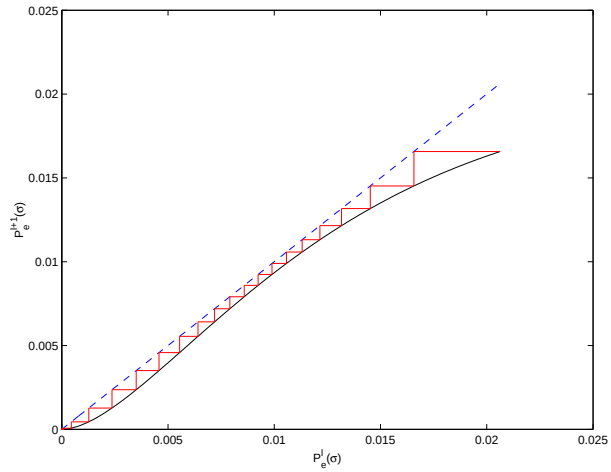

FIG. 4.3 – $P_e^{l+1}(\sigma)$ en fonction de $P_e^l(\sigma)$ en $\sigma = 0.49$ pour le $(d_v = 3, d_c = 27)$ code LDPC régulier.

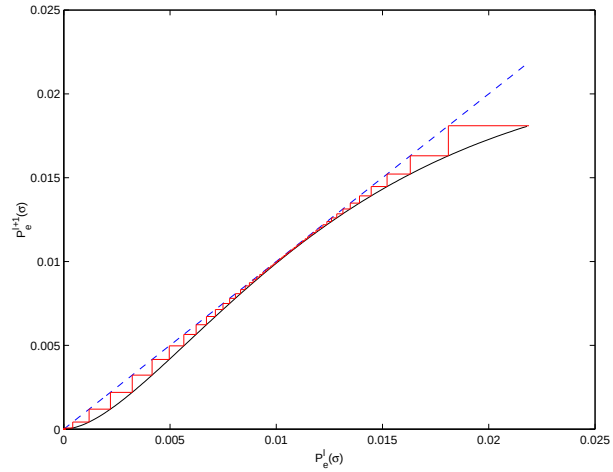

FIG. 4.4 – $P_e^{l+1}(\sigma)$ en fonction de $P_e^l(\sigma)$ à la valeur seuil $\sigma^* = 0.496$ pour le $(d_v = 3, d_c = 27)$ code LDPC régulier.

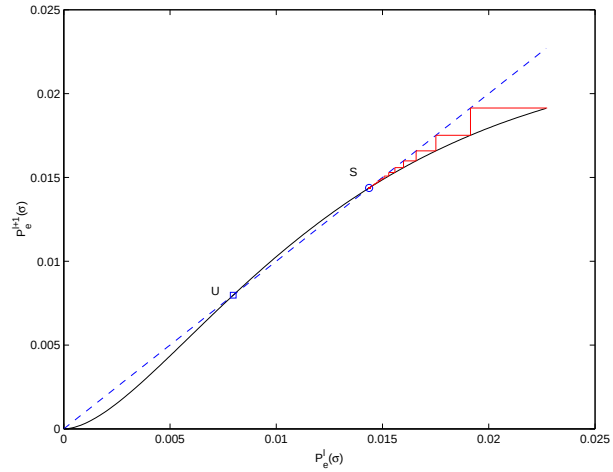

FIG. 4.5 – $P_e^{l+1}(\sigma)$ en fonction de $P_e^l(\sigma)$ à $\sigma = 0.50$ pour le $(d_v = 3, d_c = 27)$ code LDPC régulier. Notez la présence d'un point fixe stable (S) et d'un point fixe instable (I) dus à l'apparition d'une bifurcation tangente.

supérieure d'un ordre de grandeur environ ; cependant notre analyse sous forme analytique permet de mieux comprendre la dynamique du décodeur.

4.4.2 Codes Produits

Afin de travailler avec un système unidimensionnel, nous choisissons d'étudier les itérées du TEB $P_b^r(l)$ à la sortie du décodeur ligne en fonction de l'indice d'itération l et du paramètre du bruit σ. $P_b^r(0)$ est arbitrairement fixé à la probabilité de transition du canal. On peut vérifier que 0 est un point fixe du modèle de décodage itératif décrit dans la Sec. 4.3.2. Comme pour les codes LDPC, il existe un seuil $\sigma^* = \sup\{\sigma > 0 : \lim_{l\to+\infty} P_b^r(l) = 0\}$.

Exemple 4.4.4. *L'exemple suivant illustre l'effet de seuil pour le code BCH*

TAB. 4.1 – Seuils de codes LDPC réguliers obtenus par évolution de densité (σ^*_{ED}) et analyse gaussien (σ^*_{AG}) avec les $\frac{E_b}{N_0}$ correspondants.

d_v	d_c	Rendement	σ^*_{ED}	$\left(\frac{E_b}{N_0}\right)^*_{ED}$	σ^*_{AG}	$\left(\frac{E_b}{N_0}\right)^*_{AG}$
3	6	1/2	0.880	1.11 dB	0.848	1.43 dB
3	9	2/3	0.708	1.75 dB	0.690	1.97 dB
3	12	3/4	0.632	2.22 dB	0.619	2.41 dB
3	15	4/5	0.587	2.59 dB	0.577	2.74 dB
3	18	5/6	0.557	2.86 dB	0.548	3.01 dB
3	21	6/7	0.534	3.11 dB	0.527	3.22 dB
3	24	7/8	0.517	3.30 dB	0.510	3.42 dB
3	27	8/9	0.503	3.47 dB	0.496	3.59 dB

produit $BCH(64, 51, 6)^2$. *Le seuil trouvé grâce à l'analyse présentée dans la Sec. 4.3.2 vaut* $\sigma^* = 0.745$. *La Fig. 4.6 montre les itérées successives de* P_b^r *ainsi que la première bissectrice pour* $\sigma = \sigma^*$. *La trajectoire de décodage démarre dans le coin en haut à droite et se termine à l'origine. Pour* $\sigma = \sigma^*$, *la trajectoire de décodage entre dans une* région de tunnel *[54, 56] près de la bissectrice avec un ralentissement caractéristique de la vitesse de convergence.*

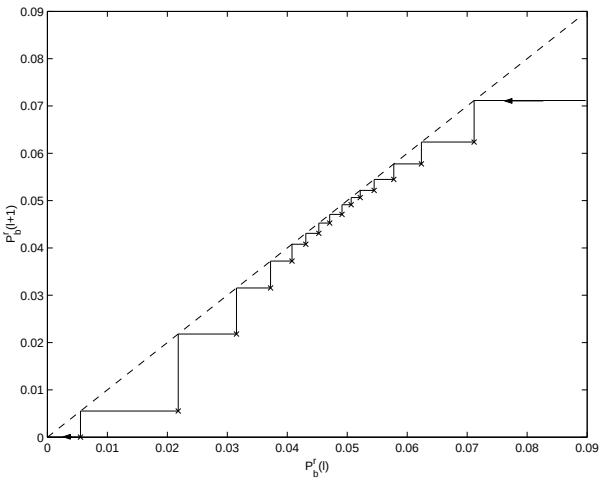

FIG. 4.6 – $P_b^r(l+1)$ en fonction de $P_b^r(l)$ à $\sigma = 0.745$ pour le code produit $BCH(64, 51, 6)^2$ sur le canal AWGN.

Intuitivement, la valeur du seuil σ^* est grande si le TEB décroît rapidement lorsque l'information extrinsèque augmente. Sans tenir compte du rendement de codage, ceci se produit lorsque la longueur du code est petite et le pouvoir de correction élevé.

TAB. 4.2 – Seuils σ^* de codes BCH produits avec le rapport $\frac{E_b}{N_0}$ correspondant pour le canal AWGN.

Code	Rendement	σ^*	$\left(\frac{E_b}{N_0}\right)^*_{dB}$
$(32, 21, 6)^2$	0.431	1.106	-0.2
$(32, 26, 4)^2$	0.660	0.803	0.7
$(64, 51, 6)^2$	0.635	0.745	1.5
$(64, 57, 4)^2$	0.793	0.615	2.2
$(128, 113, 6)^2$	0.779	0.588	2.7
$(128, 120, 4)^2$	0.879	0.513	3.3
$(256, 247, 4)^2$	0.931	0.447	4.3
$(512, 502, 4)^2$	0.961	0.401	5.1

TAB. 4.3 – Seuils σ^* de codes BCH produits avec le rapport $\frac{E_b}{N_0}$ correspondant pour le canal à évanouissements de Rayleigh.

Code	Rendement	σ^*	$\left(\frac{E_b}{N_0}\right)^*_{dB}$
$(32, 21, 6)^2$	0.431	0.896	1.6
$(32, 26, 4)^2$	0.660	0.580	3.5
$(64, 51, 6)^2$	0.635	0.508	4.8
$(64, 57, 4)^2$	0.793	0.366	6.7
$(128, 113, 6)^2$	0.779	0.330	7.7
$(128, 120, 4)^2$	0.879	0.246	9.7
$(256, 247, 4)^2$	0.931	0.170	12.7
$(512, 502, 4)^2$	0.961	0.118	15.7

Résultats Numériques

Les Tab. 4.2 et 4.3 donnent les seuils calculés avec la méthode présentée dans la Sec. 4.3.2 pour les codes produits simulés dans [18] sur le canal AWGN et le canal à évanouissements de Rayleigh, respectivement. Nous insistons sur le fait que bien qu'il ne soit pas possible de définir rigoureusement un seuil de décodage pour des codes de longueur finie, le seuil existant dans notre modèle est un bon indicateur du début de la *région de chute* des courbes de TEB prsentes dans [18], excepté pour les codes consitutifs de longueur 32. Dans ce cas particulier, les seuils calculés sont même en-dessous de la capacité. Ceci confirme que l'hypothèse, qui consiste à considérer l'information extrinsèque comme i.i.d., n'est valide que pour les longueurs de code élevées.

4.5 Conclusions

Nous avons présenté des modèles unidimensionnels du décodage itératif des codes LDPC et des codes produits, à partir de densités de probabilité gaussiennes. Ces fonctions à 1-D simples décrivent l'évolution des probabilités d'erreur en fonction du nombre d'itérations, à la fois sur le canal AWGN et le canal à évanouissements de Rayleigh. Pour les codes LDPC, notre analyse conduit à une condition de stabilité en accord avec la méthode de l'évolution de densité.

Notre approche permet une analyse qualitative de la dynamique non-linéaire de l'algorithme de décodage au voisinage du seuil. Nous avons aussi vérifié que les seuils obtenus grâce à notre analyse sont en accord avec les valeurs trouvées par évolution de densité ou par simulation Monte Carlo.

Chapitre 5

Propriétés de Convergence Asymptotique du Décodage Itératif des Codes Concaténés

Le chapitre précédent a montré que l'approximation gaussienne est capable de prédire correctement le comportement de l'algorithme de décodage itératif des codes LDPC lorsque le longueur du code tend vers l'infini et le taux d'erreur binaire (TEB) est proche de zéro. Nous proposons une étude similaire pour les codes concaténés (CC). Dans ce but, nous présentons une analyse des propriétés de convergence des systèmes de décodage itératif à rapport signal sur bruit (RSB) élevé. Nous introduisons un modèle non-linéaire simple, basé sur l'énumérateur de poids des codes constitutifs, pour décrire les itérées du taux d'erreur binaire. Ensuite, nous caractérisons la dynamique du décodeur de plusieurs codes concaténés sur le canal gaussien à entrée binaire, en terme de points fixes ainsi que de leur stabilité et vitesse de convergence associées.

5.1 Principes et Notations

Dans cette étude, nous proposons une analyse du décodage itératif des CC en faisant correspondre le TEB en sortie des décodeurs SISO avec le taux d'erreur correspondant à des rapports de vraisemblance logarithmiques ayant une distribution gaussienne, ainsi qu'il a été suggéré en premier dans [56]. Nous nous basons sur des techniques de borne par réunion pour calculer le TEB à partir des énumérateurs de poids des codes constitutifs. Cette méthode n'est pas capable de prédire la dynamique pour des valeurs élevées du TEB, à cause du problème bien connu de la divergence de la borne par réunion. Malheureusement, comme nous l'avons déjà constaté pour les codes LDPC et les codes produits, l'effet de seuil

correspond habituellement à des points fixes du TEB ayant une valeur élevée. C'est pourquoi, nous allons supposer que le RSB du canal est plus grand que le seuil pour d'éviter l'existence de points fixes entre la condition initiale et le plancher d'erreur, afin de se situer dans une région des RSB où notre méthode s'applique.

Nous insistons sur les similitudes de cette étude avec [60], bien que notre méthode de calcul du TEB repose sur la méthode du Ch. 4. Les contributions originales de ce travail comprennent une investigation de l'effet de la terminaison du treillis pour les codes convolutifs et une étude de la plupart des codes concaténés ayant un intérêt. Bien que le modèle que nous développons ne soit pas capable de fournir une explication de certains phénomènes non-linéaires typiques tels que des trajectoires périodiques et quasi-périodiques répertoriés dans le chapitre 3, le modèle analytique que nous proposons permet d'établir un lien entre la dynamique du système de décodage itératif et les paramètres des codes constitutifs. Pour des codes constitutifs de distance minimum finie, une autre limitation de la méthode proposée réside dans le fait que le plancher d'erreur prédit peut atteindre zéro. Ceci n'est pas en accord avec le fait que la distance minimum typique d'un CC, contrairement à ce qui se passe pour un code LDPC, reste finie lorsque la taille du code tends vers l'infini [19, 21, 13].

Nous réutilisons toutes les notations introduites dans la Sec. 2.3.1 en ce qui concerne la construction de CC, comportant les concaténations parallèle de codes (PCC), les concaténations série de codes (SCC) et les codes produits (PC). En particulier, nous rappelons que k, I et R désignent le nombre de bits informatifs, la taille de l'entrelaceur et le rendement de codage, respectivement. Pour un PC, le code en ligne (resp. en colonne) est un code en bloc de type (n_R, k_R) (resp. (n_C, k_C)). Par simplicité, nous nous restreignons à des CC faisant intervenir deux codes en bloc systématiques séparés par un entrelaceur. En particulier, si des codes convolutifs sont utilisés, nous supposerons que le treillis est terminé.

Nous réutilisons également les notations de la Sec. 2.3.3 en ce qui concerne le décodage itératif des CC. Un décodeur itératif générique est décrit par la Fig. 2.5. Le décodage est réalisé itérativement en utilisant deux décodeurs désignés par SISO1 et SISO2, respectivement. Nous considérons seulement un décodage au sens du *maximum a posteriori* (MAP), bien que ceci ne soit pas toujours faisable dans la pratique. SISO1 correspond au décodage du premier code, du code externe et du code en ligne pour un PCC, un SCC et un PC, respectivement. De même, SISO2 correspond au décodage du second code, du code interne et du code en colonne pour un PCC, un SCC et un PC, respectivement. Il sera avantageux de considérer le système de décodage itératif en tant qu'un système dynamique en boucle fermée, où SISO1 et SISO2 jouent le rôle des décodeurs constitutifs

non-linéaires du système en boucle ouverte correspondant.

Dans tout le chapitre, nous supposerons que les codes sont binaires et que le canal est le canal gaussien à entrée binaire ($0 \rightarrow +1$, binary $1 \rightarrow -1$) ; notons σ l'écart-type du bruit. Ainsi, sans perte de généralité, nous pouvons supposer que le mot de code tout-zéro est transmis. Nous supposerons que toutes les quantités échangées par le décodeur sont sous la forme de rapports de vraisemblance logarithmiques et peuvent être modélisées par des variables aléatoires i.i.d. ayant une distribution gaussienne symmétrique [6].

5.2 Analyse Non-linéaire du Décodage Itératif des CC

5.2.1 Modèle du Décodage Itératif des PCC

Par simplicité, nous faisons l'hypothèse standard que le PCC considéré est formé de deux codes constitutifs identiques ayant pour fonction énumératrice de poids d'entrée/redondance ou "*input-redundancy weight enumerating function*" (IRWEF) [19]

$$A(W,H) = \sum_{w} \sum_{h} A_{w,h} W^w H^h,$$

où $A_{w,h}$ représente le nombre de mots de code de poids informatif w et de poids redondant h. L'IRWEF sera utilisée dans la Sec. 5.3 pour obtenir une bonne approximation du TEB des codes constitutifs pour de faibles valeurs du TEB. Soit $2x$ la moyenne des rapports de vraisemblance logarithmiques *a priori* à l'entrée de l'un quelconque des SISO. Soient $f_{\frac{1}{\sigma^2}}$ et f_x la densité de probabilité des rapports de vraisemblance logarithmiques du canal et *a priori* à l'entrée du décodeur, respectivement.

$$f_{\frac{1}{\sigma^2}}(t) = \frac{q\left(\frac{t-\frac{2}{\sigma^2}}{2/\sigma}\right)}{2/\sigma}$$

$$f_x(t) = \frac{q\left(\frac{t-2x}{2\sqrt{x}}\right)}{2\sqrt{x}},$$

où $q(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$. Nous définissons $P(x,\sigma)$ comme le TEB après décodage d'un code constitutif, en supposant que les densités de probabilité des rapports de vraisemblance logarithmiques du canal et *a priori* sont $f_{\frac{1}{\sigma^2}}$ et f_x, respectivement. Des valeurs numériques de $P(x,\sigma)$ peuvent être obtenues par simulation Monte Carlo. Etant donné que le rapport de vraisemblance logarithmique après décodage est la somme des rapports de vraisemblance logarithmiques du ca-

nal, *a priori* et *extrinsèque*, sa densité de probabilité p est aussi une gaussienne symmétrique définie seulement par la moyenne m, et nous avons

$$P(x,\sigma) = \int_{-\infty}^{0} p(t)dt = Q\left(\sqrt{\frac{m}{2}}\right),$$

où $Q(x) = \int_{x}^{+\infty} q(t)dt$. C'est pourquoi la moyenne du rapport de vraisemblance logarithmique *extrinsèque* est donné par

$$2\left([Q^{-1}(P(x,\sigma))]^2 - \frac{1}{\sigma^2} - x\right)$$

Il s'ensuit que le taux d'erreur binaire à la sortie d'un SISO quelconque peut être décrit à chaque demi-itération l par $y_l = h(y_{l-1},\sigma)$, où h est la fonction non-linéaire suivante

$$h(y,\sigma) = P\left([Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P^{-1}(y,\sigma),\sigma\right), \tag{5.2.1}$$

La fonction inverse P^{-1} est définie puisque P est bijective. Les points fixes de la fonction pour $y > 0$ sont les solutions de $h(y,\sigma) = y$, ce qui équivaut à,

$$[Q^{-1}(y)]^2 - 2P^{-1}(y,\sigma) = \frac{1}{\sigma^2}. \tag{5.2.2}$$

Remarque 5.2.1. *Par définition du seuil de bruit σ^*, lorsque $\sigma \leq \sigma^*$, les itérées du TEB y_l commençant à $y_0 = Q\left(\frac{1}{\sigma}\right)$, atteignent un plancher d'erreur proche de 0 sans rester bloqué en aucun autre point fixe. Bien que l'objet de ce chapitre soit restreint au comportement du décodage itératif pour $\sigma \leq \sigma^*$, une méthode envisageable pour évaluer σ^* consiste à trouver la plus grande valeur de σ pour laquelle l' Eq. (5.2.2) n'a qu'un seul point fixe, à savoir le plancher d'erreur. Etant donné que la valeur du point fixe responsable de l'effet de seuil est typiquement proche de y_0, une expression valide de $P(x,\sigma)$ à TEB élevé (évalué numériquement par simulation Monte Carlo, par exemple) est nécéssaire.*

5.2.2 Modèle du Décodage Itératif des SCC

Un SCC est formé de deux codes constitutifs, qui en général ne sont pas identiques. La fonction énumératrice de poids ou "*weight enumerating function*" (WEF) du code externe est donnée par

$$A^o(Z) = \sum_{d \geq d^o_{min}} A^o_d Z^d$$

et l'IRWEF du code interne vaut

$$A^i(W,H) = \sum_w \sum_h A^i_{w,h} W^w H^h,$$

où A^o_d désigne le nombre de mots de code de poids d dans le code externe et $A^i_{w,h}$ représente le nombre de mots de code de poids informatif w et de poids redondant h dans le code interne. La distance minimum du code externe est notée d^o_{min}. Ces polynômes seront utilisés dans la Sec. 5.3 pour obtenir une bonne approximation du TEB des codes constitutifs. Nous définissons $P_o(x,\sigma)$ (resp. $P_i(x,\sigma)$) comme le TEB après décodage à la sortie du SISO correspondant au code externe (resp. interne), où le premier argument est tel que $2x$ représente la moyenne des rapports de vraisemblance logarithmiques *a priori*. En utilisant la même méthode que dans la Sec. 5.2.1, nous pouvons décrire l'évolution du TEB à la sortie de chaque SISO en définissant les fonctions non-linéaires h_o et h_i par

$$\begin{aligned} h_o(y,\sigma) &= P_o\left([Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P_i^{-1}(y,\sigma), \sigma\right) \\ h_i(y,\sigma) &= P_i\left([Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P_o^{-1}(y,\sigma), \sigma\right). \end{aligned} \tag{5.2.3}$$

Soit y^o_l (resp. y^i_l) le TEB à la sortie du SISO correspondant au code externe (resp. interne) à l'itération l,

$$\begin{aligned} y^o_l &= h_o(y^i_{l-1}, \sigma) \\ y^i_l &= h_i(y^o_l, \sigma), \end{aligned} \tag{5.2.4}$$

ou de manière équivalente,

$$\begin{aligned} y^o_l &= h_o \circ h_i(y^o_{l-1}, \sigma) \\ y^i_l &= h_i \circ h_o(y^i_{l-1}, \sigma). \end{aligned} \tag{5.2.5}$$

Il est facile de voir que h_o et h_i pour $y > 0$ ont les mêmes points fixes donnés par les solutions de

$$[Q^{-1}(y)]^2 - P_o^{-1}(y,\sigma) - P_i^{-1}(y,\sigma) = \frac{1}{\sigma^2}. \tag{5.2.6}$$

Théorème 5.2.2. *Les points fixes stables de $h_o \circ h_i$ and $h_i \circ h_o$ dans tout intervalle J sont exactement les points fixes stables de h_o and h_i si l'ensemble suivant de conditions suffisantes est satisfait $\forall y \in J$.*

1. *$h_o(y,\sigma)$ et $h_i(y,\sigma)$ sont des fonctions croissantes de y.*
2. *$h_o(y,\sigma) \leq y$.*
3. *$h_i(y,\sigma) \leq y$.*

4. *Un point fixe de h_o est stable si et seulement si c'est un point fixe stable de h_i.*

Démonstration. Nous reléguons la preuve dans l'Annexe G. □

Le Thm. 5.2.2, appliqué dans le voisinage J d'un point fixe sera utile dans la Sec. 5.3 pour trouver les points fixes et le critère de stabilité du système de décodage itératif en boucle fermée (décrit soit par $h_o \circ h_i$ soit par $h_i \circ h_o$) à partir des points fixes et des critères de stabilité des décodeurs constitutifs dans le système en boucle ouverte (décrits par h_o et h_i).

5.2.3 Modèle du Décodage Itératif des PC

Soit A^R (resp. A^C) la WEF du code en ligne (resp. en colonne)

$$\begin{aligned} A^R(Z) &= \sum_{d \geq d^R_{min}} A^R_d Z^d \\ A^C(Z) &= \sum_{d \geq d^C_{min}} A^C_d Z^d, \end{aligned}$$

où A^R_d et A^C_d désignent le nombre de mots de code de poids d dans les codes en ligne et en colonne, respectivement. Les distances minimum des code en ligne et en colonne sont notées d^R_{min} et d^C_{min}, respectivement. Ces polynômes seront utilisés dans la Sec. 5.3 pour obtenir une bonne approximation du TEB des codes constitutifs. Nous définissons $P_R(x, \sigma)$ (resp. $P_C(x, \sigma)$) comme le TEB après décodage en sortie du SISO correspondant au code en ligne (resp. en colonne), où le premier argument est tel que $2x$ représente la moyenne des rapports de vraisemblance logarithmiques *a priori*. En utilisant la même méthode que dans la Sec. 5.2.1, nous pouvons décrire l'évolution du TEB à la sortie de chaque SISO en définissant les fonctions non-linéaires h_R et h_C par

$$\begin{aligned} h_R(y, \sigma) &= P_R\left([Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P_C^{-1}(y, \sigma), \sigma \right) \\ h_C(y, \sigma) &= P_C\left([Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P_R^{-1}(y, \sigma), \sigma \right) \end{aligned} \tag{5.2.7}$$

Soit y^R_l (resp. y^C_l) le TEB à la sortie du SISO correspondant au code en ligne (resp. colonne) à l'itération l,

$$\begin{aligned} y^R_l &= h_R(y^C_{l-1}, \sigma) \\ y^C_l &= h_C(y^R_l, \sigma), \end{aligned} \tag{5.2.8}$$

ou de manière équivalente,

$$\begin{aligned} y_l^R &= h_R \circ h_C(y_{l-1}^R, \sigma) \\ y_l^C &= h_C \circ h_R(y_{l-1}^C, \sigma). \end{aligned} \tag{5.2.9}$$

On voit que cette description est très semblable à celle proposée pour les SCC. En effet, un PC est un cas particulier de SCC pour lequel le mot de code externe (resp. interne) est subdivisé en mots de code *indépendants* en ligne (resp. en colonne), grâce à l'entrelaceur matriciel. C'est pourquoi toutes les propriétés liées aux points fixes données par l'Eq. (5.2.6) et le Thm. 5.2.2 s'appliquent à condition de remplacer P_o, P_i, h_o et h_i par P_R, P_C, h_R et h_C, respectivement.

5.3 Dynamique Non-linéaire du Décodage Itératif des CC

Pour chaque CC présenté dans la Sec. 2.3.1, nous étudions la dynamique non-linéaire du décodage itératif à RSB élevé, en supposant que le TEB a atteint une valeur suffisamment faible pour pouvoir être approchée en utilisant les énumérateurs de poids définis dans la Sec. 5.2. Cette technique permet d'établir un lien entre les propriétés de convergence du processus de décodage et les paramètres des codes constitutifs.

5.3.1 Dynamique pour les PCC

Cas Général

Théorème 5.3.1. *Pour chaque code constitutif, une approximation du TEB des bits informatifs $P(x, \sigma)$ est obtenu à partir de l'IRWEF $A(W, H)$ comme suit*

$$P(x, \sigma) = \frac{1}{I} \sum_{w \geq 1} \sum_{h} w A_{w,h} Q\left(\sqrt{w \left(\frac{1}{\sigma^2} + x \right) + h \frac{1}{\sigma^2}} \right),$$

où $Q(x) = \int_x^{+\infty} q(t)dt$.

Démonstration. Le mot de code de longueur-n correspondant à chaque code constitutif est le mot de code tout-zéro $\mathbf{c}^0 = (c_1^0, \ldots, c_n^0)$. Etant donné que chaque SISO réalise un décodage au sens du MAP, la probabilité de confondre $\mathbf{c}^0$ avec n'importe quel autre mot de code $\mathbf{c}^{w,h}$ de poids informatif w et de poids redon-

dant h sur le canal gaussien à entrée binaire vaut

$$P_{w,h} = \Pr\left(\ln \frac{P(\mathbf{y}|\mathbf{c}^0)P(\mathbf{c}^0)}{P(\mathbf{y}|\mathbf{c}^{w,h})P(\mathbf{c}^{w,h})} < 0\right),$$

où $\mathbf{y} = (y_1, \ldots, y_n)$ est la séquence reçue, sachant que $\mathbf{c}^0$ est transmis.

La vraisemblance de $\mathbf{c}^0$ et $\mathbf{c}^{w,h}$ est calculée par un SISO de la manière suivante

$$P(\mathbf{y}|\mathbf{c}^0)P(\mathbf{c}^0) = C\exp\left[\sum_{k=1}^{n}(1-2c_k^0)\left(\frac{1}{\sigma^2}y_k + A_k/2\right)\right]$$
$$P(\mathbf{y}|\mathbf{c}^{w,h})P(\mathbf{c}^{w,h}) = C\exp\left[\sum_{k=1}^{n}(1-2c_k^{w,h})\left(\frac{1}{\sigma^2}y_k + A_k/2\right)\right],$$

où C est une constante et $\mathbf{A} = (A_1, \ldots, A_n)$ est le vecteur des rapports de vraisemblance logarithmiques *a priori*. Il s'ensuit que

$$\ln \frac{P(\mathbf{y}|\mathbf{c}^0)P(\mathbf{c}^0)}{P(\mathbf{y}|\mathbf{c}^{w,h})P(\mathbf{c}^{w,h})} = 2\sum_{k:c_k^{w,h}=1}\left(\frac{1}{\sigma^2}y_k + A_k/2\right).$$

Cette expression est une variable aléatoire ayant une distribution gaussienne symmétrique de moyenne $2\left(w\left(\frac{1}{\sigma^2}+x\right)+h\frac{1}{\sigma^2}\right)$. Nous rappelons que pour un PCC, $\mathbf{A}$ vaut zéro pour les bits redondants et est une variable aléatoire gaussienne symmétrique de moyenne $2x$ pour les bits informatifs. On en déduit que $P_{w,h} = Q\left(\sqrt{w\left(\frac{1}{\sigma^2}+x\right)+h\frac{1}{\sigma^2}}\right)$. Il suffit maintenant d'appliquer la borne par réunion et la preuve est achevée. □

Remarque 5.3.2. *Le lecteur peut se demander pourquoi la démonstration précédente considère une estimation au sens du MAP des mots de code, alors que le décodage itératif fonctionne avec un décodage au sens du MAP bit-à-bit des codes constitutifs. Nous avons vérifié expérimentalement que bien que l'estimation au sens du MAP des mots de code et le décodage au sens du MAP bit-à-bit peuvent mener à des décisions fermes différentes, ces deux méthodes ont néanmoins les mêmes performances en terme de taux d'erreur binaire moyen. Nous supposerons donc que le TEB du décodage au sens du MAP bit-à-bit peut toujours être assimilé au TEB de l'estimation au sens du MAP des mots de code, même si nous n'avons pas pu le prouver formellement.*

Exemple 5.3.3. *Supposons que le code convolutif* $(1, 5/7)$ *de rendement* $1/2$ *de la Fig. 2.6 avec terminaison du treillis soit choisi comme code constitutif. La Fig. 5.1 illustre* $h(y, 1.180)$ *(courbe en trait plein) pour* $k = I = 8192$*. Une trajectoire de décodage itératif typique, rebondissant de gauche à droite entre les*

courbes $z = h(y, 1.180)$ (trait plein) et $z = y$ (trait discontinu et pointillés), est aussi représentée. Dans cet exemple, h diverge vers l'infini pour $y \approx 10^{-5}$ à cause de la divergence de la borne par réunion, et les trajectoires convergent vers un point fixe $y^* \approx 1.45e-8$ lorsque la condition initiale $y_0 \leq 10^{-5}$.

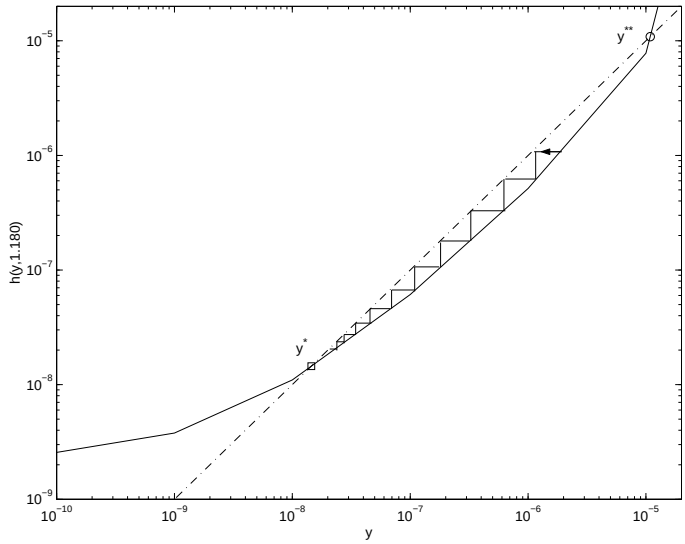

FIG. 5.1 – $h(y, 1.180)$ pour un PCC de rendement 1/3 utilisant des codes convolutifs à 4 états et tel que $I = 8192$.

D'autres exemples du même type montrent que les caractéristiques typiques de la fonction h sont les suivantes :

1. Deux points fixes y^* et y^{**}, $y^* < y^{**}$.
2. une *région de tunnel* [56] sur l'intervalle $[y^*; y^{**}]$.

Dans les paragraphes suivants, nous tentons d'expliquer qualitativement le comportement près de $y = 0$, en tant qu'une conséquence des mots de code de poids informatif un et deux dans les codes constitutifs.

Influence des Mots de Code de Poids Informatif Un

Lorsque la moyenne de l'information *a priori* $2x$ est grande, nous approximons $P(x, \sigma)$ grâce à l'inégalité $Q(\sqrt{z+y}) \leq Q(\sqrt{z})e^{-y/2}$

$$P(x, \sigma) = \alpha(\sigma) Q\left(\sqrt{x}\right), \tag{5.3.10}$$

où

$$\alpha(\sigma) = \frac{1}{I} \sum_h A_{1,h} \left[e^{-\frac{1}{2\sigma^2}}\right]^{h+1} \tag{5.3.11}$$

à condition que $\alpha(\sigma) \neq 0$ et $x \geq 0$, ce qui implique $P(x, \sigma) \leq \alpha(\sigma)/2$. Dans ce cas, seulement l'effet des mots de code de poids informatif un dans les codes

constitutifs est pris en compte. En injectant (5.3.10) dans (5.2.2), les points fixes de la fonction décrivant le décodage itératif d'un PCC sont les solutions de

$$S(y,\sigma) = [Q^{-1}(y)]^2 - 2\left[Q^{-1}\left(\frac{y}{\alpha(\sigma)}\right)\right]^2 - \frac{1}{\sigma^2} = 0. \tag{5.3.12}$$

On peut montrer que pour tout $\sigma > 0$, l'Eq. (5.3.12) n'a qu'une seule solution $y^* \in]0;\alpha(\sigma)/2]$ à condition que $\alpha(\sigma)$ soit suffisamment petit. De plus, nous avons $S(y,\sigma) \geq 0$ sur $]y^*;\alpha(\sigma)/2]$, ce qui équivaut à $[Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P^{-1}(y,\sigma) \geq P^{-1}(y,\sigma)$ et à partir de (5.2.1), on a $h(y,\sigma) \leq y$ puisque $P(x,\sigma)$ est une fonction décroissante de x. La solution y^* croît avec $\alpha(\sigma)$. En utilisant (5.3.11), on a que $\alpha(\sigma)$ croît avec σ, donc y^* augmente avec σ. Dans l'Annexe E, nous montrons que $\lim_{y^*\to 0} \frac{\partial h(y,\sigma)}{\partial y}\big|_{y=y^*} = 0$. En terme de dynamique non-linéaire, celà signifie qu'à RSB élevé, $h(y,\sigma)$ admet un point fixe stable qui augmente avec le paramètre de contrôle σ.

Remarque 5.3.4. *Nous rappelons que les itérée du TEB peuvent atteindre le point fixe y^* uniquement si celui-ci existe et s'il n'y a pas d'autre point fixe intercalé entre y^* et la condition initiale. Cette hypothèse n'est vérifiée que pour des valeurs asymptotiques du RSB du canal, i.e. une fois que le décodeur dépasse le seuil de bruit.*

Exemple 5.3.5. *La Fig. 5.2 illustre $S(y,1)$ sur l'intervalle $[0;\alpha(1)/2]$, pour $\alpha(1) = 0.1$ (courbe en trait plein, $y^* \approx 4e-3$) et $\alpha(1) = 0.01$ (courbe en trait discontinu, $y^* \approx 3e-5$).*

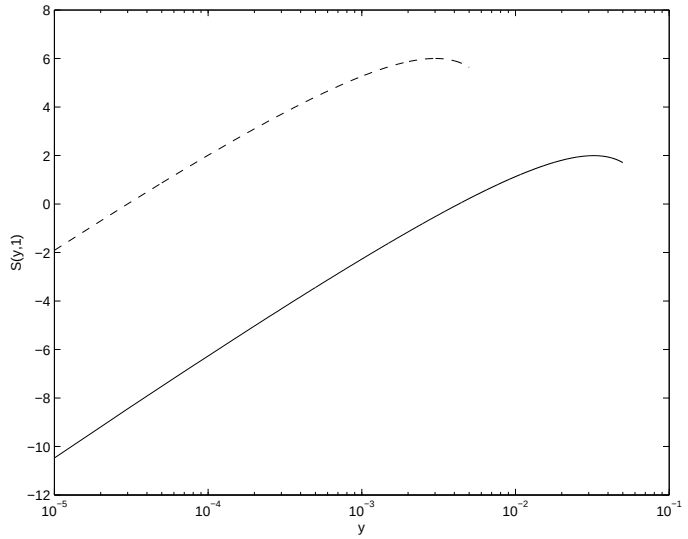

FIG. 5.2 – S(y,1) pour $\alpha(1) = 0.1$ (courbe en trait plein) et $\alpha(1) = 0.01$ (courbe en trait discontinu).

Influence des Mots de Code de Poids Informatif Deux

Si $\alpha(\sigma) = 0$, $P(x, \sigma)$ peut être approché par

$$P(x, \sigma) = \beta(\sigma) Q\left(\sqrt{2x}\right), \tag{5.3.13}$$

où

$$\beta(\sigma) = \frac{2}{I} \sum_{h} A_{2,h} \left[e^{-\frac{1}{2\sigma^2}}\right]^{h+2}, \tag{5.3.14}$$

à condition que $\beta(\sigma) \neq 0$ et $x \geq 0$, ce qui implique $P(x, \sigma) \leq \beta(\sigma)/2$. Dans ce cas, seulement l'effet des mots de code de poids informatif deux est pris en compte. Ce modèle correspond à des codes convolutifs récursifs systématiques quand $I \to \infty$ parce que le nombre de bits redondants générés par un seul bit d'information non-nul tend vers l'infini. En injectant (5.3.13) dans (5.2.2), les points fixes de la fonction décrivant le décodage itératif d'un PCC sont les solutions de

$$S(y, \sigma) = [Q^{-1}(y)]^2 - \left[Q^{-1}\left(\frac{y}{\beta(\sigma)}\right)\right]^2 - \frac{1}{\sigma^2} = 0. \tag{5.3.15}$$

On peut montrer que pour tout $\sigma > 0$, (5.3.15) n'a pas de solution dans l'intervalle $]0; \beta(\sigma)/2]$, à condition que $\beta(\sigma)$ soit suffisamment petit. De plus, nous avons $S(y, \sigma) \geq 0$ sur $]0; \beta(\sigma)/2]$, ce qui équivaut à $h(y, \sigma) \leq y$.

Exemple 5.3.6. *La Fig. 5.3 illustre $S(y, 1)$ sur l'intervalle $[0; \beta(1)/2]$ pour $\beta(1) = 0.1$ (courbe en trait plein) et $\beta(1) = 0.01$ (courbe en trait discontinu).*

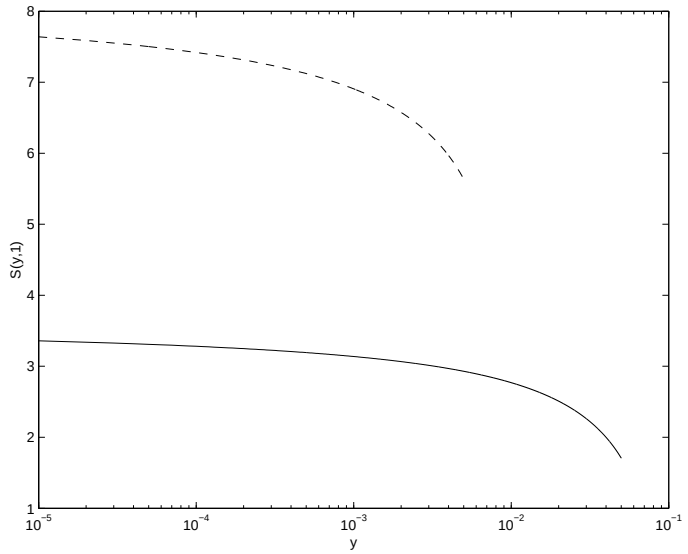

FIG. 5.3 – S(y,1) pour $\beta(1) = 0.1$ (courbe en trait plein) et $\beta(1) = 0.01$ (courbe en trait discontinu).

Cependant, en observant que h est donné par (D.0.1) en choisissant $A = B = 2$ et $\epsilon = \beta(\sigma)$, on montre dans l'Appendix D que $\lim_{y\to 0} h(y,\sigma) = 0$ et $\lim_{y\to 0} \frac{\partial h(y,\sigma)}{\partial y} = \left(\beta(\sigma)e^{\frac{1}{2\sigma^2}}\right)^2$. Il s'ensuit qu'à RSB élevé, $y = 0$ est un point fixe de h dont la stabilité est donnée par

$$\beta(\sigma)e^{\frac{1}{2\sigma^2}} < 1. \tag{5.3.16}$$

Ce résultat a déjà été obtenu dans [60].

Remarque 5.3.7. *Insistons sur le fait que les résultats, obtenus lorsqu'uniquement les mots de code de poids informatif un et deux sont pris en compte, supposent que les approximations du TEB sont valides, ce qui est vrai seulement quand I est grand. Si cette hypothèse n'est pas vérifiée, la méthode générale présentée au début de la Sec. 5.3.1 doit être appliquée pour déterminer la dynamique du décodage itératif près de zéro.*

5.3.2 Dynamique pour les SCC

Cas Général

Théorème 5.3.8. *Pour le code externe, une approximation du TEB pour les bits informatifs et redondants $P_o(x,\sigma)$ est obtenue à partir de la WEF $A^o(Z)$ comme suit*

$$P_o(x,\sigma) = \frac{1}{I}\sum_{d=d^o_{min}}^{I} dA^o_d Q\left(\sqrt{d\left(\frac{1}{\sigma^2}+x\right)}\right).$$

Pour le code interne, une approximation du TEB pour les bits informatifs $P_i(x,\sigma)$ est obtenue à partir de l'IRWEF $A^i(W,H)$ comme suit

$$P_i(x,\sigma) = \frac{1}{I}\sum_{w\geq 1}\sum_{h} wA^i_{w,h} Q\left(\sqrt{w\left(\frac{1}{\sigma^2}+x\right)+h\frac{1}{\sigma^2}}\right).$$

Démonstration. Premièrement, nous rappelons que la taille de l'entrelaceur I dans le contexte des SCC est égale au nombre total de bits générés par le code externe et au nombre de bits informatifs du code interne. Donc $P_i(x,\sigma)$ est simplement obtenu en remplaçant $A(W,H)$ par $A^i(W,H)$ dans le Thm. 5.3.1. En modifiant la preuve du Thm. 5.3.1 en prenant en compte le fait que de l'information *a priori* est disponible à la fois pour les bits informatifs et les bits redondants dans le code externe, on obtient l'expression de $P_o(x,\sigma)$. □

Exemple 5.3.9. *Supposons que le code convolutif* $(1, 5/7)$ *de rendement* $1/2$ *soit choisi comme code mère. Le code interne de rendement* $1/2$ *est obtenu en terminant le treillis du code mère. Le code externe de rendement* $2/3$ *est obtenu en terminant le treillis du code mère et en poinçonnant alternativement les bits redondants, comme il est proposé dans [61]. En choisissant* $k = 1024$*, on a* $I = \left\lceil \frac{1024+2}{2/3} \right\rceil = 1539$*. La Fig. 5.4 illustre* $h_o(y, 1.3)$ *(courbe en trait plein) et* $h_i(y, 1.3)$ *(courbe en trait discontinu). Une trajectoire de décodage itératif typique, rebondissant de gauche à droite entre les courbes* $z = h_o(y, 1.3)$*,* $z = h_i(y, 1.3)$ *et* $z = y$ *(courbe en trait discontinu et pointillés), est aussi représentée. Les trajectoires convergent vers un point fixe* $y^* \approx 5.9e - 11$*.*

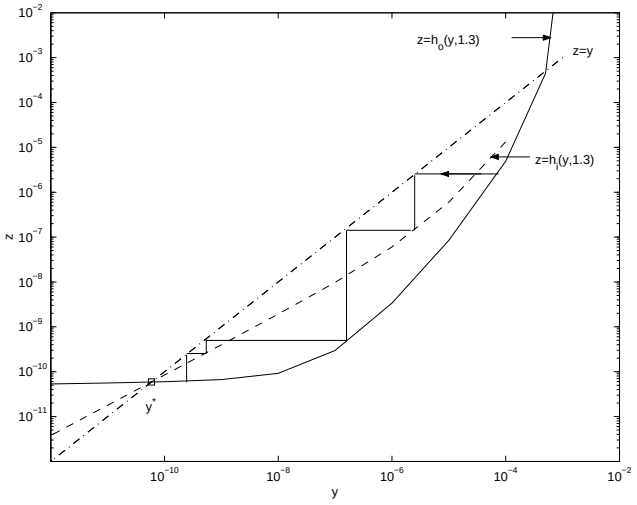

FIG. 5.4 – $h_o(y, 1.3)$ et $h_i(y, 1.3)$ pour un SCC de rendement $1/3$ utilisant des codes convolutifs terminés à 4 états et tel que $I = 1539$.

D'autres exemples du même type montrent que les caractéristiques typiques des fonctions h_o et h_i sont les suivantes

1. Un point fixe $y^* \ll 1$.
2. Une *région de tunnel* ouverte pour $h_i(y, \sigma)$, mais une convergence très rapide pour $h_o(y, \sigma)$, sauf si $d^o_{min} = 2$.
3. $\frac{1}{2} \leq \frac{\partial h_i(y,\sigma)}{\partial y} < 1$ et $0 \leq \frac{\partial h_o(y,\sigma)}{\partial y} \ll 1$, pour $y \to y^*$, sauf si $d^o_{min} = 2$.

Dans les paragraphes suivants, nous tentons d'expliquer qualitativement le comportement près de $y = 0$, en tant qu'une conséquence des mots de code de poids minimum dans les codes constitutifs.

Influence des Mots de Code de Poids Informatif Un dans le Code Interne

Lorsque la moyenne de l'information *a priori* $2x$ est grande, nous approchons $P_o(x,\sigma)$ et $P_i(x,\sigma)$ grâce à l'inégalité $Q(\sqrt{z+y}) \leq Q(\sqrt{z})e^{-y/2}$

$$\begin{aligned} P_o(x,\sigma) &= \alpha_o(\sigma) Q\left(\sqrt{d^o_{min} x}\right) \\ P_i(x,\sigma) &= \alpha_i(\sigma) Q\left(\sqrt{x}\right), \end{aligned} \tag{5.3.17}$$

où

$$\begin{aligned} \alpha_o(\sigma) &= \frac{d^o_{min} A^o_{d^o_{min}}}{I} \left[e^{-\frac{1}{2\sigma^2}}\right]^{d^o_{min}} \\ \alpha_i(\sigma) &= \frac{1}{I} \sum_h A^i_{1,h} \left[e^{-\frac{1}{2\sigma^2}}\right]^{h+1} \end{aligned} \tag{5.3.18}$$

à condition que $\alpha_i(\sigma) \neq 0$ et $x \geq 0$, ce qui implique $P_o(x,\sigma) \leq \alpha_o(\sigma)/2$ et $P_i(x,\sigma) \leq \alpha_i(\sigma)/2$. Dans ce cas, uniquement l'effet des mots de code de poids informatif un (resp. de poids d^o_{min}) est pris en compte dans le code interne (resp. externe). It résulte de (D.0.6) et (D.0.7) que h_o et h_i sont des fonctions croissantes de y. En injectant (5.3.17) dans (5.2.6), les points fixes des fonctions décrivant le décodage itératif d'un SCC sont les solutions de

$$S(y,\sigma) = [Q^{-1}(y)]^2 - \frac{1}{d^o_{min}} \left[Q^{-1}\left(\frac{y}{\alpha_o(\sigma)}\right)\right]^2 - \left[Q^{-1}\left(\frac{y}{\alpha_i(\sigma)}\right)\right]^2 - \frac{1}{\sigma^2} = 0. \tag{5.3.19}$$

On peut montrer que puisque $d^o_{min} > 1$, pour tout $\sigma > 0$, (5.3.19) n'a qu'une seule solution $y^* \in \,]0; \min\left(\alpha_o(\sigma), \alpha_i(\sigma)\right)/2]$ à condition que $\alpha_o(\sigma)$ et $\alpha_i(\sigma)$ soient suffisamment petits. De plus, nous avons $S(y,\sigma) \geq 0$ sur $]y^*; \min\left(\alpha_o(\sigma), \alpha_i(\sigma)\right)/2]$, ce qui équivaut à $h_o(y,\sigma) \leq y$ et $h_i(y,\sigma) \leq y$. On peut montrer que y^* croît avec $\alpha_o(\sigma)$ et $\alpha_i(\sigma)$ et décroît avec d^o_{min}. En utilisant (5.3.18), on a que $\alpha_o(\sigma)$ et $\alpha_i(\sigma)$ croissent avec σ, donc y^* augmente avec σ. Dans l'Annexe F, nous montrons que $\lim_{y^*\to 0} \frac{\partial h_o(y,\sigma)}{\partial y}\big|_{y=y^*} = 0$ and $\lim_{y^*\to 0} \frac{\partial h_i(y,\sigma)}{\partial y}\big|_{y=y^*} = 1 - \frac{1}{d^o_{min}}$. Nous concluons que les conditions 1 à 4 du Thm. 5.2.2 sont vérifiées sur un voisinage J de y^*. En terme de dynamique non-linéaire, à RSB élevé, $h_o \circ h_i$ et $h_i \circ h_o$ ont le même point fixe stable, qui augmente avec le paramètre de contrôle σ et décroît avec d^o_{min}.

Influence des Mots de Code de Poids Informatif Deux dans le Code Interne

Si l'approximation de $P_o(x,\sigma)$ proposée dans (5.3.17) est toujours valide, mais que $\alpha_i(\sigma)=0$, alors $P_i(x,\sigma)$ peut être approché par

$$P_i(x,\sigma)=\beta_i(\sigma)Q\left(\sqrt{2x}\right), \tag{5.3.20}$$

où

$$\beta_i(\sigma)=\frac{2}{I}\sum_h A^i_{2,h}\left[e^{-\frac{1}{2\sigma^2}}\right]^{h+2}, \tag{5.3.21}$$

à condition que $\beta^i(\sigma)\neq 0$ et $x\geq 0$, ce qui implique $P_i(x,\sigma)\leq \beta_i(\sigma)/2$. Dans ce cas, uniquement l'effet des mots de code de poids infomratif deux est pris en compte dans le code interne. Ce modèle correspond à un code interne convolutif récursif systématique quand $I\to\infty$ parce que le nombre de bits redondants générés par un seul bit d'information non-nul tend vers l'infini. Il résulte de (D.0.6) et (D.0.7) que h_o et h_i sont des fonctions croissantes de y. En injectant (5.3.20) dans (5.2.6), les points fixes des fonctions décrivant de décodage itératif d'un SCC sont les solutions de

$$S(y,\sigma)=[Q^{-1}(y)]^2-\frac{1}{d^o_{min}}\left[Q^{-1}\left(\frac{y}{\alpha_o(\sigma)}\right)\right]^2-\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\beta_i(\sigma)}\right)\right]^2-\frac{1}{\sigma^2}=0. \tag{5.3.22}$$

On peut montrer que puisque $d^o_{min}>1$, pout tout $\sigma>0$, (5.3.22) n'a pas de solution dans l'intervalle $]0;\min\left(\alpha_o(\sigma),\beta_i(\sigma)\right)/2]$, à condition que $\alpha_o(\sigma)$ et $\beta_i(\sigma)$ soient suffisamment petits.
De plus, nous avons $S(y,\sigma)\geq 0$ sur $]0;\min\left(\alpha_o(\sigma),\alpha_i(\sigma)\right)/2]$, ce qui équivaut à $h_o(y,\sigma)\leq y$ et $h_i(y,\sigma)\leq y$.

Nous étudions maintenant le comportement des fonctions au voisinage de zéro.

La fonction h_o est obtenue en multipliant (D.0.1) par $\alpha_o(\sigma)/\beta_i(\sigma)$ et en choisissant $A=d^o_{min}$, $B=2$ et $\epsilon=\beta_i(\sigma)$. De même, la fonction h_i est obtenue en multipliant (D.0.1) par $\beta_i(\sigma)/\alpha_o(\sigma)$ et en choisissant $A=2$, $B=d^o_{min}$ et $\epsilon=\alpha_o(\sigma)$. On montre dans Annexe D que $\lim_{y\to 0}h_o(y,\sigma)=\lim_{y\to 0}h_i(y,\sigma)=0$.

Cas $\mathbf{d^o_{min}=2}$

En utilisant (D.0.8), nous avons

$$\lim_{y\to 0}\frac{\partial h_o(y,\sigma)}{\partial y}=\alpha_o(\sigma)\beta_i(\sigma)\left(e^{\frac{1}{2\sigma^2}}\right)^2$$
$$\lim_{y\to 0}\frac{\partial h_i(y,\sigma)}{\partial y}=\alpha_o(\sigma)\beta_i(\sigma)\left(e^{\frac{1}{2\sigma^2}}\right)^2.$$

Nous en déduisons que les conditions 1 à 4 du Thm. 5.2.2 sont vérifiées en un voisinage J de 0. Il s'ensuit qu'à RSB élevé, $y = 0$ est un point fixe de $h_o \circ h_i$ et $h_i \circ h_o$ dont la condition de stabilité est donnée par

$$\sqrt{\alpha_o(\sigma)\beta_i(\sigma)}e^{\frac{1}{2\sigma^2}} < 1. \tag{5.3.23}$$

Cas $\mathbf{d}^o_{min} > 2$

De Annexe D.2.2.1, il suit que $\lim_{y\to 0} \frac{\partial h_o(y,\sigma)}{\partial y} = \lim_{y\to 0} \frac{\partial h_i(y,\sigma)}{\partial y} = 0$. Nous en déduisons que les conditions 1 à 4 du Thm. 5.2.2 sont vérifiées en un voisinage J de 0. Donc à RSB élevé, 0 est un point fixe superstable de $h_o \circ h_i$ et $h_i \circ h_o$.

Nous désirons caractériser la vitesse de convergence du processus de décodage itératif vers 0. Dans ce but, nous définnissons la *vitesse de convergence* de h_o par $\max\left\{r \in \mathbb{N} : \lim_{y\to 0} \frac{h_o(y,\sigma)}{y^r} = 0\right\}$ et la vitesse de convergence de h_i par $\max\left\{r \in \mathbb{N} : \lim_{y\to 0} \frac{h_i(y,\sigma)}{y^r} = 0\right\}$. Intuitivement, la vitesse de convergence mesure le nombre de digits de précision gagnés lors de la convergence vers le point fixe par une itération des fonctions. Dans l'Annexe D.2.2.2, nous montrons que la vitesse de convergence de h_o vaut $r(d^o_{min}, 2)$ et que la vitesse de convergence de h_i vaut $r(2, d^o_{min})$, où $r(.,.)$ est défini par (D.0.14). En particulier, lorsque $d^o_{min} \gg 1$, la vitesse de convergence de h_o vaut $\left\lceil \frac{d^o_{min}}{2} \right\rceil - 1$ et la vitesse de convergence de h_i vaut 1.

5.3.3 Dynamique pour les PC

Cas Général

Théorème 5.3.10. *Pour le code en ligne, une approximation du TEB des bits informatifs et redondants $P_R(x,\sigma)$ peut être obtenue à partir de la WEF $A^R(Z)$ comme suit*

$$P_R(x,\sigma) = \frac{1}{n_R} \sum_{d=d^R_{min}}^{n_R} dA^R_d Q\left(\sqrt{d\left(\frac{1}{\sigma^2} + x\right)}\right).$$

Pour le code en colonne, une approximation du TEB des bits informatifs et redondants $P_C(x,\sigma)$ peut être obtenue à partir de la WEF $A^C(Z)$ comme suit

$$P_C(x,\sigma) = \frac{1}{n_C} \sum_{d=d^C_{min}}^{n_C} dA^C_d Q\left(\sqrt{d\left(\frac{1}{\sigma^2} + x\right)}\right).$$

.

Démonstration. En modifiant la preuve du Thm. 5.3.1 en prenant en compte le fait que de l'information *a priori* est disponible à la fois pour les bits informatifs et

les bits redondants dans les lignes et les colonnes du PC, on obtient les expressions proposées de $P_R(x,\sigma)$ et de $P_C(x,\sigma)$. □

D'autres exemples du même type montrent que les caractéristiques typiques des fonctions h_R et h_C sont les suivantes

1. Un point fixe en 0.
2. Pas de *région de tunnel*, mais une convergence très rapide de h_R et h_C, pour y proche de 0, excepté quand $d^R_{min} = d^C_{min} = 2$.
3. $0 \leq \frac{\partial h_R(y,\sigma)}{\partial y} \ll 1$ et $0 \leq \frac{\partial h_C(y,\sigma)}{\partial y} \ll 1$, pour y proche de 0, excepté quand $d^R_{min} = d^C_{min} = 2$.

Dans le paragraphe suivant, nous tentons d'expliquer qualitativement le comportement près de $y = 0$, en tant qu'une conséquence des mots de code de poids minimum dans les codes constitutifs.

Influence des Mots de Code de Poids Minimum

Lorsque la moyenne de l'information *a priori* $2x$ est grande, nous approchons $P_R(x,\sigma)$ et $P_C(x,\sigma)$ par

$$\begin{aligned} P_R(x,\sigma) &= \alpha_R(\sigma) Q\left(\sqrt{d^R_{min} x}\right) \\ P_C(x,\sigma) &= \alpha_C(\sigma) Q\left(\sqrt{d^C_{min} x}\right), \end{aligned} \tag{5.3.24}$$

où

$$\begin{aligned} \alpha_R(\sigma) &= \frac{d^R_{min} A^R_{d^R_{min}}}{n_R} \left[e^{-\frac{1}{2\sigma^2}}\right]^{d^R_{min}} \\ \alpha_C(\sigma) &= \frac{d^C_{min} A^C_{d^C_{min}}}{n_C} \left[e^{-\frac{1}{2\sigma^2}}\right]^{d^C_{min}}, \end{aligned} \tag{5.3.25}$$

à condition que $x \geq 0$, ce qui implique $P_R(x,\sigma) \leq \alpha_R(\sigma)/2$ et $P_C(x,\sigma) \leq \alpha_C(\sigma)/2$. Dans ce cas, uniquement l'effet des mots de code de poids minimum dans les lignes et les colonnes est pris en compte. Il resulte de (D.0.6) et (D.0.7) que h_R et h_C sont des fonctions croissantes de y. En utilisant (5.3.24), les points fixes des fonctions décrivant le décodage itératif d'un PC sont les solutions de

$$S(y,\sigma) = [Q^{-1}(y)]^2 - \frac{1}{d^R_{min}}\left[Q^{-1}\left(\frac{y}{\alpha_R(\sigma)}\right)\right]^2 - \frac{1}{d^C_{min}}\left[Q^{-1}\left(\frac{y}{\alpha_C(\sigma)}\right)\right]^2 - \frac{1}{\sigma^2} = 0. \tag{5.3.26}$$

On peut montrer que parce que $d^R_{min} > 1$ et $d^R_{min} > 1$, pour tout $\sigma > 0$, (5.3.26) n'a qu'une seule solution dans l'intervalle $]0; \min(\alpha_R(\sigma), \alpha_C(\sigma))/2]$, à condition

que $\alpha_R(\sigma)$ et $\alpha_C(\sigma)$ soient suffisamment petits. De plus, nous avons $S(y,\sigma) \geq 0$ sur $]0; \min(\alpha_R(\sigma), \alpha_C(\sigma))/2]$, ce qui équivaut à $h_R(y,\sigma) \leq y$ et $h_C(y,\sigma) \leq y$.

Nous étudions maintenant le comportement des fonctions au voisinage de 0. La fonction h_R est obtenue en multipliant (D.0.1) par $\alpha_R(\sigma)/\alpha_C(\sigma)$ et en choisissant $A = d^R_{min}$, $B = d^C_{min}$ et $\epsilon = \alpha_C(\sigma)$. De même, la fonction h_C est obtenue en multipliant (D.0.1) par $\alpha_C(\sigma)/\alpha_R(\sigma)$ et en choisissant $A = d^C_{min}$, $B = d^R_{min}$ et $\epsilon = \alpha_R(\sigma)$. Nous montrons dans l'Annexe D que $\lim_{y\to 0} h_R(y,\sigma) = \lim_{y\to 0} h_C(y,\sigma) = 0$.

Cas $\mathbf{d^R_{min}} = d^C_{min} = 2$

D'après (D.0.8), nous avons

$$\lim_{y\to 0} \frac{\partial h_R(y,\sigma)}{\partial y} = \alpha_R(\sigma)\alpha_C(\sigma)\left(e^{\frac{1}{2\sigma^2}}\right)^2$$
$$\lim_{y\to 0} \frac{\partial h_C(y,\sigma)}{\partial y} = \alpha_R(\sigma)\alpha_C(\sigma)\left(e^{\frac{1}{2\sigma^2}}\right)^2.$$

Nous en déduisons que les conditions 1 à 4 du Thm. 5.2.2 sont vérifiées en un voisinage J de 0. Il s'ensuit qu'à RSB élevé, $y = 0$ est un point fixe de $h_R \circ h_C$ et $h_C \circ h_R$ dont la condition de stabilité est donnée par

$$\sqrt{\alpha_R(\sigma)\alpha_C(\sigma)}e^{\frac{1}{2\sigma^2}} < 1. \tag{5.3.27}$$

Cas $\max\left(d^R_{min}, d^C_{min}\right) > 2$

De l'Annexe D.2.2.1., il suit que $\lim_{y\to 0} \frac{\partial h_R(y,\sigma)}{\partial y} = \lim_{y\to 0} \frac{\partial h_C(y,\sigma)}{\partial y} = 0$. Nous en déduisons que les conditions 1 à 4 du Thm. 5.2.2 sont vérifiées en un voisinage J de 0. Donc, 0 est un point fixe superstable de $h_R \circ h_C$ et $h_C \circ h_R$.

Dans l'Annexe D.2.2.2, nous montrons que la vitesse de convergence de h_R vaut $r(d^R_{min}, d^C_{min})$ et que la vitesse de convergence de h_C vaut $r(d^C_{min}, d^R_{min})$. En particulier, lorsque $d^R_{min} \gg 1$ et $d^C_{min} \gg 1$, la vitesse de convergence de h_R vaut $d^R_{min} - 1$ et la vitesse de convergence de h_C vaut $d^C_{min} - 1$.

Remarque 5.3.11. *Dans le cas particulier où $d^R_{min} = d^C_{min} = 2$, nous pouvons supposer que les codes en ligne et en colonne sont des codes de parité. De cette façon, le rendement est maximal puisque tout code de distance minimum deux doit avoir au moins un bit de redondance. Dans un tel code produit, chaque bit vérifie exactement deux équations de parité, l'équation de parité correspondant respectivement à la ligne et à la colonne à laquelle il appartient. Par conséquent, ce code produit est un code LDPC irrégulier de distribution des degrés à gauche $\lambda(x) = x$ et de distribution des degrés à droite $\rho(x) = \frac{1}{2}(x^{n_R-1} + x^{n_C-1})$. La*

condition de stabilité du décodeur à passage de messages est alors [7]

$$\frac{(n_R-1)+(n_C-1)}{2} < e^{\frac{1}{2\sigma^2}}. \tag{5.3.28}$$

En rappelant que pour un code de parité, nous avons

$$\begin{cases} \alpha_R(\sigma) = \frac{2}{n_R}\binom{n_R}{2}\left[e^{-\frac{1}{2\sigma^2}}\right]^2 \\ \alpha_C(\sigma) = \frac{2}{n_C}\binom{n_C}{2}\left[e^{-\frac{1}{2\sigma^2}}\right]^2, \end{cases}$$

le décodeur itératif qui consiste en un décodage SISO successif des lignes et des colonnes a une condition de stabilité donnée par (5.3.27), soit

$$\sqrt{(n_R-1)(n_C-1)} < e^{\frac{1}{2\sigma^2}}.$$

Cette condition est compatible avec (5.3.28) si l'on considère que les moyennes arithmétique et géométrique de n_R-1 et n_C-1 sont approximativement les mêmes. Notons que puisque la moyenne géométrique est majorée par la moyenne arithmétique, la stabilité du décodeur ligne/colonne est légèrement meilleure que la stabilité du décodeur à passage de messages lorsque $n_R \neq n_C$.

Nous insistons sur le fait que la dynamique du décodeur itératif au voisinage de 0 est une conséquence des mots de code de poids 2 dans les codes constitutifs, bien que le code produit lui-même ne contienne aucun mot de code de poids 2, puisque sa distance minimum vaut $d^R_{min} \times d^C_{min} = 4$ [13].

5.4 Conclusions

Nous avons présenté un modèle analytique approché du décodage itératif des codes concaténés. En supposant que la densité de probabilité de l'information *extrinsèque* peut être approchée par une gaussienne, nous avons décrit l'évolution du TEB à la sortie des décodeurs constitutifs comme les itérées d'une fonction non-linéaire pour le canal gaussien à entrée binaire.

A RSB élevé, nous avons montré comment les points fixes du système de décodage itératif et leur stabilité dépendent des paramètres des codes constitutifs.

Chapitre 6

Propriétés en Terme de Distance des Codes LDPC

Dans ce chapitre, nous étudions les propriétés en terme de distance des codes LDPC en utilisant la construction de l'ensemble proposée par Richardson *et al.* [6]. En premier lieu, nous montrons que pour les codes LDPC réguliers, les propriétés en terme de distance de l'ensemble de Gallager (voir Sec. 2.2.2) et de l'ensemble de Richardson *et al.* sont identiques. Ensuite, nous étudions les propriétés en terme de distance des codes LDPC irréguliers qui ont reçu peu d'attention jusqu'à présent. En particulier, nous interprétons la condition de stabilité de l'évolution de densité en tant qu'une condition suffisante pour que la probabilité d'erreur due aux mots de code de poids faible tende vers zéro.

6.1 Construction de l'Ensemble de Richardson *et al.*

La construction de l'ensemble des codes LDPC proposée par Richardson *et al.* [6] consiste à connecter aléatoirement les noeuds de variable aux noeuds de parité dans le graphe bipartite du code (voir Sec. 2.3.1), ainsi que la Fig. 6.1 l'illustre.

6.2 Propriétés en Terme de Distance des Codes LDPC Réguliers

Les propriétés en terme de distance des codes LDPC réguliers ont déjà été présentées dans la Sec. 2.2.2 pour l'ensemble de Gallager. Bien que l'ensemble de Gallager soit seulement un sous-ensemble de l'ensemble de Richardson *et al.*,

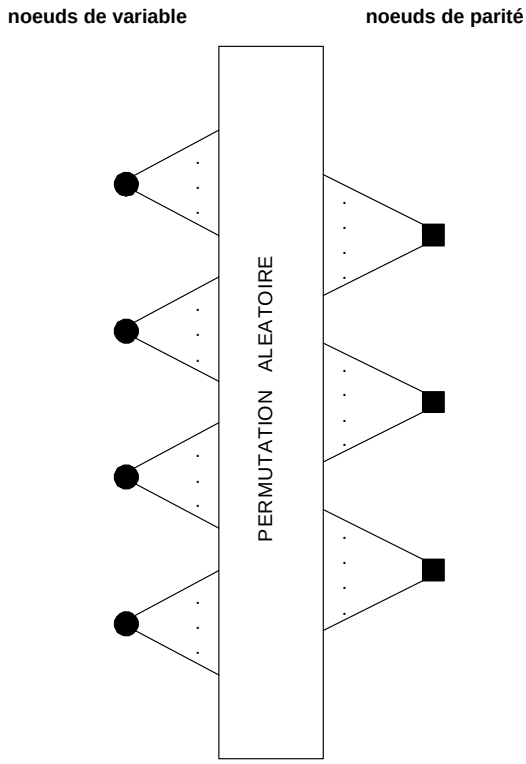

FIG. 6.1 – Construction de l'ensemble des codes LDPC par Richardson *et al.*.

nous allons montrer que les propriétés en terme de distance sont les mêmes.

Considérons l'ensemble des (n, d_v, d_c) codes LDPC réguliers défini par la construction de Richardson *et al.*. Par commodité, nous introduisons le code en bloc C qui consiste en la concaténation de $d_v n/d_c$ codes de parité de longueur d_c. De manière évidente, C correspond à toutes les équations de parité représentées sur la partie droite de la Fig. 6.1. Considérons à nouveau le code LDPC. Si chaque noeud de variable dans un mot de code de longueur n est répété $d_v - 1$ fois, les $d_v n$ digits résultants, après permutation aléatoire, doivent satisfaire les contraintes imposées par le code en bloc C. Soit $N(d)$, pour $d = 0, \ldots, nd_v$, le nombre de mots de code de poids d dans le code C. La probabilité qu'une séquence quelconque de longueur nd_v et de poids d soit un mot de code de C après permutation aléatoire vaut

$$\frac{N(d)}{\binom{d_v n}{d}}.$$

C'est pourquoi la probabilité qu'une séquence quelconque de longueur nd_v et de poids l soit un mot de code dans l'ensemble des codes LDPC vaut

$$P(l) = \frac{N(d_v l)}{\binom{d_v n}{d_v l}}.$$

En considérant que le nombre total de séquences de longueur nd_v et de poids l est $\binom{n}{l}$, le nombre moyen de mots de code de poids l dans l'ensemble des codes LDPC vaut

$$\overline{N}(l) = \binom{n}{l} P(l).$$

Théorème 6.2.1. *Le nombre moyen de mots de code de poids l dans l'ensemble des (n, d_v, d_c) codes LDPC réguliers défini par la construction de Richardson* et al. *est donné par*

$$\overline{N}(l) \leq C(\delta, n) \exp(-nB(\delta, d_v, d_c)), \tag{6.2.1}$$

où

$$\begin{aligned}
C(\delta, n) &= \sqrt{d_v} \exp\left[\frac{1}{12n}\left(1 + \frac{1}{d_v\delta(1-\delta)}\right)\right] \\
B(\delta, d_v, d_c) &= (d_v - 1)H(\delta) - \frac{d_v}{d_c}\mu_{d_c}(s) + d_v s\delta \\
\mu_{d_c}(s) &= \ln \frac{(1+e^s)^{d_c} + (1-e^s)^{d_c}}{2} \\
\delta &= \frac{\mu'_{d_c}(s)}{d_c} \\
H(\delta) &= -\delta \ln \delta - (1-\delta)\ln(1-\delta).
\end{aligned}$$

Notons que comme l'exposant $B(\delta, d_v, d_c)$ est le même que pour l'ensemble de Gallager, les propriétés en terme de distance sont les mêmes. En particulier, la plupart des codes dans l'ensemble ont une distance minimum proche de ou supérieure à $\delta_0 n$ où δ_0 est l'unique solution de $B(\delta, d_v, d_c) = 0$, pour $0 \leq \delta \leq 1/2$. De plus, lorsque d_c tend vers l'infini, δ_0 tend vers la borne de Gilbert-Varshamov qui correspond au rapport de distance minimum typique pour l'ensemble équiprobable des codes (voir Sec. 2.2.2).

Démonstration. Tout d'abord, en utilisant la formule de Stirling

$$(2\pi n)^{1/2} n^n \exp(-n) \leq n! \leq (2\pi n)^{1/2} n^n \exp\left(-n + \frac{1}{12n}\right)$$

nous avons

$$\binom{n}{l} = \binom{n}{n\delta} \leq [2\pi n\delta(1-\delta)]^{-\frac{1}{2}} \exp\left(nH(\delta) + \frac{1}{12n}\right), \tag{6.2.2}$$

où $\delta = l/n$ and $H(\delta) = -\delta \ln \delta - (1-\delta)\ln(1-\delta)$. De même, nous avons

$$\binom{d_v n}{d_v l}^{-1} = \binom{d_v n}{d_v n\delta} \leq [2\pi n d_v \delta(1-\delta)]^{\frac{1}{2}} \exp\left(-nd_v H(\delta) + \frac{1}{12nd_v\delta(1-\delta)}\right). \tag{6.2.3}$$

L'énumérateur de poids d'un code de parité de longueur j s'écrit [4]

$$A_j(Z) = \frac{(1+Z)^j + (1-Z)^j}{2}.$$

Définissons $g_j(s) = A_j(Z = e^s)$ et $\mu_j(s) = \ln g_j(s)$. Il suit que l'énumérateur de poids du code en bloc C s'écrit

$$\sum_{d=0}^{nd_v} N(d)e^{sd} = g_{d_c}(s)^{d_v n/d_c}. \tag{6.2.4}$$

La partie gauche de (6.2.4) peut être minorée par $N(d)e^{sd}$ quels que soient s et d, de sorte que

$$N(d) \leq \exp\left(nd_v \frac{\mu_{d_c}(s)}{d_c} - sd\right). \tag{6.2.5}$$

L'exposant dans (6.2.5) est minimal quand

$$d = nd_v \frac{\mu'_{d_c}(s)}{d_c}. \tag{6.2.6}$$

Il résulte de (6.2.5) et (6.2.6) que

$$N(d_v l) \leq \exp\left[nd_v\left(\frac{\mu_{d_c}(s)}{d_c} - s\delta\right)\right] \tag{6.2.7}$$

et l'exposant est minimisé quand

$$\delta = \frac{\mu'_{d_c}(s)}{d_c}. \tag{6.2.8}$$

Finalement, en combinant (6.2.2) et (6.2.3) avec (6.2.7) la preuve est achevée. □

6.3 Propriétés en Terme de Distance des Codes LDPC Irréguliers

Considérons l'ensemble des (n, λ, ρ) codes LDPC irréguliers défini par le construction de Richardson *et al.*. Le polynôme de distribution des degrés des noeuds de variable (resp. de parité) $\lambda(x)$ (resp. $\rho(x)$) est de degré d_v (resp. d_c). Notons $n_i = a_i n$ le nombre de noeuds de variable de degré i, nous avons

$$a_i = \frac{\lambda_i/i}{\sum_i \lambda_i/i}.$$

Sauf mention expresse du contraire, les notations $\sum_i$, $\prod_i$, $\min_i$ et $\max_i$ désignent une somme, un produit, une minimisation et une maximisation portant sur tous les indices i allant de 2 à d_v, pour lesquels $\lambda_i \neq 0$. De même, $\sum_j$ et $\prod_j$ désignent une somme et un produit portant sur tous les indices j allant de 2 à d_c, pour lesquels $\rho_j \neq 0$.

Rappelons que $(\rho_j/j)(\sum_i in_i)$ est le nombre de noeuds de parité de degré j dans le graphe bipartite du code LDPC. Par commodité, nous introduisons le code en bloc C qui consiste en la concaténation de $(\rho_j/j)(\sum_i in_i)$ codes de parité de longueur j, pour tous les $j = 2, \ldots, d_c$. De manière évidente, C correspond à toutes les équations de parité représentées sur la partie droite de la Fig. 6.1. Considérons à nouveau le code LDPC. Si chaque noeud de variable de degré i dans un mot de code de longueur n est répété $i-1$ fois, le nombre de digits résultant vaut

$$\sum_i in_i = \frac{n}{\sum_i \lambda_i/i}$$

Nous appellerons cette opération une *répétition irrégulière* par rapport à la distribution λ. Après permutation aléatoire, ces digits doivent satisfaire les contraintes imposées par le code en bloc C. Soit $N(d)$, pour $d = 0, \ldots, \sum_i in_i$, le nombre de mots de code de poids d dans C. Considérons une séquence de longueur n comportant l_i uns et $n_i - l_i$ zéros dans les positions correspondant à des noeuds de variable de degré i, pour tous les i allant de 2 to d_v pour lesquels $\lambda_i \neq 0$. Les contraintes $0 \leq l_i \leq n_i$, pour $i = 2, \ldots, d_v$, doivent être satisfaites et le nombre de ces séquences est

$$\prod_i \binom{n_i}{l_i}.$$

Après *répétition irrégulière* par rapport à λ et permutation aléatoire, la probabilité qu'une séquence quelconque de ce type soit un mot de code de C vaut

$$\frac{N\left(\sum_i il_i\right)}{\binom{\sum_i in_i}{\sum_i il_i}}.$$

Par conséquent, le nombre moyen de mots de code de poids l dans l'ensembles des (n, λ, ρ) codes LDPC irréguliers s'écrit

$$\overline{N}(l) = \sum_{l_i : \sum_i l_i = l} \left[\frac{N\left(\sum_i il_i\right)}{\binom{\sum_i in_i}{\sum_i il_i}} \prod_i \binom{n_i}{l_i} \right]. \tag{6.3.9}$$

La somme dans (6.3.9) porte sur toutes les décompositions possibles des l bits non-nuls en l_i bits non-nuls situés dans les positions correspondant à des noeuds de variable de degré i.

Definition 6.3.1. $\boldsymbol{\alpha} = (\alpha_2, \ldots, \alpha_{d_v})$ *est une distribution si et seulement si*

1. $0 \leq \alpha_i \leq 1$, *pour* $i = 2, \ldots, d_v$.
2. $\alpha_i = 0$ *quand* $\lambda_i = 0$, *pour* $i = 2, \ldots, d_v$.
3. $\sum_i \alpha_i = 1$.

Maintenant, soit $\delta = l/n$ et introduisons une distribution $\boldsymbol{\alpha}$ telle que $l_i = n\alpha_i\delta$, pour $i = 2, \ldots, d_v$. On vérifie aisément que $\sum_i l_i = l$. De plus, les contraintes $0 \leq l_i \leq n_i$, pour $i = 2, \ldots, d_v$, imposent $0 \leq \delta \leq \min_i \left(\frac{a_i}{\alpha_i}\right)$. Alors nous pouvons réécrire (6.3.9) comme

$$\overline{N}(l) = \sum_{\boldsymbol{\alpha}} \left[\frac{N\left(n\delta \sum_i i\alpha_i\right)}{\binom{n\sum_i ia_i}{n\delta\sum_i i\alpha_i}} \prod_i \binom{na_i}{n\alpha_i\delta} \right], \tag{6.3.10}$$

où la somme porte sur toutes les distributions possibles $\boldsymbol{\alpha}$.

Théorème 6.3.2. *En supposant que les distributions des degrés* λ *et* ρ *soient fixées, le nombre moyen de mots de code de poids* l, *dans l'ensemble des* (n, λ, ρ) *codes LDPC irréguliers défini par la construction de Richardson* et al, *est donné par*

$$\overline{N}(l) \leq \sum_{\boldsymbol{\alpha}} C(\delta, \boldsymbol{\alpha}, n) \exp(-nB(\delta, \boldsymbol{\alpha})), \tag{6.3.11}$$

où

$$\begin{aligned}
C(\delta, \boldsymbol{\alpha}, n) &= \left[\frac{2\pi n\delta(\sum_i i\alpha_i)\left(\sum_i i(a_i - \alpha_i\delta)\right)/(\sum_i ia_i)}{\prod_i (2\pi n\alpha_i\delta(a_i - \alpha_i\delta)/a_i)} \right]^{1/2} \\
&\quad \times \exp\left[\frac{1}{12n} \left(\frac{1}{\delta\sum_i i\alpha_i} + \frac{1}{\sum_i i(a_i - \alpha_i\delta)} + \sum_i \frac{1}{a_i} \right) \right] \\
B(\delta, \boldsymbol{\alpha}) &= \Big(\sum_i ia_i\Big) H\left(\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta \right) - \sum_i a_i H\left(\frac{\alpha_i}{a_i}\delta\right) \\
&\quad - \Big(\sum_i ia_i\Big) \sum_j \frac{\rho_j}{j}\mu_j(s) + s\delta \sum_i i\alpha_i \\
\delta &= \frac{\sum_i ia_i}{\sum_i i\alpha_i} \sum_j \frac{\rho_j}{j}\mu_j'(s).
\end{aligned}$$

Démonstration. Tout d'abord, en utilisant la formule de Stirling, on a

$$\binom{na_i}{n\alpha_i\delta} \leq [2\pi n\alpha_i\delta(a_i - \alpha_i\delta)/a_i]^{-\frac{1}{2}} \exp\left[na_i H\left(\frac{\alpha_i}{a_i}\delta\right) + \frac{1}{12na_i} \right] \tag{6.3.12}$$

De même, on peut montrer que

$$\binom{n\sum_i ia_i}{n\delta\sum_i i\alpha_i}^{-1} \leq \left[2\pi n\delta\Big(\sum_i i\alpha_i\Big)\left(\sum_i i(a_i-\alpha_i\delta)\right)/\Big(\sum_i ia_i\Big)\right]^{\frac{1}{2}} \times \exp\left[-n\Big(\sum_i ia_i\Big)H\left(\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta\right)+\frac{1}{12n}\left(\frac{1}{\delta\sum_i i\alpha_i}+\frac{1}{\sum_i i(a_i-\alpha_i\delta)}\right)\right]. \tag{6.3.13}$$

L'énumérateur de poids du code en bloc C s'écrit

$$\sum_{d=0}^{n\sum_i ia_i} N(d)e^{sd} = \prod_j g_j(s)^{(\rho_j/j)(n\sum_i ia_i)}. \tag{6.3.14}$$

La partie gauche de (6.3.14) peut être minorée par $N(d)e^{sd}$ quels que soient s et d, de sorte que

$$N(d) \leq \exp\left(\Big(n\sum_i ia_i\Big)\sum_j \frac{\rho_j}{j}\mu_j(s) - sd\right). \tag{6.3.15}$$

L'exposant dans (6.3.15) est minimal quand

$$d = \Big(n\sum_i ia_i\Big)\sum_j \frac{\rho_j}{j}\mu_j'(s). \tag{6.3.16}$$

Il résulte de (6.3.15) et (6.3.16) que

$$N\Big(n\delta\sum_i i\alpha_i\Big) \leq \exp\left[-n\left(-\Big(\sum_i ia_i\Big)\sum_j \frac{\rho_j}{j}\mu_j(s) + s\delta\sum_i i\alpha_i\right)\right] \tag{6.3.17}$$

et l'exposant est minimisé quand

$$\delta = \frac{\sum_i ia_i}{\sum_i i\alpha_i}\sum_j \frac{\rho_j}{j}\mu_j'(s). \tag{6.3.18}$$

Finalement, en combinant (6.3.12) et (6.3.13) avec (6.3.17) la preuve est achevée. □

A l'aide des deux théorèmes suivants, nous analysons la probabilité d'erreur due aux mots de code poids faible.

Théorème 6.3.3. *L'exposant $B(\delta, \boldsymbol{\alpha})$ vérifie*

1. *Pour tout λ, ρ et $\boldsymbol{\alpha}$*

$$\lim_{\delta\to 0} B(\delta, \boldsymbol{\alpha}) = 0.$$

2. Pour tout λ, ρ

$$\lim_{\delta\to 0}\frac{\partial B(\delta,\boldsymbol{\alpha})}{\partial\delta}=\begin{cases}-\ln\left(\lambda'(0)\rho'(1)\right) & \textit{si } \alpha_2=1 \textit{ et } \alpha_i=0, \textit{ pour } i>2\\ +\infty & \textit{sinon.}\end{cases}$$

Notons que la distribution $\boldsymbol{\alpha}$ peut être définie par $\alpha_2 = 1$ et $\alpha_i = 0$, pour $i > 2$ seulement lorsque $\lambda_2 \neq 0$.

Démonstration. 1. Nous commençons par calculer $B(\delta, \boldsymbol{\alpha})$ quand $\delta \to 0$. Premièrement, considérons l'expression de δ trouvée dans le Thm. 6.3.2. Alors $\delta \to 0$ est équivalent à $s \to -\infty$ et pour $j \geq 2$

$$\lim_{s\to-\infty} s\mu_j'(s)=\lim_{s\to-\infty}\frac{se^s\left[(1+e^s)^{j-1}-(1-e^s)^{j-1}\right]}{(1+e^s)^j+(1-e^s)^j}=0.$$

Il s'ensuit que $\lim_{s\to-\infty} s\delta\sum_i i\alpha_i = 0$.

De plus, pour $j \geq 2$, $\lim_{s\to-\infty}\mu_j(s) = 0$ et $\lim_{x\to 0} H(x) = 0$. en combinant ces résultats, on obtient $\lim_{\delta\to 0} B(\delta, \boldsymbol{\alpha}) = 0$, $\forall(\lambda, \rho, \boldsymbol{\alpha})$.

2. Nous démontrons maintenant la seconde partie du théorème. Considérons l'expression de $B(\delta, \boldsymbol{\alpha})$ trouvée dans le Thm. 6.3.2, nous obtenons

$$\begin{aligned}\frac{\partial B(\delta,\boldsymbol{\alpha})}{\partial\delta}=&-\sum_i\alpha_i H'\left(\frac{\alpha_i}{a_i}\delta\right)+\Big(\sum_i i\alpha_i\Big)H'\left(\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta\right)+s\Big(\sum_i i\alpha_i\Big)\\&+\frac{\partial s}{\partial\delta}\left[-\Big(\sum_i ia_i\Big)\sum_j\frac{\rho_j}{j}\mu_j'(s)+\Big(\sum_i i\alpha_i\Big)\delta\right].\end{aligned}$$

En utilisant l'expression de δ trouvée dans le Thm. 6.3.2, nous obtenons la simplification suivante

$$\frac{\partial B(\delta,\boldsymbol{\alpha})}{\partial\delta}=-\sum_i\alpha_i\ln\frac{1-\frac{\alpha_i}{a_i}\delta}{\frac{\alpha_i}{a_i}\delta}+\Big(\sum_i i\alpha_i\Big)\ln\frac{1-\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta}{\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta}+s\Big(\sum_i i\alpha_i\Big). \tag{6.3.19}$$

En procédant à la substitution

$$z=\frac{1-e^s}{1+e^s},$$

$\delta \to 0$ devient équivalent à $z \to 1$. Dans l'Annexe H, nous montrons que

(6.3.19) est de la forme

$$\frac{\partial B(\delta,\boldsymbol{\alpha})}{\partial\delta} = \ln\left[\frac{1-z}{1+z}\left(\frac{1+z^{d_c-1}}{1-z^{d_c-1}}\right)^{\left(\sum_i i\alpha_i\right)-1}\right] - \left(\left(\sum_i i\alpha_i\right)-1\right)\ln(1+\epsilon_1(z)) + \left(\sum_i i\alpha_i\right)\ln(1-\epsilon_2(z)) - \sum_i \alpha_i \ln(\Gamma_i(z)-\epsilon_2(z)), \tag{6.3.20}$$

avec

$$\begin{aligned}
&\lim_{z\to 1}\epsilon_1(z) = -\frac{1}{d_c-1}\sum_{j<d_c}\rho_j(d_c-j)\\
&\lim_{z\to 1}\epsilon_2(z) = 0\\
&\lim_{z\to 1}\Gamma_i(z) = \frac{a_i}{\alpha_i}\frac{\sum_i i\alpha_i}{\sum_i i a_i}.
\end{aligned}$$

Par conséquent, nous avons

$$\begin{aligned}
\lim_{z\to 1}\ln(1+\epsilon_1(z)) &= -\ln(d_c-1) + \ln\Big(d_c - 1 - \sum_{j<d_c}\rho_j(d_c-j)\Big)\\
&= -\ln(d_c-1) + \ln\Big(d_c - 1 - d_c\sum_{j<d_c}\rho_j + \sum_{j<d_c}\rho_j j\Big)\\
&= -\ln(d_c-1) + \ln\Big(-1 + d_c\Big(1-\sum_{j<d_c}\rho_j\Big) + \sum_{j<d_c}\rho_j j\Big)\\
&= -\ln(d_c-1) + \ln\Big(-1 + \sum_{j}\rho_j j\Big)\\
&= -\ln(d_c-1) + \ln\Big(\sum_{j}\rho_j(j-1)\Big)\\
&= -\ln(d_c-1) + \ln\rho'(1).
\end{aligned}$$

Pour la distribution particulière $\boldsymbol{\alpha}$ définie par $\alpha_2 = 1$ et $\alpha_i = 0$, pour $i > 2$

$$\begin{aligned}
&\lim_{z\to 1}\Gamma_2(z) = \lambda_2 = \lambda'(0)\\
&\lim_{z\to 1}\sum_i \alpha_i \ln(\Gamma_i(z)-\epsilon_2(z)) = \ln\lambda'(0).
\end{aligned}$$

Pour la distribution particulière $\boldsymbol{\alpha}$ définie par $\alpha_2 = 1$ et $\alpha_i = 0$, pour $i > 2$ nous avons $\sum_i i\alpha_i = 2$. Pour toute autre distribution, $\sum_i i\alpha_i > 2\sum_i \alpha_i = 2$. Maintenant, en utilisant l'identité $(1-z^n) = (1-z)(1+z+\cdots+z^{n-1})$,

nous obtenons

$$\begin{aligned}
&\lim_{z\to 1} \ln \left[\frac{1-z}{1+z} \left(\frac{1+z^{d_c-1}}{1-z^{d_c-1}} \right)^{\left(\sum_i i\alpha_i\right)-1} \right] \\
&= \lim_{z\to 1} \ln \left\{ \frac{(1+z^{d_c-1})^{\left(\sum_i i\alpha_i\right)-1}}{(1-z^{d_c-1})^{\left(\sum_i i\alpha_i\right)-2}} \frac{1}{1+z} \frac{1}{1+z+\cdots+z^{d_c-2}} \right\} \\
&= \begin{cases} -\ln(d_c-1) & \text{si } \alpha_2 = 1 \text{ et } \alpha_i = 0, \text{ pour } i > 2 \\ +\infty & \text{sinon.} \end{cases}
\end{aligned}$$

En combinant ces limites, on obtient immédiatement le second résultat du théorème. □

Théorème 6.3.4. *Considérons l'ensemble des (n, λ, ρ) avec λ et ρ choisis de manière arbitraire. Pour une modulation BPSK sur les canaux AWGN et à évanouissement de Rayleigh, une condition suffisante pour que la probabilité d'erreur due aux mots de code de poids faible tende vers zéro quand n tend vers l'infini, est donnée par la condition de stabilité de l'évolution de densité.*

Démonstration. Soient σ, r et E_b/N_0 l'écart-type du bruit gaussien, le rendement de codage et le rapport énergie par bit sur densité spectrale de puissance du bruit, respectivement. Nous avons

$$\sigma^2 = \frac{1}{2rE_b/N_0}.$$

Supposons que les bits encodés soient transmis par modulation BPSK. Alors, sur le canal AWGN, la probabilité de détecter un mot de code qui diffère du mot de code transmis en l positions binaires vaut [24]

$$P(l) = Q\left(\sqrt{\frac{2lrE_b}{N_0}}\right) \leq \left(e^{-\frac{1}{2\sigma^2}}\right)^l.$$

La borne par réunion de la probabilité d'erreur des mots de code P_w, correspondant aux mots de code de poids faible, est obtenue pour $\delta = l/n \to 0$ comme suit

$$\begin{aligned}
P_w &\leq \lim_{\delta\to 0} \sum_{\boldsymbol{\alpha}} C(\delta, \boldsymbol{\alpha}, n) \exp(-nB(\delta, \boldsymbol{\alpha})) \left(e^{-\frac{1}{2\sigma^2}}\right)^{n\delta} \\
&\leq \sum_{\boldsymbol{\alpha}} \lim_{\delta\to 0} C(\delta, \boldsymbol{\alpha}, n) \exp\left[-n\left(B(\delta, \boldsymbol{\alpha}) + \frac{\delta}{2\sigma^2}\right)\right].
\end{aligned}$$

Une condition suffisante pour que $P_w \to 0$, lorsque $n \to \infty$ est donc

$$\lim_{\delta\to 0} B(\delta, \boldsymbol{\alpha}) + \frac{\delta}{2\sigma^2} > 0, \quad \forall \boldsymbol{\alpha}.$$

Cette condition peut être réécrite de la manière suivante

$$\max_{\boldsymbol{\alpha}} \lim_{\delta \to 0} \left(-\frac{B(\delta, \boldsymbol{\alpha})}{\delta} \right) < \frac{1}{2\sigma^2}. \tag{6.3.21}$$

D'après la première partie du Thm. 6.3.3, nous avons $\lim_{\delta \to 0} B(\delta, \boldsymbol{\alpha}) = 0$, par conséquent

$$\max_{\boldsymbol{\alpha}} \lim_{\delta \to 0} \left(-\frac{B(\delta, \boldsymbol{\alpha})}{\delta} \right) = \max_{\boldsymbol{\alpha}} \lim_{\delta \to 0} \left(-\frac{\partial B(\delta, \boldsymbol{\alpha})}{\partial \delta} \right).$$

Finalement, en utilisant la deuxième partie du Thm. 6.3.3, (6.3.21) est bien équivalent à la condition de stabilité de l'évolution de densité [7], à savoir

$$\lambda'(0)\rho'(1) < e^{\frac{1}{2\sigma^2}}.$$

De même, sur le canal de Rayleigh sans mémoire, la probabilité de détecter un mot de code qui diffère du mot de code transmis en l positions binaires est majorée par [24]

$$P(l) \leq \left(\frac{1}{1 + \frac{1}{2\sigma^2}} \right)^l.$$

La borne par réunion de la probabilité d'erreur des mots de code P_w, correspondant aux mots de code de poids faible, est obtenue pour $\delta = l/n \to 0$ comme suit

$$\begin{aligned} P_w &\leq \lim_{\delta \to 0} \sum_{\boldsymbol{\alpha}} C(\delta, \boldsymbol{\alpha}, n) \exp(-nB(\delta, \boldsymbol{\alpha})) \left(\frac{1}{1 + \frac{1}{2\sigma^2}} \right)^{n\delta} \\ &\leq \sum_{\boldsymbol{\alpha}} \lim_{\delta \to 0} C(\delta, \boldsymbol{\alpha}, n) \exp\left[-n \left(B(\delta, \boldsymbol{\alpha}) + \delta \ln \left(1 + \frac{1}{2\sigma^2} \right) \right) \right]. \end{aligned}$$

Une condition suffisante pour que $P_w \to 0$, lorsque $n \to \infty$ est donc

$$\lim_{\delta \to 0} B(\delta, \boldsymbol{\alpha}) + \delta \ln \left(1 + \frac{1}{2\sigma^2} \right) > 0, \quad \forall \boldsymbol{\alpha}.$$

En utilisant le Thm. 6.3.3, cette condition devient équivalente à la condition de stabilité de l'évolution de densité [10], à savoir

$$\lambda'(0)\rho'(1) < 1 + \frac{1}{2\sigma^2}.$$

□

Nous concluons ce chapitre avec une remarque à propos de la comparaison entre l'ensemble des codes LDPC irréguliers et l'ensemble aléatoire des codes.

Definition 6.3.5. *La distribution des degrés à droite* $\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}$ *est*

concentrée sur un nombre fini de termes s'il existe un entier fini F tel que ρ_j soit non-nul que pour $j = d_c - F + 1, \ldots, d_c$.

Remarque 6.3.6. *Nous pouvons décomposer l'exposant obtenu dans le Thm. 6.3.11 de la manière suivante*

$$B(\delta, \boldsymbol{\alpha}) = -\sum_i a_i H\left(\frac{\alpha_i}{a_i}\delta\right) + \Big(\sum_i i a_i\Big) \sum_j \frac{\rho_j}{j} \ln 2 + R(\delta, \boldsymbol{\alpha}),$$

où

$$R(\delta, \boldsymbol{\alpha}) = \Big(\sum_i i a_i\Big) H\left(\frac{\sum_i i\alpha_i}{\sum_i i a_i}\delta\right) - \Big(\sum_i i a_i\Big) \sum_j \frac{\rho_j}{j} \ln\left[(1+e^s)^j + (1-e^s)^j\right] + s\delta \sum_i i\alpha_i.$$

On peut montrer que si ρ est concentré sur un nombre fini de termes, $R(\delta, \boldsymbol{\alpha}) \to 0$, lorsque $d_c \to \infty$. En appliquant l'inégalité de Jensen à la fonction convexe H, la borne suivante est valide quand $d_c \to \infty$

$$B(\delta, \boldsymbol{\alpha}) = -\sum_i a_i H\left(\frac{\alpha_i}{a_i}\delta\right) + (1-r)\ln 2 \geq -H(\delta) + (1-r)\ln 2$$

et la borne est atteinte pour la distribution $\boldsymbol{\alpha}$ définie par $\alpha_i = a_i$, quel que soit i. Par conséquent, lorsque $n \to \infty$, la somme dans (6.3.11) est dominée par l'exposant de l'ensemble aléatoire des codes, à savoir $-H(\delta)+(1-r)\ln 2$ [3]. Cependant, dans la pratique, d_c reste inférieur à 20. Nous avons constaté expérimentalement que pour $d_c \leq 20$, de mauvaises distributions $\boldsymbol{\alpha}$ existent, tel que $B(\delta, \boldsymbol{\alpha}) = 0$ a une racine positive δ_0 au voisinage de zéro. Pour les codes LDPC irréguliers, la distance minimum ne s'approche pas de la borne de Gilbert-Varshamov, car il existe des distributions telles que $R(\delta, \boldsymbol{\alpha})$ tend lentement vers zéro quand d_c augmente.

6.4 Conclusions

Nous avons présenté une méthode pour évaluer le spectre des distances moyen de l'ensemble des codes LDPC pour la construction proposée par Richardson *et al.* [6], en nous basant sur les équations de parité définissant cet ensemble.

Il est intéressant de constater que l'ensemble proposé par Richardson *et al.* et l'ensemble de Gallager ont les mêmes propriétés en terme de distance.

Pour l'ensemble des codes LDPC irréguliers, un lien entre l'évolution de densité et les propriétés en terme de distance a été établi. En effet, la condition de stabilité de l'évolution de densité est une condition suffisante pour que la probabilité d'erreur due aux mots de code de poids faible tende vers zéro lorsque la

taille du code tend vers l'infini.

Troisième partie

Application des Systèmes de Décodage Itératif aux Modems Filaires et Non-filaires

Chapitre 7

Analyse de Performance des Systèmes VDSL avec Modulation DMT de Type BICM

Les systèmes VDSL ("very high bit-rate digital subscriber line") basés sur la modulation DMT ("discrete multitone") divisent le spectre des câbles téléphoniques en un grand nombre de sous-porteuses orthogonales utilisant une modulation d'amplitude en quadrature (MAQ) codée. Une analyse approchée qui estime le taux d'erreur binaire en fonction du rapport signal sur bruit (RSB) est en général employée pour déterminer la taille de la constellation MAQ utilisée par chaque sous-porteuse. Nous utiliserons la technique connue sous le nom de anglo-saxon de "bit interleaved coded modulation" (BICM), qui consiste simplement à insérer un entrelaceur entre l'encodeur et le modulateur MAQ, et qui se solde par une perte de capacité négligeable. Nous proposons la BICM afin de simplifier l'évaluation des performances des systèmes VDSL codés. Il en résulte un moyen plus rigoureux d'allouer les bits aux sous-porteuses. Nous illustrons cette technique pour les systèmes VDSL utilisant les codes Reed-Solomon et les codes BCH concaténés en série.

7.1 Introduction

La DMT est une technique de modulation utilisée pour combattre les interférences inter-symbole (ISI) en divisant le spectre du canal en sous-porteuses orthogonales recouvrantes [62, 63]. Les processus de modulation et de démodulation peuvent être implémentés efficacement, même pour un grand nombre de sous-porteuses en utilisant la transformée de Fourier rapide [64], c'est pourquoi la DMT est un choix convenable pour les systèmes à haut débit.

Dans un système VDSL basé sur la modulation DMT [65], l'effet de la fonction de transfert du canal sélectif en fréquence, des interférences et du bruit est pris en compte en assignant à chaque sous-porteuse le nombre maximal de bits étant donnés le taux d'erreur binaire ciblé après décodage P_b et le RSB disponible. Une solution optimale pour une 2^n-MAQ non-codée, connue sous le nom de chargement adaptatif [66, 67], est basée sur le calcul du taux d'erreur binaire en fonction du RSB, mais la méthode est exacte seulement pour n pair. Cette technique a été généralisée par la suite au cas de la MAQ codée en treillis en approchant le taux d'erreur binaire de la modulation codée par la performance de la modulation non-codée décalée par le gain de codage [68, 69]. Une méthode approchée est aussi disponible pour la MAQ codée en bloc [70].

La BICM, qui sépare l'encodage de la modulation en insérant un entrelaceur, a été introduite en premier par Zehavi [71] afin d'augmenter la diversité d'une modulation codée sur le canal à évanouissements de Rayleigh aussi haut que la distance de Hamming minimum du code. Dans [72], Caire *et al.* ont montré que l'entrelacement bit à bit n'induisait qu'une perte de capacité négligeable sur le canal AWGN. De plus, ces auteurs ont calculé une borne supérieure très proche de la probabilité d'erreur binaire de la BICM et ont montré que le meilleur choix de constellation était l'étiquetage de Gray.

Dans ce chapitre, nous envisageons la BICM et des constellations MAQ avec étiquetage de Gray pour les systèmes VDSL, comme une alternative au système MAQ codé adopté dans [65]. Nous montrons que l'analyse des performances de ce système est possible grâce à la séparation du codage et de la modulation, tandis que la capacité pour les codes de rendement élevé est approximativement la même. Les algorithmes de chargement existants, utilisant des approximations, peuvent mener à un décalage entre le taux d'erreur binaire effectif et ciblé après décodage. La contribution principale est un algorithme de chargement précis basé sur une étude de performance analytique.

7.2 Modèle du Système

Le modèle complexe discret en bande de base du système est illustré par la Fig. 7.1. L'émetteur consiste en un encodeur binaire de rendement R, un entrelaceur binaire et un mappeur qui produit un vecteur complexe de symboles MAQ $\mathbf{x} = (x_1, \ldots, x_N)$, où N représente le nombre de sous-porteuses. Un algorithme de chargement génère le vecteur $\mathbf{b} = (b_1, \ldots, b_N)$ contenant le nombre de bits que le mappeur alloue sur chaque porteuse ; en d'autres termes x_i est un point d'une constellation de type 2^{b_i}-MAQ. Nous considérons la modulation/démodulation DMT et le canal comme une seule entité. A condition que l'ISI soit éliminé par un

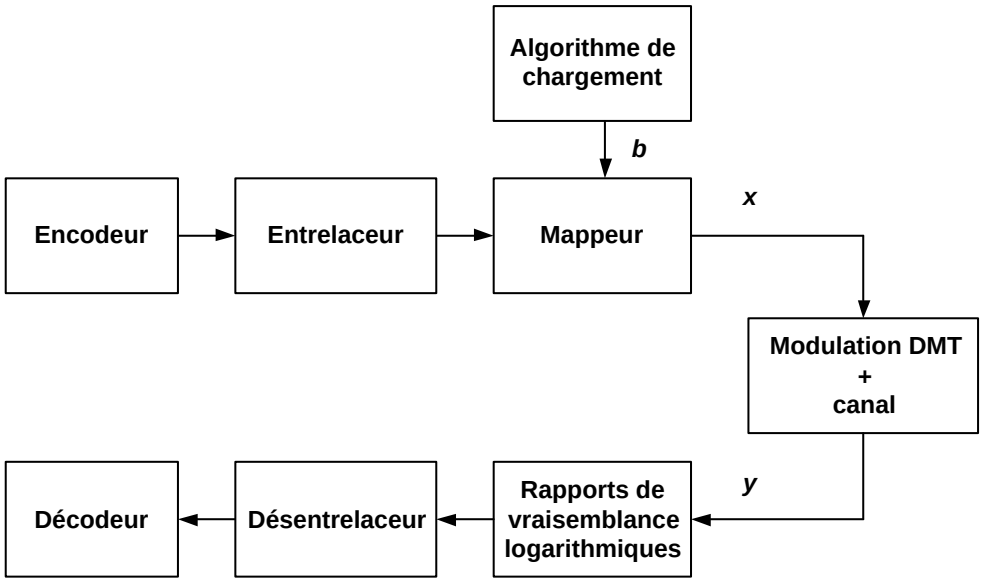

FIG. 7.1 – Schéma bloc d'un système de transmission VDSL utilisant la BICM.

préfixe cyclique aussi long que la réponse impulsionnelle du canal [69], la sortie du démodulateur DMT $\mathbf{y} = (y_1, \ldots, y_N)$ s'écrit

$$y_i = H_i x_i + n_i, \quad i = 1, \ldots, N \tag{7.2.1}$$

où H_i représente la réponse fréquentielle du canal et n_i la contribution de l'interférence et du bruit pour la $i$ème sous-porteuse. Au récepteur, le rapport de vraisemblance logarithmique des bits reçus est calculé [71], [72], puis désentrelacé et décodé.

Dans la suite de ce chapitre, nous supposerons que les coefficients de la réponse fréquentielle complexe du canal H_i sont fixés et que n_i peut être modélisé par un processus blanc et gaussien de variance N_i [73]. Notons aussi que puisque le rôle de l'entrelaceur bit à bit est de retirer la corrélation introduite par la modulation, une profondeur d'entrelacement égale à plusieurs fois la valeur maximale des b_i est suffisante [72]. De plus, nous supposerons que l'étiquetage de Gray est utilisé pour les constellations puisque ce choix mène aux meilleures performances [72] ; lorsque ceci n'est pas possible, un étiquetage quasi-Gray est utilisé à la place [74].

7.3 Analyse de Performance

7.3.1 Système Non-codé

Tout d'abord, nous rappellons brièvement l'analyse classique d'un système DMT non-codé [66]. La $i$ème sous-porteuse transmet des symboles pris dans une constellation de type 2^{b_i}-MAQ dont l'énergie est normalisée à un, où $d^2_{b_i}$

représente le carré de la distance euclidienne minimum et $b_i K_{b_i}$ est le nombre moyen d'erreurs binaires entraînées par des erreurs symbole à distance euclidienne minimum. Alors, d'après l'équation (7.2.1) la probabilité d'erreur binaire s'écrit

$$P_i = K_{b_i} Q\left(\sqrt{\frac{d_{b_i}^2 |H_i|^2}{2N_i}}\right), \tag{7.3.2}$$

où $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}} dt$. Cette expression est valide à RSB élevé lorsque les erreurs symbole à distance euclidienne minimum dominent la probabilité d'erreur. Pour les constellations paires $d_b^2 = \frac{6}{2^b - 1}$ [66] et avec un étiquetage de Gray $K_b \leq 1$ [72], alors P_i peut être majoré par

$$P_i \leq Q\left(\sqrt{\frac{3|H_i|^2/N_i}{2^{b_i} - 1}}\right). \tag{7.3.3}$$

Soit P_b la probabilité d'erreur binaire ciblé pour les N sous-porteuses, d'après l'équation (7.3.3) le choix optimal de b_i est donné par

$$b_i = \log_2\left(1 + \frac{|H_i|^2/N_i}{\Gamma}\right), \tag{7.3.4}$$

où $\Gamma = \left[Q^{-1}\left(P_b\right)\right]^2 / 3$ peut être interprété comme l'écart entre le RSB du canal $|H_i|^2/N_i$ et le RSB nécéssaire pour atteindre la capacité sur la $i$ème sous-porteuse. Notons que le résultat de l'algorithme de chargement donné par l'équation (7.3.4) doit être arrondi à l'entier le plus proche et est exact seulement si cet entier est pair.

Exemple 7.3.1. *Pour un taux d'erreur binaire ciblé de* 10^{-7} *(resp.* 10^{-9}*), l'écart* Γ *par rapport à la capacité vaut* 9.55 *dB (resp.* 10.79 *dB).*

La généralisation suivante de la procédure précédente conduit à un algorithme de chargement exact pour toute constellation MAQ avec étiquetage de Gray et pour la modulation BPSK ($b = 1$). Considérons que la plus grande constellation utilisée en VDSL est la 2^{15}-MAQ [65], l'algorithme de chargement choisit b_i de la manière suivante

$$b_i = \max\{1 \leq b \leq 15 : P_i \leq P_b\}, \tag{7.3.5}$$

où P_i est donné par l'équation (7.3.2) et les valeurs de d_b^2 et bK_b, $b = 2, \ldots, 15$ sont listées dans le Tab. 7.1. Cette procédure est valide à condition qu'une sous-porteuse dont le RSB est trop faible pour transporter un symbole BPSK, soit désactivée.

TAB. 7.1 – Carré de la distance euclidienne minimum et nombre moyen d'erreurs binaires dues aux erreurs symbole à distance euclidienne minimum pour des constellations MAQ normalisées avec étiquetage de Gray.

b	d_b^2	bK_b
1	4	1
2	2	2
3	4/6	2.5
4	4/10	3
5	4/20	3.75
6	4/42	3.5
7	4/82	3.875
8	4/170	3.75
9	4/330	3.9375
10	4/682	3.875
11	4/1322	3.96875
12	4/2730	3.9375
13	4/5290	3.984375
14	4/10922	3.96875
15	4/21162	3.992188

7.3.2 Systèmes Codés

A partir de l'exemple 7.3.1 nous constatons que l'écart de RSB par rapport à la capacité est grand, en conséquence, des gains de codage substantiels peuvent être atteints. Cependant, l'introduction d'un encodeur diminue le débit (informatif) net du rendement de codage R. Etant donnés les plans de fréquence disponibles pour les systèmes VDSL, le débit est dominé par les sous-porteuses à basses fréquences qui portent un grand nombre de bits, ceci entraîne que seuls les codes de rendement élevé conviennent. Le plus simple pour comprendre ce phénomène est de considérer l'exemple suivant.

Exemple 7.3.2. *Supposons que le plan de fréquence soit tel qu'un système DMT non-codé de durée symbole T transporte* 12 *bits sur toutes les N sous-porteuses. Si un code de rendement R, qui réduit l'écart par rapport à la capacité de* 6 *dB, est utilisé, chaque sous-porteuse sera capable de transporter environ* 2 *bits de plus. Il s'ensuit que le débit net vaut $12N/T$ et $14NR/T$ pour le système non-codé et codé, respectivement. Ainsi, le code amméliore le débit net uniquement si $R > 12/14 \approx 0.86$.*

Dans cette section, nous allons étudier les performances de codes Reed-Solomon et de codes BCH concaténés en série à haut rendement.

Codes Reed-Solomon

Les codes Reed-Solomon sont actuellement adoptés par le standard VDSL [65]. Les auteurs de [70] proposent une analyse approchée des systèmes DMT encodés par des codes Reed-Solomon, en supposant que la probabilité d'erreur des symboles Reed-Solomon est proportionnelle à la probabilité d'erreur des symboles MAQ. Malheureusement, comme il a été montré dans [75], cette hypothèse n'est pas vérifiée en général .

Nous allons montrer que si l'encodeur dans le système de la Fig. 7.1 est un code $RS(n,k,t)$ sur $GF(m)$ avec $n = 2^m - 1$, le calcul de la probabilité d'erreur binaire après décodage est grandement simplifié grâce aux propriétés de la BICM. Soient P_b, P_s, P_{RS} et P_{bit} le taux d'erreur binaire à la sortie du démodulateur, le taux d'erreur symbole RS avant décodage, le taux d'erreur symbole RS après décodage et le le taux d'erreur binaire après décodage, respectivement. En supposant que le cahier des charges du système impose la valeur de P_{bit}, le but est de trouver la valeur correspondante de P_b à utiliser comme taux d'erreur binaire ciblé dans l'algorithme de chargement donné par l'équation (7.3.5). Comme le désentrelaceur en réception détruit la corrélation introduite par la modulation, nous pouvons supposer ques les bits à la sortie du démodulateur sont i.i.d., donc

$$P_s = 1 - (1 - P_b)^m \tag{7.3.6}$$

L'expression de P_{RS} en fonction de P_s est donnée par [24]

$$P_{RS} = \sum_{i=t+1}^{n} \frac{i}{n} \binom{n}{i} P_s^i (1 - P_s)^{n-i} \tag{7.3.7}$$

Si nous supposons aussi que les bits en sortie du décodeur RS sont i.i.d., ce que nous avons vérifié par simulation, il suit que

$$P_{bit} = 1 - (1 - P_{RS})^{\frac{1}{m}} \tag{7.3.8}$$

En combinant les équations (7.3.6)-(7.3.8), nous pouvons tracer P_{bit} en fonction de P_b et obtenir le taux d'erreur binaire ciblé à utiliser par l'algorithme de chargement.

Exemple 7.3.3. *La Fig. 7.2 montre P_{bit} en fonction de P_b pour un code $RS(144, 128, 8)$ raccourci. En particulier, pour atteindre un taux d'erreur binaire après décodage de 10^{-7} (resp. 10^{-9}), un taux d'erreur binaire après démodulation de $1.2e-3$ (resp. $6.8e-4$) est requis.*

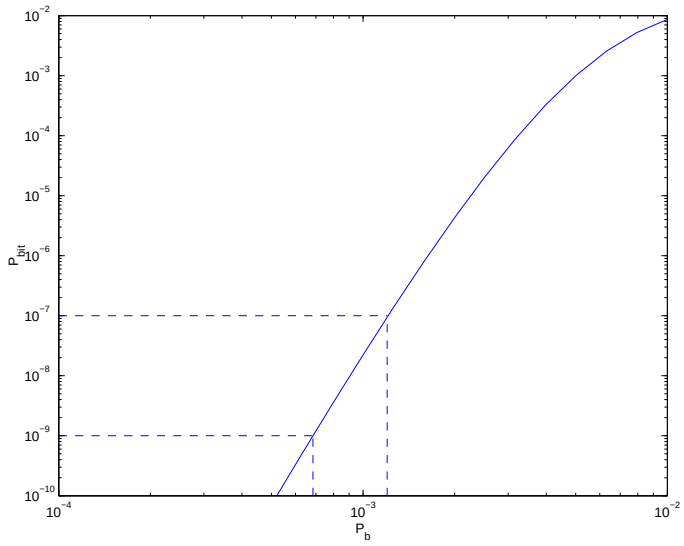

FIG. 7.2 – Taux d'erreur binaire après décodage en fonction du taux d'erreur binaire après démodulation pour un code $RS(144, 128, 8)$.

Codes BCH concaténés en série

Un code en bloc $C(n, k)$ concaténé en série, illustré par la Fig. 7.3, consiste en un code externe $C_o(n_o, k_o)$ suivi d'un entrelaceur et d'un code interne $C_i(n_i, k_i)$ [21]. Nous considérons le cas où la taille de l'entrelaceur est un multiple (par un fac-

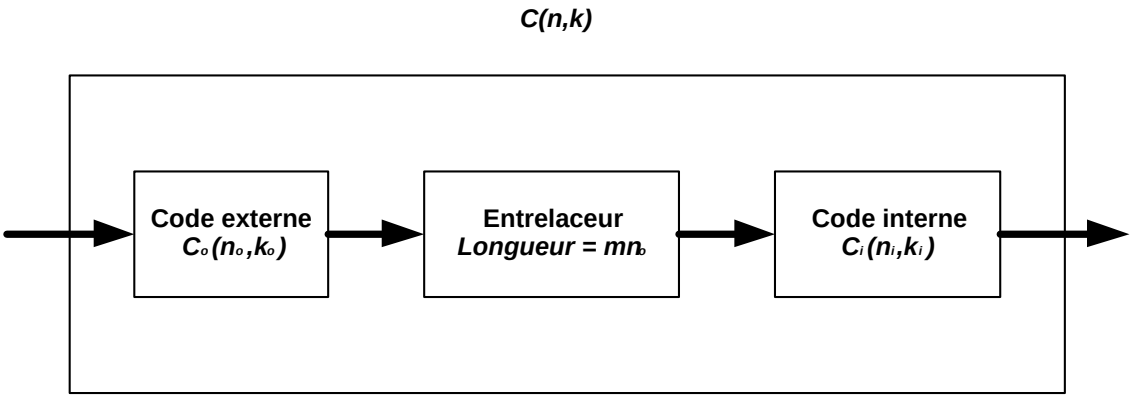

FIG. 7.3 – Schéma bloc d'une concaténation en série de codes en bloc.

teur m) de la longueur des mots de code du code externe. Nous supposerons aussi que la taille de l'entrelaceur mn_o est un multiple du nombre de bits informatifs du code interne k_i. Ces codes ont des taux d'erreur faibles à RSB faible et peuvent être décodés avec une procédure de décodage itératif quasi-optimale à faible complexité [18].

La procédure de chargement simplifiée adaptée au décodage souple proposée dans [68, 69] consiste essentiellement à remplacer la distance euclidienne minimum d_{b_i} du système non-codé par la distance euclidienne minimum du système codé dans l'équation (7.3.2). Malheureusement, cette approche n'est valide que

pour les RSB élevés où les erreurs à distance euclidienne minimum dominent la performance. Afin d'évaluer plus précisemment le taux d'erreur binaire d'un système VDSL utilisant des codes en bloc concaténés en série, nous modifions la technique de borne par réunion suggérée dans [21]. La borne par réunion de la probabilité d'erreur des mots de code est donnée par

$$P_e \leq \sum_{c \in C} p(c) \sum_{c' \neq c} P(c \rightarrow c'),$$

où $p(c)$ représente la probabilité d'émettre le mot de code c et $P(c \rightarrow c')$ est la probabilité que le décodeur détecte le mot de code c' à la place du mot de code correct c. Supposons que toutes les sous-porteuses transmettent des symboles d'énergie E_s pris dans une constellation de type 2^b-MAQ utilisant un étiquetage de Gray et que le bruit du canal à la même puissance N_0 sur toutes les sous-porteuses ; avec cette simplification la borne par réunion devient

$$P_e \leq \sum_{c \in C} p(c) \sum_{c' \neq c} Q\left(\sqrt{\frac{d_e(c,c')^2 E_s}{2N_0}}\right),$$

où $d_e(c,c')^2$ représente le carré de la distance euclidienne entre les séquences modulées en MAQ correspondant aux mots de code c and c'. Afin de s'affranchir de la dépendance de la borne en la séquence correcte, nous pouvons utiliser la borne inférieure $d_e(c,c')^2 \geq d_b^2 d_H(c,c')$ puisque l'étiquetage de Gray est utilisé ($d_H(.)$ représente la distance de Hamming). Alternativement, la méthode développée dans [72] est envisageable, malheureusement nous avons observé que pour des codes en bloc concaténés en série, cette borne ne converge que pour des valeurs extrêmement faibles de la probabilité d'erreur binaire. Puisque le code concaténé en série C est linéaire et que les mots de codes sont transmis de manière équiprobable, la borne est de la forme

$$P_e \leq \sum_{h=d_{min}}^{n} \sum_{w=1}^{h} A_{w,h} Q\left(\sqrt{\frac{h d_b^2 E_s}{2N_0}}\right),$$

où $A_{w,h}$ représente le nombre de mots de code de poids de Hamming h générés par des séquences d'information de poids w et d_{min} est la distance minimum du code. Soit $A(W,H) = \sum_{h=d_{min}}^{n} \sum_{w=1}^{h} A_{w,h} W^w H^h$ la fonction énumératrice des poids d'entrée/sortie (IOWEF) de C, alors

$$P_e \leq Q\left(\sqrt{\frac{d_{min} d_b^2 E_s}{2N_0}}\right) e^{\frac{d_{min} d_b^2 E_s}{4N_0}} A(W,H)\Big|_{W=1, H=e^{-\frac{d_b^2 E_s}{4N_0}}},$$

où nous avons utilisé l'inégalité $Q(\sqrt{x+y}) \leq Q(\sqrt{x})e^{-y/2}$ for $x \geq 0$, $y \geq 0$.

De même, le taux d'erreur binaire peut être majoré par

$$P_{bit} \leq \sum_{h=d_{min}}^{n} \sum_{w=1}^{h} \frac{w}{k} A_{w,h} Q\left(\sqrt{\frac{h d_b^2 E_s}{2N_0}}\right),$$

et en utilisant la même inégalité, on obtient

$$P_{bit} \leq \frac{1}{k} Q\left(\sqrt{\frac{d_{min} d_b^2 E_s}{2N_0}}\right) e^{\frac{d_{min} d_b^2 E_s}{4N_0}} \frac{\partial A(W,H)}{\partial W}\bigg|_{W=1, H=e^{-\frac{d_b^2 E_s}{4N_0}}}, \tag{7.3.9}$$

En se servant du changement de variable $p = e^{-\frac{d_b^2 E_s}{4N_0}}$, où p peut être interprété comme une borne supérieure de la probabilité d'erreur binaire après démodulation, l'équation (7.3.9) devient

$$P_{bit} \leq \frac{1}{k} Q\left(\sqrt{-2 d_{min} \ln p}\right) p^{-d_{min}} \frac{\partial A(W,H)}{\partial W}\bigg|_{W=1, H=p}, \tag{7.3.10}$$

Nous proposons l'algorithme de chargement suivant partant du principe que la probabilité d'erreur doit être maintenue constante sur toutes les sous-porteuses [68]. Nous traçons P_{bit} en fonction de p en utilisant l'équation (7.3.10) ; appellons p_0 la valeur de p correspondant à la valeur de probabilité d'erreur après décodage ciblée P_{bit}, alors pour la $i$ème sous-porteuse nous choisissons l'ordre de la modulation comme suit

$$b_i = \max\left\{1 \leq b \leq 15 : e^{-\frac{d_b^2 |H_i|^2}{4N_i}} \leq p_0\right\} \tag{7.3.11}$$

Exemple 7.3.4. *Choisissons comme codes interne et externe le code étendu* $BCH(64, 57, 1)$ *et* $m = 57$*, le code concaténé en série résultant* $C(64^2, 57^2)$ *utilise un entrelaceur de taille* 3648*. La Fig. 7.4 illustre* P_{bit} *en fonction de* p *pour ce code. En particulier, pour atteindre un taux d'erreur binaire après décodage de* 10^{-9}*, la valeur de* p *requise est* $p_0 = 0.114$*. Observons que la borne diverge pour* $P_{bit} \geq 2e-9$*. L'IOWEF du code* $BCH(64, 57, 1)$ *a été caclulée en construisant le treillis correspondant [38] et l'IOWEF de* C *a été obtenue grâce à la technique introduite dans [21].*

7.4 Résultats Numériques

Nous considérons un système VDSL synchrone ayant les caractéristiques suivantes [73, 76] :

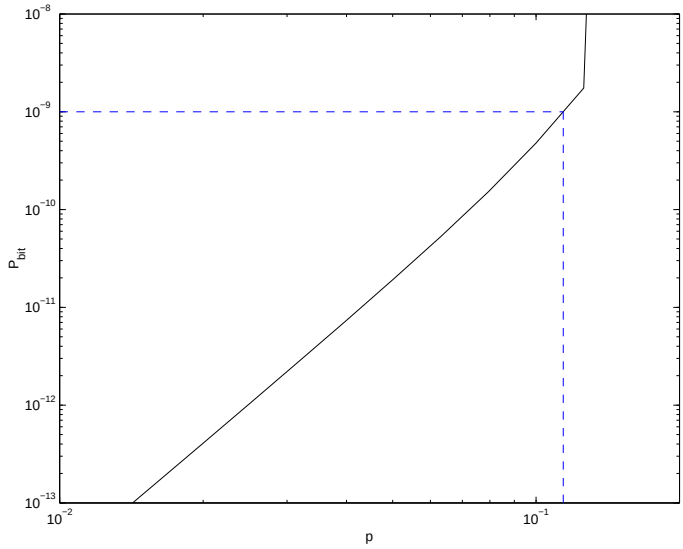

FIG. 7.4 – Taux d'erreur binaire après décodage en fonction du taux d'erreur binaire après démodulation pour la concaténation en série de deux codes $BCH(64, 57, 1)$ avec $m = 57$.

1. La bande de fréquence totale allant de 0 à 17.664 MHz est subdivisée en $N = 4096$ sous-porteuses.
2. Le signal VDSL est transmis avec une densité spectrale de puissance de -60 dBm/Hz.
3. Nous utilisons le mode FDD ("frequency division duplexing"), connu sous le nom de plan de fréquence 998, et dont les bandes en upstream (terminal vers la station de base) et en downstream (station de base vers le terminal) sont illustrées par la Fig. 7.5.
4. Nous utilisons les modèles de câbles téléphoniques ANSI VDSL1 et ETSI LOOP1.
5. Les caractéristiques des interférences correspondent à l'effet de 25 perturbateurs VDSL de type FEXT ("far-end crosstalk") et au modèle A pour les perturbateurs autres que VDSL.
6. La densité spectrale de puissance du bruit de fond est de -140 dBm/Hz.
7. Les sous-porteuses présentes dans les bandes réservées aux radio amateurs sont éteintes.
8. La réponse impulsionnelle du canal est suffisamment courte pour pouvoir négliger le rallongement de la durée symbole due au préfixe cyclique.

Les Fig. 7.6 et 7.7 montrent le débit (informatif) net en downstream et en upstream pour le modèle de câble téléphonique ANSI VDSL1 lorsque le taux d'erreur binaire après décodage ciblé est fixé à $P_{bit} = 10^{-9}$. Ces résultats incluent les performances du système VDSL considéré sans codage, avec un code

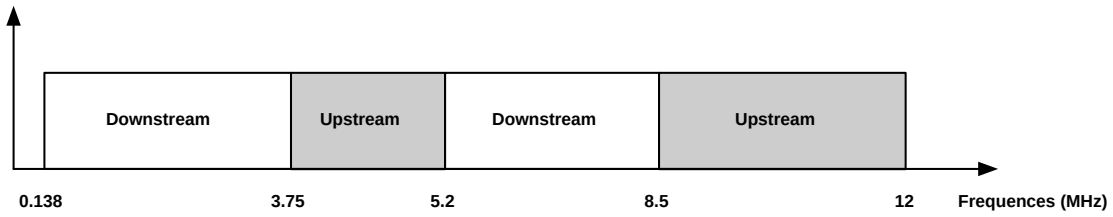

FIG. 7.5 – Plan de fréquence 998 pour les systèmes VDSL.

$RS(144, 128, 8)$ et avec le code BCH concaténé $C(64^2, 57^2)$ de l'exemple 7.3.4, en utilisant respectivement les algorithmes de chargement présentés dans la Sec. 7.3.1, 7.3.2, et 7.3.2. Nous représentons aussi la capacité du système donnée par $\sum_{i=1}^{N} \log_2 (1 + |H_i|^2/N_i)$. On observe que pour des câbles courts (typiquement de longueur inférieure à 1 km), l'augmentation du débit obtenue grâce au codage est à peu près constante. Ceci s'explique par le fait que pour les câbles courts, la majorité des sous-porteuses sont utilisées et transportent environ le nombre de bits donné par l'équation (7.3.4) à condition de diminuer l'écart à la capacité par le gain de codage. En particulier, l'écart résiduel à la capacité pour le code $RS(144, 128, 8)$ et le code concaténé en série C vaut 7.8 et 4.6 dB respectivement. Cependant, pour des longueurs de câble plus importantes, le RSB du canal décroît rapidement, tout particulièrement pour les hautes fréquences. En conséquence, de plus en plus de sous-porteuses doivent être éteintes, ce qui diminue d'autant le gain de débit obtenu par codage canal. Ce phénomène est particulièrement évident pour l'upstream qui utilise plus de sous-porteuses dans les hautes fréquences que le downstream (voir Fig. 7.5). Les Fig. 7.8 et 7.9 montrent des résultats similaires pour le modèle de câble téléphonique ETSI LOOP1. La différence principale réside dans le fait que le débit décroît moins sévèrement après 1 km. On observe que le code $RS(144, 128, 8)$ et le code BCH concaténé C augmentent repectivement le débit net de 5-6 et 7-8 Mbits/s pour les câbles courts.

7.5 Conclusions

Nous avons réalisé une étude théorique d'un système VDSL basé sur le modulation DMT utilisant la BICM et des constellations MAQ à étiquetage de Gray. Cette étude nous a permis de trouver de nouveaux algorithmes de chargement pour des systèmes VDSL sans codage, avec un code Reed-Solomon et des codes BCH concaténés en série. La différence principale avec les procédures de chargement existantes est que le taux d'erreur binaire après décodage ciblé est toujours atteint au lieu d'être approximativement atteint. En fait, la méthode décrite dans la Sec. 7.3.2 peut être appliquée à tout code en bloc utilisant un décodeur

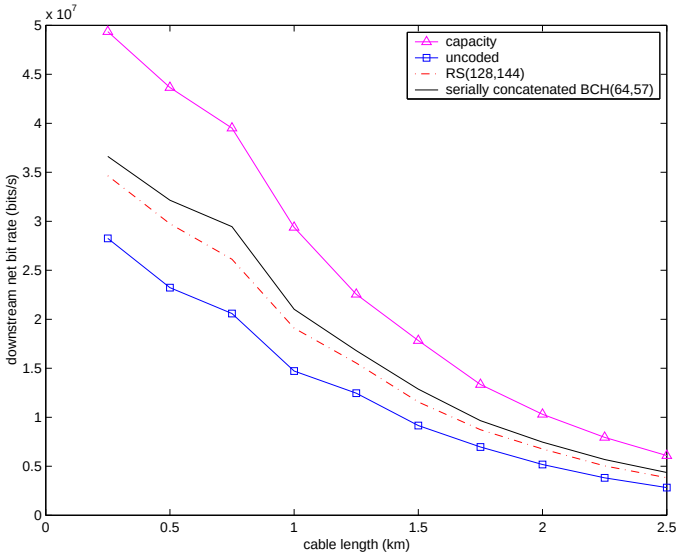

FIG. 7.6 – Débit net en downstream (bits/s) pour le modèle de câble ANSI VDSL1 et pour $P_{bit} = 10^{-9}$.

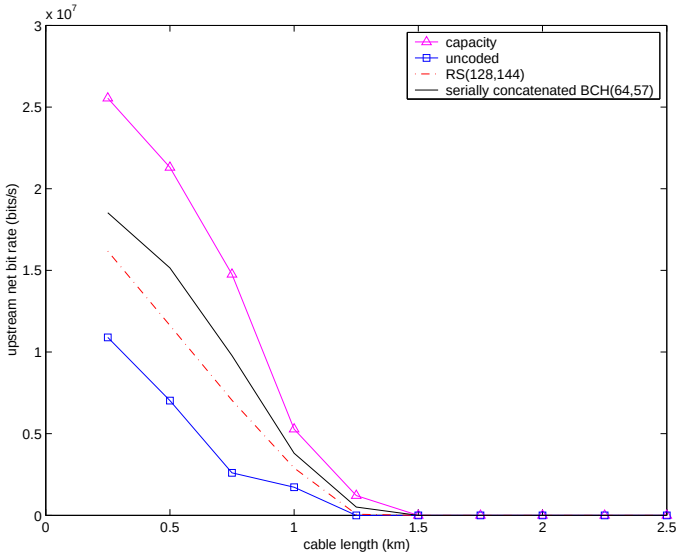

FIG. 7.7 – Débit net en upstream (bits/s) pour le modèle de câble ANSI VDSL1 et pour $P_{bit} = 10^{-9}$.

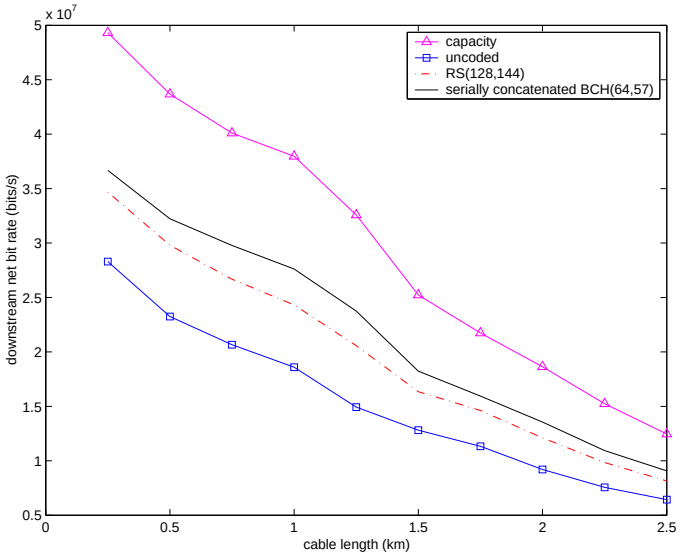

FIG. 7.8 – Débit net en downstream (bits/s) pour le modèle de câble ETSI LOOP1 et pour $P_{bit} = 10^{-9}$.

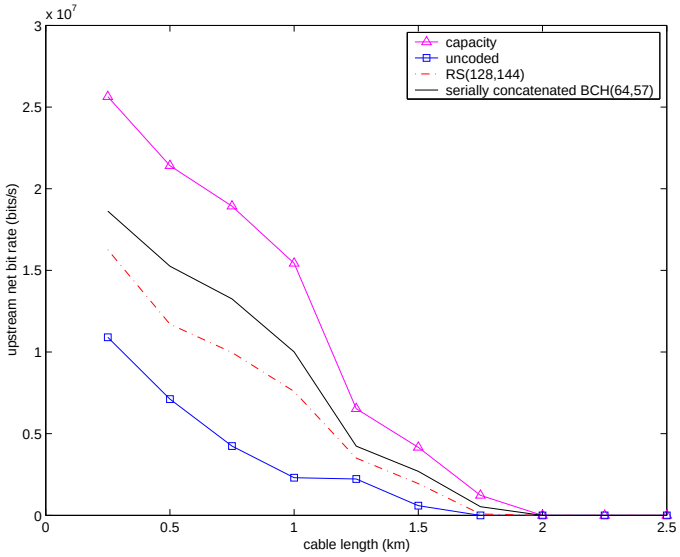

FIG. 7.9 – Débit net en upstream (bits/s) pour le modèle de câble ETSI LOOP1 et pour $P_{bit} = 10^{-9}$.

algébrique et la méthode de la Sec. 7.3.2 est valide pour tout code utilisant un décodeur souple à vraisemblance maximale. Enfin, des résultats numériques, pour des modèles de câbles téléphoniques courants fournis par les standards ANSI et ETSI, montrent que le codage canal peut augmenter de manière significative le débit d'un système VDSL.

Chapitre 8

Codes Produits Optimisés pour l'ATM Sans Fil

8.1 Introduction

Les réseaux locaux sans fil permettent des communications fiables sur le canal radio. Le modèle de réseaux basé sur l'ATM ("asynchronous transfer mode") consiste en un anneau fédérateur ou réseau "backbone" ATM, étendu dans le canal radio. Afin de rendre cette extension aussi transparente que possible, nous supposons que les packets ATM sans fil sont formés d'une cellule ATM augmentée d'un en-tête pour les besoins de la couche MAC ("medium access control"). Dans le scénario que nous considérons, le packet ATM sans fil contient 54 octets. De plus, afin d'éviter une couche de convergence complexe, la longueur du code correcteur de la couche physique est choisie égale à la taille du packet ATM sans fil, i.e. 432.

Le problème principal du système proposé est lié au fait que le protocole ARQ ("automatic repeat request") de l'anneau fédérateur ATM est dimensionné pour des taux d'erreur de transmission faibles. Cependant, les erreurs de transmission sont fréquentes dans le canal radio à cause des évanouissements. En conséquence, un codage canal efficace est nécéssaire dans la couche physique de la station de base et du terminal mobile, afin d'éviter une forte baisse du débit net due à de fréquentes retransmissions. Nous montrons que sous certaines conditions sur la technique de modulation employée dans la couche physique, de bonnes performances sont obtenues sur le canal à évanouissements de Rayleigh lorsque la distance minimum du code est grande [72]. Comme les codes BCH produits sont des codes en bloc bien connus pour leur grande distance minimum, nous optimisons des codes BCH produits de longueur 432 pour les rendements usuels 1/2, 9/16, 2/3 and 3/4. Ensuite, nous comparons les performances de ces codes avec

les performances du code convolutif classique à 64 états sur le canal de Rayleigh sans mémoire. Notons qu'une application possible est le standard Hiperlan2 [82] lorsque l'entrelacement temps-fréquence est parfait.

8.2 Critère de Performance des Codes en Bloc sur le Canal à Evanouissements de Rayleigh

Nous analysons le taux d'erreur binaire des codes en bloc sur les canaux de Rayleigh sans mémoire et complètement corrélé. Sur le canal de Rayleigh sans mémoire, nous utiliserons la BICM ("bit interleaved coded modulation") afin d'augmenter la diversité aussi haut que la distance de Hamming minimum du code. Sauf mention expresse du contraire, nous supposerons que la technique de modulation est basée sur des constellations à étiquetage de Gray. De plus, le code correcteur d'erreurs est un code en bloc binaire $C(n,k)$, où n désigne la longueur du code, $k = 432$ est le nombre de bits informatifs, R est le rendement de codage et d_{min} la distance minimum.

8.2.1 Canal de Rayleigh Sans Mémoire

Transmettons le mot de code $\mathbf{c} = (c_1, \ldots, c_n)$, où les c_i, pour $i = 1, \ldots, n$ sont des digits binaires. En supposant que la modulation BPSK est utilisée, la séquence reçue $\mathbf{y} = (y_1, \ldots, y_n)$ est définie par

$$y_i = \sqrt{E_s} a_i (1 - 2c_i) + n_i, \quad \text{for } i = 1, \ldots, n,$$

où E_s désigne l'énergie moyenne par symbole, $\mathbf{n} = (n_1, \ldots, n_n)$ est une séquence de bruit gaussien i.i.d. de densité spectrale de puissance N_0 et $\mathbf{a} = (a_1, \ldots, a_n)$ est une séquence d'amplitudes de Rayleigh i.i.d. de densité de probabilité

$$p(a) = 2ae^{-a^2}.$$

Soit $\mathbf{c}' = (c'_1, \ldots, c'_n)$ un mot de code qui diffère de $\mathbf{c}$ en d positions. La probabilité qu'un décodeur à vraisemblance maximale confonde $\mathbf{c}$ avec $\mathbf{c}'$, sachant que $\mathbf{c}$ est transmis et sachant que $\mathbf{a}$ est le vecteur d'amplitudes de Rayleigh indépendantes, s'écrit

$$P(\mathbf{c} \to \mathbf{c}'|\mathbf{a}) = Q\left(\sqrt{\frac{2dRE_b}{N_0} \sum_{i=1}^{d} (a_i^*)^2}\right),$$

où $\mathbf{a}^* = (a_1^*, \ldots, a_d^*)$ est le vecteur d'amplitudes de Rayleigh correspondant aux d positions en lesquelles $\mathbf{c}$ et $\mathbf{c}'$ diffèrent. La probabilité qu'un décodeur à vrai-

semblance maximale confonde $\mathbf{c}$ avec $\mathbf{c}'$ est obtenue en calculant la probabilité marginale

$$P(\mathbf{c} \to \mathbf{c}') = \int_{a_1^*} \dots \int_{a_d^*} Q\left(\sqrt{\frac{2dRE_b}{N_0} \sum_{i=1}^{d} (a_i^*)^2}\right) \prod_{i=1}^{d} 2a_i^* e^{-(a_i^*)^2} da_i^*,$$

Observons que cette expression est indépendante du mot de code transmis. C'est pourquoi, nous pouvons supposer sans perte de généralité que le mot de code tout-zéro est transmis. En écrivant l'expression de la fonction queue de distribution gaussienne de la manière suivante

$$Q(x) = \frac{1}{\pi} \int_0^{\pi/2} \exp\left(-\frac{x^2}{2\sin^2\phi}\right) d\phi,$$

la probabilité de détecter un mot de code de poids d à la place du mot de code tout-zéro au récepteur devient

$$P(d) = \frac{1}{\pi} \int_0^{\pi/2} \left(\frac{\sin^2\phi}{\sin^2\phi + \frac{RE_b}{N_0}}\right)^d d\phi$$

Soit $A_{w,d}$ le nombre de mots de code de poids d et de poids informatif w, la borne par réunion de la probabilité d'erreur binaire s'écrit

$$P_b \leq \sum_w \frac{w}{k} \sum_d A_{w,d} P(d). \tag{8.2.1}$$

Pour des modulations d'ordre plus élevé, nous pouvons utiliser l'expression générale du taux d'erreur binaire trouvée en [72] pour un RSB élevé

$$\log_{10} P_b = -d_{min} \left[\left(Rbd_h^2\right)_{dB} + \left(\frac{E_b}{N_0}\right)_{dB}\right]/10 + const,$$

où b est le nombre de bits dans la constellation et d_h^2 représente la moyenne harmonique du carré des distances euclidiennes entre sous-ensembles complémentaires de la constellation. Les valeurs de d_h^2 sont listées dans le Tab. 8.1 pour les constellations MAQ à étiquetage de Gray. Donc, sur une échelle logarithmique, le taux d'erreur binaire est proche d'une ligne droite de pente proportionnelle à $-d_{min}$, translatée horizontalement par le décalage $(Rbd_h^2)_{dB}$. Il s'ensuit que pour un même code correcteur, la perte d'efficacité de puissance d'une modulation d'ordre supérieur par rapport à une modulation BPSK vaut

$$L_{dB} = 10\log_{10}\left(\frac{4}{bd_h^2}\right) dB.$$

TAB. 8.1 – Canal de Rayleigh sans mémoire : Valeurs de d_h^2 (moyenne harmonique du carré des distances euclidiennes entre sous-ensembles complémentaires de la constellation) et L_{dB} (perte d'efficacité de puissance par rapport à une modulation BPSK) pour des constellations MAQ normalisées à étiquetage de Gray.

b	d_h^2	L_{dB}
2	2	0
4	0.4923	3.1
6	0.1442	6.6
8	0.043	10.7

Le Tab. 8.1 donne les valeurs de L_{dB} pour des constellations MAQ à étiquetage de Gray sur le canal de Rayleigh sans mémoire.

8.2.2 Canal de Rayleigh Complètement Corrélé

Nous supposerons que le temps de cohérence des amplitudes de Rayleigh est supérieur au temps nécéssaire à la transmission d'un mot de code sur le canal. Par conséquent, tous le symboles dans la séquence modulée correspondant à un mot de code sont affectés par la même amplitude de Rayleigh a. Alors la probabilité de détecter $\mathbf{c}'$ à la place de $\mathbf{c}$, sachant la valeur de a, s'écrit

$$P(\mathbf{c} \to \mathbf{c}'|a) = Q\left(\sqrt{\frac{d_e(\mathbf{c},\mathbf{c}')^2 E_s}{2N_0} a^2}\right),$$

où $d_e(\mathbf{c},\mathbf{c}')^2$ représente le carré de la distance euclidienne entre les séquences modulées correspondant à $\mathbf{c}$ et $\mathbf{c}'$. Pour s'affranchir de la dépendance par rapport à la séquence correcte $\mathbf{c}$, nous utilisons la borne inférieure $d_e(\mathbf{c},\mathbf{c}')^2 \geq d_b^2 d_H(\mathbf{c},\mathbf{c}')$ puisque l'étiquetage de Gray est utilisé. Par conséquent, la probabilité de détecter un mot de code qui diffère du mot de code correct en d positions est majoré par

$$\begin{aligned} P(d) &\leq \int_0^\infty 2ae^{-a^2} Q\left(\sqrt{\frac{dd_b^2 RbE_b}{2N_0} a^2}\right) da \\ &\leq \frac{1}{2}\left[1 - \frac{\sqrt{\frac{dd_b^2 RbE_b}{4N_0}}}{\sqrt{1+\frac{dd_b^2 RbE_b}{4N_0}}}\right] \end{aligned} \tag{8.2.2}$$

et la borne est atteinte pour une modulation BPSK i.e. $b = 1$, $d_b^2 = 4$. A RSB élevé, la borne (8.2.2) peut être approximée par

$$P(d) \leq \frac{1}{d d_b^2 R b \frac{E_b}{N_0}}.$$

Nous concluons cette section en notant que pour les canaux de Rayleigh sans mémoire et complètement corrélé le taux d'erreur binaire est faible lorsque la distance minimum du code est élevée, à condition que des constellations à étiquetage de Gray soient utilisées et que la BICM soit employée dans le cas du canal de Rayleigh sans mémoire. Celà signifie qu'en cherchant de bons codes, le critère d'optimisation principal doit être une distance minimum élevée. Un critère d'optimisation secondaire consiste à choisir des codes avec le moins possible de mots de code de poids minimum.

8.3 Optimisation de Codes Produits

8.3.1 Algorithme d'Optimisation

La raison pour laquelle nous avons retenu les codes produits comme de bons candidats pour réaliser des transmissions fiables sur le canal de Rayleigh est que ces codes ont de bonnes propriétés en terme de distance ainsi qu'une complexité de décodage faible grâce au décodage itératif [18]. Considérons un code produit $C(n, k, d_{min})$ formé d'un code en ligne $C_R(n_R, k_R, d_{min}^R)$ de rendement R_R et d'un code en colonne $C_C(n_C, k_C, d_{min}^C)$ de rendement R_C. On a alors la relation $d_{min} = d_{min}^R \times d_{min}^C$ [13]. Cette propriété va simplifier considérablement la recherche de bons codes. Nous nous restreignons à des codes constitutifs BCH et BCH étendus binaires, qui sont bien connus pour leurs bonnes propriétés en terme de distance. Une liste de ces codes peut être trouvée dans [79]. Afin d'atteindre les rendements désirés 1/2, 9/16, 2/3 et 3/4, on choisit de raccourcir le code en ligne (resp. en colonne) de s_R (resp. s_C) bits informatifs. Notons que la distance minimum d'un code BCH peut diminuer si les valeurs de s_R ou s_C deviennent trop grandes. Par la suite, nous nous assurerons toujours que ce cas limite ne se produit pas. Dans le but de conserver une complexité de décodage itératif faible, nous restreindrons l'optimisation aux codes BCH de longueur inférieure ou égale à 256 et de pouvoir de correction d'erreurs inférieur ou égal à 2 bits. Cette dernière hypothèse correspond au fait que le décodeur algébrique, nécéssaire à l'algorithme SISO de Chase à sorties pondérées, peut être implémenté aisément grâce à une table lorsque le pouvoir de correction d'erreurs est faible. Notons aussi que l'encodage et le décodage algébrique des codes raccourcis peuvent être

TAB. 8.2 – Couples valides (k_R, k_C) tell que $k_R \times k_C = 432$ avec $k_R < k_C$.

k_R	k_C
2	216
3	144
4	408
6	72
8	54
9	48
12	36
16	27
18	24

simplifiés en appliquant la technique proposée dans [79] [pp. 116-121].

Comme le nombre de bits informatifs dans le code produit est fixé à 432, les couples (k_R, k_C) valides peuvent prendre uniquement les valeurs listées dans le Tab. 8.2. Nous décrivons maintenant la procédure d'optimisation en notant R_d le rendement de codage désiré :

1. Etape 1. Pour chaque couple de codes BCH $C_1(n_1, k_1, d_1)$, $C_2(n_2, k_2, d_2)$ avec $k_1 < k_2$ choisis dans la liste précédemment mentionnée et pour chaque couple (k_R, k_C) choisi dans le Tab. 8.2, calculons le nombre de bits raccourcis de la manière suivante

$$s_R = k_1 - k_R$$
$$s_C = k_2 - k_C.$$

   Si $s_R \geq 0$ et $s_C \geq 0$, le code en ligne est choisi comme $C_1(n_1, k_1, d_1)$ avec les s_R premiers bits informatifs raccourcis et le code en colonne est choisi comme $C_2(n_2, k_2, d_2)$ avec les s_C premiers bits informatifs raccourcis. Le rendement R et la distance minimum d_{min} du code produit résultant sont alors

$$R = \frac{(k_1 - s_R)(k_2 - s_C)}{(n_1 - s_R)(n_2 - s_C)}$$
$$d_{min} = d_1 \times d_2.$$

   Stockons les paramètres de ce code si $|R - R_d| < \epsilon$, où ϵ est fixé arbitrairement à 0.03.

2. Etape 2. Pour tout code BCH $C_1(n_1, k_1, d_1)$ choisi dans la liste précédemment mentionnée et pour tout diviseur j de 432, le code en ligne est choisi comme $C_1(n_1, k_1, d_1)$ avec les $s_R = k_1 - 432/j$ premiers bits informatifs raccourcis

TAB. 8.3 – Codes produits optimisés pour les rendements de codage désirés 1/2, 9/16, 2/3 and 3/4.

n_R	k_R	s_R	d^R_{min}	n_C	k_C	s_C	d^C_{min}	n	k	R	d_{min}
32	26	8	4	63	51	27	5	864	432	1/2	20
32	26	8	4	128	120	96	4	768	432	9/16	16
64	51	15	6	13	12	0	2	637	432	0.678	12
128	120	84	4	13	12	0	2	572	432	0.755	8

et le code en colonne est choisi comme le code de parité de longueur $j+1$, à condition que $s_R \geq 0$. Le rendement R et la distance minimum d_{min} du code produit résultant sont alors

$$R = \frac{(k_1 - s_R)j}{(n_1 - s_R)(j+1)}$$
$$d_{min} = d_1 \times 2.$$

Stockons aussi les paramètres de ce code si $|R - R_d| < \epsilon$, où ϵ est fixé arbitrairement à 0.03.

3. Etape 3. Parmi les codes produits retenus durant les étapes 1 et 2, choisissons celui à distance minimum la plus grande. S'il advient que plusieurs codes aient la même distance minimum, retenons celui qui a le moins possible de mots de code de poids minimum.

Remarque 8.3.1. *L'étape 2 est introduite pour trouver des codes lorsque le rendement R_p est grand. Le coefficient ϵ est choisi de telle sorte que des codes produit de distance minimum élevée et de rendement raisonnablement proche de R_d ne soient pas éliminés. Enfin, une méthode pour trouver le nombre de mots de code de poids minimum sera décrite dans la Sec. 8.4.1.*

8.3.2 Résultats de l'Optimisation

Les codes produits optimisés avec l'algorithme décrit dans la Sec. 8.3.1 sont listés dans le Tab. 8.3 pour les rendements de codage désirés 1/2, 9/16, 2/3 et 3/4.

A titre de référence, le masque de poinçonnage et la distance minimum pour le code convolutif à 64 états illustré par la Fig. 8.1 sont donnés dans le Tab. 8.4. Observons que la distance minimum des codes produits optimisés est relativement supérieure à la distance minimum du code convolutif poinçonné de même rendement. C'est pourquoi on peut s'attendre à ce que la complexité de décodage plus

TAB. 8.4 – Paramètres des codes convolutifs poinçonnés à 64 états.

Rendement	Masque de poinçonnage	Distance minimum
1/2	X : 1 Y : 1	10
9/16	X : 111111110 Y : 111101111	7
2/3	X : 11 Y : 10	6
3/4	X : 110 Y : 101	5

élevée des codes produits optimisés soit compensée par des gains de performance substantiels.

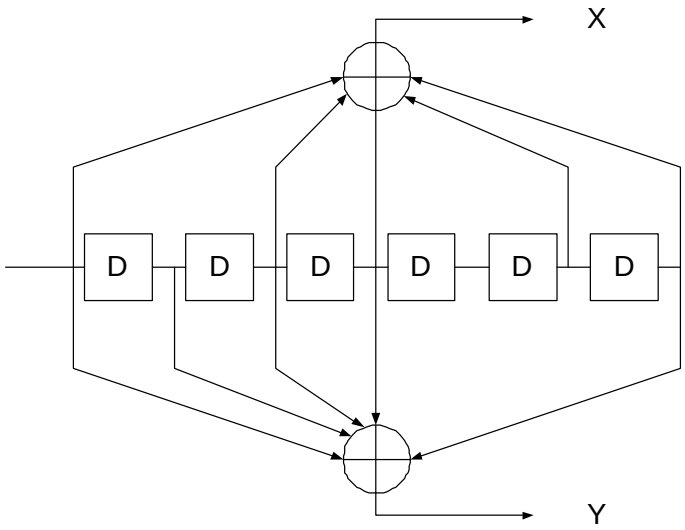

FIG. 8.1 – Code convolutif mère à 64 états de rendement 1/2.

8.4 Evaluation des Performances des Codes Produits Optimisés

8.4.1 Enumérateur de Poids des Codes Produits

Soient A_d^R, A_d^C et A_d le nombre de mots de code de poids d dans le code en ligne, le code en colonne et le code produit, respectivement. En supposant que les codes soient binaires, il a été montré dans [80] que pour $1 \leq d \leq d_{min}^R d_{min}^C +$

$\max\left(\left\lceil \frac{d^R_{min}}{2} \right\rceil, \left\lceil \frac{d^C_{min}}{2} \right\rceil\right)$

$$A_d = \sum_{i \mid d} A^R_i A^C_{d/i},$$

où la somme s'étend sur tous les diviseurs i de d. Ceci résulte d'une extension du Thm. 3 dans [81], qui démontre qui tout mot de code $\mathbf{c}$ de poids d dans le code produit vérifie

$$\mathbf{c}_{i,j} = (\mathbf{c}_R)_i (\mathbf{c}_C)_j, \quad \forall (i,j)$$

où $\mathbf{c}_R$ est un mot de code du code en ligne et $\mathbf{c}_C$ est un mot de code du code en colonne, sous réserve que $d^R_{min} d^C_{min} \leq d \leq d^R_{min} d^C_{min} + \min\left(d^R_{min}, d^C_{min}\right)$.

De même, soient $A^R_{w,d}$, $A^C_{w,d}$ et $A_{w,d}$, le nombre de mots de code de poids d correspondant à des séquences informatives de poids w, dans le code en ligne, le code en colonne et le code produit, respectivement. Définissons les fonctions énumératrices de poids conditionnelles ou "conditional weight enumerating functions" (CWEF) de la manière suivante

$$A^R_d(W) = \sum_w A^R_{w,d} W^w$$
$$A^C_d(W) = \sum_w A^C_{w,d} W^w$$
$$A_d(W) = \sum_w A_{w,d} W^w.$$

Alors, pour $1 \leq d \leq d^R_{min} d^C_{min} + \max\left(\left\lceil \frac{d^R_{min}}{2} \right\rceil, \left\lceil \frac{d^C_{min}}{2} \right\rceil\right)$ nous avons

$$A_d(W) = \sum_{i \mid d} A^R_i(W) A^C_{d/i}(W),$$

où la somme s'étend sur tous les diviseurs i de d. En injectant ce résultat dans (8.2.1) on obtient la borne par réunion tronquée de la probabilité d'erreur binaire pour une modulation BPSK sur le canal de Rayleigh sans mémoire.

Remarque 8.4.1. *Les CWEF $A^R_d(W)$ et $A^C_d(W)$ peuvent être obtenues en construisant le treillis [38] des codes en ligne et en colonne.*

8.4.2 Résultats Numériques

Nous avons simulé les codes produits optimisés en utilisant le décodage itératif pour une modulation BPSK sur le canal de Rayleigh sans mémoire. Les codes BCH constitutifs sont décodés à l'aide de l'algorithme de Chase à sortie pondérée décrit dans la Sec. 2.4.3. L'algorithme SISO correspondant à un code de parité est donné par la "règle de la tanh" [6]. Les Fig. 8.2 et 8.3 présentent les courbes

de taux d'erreur binaire et trame simulées (courbes en pointillés) et théoriques (courbes en trait plein). Six itérations de décodage sont utilisées pour le code de rendement 1/2, tandis que quatre itérations suffisent pour les autres rendements. Par construction, le taux d'erreur trame correspond au taux d'erreur des cellules ATM. Notons qu'à cause de la nature sous-optimale du décodage itératif, les performances simulées ne rejoignent pas exactement les bornes théoriques à RSB élevé.

Il est aussi intéressant de comparer les performances des codes produits optimisés aux performances obtenues avec le code convolutif standard à 64 états. Les Fig. 8.4 et 8.5 présentent les bornes théoriques de la probabilité d'erreur binaire et trame pour les codes produits (courbes en trait plein) et les codes convolutifs (courbes en pointillés). Comme prévu, étant donné que les codes produits optimisés ont une distance minimum plus élevée que les codes convolutifs poinçonnés, de larges gains de diversité sont atteints.

8.5 Conclusions

Nous avons présenté une méthode pour trouver de bons codes produits pour les applications ATM sans fil à partir de codes BCH raccourcis. Les codes produits optimisés résultants ont une distance minimum qui est en général relativement supérieure à la distance minimum des codes convolutifs poinçonnés à 64 états correspondants. En se servant de bornes et de simulations, nous avons montré que des gains de performance significatifs peuvent être obtenus sur le canal de Rayleigh sans mémoire en remplaçant le code convolutif standard à 64 états par un code produit optimisé. Ces codes sont performants pour la modulation BPSK, mais aussi pour des modulations d'ordre supérieur, si la technique de modulation est basée des constellations à étiquetage de Gray et sur la BICM pour le canal de Rayleigh sans mémoire.

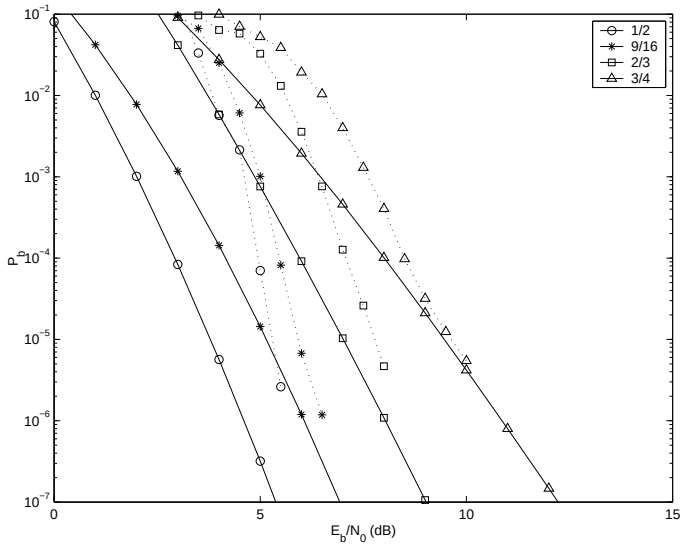

FIG. 8.2 – Taux d'erreur binaire théorique (trait plein) et simulé (pointillés) des codes produits optimisés sur le canal de Rayleigh sans mémoire. Les rendements sont indiqués dans la légende.

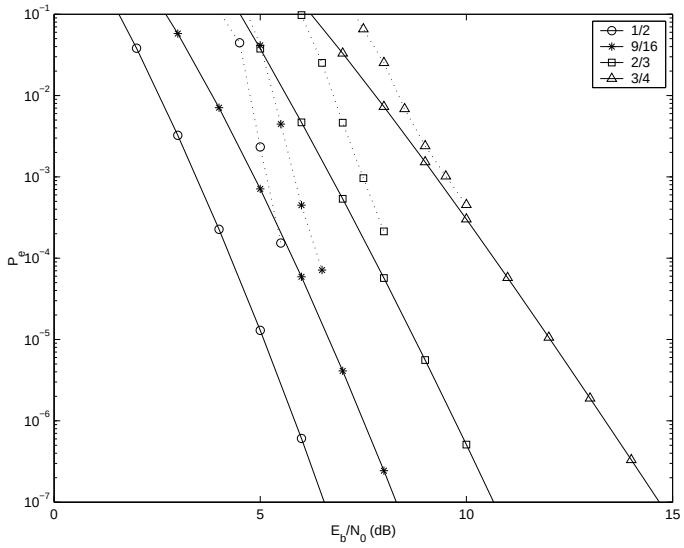

FIG. 8.3 – Taux d'erreur trame théorique (trait plein) et simulé (pointillés) des codes produits optimisés sur le canal de Rayleigh sans mémoire. Les rendements sont indiqués dans la légende.

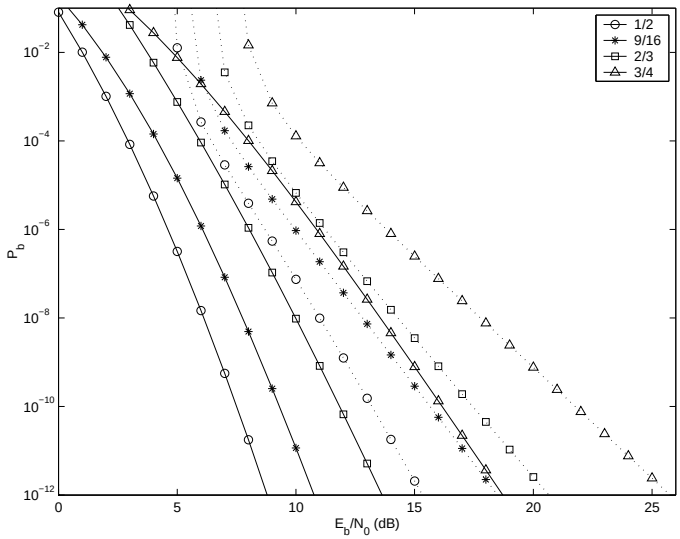

FIG. 8.4 – Performance des code produits optimisés (trait plein) et des codes convolutifs poinçonnés à 64 états (pointillés). Taux d'erreur binaire théorique sur le canal de Rayleigh sans mémoire. Les rendements sont indiqués dans la légende.

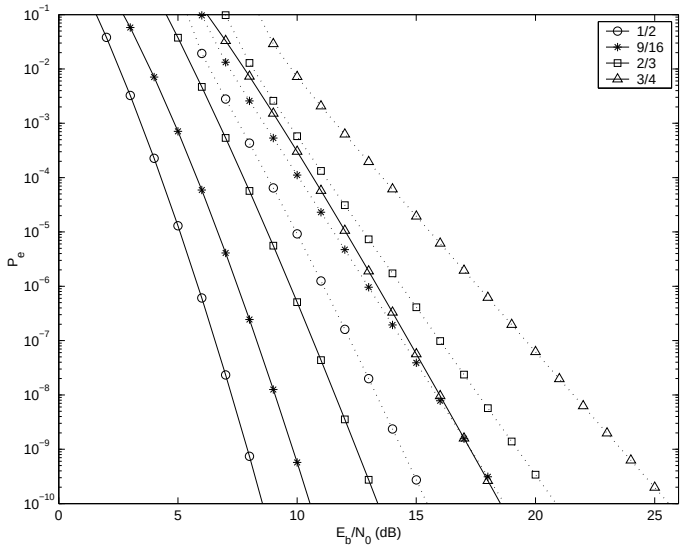

FIG. 8.5 – Performance des code produits optimisés (trait plein) et des codes convolutifs poinçonnés à 64 états (pointillés). Taux d'erreur trame théorique sur le canal de Rayleigh sans mémoire. Les rendements sont indiqués dans la légende.

Conclusion générale

L'objectif principal de cette thèse était de faire progresser la compréhension du fonctionnement des algorithmes de décodage itératif en trouvant des liens avec la structure des codes sous-jacents. En plus de ce but théorique, cette thèse apporte aussi une contribution pratique en montrant le potentiel du décodage itératif pour les applications filaires et non-filaires.

Contributions principales

La **première partie** de la thèse récapitule l'état de l'art des systèmes de décodage itératif. Elle a permis de préciser la construction des codes utilisés, leurs propriétés en terme de distance ainsi que les algorithmes de décodage itératif associés.

Dans la **deuxième partie** nous rappelons que les systèmes de décodage itératif sont des systèmes dynamiques non-linéaires discrets à dimension élevée avec un grand nombre de paramètres. C'est pourquoi les algorithmes de décodage itératif présentent des trajectoires complexes avec des points fixes, des orbites périodiques, des bifurcations et du chaos. Nous avons vérifié ce fait par simulation pour de nombreux systèmes de décodage itératif.

En approchant les densités de probabilité des messages échangés dans le décodeur par des gaussiennes, la dynamique du système peut être decrite par un modèle simple à une dimension. Ce modèle permet de comprendre l'effet de seuil du bruit et la vitesse de convergence pour le décodage itératif des codes produits et LDPC. En particulier, nous avons montré comment utiliser la fonction énumératrice de poids des codes constitutifs des codes concaténés pour obtenir un modèle simplifié à 1-D de l'évolution de densité.

Puisque le décodage itératif est sous-optimal, il est important d'étudier les performances du décodeur à vraisemblance maximale en évaluant les propriétés relatives aux distances des structures de code sous-jacentes. Ceci a été fait dans cette thèse pour les codes LDPC irréguliers. Il a ainsi été possible d'établir un lien entre la fonction énumératrice de poids des codes LDPC et l'évolution de densité en interprétant la condition de stabilité de l'évolution de densité en tant qu'une condition suffisante pour que la probabilité d'erreur due aux mots de code de poids faible tende vers zéro.

Dans la **troisième partie**, nous utilisons le fait que le décodage itératif offre une solution à bas coût pour décoder des codes complexes. Ainsi, les systèmes de décodage itératif sont des candidats prometteurs pour de futures applications

filaires et non-filaires. En particulier nous avons étudié le potentiel des codes concaténés avec des codes constitutifs de type BCH pour les systèmes VDSL et ATM non-filaires.

L'apport principal de cette thèse est d'exposer le lien entre les différents moyens existant à ce jour pour appréhender le fonctionnement du décodage itératif, à savoir :

1. la théorie des systèmes dynamiques non-linéaires
2. les techniques classiques du codage canal qui se rapportent à l'étude de performance d'un code à partir de sa fonction énumératrice de poids.

Nous avons montré que ces procédés, a priori sans lien apparent, sont liés par l'intermédiaire de l'évolution de densité en ce qui concerne le point fixe au voisinage de zéro.

Perspectives

Les méthodes de contrôle du chaos pourraient être étudiées dans le futur pour amméliorer les performances du décodage itératif. De même, de futurs travaux pourraient se focaliser sur les approximations gaussiennes pour les codes convolutifs.

La construction de nouveaux ensembles de codes avec de bonnes propriétés en terme de distance et un décodeur itératif associé est un domaine de recherche prometteur.

Enfin, l'optimisation de codes produits pour les systèmes ATM sans fil pourrait être étendue aux codes Reed-Solomon.

Annexe A

Dérivée partielle de $f(x, \sigma)$ pour le canal AWGN

La dérivée partielle de la fonction donnée par l'équation (4.2.7) admet une expression analytique. En effet, on voit aisément que la dérivée partielle par rapport à x a pour expression :

$$\frac{\partial f}{\partial x}(x,\sigma) = e^{-\frac{1}{2\sigma^2}} \sum_{j=2}^{d_c} \rho_j (j-1) \left(1-2x\right)^{(j-2)} \times \sum_{i=2}^{d_v} \lambda_i \frac{(i-1)Q^{-1}\left(s(x)\right)}{\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s(x)\right)\right\}^2}} \exp\left[\left(1-\frac{i}{2}\right)\left\{Q^{-1}\left(s(x)\right)\right\}^2\right],$$

et si nous prenons la limite en $x = 0$, on peut vérifier que

$$\lim_{x\to 0} \frac{\partial f}{\partial x}(x,\sigma) = e^{-\frac{1}{2\sigma^2}} \lambda_2 \sum_{j=2}^{d_c} \rho_j (j-1) = e^{-\frac{1}{2\sigma^2}} \lambda'(0)\rho'(1)$$

De même, la dérivée partielle par rapport à σ a pour expression :

$$\frac{\partial f}{\partial \sigma}(x,\sigma) = \frac{e^{-\frac{1}{2\sigma^2}}}{\sigma^3\sqrt{2\pi}} \sum_{i=2}^{d_v} \lambda_i \frac{1}{\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s(x)\right)\right\}^2}} \exp\left[\frac{1}{2}(1-i)\left\{Q^{-1}\left(s(x)\right)\right\}^2\right] \tag{A.0.1}$$

Annexe B

Dérivée partielle de $f(x, \sigma)$ pour le canal de Rayleigh

La dérivée partielle de la fonction $f(x, \sigma)$ par rapport à x est donnée par :

$$\begin{aligned}\frac{\partial f(x,\sigma)}{\partial x} = \frac{2\sigma^2}{\sqrt{1+2\sigma^2}} \sum_{j=2}^{d_c} \rho_j (j-1)(1-2x)^{j-2} \\ \times \sum_{i=2}^{d_v} \lambda_i \Bigg\{ \sqrt{2\pi} \exp\left(\frac{1}{2}(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2\right) \\ Q\left(\sqrt{(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2}\right) \\ \times (i-1)Q^{-1}(s(x)) \exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right) \Bigg\}\end{aligned}$$

Notons que pour $x \to 0$: $Q^{-1}(s(x)) \to +\infty$ et en utilisant le fait que pour $y \to +\infty$: $Q(\sqrt{y}) \to \frac{1}{\sqrt{2\pi}} \frac{e^{-\frac{y}{2}}}{\sqrt{y}}$, il s'ensuit que $(x \to 0)$:

$$\begin{aligned}\frac{\partial f(x,\sigma)}{\partial x} \to \left(\sum_{j=2}^{d_c} \rho_j (j-1)\right) \frac{2\sigma^2}{\sqrt{1+2\sigma^2}} \\ \times \sum_{i=2}^{d_v} \lambda_i \frac{(i-1)Q^{-1}(s(x))}{\sqrt{(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2}} \exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right)\end{aligned}$$

et :

$$\frac{\partial f(x,\sigma)}{\partial x} \to \left(\sum_{j=2}^{d_c} \rho_j (j-1)\right) \frac{1}{1+\frac{1}{2\sigma^2}} \times \sum_{i=2}^{d_v} \lambda_i \sqrt{i-1} \exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right), \quad x \to 0$$

Finalement :

$$\lim_{x \to 0} \frac{\partial f}{\partial x}(x, \sigma) = \frac{1}{1 + \frac{1}{2\sigma^2}} \lambda_2 \sum_{j=2}^{d_c} \rho_j(j-1) = \frac{1}{1 + \frac{1}{2\sigma^2}} \lambda^{'}(0) \rho^{'}(1)$$

Annexe C

Analyse de l'algorithme de Chase

Supposons que dans une séquence reçue de longueur n, i erreurs de transmission binaires se soient produites. Les fiabilités des digits, définies comme la valeur absolue des rapports de vraisemblance du canal, sont réordonnées de la manière suivante

- les fiabilités correspondant aux i décisions fermes érronées sont réordonnées dans l'ordre décroissant : $\beta_1(i) \geq \beta_2(i) \geq \cdots \geq \beta_i(i)$.
- les fiabilités correspondant aux $n-i$ décisions fermes correctes sont réordonnées dans l'ordre décroissant : $\gamma_1(n-i) \geq \gamma_2(n-i) \geq \cdots \geq \gamma_{n-i}(n-i)$.

Soit $f_\alpha^c(x)$ (resp. $f_\alpha^e(x)$) la densité de probabilité associée aux fiabilités correspondant à des décisions fermes correctes (resp. incorrectes). Soit $F_\alpha^c(x)$ (resp. $F_\alpha^e(x)$) la fonction de répartition associée aux fiabilités correspondant à des décisions fermes correctes (resp. incorrectes). Alors les densités de probabilité des statistiques ordonnées $\beta_j(i)$ et $\gamma_l(n-i)$ sont données par by [59]

$$f_{\beta_j(i)}(x) = \frac{i!}{(j-1)!(i-j)!}[1 - F_\alpha^e(x)]^{j-1} f_\alpha^e(x)[F_\alpha^e(x)]^{i-j}$$
$$f_{\gamma_l(n-i)}(x) = \frac{(n-i)!}{(l-1)!(n-i-l)!}[1 - F_\alpha^c(x)]^{l-1} f_\alpha^c(x)[F_\alpha^c(x)]^{n-i-l}$$

Il s'ensuit que

$$P\left(\beta_j(i) \geq \gamma_l(n-i)\right) = \int_0^\infty f_{\gamma_l(n-i)}(x) \left(\int_x^\infty f_{\beta_j(i)}(y) dy \right) dx,$$

pour $1 \leq j \leq i$ et $1 \leq l \leq n-i$.

Annexe D

Fonctions approchées pour l'étude des CC

Afin d'étudier les propriétés des fonctions approchées introduite dans la Sec. 5.3, nous définissons la fonction

$$f(y,\sigma)=\epsilon Q\left(\sqrt{A\left([Q^{-1}(y)]^2-\frac{1}{\sigma^2}-\frac{1}{B}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2\right)}\right), \tag{D.0.1}$$

où A, B sont des entiers >1 et $y\geq 0$, ϵ sont des nombres réels. Sauf mention expresse du contraire, le paramètre de contrôle σ est un nombre réel positif fixé.

D.1. Limite de $f(x,\sigma)$ quand $y\to 0$

En utilisant l'inégalité $\left(1-\frac{1}{x^2}\right)<\sqrt{2\pi}xe^{x^2/2}Q(x)<1$, lorsque $x>0$, nous obtenons $\lim_{x\to+\infty}\sqrt{2\pi}Q(x)xe^{x^2/2}=1$. Quand $0<y<1/2$, choisissons $x=Q^{-1}(y)$ et $x=Q^{-1}\left(\frac{y}{\epsilon}\right)$ pour obtenir

$$\begin{aligned}&\lim_{y\to 0}\sqrt{2\pi}yQ^{-1}(y)e^{\frac{1}{2}[Q^{-1}(y)]^2}=1\\&\lim_{y\to 0}\sqrt{2\pi}\frac{y}{\epsilon}Q^{-1}\left(\frac{y}{\epsilon}\right)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}=1.\end{aligned} \tag{D.0.2}$$

Par conséquent,

$$\lim_{y\to 0}\frac{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}Q^{-1}\left(\frac{y}{\epsilon}\right)}{\epsilon e^{\frac{1}{2}[Q^{-1}(y)]^2}Q^{-1}(y)}=1. \tag{D.0.3}$$

D'après la règle L'Hôpital, nous avons

$$\lim_{y\to 0}\frac{Q^{-1}\left(\frac{y}{\epsilon}\right)}{Q^{-1}(y)}=\lim_{y\to 0}\frac{\frac{\partial Q^{-1}\left(\frac{y}{\epsilon}\right)}{\partial y}}{\frac{\partial Q^{-1}(y)}{\partial y}}=\lim_{y\to 0}\frac{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}Q^{-1}\left(\frac{y}{\epsilon}\right)}{\epsilon e^{\frac{1}{2}[Q^{-1}(y)]^2}Q^{-1}(y)}.$$

et (D.0.3) implique que

$$\lim_{y\to 0}\frac{Q^{-1}\left(\frac{y}{\epsilon}\right)}{Q^{-1}(y)}=1. \tag{D.0.4}$$

En utilisant l'inégalité $Q(x)<e^{-x^2/2}$ et (D.0.1), nous obtenons

$$0\leq f(y,\sigma)\leq \epsilon e^{\frac{A}{2\sigma^2}}\exp\left\{-\frac{A}{2}[Q^{-1}(y)]^2\left(1-\frac{1}{B}\frac{\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}{[Q^{-1}(y)]^2}\right)\right\}. \tag{D.0.5}$$

Etant donné que $A>1$, $B>1$ et $\lim_{y\to 0}Q^{-1}(y)=+\infty$, en combinant (D.0.4) et (D.0.5) on a $\lim_{y\to 0}f(y,\sigma)=0$.

D.2. Dérivée de $f(y,\sigma)$ par rapport à y

L'expression de la dérivée partielle par rapport à y s'écrit

$$\frac{\partial f(y,\sigma)}{\partial y}=C(A,B,\epsilon)\gamma(y,B,\epsilon)\delta(y,A,B,\epsilon)\geq 0, \tag{D.0.6}$$

où

$$\begin{aligned}
C(A,B,\epsilon)=&\epsilon^2 e^{\frac{A}{2\sigma^2}}\sqrt{A\left(1-\frac{1}{B}\right)}\\
\gamma(y,B,\epsilon)=&\sqrt{1-\frac{1}{B}}\frac{\frac{[Q^{-1}(y)]^2}{\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}}{\sqrt{\frac{[Q^{-1}(y)]^2}{\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}-\frac{1}{B}-\frac{1/\sigma^2}{\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}}}\\
&\times\frac{1}{B-1}\left(B-\frac{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}Q^{-1}\left(\frac{y}{\epsilon}\right)}{\epsilon e^{\frac{1}{2}[Q^{-1}(y)]^2}Q^{-1}(y)}\right)\\
\delta(y,A,B,\epsilon)=&\frac{e^{\frac{A}{2B}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}Q^{-1}\left(\frac{y}{\epsilon}\right)}{\epsilon e^{\frac{A-1}{2}[Q^{-1}(y)]^2}Q^{-1}(y)}.
\end{aligned} \tag{D.0.7}$$

Etant donné que $\lim_{y\to 0}Q^{-1}\left(\frac{y}{\epsilon}\right)=+\infty$, d'après (D.0.3) et (D.0.4), nous avons immédiatement $\lim_{y\to 0}\gamma(y,B,\epsilon)=1$. Donc, la limite de $\frac{\partial f(y,\sigma)}{\partial y}$ est déterminée uniquement par le comportement de $\delta(y,A,B,\epsilon)$, pour $y\to 0$.

Dans l'Annexe D.1, nous supposons que A et B parcourent les entiers strictement supérieurs à un, en conséquence nous pouvons introduire deux cas distincts, à savoir $A=B=2$ and $\max(A,B)>2$.

D.2.1. Cas $A = B = 2$

En utilisant (D.0.3), nous obtenons $\lim_{y\to 0}\delta(y, A, B, \epsilon) = 1$, donc

$$\lim_{y\to 0}\frac{\partial f(y,\sigma)}{\partial y} = C(2,2,\epsilon) = \left(\epsilon e^{\frac{1}{2\sigma^2}}\right)^2. \tag{D.0.8}$$

D.2.2. Cas $\max(A, B) > 2$

D'après (D.0.2)

$$\begin{aligned}
&\lim_{y\to 0}\left(\sqrt{2\pi}yQ^{-1}(y)e^{\frac{1}{2}[Q^{-1}(y)]^2}\right)^{A-1} = 1\\
&\lim_{y\to 0}\left(\sqrt{2\pi}\frac{y}{\epsilon}Q^{-1}\left(\frac{y}{\epsilon}\right)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^2}\right)^{\frac{A}{B}} = 1.
\end{aligned} \tag{D.0.9}$$

Il s'ensuit que

$$\begin{aligned}
\lim_{y\to 0}\delta(y, A, B, \epsilon) &= \lim_{y\to 0}\frac{(\sqrt{2\pi})^{A-1}y^{A-1}[Q^{-1}(y)]^{A-2}}{(\sqrt{2\pi})^{\frac{A}{B}}\frac{1}{\epsilon^{\frac{A}{B}-1}}y^{\frac{A}{B}}\left[Q^{-1}\left(\frac{y}{\epsilon}\right)\right]^{\frac{A}{B}-1}}\\
&= \lim_{y\to 0}(\sqrt{2\pi})^{A\left(1-\frac{1}{B}\right)-1}\epsilon^{\frac{A}{B}-1}\left(\frac{Q^{-1}\left(\frac{y}{\epsilon}\right)}{Q^{-1}(y)}\right)^{1-\frac{A}{B}}\left(yQ^{-1}(y)\right)^{A\left(1-\frac{1}{B}\right)-1}.
\end{aligned}$$

Finalement, (D.0.4) conduit à

$$\lim_{y\to 0}\delta(y, A, B, \epsilon) = (\sqrt{2\pi})^{A\left(1-\frac{1}{B}\right)-1}\epsilon^{\frac{A}{B}-1}\times\lim_{y\to 0}\left(yQ^{-1}(y)\right)^{A\left(1-\frac{1}{B}\right)-1}. \tag{D.0.10}$$

D.2.2.1. Limite de $\frac{\partial f(y,\sigma)}{\partial y}$ quand $y \to 0$

Nous utilisons l'inégalité $Q(x) < e^{-x^2/2}$, $x > 0$. Lorsque $0 < y < 1/2$, choisissons $x = Q^{-1}(y)$ et après quelques manipulations évidentes nous obtenons la borne

$$0 < Q^{-1}(y) < \sqrt{-2\ln y}.$$

Il est immédiat que

$$0 < \left(yQ^{-1}(y)\right)^{A\left(1-\frac{1}{B}\right)-1} < \left(-2y^2\ln y\right)^{\frac{1}{2}\left[A\left(1-\frac{1}{B}\right)-1\right]}. \tag{D.0.11}$$

Notons que $\lim_{y\to 0} y^2\ln y = 0$ et $A\left(1-\frac{1}{B}\right)-1 > 0$, puisque $A > 1$ et $B > 1$, donc en combinant (D.0.11) avec (D.0.10) il resulte que $\lim_{y\to 0}\delta(y, A, B, \epsilon) = 0$. Nous concluons que $\lim_{y\to 0}\frac{\partial f(y,\sigma)}{\partial y} = 0$ quand $\max(A, B) > 2$.

D.2.2.2. Limite de $\frac{f(y,\sigma)}{y^r}$ quand $y \to 0$ pour $r > 0$

Nous recherchons

$$\lim_{y\to 0} \frac{\delta(y,A,B,\epsilon)}{ry^{r-1}} = (\sqrt{2\pi})^{A\left(1-\frac{1}{B}\right)-1} \frac{\epsilon^{\frac{A}{B}-1}}{r} \times \lim_{y\to 0} y^{A\left(1-\frac{1}{B}\right)-r} \left(Q^{-1}(y)\right)^{A\left(1-\frac{1}{B}\right)-1}. \tag{D.0.12}$$

Commençons par évaluer les limites des fonctions suivantes

$$\eta(y) = y^{A\left(1-\frac{1}{B}\right)-r}(-2\ln y)^{\frac{1}{2}\left[A\left(1-\frac{1}{B}\right)-1\right]}$$

$$\theta(y) = \left(1 + \frac{\ln\sqrt{2\pi}}{\ln y} + \frac{\ln Q^{-1}(y)}{\ln y} - \frac{\ln\left(1-\frac{1}{[Q^{-1}(y)]^2}\right)}{\ln y}\right)^{\frac{1}{2}\left[A\left(1-\frac{1}{B}\right)-1\right]}.$$

Lorsque $y \to 0$, nous avons $Q^{-1}(y) \to +\infty$, $-\ln y \to +\infty$ et en employant le changement de variable $X = -2\ln y$ pour $0 < y < 1/2$, nous avons

$$\left|\frac{\ln Q^{-1}(y)}{\ln y}\right| < \frac{\ln\left(\sqrt{-2\ln y}\right)}{-\ln y} = \frac{\ln X}{X} \to 0,$$

c'est pourquoi $\lim_{y\to 0}\theta(y) = 1$.

Avec le changement de variable $Y = (-\ln y)^{\frac{1}{2}\left[A\left(1-\frac{1}{B}\right)-1\right]}$, pour $0 < y < 1/2$, $y \to 0$ est équivalent à $Y \to +\infty$ et

$$\eta(y) = 2^{\frac{1}{2}\left[A\left(1-\frac{1}{B}\right)-1\right]} Y \exp\left[\left(r - A\left(1-\frac{1}{B}\right)\right) Y^{\frac{2}{A\left(1-\frac{1}{B}\right)-1}}\right].$$

De $A\left(1-\frac{1}{B}\right)-1 > 0$, nous déduisons que

$$\lim_{y\to 0}\eta(y) = \begin{cases} 0, & \text{si } r < A\left(1-\frac{1}{B}\right) \\ +\infty, & \text{sinon} \end{cases}. \tag{D.0.13}$$

Cas $\mathbf{r < A\left(1-\frac{1}{B}\right)}$

Pour $0 < y < 1/2$, nous utilisons $Q^{-1}(y) < \sqrt{-2\ln y}$ et obtenons

$$0 < y^{A\left(1-\frac{1}{B}\right)-r}[Q^{-1}(y)]^{A\left(1-\frac{1}{B}\right)-1} < \eta(y).$$

En combinant ce résultat avec (D.0.13) on a $\lim_{y\to 0}\frac{\delta(y,A,B,\epsilon)}{ry^{r-1}} = 0$.

Cas $\mathbf{r \geq A\left(1-\frac{1}{B}\right)}$

Utilisons l'inégalité $\frac{1}{\sqrt{2\pi}x}\left(1-\frac{1}{x^2}\right)e^{-x^2/2} < Q(x)$, pour $x > 0$. Pour $0 < y < 1/2$, choisissons $x = Q^{-1}(y) > 0$ et après quelques manipulations, nous obtenons

la borne

$$Q^{-1}(y) > (-2\ln y)^{1/2}\left(1+\frac{\ln\sqrt{2\pi}}{\ln y}+\frac{\ln Q^{-1}(y)}{\ln y}-\frac{\ln\left(1-\frac{1}{[Q^{-1}(y)]^2}\right)}{\ln y}\right)^{1/2}.$$

Ensuite, en prenant en compte le fait que $A\left(1-\frac{1}{B}\right)-1>0$, il suit que

$$y^{A\left(1-\frac{1}{B}\right)-r}[Q^{-1}(y)]^{A\left(1-\frac{1}{B}\right)-1} > \eta(y)\times\theta(y).$$

En combinant ce résultat avec (D.0.13) on a $\lim_{y\to 0}\frac{\delta(y,A,B,\epsilon)}{ry^{r-1}}=+\infty$.

Si r est un entier, définissons

$$r(A,B)=\max\left\{r\in\mathbb{N}:\lim_{y\to 0}\frac{f(y,\sigma)}{y^r}=0\right\}.$$

En appliquant la règle de L'Hôpital, nous concluons que

$$r(A,B)=\max\left\{r\in\mathbb{N}:\lim_{y\to 0}\frac{1}{ry^{r-1}}\frac{\partial f(y,\sigma)}{\partial y}=0\right\}.$$

Finalement, lorsque $\max(A,B)>2$, nous avons

$$r(A,B)=\begin{cases} A\left(1-\frac{1}{B}\right)-1, & \text{si } A \mod B=0 \\ \left\lfloor A\left(1-\frac{1}{B}\right)\right\rfloor, & \text{sinon}\end{cases}. \tag{D.0.14}$$

Annexe E

Dérivée partielle de la fonction représentant le décodage itératif des PCC

La fonction h représentant le décodage itératif des PCC a une dérivée partielle par rapport à y de la forme

$$\frac{\partial h(y,\sigma)}{\partial y} = \left[-2\sqrt{2\pi}Q^{-1}(y)\exp\left(\frac{1}{2}[Q^{-1}(y)]^2\right) - \left(\frac{\partial P(x,\sigma)}{\partial x}\Big|_{x=P^{-1}(y,\sigma)}\right)^{-1}\right] \times \frac{\partial P(x,\sigma)}{\partial x}\Big|_{x=[Q^{-1}(y)]^2-\frac{1}{\sigma^2}-P^{-1}(y,\sigma)}, \tag{E.0.1}$$

où $P(x,\sigma)$ est défini dans la Sec. 5.2.1. En un point fixe $y > 0$, $[Q^{-1}(y)]^2 - \frac{1}{\sigma^2} - P^{-1}(y,\sigma) = P^{-1}(y,\sigma)$, ainsi, la dérivée partielle se simplifie en

$$\frac{\partial h(y,\sigma)}{\partial y} = -2\sqrt{2\pi}Q^{-1}(y)\exp\left(\frac{1}{2}[Q^{-1}(y)]^2\right) \times \frac{\partial P(x,\sigma)}{\partial x}\Big|_{x=P^{-1}(y,\sigma)} - 1 \tag{E.0.2}$$

En remplaçant $P(x,\sigma)$ dans (E.0.2) par l'approximation suggérée par (5.3.10),

$$\frac{\partial h(y,\sigma)}{\partial y}\Big|_{y=y^*} = \frac{\alpha(\sigma)e^{\frac{1}{2}\left[Q^{-1}(y^*)\right]^2}Q^{-1}(y^*)}{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha(\sigma)}\right)} - 1, \tag{E.0.3}$$

où $y^* > 0$ est un point fixe de h. En choisissant $\epsilon = \alpha(\sigma)$ dans (D.0.3) on obtient $\lim_{y^*\to 0}\frac{\partial h(y,\sigma)}{\partial y}\Big|_{y=y^*} = 0$.

Annexe F

Dérivée partielle des fonctions représentant le décodage itératif des SCC

Les fonctions h_o et h_i représentant le décodage itératif des SCC ont une dérivée partielle par rapport à y de la forme

$$\begin{aligned}\frac{\partial h_o(y,\sigma)}{\partial y} &= \left[-2\sqrt{2\pi}Q^{-1}(y)\exp\left(\frac{1}{2}[Q^{-1}(y)]^2\right) - \left(\frac{\partial P_i(x,\sigma)}{\partial x}\Big|_{x=P_i^{-1}(y,\sigma)}\right)^{-1}\right] \\ &\quad \times \frac{\partial P_o(x,\sigma)}{\partial x}\Big|_{x=[Q^{-1}(y)]^2-\frac{1}{\sigma^2}-P_i^{-1}(y,\sigma)} \\ \frac{\partial h_i(y,\sigma)}{\partial y} &= \left[-2\sqrt{2\pi}Q^{-1}(y)\exp\left(\frac{1}{2}[Q^{-1}(y)]^2\right) - \left(\frac{\partial P_o(x,\sigma)}{\partial x}\Big|_{x=P_o^{-1}(y,\sigma)}\right)^{-1}\right] \\ &\quad \times \frac{\partial P_i(x,\sigma)}{\partial x}\Big|_{x=[Q^{-1}(y)]^2-\frac{1}{\sigma^2}-P_o^{-1}(y,\sigma)},\end{aligned} \tag{F.0.1}$$

où $P_o(x,\sigma)$ et $P_i(x,\sigma)$ sont définis dans la Sec. 5.2.2. En remplaçant $P_o(x,\sigma)$ et $P_i(x,\sigma)$ dans (F.0.1) par les approximations données par (5.3.17),

$$\begin{aligned}\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*} &= d^o_{min}\left(\frac{\alpha_o(\sigma)e^{\frac{1}{2}\left[Q^{-1}(y^*)\right]^2}Q^{-1}(y^*)}{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)} - \frac{\alpha_o(\sigma)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)}{\alpha_i(\sigma)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)}\right) \\ \frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*} &= \frac{\alpha_i(\sigma)e^{\frac{1}{2}\left[Q^{-1}(y^*)\right]^2}Q^{-1}(y^*)}{e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)} - \frac{1}{d^o_{min}}\frac{\alpha_i(\sigma)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_o(\sigma)}\right)}{\alpha_o(\sigma)e^{\frac{1}{2}\left[Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)\right]^2}Q^{-1}\left(\frac{y^*}{\alpha_i(\sigma)}\right)},\end{aligned} \tag{F.0.2}$$

où $y^* > 0$ est un point fixe de h_o et h_i. A l'aide d'une méthode similaire à la démonstration de (D.0.3), nous avons $\lim_{y^* \to 0} \frac{\partial h_o(y,\sigma)}{\partial y}\big|_{y=y^*} = 0$ et $\lim_{y^* \to 0} \frac{\partial h_i(y,\sigma)}{\partial y}\big|_{y=y^*} = 1 - \frac{1}{d^o_{min}}$.

Annexe G

Démonstration du Thm. 5.2.2

Nous donnons ici la preuve du Thm. 5.2.2 énoncé dans la Sec. 5.2.2.

Nous montrons d'abord que sous les conditions 1 à 3, les points fixes de $h_o \circ h_i$ et $h_i \circ h_o$ sont exactement les points fixes h_o et h_i. Soit $y^* \in J$ un point fixe de h_o et h_i ; par définition $h_o(y^*, \sigma) = y^*$ et $h_i(y^*, \sigma) = y^*$, alors

$$h_o \circ h_i(y^*, \sigma) = h_o(y^*, \sigma) = y^*$$
$$h_i \circ h_o(y^*, \sigma) = h_i(y^*, \sigma) = y^*.$$

C'est pourquoi, les points fixes de h_o et h_i sont un sous-ensemble des points fixes de $h_o \circ h_i$ et $h_i \circ h_o$. Réciproquement, soit $y^* \in J$ un point fixe de $h_o \circ h_i$, par définition $h_o \circ h_i(y^*, \sigma) = y^*$. Supposons que $h_i(y^*, \sigma) \neq y^*$, la condition 3 implique que $h_i(y^*, \sigma) < y^*$ et à partir de la condition 1, nous avons $h_o \circ h_i(y^*, \sigma) = y^* \leq h_o(y^*, \sigma)$. En appliquant la condition 2, il s'ensuit que $h_o(y^*, \sigma) = y^*$. Puisque h_i et h_o ont les mêmes points fixes, $h_i(y^*, \sigma) = y^*$, ce qui est en contradiction avec l'hypothèse. De plus, puisque h_i et h_o ont les mêmes points fixes, $h_o(y^*, \sigma) \neq y^*$ implique $h_i(y^*, \sigma) \neq y^*$ et conduit à la même contradiction. Donc y^* est un point fixe de h_i et h_o. Le même raisonnement s'applique à $h_i \circ h_o$. Par conséquent, les points fixes de $h_o \circ h_i$ et $h_i \circ h_o$ sont un sous-ensemble des points fixes de h_o et h_i.

Nous achevons la preuve en montrant que sous les conditions 1 à 4, les points fixes stables de $h_o \circ h_i$ et $h_i \circ h_o$ sont exactement les points fixes stables de h_o and h_i. Puisque nous avons déjà montré que sous les conditions 1 à 3, h_o, h_i, $h_o \circ h_i$

et $h_i \circ h_o$ ont les mêmes points fixes, soit $y^* \in J$ un tel point fixe, alors

$$\begin{aligned}
\left|\frac{\partial h_o \circ h_i}{\partial y}(y,\sigma)\Big|_{y=y^*}\right| &= \left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=h_i(y^*,\sigma)} \times \frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \\
&= \left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \times \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \\
\left|\frac{\partial h_i \circ h_o}{\partial y}(y,\sigma)\Big|_{y=y^*}\right| &= \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=h_o(y^*,\sigma)} \times \frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \\
&= \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \times \left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right|.
\end{aligned} \tag{G.0.1}$$

Soit $y^* \in J$ un point fixe stable de h_o et h_i ; par définition, nous avons

$$\begin{aligned}
\left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| &< 1 \\
\left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| &< 1.
\end{aligned}$$

Nous déduisons de (G.0.1) que les points fixes stables de h_o et h_i sont un sous-ensemble des points fixes stables de $h_o \circ h_i$ et $h_i \circ h_o$. Réciproquement, soit $y^* \in J$ un point fixe stable de $h_o \circ h_i$ et $h_i \circ h_o$, par définition

$$\begin{aligned}
\left|\frac{\partial h_o \circ h_i}{\partial y}(y,\sigma)\Big|_{y=y^*}\right| &< 1 \\
\left|\frac{\partial h_i \circ h_o}{\partial y}(y,\sigma)\Big|_{y=y^*}\right| &< 1.
\end{aligned}$$

En prenant en compte la constrainte donnée par les conditions 2 et 3, (G.0.1) impose

$$\left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \leq 1 \quad \text{et} \quad \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| < 1,$$

ou

$$\left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| < 1 \quad \text{et} \quad \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| \leq 1$$

et finalement, à cause de la condition 4, nous avons nécéssairement

$$\left|\frac{\partial h_o(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| < 1 \quad \text{et} \quad \left|\frac{\partial h_i(y,\sigma)}{\partial y}\Big|_{y=y^*}\right| < 1.$$

C'est pourquoi les points fixes stables de $h_o \circ h_i$ and $h_i \circ h_o$ sont un sous-ensemble des points fixes stables de h_o and h_i.

Annexe H

Démonstration de l'Eq. (6.3.20)

Nous montrons comment obtenir le résultat (6.3.20) nécéssaire pour démontrer la seconde partie du Thm. 6.3.3. En procédant à la substitution

$$z = \frac{1-e^s}{1+e^s}$$
$$s = \ln \frac{1-z}{1+z},$$

on obtient

$$\begin{aligned}
\sum_j \frac{\rho_j}{j} \mu_j'(s) &= \sum_j \rho_j \frac{e^s \left[(1+e^s)^{j-1} - (1-e^s)^{j-1}\right]}{(1+e^s)^j + (1-e^s)^j} \\
&= \frac{1-z}{2} \sum_j \rho_j \frac{1-z^{j-1}}{1+z^j} \\
&= \frac{1-z}{2} \left\{ \rho_{d_c} \frac{1-z^{d_c-1}}{1+z^{d_c}} + \sum_{j<d_c} \rho_j \frac{1-z^{j-1}}{1+z^j} \right\} \\
&= \frac{1-z}{2} \left\{ \rho_{d_c} \frac{1-z^{d_c-1}}{1+z^{d_c}} + \sum_{j<d_c} \rho_j \frac{(1-z^{d_c-1})(1+z^j) + (1+z)(z^{d_c-1} - z^{j-1})}{(1+z^{d_c})(1+z^j)} \right\} \\
&= \frac{1-z}{2} \left\{ \rho_{d_c} \frac{1-z^{d_c-1}}{1+z^{d_c}} + \sum_{j<d_c} \rho_j \frac{1-z^{d_c-1}}{1+z^{d_c}} + \frac{1+z}{1+z^{d_c}} \sum_{j<d_c} \rho_j \frac{z^{d_c-1} - z^{j-1}}{1+z^j} \right\} \\
&= \frac{1-z}{2} \frac{1-z^{d_c-1}}{1+z^{d_c}} (1+\epsilon_1(z)),
\end{aligned}$$

où

$$\epsilon_1(z) = \frac{1+z}{1-z^{d_c-1}} \sum_{j<d_c} \rho_j \frac{z^{d_c-1} - z^{j-1}}{1+z^j}.$$

Notons qu'en utilisant l'identité $(1-z^n) = (1-z)(1+z+\cdots+z^{n-1})$, nous avons

$$\begin{aligned}\lim_{z\to 1}\epsilon_1(z) &= \lim_{z\to 1} -\frac{1+z}{1-z^{d_c-1}}\sum_{j<d_c}\rho_j\frac{z^{j-1}(1-z^{d_c-j})}{1+z^j}\\ &= \lim_{z\to 1} -\frac{1+z}{1+z+\cdots+z^{d_c-2}}\sum_{j<d_c}\rho_j\frac{z^{j-1}(1+z+\cdots+z^{d_c-j-1})}{1+z^j}\\ &= -\frac{1}{d_c-1}\sum_{j<d_c}\rho_j(d_c-j).\end{aligned}$$

De l'expression de δ obtenue dans le Thm. 6.3.2, il suit que

$$\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta = \frac{1-z}{2}\frac{1-z^{d_c-1}}{1+z^{d_c}}(1+\epsilon_1(z))$$

et après quelques manipulations algébriques simples

$$1-\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta = \frac{1+z}{2}\frac{1+z^{d_c-1}}{1+z^{d_c}}(1-\epsilon_2(z)),$$

où

$$\epsilon_2(z) = \frac{1-z}{1+z^{d_c-1}}\sum_{j<d_c}\rho_j\frac{z^{d_c-1}-z^{j-1}}{1+z^j}.$$

Notons que $\lim_{z\to 1}\epsilon_2(z) = 0$. De plus, nous obtenons

$$\frac{\alpha_i}{a_i}\delta = \frac{\alpha_i}{a_i}\frac{\sum_i ia_i}{\sum_i i\alpha_i}\left(\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta\right) = \frac{\alpha_i}{a_i}\frac{\sum_i ia_i}{\sum_i i\alpha_i}\frac{1-z}{2}\frac{1-z^{d_c-1}}{1+z^{d_c}}(1+\epsilon_1(z))$$

et après quelques manipulations

$$1-\frac{\alpha_i}{a_i}\delta = \frac{\alpha_i}{a_i}\frac{\sum_i ia_i}{\sum_i i\alpha_i}\frac{1+z}{2}\frac{1+z^{d_c-1}}{1+z^{d_c}}(\Gamma_i(z)-\epsilon_2(z)),$$

où

$$\Gamma_i(z) = \frac{\left(2\frac{a_i}{\alpha_i}\frac{\sum_i i\alpha_i}{\sum_i ia_i}-1\right)(1+z^{d_c})+z+z^{d_c-1}}{(1+z)(1+z^{d_c-1})}.$$

Notons que

$$\lim_{z\to 1}\Gamma_i(z) = \frac{a_i}{\alpha_i}\frac{\sum_i i\alpha_i}{\sum_i ia_i}.$$

Finalement, en combinant ces résultats, nous obtenons

$$\begin{aligned}\frac{1-\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta}{\frac{\sum_i i\alpha_i}{\sum_i ia_i}\delta} &= \frac{1+z}{1-z}\frac{1+z^{d_c-1}}{1-z^{d_c-1}}\frac{1-\epsilon_2(z)}{1+\epsilon_1(z)} \\ \frac{1-\frac{\alpha_i}{a_i}\delta}{\frac{\alpha_i}{a_i}\delta} &= \frac{1+z}{1-z}\frac{1+z^{d_c-1}}{1-z^{d_c-1}}\frac{\Gamma_i(z)-\epsilon_2(z)}{1+\epsilon_1(z)}.\end{aligned} \tag{H.0.1}$$

Le résultat (6.3.20) est obtenu en injectant (H.0.1) dans (6.3.19).

Bibliographie

[1] B. Sklar, "A primer on turbo code concepts," *IEEE Comm. Mag.*, pp. 94-102, December 1997.

[2] C.E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379-423, pp. 623-56, 1948.

[3] R.G. Gallager, "Low Density Parity Check Codes," *IRE Trans. Inf. Theory*, vol. 8, pp. 21-28, 1962.

[4] R.G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA, MIT Press, 1963.

[5] R.M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, pp. 533-547, September 1981.

[6] T.J. Richardson and R.L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 599-618, February 2001.

[7] T.J. Richardson, M.A. Shokrollahi and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 619-637, February 2001.

[8] D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of Low Density Parity Check Codes," *Electronics Letters*, vol. 32, pp. 1645-46, 1996.

[9] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498-519, Feb. 2001.

[10] J. Hou, P.H. Siegel, L.B. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," *IEEE JSAC*, vol. 19, no. 5, pp. 924-34 May 2001.

[11] L. Bazzi, T. Richardson, R. Urbanke, "Exact thresholds and optimal codes for the binary symmetric channel and Gallager's decoding algorithm A," *Proc ISIT 2000*, Sorrento, Italy, June 2000.

[12] M.A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," *Proc. AAECC'99 : 13th AAECC Symposium on Ap-*

plied Algebra, Algebraic Algorithms, and Error-Correcting Codes, p.65-76, Honolulu, USA, Nov. 1999.

[13] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. 4, pp.29-37, September 1954.

[14] G.Forney, *Concatenated codes.* Cambridge, MA, MIT Press, 1966.

[15] P. Thitimajshima, *Les codes convolutifs récursifs systématiques et leur application à la concaténation parallèle*, Ph.D. Thesis, Université de Bretagne Occidentale, December 1993.

[16] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : turbo codes," *Proc. IEEE Int. Conf. Comm.,* pp. 1064-1070, Geneva, Switzerland, May 1993.

[17] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, "Near optimum decoding of product codes," *Proc. GLOBECOM'94,* vol. 1, pp. 339-343, San Francisco, CA, November-December 1994.

[18] R.M. Pyndiah, "Near optimum decoding of product codes : block turbo codes," *IEEE Trans. Comm.*, vol. 46, no. 8, pp. 1003-1010, August 1998.

[19] S. Benedetto, G. Montorsi, "Unveiling turbo-codes : some results on parallel concatenated coding schemes," *IEEE Trans. Inf. Theory,* vol. 42, no. 2, pp. 409-428, March 1996.

[20] D. Divsalar and F. Pollara, "On the design of turbo codes," *TDA Progress Report,* vol. 42-123, pp. 99-121, November 1995.

[21] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, "Serial concatenation of interleaved codes : performance analysis, design and iterative decoding," *IEEE Trans. Inf. Theory,* vol. 44, no. 3, pp. 909-926, May 1998.

[22] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *TDA Progress Report,* vol. 42-122, pp. 56-65, August 1995.

[23] F. Daneshgaran and M. Mondin, "Design of interleavers for turbo codes : iterative interleaver growth algorithms of polynomial complexity," *IEEE Trans. Inf. Theory,* vol. 45, no. 6, pp. 1845-1959, September 1999.

[24] J.G. Proakis, *Digital Communications.* New-York, NY, McGraw-Hill, 2001.

[25] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, Vol. 20, pp. 284-287, March 1974.

[26] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," *Proc. GLOBECOM'94*, pp. 1298-1303, 1994.

[27] J. Hagenauer, P. Robertson, L. Papke, "Iterative ("TURBO") decoding of systematic convolutional codes with MAP and SOVA algorithms," *Proc. ITG'94*, 1994.

[28] J. Lodge, R, Young, P. Hoeher, J. Hagenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," *Proc. ICC'93*, Geneva, Switzerland, pp. 1740-1745, May 1993.

[29] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log-domain," *Proc. ICC'95*, Seattle, Washington, pp. 1009-1013, June 1995.

[30] J. Hagenauer, E. Offer, L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE. Trans. Inform. Theory*, Vol. 42, No. 2, pp. 284-287, March 1996.

[31] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes," *TDA Progress rep. 42-124*, Jet Propulsion Lab., Pasadena, CA, pp. 63-87, February 1996.

[32] 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 rep. 42-127*, Jet Propulsion Lab., Pasadena, CA, pp. 63-87, November 1996.

[33] H. Koorapaty, S. Chennakeshu, Y.P. Wang and R. Ramesh, "MAP decoding for satellite channels," *Proc. IEEE Veh. Tech. Conf.*, pp. 477-481, 1996.

[34] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "A Soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, Vol. 1, No. 1, pp. 22-24, January 1997.

[35] P. Robertson, E. Villebrun and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecommun.*, Vol. 8, pp. 119-125, March/April 1997.

[36] J.K. Wolf, "Efficient maximum likelihood decoding of linear block codes," *IEEE. Trans. Inform. Theory*, Vol. 24, pp. 76-80, January 1978.

[37] F.R. Kschischang and V. Sorokine, "On the treillis structure of block codes," *IEEE. Trans. Inform. Theory*, Vol. 41, pp. 1924-1937, November 1995.

[38] R.J. McEliece, "On the BCJR treillis for linear block codes," *IEEE. Trans. Inform. Theory*, Vol. 42, No. 4, pp. 1072-1092, July 1996.

[39] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE. Trans. Inform. Theory*, Vol. 13, pp. 260-269, April 1967.

[40] G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi," *Annales des Télécommunications*, Vol. 42, No. 1-2, pp. 31-38, Janvier-Fevrier 1987.

[41] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *GLOBECOM'89*, Dallas, Texas, pp. 1680-1686, November 1989.

[42] C. Berrou, P, Adde, E. Angui and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," *Proc. ICC'93*, Geneva, Switzerland, pp. 737-740, May 1993.

[43] J. Hagenauer and L. Papke, "Decoding 'turbo' codes with the soft output Viterbi algorithm (SOVA)," *Proc. Int. Symp. on Information Theory*, Trondheim, Norway, p. 164, June 1994.

[44] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, pp. 170-182, January 1972.

[45] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 9-23, January 2000.

[46] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 699-722, February 2001.

[47] S.H. Strogatz, *Nonlinear dynamics and chaos.* Cambridge, MA, Perseus publishing, 1994.

[48] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos.* Springer-Verlag, New-York, 1990.

[49] Z. Tasev, L. Kocarev, F. Lehmann and G.M. Maggio, "Nonlinear phenomena in the turbo decoding algorithm," *accepted at NOLTA 2002*, Xi'an, PRC, October 2002.

[50] T. Richardson and R. Urbanke, "Thresholds for turbo codes", *Proc. ISIT 2000*, vol. 10, p. 317, Sorrento, Italy, June 2000.

[51] N. Wiberg, *Codes and decoding on general graphs.* Ph.D. Thesis, Linköping University, Sweden, 1996.

[52] S.-Y. Chung, T.J. Richardson and R.L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 657-670, February 2001.

[53] H. El Gammal and A.R. Hammons, "Analysis the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 671-686, February 2001.

[54] D. Divsalar, S. Dolinar and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *J. Select. Areas Comm.,* vol. 19, no. 5, pp. 891-907, May 2001.

[55] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters,* pp. 806-808, vol. 35, no. 10, May 1999.

[56] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Comm.,* Vol. 49, No. 10, pp. 1727-1737, October 2001.

[57] F. Lehmann and G. M. Maggio, "An approximate analytical model of the message passing decoder of LDPC codes," *ISIT 2002,* Lausanne, Switzerland, July 2002.

[58] P. Rusmevichientong and B. Van Roy, "An analyzis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 745-765, February 2001.

[59] M.P.C. Fossorier and S. Lin, "Error performance analysis for reliability-based decoding algorithms," *IEEE Trans. Inf. Theory,* vol. 48, no. 1, pp. 287-293, January 2002.

[60] M. Lentmaier, D.V. Truhachev and K. Sh. Zigangirov, "On the theory of low-density convolutional codes," *Probl. Peredach. Inform.,* vol. 37, no. 4, pp. 288-306, October-December 2001.

[61] S. Benedetto, G. Montorsi, R. Garello, "Concatenated codes with interleaver for digital transmission over mobile channels", *Proc. ICC 2001*, vol. 10, pp. 3026 -3030, Helsinki, Finland, June 2001.

[62] R.W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," *Bell System Tech. J.,* vol. 45, pp. 1775-1796, December 1966.

[63] B.R. Saltzberg, "Performance of an efficient parallel transmission system," *IEEE Trans. Comm. Tech.,* vol. 15, no. 6, pp. 805-811, December 1967.

[64] S.B. Weinstein and P.M. Ebert, "Performance of an efficient parallel transmission system," *IEEE Trans. Comm. Tech.,* vol. 19, no. 5, pp. 628-634, October 1971.

[65] ANSI, "Very-high bit-rate digital subscriber lines (VDSL) metallic interface, Part 3 : technical specification of a multi-carrier modulation transceiver", T1E1.4/2000-013R4, Savannah, GA, November 2000.

[66] I. Kalet,"The multitone channel," *IEEE Trans. Comm.,* vol. 37, no. 2, pp. 119-124, February 1989.

[67] J.A.C. Bingham, "Multicarrier modulation for data transmission : an idea whose time has come," *IEEE Comm. Mag.,* vol. 28, no. 5, pp. 5-14, May 1990.

[68] J.C. Tu and J.M. Cioffi, "A loading algorithm for the concatenation of coset codes with multichannel modulation methods," *Proc. GLOBECOM'90,* San Diego, CA, December 1990.

[69] A. Ruiz, J.M. Cioffi and S. Kasturia, "Discrete multiple tone modulation with coset coding for the spectrally shaped channel," *IEEE Trans. Comm.,* vol. 40, no. 6, pp. 1012-1029, June 1992.

[70] T.N. Zogakis, J.T. Aslanis and J.M. Cioffi, "A coded and shaped discrete multitone system," *IEEE Trans. Comm.,* vol. 43, no. 12, pp. 2941-2949, December 1995.

[71] E. Zehavi, "8-PSK trellis code for a Rayleigh channel," *IEEE Trans. Comm.,* vol. 40, no. 5, pp. 873-884, May 1992.

[72] C. Caire, G. Taricco and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory,* vol. 44, no. 3, pp. 932-946, May 1998.

[73] ANSI, "Very-high bit-rate digital subscriber lines (VDSL) metallic interface, Part 1 : requirements and common specification", T1E1.4/2000-009R2, Savannah, GA, November 2000.

[74] R.D. Wesel, Xueting Liu, J.M. Cioffi, C. Komminakis, "Constellation labeling for linear encoders," *IEEE Trans. Inf. Theory,* vol. 47, no. 6, pp. 2417-2431, September 2001.

[75] T.A. Gulliver, "Matching Q-ary Reed-Solomon codes with M-ary modulation," *IEEE Trans. Comm.,* vol. 45, no. 11, pp. 1349-1353, November 1997.

[76] ETSI, "Transmission and Multiplixing (TM), Access Transmission systems on metallic access cables, Very high speed Digital Subscriber Line (VDSL)," Technical Specification 101 270-1 v 1.2.1, October 1999.

[77] O. Kubbar and H.T. Mouftah, "Multiple access control protocols for wireless ATM : problems definition and design objectives," *IEEE Comm. Mag.,* pp. 96-99, November 1997.

[78] M.K. Simon and M.S. Alouini,"A unified approach to the performance analysis of digital communication over generalized fading channels," *Proc. of the IEEE*, vol. 86, no. 9, pp. 1860-1877, September 1998.

[79] S. Lin and D.J. Costello, *Error control coding.* Prentice Hall, Englewood Cliffs, NJ, 1983.

[80] L. Tolhuizen, S. Baggen and E. Hekstra-Nowacka, "Union bounds on the performance of product codes," *Proc ISIT 1998*, Cambridge, MA, USA, August 1998.

[81] L. Tolhuizen, C. Baggen, "On the weight enumerator of product codes," *Discr. Math.*, 106/107, pp. 483-488, 1992.

[82] ETSI TS 101 475, "Broadband Radio Access Networks (BRAN) ; Hiperlan Type 2 ; Physical (PHY) layer" , Sophia-Antipolis, France, 2000.

Résumé :

Récemment, il a été démontré que les systèmes de décodage itératif peuvent fonctionner à des rendements très proches de la limite de Shannon avec toutefois une complexité raisonnable. En particulier, les codes LDPC (" low-density parity-check ") irréguliers et les turbo codes sont des candidats prometteurs pour de futures applications. En premier lieu, la dynamique non-linéaire du décodage itératif est étudiée de manière expérimentale. Celle-ci étant très complexe, un modèle simplifié basé sur les densités de probabilité gaussiennes est proposé pour analyser le décodage itératif. En particulier, nous verrons comment expliquer le seuil de bruit et comment caractériser la dynamique en fonction des paramètres du code. Ensuite, grâce à une analyse théorique, un lien est établi entre les propriétés des codes LDPC et la dynamique du décodage itératif associé. Finalement, les performances des systèmes de décodage itératif sont évaluées pour les applications VDSL et ATM sans fil.