Dualité entre codage de source et codage de canal

François-Xavier Bergot

Thèse

présentée pour obtenir le grade de docteur

de l'Ecole nationale supérieure
des télécommunications

Spécialité : Electronique et Communications

François-Xavier Bergot

Dualité entre codage de source et codage de canal

soutenue le 16 juin 2000 devant le jury composé de

Michel Barlaud	Président et rapporteur
Benoît Macq	Rapporteur
Jean-Claude Belfiore	Examinateurs
Pierre Duhamel	
Pierre Siohan	
Olivier Rioul	Directeur de thèse

Ecole nationale supérieure des télécommunications

Car avec beaucoup de sagesse on a beaucoup de chagrin, et celui qui augmente sa science augmente sa douleur.

Ecclésiaste, 1, 18 (trad. Louis Segond)

Ne passons pas à côté des choses simples.

Herta

Remerciements

Je tiens tout d'abord à remercier M. Benoît Macq de l'Université Catholique de Louvain et M. Michel Barlaud de l'Université de Nice qui ont tous les deux contribué à améliorer ce rapport, tant au niveau de la forme que du fond. Je remercie également M. Pierre Siohan, du centre de Recherche et Développement de France Télécom à Rennes, dont l'intérêt pour le travail de Bahram Zahir-Azami a suscité quelques-uns des travaux présentés dans cette thèse. Je n'oublie pas M. Pierre Duhamel, chef du département TSI, que je remercie non seulement pour avoir accepté le rôle d'examinateur mais aussi pour les conseils qu'il a pu me donner avant que cette thèse ne commence, et M. Jean-Claude Belfiore du département COMÉLEC qui a également inspiré quelques-uns des travaux présentés dans cette thèse.

Mes plus grands remerciements sont bien évidemment adressés à Olivier Rioul qui a dirigé mes travaux. La clarté de ses cours, ses idées et sa persévérance m'ont beaucoup aidé : j'espère que ses qualités m'auront imprégné et qu'il gardera un bon souvenir de ce travail accompli ensemble.

Je remercie également M. Bernard Robinet, directeur scientifique de l'ENST, qui a accepté de financer ce travail, ce qui est bien sûr une des conditions nécessaires à son accomplissement.

Bien qu'absent du jury, M. Solé, de l'Université de Nice, a également contribué à l'évaluation de ce mémoire, spécialement grâce à ses connaissances des codes réels : je le remercie pour le temps qu'il y a consacré et pour son aide sur le rayon de recouvrement.

Cette thèse a prolongé une scolarité commencée à l'ENST en 1993 en tant qu'élève-ingénieur. Je serais donc ingrat si j'oubliais de remercier pour leur enseignement les enseignants-chercheurs des différents départements, notamment ceux du groupe de communications numériques du département COMÉLEC et ceux du département TSI : leur enseignement passionné m'a beaucoup apporté. Distinguons parmi eux, en plus de ceux cités précédemment, Joseph Boutros, Éric Moulines, Philippe Loubaton, Jacques Prado, Maurice Charbit, Dominique Ventre, Jean Provost et Yves Mathieu. Je tiens également à remercier Sylvie Mayrargue du centre de Recherche et Développement de

France Télécom alors appelé CNET : elle m'a permis de découvrir à travers deux stages ce centre de recherche.

La communauté de situation crée toujours des liens : je remercie donc tous les « thésards » avec qui j'ai eu l'occasion d'échanger quelques mots, voire des idées, pire des blagues, pendant ces années. La diversité culturelle de cette communauté bien réelle fut un véritable enrichissement humain. Je souhaite à ceux et celles qui ne sont pas encore « au bout du tunnel » d'y parvenir brillament. Je remercie également mes amis qui, pendant que je demeurais étudiant, ont choisi d'entrer dans la vie où, paraît-il, on travaille vraiment : dans quelques mois je serai certainement l'un des vôtres.

Je remercie le personnel du Ministère de la Défense qui m'a confié une tâche intéressante pendant mon service national tout en me laissant rédiger cette thèse. Une pensée également pour les scientifiques et musiciens de la troisième section de la 11^{e} compagnie du 8^{e} régiment de transmission du contingent 9912, apprentis petits hommes verts qui ont su instaurer une bonne camaraderie pendant qu'ils faisaient des « classes » (mais où donc étaient les profs ?).

Il n'y a pas de mots pour dire tout ce qu'ont su m'apporter mes parents, mon frère, son épouse Caroline et mon filleul Louis. Puissé-je en profiter longtemps encore !

Enfin, j'en appelle à Guillaume de Machaut pour remercier Catherine de son soutien et de sa présence :

« Ce qui soustient moy, m'onneur et ma vie
Aveuc Amours, c'estes vous, douce dame.

Long, près, toudis serez, quoy que nuls die,
Ce qui soustient moy m'onneur et ma vie.

Et quant je vif par vous anemie,
Qu'aime mieux que moy, bien dire doy, par m'ame :
"Ce qui soustient moy, m'onneur et ma vie
Aveuc Amours, c'estes vous, douce dame. »

Table des matières

Table des figures

Liste des tableaux

Abréviations et notations

Abréviations

Nous donnons ici la liste des abréviations les plus utilisées dans ce document.

AWGN (channel)	*Additive White Gaussian Noise* ((canal à) bruit blanc gaussien additif).
(code) BCH	code de Bose-Chaudhuri-Hocquenghem.
CBS	canal binaire symétrique.
CCE	codes correcteurs d'erreurs.
CqS	canal q-aire symétrique.
EQM	erreur quadratique moyenne.
i.i.d.	indépendants et identiquement distribués.
(code) MDS	*Maximum Distance Separable* (code séparable et à distance (minimale) maximale).
OPTA	*Optimum Performance Theoretically Attainable* (performances optimales théoriquement possibles).
SAS	source arithmétique symétrique.
SBA	source binaire asymétrique.
SBS	source binaire symétrique.
SqS	source q-aire symétrique.

Notations

Nous récapitulons ici les notations les plus courantes employées tout au long des cinq chapitres de ce document.

$\lvert A\rvert$	cardinal de l'ensemble A ou module du nombre complexe A.
$\lfloor x \rfloor$	partie entière du réel x.
$\lceil x \rceil$	plus petit entier supérieur à x.
A^t	transposée de A.
A^*	conjugué de A.
$A^\dagger$	transconjugué de A.
AN	code arithmétique constitué des B premiers multiples de A.
B	cardinal d'un code arithmétique.

$\mathcal{C}$	code.
$\mathcal{C}^{\perp}$	code dual du code linéaire $\mathcal{C}$.
$\mathbb{C}$	corps des complexes.
CT_i	classe cyclotomique de l'entier i (pour n donné).
$\underline{c}$	mot de code.
$\underline{c}_m$	mot de code modulé.
$\underline{\hat{c}}$	mot de code estimé.
$\underline{\hat{c}}_m$	mot de code estimé modulé.
$d_A()$	distance arithmétique.
$d_H()$	distance de Hamming.
$d_H^{i\ldots j}(\underline{u},\underline{v})$	distance de Hamming calculée uniquement sur les composantes $i, i+1, \cdots, j-1, j$ de $\underline{u}$ et $\underline{v}$.
D	distorsion (suivant un certain critère de fidélité).
D_{inf}	borne inférieure de la distorsion (pour une SBS comprimée suivant le critère de la distance de Hamming).
D_{sci}	borne inférieure de la distorsion (pour une SBS comprimée suivant le critère de la distance de Hamming et transmise sur le CBS).
F_q	corps fini à q éléments.
F_q^m	espace vectoriel de dimension m construit sur le corps F_q.
G	matrice génératrice d'un code linéaire.
g	polynôme générateur d'un code cyclique.
H	matrice de parité d'un code linéaire.
H_2	entropie binaire.
h	polynôme de parité d'un code cyclique.
k	dimension d'un code linéaire.
$\underline{\imath}$	mot d'information.
$K[x]$	anneau des polynômes sur le corps K ($K = \mathbb{C}, \mathbb{R}, F_q$ dans cette thèse).
$K[x]/(x^n-1)$	anneau quotient de $K[x]$ par le polynôme x^n-1, i.e. ensemble des restes de la division euclidienne des polynômes de $K[x]$ par x^n-1.
L	matrice d'étiquetage.
$\log_q$	logarithme à base q.
m	longueur d'un code.
M	taille d'un code.
$M^{(i)}$	polynôme minimal de l'élément α^i.
M_j	matrice formée à partir de $2j-1$ syndromes distincts.
$\mathbb{N}$	ensembles des entiers naturels.
n	longueur d'un code.
$o(f)$	fonction négligeable devant la fonction f en un point donné (notation de Landau).
$O(f)$	fonction dominée par f en un point donné (notation de Landau).
p	paramètre du CqS qui définit aussi $P=(q-1)p$.
p_s	probabilité qu'une source binaire émette un « 1 ».
P_{em}	probabilité d'erreur par mot.
P_{emi}	borne inférieure de la probabilité d'erreur par mot.
P_{es}	probabilité d'erreur par symbole.

P_{esi}	borne inférieure de la probabilité d'erreur par symbole.
$Pr(A)$	probabilité de l'événement A.
q	puissance d'un nombre premier.
Q	fonction d'erreur normalisée $Q(x) = 1/\sqrt{2\pi}\int_x^\infty \exp(-t^2/2)dt$.
$\mathbb{R}$	corps des réels.
R	rendement (taux) d'un code.
R_c	rendement d'un code de canal.
R_s	rendement d'un code de source.
$R_{\text{source,critère}}(D)$	fonction rendement-distorsion correspondant à la source indiquée pour un critère donné (si le critère n'est pas précisé, le critère choisi est la distance de Hamming).
S	syndrome d'un mot
$S(x)$	polynôme formé des syndromes.
T_y	tangente en y à la fonction H_2.
TF_n	matrice correspondant à la transformée de Fourier discrète de dimension n sur le corps des complexes.
$V(\underline{c})$	cellule de Voronoï du mot de code $\underline{c}$.
$x[y]$	x modulo y (i.e. le reste de la division euclidienne de x par y).
$\mathbb{Z}$	anneau des entiers relatifs.
$\mathbb{Z}_m$	$\mathbb{Z}/m\mathbb{Z}$ (anneau quotient de l'anneau des entiers relatifs par le sous-anneau des multiples de m).
α	racine n^{e} de l'unité.
α_i	nombre de mots de poids i dans la cellule $V(\underline{0})$.
Λ	polynôme localisateur d'erreurs.
Ω	polynôme évaluateur d'erreurs.

Résumé

Codage de source et codage de canal sont deux activités duales : l'une supprime de la redondance tandis que l'autre en ajoute. Cette dualité est donc une clé pour simplifier l'étude complexe du codage conjoint source-canal, étude rendue nécessaire pour améliorer les performances des sytèmes de transmission dont la complexité est toujours limitée.

Afin de mener à bien cette conception, nous étudions dans cette thèse différentes briques de base qui impliquent de faibles délais de traitement, s'appuient sur des modèles simples de source, de canal et de critère de performances et exploitent des outils de codage de canal pour le codage de source grâce à la dualité. Les briques de bases étudiées sont les suivantes :

- source binaire symétrique ou asymétrique, canal binaire symétrique et distance de Hamming. Utilisant les codes linéaires ou les codes arithmétiques, nous comparons le système conçu de manière séparée (par opposition à conjoint) à un système plus performant dit « source ou canal ». Nous déterminons également des bornes de performances du système source ou canal dépendant de la complexité du codeur considéré. Ce système permet un codage conjoint efficace ;
- nous exploitons la dualité pour comprimer la source gaussienne selon le critère de la distance de Hamming (mesurant un bruit impulsif) grâce aux codes BCH réels MDS. Cette compression minimise également la puissance du bruit impulsif ;
- lorsque la source gaussienne doit être transmise sur un canal binaire symétrique suivant le critère de l'erreur quadratique moyenne, nous étudions l'optimisation du quantificateur à code de canal fixé en imposant une relation linéaire entre les dictionnaires de source et de canal. Pour des systèmes de faible complexité, les meilleurs systèmes obtenus sont ceux pour lesquels il n'y a pas de codage de canal !

Ces briques de base peuvent être utilisées par exemple dans des systèmes à protection hiérarchique de données.

Mots-clés : codage de source, codage de canal, codage conjoint , théorie de l'information, algorithme de Lloyd, code linéaire binaire, code arithmétique, code BCH réel MDS, étiquetage linéaire, source binaire, source gaussienne, canal binaire symétrique, distance de Hamming, distance euclidienne.

Abstract

Source coding and channel coding are dual tasks: the former removes redundancy from source while the latter adds some. This duality is a tool to simplify the hard and painful study of joint coding, which is mandatory to improve performances of transmission systems with a bounded complexity.

In order to design a joint coding system, we study several simple typical situations, using channel coding tools to make source coding (by duality) and simple source and channel models and simple criteria. We limit system complexity to achieve low processing delays. The typical simple situations studied are:

- binary symmetric or asymmetric source transmitted on a binary symmetric channel with the Hamming distance fidelity criterion. We use linear and arithmetic codes to compare the performances of a tandem system (in contrast with a joint one) to those better of a system called "source or channel". We give performances lower bound with a given system complexity. The system "source or channel" achieves an efficient joint coding;
- by duality, a gaussian source is compressed by using real BCH MDS codes with Hamming distance fidelity criterion which is related to impulsive noise. By the way, we minimize also impulsive noise power;
- given a channel code, we optimize the quantizer of a gaussian source transmitted on a binary symmetric channel with a mean squared error fidelity criterion. In order to optimize the quantizer, we set a linear relationship between source code and channel code. The better low complexity systems are those where the channel code is identity!

The developed tools in these typical situations may be used e.g.in a full transmission system with unequal error protection.

Keywords: source coding, channel coding, joint coding, information theory, Lloyd's algorithm, linear code, arithmetic code, real BCH MDS code, linear mapping, binary source, gaussian source, binary symmetric channel, Hamming distance, euclidean distance.

Introduction

Pourquoi choisir une conception conjointe au lieu d'une conception séparée ?

Depuis quelques années, nous assistons à un regain d'intérêt pour l'étude des techniques de codage conjoint source - canal. Le « théorème du codage source/canal », dû à Shannon [55] et bien connu en théorie de l'information, donne les performances optimales que l'on est en droit d'attendre d'un système où source et canal suivent un modèle déterminé : ces performances sont appelées OPTA (*Optimum Performances Theoretically Attainable*). La preuve montre qu'un système séparé permet d'atteindre asymptotiquement ces performances : le codeur de source détermine alors en grande partie les performances du système alors que le code de canal est suffisamment puissant et suffisamment long pour combattre presque toutes les perturbations introduites par le canal, assurant ainsi l'intégrité des données transmises au décodeur de source. En raison de cette preuve, le théorème du codage conjoint est aussi appelé théorème de la séparation : en choisissant, d'une part, un bon codeur de source optimisé indépendamment du canal et, d'autre part, un bon codeur de canal optimisé indépendamment de la source, nous sommes assurés d'être proches des performances optimales.

Cependant, les systèmes de transmission existants diffèrent en plusieurs points du modèle de ce théorème. Tout d'abord, il est évident que le concepteur d'un système de transmission ne peut s'autoriser des codeurs de source et de canal de grande longueur, comme préconisé par le théorème de Shannon : dans le cas contraire, le délai de traitement des données serait alors assez long, l'émetteur ou le récepteur pourrait ne pas répondre à d'autres spécifications telles que poids, taille, consommation éléctrique.... Autant de contraintes supplémentaires, parfois aussi importantes que les performances du système, qui nous obligent à tenir compte de la complexité de ce dernier.

Par ailleurs, depuis quelques dizaines d'années, le développement foudroyant des services nécessite de plus en plus de transmettre sur un seul et même réseau des sources hétérogènes : images, écrits, messages et données de toutes sortes circulent sur l'internet, voix et données sont transmises sur le réseau téléphonique (fixe ou mobile). De surcroît, le canal physique de transmission est souvent variable : la conception d'un système doit alors prendre en compte cette variation afin de garantir une qualité de service minimale. Il est donc nécessaire de concevoir un système qui permet de transmettre

des sources de natures différentes sur des canaux variables, alors que le théorème de Shannon ne s'intéresse qu'à un modèle de source et de canal bien spécifique.

Une approche conjointe du système de transmission paraît donc toute indiquée lorsque la complexité de ce système est limitée, garantissant ainsi un court délai de traitement et pouvant réduire cette complexité (cette réduction est observée en considérant certaines sources et certains canaux lorsque le rendement est égal à l'unité). De plus, son application à des modèles génériques de source et de canal permet d'assurer la transparence des données par rapport aux réseaux utilisés pour les transmettre : le codage conjoint est donc une solution envisageable pour améliorer les transmissions, par exemple sur canal radio-mobile (afin de proposer de nouveaux services, comme le téléreportage), ou par satellite (avec des applications comme la télésurveillance ou le télécinéma).

Jusqu'à présent, une telle approche a rarement été mise en œuvre : la séparation du codage de source et du codage de canal facilite grandement leur étude respective. Dès lors que le modèle étudié prend en compte ces deux domaines, l'étude devient complexe et pénible. Pourtant, les deux codeurs semblent poursuivre des buts contradictoires : le codeur de source supprime toute ou partie de la redondance de la source alors que le codeur de canal en ajoute, le plus souvent de manière contrôlée. Codage de source et codage de canal sont donc des activités duales, dualité dont on pourrait tirer parti pour simplifier l'étude du codage conjoint ou pour transposer des outils d'un domaine à l'autre.

La dualité du codage de source et du codage de canal : une idée récurrente mais relativement peu explorée

Le codage conjoint, bien qu'absent des applications actuelles, apparaît de manière récurrente dans les articles de recherche en faible nombre devant la pléthore d'articles traitant des deux problèmes séparément. Depuis les débuts de la théorie de l'information, des chercheurs se sont intéressés à l'impact des perturbations introduites par le canal sur le codage de source. Plusieurs solutions ont été proposées et peuvent être classées selon deux thèmes :

- la conception de codeurs de source robustes aux erreurs dues au canal. La robustesse découle soit de l'optimisation de la correspondance entre dictionnaire de source et code de canal, soit de l'optimisation du codeur de source à modèle de canal donné ;
- la protection hiérarchique des données à émettre.

L'optimisation de la correspondance entre dictionnaire de source et code de canal consiste à associer à des éléments proches du dictionnaire de source des mots de canal proches afin que les erreurs dues au canal les plus probables créent une distorsion la plus faible possible. Cette optimisation est complexe puisqu'elle en prend en compte la quasi-totalité de la chaîne de transmission. Skinnemoen [58] a mené une telle optimisation grâce à un réseau de neurones, développant ainsi un quantificateur dont la sortie

est directement le mot de canal modulé, sans passer par une étape d'étiquetage binaire. Cette technique, appelée MORVQ (*Modulation Organized Vector Quantization*), généralise une autre technique, bien connue des ingénieurs des réseaux téléphoniques, les PCM (*Pulse Code Modulation*). La première étape d'un codeur PCM consiste à effectuer un échantillonnage du signal, obtenant ainsi une modulation d'amplitude, modulation courante en codage de canal. Vaishampayan et Farvardin [62] ont également proposé une conception conjointe de la modulation et du dictionnaire de source en imposant un décodeur linéaire ainsi que des contraintes portant sur l'énergie et la bande passante.

Lorsque le système ne comprend pas, à proprement dit, de code correcteur d'erreur, cette approche conduit à optimiser l'étiquetage inclus dans le codeur de source, la quantification étant indépendante du canal. Récemment, Knagenhjelm [29, 30] a montré l'utilité de la transformée de Hadamard, outil bien connu dans le domaine des codes correcteurs d'erreurs (CCE) construits sur les corps finis, pour apprécier les performances de l'étiquetage.

Il est également possible d'optimiser à la fois l'étiquetage et le quantificateur. En 1964, Fine [21] a étudié l'influence des canaux sur la quantification, qu'elle soit synchrone ou prédictive, et donné des conditions d'optimalité. Plus récemment, Farvardin [17, 18, 19] a utilisé un algorithme de Lloyd prenant en compte les perturbations introduites par le canal pour construire des codeurs de source, scalaires ou vectoriels, optimisés en fonction du canal considéré. Une telle technique est appelée COSQ ou COVQ (respectivement *Channel Optimized Scalar Quantization* et *Channel Optimized Vector Quantization*).

La protection hiérarchique consiste à rendre moins sensibles au bruit les informations les plus importantes en leur attribuant plus d'énergie ou en choisissant un code de canal plus puissant que celui protégeant les informations moins importantes. Cette idée est duale de celle qui mène aux travaux sur les modulations codées où tous les bits n'ont pas la même influence sur la probabilité d'erreur et sont donc plus ou moins protégés. Bedrosian [3], en 1958, appliquait cette idée aux PCM en proposant de les pondérer afin de mieux protéger les bits de poids fort. Récemment, Zahir-Azami [69] a appliqué ce type de protection à la décomposition binaire d'une source.

Certaines techniques de codage conjoint ont donc mis à profit la dualité entre codage de source et codage de canal. Certains articles, en faible nombre, n'ont exploité que cette dualité pour proposer de nouvelles solutions : Ancheta [1] se sert des codes correcteurs d'erreurs binaires pour comprimer une source binaire asymétrique. Goblick [24], Berger [4], Van der Horst [63], Kerdock et Wolf [27] ont employé les mêmes codes pour comprimer la source binaire symétrique. Cette dualité s'illustre aussi par les constellations issues des réseaux de points : elles sont utilisées à la fois dans le codage de canal [31] et en codage de source [44, 62].

Le regain d'intérêt pour le codage conjoint nous invite, dans cette thèse, à poursuivre l'étude de cette dualité afin, d'une part, de transposer des outils du codage de canal au codage de source et, d'autre part, d'en appliquer les résultats dans le domaine du codage conjoint en considérant des systèmes de faible complexité.

Choix du modèle

Afin d'évaluer les performances du codage conjoint, il est nécessaire de définir une mesure de l'erreur commise entre la source originale et la source reconstruite que le destinataire de la communication reçoit. La mesure d'erreur pertinente est le plus souvent subjective lorsque le destinataire est un être humain aux capacités de perception naturellement limitées et que la source à transmettre est un signal audio et/ou vidéo. Du fait de la prise en compte de l'élément humain, ces mesures d'erreurs ne sont pas simples à traiter. Dans les autres cas, la mesure d'erreur pertinente est objective (indépendante de la nature du destinataire) et rigoureusement définie. Trois **distances** peuvent servir de mesure d'erreur objective :

- la distance euclidienne, plus couramment appelée erreur quadratique moyenne (EQM) ;
- la distance de Hamming, utilisée le plus souvent en codage de canal pour définir les probabilités d'erreur ;
- la distance arithmétique, définie entre deux entiers et peu employée jusqu'à présent.

Il est également nécessaire de choisir **un modèle de canal**. Les trois modèles les plus étudiés actuellement en codage conjoint demeurent assez simples :

- le canal binaire symétrique (noté CBS), cas particulier d'un canal discret ;
- le canal à bruit blanc gaussien additif (AWGN, *Additive White Gaussian Noise (Channel)*) ;
- le canal de Rayleigh.

Ces trois modèles ne créent pas d'interférence entre symbole, pourtant l'étude d'un système conjoint faisant intervenir un de ces modèles devient vite complexe.

Enfin un **modèle de source** doit être retenu afin de mener l'étude. Là encore, plusieurs choix simples [1] s'offrent à nous :

- une source gaussienne aux échantillons éventuellement corrélés (source de Gauss-Markov) ;
- la source gaussienne généralisée blanche (regroupant, entre autres, les sources gaussienne, laplacienne et uniforme), notée SGGB ;
- une source binaire symétrique (SBS) ou asymétrique (SBA) aux échantillons indépendants ;
- une source arithmétique symétrique (qui émet des entiers naturels inférieurs à un entier donné) notée SAS.

Nous pouvons alors, grâce au tableau 1, resituer les travaux déjà accomplis dans

[1] il existe par ailleurs des modèles très complexes.

leurs contextes [2].

Source considérée	Modèle de canal retenu	Critère retenu	Outil et référence
Gauss-Markov (parole)	AWGN	EQM	MORVQ, Skinnemoen [58]
Gauss-Markov	AWGN	EQM	Modulation adaptée à la source et au canal, Vaishampayan *et al.* [62]
gaussienne	AWGN, Rayleigh	EQM	Réseaux de points et rotations, Pépin [51]
sans mémoire	AWGN	EQM	Réseaux de points, Laroia *et al.* [32]
Gauss-Markov	discret	EQM	Réseaux de points, CCE et étiquetage affine, Méhes et Zeger [44]
gaussienne	discret	EQM	Fine [21]
continue	discret	EQM	PCM pondérée (protection inégale de l'information), Bedrosian [3]
Gauss-Markov	CBS	EQM	COVQ, COSQ, Farvardin *et al.* [17, 18, 19]
SGGB	CBS	EQM	optimisation conjointe par algorithme de Lloyd et protection hiérarchique, Zahir-Azami [69]
SBA	CBS	Hamming	CCE, Massey [37]
SAS	CBS	EQM	CCE, Crimmins *et al.* [14]
			CCE, Wolf et Redinbo [65]
codeur de source max-entropique	CBS	EQM	optimisation de l'étiquetage par transformée de Hadamard, Knagenhjelm [29, 30]
			optimisation de l'étiquetage, McLaughlin [42]

TAB. 1 – Quelques travaux déjà accomplis en codage conjoint

À l'exception des systèmes utilisant des modulations adaptées à la fois au canal et à la source, les systèmes conjoints énumérés dans ce tableau peuvent être décrits par le schéma 1, les divers travaux jouant sur la nature des outils utilisés pour accomplir telle ou telle tâche et sur la manière de les optimiser conjointement. Sur ce schéma, le codeur de source produit une étiquette qui est protégée par le codeur de canal. La sortie du codeur de source est donc assimilable à une source binaire. Modulation, canal et démodulation consituent un supercanal qui est, dans certains cas, modélisable par un canal binaire symétrique. Nous nous focaliserons donc au début de cette thèse sur le système source binaire - canal binaire symétrique, les performances étant mesurées

[2] Remarquons que ce modèle de source, c'est-à-dire un codeur où les étiquettes sont équiprobables est assimilable à une source arithmétique symétrique, la seule différence résidant dans l'éventuelle sémantique à accorder à une telle source : la présence d'un codeur max-entropique signifie que le destinataire devra accomplir la fonction inverse pour retrouver l'information pertinente.

grâce à la distance de Hamming. Ce choix permet de prendre en compte l'hétérogénéité des sources à transmettre sur un seul et même réseau et de garantir la transparence des données vis-à-vis du réseau. Un tel modèle est donc une brique de base d'un système plus complet : il a l'avantage d'être simple et précis. Zahir-Azami [69] a employé les mêmes outils que ceux qui seront décrits ici en reprenant le même modèle mais en considérant le critère de l'EQM. Accessoirement, nous examinerons brièvement le critère de distance arithmétique appliqué à la source arithmétique symétrique et l'EQM associée à la transmission d'une source gaussienne.

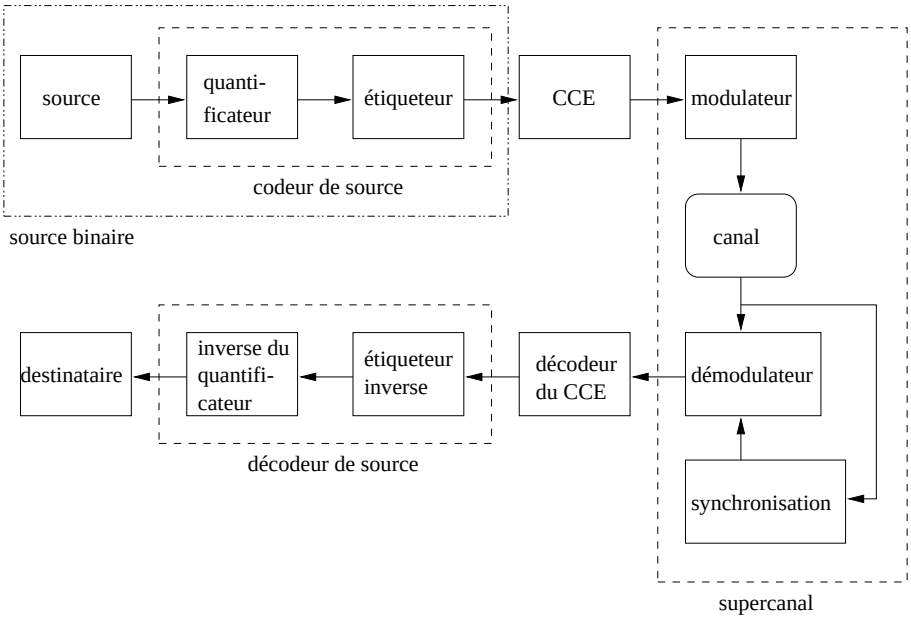


FIG. 1 – Système simple de transmission

Organisation du document

Le **premier chapitre** met en évidence la dualité entre codage de source binaire symétrique et codage de canal binaire symétrique. Les résultats se généralisent, le plus souvent facilement, au cas q-aire, qui sera par conséquent étudié. Les outils développés en codage de canal seront alors directement transposés au codage de source.

Le **second chapitre** exploite la dualité de ces deux domaines pour examiner le codage conjoint SBS-CBS : un système conjoint est comparé au système séparé. L'algorithme de Lloyd généralisé est présenté, permettant d'obtenir des codes de source ou des codes de canal.

Le **troisième chapitre** reprend la dualité exposée au premier chapitre pour l'appliquer à la source binaire asymétrique. Nous comparons alors nos résultats avec ceux d'Ancheta [1] et utilisons à nouveau l'algorithme de Lloyd pour trouver de bons codes de source non linéaires.

Le **quatrième chapitre** reprend l'étude menée lors des deux premiers chapitres

en substituant aux codes linéaires les codes arithmétiques. Dans un premier temps, la source considérée est arithmétique : par dualité, ces codes, habituellement utilisés en correction d'erreurs, peuvent comprimer cette source selon le critère de distance arithmétique. Dans un deuxième temps, nous revenons à la source binaire symétrique et au critère de la distance de Hamming en vue de comprimer cette source par le biais des codes arithmétiques. Le codage conjoint à partir de ces codes est ensuite examiné.

Le **dernier chapitre** exploite une nouvelle fois la dualité pour comprimer une source gaussienne réelle en utilisant des codes BCH réels selon la distance de Hamming : cette compression enlève du bruit impulsif à la source, le canal est supposé parfait. Les codes réels sont ensuite optimisés par un algorithme de Lloyd par rapport à la source réelle et au code de canal en considérant le critère de l'EQM, le canal considéré étant le canal binaire symétrique.

Enfin, nous ferons le bilan de nos travaux et évoquerons différentes perspectives de recherche future.

Contributions de cette thèse

L'exploitation de la dualité nous a permis de transposer les outils du codage de canal au codage de source, notamment :

- en codage de source q-aire symétrique (SqS), l'étude des cellules de Voronoï nous a permis de formuler une borne inférieure de la distorsion utile, car elle donne les performances optimales des codes de longueur et de dimension données (c'est-à-dire une à complexité limitée), l'OPTA ne donnant que les performances asymptotiques. Nous avons proposé un algorithme pour construire de bons codes comprimant la SqS et démontré que les codes linéaires pouvaient atteindre l'OPTA. La SBS a aussi été comprimée avec succès par des codes arithmétiques ;
- nous avons aussi étudié les performances des codes linéaires binaires en compression de source binaire asymétrique, mettant en évidence la complémentarité pour les petites longueurs du codage par syndrome d'Ancheta et du codage par région de Voronoï. L'algorithme de Lloyd nous permet de mesurer l'impact de la linéarité du code en nous permettant de trouver de bons codes non linéaires ;
- nous avons proposé un algorithme complet de codage de source utilisant les codes BCH réels. Cet algorithme permet d'enlever le bruit impulsif de la source par un « filtrage » non linéaire. Si la complexité le permet, cet algorithme peut supprimer le bruit impulsif de moindre énergie, opérant ainsi une compression à double critère.

Dans le domaine du codage conjoint CBS-SBS, nous avons mis en évidence l'intérêt d'un système conjoint face au sytème séparé lorsque la complexité est limitée : ses performances sont meilleures pour une complexité moindre. Les cellules de Voronoï nous ont permis de donner des bornes inférieures de performances utiles car dépendantes de la complexité. L'algorithme de Lloyd généralisé nous a permis d'améliorer légèrement les performances du système conjoint et permet d'obtenir facilement le décodeur optimal lorsque la sortie du canal est quantifiée sur plus de deux niveaux. Nous

avons aussi montré l'efficacité du système le plus simple dans le codage conjoint source gaussienne-CBS en considérant des systèmes optimisés par algorithme de Lloyd.

Ce travail comporte plusieurs avantages :
- le modèle le plus étudié, à savoir source binaire et canal binaire symétrique est assez générique pour représenter bon nombre de réseaux de communication et les données qui y sont transmises. Nous avons donc étudié une brique de base qui, par sa flexibilité, permet de considérer alors des systèmes plus complexes. En décomposant binairement une source et en utilisant une protection hiérarchique, une amélioration des performances est prévisible ;
- nous avons obtenu des bornes permettant d'apprécier les performances des codes trouvés en fonction de leur longueur et de leur dimension. Ces bornes donnent ainsi une idée des performances à complexité donnée, même si l'on ne peut garantir théoriquement qu'elles sont atteignables par un code de mêmes paramètres.

Les inconvénients sont au nombre de deux :
- le critère de la distance de Hamming, utilisé pour le modèle de source binaire, ne facilite pas l'étude du codage conjoint. Zahir-Azami [69] a effectué, parallèlement à cette thèse, un travail similaire en considérant le critère de l'EQM : l'algorithme de Lloyd permet alors d'améliorer nettement les codeurs et décodeurs, ce qui est difficile avec la distance de Hamming. Le critère de distance de Hamming ne facilite pas non plus la compression de source réelle, comme le montre l'algorithme que nous avons développé ;
- pour des raisons de complexité d'optimisation, les longueurs de code demeurent petites et nous sommes relativement éloignés de l'OPTA. Augmenter cette longueur sans pour autant considérer des complexités faramineuses permettrait sans doute de se rapprocher de performances optimales.

Insistons encore une fois sur la difficulté de mener l'étude, même dans des cas simples : le passage d'une source symétrique à une source asymétrique, par exemple, ne nous permet pas d'appliquer les mêmes outils pour trouver des bornes des performances.

Cependant, le but de cette thèse est atteint : nous fournissons plusieurs briques de base génériques utilisables par la suite dans des systèmes spécifiques. Ces briques de base s'appuient volontairement sur des modèles simples mais assez répandus.

Chapitre 1

Dualité entre le codage de canal q-aire symétrique et le codage de source q-aire symétrique

Nous souhaitons montrer dans ce chapitre la forte dualité qui existe entre le codage de canal et le codage de source, particulièrement quand nous nous intéressons au cas du canal q-aire symétrique (**CqS**) ou de la source q-aire symétrique (**SqS**). Nous commençons d'abord par quelques rappels et travaux sur le codage de canal : définition, performances exactes, bornes de ces performances, recherche de bons codes. Puis nous montrons la dualité entre les deux types de codage afin de pouvoir transposer par la suite les outils du codage de canal dans le domaine du codage de source.

1.1 Codage de canal

La locution codage de canal désigne l'ensemble des techniques visant à protéger l'information à émettre des perturbations introduites par le canal de transmission. La plupart de ces techniques reposent sur deux idées :

- l'ajout de redondance à l'information à transmettre par l'intermédaire d'un code ;
- le choix d'une modulation adaptée au canal, c'est-à-dire d'un alphabet de canal et un dispositif de mise en forme du signal.

Ces deux idées peuvent se combiner : il est alors nécessaire de choisir une bonne correspondance entre mots de code et symboles émis, ce qui conduit par exemple aux travaux sur les modulations codées en treillis. Le paradigme du codage de canal est représenté sur la figure 1.1. Les symboles composant les mots d'information sont supposés appartenir au corps fini de q éléments F_q (où q est une puissance d'un nombre premier). Le codeur de canal ajoute au mot d'information de la redondance : il transforme le mot d'information $\underline{\imath} = (\imath_1, \cdots, \imath_k)$ de longueur k en un mot de canal $\underline{v} = (v_1, \cdots, v_n)$

de longueur n, chacun des n symboles appartenant à l'alphabet de canal. Nous supposons dans ce chapitre que la correspondance entre $\underline{i}$ et $\underline{v}$ est bijective. Pour que de la redondance soit effectivement introduite, n doit être plus grand que k. Le codeur de canal met aussi en forme le signal qui est perturbé par le canal. Le décodeur de canal utilise alors la redondance introduite pour fournir au destinataire une estimée du mot d'information émis, $\hat{\underline{i}}$. Cette estimée peut se calculer directement à partir du mot $\underline{u} = (u_1, \cdots, u_n)$ ou bien en cherchant d'abord le mot de code $\hat{\underline{v}}$ le plus vraisemblable sachant que le décodeur a reçu $\underline{u}$ pour en déduire $\hat{\underline{i}}$.

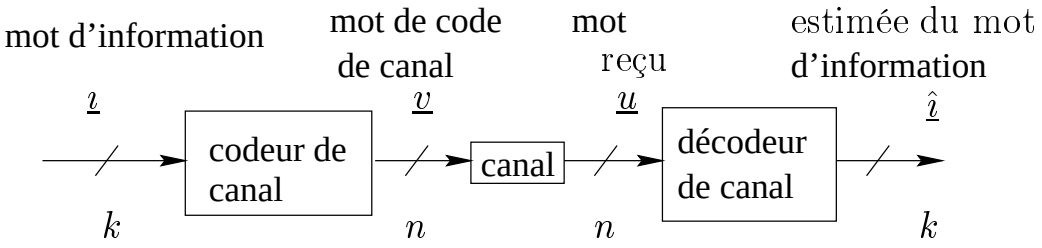


Fig. 1.1 – Paradigme du codage de canal.

Deux grandes familles de codes de canal existent : les codes en bloc travaillant sur des blocs de données de taille fixe et les codes convolutifs qui opèrent sur une fenêtre temporelle glissante. Les codes convolutifs sont souvent employés car leurs performances sont bonnes. Néanmoins, le décodage des codes en bloc linéaires est plus simple et à faible délai. Nous considérerons dans cette thèse uniquement des codes en bloc, le plus souvent linéaires, en raison de leur simplicité de description.

1.1.1 Le canal q-aire symétrique

Le canal représenté sur la figure 1.1 est totalement décrit par la densité de probabilité conditionnelle $p(\underline{u}|\underline{v})$, c'est-à-dire la probabilité que le mot reçu soit $\underline{u}$ sachant que l'on émet le mot $\underline{v}$. Nous nous limitons dans ce chapitre au canal q-aire symétrique sans mémoire, noté CqS, de paramètre p, représenté sur la figure 1.2. Un tel canal peut être représenté par une matrice de transition de taille $q \times q$:

$$\begin{pmatrix} 1-(q-1)p & p & \cdots & p & p \\ p & 1-(q-1)p & \cdots & p & p \\ \vdots & & & & \vdots \\ p & p & \cdots & 1-(q-1)p & p \\ p & p & \cdots & p & 1-(q-1)p \end{pmatrix} \tag{1.1}$$

Si le canal est utilisé n fois, alors

$$p(\underline{u}|\underline{v}) = \prod_{i=1}^{n} p(u_i|v_i)$$

$$p(u_i|v_i) = \begin{cases} p & \text{si} \quad u_i \neq v_i \\ 1-(q-1)p & \text{si} \quad u_i = v_i \end{cases}$$

Dans le cas binaire ($q = 2$), p est aussi appelé **probabilité de transition** ou **probabilité d'erreur du CBS**. Définissons P par $P = (q-1)p$. La probabilité conditionnelle $p(\underline{u}|\underline{v})$ peut s'exprimer à l'aide de la **distance de Hamming** $d_H(\underline{u}, \underline{v})$, i.e. le nombre de composantes non nulles de $\underline{u} - \underline{v}$:

$$p(\underline{u}|\underline{v}) = p^{d_H(\underline{u},\underline{v})}(1-P)^{n-d_H(\underline{u},\underline{v})}$$

Par construction, le CqS est un canal additif : il ajoute un mot d'erreur $\underline{e}$ au mot émis $\underline{v}$ de sorte que le mot reçu s'écrive $\underline{u} = \underline{v} + \underline{e}$, l'addition étant calculée dans F_q^n.

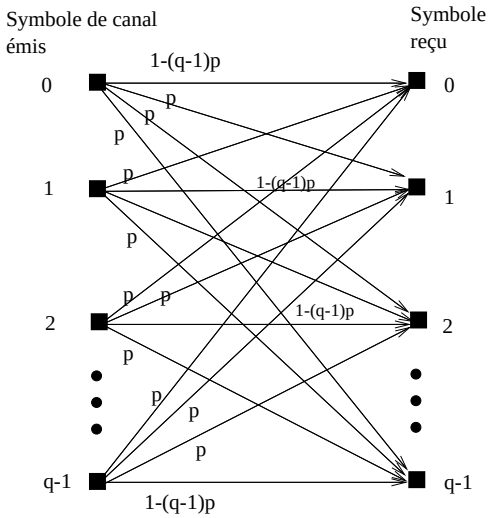


FIG. 1.2 – Canal q-aire symétrique : de chaque symbole émis partent q branches et q branches arrivent à un symbole reçu.

Ce modèle un peu abstrait mais générique et simple représente (approximativement, lorsque q est strictement supérieur à 2) un canal à bruit additif blanc et gaussien suivi d'un organe de décision dure.

1.1.2 Taux de codage canal et probabilités d'erreur

Le **taux de codage canal** aussi appelé **rendement canal** et noté $\mathbf{R_c}$ est défini par le rapport suivant :

$$R_c = \frac{k}{n}\log_2 q \text{ bits d'information par symbole de canal émis,} \tag{1.2}$$

où $\log_2$ désigne la fonction logarithme à base 2.

La qualité de la transmission se mesure en terme de probabilité d'erreur. On suppose que les mots d'information sont tous équiprobables. La probabilité d'erreur la plus naturelle est la **probabilité d'erreur par symbole**, notée P_{es}, c'est-à-dire la

moyenne des k probabilités qu'un symbole d'information estimé soit différent du symbole d'information émis

$$P_{es} = \frac{1}{k} \sum_{j=1}^{k} \Pr(\hat{i}_j \neq i_j) \tag{1.3}$$

où $\underline{i}$ et $\underline{\hat{i}}$ sont les mots d'information définis sur la figure 1.1. Cette probabilité est en général difficile à estimer théoriquement car elle requiert l'évaluation de k probabilités [1]. Par conséquent, on simplifie le critère mesurant la qualité de transmission en comptant le nombre de fois où le mot de canal estimé $\underline{\hat{v}}$ par le décodeur de canal est différent du mot de canal émis $\underline{v}$, définissant ainsi la **probabilité d'erreur par mot** notée P_{em}. Lorsque le décodeur de canal se trompe sur le mot de canal, cela signifie qu'au moins un symbole d'information estimé est faux et qu'au plus k symboles estimés seront erronés, i.e.

$$\frac{P_{em}}{k} \leq P_{es} \leq P_{em}. \tag{1.4}$$

Cet encadrement permet donc d'estimer plus simplement la qualité du codage effectué. Les performances des codes en bloc sur le CqS doivent être comparées avec les meilleures que l'on puisse obtenir, abstraction faite de la longueur des codes employés. Ces dernières sont formulées dans un théorème étonnament simple au regard de la difficulté d'évaluer les performances exactes d'un code en bloc, comme nous allons le voir.

1.1.3 Performances optimales : capacité du canal

Le théorème de codage de canal de Shannon [41] indique qu'une communication sur un canal peut être aussi fiable que possible (i.e. avec une probabilité d'erreur par symbole ou par mot aussi faible que voulue) à condition que le rendement de canal R_c soit inférieur ou égal à la **capacité** C du canal. La réciproque de ce théorème affirme que si le rendement canal est supérieur à la capacité alors la probabilité d'erreur [2] est minorée par un terme non-nul. La capacité est définie comme le maximum de l'information mutuelle entre l'entrée et la sortie du canal [3] :

$$C = \sup_{n} \max_{p(\underline{v})} \frac{1}{n} I(\underline{U}, \underline{V}) \tag{1.5}$$

$$I(\underline{U}, \underline{V}) = \sum_{\underline{u},\underline{v}} p(\underline{u}, \underline{v}) \log_2 \left(\frac{p(\underline{u}|\underline{v})}{p(\underline{u})} \right) \tag{1.6}$$

$I(\underline{U}, \underline{V})$ est appelée **information mutuelle** entre les deux variables aléatoires $\underline{U}$ et $\underline{V}$ de dimension n ayant respectivement pour réalisation $\underline{u}$ et $\underline{v}$. Introduisons l'entropie

[1] sauf, bien sûr, pour les codes qui assurent une protection égale de tous les symboles d'information !

[2] par mot ou par symbole, peu importe

[3] cette définition ne fait volontairement pas apparaître la contrainte portant sur le coût d'utilisation du canal car cette notion n'a pas de signification pratique pour le canal q-aire symétrique.

de $\underline{U}$, notée $H_2(\underline{U})$, et l'entropie conditionnelle de $\underline{U}$ sachant $\underline{V}$, notée $H_2(\underline{U}|\underline{V})$:

$$H_2(\underline{U}) = -\sum_{\underline{u}} p(\underline{u}) \log_2 p(\underline{u}) \tag{1.7}$$

$$H_2(\underline{U}|\underline{V}) = -\sum_{\underline{u},\underline{v}} p(\underline{u},\underline{v}) \log_2 p(\underline{u}|\underline{v}) \tag{1.8}$$

Entropie et information mutuelle vérifient alors

$$I(\underline{U},\underline{V}) = H_2(\underline{U}) - H_2(\underline{U}|\underline{V}) \tag{1.9}$$

Avant de donner une expression concise de la capacité du CqS, nous définissons l'**entropie binaire** H_2 par

$$\begin{array}{rccl} H_2 : & [0,1] & \to & [0,1] \\ & x & \mapsto & -x\log_2 x - (1-x)\log_2(1-x). \end{array}$$

La capacité du canal q-aire symétrique C_{CqS} s'écrit alors [6]

$$C_{CqS}(P) = \log_2(q) - H_2(P) - P\log_2(q-1), \quad 0 \le P \le 1 - \frac{1}{q}. \tag{1.10}$$

Dans le cas binaire, l'expression de la capacité devient

$$C_{CBS}(p) = 1 - H_2(p), \quad 0 \le p \le 1. \tag{1.11}$$

Dans ces deux cas, la capacité est obtenue pour $n = 1$ et en choisissant une variable aléatoire $\underline{V}$ de distribution uniforme. Elle est représentée pour différentes valeurs de q sur la figure 1.3.

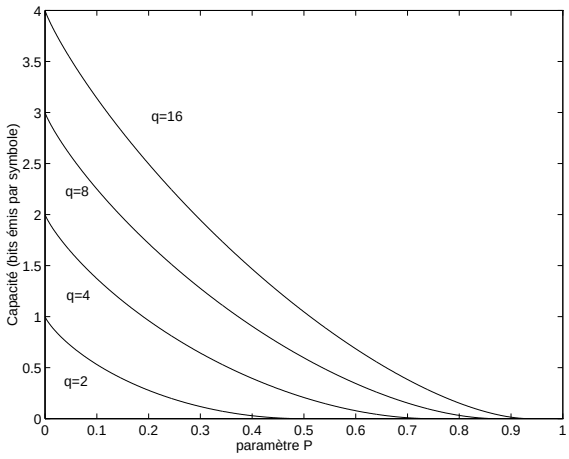


FIG. 1.3 – Capacité du CqS pour différentes valeurs de q.

Notons que la capacité vérifie une égalité qui sera utile par la suite :

$$2^{n(R_c - C_{CqS}(P))} = \frac{(q-1)^{nP} 2^{nH_2(P)}}{q^{n-k}} = \frac{(q-1)^{nP}}{P^{nP}(1-P)^{n(1-P)} q^{n-k}}, \quad 0 \le P \le 1 - \frac{1}{q}. \tag{1.12}$$

1.1.4 Codes en bloc linéaires

L'utilisation de codes en bloc linéaires représente l'une des techniques possibles de codage de canal. Nous allons définir dans ce paragraphe ces codes ainsi que les outils nécessaires à leur décodage : de nombreux ouvrages de références, [5, 7, 41, 43, 50] par exemple, présentent dans sa quasi-totalité ce domaine.

Un code en bloc linéaire q-aire de longueur n et de dimension k, noté (n, k), est un sous-espace vectoriel de dimension k de F_q^n, l'espace vectoriel de dimension n construit sur le corps fini F_q. La taille d'un tel code, c'est-à-dire son nombre de mots, est donc q^k. La distance de Hamming minimale entre deux mots de code distincts définit la distance minimale du code, notée $d_{\min}$. À cause de la linéarité du code, il suffit de chercher le mot de code non-nul de poids minimal, i.e. le plus proche du mot nul. Cette distance est liée à la capacité de correction t, i.e. le nombre de symboles erronés qu'un bon décodeur de canal est sûr de pouvoir corriger, par la relation

$$t = \lfloor \frac{d_{\min} - 1}{2} \rfloor.$$

Si le code ne corrige aucun motif d'erreur de poids de Hamming strictement supérieur à t, le code est dit **parfait**. Si le code corrige des motifs d'erreurs de poids $t+1$ et aucun de poids strictement supérieur à $t+1$, le code est dit **quasi-parfait**.

Matrice géneratrice et matrice de parité

Par définition, un code linéaire $\mathcal{C}$ (n, k) se confond avec l'espace image d'une matrice de taille $k \times n$ dite **matrice génératrice** G : un mot de code $\underline{v}$ est tiré du mot d'information $\underline{i}$ par la relation $\underline{v} = \underline{i}G$. L'espace orthogonal du code linéaire est representé par une matrice H dite **matrice de parité** et de taille $(n-k) \times n$. Matrice génératice et matrice de parité vérifient donc la relation :

$$GH^t = 0$$

Le code engendré par la matrice de parité est appelé **code dual** de $\mathcal{C}$ et noté $\mathcal{C}^\perp$. Les matrices G et H associées à un code donné ne sont pas uniques. On distingue notamment la forme de la matrice génératrice s'écrivant $[I_k \ Par]$ où Par est une matrice de taille $k \times (n-k)$ à éléments dans F_q, forme dite **systématique**[4]. La matrice de parité associée à la matrice génératrice systématique s'écrit $[-Par^t \ I_{n-k}]$[5]. Deux codes qui ont même représentation systématique sont dits **équivalents** : leurs cellules de Voronoï sont identiques, seule la correspondance entre mots d'information et mots de code les distingue.

[4] la forme sytématique est unique.

[5] dans le cas binaire, soustraction et addition se confondent : par conséquent la matrice de parité s'écrit également $[Par^t \ I_{n-k}]$.

Décodage des codes linéaires

Le décodeur reçoit une version erronée $\underline{u}$ du mot de code émis. Afin de minimiser la probabilité d'erreur par mot, le décodeur cherche le mot de code le plus probable connaissant $\underline{u}$ (critère de maximum de vraisemblance a posteriori), ce qui revient à trouver le mot de code le plus proche de $\underline{u}$ au sens de la distance de Hamming si

- $p < 1/q$;
- les mots d'information sont équiprobables.

Nous supposerons ces deux hypothèses vérifiées par la suite. Afin de pouvoir décoder le mot $\underline{u}$, nous définissons la relation d'équivalence $\mathcal{R}$ par

$$\underline{v}\mathcal{R}\underline{u} \iff \underline{v} - \underline{u} \in \mathcal{C}$$

$\mathcal{C}$ étant un groupe, $\mathcal{R}$ définit les q^{n-k} classes d'équivalence de F_q^n, classes qui comportent toutes le même nombre d'éléments. Le **chef** d'une classe donnée (*coset leader*) est l'un des éléments de la classe de poids minimal. L'ensemble des chefs de classe forme la **cellule de Voronoï** du mot nul, c'est-à-dire l'ensemble des mots reçus pour lesquels le décodeur produira comme sortie le mot nul. Cette cellule est notée $V(\underline{0})$. La cellule de Voronoï d'un mot de code quelconque $\underline{v}$ est le translaté par $\underline{v}$ de $V(\underline{0})$: on note $V(\underline{v}) = \underline{v} + V(\underline{0})$. H engendrant le code dual de $\mathcal{C}$, le **syndrome** S défini par $S = \underline{u}H^t$ caratérise de manière unique une classe d'équivalence.

Le **tableau standard de décodage** consiste à écrire tous les mots de F_q^n en q^{n-k} lignes. Chaque ligne comporte les q^k éléments d'une classe ordonnés de telle manière que chaque colonne puisse s'écrire comme le translaté d'une autre colonne. Les colonnes de ce tableau sont donc les cellules de Voronoï des mots du code. Deux codes équivalents ont donc même tableau de décodage.

Exemple 1 *Soit G la matrice génératrice d'un code binaire de longueur 5 et de dimension 3 :*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

La forme systématique de G, notée G_{sys}, est alors

$$G_{sys} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} = [I_3\ Par].$$

La matrice de parité H associée à cette matrice systématique est donc

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} = [Par^t\ I_2].$$

Le tableau de décodage est donc :

Cellules de Voronoï								*Syndrome*
00000	*10011*	*01001*	*00110*	*11010*	*10101*	*01111*	*11100*	*00*
00001	*10010*	*01000*	*00111*	*11011*	*10100*	*01110*	*11101*	*01*
00010	*10001*	*01011*	*00100*	*11000*	*10111*	*01101*	*11110*	*10*
10000	*00011*	*11001*	*10110*	*01010*	*00101*	*11111*	*01100*	*11*

On peut obtenir un autre tableau de décodage à partir de celui-ci en échangeant dans la deuxième ligne 00001 et 01000 car ces deux éléments d'une même classe ont un poids égal au poids minimal de la classe. Les autres éléments de la classe doivent alors être réarrangés pour conserver la propriété de translation des cellules de Voronoï. Le choix d'une matrice de parité associée à G et non G_{sys} conduirait à permuter les trois dernières lignes de la colonne des syndromes.

Ce code est quasi-parfait : sa capacité de correction est nulle et il corrige quelques erreurs de poids 1 *et aucune de poids supérieur ou égal à* 2*.*

Nous appuyant sur ce tableau, il nous est possible de décoder par la méthode du tableau standard (*standard array decoding*) le mot reçu $\underline{u}$:

- calcul du syndrome $S = \underline{u}H^t$;
- le chef de classe associé au syndrome S est soustrait au mot reçu ;
- extraction, si nécessaire, des bits d'information du mot de code ainsi formé.

Cette dernière étape est spécifique à la matrice génératrice G choisie : les décodeurs associés à deux codes équivalents et recevant la même sortie de canal $\underline{y}$ produiront *a priori* un mot d'information différent. Le choix de la matrice de parité n'influe pas sur le mot de code fourni par ce décodeur : elle ne sert qu'à trouver la classe de l'erreur.

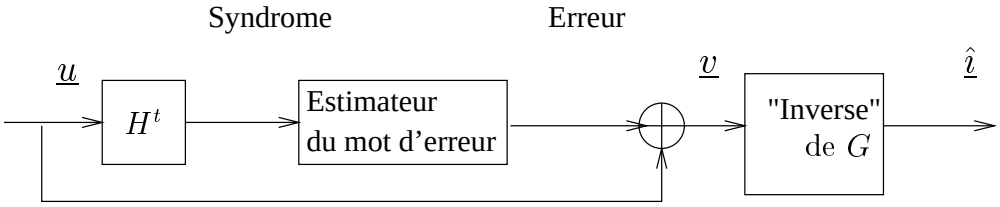


Fig. 1.4 – Décodeur de canal pour un code linéaire binaire.

Un tel algorithme de décodage, illustré sur la figure 1.4, est dit **complet** car il trouve le mot de code le plus proche quel que soit le mot reçu. La recherche des chefs de classe est difficile pour les codes de rendement faible [41]. Enfin il ne nous est pas apparu possible de relier les poids des chefs de classe du code à ceux de son dual, contrairement aux poids des mots de code reliés aux poids des mots de code du dual par l'identité de Pless-McWilliams : ce point constitue d'ailleurs un problème ouvert référencé par McWilliams et Sloane [43].

Performances des codes linéaires

Nous allons donner dans ce paragraphe les expressions exactes des probabilités d'erreur par mot et par symbole des codes linéaires (nous supposons toujours que les mots d'information sont équiprobables et que $p < \frac{1}{q}$). La probabilité d'erreur par mot P_{em} est minimale lorsque le décodeur choisit le mot de code le plus probablement émis connaissant le mot reçu. Pour déterminer cette probabilité d'erreur, nous considérons l'événement complémentaire, c'est-à-dire à la probabilité que le mot décidé (la sortie de l'algorithme décrit précédemment) soit le mot effectivement émis. Cet événement se produit lorsque le mot reçu est un des mots composant la cellule de Voronoi du mot émis, ce qui revient à écrire :

$$P_{em} = 1 - \sum_{\underline{x} \in \mathcal{C}} p(\underline{x}) \Pr(\underline{y} \in V(\underline{x}) | \underline{x}).$$

Récrivons $\Pr(\underline{y} \in V(\underline{x})|\underline{x})$:

$$\Pr(\underline{y} \in V(\underline{x})|\underline{x}) = \sum_{\underline{y} \in V(\underline{x})} p(\underline{y}|\underline{x}) = \sum_{i=0}^{n} p^i (1-P)^{n-i} \alpha_i(\underline{x})$$

où $\alpha_i(\underline{x})$ désigne le nombre de mots de $V(\underline{x})$ à distance i de $\underline{x}$. En tenant compte de l'équiprobabilité des mots d'information, P_{em} devient :

$$P_{em} = 1 - \sum_{i} p^i (1-P)^{n-i} \alpha_i \tag{1.13}$$

où α_i est la moyenne des $(\alpha_i(\underline{x}))$ (plus de détails sur ces derniers coefficients sont donnés dans l'annexe B) : ils dépendent *a priori* de toutes les cellules de Voronoï. Cependant, dans le cas des codes linéaires, comme les cellules de Voronoï sont translatées les unes des autres, les α_i ne dépendent que de la cellule de Voronoï du mot nul : la probabilité d'erreur par mot est alors plus simple à calculer. Deux codes linéaires équivalents produiront donc la même probabilité d'erreur par mot.

La probabilité d'erreur par symbole d'un code linéaire systématique, P_{es}, définie par (1.3), se récrit en tenant compte des hypothèses

$$P_{es} = \frac{1}{k} \sum_{\underline{\hat{v}} \in \mathcal{C}} d_H^{1 \cdots k}(\underline{\hat{v}}, \underline{0}) \Pr(V(\underline{\hat{v}}) | \underline{v} = \underline{0}), \tag{1.14}$$

où $d_H^{1 \cdots k}(\underline{v}, \underline{u})$ désigne le nombre de composantes distinctes entre les mots $\underline{v}$ et $\underline{u}$ parmi les composantes de ces mots dont l'indice varie entre 1 et k. Supposons en effet que le mot émis soit le mot nul. Le mot reçu $\underline{y}$ appartenant à la cellule de Voronoï de $\underline{c}$, le mot d'information décidé est donc celui correspondant à $\underline{c}$, ce qui, dans le cas d'un code systématique, correspond à ses k premières composantes, ce que traduit (1.14). Si l'on s'intéresse à un symbole d'information particulier, celui en position i, la probabilité d'erreur associée, notée $P_{es}^{(i)}$ a pour expression

$$P_{es}^{(i)} = \sum_{\underline{\hat{v}} \in \mathcal{C}, \hat{v}_i \neq 0} \Pr(V(\underline{\hat{v}}) | \underline{v} = \underline{0}). \tag{1.15}$$

Deux codes équivalents ne produiront pas *a priori* la même probabilité d'erreur par symbole.

Suite de l'exemple 1 *Reprenons le code systématique de longueur* 5 *et de dimension* 3 *défini dans l'exemple 1. Le tableau standard de décodage nous permet de calculer ses performances exactes. La probabilité d'erreur par mot de ce code est*

$$P_{em} = 1 - (1-p)^5 - 3p(1-p)^4 = 2p + 2p^2 - 8p^3 + 7p^4 - 2p^5. \tag{1.16}$$

Le tableau de décodage du code systématique permet de calculer la probabilité d'erreur pour chaque bit d'information

$$\begin{aligned} P_{es}^{(1)} &= 8p^2 - 20p^3 + 20p^4 - 8p^5 \\ P_{es}^{(2)} &= p \\ P_{es}^{(3)} &= p \end{aligned}$$

La probabilité d'erreur par symbole de ce même code est donc

$$\begin{aligned} P_{es} &= \frac{1}{3}(2p(1-p)^4 + 16p^2(1-p)^3 + 16p^3(1-p)^2 + 12p^4(1-p) + 2p^5) \\ &= \frac{1}{3}(2p + 8p^2 - 20p^3 + 20p^4 - 8p^5) \end{aligned} \tag{1.17}$$

Quant au code non systématique défini par la matrice G, les probabilités d'erreur par symbole sont données, tous calculs faits, par :

$$P_{es}^{(1)} = 8p^2 - 20p^3 + 20p^4 - 8p^5 \tag{1.18}$$
$$P_{es}^{(2)} = 2p - 2p^2 \tag{1.19}$$
$$P_{es}^{(3)} = p \tag{1.20}$$
$$P_{es} = \frac{1}{3}(3p + 6p^2 - 20p^3 + 20p^4 - 8p^5) \tag{1.21}$$

Il apparaît alors que ce code non systématique a une probabilité d'erreur par symbole plus grande que celle du code systématique associé !

1.1.5 Bornes sur les probabilités d'erreur

Après ces rappels permettant de déterminer les performances exactes des codes linéaires, nous allons présenter dans cette section plusieurs nouvelles bornes des probabilités d'erreur afin d'évaluer plus facilement les performances du système représenté sur la figure 1.1. Ces bornes, contrairement aux performances optimales, dépendent de la longueur et de la taille des codes.

Borne inférieure de P_{em}

Puisque $p < 1/q$, $i \mapsto -p^i(1-P)^{n-i}$ est une fonction strictement croissante. Nous appliquons alors les résultats de l'annexe B en choisissant pour fonction f, la fonction définie par $f(i) = -p^i(1-P)^{n-i}$. Ces résultats nous permettent d'obtenir une borne inférieure de P_{em} (en omettant dans les notations la dépendance vis-à-vis du paramètre d de la borne inférieure).

Théorème 1.1.1 *Soit*

$$P_{emi} = 1 - \sum_{i=0}^{d-1}(q-1)^i C_n^i p^i (1-P)^{n-i} - \left(q^{n-k} - \sum_{i=0}^{d-1}(q-1)^i C_n^i\right) p^d (1-P)^{n-d}, \quad (1.22)$$

d étant le plus grand entier vérifiant :

$$\sum_{i<d}(q-1)^i C_n^i \leq q^{n-k}.$$

P_{emi} est une borne inférieure de la probabilité d'erreur par mot valable pour les codes linéaires (n,k) et non linéaires. Cette borne donne la probabilité exacte d'erreur par mot si et seulement si le code est parfait ou quasi-parfait.

L'annexe D dresse une liste des codes parfaits et quasi-parfaits (liste non exhaustive pour ces derniers) et indique la valeur ou l'expression de P_{em} pour ces codes.

Interprétons maintenant cette borne dans le cas d'un code linéaire. Nous allons sous-estimer le poids des chefs de classe triés par poids croissant et les enlever au fur et à mesure de cette liste. Fixons la variable i à 0. Le poids du premier ($(q-1)^i C_n^i = 1$) chef de classe est sous-estimé par i : nous enlevons ce chef de classe (qui est le mot nul puisque le code est linéaire) de la liste. Incrémentons alors i : i vaut 1. Il nous reste $q^{n-k}-1$ chefs de classe. Prenons les $(q-1)^i C_n^i$ premiers restants et sous-estimons leurs poids par i. Généralisons cette procédure : i prend alors une valeur quelconque. À l'étape i, il restera $q^{n-k} - \sum_{j=0}^{i-1}(q-1)^j C_n^j$ chefs de classe à traiter. Leur poids est par récurrence supérieur ou égal à i. Si $\sum_{j=0}^{i}(q-1)^j C_n^j < q^{n-k}$, nous choisissons alors les $(q-1)^i C_n^i$ premiers chefs de classe restants et nous sous-estimons leur poids par i. Leur contribution à P_{em} est donc minorée par $(q-1)^i C_n^i p^i (1-P)^{n-i}$. Il nous reste $q^{n-k} - \sum_{j=0}^{i}(q-1)^j C_n^j$ chefs de classes non marqués de poids supérieur ou égal à $i+1$. Nous répétons cette procédure jusqu'à $i = d-1$: après cette dernière étape, il reste $q^{n-k} - \sum_{i<d}(q-1)^i C_n^i$ chefs de classe non marqués de poids supérieur ou égal à d. Nous retrouvons ainsi l'expression de P_{emi}.

Lorsque le canal est **opaque** ($p = 1/q$) ou **parfait** ($p = 0$), cette borne donne la valeur exacte de la probabilité d'erreur par mot ($1 - q^{-k}$ pour le canal opaque et 0 pour le canal parfait). Nous pouvons pousser plus loin l'étude de cette borne en nous intéressant à son comportement asymptotique et la relier ainsi à la capacité du canal.

Théorème 1.1.2 *P_{emi} converge vers 0 lorsque $R_c < C_{CqS}(P)$ et vers 1 lorsque $R_c > C_{CqS}(P)$.*

Preuve
La preuve, un peu longue, est donnée dans l'annexe C.2.

□

Ce théorème est la réciproque du théorème de codage de canal de Shannon : il donne une limite inférieure des performances des codes en bloc pour une complexité donnée alors que le théorème de Shannon donne des performances asymptotiques. La figure 1.5 illustre la convergence, dans le cas binaire, de cette borne pour trois longueurs n distinctes (10, 20 et 50). .

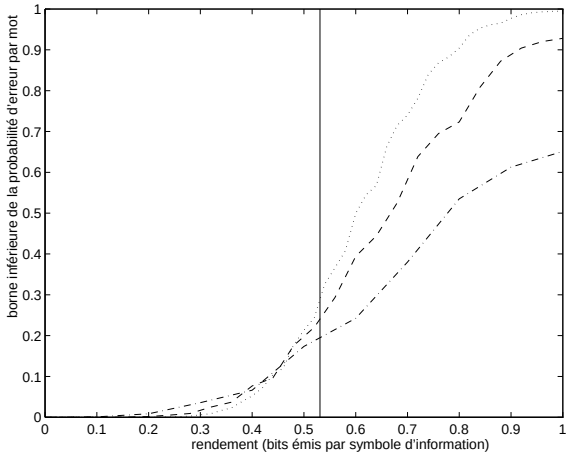


Fig. 1.5 – Borne inférieure de la probabilité d'erreur pour différentes longueurs et différents rendements : $n = 10$ (-.), $n = 25$ (–), $n = 50$ (..) pour $q = 2$ et $p = 0.1$. La ligne continue représente la capacité du CBS.

Borne inférieure sur P_{es}

La probabilité d'erreur par mot P_{em} a été largement plus étudiée dans la littérature que la probabilité d'erreur par symbole, cependant l'étude de la probabilité d'erreur par symbole est nécessaire lorsque nous aborderons le codage conjoint au chapitre suivant.

Pour minorer P_{es}, il suffit de classer les mots d'information par poids croissant et d'attribuer à leur cellule de Voronoï associée une probabilité de plus en plus faible en oubliant les contraintes imposées par la linéarité du code sur ces cellules. Cette idée se traduit par le théorème suivant :

Théorème 1.1.3 *Soit p le paramètre d'un CqS tel que $p < 1/q$. Posons $P = (q-1)p$. Considérons la suite $(s_i)_{i\in\mathbb{N}}$ définie par*

$$s_i = \sum_{w=0}^{i} (q-1)^w C_k^w$$

et la suite $(w_i)_{i=1,\cdots,q^k}$ *définie par*

$$w_i = \arg\max\{j, \sum_{w=0}^{j} (q-1)^w C_n^w \leq iq^{n-k}\}. \tag{1.23}$$

Enfin, posons $p(w) = p^w(1-P)^{n-w}$.

La probabilité d'erreur par symbole d'un code linéaire de longueur n *et de dimension* k *sur CqS est minorée par la probabilité* P_{esi} *définie par*

$$P_{esi} = \frac{1}{k}\left(\sum_{a=1}^{k} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} (q-1)^w C_n^w p(w) + \sum_{a=0}^{k} (\sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w - s_a q^{n-k}) p(w_{s_a}+1)\right) \tag{1.24}$$

PREUVE

La démonstration repose sur deux idées :

- le classement des mots de F_q^n par poids croissant, i.e. par probabilité décroissante, et leur regroupement en ensembles de q^{n-k} éléments (qui jouent le rôle de cellules de Voronoï) ainsi que le classement des mots d'information par poids croissant ;
- les $(q-1)^w C_k^w$ « cellules de Voronoï » associées aux mots d'information de poids w voient leurs probabilités multipliées par w.

La première « cellule de Voronoï » associée au mot d'information nul contient les mots de poids $0, \cdots, w_1$ et $q^{n-k} - \sum_{w=0}^{w_1} (q-1)^w C_n^w$ mots de poids $w_1 + 1$. La i^{e} « cellule » est composée de :

- $(q-1)^{w_{i-1}+1} C_n^{w_{i-1}+1} - ((i-1)q^{n-k} - \sum_{w=0}^{w_{i-1}} (q-1)^w C_n^w)$ mots de poids $w_{i-1}+1$;
- $(q-1)^{w_{i-1}+2} C_n^{w_{i-1}+2}$ mots de poids $w_{i-1}+2$;
- $\ldots$;
- $(q-1)^{w_i} C_n^{w_i}$ mots de poids w_i ;
- $iq^{n-k} - \sum_{w=0}^{w_i} (q-1)^w C_n^w$ mots de poids $w_i + 1$.

Par conséquent, la contribution des $(q-1)^a C_k^a$ mots d'information estimés de poids a à la probabilité d'erreur s'écrit

$$\begin{aligned}
&\sum_{l=s_{a-1}+1}^{s_a} \left\{ \sum_{w=w_{l-1}+2}^{w_l} (q-1)^w C_n^w p(w) + \left(\sum_{w=0}^{w_{l-1}+1} (q-1)^w C_n^w - (l-1)q^{n-k} \right) p(w_{l-1}+1) \right. \\
&\left. + \left(lq^{n-k} - \sum_{w=0}^{w_l} (q-1)^w C_n^w \right) p(w_l+1) \right\} \\
= &\sum_{w=w_{s_{a-1}}+2}^{w_{s_a}} (q-1)^w C_n^w p(w) + \left(\sum_{w=0}^{w_{s_{a-1}}+1} (q-1)^w C_n^w - s_{a-1} q^{n-k} \right) p(w_{s_{a-1}}+1) \\
&+ \left(s_a q^{n-k} - \sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w \right) p(w_{s_a}+1)
\end{aligned}$$

La probabilité d'erreur est donc minorée, après quelques manipulations algébriques, par

$$P_{es} \geq \frac{1}{k} \sum_{a=1}^{k} a \left\{ \sum_{w=w_{s_{a-1}+2}}^{w_{s_a}} (q-1)^w C_n^w p(w) + \left(\sum_{w=0}^{w_{s_{a-1}+1}} (q-1)^w C_n^w - s_{a-1} q^{n-k} \right) p(w_{s_{a-1}}+1) \right.$$

$$+\left(s_a q^{n-k} - \sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w\right) p(w_{s_a}+1)\Bigg\}$$

$$\geq \frac{1}{k}\left(\sum_{a=1}^{k} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} (q-1)^w C_n^w p(w) + \sum_{a=0}^{k}\left(\sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w - s_a q^{n-k}\right) p(w_{s_a}+1)\right)$$

On reconnaît dans le dernier membre de l'inégalité la définition de P_{esi}.

□

Remarquons que cette borne vaut $1-1/q$ lorsque le canal est opaque, i.e. pour $p = 1/q$, ce qui est la valeur exacte de la probabilité d'erreur par symbole dans ce cas. Plus intéressant est le comportement asymptotique de cette borne, que nous allons voir à présent.

Théorème 1.1.4 *Supposons que l'on considère une famille de codes (n,k) de rendement limite R_c (i.e. vérifiant $\lim_{n\to\infty} \frac{k}{n}\log_2 q = R_c$) tel que $R_c < C_{CqS}(P)$ alors la borne inférieure de la probabilité d'erreur par symbole P_{esi} tend vers 0 :*

$$\lim_{n\to\infty} P_{esi} = 0$$

Preuve
La démonstration est donnée dans l'annexe C.4.

□

Nous illustrons le comportement asymptotique de la borne par la figure 1.6. Nous avons choisi un CBS de paramètre $p = 0.1$. La capacité est donc égale à 0.5310. Le rendement R_c prend trois valeurs : 0.331, 0.6 et 0.731. On remarque que la borne semble avoir une limite lorsque le rendement est supérieur à la capacité. Le théorème qui suit justifie cette remarque, complétant ainsi l'étude du comportement asymptotique de P_{esi}

Théorème 1.1.5 *Supposons que l'on considère une famille de codes (n,k) de rendement limite R_c (i.e. vérifiant $\lim_{n\to\infty} \frac{k}{n}\log_2 q = R_c$) tel que $\log_2 q > R_c > C_{CqS}(P)$ alors la borne inférieure de la probabilité d'erreur par symbole P_{esi} admet une limite :*

$$\lim_{n\to\infty} P_{esi} = C_{CqS}^{-1}\left(\frac{\log_2 q}{R_c} C_{CqS}(P)\right)$$

Preuve
La preuve est donnée, encore une fois, dans l'annexe C.4.

□

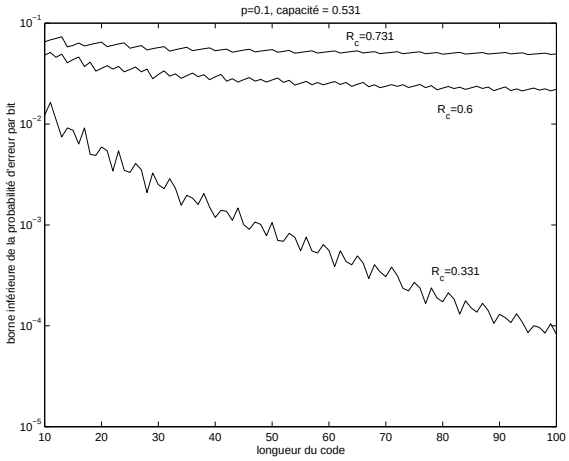


FIG. 1.6 – Comportement de P_{esi} en fonction du taux considéré et de la longueur des codes sur un canal binaire symétrique.

Les deux théorème précédents constituent à eux deux une réciproque du théorème de codage de canal de Shannon : la borne P_{esi} est une borne inférieure des performances des codes linéaires dont la convergence est en accord avec le théorème de Shannon. La borne présentée dépend des paramètres du code, ce qui constitue tout son intérêt par rapport au théorème de Shannon.

Le théorème ci-dessus montre que la borne inférieure de la probabilité d'erreur par symbole P_{em}/k (double inégalité (1.4)) est mauvaise lorsque le rendement est supérieur à la capacité du canal puisque cette borne converge vers 0. Terminons l'étude de cette borne par un exemple illustrant à la fois la démonstration de la borne et la borne elle-même.

Suite de l'exemple 1 *Nous devons classer les mots de* F_2^5 *en huit ensembles de* $2^{n-k} = 4$ *éléments. Ces huit ensembles joueront le rôle des cellules de Voronoï. Le premier, associé au mot d'information de poids nul, contient le mot nul de* F_2^5 *ainsi que trois des mots de* F_2^5 *de poids* 1. *Le deuxième, associé à un mot d'information de poids* 1, *par exemple* $(0,0,1)$, *contient les deux mots restants de poids* 1 *et deux mots de poids* 2. *Les ensembles associés aux mots d'information* $(0,1,0)$ *et* $(1,0,0)$ *contiennent chacun quatre mots de poids* 2. *Il n'y a donc plus de mots de* F_2^5 *de poids* 2 *disponibles. Les ensembles associés aux mots d'information* $(0,1,1)$ *et* $(1,0,1)$ *contiennent chacun quatre mots de poids* 3. *L'ensemble associé à* $(1,1,0)$ *contient les deux mots de poids* 3 *restants et deux mots de poids* 4. *Enfin, l'ensemble associé au mot d'information* $(1,1,1)$ *contient les trois mots de poids* 4 *restants et l'unique mot de poids* 5.
Cela revient à écrire que

$$w_{s_0} = w_1 = 0,\ w_{s_1} = w_4 = 2,\ w_{s_2} = w_7 = 3,\ w_{s_3} = w_8 = 5.$$

Ces équations signifient que l'union des « cellules de Voronoï » associées aux mots d'informations de poids de Hamming inférieur ou égal à l'indice i *de* s *(respectivement 0, 1, 2, 3) contient tous les mots de* F_2^5 *de poids de Hamming inférieur ou égal à* w_{s_i}

(respectivement 0, 2, 3, 5). Nous sommes en mesure de donner alors l'expression de la borne P_{esi} en fonction du paramètre p du CBS :

$$P_{esi} = \frac{1}{3}(2p(1-p)^4 + 10p^2(1-p)^3 + 20p^3(1-p)^2 + 13p^4(1-p) + 3p^5) \quad (1.25)$$
$$= \frac{1}{3}(2p + 2p^2 + 2p^3 - 5p^4 + 2p^5). \quad (1.26)$$

Nous pouvons remarquer que, pour ces valeurs de n et k, la borne P_{esi} est supérieure à P_{emi}/k, nous apportant par là plus de précision.

Borne supérieure tirée du codage aléatoire

Les bornes inférieures de performances ne sont pas suffisantes pour juger des performances d'un code donné : une borne supérieure aléatoire permet de juger des performances d'un code par rapport à un code choisi au hasard. Nous allons donc rappeler ces bornes et leur principe.

L'idée du codage aléatoire, c'est-à-dire de calculer une probabilité d'erreur moyennée sur un ensemble de codes déterminé, a permis d'aboutir au théorème de codage de canal énoncé au paragraphe 1.1.3. Elle donne, en outre, une borne supérieure de la probabilité d'erreur par mot. Nous présentons ici des résultats classiques, traditionnellement présentés pour le cas binaire (troisième chapitre de [64] qui reprend les travaux de Gallager), en les adaptant au cas q-aire.

Sans reprendre toutes les étapes de calcul, la probabilité P_{em} est majorée par

$$P_{em} < 2^{-n(E_0(\rho,\tilde{p})-\rho R)}, \quad 0 \leq \rho \leq 1$$

où $\tilde{p}$ est une distribution de probabilités sur l'alphabet de canal F_q et où $E_0(\rho, \tilde{p})$ est défini par

$$E_0(\rho, \tilde{p}) = -\log_2 \sum_{y \in F_q} \left\{ \sum_{x \in F_q} \tilde{p}(x) p(y|x)^{\frac{1}{1+\rho}} \right\}^{1+\rho}.$$

En vue d'optimiser la borne, il faut maximiser $E_0(\rho, \tilde{p})$ par rapport à ρ et à $\tilde{p}$. Nous restreignons l'étude de ces bornes au cas du CqS. Intuitivement, nous choisirions une distribution $\tilde{p}$ uniforme, ce qui est confirmé mathématiquement [64]. Posons $E_0(\rho) = E_0(\rho, \tilde{p}$ uniforme) et $E(R) = \max_{0 \leq \rho \leq 1} E_0(\rho) - \rho R$. Les probabilités du CqS permettent de calculer ce dernier :

$$E_0(\rho) = \rho \log_2 q - (1+\rho) \log_2 \left\{ (q-1) p^{\frac{1}{1+\rho}} + (1-P)^{\frac{1}{1+\rho}}) \right\}.$$

Afin d'alléger les équations suivantes, posons

$$\hat{p}(\rho) = \frac{(q-1)p^{\frac{1}{1+\rho}}}{(q-1)p^{\frac{1}{1+\rho}} + (1-P)^{\frac{1}{1+\rho}}}.$$

Il ne reste plus qu'à se préoccuper du choix de ρ : une étude de la convexité de $E_0(\rho)$ met en évidence l'existence d'un maximum. Par conséquent, le maximum de $\rho \mapsto E_0(\rho) - \rho R$ est atteint soit par annulation de la dérivée par rapport à ρ soit en $\rho = 1$, ce que l'on résume en

$$\begin{array}{ll} 0 \leq R \leq E_0'(1) & E(R) = \log_2(q) - 2\log_2\{(q-1)\sqrt{p} + \sqrt{1-P}\} - R \\ E_0'(1) \leq R \leq C_{CqS}(P) & \left\{ \begin{array}{l} E(R) = T_p(\hat{p}(\rho)) - H_2(\hat{p}(\rho)) \\ R = C_{CqS}(\hat{p}(\rho)) \end{array} \right. \end{array} \tag{1.27}$$

où $T_y : x \mapsto -x\log_2 y - (1-x)\log_2(1-y)$ est la tangente en y à la fonction H_2. $E_0'(1)$ s'exprime comme une capacité de CqS :

$$E_0'(1) = \log_2 q - H_2(\hat{p}(1)) - \hat{p}(1)\log_2(q-1) = C_{CqS}(\hat{p}(1))$$

Cette borne a été améliorée dans le cas de faibles rendements par Gallager : la probabilité P_{em} est alors majorée par

$$P_{em} < 2^{-n \max_{\tilde{p}} \sup_{\rho \geq 1}(E_x(\rho,\tilde{p}) - \rho R)}$$

avec

$$E_x(\rho, \tilde{p}) = -\rho \log_2 \left\{ \sum_{(x,x') \in F_q \times F_q} \tilde{p}(x)\tilde{p}(x') \left(\sum_{y \in F_q} \sqrt{p(y|x)p(y|x')} \right)^{\frac{1}{\rho}} \right\}.$$

Là encore, dans le cas du CqS, la distribution $\tilde{p}$ qui maximise $E_x(\rho, \tilde{p})$ est la distribution uniforme :

$$\max_{\tilde{p}} E_x(\rho, \tilde{p}) = -\rho \log_2 \left\{ \left(1 - \frac{1}{q}\right) \left(\sqrt{4p(1-P)} + (q-2)p \right)^{\frac{1}{\rho}} + \frac{1}{q} \right\}.$$

Posons $Z = \sqrt{4p(1-P)} + (q-2)p$ et $E_x(R) = \max_{\tilde{p}} \sup_{\rho \geq 1}(E_x(\rho, \tilde{p}) - \rho R)$. L'étude de la maximisation par rapport à ρ permet d'affirmer que

$$\begin{aligned} E_x(R) &= -\frac{(q-1)Z^{\frac{1}{\rho}}}{1 + (q-1)Z^{\frac{1}{\rho}}} \log_2 Z, \\ R &= \log_2 q - H_2\left(\frac{(q-1)Z^{\frac{1}{\rho}}}{1 + (q-1)Z^{\frac{1}{\rho}}} \right) - \frac{(q-1)Z^{\frac{1}{\rho}}}{1 + (q-1)Z^{\frac{1}{\rho}}} \log_2(q-1) = C_{CqS}\left(\frac{(q-1)Z^{\frac{1}{\rho}}}{1 + (q-1)Z^{\frac{1}{\rho}}} \right) \end{aligned}$$

lorsque R est compris entre 0 et $C_{CqS}(\frac{(q-1)Z}{1+(q-1)Z})$.

1.1.6 À la recherche de bons codes linéaires

Nous souhaitons trouver de bons codes linéaires binaires : il est possible d'examiner tous les codes linéaires, cependant l'évaluation exacte de leurs performances nécessite

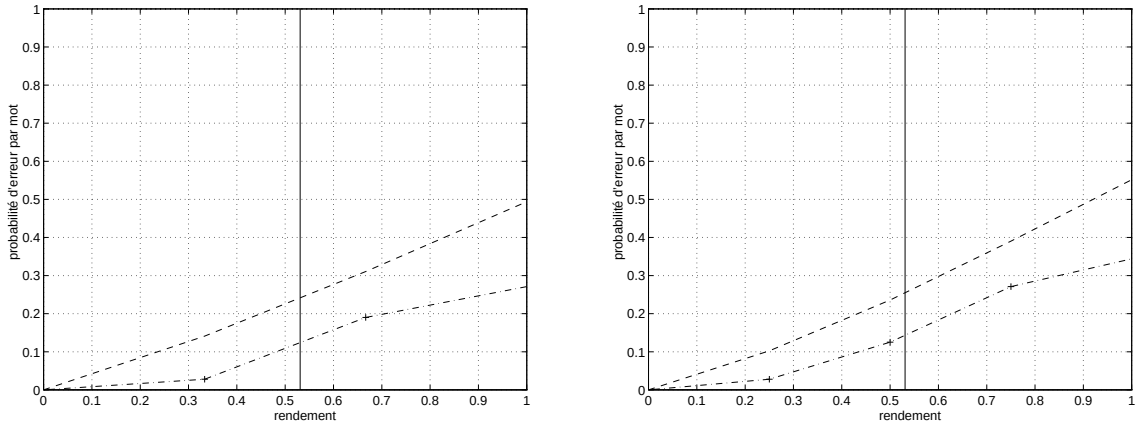
de calculer le tableau standard de décodage, ce qui limite le rendement. La probabilité d'erreur par mot étant plus facile à calculer que celle par symbole, nous nous contenterons de la première. De plus, les sous-espaces vectoriels de F_2^n de dimension k sont au nombre de

$$\frac{(2^n-1)(2^{n-1}-1)\cdots(2^2-1)(2-1)}{(2^k-1)\cdots(2-1)(2^{n-k}-1)(2^{n-k-1}-1)\cdots(2-1)},$$

ce qui croît rapidement. Nous nous restreignons alors aux codes linéaires systématiques, sans perte de généralité puisque tout code linéaire en bloc admet une représentation systématique et que leurs performances sont équivalentes. À ce stade, la recherche exhaustive examinerait donc $2^{k(n-k)}$ codes. Il est possible de réduire ce nombre à $C^{n-k}_{n-k+2^k-1}$ en tirant parti de l'équivalence entre matrices génératrices (l'ordre des colonnes de la matrice Par n'importe pas). Enfin, puisque mener la recherche sur les matrices génératrices systématiques est équivalent à mener la recherche sur les matrices de parité systématique, le nombre de codes à examiner est réduit à

$$\min(C^{n-k}_{n-k+2^k-1}, C^{k}_{k+2^{n-k}-1}).$$

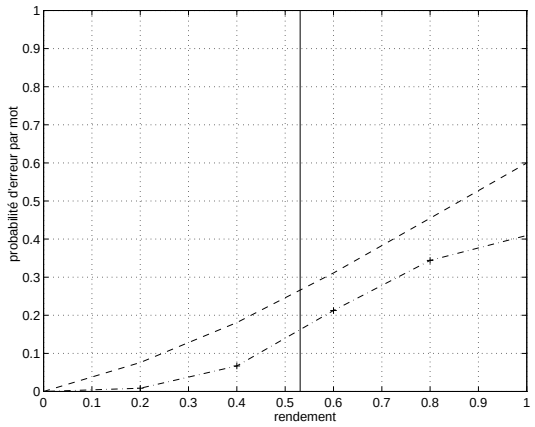
L'évaluation des performances nécessite de calculer la cellule de Voronoï du mot nul, c'est-à-dire la première colonne du tableau standard de décodage. Il faut donc trouver les 2^{n-k} chefs de classe en testant successivement les mots de F_2^n classés par poids croissant. Les figures 1.7-1.10 montrent la probabilité d'erreur par mot des meilleurs codes trouvés[6] et leur rendement. Ces codes sont marqués par une croix (+). Le paramètre p du CBS est fixé à 0.1 et la capacité de ce canal est représentée par la ligne verticale sur ces figures. La borne P_{emi} est tracée en traits mixtes et la borne supérieure correspondant à (1.27) en pointillés.



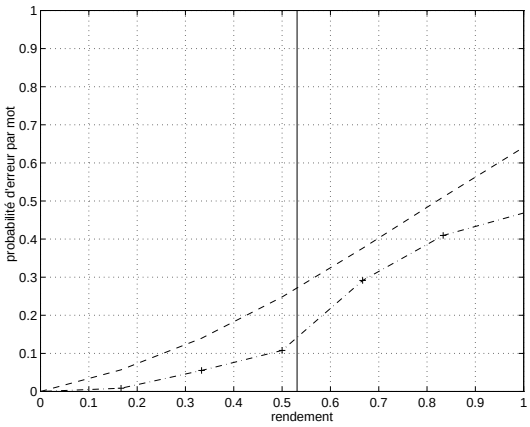
(a) : $n = 3$ (b) : $n = 4$

Fig. 1.7 – Meilleurs codes pour le codage de CBS, $p = 0.1$, $n = 3, 4$.

[6]Les codes linéaires trouvés dans ce chapitre ont été réutilisés par Zahir Azami dans un contexte de codage conjoint [69] : ils servent de points d'initialisation de l'algorithme JOB (Joint Optimization Binary (Coding)).

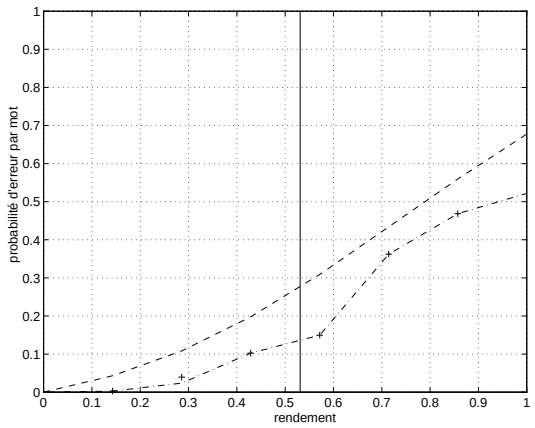


(a) : $n = 5$

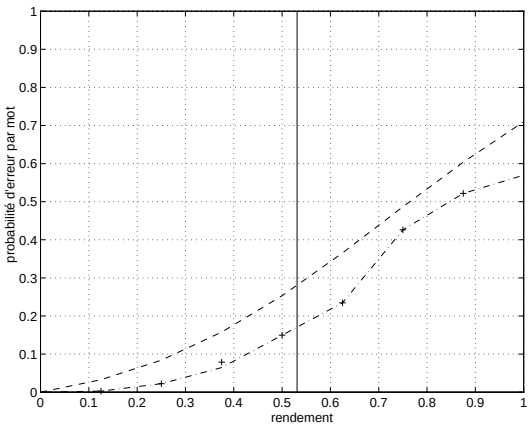


(b) : $n = 6$

FIG. 1.8 – Meilleurs codes pour le codage de CBS, $p = 0.1$, $n = 5, 6$.



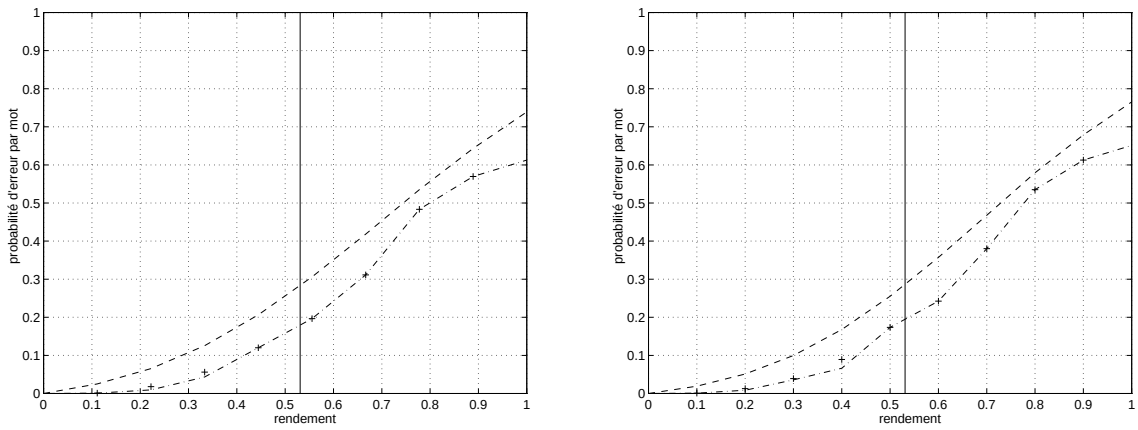
(a) : $n = 7$



(b) : $n = 8$

FIG. 1.9 – Meilleurs codes pour le codage de CBS, $p = 0.1$, $n = 7, 8$.

Nous avons vérifié que les meilleurs codes obtenus sont ceux indiqués dans [50]. Une très grande partie de ces codes sont quasi-parfaits ou parfaits : la borne inférieure P_{emi} est souvent atteinte.



(a) : $n = 9$ (b) : $n = 10$

Fig. 1.10 – Meilleurs codes pour le codage de CBS, $p = 0.1$, $n = 9, 10$.

1.2 Codage de source

1.2.1 Description du problème et mise en évidence de la dualité

Nous supposons que la source S est une source stationnaire q-aire, symétrique et sans mémoire. Les symboles de sources sont des éléments de l'alphabet de source $F_q = \{0, 1 \cdots, q-1\}$. Une telle telle source est appelée **source q-aire symétrique (SqS)**. Nous supposerons que q est une puissance d'un nombre premier : F_q est donc un corps fini. Considérons l'extension m^{e} de la source : un mot de source $\underline{u} = (u_0, \cdots, u_{m-1}) \in F_q^m$ est émis avec une probabilité $p(\underline{u}) = q^{-m}$. Le codage de source s'effectue en choisissant un code $\mathcal{C}$, c'est-à-dire M mots composés chacun de m symboles : $\mathcal{C} = \{\underline{v}_1, \cdots, \underline{v}_M\} \subset F_q^m$. Posons $k = \log_q(M)$, k est la dimension du code $\mathcal{C}$ lorsque ce dernier est linéaire. Le codeur de source procède en deux étapes :

- le quantificateur associe à chaque mot de source $\underline{u}$ un mot de code $\underline{v}$. $\underline{v}$ doit être proche de $\underline{u}$ afin de ne pas trop détériorer la source ;
- l'étiqueteur associe au mot $\underline{v}$ un mot $\underline{i}$ composé de $\log_2(M)$ bits.

La source est donc comprimée avec pertes dès que M est strictement inférieur à q^m. Nous supposons dans ce chapitre que les bits d'étiquetage sont transmis sur un canal sans bruit. Le décodeur de source se contente donc de fournir à l'utilisateur à partir du mot reçu $\underline{i}$ le mot du code de source associé $\underline{v}$. Ce schéma de codage de source est représenté sur la figure 1.11.

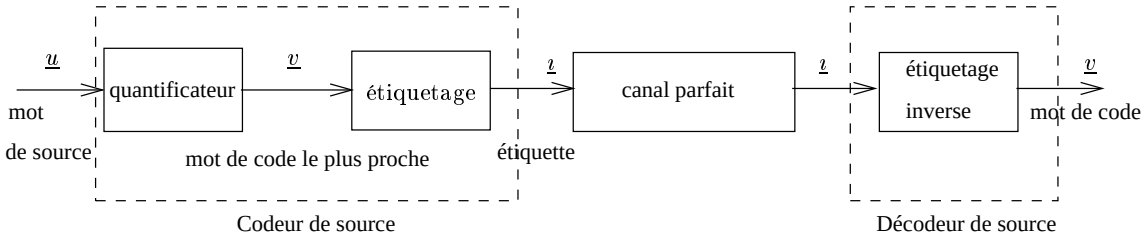


FIG. 1.11 – Modèle du codage de source.

Il apparaît donc clairement que **le codeur de source et le décodeur de canal** décrit au paragraphe 1.1.4 **remplissent les mêmes tâches, à savoir retrouver suivant un certain critère un mot de code à partir d'un mot reçu et fournir l'étiquette ou le mot d'information associé**. De même, **le décodeur de source et le codeur de canal accomplissent la même tâche**. Par conséquent, il est possible de comprimer la source S en utilisant le décodeur de canal comme codeur de source et le codeur de canal comme décodeur de source : codage de source et codage de canal sont donc deux activités duales, la première éliminant la redondance d'un message et la deuxième en ajoutant pour le prémunir des perturbations introduites par le canal. Cette dualité avait déjà été remarquée par T. Berger [4] : nonobstant, l'utilisation des codes en bloc linéaires en tant que codes de source ne semble pas avoir été étudiée intensivement.

En raison de cette dualité, nous espérons tirer profit des outils de codage de canal développés dans la section précédente afin d'obtenir des outils de codage de source sans travail supplémentaire ou presque !

Cependant, une différence notable existe entre le codage de source et le codage de canal : le codeur de source doit être capable de trouver le mot de code le plus proche du mot de source, quel que soit ce dernier. Or, le plus souvent, le décodeur de canal reçoit un mot qui est proche de celui émis : il peut se contenter de ne savoir décoder que ces mots-là. Ce décodage est dit **incomplet**, par opposition au codage de source, qui, lui, doit être **complet**.

1.2.2 Performances

Le rendement R_s du codage de source s'exprime en bits par symbole de source :

$$R_s = \frac{\log_2 M}{m} = \frac{k}{m} \log_2 q \tag{1.28}$$

Pour tirer profit de la dualité, le critère de fidélité choisi est celui de la probabilité d'erreur par symbole. Le codeur de source introduit une distorsion moyenne par symbole de source, notée D, égale à

$$D = \frac{1}{m} E[d_H(\underline{u}, \underline{v})] \tag{1.29}$$

où $E[.]$ désigne l'espérance mathématique (qui porte sur les mots de source $\underline{u}$). Le quantificateur qui minimise la distorsion à code de source donné vérifie la condition du plus proche voisin : il choisit le mot de code $\underline{v}$ tel que

$$d_H(\underline{u}, \underline{v}) = \min_{\underline{x} \in \mathcal{C}} d_H(\underline{u}, \underline{x}) \quad (= d_H(\underline{u}, \mathcal{C})).$$

Des mots de source peuvent être à égale distance de deux mots de code, voire plus. Ces cas d'égalité sont résolus par le hasard (si le code de source est linéaire, nous avons vu qu'on pouvait alors déterminer une fois pour toutes la sortie du codeur). Notons que le choix d'un mot de code particulier dans ces cas d'égalité n'a pas d'influence sur la distorsion : par conséquent, la distorsion D ne dépend que du code $\mathcal{C}$.

La distorsion dans le cas d'un code en bloc linéaire q-aire s'écrit de manière plus concise. En utilisant la symétrie de la source et la définition des cellules de Voronoï, on récrit (1.29)

$$\begin{aligned} D &= \frac{1}{mq^m} \sum_{v \in \mathcal{C}} \sum_{\underline{u} \in V(\underline{v})} d_H(\underline{u}, \underline{v}) \\ &= \frac{1}{mq^{m-k}} \sum_{i=0}^{m} i\alpha_i \end{aligned} \tag{1.30}$$

où les (α_i) ont été définis au paragraphe 1.1.4. Deux codes linéaires équivalents produiront donc la même distorsion. Remarquons que la sortie du codeur de source associé à un code linéaire est elle-même l'extension k^{e} d'une source q-aire symétrique. Illustrons ce calcul en poursuivant l'exemple simple introduit lors de l'étude du codage de canal.

Suite de l'exemple 1 *Le tableau de décodage du code engendré par la matrice G (ou par G_{sys}), ou plus exactement sa première colonne, nous permet de calculer la distorsion résultant de la compression d'une SBS par ce code* $D = \frac{1}{5\times 2^{5-3}}(0\times 1+3\times 1) = \frac{3}{20}$.

Quelle est la plus petite distorsion possible à rendement fixé ? Là encore, Shannon [13] a répondu à cette question.

Performances optimales : théorème du codage de source

Définissons, en premier lieu, la **fonction taux-distorsion** d'une source, notée $R(D)$:

$$R(D) = \inf_{m\in\mathbb{N}} \min_{p(\underline{v}|\underline{u})/E[d(\underline{U},\underline{V})]\leq mD} \frac{1}{m} I(\underline{U},\underline{V}) \tag{1.31}$$

où $I(\underline{U},\underline{V})$ est l'information mutuelle entre la source représentée par la variable aléatoire $\underline{U}$ de dimension m (émettant des mots $\underline{u}$) et la source reconstruite représentée par la variable aléatoire $\underline{V}$ (émettant des mots $\underline{v}$). $d(\underline{U},\underline{V})$ mesure la distance entre les deux variables. Cette fonction est aussi appelée **fonction rendement-distorsion**. La fonction taux-distorsion de la SqS noté R_{SqS} est définie, lorsque la distance choisie est celle de Hamming, par

$$R_{SqS}(D) \;=\; \begin{cases} \log_2 q - H_2(D) - D\log_2(q-1) & \text{si } 0 \leq D \leq 1-\frac{1}{q} \\ 0 & \text{si } 1-\frac{1}{q} < D \leq 1 \end{cases} \tag{1.32}$$

Cette fonction, décroissante et convexe, est représentée sur la figure 1.12 pour différentes valeurs de q. Sur le segment $[0, 1-1/q]$, les fonctions R_{SqS} et C_{CqS} se confondent.

Le théorème de codage de source de Shannon stipule que, si l'on choisit le taux R_s et la distorsion D tels que $R_s > R_{SqS}(D)$, alors il existe un code en bloc (de longueur suffisamment grande) tel que son taux est inférieur ou égal à R_s et sa distorsion inférieure ou égale à D. La réciproque de ce théorème affirme que n'importe quel code de source de taux R_s et de distorsion D vérifie la relation : $R_s \geq R_{SqS}(D)$. Par conséquent, la fonction taux-distorsion apparaît comme la limite asymptotique théorique.

Notons que cette fonction vérifie une égalité qui sera utile par la suite :

$$2^{m(R_s - R_{SqS}(D))} = \frac{(q-1)^{mD}2^{mH_2(D)}}{q^{m-k}} = \frac{(q-1)^{mD}}{D^{mD}(1-D)^{m(1-D)}q^{m-k}}, \quad 0 \leq D \leq 1-\frac{1}{q}. \tag{1.33}$$

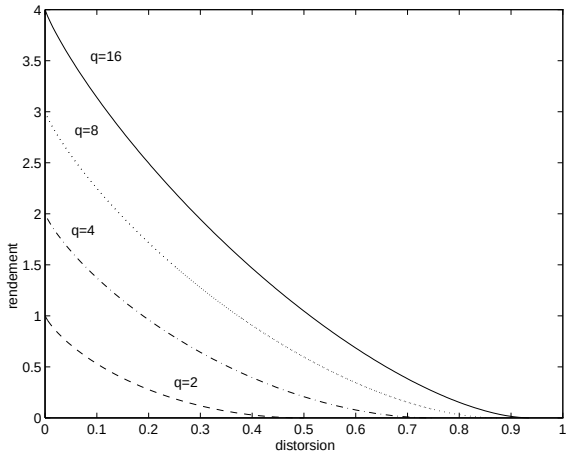


Fig. 1.12 – Fonction taux-distorsion $R_{SqS}(D)$ pour différentes valeurs de q.

1.2.3 Bornes sur la distorsion

À l'instar de la probabilité d'erreur, la distorsion peut être bornée en fonction de la longueur et de la dimension des codes utilisés, ce qui nous donnera des informations plus précises que les performances asymptotiques énoncées dans le théorème du codage de source de Shannon.

Borne inférieure

La borne inférieure sur la distorsion est déduite des résultats de l'annexe B utilisant les propriétés des cellules de Voronoï. La fonction f choisie pour appliquer les résultats de cette annexe est l'identité (qui est, bien sûr, strictement croissante).

Théorème 1.2.1 *La distorsion créée par l'utilisation d'un code linéaire q-aire de longueur n et de dimension k pour comprimer une SqS est minorée par*

$$D_{inf} = \frac{d}{m} - \frac{1}{mq^{m-k}} \sum_{i<d} (d-i)(q-1)^i C_m^i \tag{1.34}$$

où d est le plus grand entier tel que :

$$\sum_{i<d} (q-1)^i C_m^i \leq q^{m-k}.$$

Cette borne est valable pour les codes linéaires et non-linéaires. De surcroît, elle équivaut à la distorsion lorsque le code est parfait ou quasi-parfait.

L'annexe D présente la distorsion des codes parfaits et des codes quasi-parfaits les plus connus.

Il est possible d'interpréter la borne D_{inf} de la même manière que la borne P_{emi} : ces deux bornes sous-estiment le poids des chefs de classe. Le seul inconvénient de

cette borne est qu'elle ne peut garantir l'existence de codes l'atteignant car elle fait fi des contraintes algébriques que vérifient les chefs de classe. Nous avons retrouvé cette borne dans les travaux de Kerdock et Wolf [27] : ils considèrent un codeur qui, à chaque de mot de source, choisit au hasard parmi deux codes linéaires de même longueur m le code linéaire servant à coder le mot. La distorsion est alors une moyenne pondérée des distorsions produites par chaque code et peut être inférieure à la borne D_{inf} calculée pour une longueur m et une dimension qui est la moyenne pondérée des deux dimensions. Ce fait n'est pas surprenant car nous ne sommes pas dans les conditions dans lesquelles la borne a été établie. Kerdock et Wolf remarquent que ce codeur au comportement aléatoire engendre une distorsion égale à celle d'un code linéaire de longueur plus grande et de dimension la moyenne pondérée des deux dimensions. Cette distorsion est bien sûr supérieure à la borne inférieure D_{inf} calculée pour ces valeurs.

Par analogie avec les outils développés en codage de canal, nous nous intéressons au comportement asymptotique de cette borne de la distorsion, comportement qui ne figure pas dans [27].

Théorème 1.2.2 *Supposons le rendement R_s fixé. La borne D_{inf} est supérieure à la distorsion minimale ${R_{SqS}}^{-1}(R_s)$ donnée par le théorème de codage de source de Shannon ($D_{inf} > {R_{SqS}}^{-1}(R_s)$). La borne inférieure D_{inf} converge vers la fonction rendement-distorsion, i.e.*

$$\lim_{m\to\infty} D_{inf} = {R_{SqS}}^{-1}(R_s).$$

PREUVE
La preuve est donnée dans l'annexe C.3.

□

Ce théorème nous permet donc de conclure que cette borne est utile en raison de sa convergence et de sa dépendance vis-à-vis de la complexité. La convergence vers la fonction rendement-distorsion dans le cas binaire est illustrée sur la figure 1.13 où la fonction rendement-distorsion est représentée en trait plein : plus la longueur du code augmente, plus la borne se rapproche des meilleures performances optimales possibles.

Borne supérieure

Goblick [24], dont une partie des travaux a été reprise dans [64], a montré que dans le cas binaire les codes linéaires atteignent asymptotiquement l'OPTA. Plus précisément, D étant fixé, il existe un code linéaire binaire $\mathcal{C}$, de rendement $R > R_{SBS}(D)$ et de longueur m, tel que sa distorsion $D(\mathcal{C})$ vérifie :

$$D(\mathcal{C}) \leq D + \exp(-2^{m(R-R_{SBS}(D)+f(\epsilon))})$$

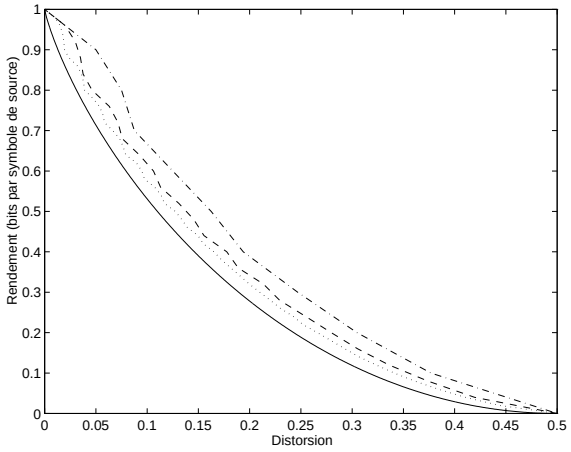


Fig. 1.13 – Borne inférieure D_{inf} pour différentes longueurs de code : m=10 (-.), m=25 (–), m=50 (..), OPTA en trait plein.

où ϵ est compris strictement entre 0 et D et où

$$f(\epsilon) = 2\epsilon \log_2 \frac{D-\epsilon}{1-D+\epsilon} + \frac{1}{m}\log_2\left(1 - \frac{1}{4m\epsilon^2}\right)$$

$f(\epsilon)$ est négligeable devant m lorsque ce dernier tend vers l'infini. La latitude sur le choix de ϵ permet d'optimiser la borne. La preuve de cette borne s'appuie sur le codage aléatoire et sur le choix séquentiel des générateurs du code : elle démontre donc qu'il est possible de construire une suite de bons vecteurs générateurs mais ne donne aucun moyen pratique de construire cette suite !

Cependant, nous pouvons utiliser les travaux de Cohen et Frankl [11] pour montrer l'existence de codes en bloc linéaires q-aires atteignant asymptotiquement la fonction rendement-distorsion. Nous allons démontrer ce résultat en reformulant les résultats de [11]. Nous obtiendrons au passage une borne supérieure de la distorsion.

Théorème 1.2.3 *Soit R_s un réel vérifiant $0 < R_s \leq \log_2 q$ et D un réel tel que $D > R^{-1}_{SqS}(R_s)$. Soit ϵ un réel positif. Il existe un code en bloc linéaire q-aire $\mathcal{C}$ de longueur suffisamment grande tel que :*

- *son rendement R vérifie $R_s < R < R_s + \epsilon$;*
- *sa distorsion notée $D(\mathcal{C})$ satisfait $D < D(\mathcal{C}) < D + \epsilon$.*

Autrement dit, il existe asymptotiquement de bons codes linéaires q-aires pour comprimer une SqS suivant le critère de distorsion de Hamming.

Preuve

Commençons la démonstration par rappeler une inégalité et énoncer un lemme.

Lemme 1.2.1 *Soient y un réel positif et x un réel compris entre 0 et 1, alors*

$$1 - xy \leq \exp(-xy) \leq (1-x)^y$$

Lemme 1.2.2 *(Cohen et Frankl, [11])*
Soient Y et Z deux sous-ensembles de F_q^m. Soit $\underline{x}$ un vecteur aléatoire de F_q^m, de distribution uniforme. Le cardinal moyen de l'intersection de Z et du translaté de Y par $\underline{x}$ est $q^{-m}|Y||Z|$, i.e.

$$E[|(\underline{x}+Y)\cap Z|] = q^{-m}|Y||Z|.$$

Notons, pour les besoins de la démonstration, $\mathcal{C}_j$ un code en bloc linéaire q-aire de dimension j et de longueur m. Nous allons construire de manière itérative des codes de dimension croissante en choisissant de bons vecteurs générateurs. Soit un réel $D \in [0, 1-1/q[$. Notons P_j la probabilité qu'un mot de source $\underline{u}$ soit à une distance du code $\mathcal{C}_j$ strictement supérieure à mD. Soit $\underline{x}$ un vecteur aléatoire de F_q^m de distribution uniforme, candidat pour être le $(j+1)^{\text{e}}$ vecteur générateur.

Corollaire 1 ***du lemme 1.2.2****(Cohen et Frankl [11], Goblick [24]) Il est possible de construire une suite de générateurs tels que*

$$P_{j+1} \leq P_j^2 \leq P_0^{2^j}.$$

Ce corollaire se prouve en appliquant le lemme 1.2.2 avec $Y = Z = \{\underline{u}, d_H(\underline{u}, \mathcal{C}_j) > mD\}$ ou directement en moyennant la probabilité de Z sur la distribution de probabilités du vecteur générateur aléatoire $\underline{x}$:

$$\begin{aligned} E[\Pr(d_H(\underline{u}, \mathcal{C}_j \cup (\underline{x}+\mathcal{C}_j)) > mD)] &= E[\Pr(d_H(\underline{u}, \mathcal{C}_j) > mD, d_H(\underline{u}, \underline{x}+\mathcal{C}_j) > mD)] \\ &= E[\Pr(d_H(\underline{u}, \mathcal{C}_j) > mD, d_H(\underline{u}-\underline{x}, \mathcal{C}_j) > mD)] \end{aligned}$$

Or $\underline{u}$ et $\underline{u}-\underline{x}$ sont deux vecteurs aléatoires indépendants donc

$$E[\Pr(d_H(\underline{u}, \mathcal{C}_j \cup (\underline{x}+\mathcal{C}_j)) > mD)] = \Pr(d_H(\underline{u}, \mathcal{C}_j) > mD)^2.$$

Par conséquent, il est possible de choisir un vecteur $\underline{x}$ tel que $P_{j+1} \leq P_j^2$. À partir de ce corollaire, on peut montrer l'existence de codes en bloc linéaires binaires atteignant l'OPTA. Il n'est pas suffisant dans le cas q-aire mais sera néanmoins utile à la démonstration. Nous allons donc améliorer ce résultat, cependant il ne sera pas valable pour toutes les dimensions.

Lemme 1.2.3 *(Cohen et Frankl, [11])*
Lorsque m est assez grand, il est possible de choisir j_1 vecteurs générateurs ($j_1 \leq m$) vérifiant

$$\begin{array}{cccccccc} & P_{j_1} & \leq & 1-(qm)^{-1} & \leq & P_{j_1-1} & \leq & P_0^{q^{j_1-1}\exp(-1/2)} \\ 0 \leq j \leq j_1-1 & & & P_{j+1} & \leq & P_j^{q(1-(2m)^{-1})} & & \end{array} \tag{1.35}$$

Démontrons ce lemme. Une fois le $(j+1)^{\text{e}}$ générateur choisi, $\mathcal{C}_{j+1}$ est la réunion de q translatés du code $\mathcal{C}_j$. Moyennant alors sur l'ensemble des vecteurs de F_q^m la probabilité P_{j+1}, on écrit

$$E[P_{j+1}] = E[\Pr(d_H(\underline{u}, \mathcal{C}_{j+1}) > mD)] = E[\Pr(d_H(\underline{u}, \cup_{\lambda \in F_q}(\lambda \underline{x} + \mathcal{C}_j)) > mD)].$$

Posons cette fois-ci $Z = \{\underline{u}, d_H(\underline{u}, \mathcal{C}_j) \leq mD\}$. Alors $P_j = 1 - q^{-m}|Z|$, donc

$$E[\Pr(d_H(\underline{u}, \mathcal{C}_{j+1}) > mD)] = E[1 - q^{-m}|\cup_{\lambda \in F_q}(\lambda \underline{x} + Z)|] \tag{1.36}$$

En appliquant le principe d'inclusion-exclusion et le lemme 1.2.2, nous obtenons

$$|\cup_{\lambda \in F_q}(\lambda \underline{x} + Z)| \geq \sum_{\lambda \in F_q} |\lambda \underline{x} + Z| - \sum_{\substack{(\lambda, \lambda') \in F_q^2 \\ \lambda \neq \lambda'}} |(\lambda \underline{x} + Z) \cap (\lambda' \underline{x} + Z)|$$

$$E\left[\sum_{\substack{(\lambda, \lambda') \in F_q^2 \\ \lambda \neq \lambda'}} |(\lambda \underline{x} + Z) \cap (\lambda' \underline{x} + Z)|\right] = q^{-m} C_q^2 |Z|^2.$$

Combinons ces deux résultats avec l'équation (1.36) et utilisons l'inégalité indiquée dans le lemme 1.2.1

$$\begin{aligned} E[\Pr(d_H(\underline{u}, \mathcal{C}_{j+1}) > mD)] &\leq 1 - q\frac{|Z|}{q^n} + C_q^2 \frac{|Z|^2}{q^{2m}} \\ &\leq \left(1 - \frac{|Z|}{q^m}\right)^{q(1-q|Z|/(2q^m))} \\ &\leq P_j^{q(1-q(1-P_j)/2)}. \end{aligned}$$

P_j doit être supérieure à $1 - 2/q$. Si cette condition est vérifiée, alors il est possible de trouver un vecteur $\underline{x}$ tel que

$$P_{j+1} \leq P_j^{q(1-q(1-P_j)/2)}.$$

Si la probabilité P_j est telle que $P_j > 1 - 1/(qm)$ alors

$$P_{j+1} \leq P_j^{q(1-(2m)^{-1})}.$$

Remarquons qu'en vertu du théorème A.2.1, il est toujours possible de choisir m assez grand pour que P_0 vérifie $P_0 > 1 - 1/(2m)$. Par conséquent, nous pouvons itérer l'inégalité précédente à partir du rang 0, obtenant alors

$$\forall j \in \{0, \cdots, j_1\}, \quad P_j \leq P_0^{(q(1-(2m)^{-1}))^j}$$

où j_1 est le plus petit entier vérifiant $P_{j_1} \leq 1 - 1/(qm)$. Simplifions encore l'inégalité en remarquant que $(1 - (2m)^{-1})^j \geq e^{-1/2}$ par application du lemme 1.2.1. Il existe donc un entier j_1 tel que

$$1 - (qm)^{-1} \leq P_{j_1 - 1} \leq P_0^{q^{j_1 - 1} \exp(-1/2)} \tag{1.37}$$

Le lemme 1.2.3 est donc démontré. Grâce à ce lemme, nous construisons un code de dimension j_1. En appliquant le corollaire 1, nous pouvons choisir j_2 générateurs supplémentaires de sorte que $P_{j_1+j_2} \leq q^{-m}$ Il suffit, pour cela, de choisir j_2 tel que

$$(1-(qm)^{-1})^{2^{j_2}} \leq q^{-m},$$

ce qui implique que $j_2 = 2\log_2 m + O(1)$: j_2/m est donc négligeable devant 1. Relions maintenant $P_{j_1+j_2}$ au rendement R du code $\mathcal{C}_{j_1+j_2}$.

$$P_{j_1+j_2} \leq P_{j_1}^{2^{j_2}} \leq P_0^{2^{j_2} q^{j_1} \exp(-1/2)}$$

Utilisons les résultats de l'annexe 8A de [64] :

$$1 - P_0 \geq 2^{-m(R_{SqS}(D)+o(1))}$$

En appliquant une fois de plus le lemme 1.2.1, il vient

$$P_{j_1+j_2} \leq \exp(-\exp(-1/2)2^{-m(R-R_{SqS}(D)-\frac{j_2}{m}(\log_2 q-1)+o(1))})$$

Puisque j_2/m est donc négligeable devant 1, on déduit de l'inégalité précédente que $P_{j_1+j_2}$ tend vers 0 si le rendement R du code est supérieur à $R_{SqS}(D)$. La distorsion du code $\mathcal{C}_{j_1+j_2}$ est donc bornée par $D + P_{j_1+j_2}$, ce qui achève de démontrer l'existence de bons codes en bloc linéaires q-aires. Remarquons au passage que le codeur de source peut se contenter de bien savoir coder les mots de source dans les sphères de rayon D centrées sur les mots de code, ceux en dehors de ces sphères pouvant être arbitrairement codés car étant en faible proportion : la démonstration prouve non seulement que les codes linéaires permettent d'atteindre les performances asymptotiques mais que la distorsion maximale sera bornée par D.

□

Nous disposons donc de deux bornes nous permettant de juger des performances d'un code vis-à-vis des autres codes partageant même longueur et même dimension.

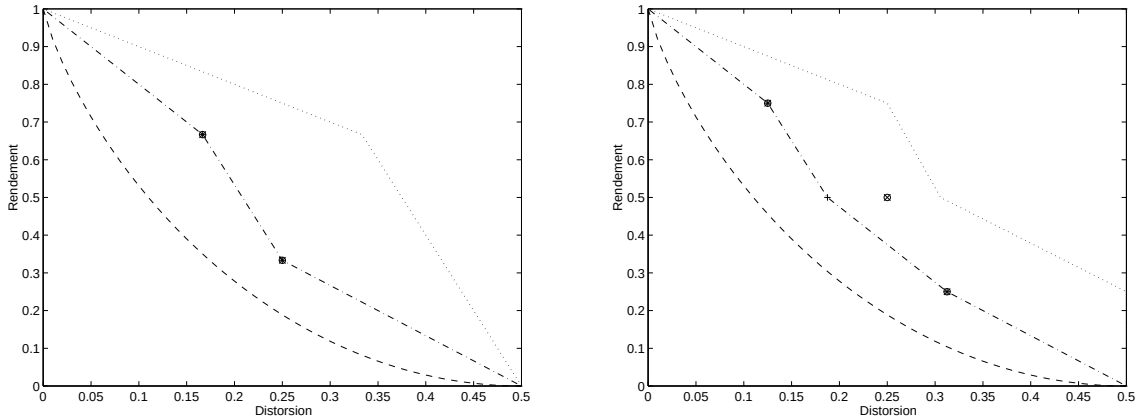
1.2.4 À la recherche de bons codes

Recherche exhaustive

À l'instar de la recherche de codes de canal linéaires binaires, nous recherchons les meilleurs codes de source linéaires binaires de manière exhaustive. Deux codes équivalents produisant la même distorsion, la recherche systématique peut être allégée. Nous l'avons menée jusqu'à une longueur égale à 10. Les figures 1.14-1.17 montrent la distorsion des meilleurs codes trouvés[7]. Elle est représentée par des croix (+), les bornes supérieure par des points (..) et inférieure en traits mixtes (-.). Les cercles (o) et les

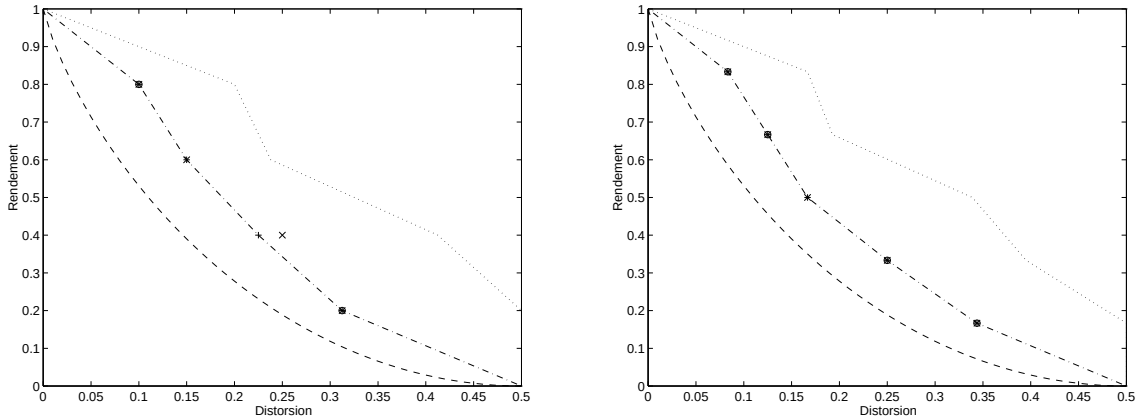
[7] Les codes linéaires trouvés ont été réutilisés par Zahir Azami pour initialiser l'algorithme JOB [69].

croix obliques (x) présents sur ces figures seront expliqués ultérieurement. La fonction rendement-distorsion R_{SBS}, déterminant les performances optimales indiquées par le théorème de codage de source de Shannon, est représentée par des tirets (--). Les codes atteignent le plus souvent la borne inférieure D_{inf}, cependant ils demeurent assez loin des meilleures performances possibles en raison de leur petite longueur.



(a) : $m = 3$ (b) : $m = 4$

Fig. 1.14 – Meilleurs codes linéaires pour le codage de SBS, $m = 3, 4$.

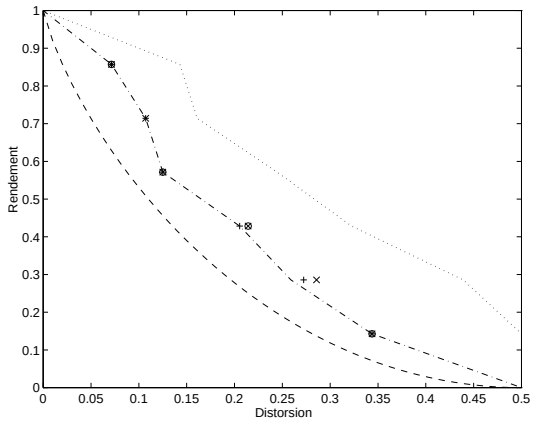


(a) : $m = 5$ (b) : $m = 6$

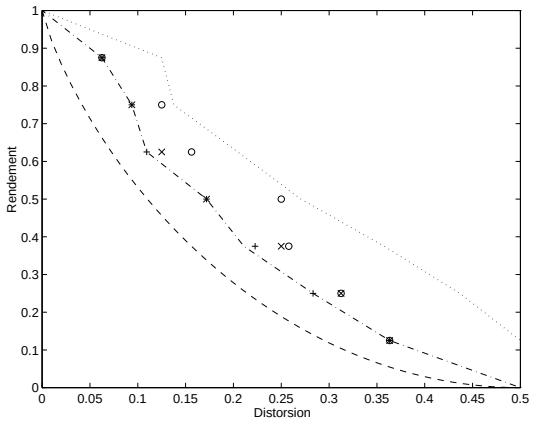
Fig. 1.15 – Meilleurs codes linéaires pour le codage de SBS, $m = 5, 6$.

Codes cycliques

Les performances des meilleurs codes linéaires trouvés précédemment par recherche exhaustive étant éloignées de la fonction taux-distorsion, nous allons chercher à augmenter la longueur des codes. La recherche exhaustive devenant trop coûteuse, nous

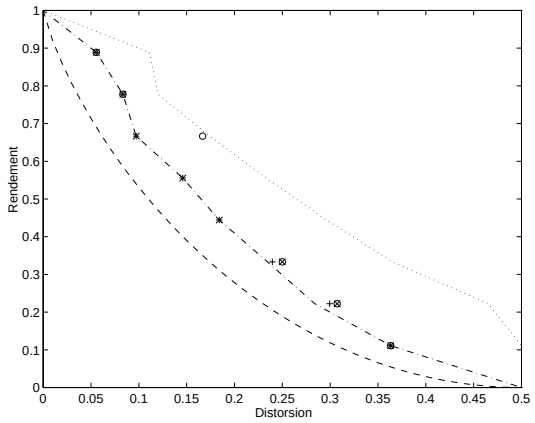


(a) : $m = 7$

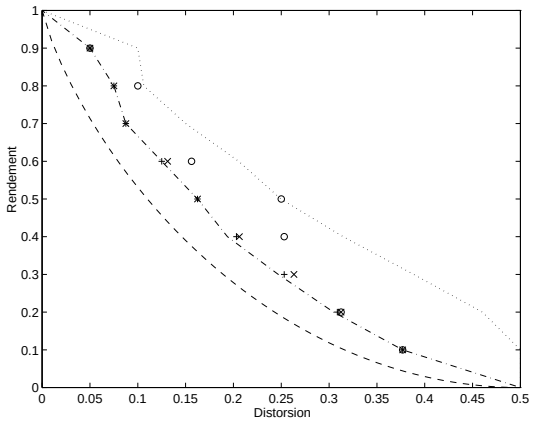


(b) : $m = 8$

FIG. 1.16 – Meilleurs codes linéaires pour le codage de SBS, $m = 7, 8$.



(a) : $m = 9$

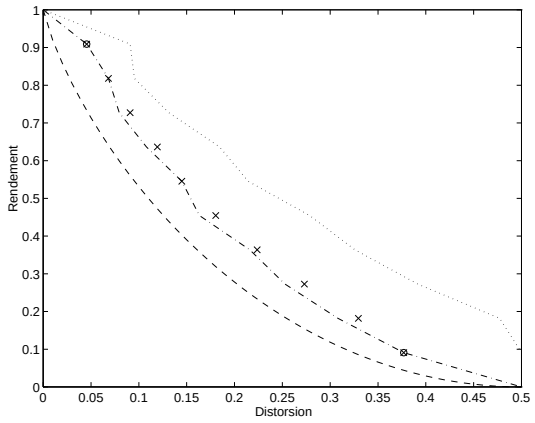


(b) : $m = 6$

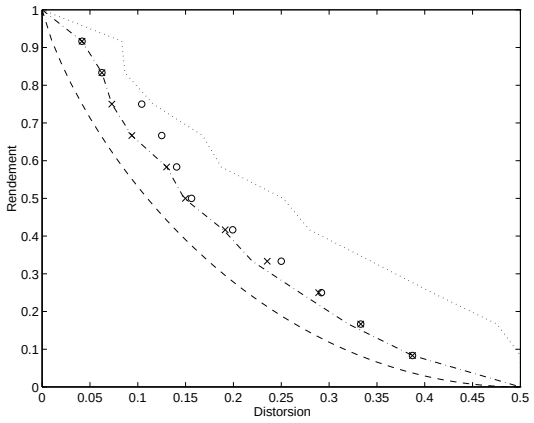
FIG. 1.17 – Meilleurs codes linéaires pour le codage de SBS, $m = 9, 10$.

allons nous restreindre aux codes cycliques [8] ayant constaté que les codes cycliques de longueur impaire ont presque les mêmes performances que les meilleurs codes linéaires de même longueur et de même rendement. La distorsion des codes cycliques est représentée sur les figures 1.14-1.24 par des cercles, pour des longueurs allant jusqu'à $m = 23$ (la liste exhaustive de ces codes est donnée dans l'ouvrage de McWilliams et Sloane [43]).

[8] le lecteur souhaitant connaître la définition précise des codes cycliques binaires est invité à se reporter à l'ouvrage de McWilliams et Sloane [43]. Précisons tout de même que les codes cycliques sont des codes linéaires bien particuliers. Au chapitre 5, nous expliquons la construction des codes cycliques sur le corps des réels, construction analogue à celle des codes cycliques sur le corps F_2.

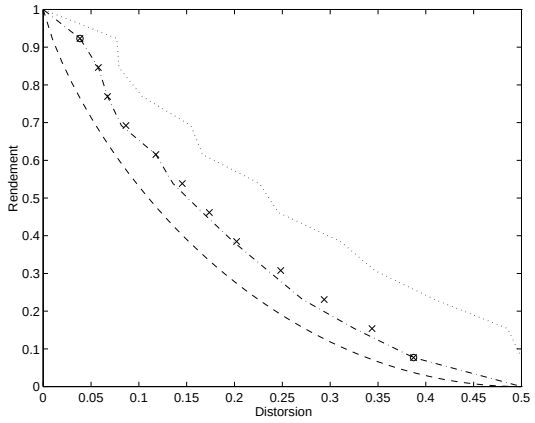


(a) : $m = 11$

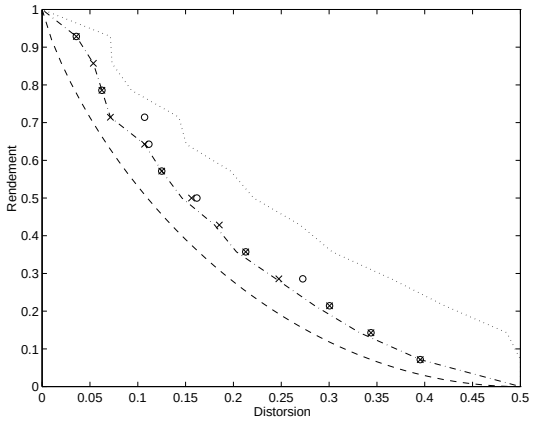


(b) : $m = 12$

Fig. 1.18 – Codage linéaire de SBS, $m = 11, 12$.

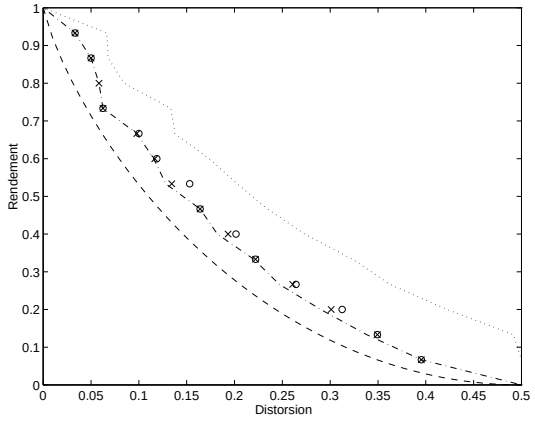


(a) : $m = 13$

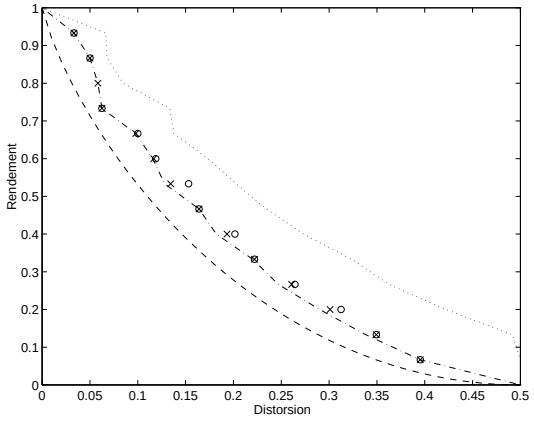


(b) : $m = 14$

Fig. 1.19 – Codage linéaire de SBS, $m = 13, 14$.



(a) : $m = 15$



(b) : $m = 16$

Fig. 1.20 – Codage linéaire de SBS, $m = 15, 16$.

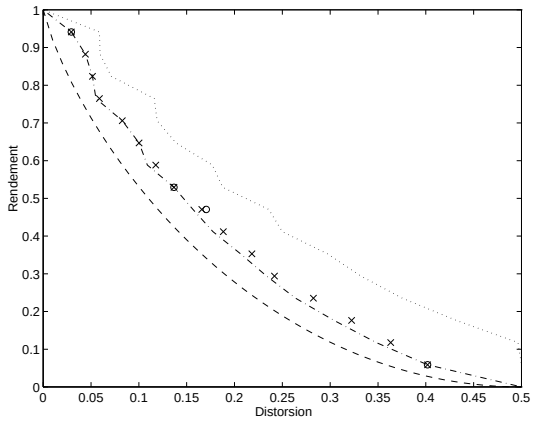
Un algorithme de construction des bons codes de source linéaires

En raison du coût exponentiel de la recherche exhaustive, l'existence d'un algorithme de construction de bons codes se révèle fortement intéressante. De tels algorithmes existent dans le cas du codage de canal [43] : ils se focalisent, pour la plupart, sur l'augmentation de la distance minimale des codes [9]. Dans le cas du codage de source, nous n'avons trouvé aucun algorithme dans la littérature. Nous proposons l'algorithme suivant :

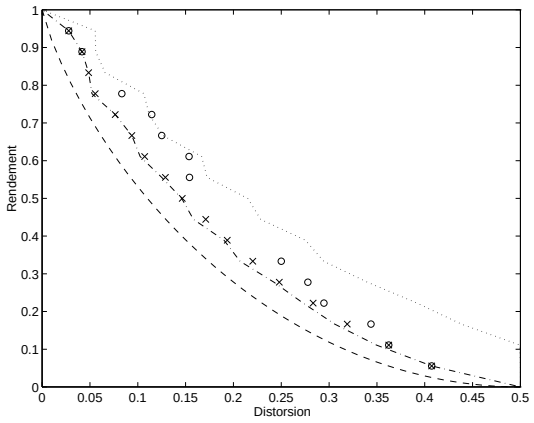
- nous calculons tout d'abord le tableau standard de décodage d'une matrice de parité ;
- nous créons à partir de ce code un nouveau code linéaire en lui ajoutant un nouveau générateur choisi parmi les chefs de classe de poids maximum. Choisir le chef de classe qui minimiserait la distorsion du nouveau code nécessiterait d'évaluer exactement la nouvelle distorsion pour tous les candidats, ce qui peut s'avérer complexe. Par conséquent nous choisissons le dernier chef de classe de poids maximum trouvé lors du calcul du tableau standard de décodage du premier code.

Le code obtenu est de même longueur et de dimension $k+1$. Bien sûr, cet algorithme peut être itéré afin d'obtenir des codes à haut rendement. Nous avons programmé cet algorithme en choisissant comme codes initiaux les codes cycliques. Les performances des meilleurs codes trouvés par cet algorithme sont représentées par des croix obliques (x) sur les figures 1.14-1.21 Ces codes restent proches de la borne inférieure D_{inf} : en général, leurs distorsions sont inférieures à celles des codes cycliques. Cependant, notre algorithme nécessite toujours le calcul du tableau standard de décodage, ce qui impose de limiter $m-k$: le code servant à initialiser l'algorithme ne doit pas être de trop petite dimension. Nous sommes donc limités à une longueur m inférieure à 18. Pour une longueur m égale à 23, nous avons initialisé l'algorithme avec le code de Golay de dimension 12 [43] : les résultats de notre algorithme sont portés sur la figure 1.24.

[9] le critère de distance minimale n'est pas *a priori* le plus pertinent en terme de probabilité d'erreur comme l'a montré Battail [2].

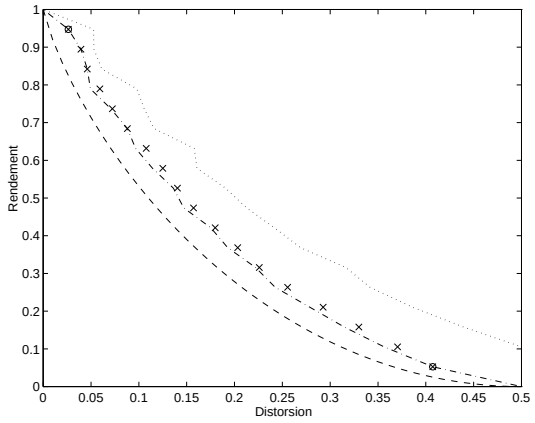


(a) : $m = 17$

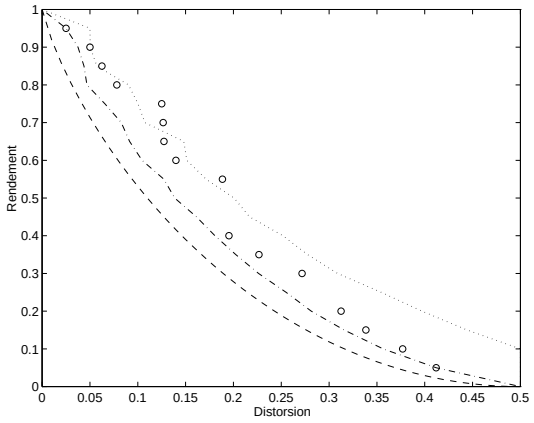


(b) : $m = 18$

Fig. 1.21 – Codage linéaire de SBS, $m = 17, 18$.

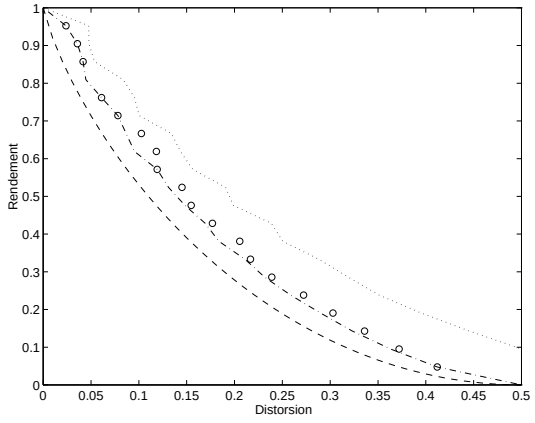


(a) : $m = 19$

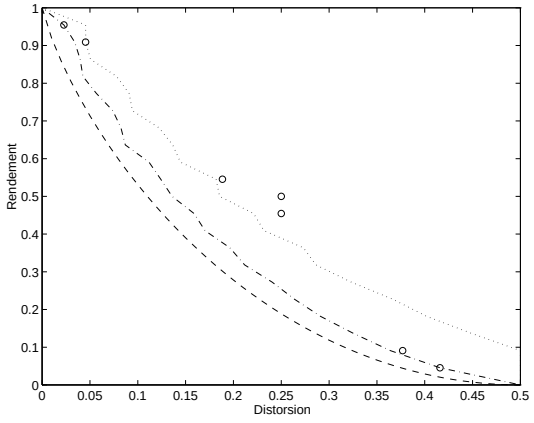


(b) : $m = 20$

Fig. 1.22 – Codage linéaire de SBS, $m = 19, 20$.



(a) : $m = 21$



(b) : $m = 22$

Fig. 1.23 – Codage linéaire de SBS, $m = 21, 22$.

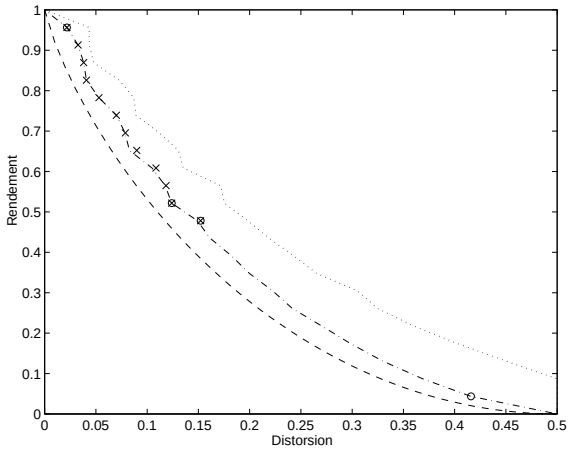


Fig. 1.24 – Codage linéaire de source $m = 23$.

1.3 Conclusions

Ayant rappelé tout d'abord des notions bien connues sur l'utilisation des codes en bloc linéaires dans le cadre de la théorie du codage de canal, nous avons mis en évidence la dualité existante entre cette dernière et celle du codage de source : le décodeur de canal et le codeur de source, appliqués respectivement au canal q-aire symétrique et à la source q-aire symétrique, accomplissent la même tâche. Il en est de même pour le codeur de canal et le décodeur de source. La symétrie est telle entre ces deux domaines que la capacité et la fonction rendement-distorsion ont même expression.

Des bornes inférieures précises ont été déterminées pour évaluer les performances des codes en blocs dans ces deux domaines par des outils similaires. Nous avons également rappelé des bornes supérieures issues de la technique du codage aléatoire. Nous avons alors montré en utilisant les travaux de Frankl et Cohen que les codes linéaires q-aires permettent d'atteindre asymptotiquement la plus petite distorsion possible à rendement fixé.

Nous avons enfin recherché de « bons » codes : cette recherche, déjà menée dans le cadre du codage de canal [50, 43], nous a permis de voir que, pour les petites longueurs, les meilleurs codes binaires pour la compression de SBS sont aussi les meilleurs pour la correction d'erreurs. Hélas, nous n'avons pu préciser formellement ce lien. Nous avons proposé un algorithme simple pour construire des codes de rendement croissant et de longueur fixe aux résultats satisfaisants en compression de source.

Cette étude préliminaire terminée, nous allons tout naturellement nous intéresser à la combinaison des deux domaines afin de nous approcher d'un système réel : tournons la page !

Chapitre 2

Codage conjoint source binaire symétrique - canal binaire symétrique

2.1 Introduction

Nous ne nous sommes intéressés au chapitre précédent qu'à une partie de la chaîne de transmission, considérant exclusivement le codage de canal ou le codage de source. Un système réel de communication incorpore bien évidemment ces deux tâches. Nous supposerons dans cette étude que nous devons transmettre la sortie d'une source binaire symétrique sur un canal binaire symétrique. La question qui se pose alors est, d'une part, de savoir comment combiner habilement codage de source et codage de canal et, d'autre part, de déterminer les performances en termes de distorsion et de rendement que nous sommes en droit d'attendre. Historiquement, la réponse apportée par le théorème de codage conjoint de Shannon à cette deuxième question a déterminé la réponse à la première, conduisant ainsi à une conception séparée du codeur de source et de canal, ce qui suppose des systèmes complexes et dont le délai de traitement peut être grand. Une approche conjointe du codage vise à garantir, pour une complexité moindre, des performances meilleures ou équivalentes à des systèmes conçus séparément. En contrepartie, les techniques de codage conjoint sont potentiellement plus complexes à étudier.

Le but de ce chapitre est d'étudier la conception d'un système conjoint SBS-CBS en mesurant les performances au moyen de la distance de Hamming [1], critère pertinent en codage de canal et qui, par dualité, est pertinent pour la compression de SBS. Ce choix de source et de canal présente l'avantage d'être général et flexible. En effet, depuis l'avènement des communications numériques, toute source est échantillonnée et quantifiée : le modèle que nous avons retenu vient donc tout naturellement se greffer

[1] Zahir-Azami [69] a accompli un travail similaire en utilisant la distance euclidienne en partant des codes trouvés au chapitre précédent.

après cette étape. Un codage conjoint CBS-SBS efficace permettrait donc de faire varier le débit binaire émis en fonction de l'état du canal ou des ressources disponibles en terme de bande passante, le système devenant ainsi aisément flexible.

Nous allons donc tout d'abord présenter le système séparé, qui réutilise les notions introduites dans le précédent chapitre et qui est pris comme système de référence. Puis nous donnerons les meilleures performances théoriquement possibles en rappelant précisément le théorème du codage conjoint. Nous présenterons ensuite les performances pratiques de ce système et exposerons un nouveau système dit « source ou canal » plus simple que le système séparé et aux performances meilleures. Nous mènerons alors une petite étude théorique à son sujet. Enfin nous utiliserons l'algorithme de Lloyd pour améliorer ce système. Reprenant les codes de petite longueur trouvés au chapitre précédent, les systèmes étudiés présentent de très faibles délais de traitements (une dizaine de bits).

2.2 Présentation du système séparé

Si nous reprenons directement les codeurs et décodeurs décrits dans les deux premières sections du premier chapitre, nous aboutissons au système représenté sur la figure 2.1 : la source binaire symétrique émet un mot $\underline{u} = (u_1, \cdots, u_m)$ qui est comprimé par le codeur de source en un mot d'information $\underline{\imath} = (\imath_1, \cdots, \imath_k)$. Le codeur de canal calcule à partir de ce mot le mot de canal $\underline{x} = (x_1, \cdots, x_n)$. Le codeur global associe donc à un mot de source $\underline{u}$ un mot de canal $\underline{x}$. Le canal binaire symétrique ajoute du bruit à ce mot : le décodeur de canal reçoit donc le mot bruité $\underline{y} = (y_1, \cdots, y_n)$ à partir duquel il calcule une estimée du mot d'information $\underline{\imath}$ notée $\underline{\hat{\imath}} = (\hat{\imath}_1, \cdots, \hat{\imath}_k)$. Enfin, le décodeur de source fournit au destinataire une estimée de $\underline{u}$, $\underline{\hat{u}} = (\hat{u}_1, \cdots, \hat{u}_m)$, calculée à partir de $\underline{\hat{\imath}}$. Le décodeur global associe donc au mot $\underline{y}$ le mot de source reconstruit $\underline{\hat{u}}$.

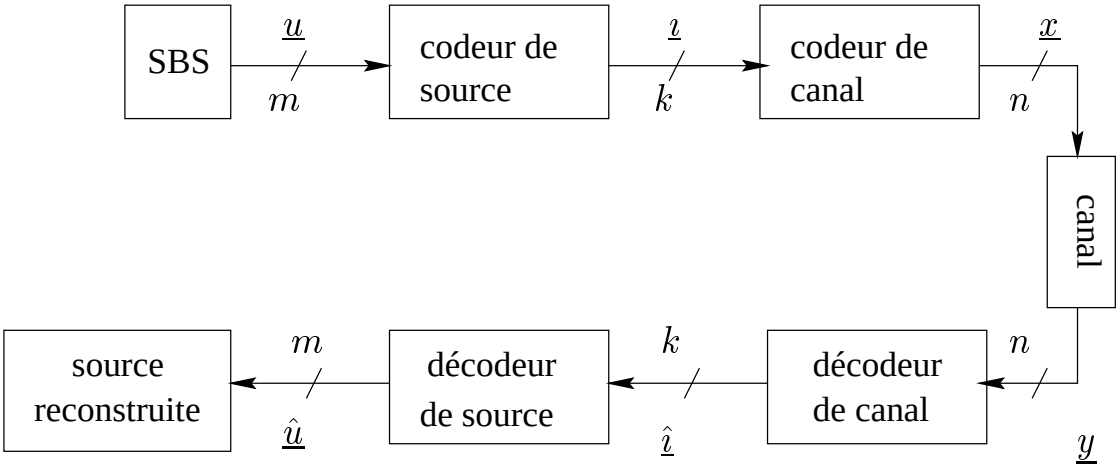


Fig. 2.1 – Codage séparé SBS-CBS.

2.3 Bornes : OPTA et schémas triviaux

Définissons tout d'abord le **rendement global** R du système séparé :

$$R = \frac{n}{m} \text{ symboles de canal par symbole de source.} \tag{2.1}$$

Ce rendement est donc indépendant des mécanismes internes du codeur global.

La distorsion globale par bit D entre la source originale et la source reconstruite est donnée par

$$D = \frac{1}{m} E[d_H(\underline{U}, \hat{\underline{U}})]. \tag{2.2}$$

Or, le théorème de traitement des données [13] affirme que

$$I(\underline{U}, \hat{\underline{U}}) \leq I(\underline{X}, \underline{Y}),$$

et les définitions de la capacité et de la fonction rendement-distorsion impliquent que

$$\begin{aligned} I(\underline{X}, \underline{Y}) &\leq nC_{CBS}(p) \\ mR_{SBS}(D) &\leq I(\underline{U}, \hat{\underline{U}}), \end{aligned}$$

où p est la paramètre du CBS. Combinant alors le théorème de traitement des données à ces inégalités, nous obtenons

$$R \geq \frac{R_{SBS}(D)}{C_{CBS}(p)} = \frac{1 - H_2(D)}{1 - H_2(p)}. \tag{2.3}$$

Le **théorème du codage conjoint**, formulé par Shannon [41], établit qu'à distorsion D et rendement R donnés tels que $R > R_{SBS}(D)/C_{CBS}(p)$, il existe un code conjoint de rendement inférieur à R et de distorsion inférieure à D. L'équation $R = (1 - H_2(D))/(1 - H_2(p))$ représente donc les meilleures performances théoriquement possibles d'un schéma de codage conjoint.

Ce théorème est établi en appliquant séparément les théorèmes de codage de source et de canal : en choisissant des longueurs m et n suffisamment grandes, il est possible de trouver un codeur et un décodeur de source qui créent une distorsion proche de D et un codeur et un décodeur de canal dont la probabilité d'erreur est arbitrairement petite. Lorsque le décodeur de canal se trompe, ce qui arrive très rarement, la distorsion restera bornée [2]. Cette preuve suggère donc une conception séparée du codeur de source et du codeur de canal, cette séparation se faisant au détriment de la simplicité.

Il est d'ores et déjà possible de donner des bornes supérieures de performances en utilisant des codes linéaires triviaux (c'est-à-dire dont la matrice Par est nulle). Dans le cas d'un rendement global R inférieur à 1, le code de canal choisi est le code identité ($k = n$), appelé également code universel, et le code de source est le code trivial

[2] Une démonstration plus rigoureuse de ce théorème est donnée dans [41].

qui supprime $m-n$ des m composantes du mot de source $\underline{u}$. Un tel système est représenté sur la figure 2.2. Supposons que ces composantes soient les $m-n$ dernières. Les n premières composantes de $\underline{\hat{u}}$ sont égales à celles de $\underline{y}$ et les autres sont tirées au hasard (suivant une loi uniforme puisque la source est symétrique) ou fixées arbitrairement à une certaine valeur (0, par exemple). Les n premières composantes ont donc une probabilité p d'être différentes des n premières du mot de source $\underline{u}$ alors que chacune des autres a une probabilité d'être différente égale à $1/2$. La distorsion est $D = \frac{np+0.5(m-n)}{m} = 0.5 - R(0.5-p)$.

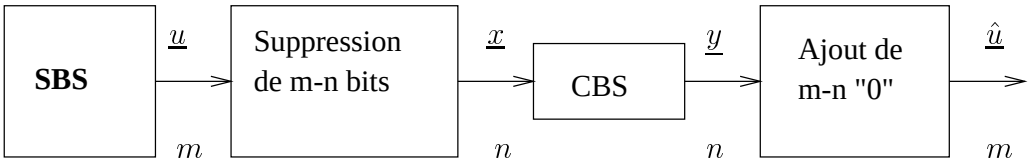


Fig. 2.2 – Schéma trivial de codage conjoint SBS-CBS, $R < 1$.

Lorsque le rendement global R est supérieur à 1, le système trivial consiste à choisir un code de source universel ($k = m$) et à ajouter $n-m$ zéros au mot de source (c'est-à-dire ne pas tirer profit des bits de parité). Ce système est représenté sur la figure 2.3 dans le cas $R > 1$. Lorsque le rendement global R vaut 1, la source est directement émise sur le canal (ce qui correspond au choix d'un code universel). La distorsion de ces deux systèmes est $D = p$.

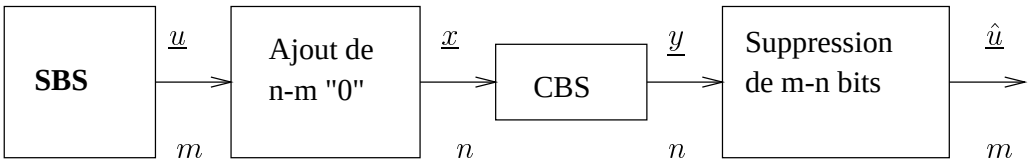


Fig. 2.3 – Schéma trivial de codage conjoint SBS-CBS, $R > 1$.

Les performances des codes triviaux, indépendantes de la longueur du code considérée, et l'OPTA sont représentées (respectivement en pointillés et en trait continu) sur la figure 2.4. Remarquons que dans le cas $R = 1$, nous avons atteint les meilleures performances théoriquement possibles alors qu'une conception de système s'appuyant sur la preuve du théorème du codage conjoint nous aurait poussé à concevoir un codeur de source créant une distorsion égale à p et un codeur de canal assez puissant pour pratiquement corriger toutes les erreurs, ce qui peut s'avérer difficile si p n'est pas petit devant 1. Le choix d'un code universel est donc le meilleur possible : si nous choisissons d'émettre sur le canal les mots d'un code non universel, la distorsion obtenue sera supérieure ou égale à p. Lorsque le rendement global R est proche de 1, il sera donc difficile de battre les codes triviaux.

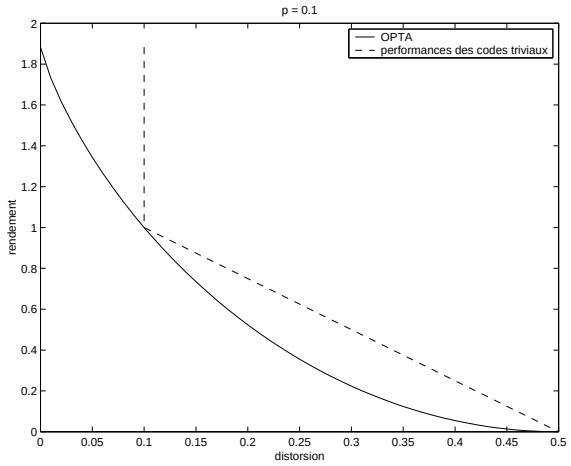


Fig. 2.4 – OPTA et performances des codes triviaux dans le cas du codage conjoint SBS-CBS, $p = 0.1$.

2.4 Performances de différents systèmes

L'écart entre les meilleures performances théoriquement possibles et celles des codes triviaux conduit naturellement à rechercher des systèmes plus efficaces. Nous allons donc examiner deux systèmes :

- le système tandem ;
- le système source ou canal.

2.4.1 Schéma tandem

Le système tandem représenté sur la figure 2.1 s'inspire de la preuve du théorème du codage conjoint : codage de source et codage de canal sont réalisés séparément par deux codes linéaires $\mathcal{C}_s$ et $\mathcal{C}_c$. Encore faut-il choisir judicieusement ces codes. Nous avons retenu la stratégie suivante : nous souhaitons trouver un système tandem présentant une distorsion totale égale à D fixé. Nous recherchons alors, parmi les codes trouvés au paragraphe 1.2.4, le code de source $\mathcal{C}_s$ dont la distorsion est la plus proche de D. Une fois fois $\mathcal{C}_s$ choisi, nous recherchons alors parmi les codes trouvés au paragraphe 1.1.6 le code $\mathcal{C}_c$ de plus haut rendement et dont la probabilité d'erreur par bit P_{eb} est la plus proche d'une valeur dépendant de D et petite par rapport à cette dernière, espérant ainsi que le codeur de canal trouvé ne dégradera pas trop les performances du codeur de source. Il est difficile de donner une expression analytique des performances d'un tel système. Nous avons donc eu recours à des simulations. Nous choisissons le code $\mathcal{C}_c$ de plus haut rendement dont la probabilité d'erreur est proche de $D/100$ ($P_{eb} \approx D/100$). La recherche des codes est menée lorsque le paramètre p vaut 5.10^{-4}. Puis nous augmentons p afin d'étudier la robustesse du système séparé : p prend alors les valeurs 10^{-3}, 5.10^{-3}, 10^{-2}, 5.10^{-2}, 10^{-1}. La figure 2.5 montre la dis-

torsion de trois systèmes séparés (correspondant à trois rendements globaux différents) en fonction de la probabilité de transition du CBS. L'OPTA pour chacun de ces trois systèmes est également représentée : les systèmes séparés restent loin des meilleures performances théoriquement possibles en raison des petites longueurs de code. Ces systèmes ne montrent pas de robustesse particulière : leurs courbes de distorsion semblent parallèles à celles représentant l'OPTA.

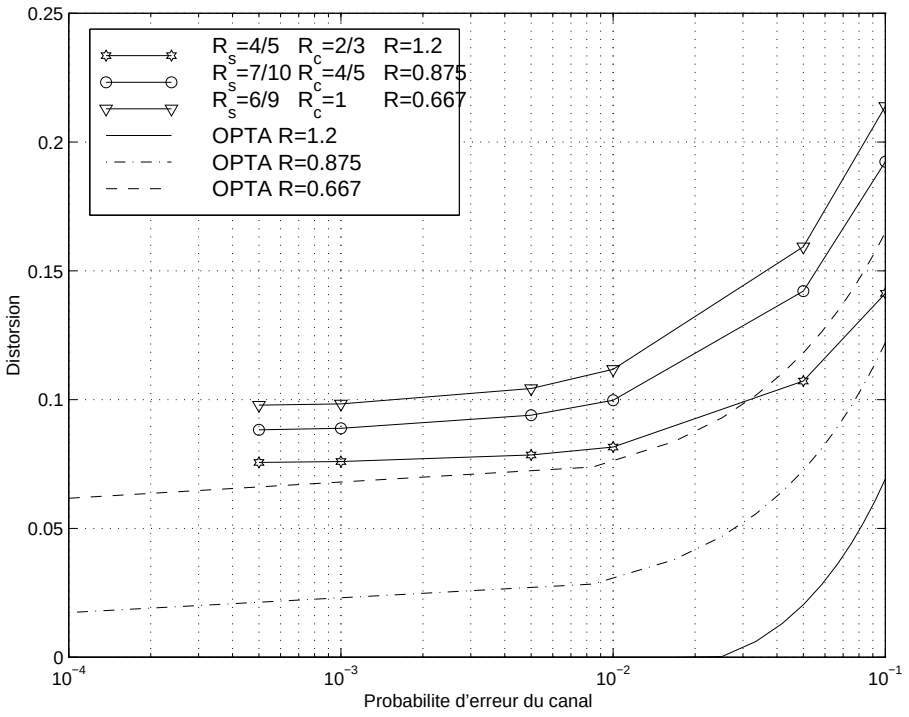


Fig. 2.5 – Performances du système tandem (codes linéaires de longueur ≤ 10).

2.4.2 Système « source ou canal »

Présentation du système

Au lieu de concevoir un système séparé, nous pouvons simplement choisir de supprimer le codeur de source ou de canal suivant la valeur du rendement global R par rapport à 1 :

- lorsque $R < 1$, un code linéaire binaire réalise le codage de source, la sortie de ce codeur est émise directement sur le canal (le code de canal est universel) ;
- lorsque $R > 1$, les mots de source alimentent l'entrée du codeur de canal. Le code de canal est choisi parmi les codes linéaires binaires (le code de source est universel) ;
- lorsque $R = 1$, les codes de source et de canal sont tous les deux universels.

Ce système appelé « source ou canal » en raison de la suppression d'une étape de codage est représenté sur la figure 2.6.

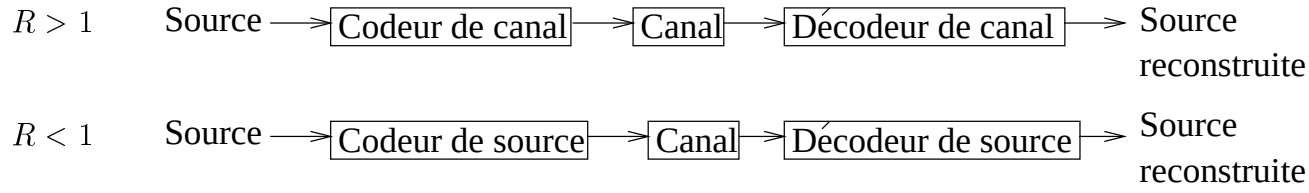


FIG. 2.6 – Schéma du système source ou canal.

Étude théorique du système

Outre le cas trivial $R = 1$, les théorèmes de codage de canal et de codage de source montrent qu'un tel système permet d'atteindre asymptotiquement l'OPTA lorsque :

- $R = 1/R_c > 1/(1-H_2(p))$ (la distorsion totale est alors asymptotiquement nulle : il existe donc de bons codes linéaires permettant une communication fiable) ;
- $R = R_s > 1 - H_2(D)$ ($p = 0$, c'est-à-dire lorsque le canal binaire symétrique est parfait : il existe alors de bons codes linéaires pour comprimer la source binaire symétrique).

Comment se comporte ce système lorsque le rendement global ne permet pas une communication fiable et que le canal n'est pas parfait, c'est-à-dire lorsque $1-H_2(D) < R < 1/(1-H_2(p))$? Nous pouvons apporter un début de réponse en remarquant que la borne inférieure P_{esi} est, dans le cas $R > 1$, une borne inférieure de la distorsion totale D de notre système. De plus, en vertu des théorèmes 1.1.4 et 1.1.5, cette borne converge vers l'OPTA du codage conjoint SBS-CBS.

Qu'en est-il alors pour $R < 1$? La borne D_{inf}, hélas, suppose le canal parfait et n'est donc pas applicable lorsque p est non nul. Qu'à cela ne tienne, nous pouvons reprendre l'idée nous ayant menés à la formule de P_{esi} (1.24) pour développer une borne de la distorsion totale D du système source ou canal lorsque $R < 1$.

Théorème 2.4.1 *La distorsion globale D du système source ou canal lorsque le rendement global R est inférieur à 1 est minorée par D_{sci} :*

$$D_{sci} = \frac{1}{m2^{m-n}} \left(\sum_{a=0}^{n} p^a (1-p)^{n-a} \left[\sum_{l=w_{s_{a-1}}+1}^{w_{s_a}} l C_m^l + (1+w_{s_a})(s_a 2^{m-n} - \sum_{l=0}^{w_{s_a}} C_m^l) \frac{1-2p}{1-p} \right] \right) \tag{2.4}$$

où le coefficient s_a est la somme des a premiers coefficients binomiaux

$$s_a = \sum_{l=0}^{a} C_n^l$$

et où w_i est l'entier défini par

$$w_i = \arg\max\{j \in \mathbb{N},\ \sum_{l=0}^{j} C_m^l \leq i 2^{m-n}\}. \tag{2.5}$$

Preuve
La borne est obtenue en classant par poids croissant les éléments de F_2^n et ceux de F_2^m. Nous prenons les éléments de F_2^m ainsi triés consécutivement pour former 2^n ensembles de 2^{m-n} éléments. L'ensemble numéroté i est associé au i^{e} mot de F_2^n, ses mots étant classés par poids croissant. Quelques manipulations algébriques similaires à celles de la démonstration de P_{esi} nous conduisent à la formule (2.4).

□

Lorsque le canal est parfait, D_{sci} se confond avec la borne D_{inf}. Lorsque le canal est opaque ($p = 1/2$), cette borne vaut $1/2$. À l'instar de P_{esi}, nous étudions le comportement asymptotique de cette borne.

Théorème 2.4.2 *À rendement R fixé, la borne inférieure D_{sci} de la distorsion globale converge vers la meilleure distorsion théoriquement possible, i.e.*

$$\lim_{m\to\infty} \frac{n}{m} = R \Longrightarrow \lim_{m\to\infty} D_{sci} = H_2^{-1}(1 - R(1 - H_2(p))) \tag{2.6}$$

Preuve
La preuve est donnée à la section C.5.

□

Ce théorème montre que la borne D_{sci}, qui dépend de la complexité du code utilisé, a un comportement asymptotique en accord avec le théorème du codage conjoint de Shannon. Nous terminons cette brève étude théorique en reprenant l'exemple introduit lors du premier chapitre.

Suite de l'exemple 1 *Le code de longueur 5 et de dimension 3 défini par la matrice G_{sys} est utilisé pour comprimer une SBS et la transmettre sur le canal binaire symétrique. La distorsion globale est alors*

$$D = \frac{1}{20}(3 + 18p - 8p^2). \tag{2.7}$$

Celle créée par la version non systématique G est

$$D = \frac{1}{20}(3 + 26p - 32p^2 + 16p^3). \tag{2.8}$$

Encore une fois, la version systématique a de meilleures performances (i.e. une distorsion plus petite) que celle non systématique (p étant compris entre 0 et 1/2). La borne inférieure D_{sci} est donnée par

$$D_{sci} = \frac{1}{20}(3 + 13p + 3p^2 - 2p^3). \tag{2.9}$$

Cet exemple montre que le choix de l'étiquetage (*index assignment*), i.e. la correspondance entre mots d'information et mots de code, est crucial puisque deux codes équivalents n'ont pas les mêmes performances. Cet étiquetage a aussi fait le fruit de recherche dans d'autres contextes [17, 16, 29]. Nous nous limiterons, pour notre part, à des codes systématiques.

Étude pratique et comparaison avec le système séparé

Pour comparer avec le système séparé exposé précédemment, nous fixons la probabilité p de transition du CBS ($p = 5.10^{-4}$) ainsi que le rendement global R. Nous choisissons le code linéaire au cœur du système source ou canal parmi ceux trouvés au chapitre précédent. Une fois ce code déterminé, nous faisons croître la probabilité de transition du canal afin d'étudier la robustesse de ce système : p prend alors les valeurs 10^{-3}, 5.10^{-3}, 10^{-2}, 5.10^{-2} et 10^{-1}. La distorsion globale en fonction de p est indiquée par des cercles sur les figures 2.7 - 2.10 pour quatres valeurs du rendement global (1/2, 7/8, 6/5 et 2). Les meilleures performances théoriquement possibles sont indiquées par un trait continu. Les croix représentent les performances de divers systèmes séparés de même rendement global (nous avons considéré plusieurs paires de codes linéaires définissant le codeur global du système séparé). En pointillés figure, suivant la valeur du rendement global, la borne inférieure des performances du système source ou canal, à savoir D_{sci} ou P_{esi}. L'axe des abscisses représente le rapport signal-à-bruit du canal, noté RSB [3]. Nous considérons pour cela le canal binaire symétrique comme la concaténation d'une modulation de phase à deux états émise sur un canal à bruit blanc additif gaussien (dont la variance est déterminée par le rapport signal à bruit) suivi d'un détecteur à seuil. La probabilité de transition p du CBS est liée à ce rapport par la formule $p = Q(\sqrt{2\mathrm{RSB}})$ où Q est la fonction d'erreur normalisée ($Q(x) = 1/\sqrt{2\pi}\int_x^\infty \exp(-t^2/2)dt$).

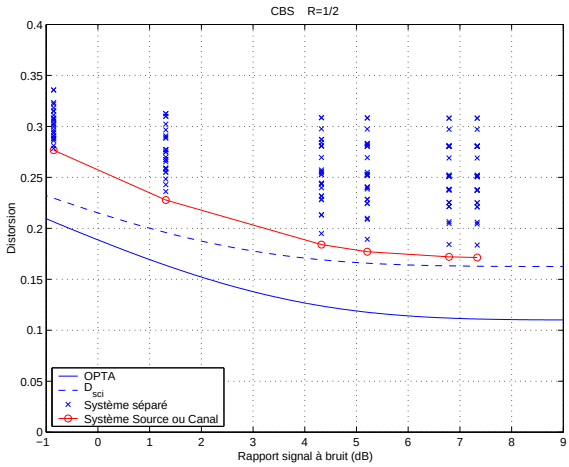


Fig. 2.7 – Performances du système source ou canal pour un rendement $R = 1/2$.

Dans nos exemples, quelle que soit la valeur du rendement global par rapport à 1, le système source ou canal présente de meilleures performances que le système séparé, même lorsque le canal se dégrade. De plus, les performances du système séparé sont proches des bornes inférieures (sans être tout à fait égales, même si cela ne se voit pas sur toutes les figures). Étant donné le comportement asymptotique des bornes des

[3] ce choix permettra de comparer la distorsion lorsque nous modifierons légèrement le modèle de canal à la section suivante.

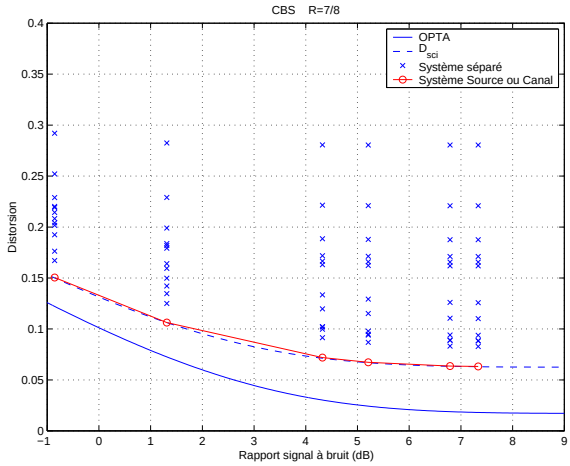


Fig. 2.8 – Performances du système source ou canal pour un rendement $R = 7/8$.

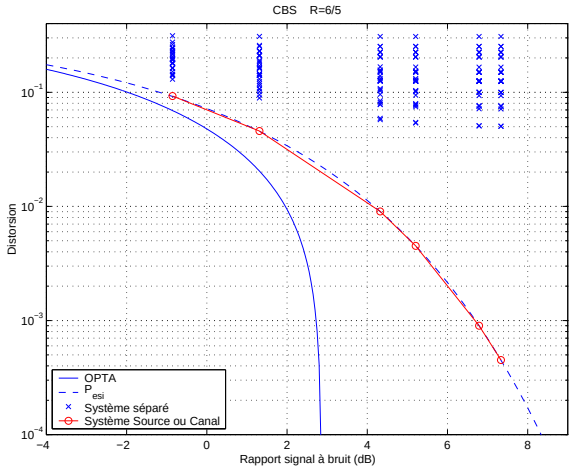


Fig. 2.9 – Performances du système source ou canal pour un rendement $R = 6/5$.

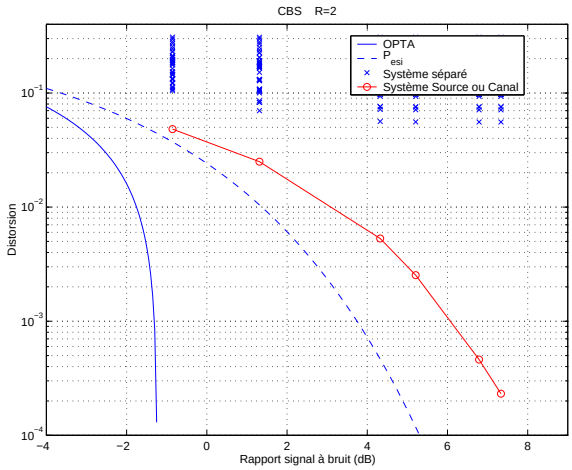


Fig. 2.10 – Performances du système source ou canal pour un rendement $R = 2$.

performances de ce système, nous pouvons supposer que le système source ou canal a des performances égales sinon meilleures que le système séparé, alors que sa conception est nettement plus aisée.

2.5 Algorithme de Lloyd généralisé doublement

Les performances du système source ou canal présenté dans la section précédente restent assez éloignées des meilleures performances possibles et, parfois, des bornes inférieures de performances dépendant de la longueur et de la dimension considérées. Une solution possible pour améliorer ces performances est alors de recourir à l'algorithme de Lloyd dont nous rappellerons brièvement le principe général avant d'expliquer comment nous l'avons adapté à nos besoins en prenant en compte les effets du canal : l'algorithme de Lloyd est alors généralisé doublement car il traite les effets du canal (première généralisation) et modélise codeur et décodeur comme des traitements aléatoires (seconde généralisation).

2.5.1 Rappels sur l'algorithme de Lloyd

L'algorithme de Lloyd, ou plus précisément de Lloyd-Max, a été proposé par Lloyd [34] en 1957 puis par Max [39] en 1960 dans le domaine du codage de source à valeur réelle scalaire. En 1980, Linde, Buzo et Gray [33] l'ont généralisé en en proposant une version vectorielle, couramment appelée LBG, appliquée dans leur article à des sources continues. Supposons que nous disposions d'un codeur et d'un décodeur de source. L'algorithme de Lloyd va améliorer alternativement ces deux fonctions en gardant l'autre fixe.

Supposons le décodeur de source fixé, c'est-à-dire les mots composant le code de source fixés. Lors du codage, la minimisation de la distorsion impose de choisir pour chaque mot de source le mot de code de source le plus proche. Ce choix est appelé la **condition du plus proche voisin**. Cette étape dessine donc le contour des régions de Voronoï de chaque mot du code de source.

Supposons maintenant le codeur de source fixé, c'est-à-dire les régions de Voronoï fixées. La distorsion est minimale lorsque le mot du code de source est le barycentre de la région de Voronoï. Cette étape est appelée la **condition du centroïde**.

L'algorithme de Lloyd consiste à itérer ces deux étapes, chaque itération diminuant la distorsion. Lorsque l'amélioration du système devient négligeable, l'algorithme prend fin. L'algorithme de Lloyd a donc l'avantage d'être général, simple à comprendre et à implémenter. Il peut s'avérer toutefois coûteux en temps de calcul lorsque l'on souhaite construire des codeurs de haute résolution (i.e. de faible distorsion).

Afin de pallier à cela, la version LBG propose d'augmenter progressivement la dimension du code de source en effectuant une optimisation pour chaque valeur de la dimension. Le nombre de vecteurs double à chaque augmentation de la dimension en appliquant une technique de dédoublement (*splitting*) : un mot du code de source engendre deux nouveaux mots, chacun translaté du premier dans une direction opposée à l'autre.

Le second inconvénient est de nature théorique : rien ne garantit que cet algorithme atteint la distorsion minimale. Il est en effet possible que les itérations nous conduisent dans un minimum local mais non global : le choix de l'initialisation est alors crucial pour obtenir de bons résultats.

Signalons enfin que Farvadin [17, 18, 19] a généralisé la version LBG en incorporant les perturbations introduites par le canal (en l'appliquant toujours à des sources continues). Le quantificateur et l'étiqueteur sont alors optimisés conjointement.

Nous allons maintenant détailler cet algorithme dans le cas d'une source discrète, la SBS, et d'une distance discrète, la distance de Hamming. Zahir-Azami [69] a développé la version de cet algorithme dans le cas d'une source discrète et d'une distance continue (distance euclidienne). Le choix de la distance mesurant la fidélité de la source reconstruite est alors analogue au choix entre décodage de canal dur ou souple.

2.5.2 Cas de la SBS seule

Avant d'expliquer comment nous allons prendre en compte le canal, nous allons le supposer dans un premier temps parfait. Cela nous permettra de détailler l'algorithme de Lloyd dans le cadre de la compression de SBS selon le critère de la distance de Hamming. Considérons l'extension m^{e} de la SBS. Soit $\mathcal{C} = \underline{v}_1, \cdots, \underline{v}_M$ un code de source de taille M et de longueur m. Chacun des mots de codes est associé à une étiquette ou mot d'information $\underline{i}$ représenté sur $n = \log_2 M$ bits. La condition du plus proche voisin définit les cellules de Voronoï $V(\underline{v}_k)$ associées à chacun des mots du code, les mots de source équidistants de deux mots de code ou plus étant attribués arbitrairement à l'une ou l'autre des cellules de Voronoï concernées [4]. Le mot de source reconstruit $\underline{\hat{u}}$ est ainsi l'un des éléments du code de source. La distorsion D d'un tel système est alors :

$$D = \frac{1}{m} \sum_{k=1}^{M} \sum_{\underline{u} \in V(\underline{v}_k)} p(\underline{u}) d_H(\underline{u}, \underline{v}_k).$$

Une fois ces cellules déterminées, la condition du centroïde nous impose de choisir de nouveaux mots de code notés $\underline{v}'_k$ qui vont minimiser chacun la distorsion dans leur cellule. La distance de Hamming $d_H(\underline{u}, \underline{v}_k)$ étant la somme des distances de Hamming

[4] cette situation ne manquera pas de se produire vu le peu de nombre de codes parfaits

calculées sur chacune des composantes, il nous suffit de minimiser la distorsion sur chacune d'elles. La j^{e} composante du mot $\underline{v}'_k$, notée $v'_{k,j}$, est donc déterminée par la j^{e} composante des mots de la cellule $V(\underline{v}_k)$:

$$v'_{k,j} = \begin{cases} 1 & \text{si } |\{\underline{u} \in V(\underline{v}_k),\ u_j = 1\}| > |\{\underline{u} \in V(\underline{v}_k),\ u_j = 0\}|, \\ 0 & \text{si } |\{\underline{u} \in V(\underline{v}_k),\ u_j = 1\}| < |\{\underline{u} \in V(\underline{v}_k),\ u_j = 0\}|, \\ 1 \text{ ou } 0 & \text{sinon.} \end{cases} \tag{2.10}$$

Dans le dernier cas de l'égalité (2.10), la valeur de $v'_{k,j}$ peut être choisie arbitrairement ou aléatoirement. Nous pouvons alors recommencer en réappliquant la condition du plus proche voisin.

Formulons autrement l'algorithme de Lloyd en choisissant un cadre probabiliste. Le codeur de source produit une étiquette $\underline{i}$, élément de F_2^n, à partir du mot de source $\underline{u}$ de F_2^m, le décodeur produit un mot de code de source $\underline{v}$ à partir de l'étiquette $\underline{i}$. La distorsion résultante s'écrit

$$D = \frac{1}{m} E[d_H(\underline{u}, \underline{v})] = \frac{1}{m} \sum_{\substack{(\underline{u},\underline{v}) \in F_2^m \times F_2^m \\ \underline{i} \in F_2^n}} p(\underline{u}) p(\underline{i}|\underline{u}) p(\underline{v}|\underline{i}) d_H(\underline{u}, \underline{v}).$$

$p(\underline{i}|\underline{u})$ et $p(\underline{v}|\underline{i})$ caractérisent respectivement le codeur et le décodeur de source : ce sont des matrices rectangulaires (de tailles respectives $2^n \times 2^m$ et $2^m \times 2^n$) dont une seule entrée par colonne est non nulle si les cellules de Voronoï ont des frontières définies une fois pour toutes.

Si on suppose le décodeur fixé alors nous pouvons récrire D

$$D = \frac{1}{m} \sum_{\substack{\underline{u} \in F_2^m \\ \underline{i} \in F_2^n}} p(\underline{i}|\underline{u}) \phi(\underline{u}, \underline{i}) \text{ où } \phi(\underline{u}, \underline{i}) = \sum_{\underline{v} \in F_2^m} p(\underline{u}) p(\underline{v}|\underline{i}) d_H(\underline{u}, \underline{v})$$

Lorsque le mot de source est fixé, la minimisation de la distorsion impose de choisir un codeur tel que

$$\begin{aligned} p(\underline{i}|\underline{u}) &= 0 \text{ si } \underline{i} \neq \underline{i}_0 \\ p(\underline{i}_0|\underline{u}) &= 1 \end{aligned}$$

avec $\underline{i}_0 = \arg\min_{\underline{i}} \phi(\underline{u}, \underline{i})$ (car D est une fonction convexe des probabilités de transition du codeur qui appartiennent à un ensemble convexe). Autrement dit, à mot de source fixé il faut choisir l'étiquette qui correspond au mot de code le plus proche : nous retrouvons la condition du plus proche voisin.

Lorsque le codeur est fixé, nous récrivons D en mettant en avant le décodeur

$$D = \frac{1}{m} \sum_{\substack{\underline{v} \in F_2^m \\ \underline{i} \in F_2^n}} p(\underline{v}|\underline{i}) \psi(\underline{v}, \underline{i}) \text{ où } \psi(\underline{v}, \underline{i}) = \sum_{\underline{u} \in F_2^m} p(\underline{u}) p(\underline{i}|\underline{u}) d_H(\underline{u}, \underline{v}).$$

À étiquette fixée, la minimisation de D conduit à choisir le décodeur tel que

$$\begin{aligned} p(\underline{v}|\underline{i}) &= 0 \text{ si } \underline{v} \neq \underline{v}_0 \\ p(\underline{v}_0|\underline{i}) &= 1 \end{aligned}$$

avec $\underline{v}_0 = \arg\min_{\underline{v}} \psi(\underline{v}, \underline{i})$ (les mêmes arguments s'appliquent). Ceci est l'équivalent de la condition du centroïde (2.10).

2.5.3 Algorithme de Lloyd généralisé doublement

Le cadre probabiliste introduit précédemment permet tout naturellement de prendre en compte le CBS défini lui aussi par sa matrice de transition (équation (1.1)). Tous les éléments de la chaîne de transmission sont représentés au moyen des probabilités de transition, comme indiqués sur le schéma 2.11. Ce dernier représente aussi bien le système séparé que le système source ou canal. Le codeur et le décodeur sont variables dans le sens où ils sont modifiés itérativement par l'algorithme de Lloyd modifié que nous allons présenter ici. Cet algorithme est dit généralisé doublement car :

- il prend en compte un modèle probabiliste de codeur et de décodeur ;
- il prend en compte le canal.

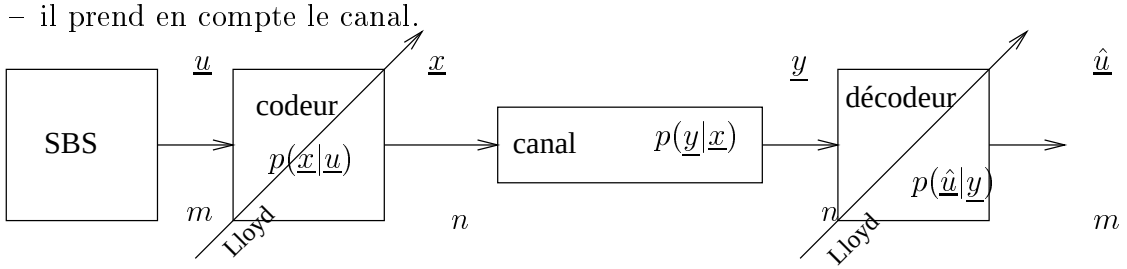


Fig. 2.11 – Algorithme de Lloyd généralisé doublement.

La distorsion globale au sens de Hamming du système représenté sur la figure 2.11 s'écrit en fonction des probabilités de transition de chacune des différentes « boîtes »

$$D = \frac{1}{m} E[d_H(\underline{u}, \underline{\hat{u}})] = \frac{1}{m} \sum_{\substack{(\underline{u},\underline{\hat{u}}) \in F_2^m \times F_2^m \\ (\underline{x},\underline{y}) \in F_2^n \times F_2^n}} p(\underline{u}) p(\underline{x}|\underline{u}) p(\underline{y}|\underline{x}) p(\underline{\hat{u}}|\underline{y}) d_H(\underline{u}, \underline{\hat{u}}).$$

Supposons que le codeur, c'est-à-dire les probabilités de transition $p(\underline{x}|\underline{u})$, est fixé. La minimisation de la distorsion globale nous conduit à optimiser le décodeur, c'est-à-dire les probabilités $p(\underline{\hat{u}}|\underline{y})$. À l'instar du cas de la SBS sur canal parfait étudié précédemment, nous récrivons la distorsion en isolant les probabilités à déterminer :

$$mD = \sum_{\substack{\underline{v} \in F_2^m \\ \underline{y} \in F_2^n}} p(\underline{\hat{u}}|\underline{y}) \psi(\underline{\hat{u}}, \underline{y}) \text{ où } \psi(\underline{\hat{u}}, \underline{y}) = \sum_{\substack{\underline{u} \in F_2^m \\ \underline{x} \in F_2^n}} p(\underline{u}) p(\underline{x}|\underline{u}) p(\underline{y}|\underline{x}) d_H(u, \underline{\hat{u}}).$$

Encore une fois, la distorsion D est une fonction convexe des probabilités représentant le décodeur, probabilités variant elles-mêmes dans un espace lui aussi convexe. La prise en compte du canal influe seulement sur la détermination du mot $\underline{\hat{u}}$ optimum, à mot reçu $\underline{y}$ fixé, par le biais du calcul de la fonction ψ. À codeur fixé, la distorsion est donc minimale lorsque le décodeur vérifie

$$\forall y \in F_2^n \; p(\underline{\hat{u}}|\underline{y}) = 0 \Longleftrightarrow \psi(\underline{\hat{u}}, \underline{y}) > \min_{\underline{\hat{u}}'} \psi(\underline{\hat{u}}', \underline{y}) \tag{2.11}$$

La somme des probabilités de transition du décodeur non spécifiées par l'équation ci-dessus doit bien entendu être égale à l'unité.

Supposons à présent ce décodeur fixé et optimisons le codeur représenté par les probabilités $p(\underline{x}|\underline{u})$.

$$mD = \sum_{\substack{\underline{u} \in F_2^m \\ \underline{x} \in F_2^n}} p(\underline{x}|\underline{u})\phi(\underline{u}, \underline{x}) \text{ où } \phi(\underline{u}, \underline{x}) = \sum_{\substack{\underline{\hat{u}} \in F_2^m \\ \underline{y} \in F_2^n}} p(\underline{u})p(\underline{y}|\underline{x})p(\underline{\hat{u}}|\underline{y})d_H(\underline{u}, \underline{\hat{u}})$$

Les remarques formulées lors de l'optimisation précédente sont encore valables. La distorsion est donc minimale lorsque le codeur vérifie :

$$\forall \underline{u} \in F_2^m \; p(\underline{x}|\underline{u}) = 0 \Longleftrightarrow \phi(\underline{u}, \underline{x}) > \min_{\underline{x}'} \phi(\underline{u}, \underline{x}') \tag{2.12}$$

La somme des probabilités de transition du codeur non spécifiées par l'équation ci-dessus doit être égale à l'unité.

Les équations (2.11) et (2.12) montrent que la solution optimale et les minima locaux du système correspondent à des codeurs et décodeurs déterministes, i.e. dont la sortie correspondant à une entrée donnée est toujours la même. Autrement dit, il s'agit de choisir un élément et un seul de l'ensemble des mots $\underline{x}$ minimisant $\phi(\underline{u}, \underline{x})$ à mot de source $\underline{u}$ donné pour spécifier la sortie du codeur. De même, on choisit un élément et un seul de l'ensemble des mots de source reconstruits $\underline{\hat{u}}$ minimisant $\psi(\underline{\hat{u}}, \underline{y})$ à mot reçu $\underline{y}$ fixé : cet élément détermine alors la sortie du codeur.

Des simulations de cet algorithme de Lloyd généralisé doublement nous ont montré que celui-ci se terminait en quelques itérations : afin de pallier à cela et espérant pouvoir sortir des minima locaux, **nous autorisons codeur et décodeur à être aléatoires (ou flous)**, comme le permettent les équations (2.11) et (2.12) lorsque les fonctions ψ et ϕ ont un minimum atteint en plusieurs points. Lorsqu'une entrée se présente, ces « boîtes » floues choisissent au hasard une sortie parmi un ensemble restreint. Si nécessaire, à la fin de l'algorithme, les mêmes équations nous garantissent que nous pourrons revenir à une solution déterministe.

Nous avons utilisé l'algorithme ainsi développé en choisissant pour points d'initialisation le système source ou canal. Les figures 2.12 et 2.13 montrent les performances du système séparé (pour rappel) indiquées par des carrés, du système source ou canal indiquées par des croix et enfin celles de ce même système amélioré par l'algorithme

de Lloyd, représentées par des cercles. L'OPTA est tracé en trait continu. Nous observons que l'algorithme de Lloyd n'apporte qu'une amélioration marginale en dépit des efforts faits pour sortir des minima locaux que représente le système source ou canal : le gain obtenu par l'algorithme de Lloyd est de l'ordre de quelques millièmes de la distorsion initiale, ce qui explique que cette amélioration n'est pas visible sur les figures. L'algorithme de Lloyd modifie donc légèrement les codeurs et les décodeurs.

Même si les résultats ne sont pas à la hauteur de nos espérances, l'algorithme de Lloyd généralisé doublement présente des avantages non négligeables :

- la modélisation retenue est la plus large possible car elle ne s'appuie pas sur une structure bien particulière de codeur ou de décodeur mais sur une approche probabiliste qui est celle retenue en théorie de l'information. L'algorithme de Lloyd nous permet alors de concevoir un système véritablement conjoint ;
- à chaque itération, les performances ne peuvent que s'améliorer ou être identiques (propriété de l'algorithme de Lloyd initial). L'algorithme de Lloyd est un outil permettant d'améliorer n'importe quel système ;
- le changement de modèle de canal ne modifie pas la nature de l'algorithme : il est facilement transposable d'un canal à l'autre.

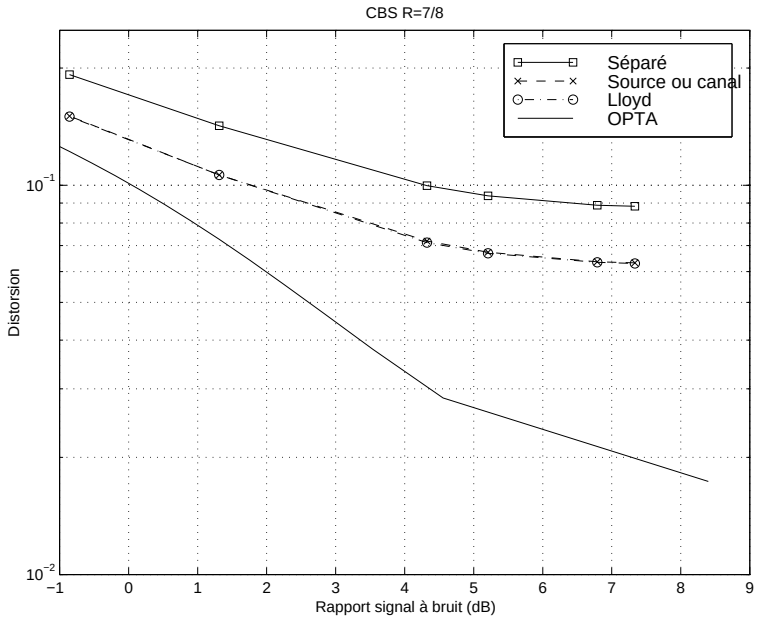


Fig. 2.12 – Performances de l'algorithme de Lloyd généralisé doublement appliqué au codage SBS-CBS pour un rendement $R = 7/8$.

Nous allons illustrer ce dernier point en nous limitant à des canaux sans mémoire à entrée binaire et dont la sortie sera ternaire ou quaternaire. Ces canaux représentent alors une modulation de phase à deux états émise sur un canal à bruit blanc additif gaussien suivi d'un détecteur multiniveaux (en l'occurence trois ou quatre). Une telle modélisation permet d'introduire un peu de souplesse dans le décodage : la sortie du détecteur à seuil permet de mesurer la confiance que l'on peut avoir en celle-ci.

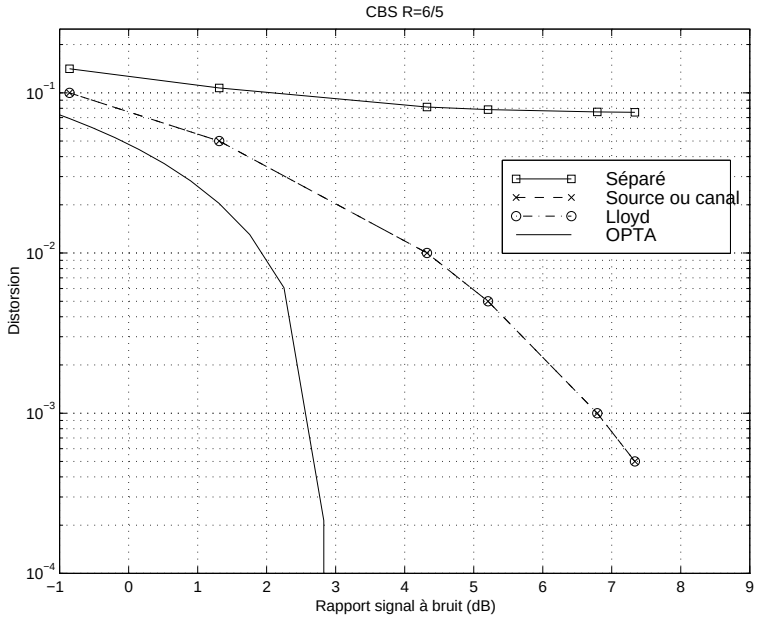


Fig. 2.13 – Performances de l'algorithme de Lloyd généralisé doublement appliqué au codage SBS-CBS pour un rendement $R = 6/5$.

D'un point de vue pratique pour appliquer l'algorithme de Lloyd, nous devons calculer la matrice de transition du canal étudié : nous utilisons, comme nous l'avons déjà fait pour le cas binaire, l'équivalence entre le canal à entrée binaire et à sortie quantifiée et le canal à bruit additif blanc gaussien à entrée binaire et sortie quantifiée. Notons E_s l'énergie du signal émis par le modulateur. Afin de faire des comparaisons justes entre les systèmes, nous fixons le rapport signal-à-bruit RSB et équirépartissons les seuils dans l'intervalle $[-\sqrt{E_s}, +\sqrt{E_s}]$. Enfin, puisque le changement de canal n'affecte que le décodeur, nous utiliserons le codeur du système source ou canal pour initialiser l'algorithme de Lloyd : les figures représentant les résultats pour les canaux à sortie ternaire ou quaternaire n'indiqueront que la distorsion obtenue à la fin de l'algorithme de Lloyd généralisé doublement [5]. Le décodeur de canal du système séparé est un décodeur à maximum de vraisemblance (qui suppose donc connue la matrice de transition du canal).

Canal à sortie ternaire

Le canal à entrée binaire et sortie ternaire est aussi appelé canal à effacements car la sortie intermédiaire de ce canal n'est pas digne de confiance : la probabilité de se tromper en décidant quel symbole a été émis est élevée. Dans le domaine du codage de canal, il est depuis longtemps reconnu [50] qu'il est plus facile de corriger des effacements que des erreurs : en effet, les positions des effacements sont connues

[5] Les performances du système source ou canal adapté au canal à sortie binaire ne peuvent être représentées sur ces figures car le modèle de canal a changé.

par le décodeur, contrairement à celles des erreurs. La matrice de transition du canal à sortie binaire et sortie ternaire est alors donnée par

$$\begin{pmatrix} 1-Q(\frac{2}{3}\sqrt{2\mathrm{RSB}}) & Q(\frac{4}{3}\sqrt{2\mathrm{RSB}}) \\ Q(\frac{2}{3}\sqrt{2\mathrm{RSB}})-Q(\frac{4}{3}\sqrt{2\mathrm{RSB}}) & Q(\frac{2}{3}\sqrt{2\mathrm{RSB}})-Q(\frac{4}{3}\sqrt{2\mathrm{RSB}}) \\ Q(\frac{4}{3}\sqrt{2\mathrm{RSB}}) & 1-Q(\frac{2}{3}\sqrt{2\mathrm{RSB}}) \end{pmatrix}.$$

Les figures 2.14 et 2.15 représentent les performances des systèmes séparés et source ou canal pour deux valeurs de rendement ($R = 7/8$ et $R = 6/5$). Les codes linéaires systématiques composant le système séparé sont les mêmes que dans le cas du CBS : ils sont fixés par le plus grand rapport signal-à-bruit simulé. Nous faisons diminuer ce rapport : le décodeur de canal du système séparé va prendre en compte les nouvelles valeurs des probabilités de transition. Le système source ou canal est optimisé par l'algorithme de Lloyd pour le plus grand rapport signal-à-bruit simulé. Codeur et décodeur resteront fixes lorsque ce rapport va diminuer, mesurant ainsi la robustesse du système face à un canal variant lentement dans le temps. Enfin, nous avons tracé en trait continu l'OPTA correspondant à ce canal.

Encore une fois le système source ou canal bat largement le système séparé à complexité égale. De plus, la lente variation du canal dans le temps ne vient pas troubler ce classement des systèmes.

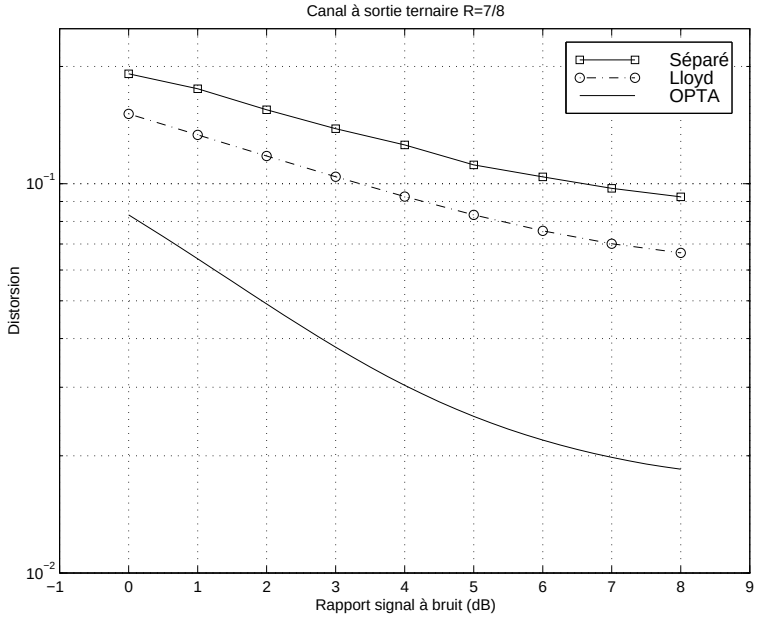


Fig. 2.14 – Performances du système séparé et du système optimisé par algorithme de Lloyd initialisé par le codeur source ou canal pour le canal à sortie ternaire avec un rendement$R = 7/8$.

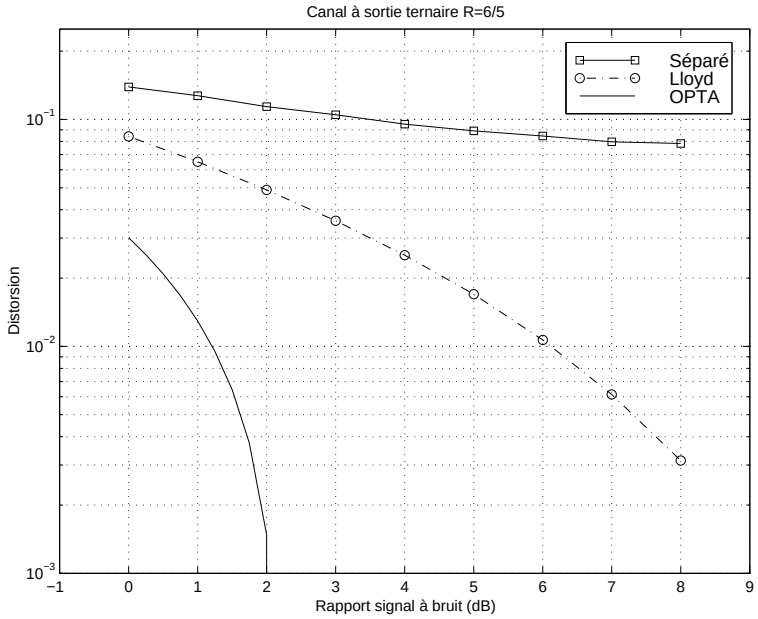


Fig. 2.15 – Performances du système séparé et du système optimisé par algorithme de Lloyd initialisé par le codeur source ou canal pour le canal à sortie ternaire avec un rendement $R = 6/5$.

Canal à sortie quaternaire

Nous recommençons l'expérience menée avec le canal à sortie ternaire en ajoutant une sortie supplémentaire. La sortie d'un tel canal peut être considérée comme très fiable si elle est supérieure à $\frac{\sqrt{E_s}}{2}$ en valeur absolue ou moins fiable si elle est inférieure à ce seuil. Le signe de la sortie détermine *a priori* le symbole émis. La matrice de transition ce canal à sortie quaternaire est donnée par

$$\begin{pmatrix} 1-Q(\frac{1}{2}\sqrt{2\mathrm{RSB}}) & Q(\frac{3}{2}\sqrt{2\mathrm{RSB}}) \\ Q(\frac{1}{2}\sqrt{2\mathrm{RSB}})-Q(\sqrt{2\mathrm{RSB}}) & Q(\sqrt{2\mathrm{RSB}})-Q(\frac{3}{2}\sqrt{2\mathrm{RSB}}) \\ Q(\sqrt{2\mathrm{RSB}})-Q(\frac{3}{2}\sqrt{2\mathrm{RSB}}) & Q(\frac{1}{2}\sqrt{2\mathrm{RSB}})-Q(\sqrt{2\mathrm{RSB}}) \\ Q(\frac{3}{2}\sqrt{2\mathrm{RSB}}) & 1-Q(\frac{1}{2}\sqrt{2\mathrm{RSB}}) \end{pmatrix}.$$

Les figures 2.16 et 2.17 présentent les performances des deux systèmes étudiés à rendement global fixé : le système source ou canal bat le système séparé, et ce même lorsque le canal varie. Par rapport au canal à sortie ternaire, les performances sont, bien entendu, améliorées mais la distorsion globale atteignable avec un canal à sortie quaternaire est inférieure à celle atteignable avec un canal à sortie ternaire (i.e. l'OPTA a, elle aussi, changé).

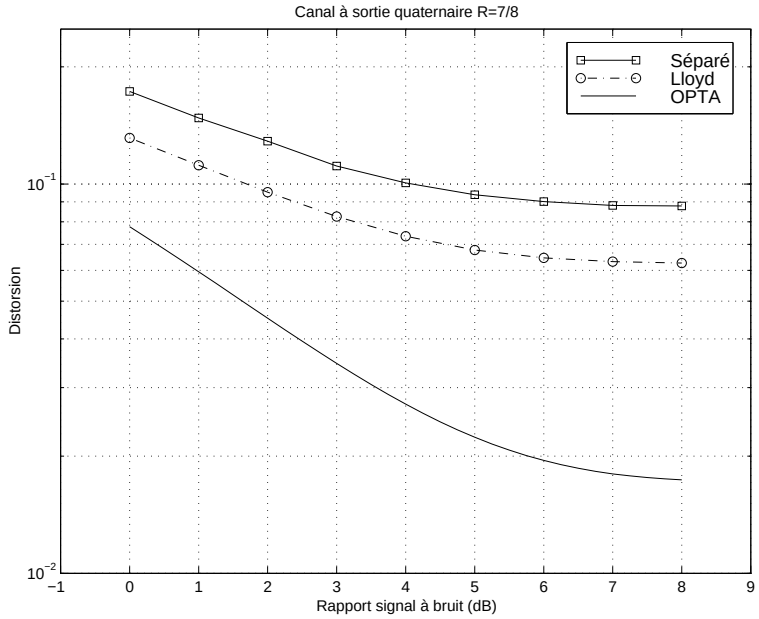


Fig. 2.16 – Performances du système séparé et du système optimisé par algorithme de Lloyd initialisé par le codeur source ou canal, canal à sortie quaternaire pour un rendement $R = 7/8$.

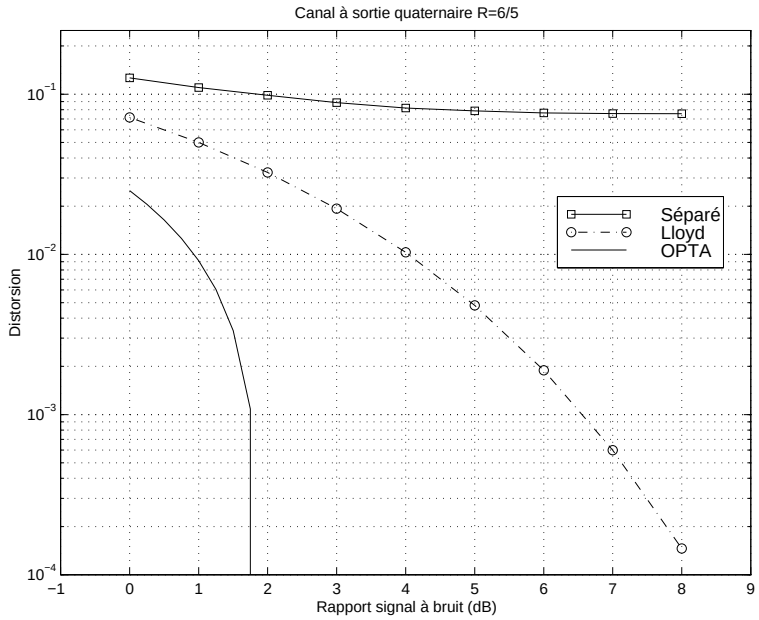


Fig. 2.17 – Performances du système séparé et du système optimisé par algorithme de Lloyd initialisé par le codeur source ou canal, canal à sortie quaternaire pour un rendement $R = 6/5$.

2.6 Conclusions

Nous nous sommes penchés dans ce chapitre sur le problème du codage conjoint SBS-CBS en mesurant la distorsion par le biais de la distance de Hamming. La preuve

du théorème de codage conjoint nous conduit à envisager un système séparé ou tandem qui est la concaténation du codage de source et du codage de canal. Ce système a été simulé à l'aide des codes trouvés au premier chapitre.

Puis nous avons proposé un système plus simple, appelé source ou canal, et qui supprime l'une ou l'autre des étapes du codage séparé en fonction du rendement global. Des bornes inférieures de performances de ce système ont été présentées : elles convergent asymptotiquement vers les meilleures performances théoriquement possibles. Nous supposons donc que ce système permet, lui aussi, d'atteindre asymptotiquement ces performances.

Nous avons comparé ces deux systèmes à complexité à peu près égale et pour de faibles retards (c'est-à-dire de faibles longueurs) : le système le plus simple à mettre en œuvre, à savoir le système source ou canal présente les meilleures performances. Nous avons ensuite essayé d'améliorer ces dernières en recourant à un outil bien classique du codage de source, l'algorithme de Lloyd. Nous avons détaillé son fonctionnement pour une source discrète et une distance discrète, elle aussi (distance de Hamming) tout en prenant en compte les effets du CBS. Cela a permis d'améliorer marginalement le système. Nous avons profité de la modélisation assez large de celui-ci pour l'appliquer à des canaux à entrée binaire et à sortie ternaire ou quaternaire : l'algorithme, outre son rôle habituel d'optimisation, nous donne alors des décodeurs prêts à l'emploi. Nous avons alors comparé les performances du système optimisé par l'algorithme de Lloyd initialisé par le codeur du système source ou canal avec celles du système séparé (avec un décodeur de canal à maximum de vraisemblance) en nous attardant sur la robustesse des performances par rapport à des variations lentes du canal. Là encore, le système source ou canal est, à complexité égale, meilleur que le système séparé. L'algorithme de Lloyd nous permet également de trouver un bon décodeur lorsque le nombre de sorties du canal à entrée binaire augmente, c'est-à-dire lorsque le récepteur quantifie plus finement le signal reçu. Le décodeur fourni par cet algorithme est déterministe mais lors des itérations, il est considéré comme flou.

Ce chapitre illustre donc l'intérêt de concevoir conjointement un système [15] : la solution recommandée par la preuve du théorème du codage conjoint ne conduit pas à la solution la plus simple et la plus performante en pratique dans le cas étudié.

Chapitre 3

Codage de source binaire asymétrique suivant le critère de distance de Hamming

3.1 Introduction

Nous avons considéré dans les deux premiers chapitres la source binaire symétrique. Cependant, si cette source présente l'avantage d'être facile à étudier, elle modélise assez mal les sources binaires réelles (comme par exemple un fichier texte, un fichier d'une suite bureautique, une image ou encore les bits en sortie d'un appareil de mesure spatial [1]). Ces sources réelles se distinguent de la SBS par deux aspects cruciaux :

- les bits ne sont pas statistiquement indépendants ;
- les bits ne sont pas équiprobables.

Nous prenons en compte dans ce chapitre ce deuxième aspect en étudiant la compression des sources binaires asymétriques (**SBA**). Ces sources subissent généralement un codage entropique au moyen de codes de Huffman ou à longueur variable [13]. Cependant, ces codes, qui atteignent asymptotiquement l'entropie de la source, sont sensibles aux erreurs que le canal peut introduire. Si le décodeur de canal ne corrige pas toutes les erreurs parfaitement, le décodeur du code de longueur variable se trompe, et pire, perd la synchronisation : l'erreur se propage alors aux symboles de source suivants. Il est possible de remédier à ce problème en introduisant un mot de synchronisation (*comma codes*) afin de limiter la propagation de l'erreur. Une autre solution consiste à choisir parmi des codes de même longueur variable celui qui limite le mieux la propagation de l'erreur [48, 40, 45]. Enfin, on peut utiliser les codes à longueur variable auto-synchronisés [20, 46] qui détectent une erreur assez rapidement et ne la propagent pas aux autres mots codés.

La source binaire asymétrique n'est donc pas reproduite fidèlement lorsque le codeur de canal ne corrige pas toutes les erreurs. Nous allons donc essayer de coder par bloc et avec pertes cette source, l'utilisation de blocs ne posant pas de problème de synchronisation au niveau du décodeur de source. Nous rappellerons dans un premier temps la fonction rendement-distorsion d'une source binaire asymétrique puis nous proposerons un codeur en bloc simple et à étiquetage complexe, le codeur par couronne, dont nous évaluerons les performances asymptotiques. En vue de simplifier, l'étiquetage, nous présenterons le codage de source par syndrome développé par Ancheta [1] utilisé pour réaliser un codeur conjoint SBA-CBS [37]. Nous donnerons les performances théoriques de ce codage de source. Toujours dans un souci de faible complexité, nous utiliserons le codeur de source linéaire développé en 1.2.1 pour comprimer la SBA. Enfin, nous étudierons les résultats de l'algorithme de Lloyd appliqué à la compression de la SBA.

3.2 Définition et performances optimales (OPTA)

Définition 3.2.1 (SBA) *Une source binaire asymétrique sans mémoire, notée SBA, est une source émettant le symbole* 1 *avec une probabilité* p_s *et le symbole* 0 *avec une probabilité* $1-p_s$*. Les symboles émis sont statistiquement indépendants.*

D'après cette définition, la probabilité que la source émette un mot $\underline{u}$ donné (de longueur m) est simplement

$$p(\underline{u}) = p_s^{w_H(\underline{u})}(1-p_s)^{m-w_H(\underline{u})}.$$

Le critère de fidélité choisi pour comprimer la source est la distorsion de Hamming par bit. L'égalité (1.31) détermine la fonction rendement-distorsion [41], notée $R_{SBA}(D)$

$$R_{SBA}(D) = \begin{cases} H_2(p_s) - H_2(D) & \text{si} \quad 0 \leq D \leq \min(p_s, 1-p_s) \\ 0 & \text{si} \quad \min(p_s, 1-p_s) \leq D \leq 1 \end{cases} \tag{3.1}$$

Le théorème de codage de source [41] affirme qu'étant donnés D et $R_s > R_{SBA}(D)$, il existe un code de rendement R compris entre $R_{SBA}(D)$ et R_s et dont la distorsion est aussi proche que l'on veut de D.

Nous supposerons par la suite que $p_s \leq 1/2$: dans le cas contraire, il suffit de complémenter à un les mots de source émis pour se ramener au cas $p_s \leq 1/2$. $H_2(p_s)$ mesure l'entropie de la SBA : la longueur moyenne des mots de source codés par de bons codes à longueur variable atteint asymptotiquement cette entropie. Notre but étant d'étudier de bons codes de longueur fixe pour comprimer la SBA, nous allons tout d'abord montrer qu'il existe des codes de longueur fixe dont le rendement atteint asymptotiquement l'entropie et dont la distorsion tend vers 0.

3.3 Compression par une couronne

Considérons l'extension m^{e} d'une SBA de paramètre p_s : cette source émet des mots $\underline{u}$. Pour comprimer la source, nous comparons le poids de Hamming du mot $\underline{u}$ à un seuil $m\delta$ (δ étant un réel positif plus petit que 1) :

- si $w_H(\underline{u}) \leq m\delta$; alors la sortie du décodeur de source $\underline{v}$ est $\underline{v} = \underline{u}$;
- si $w_H(\underline{u}) > m\delta$ alors le mot de source est « projeté » sur la sphère des mots de poids $\lfloor m\delta \rfloor$ en mettant $w_H(\underline{u}) - \lfloor m\delta \rfloor$ composantes non nulles de $\underline{u}$ à 0.

Enfin, un organe d'étiquetage restitue une étiquette $\underline{\imath}$ sur k bits. Un tel schéma de compression est représenté sur la figure 3.1.

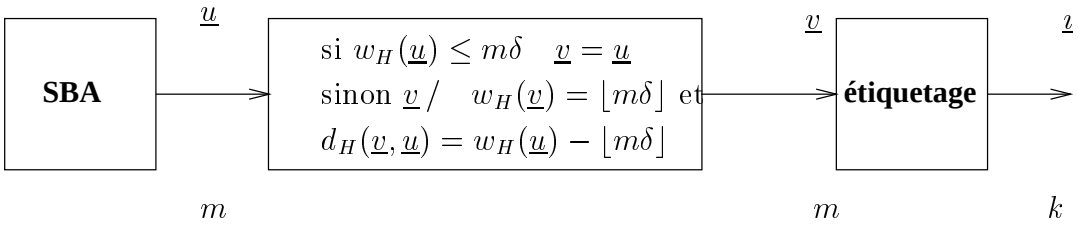


FIG. 3.1 – Compression d'une SBA par une boule.

Le rendement R_s d'un tel codeur de source est

$$R_s = \frac{k}{m} = \frac{\lceil \log_2 \sum_{i=0}^{\lfloor m\delta \rfloor} C_m^i \rceil}{m}. \tag{3.2}$$

Par construction, la distorsion est donc uniquement due aux mots en dehors de la boule de rayon $\lfloor m\delta \rfloor$, soit :

$$D = \frac{1}{m} E[d_H(\underline{u}, \underline{v})] = \frac{1}{m} \sum_{i=\lceil m\delta \rceil}^{m} C_m^i p_s^i (1-p_s)^{m-i} (i - \lfloor m\delta \rfloor). \tag{3.3}$$

Nous allons maintenant étudier les performances asymptotiques de ce codeur de source : supposons alors que $\delta < p_s$. Le théorème A.2.1 montre que R converge alors vers $H_2(\delta)$. La distorsion se récrit, en tenant compte du fait que le poids moyen d'un mot de source est mp_s,

$$D = p_s - \frac{\lfloor m\delta \rfloor}{m} + \sum_{i=0}^{\lfloor m\delta \rfloor - 1} \frac{\lceil m\delta \rceil - i}{m} C_m^i p_s^i (1-p_s)^{m-i}.$$

Le théorème A.2.2 montre alors que la distorsion D converge vers $p_s - \delta$. La distorsion et le rendement de ce codeur vérifient donc asymptotiquement

$$R_s = H_2(p_s - D). \tag{3.4}$$

Par conséquent, asymptotiquement, il existe des codes en blocs permettant de coder presque sans perte la SBA avec un rendement proche de l'entropie. Les performances du codeur par boule sont illustrées sur la figure 3.2.

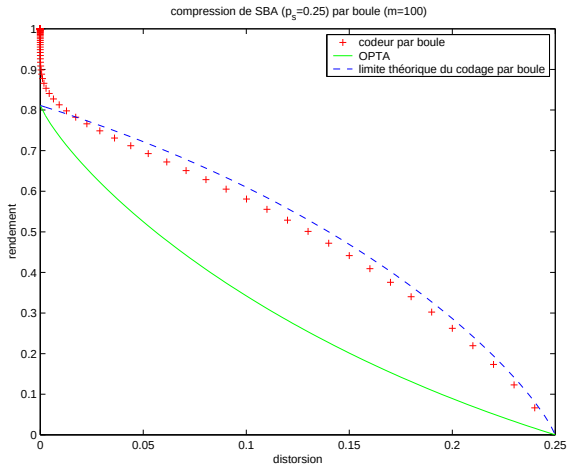


Fig. 3.2 – Performances du codeur par boule ($p_s = 0.25$, $m = 100$).

Le code de source ainsi défini contient des mots de source atypiques. Un mot de source est dit ϵ-atypique [13], ϵ étant un réel positif, si et seulement si

$$\left| -\frac{1}{m} \log_2 p(\underline{u}) - H_2(p_s) \right| > \epsilon.$$

La loi faible des grands nombres implique que ces mots atypiques sont en proportion négligeable. Le code de source précédent contient les mots atypiques de poids faible par rapport à mp_s. Le nouveau codeur de source va donc « projeter » ces derniers sur la « surface » d'une sphère de rayon $m\delta' < m\delta$ et se comporter comme le précédent codeur pour les autres mots. Le code de source ainsi défini ne contient que les mots dont le poids est compris entre $m\delta'$ et $m\delta$, définissant ainsi une couronne. Le rendement d'un tel codeur sera légèrement inférieur à celui du codeur par boule. Quant à la distorsion, elle sera légèrement supérieure à celle du codeur précédent : la contribution des mots atypiques de poids faible à la distorsion sera négligeable lorsque la longueur des codes tends vers l'infini (toujours en raison de la loi faible des grands nombres), comme le montre la figure 3.3. Sur cette figure, nous observons que les meilleurs codeurs par couronne obtenus (c'est-à-dire ceux de plus faible rendement à distorsion donnée) se comportent asymptotiquement comme les codeurs par boule.

Le codeur par boule ou par couronne permet donc de coder la SBA presque sans perte et avec un rendement proche de l'entropie. Cependant, lorsque la distorsion visée n'est pas nulle, ces codeurs sont asymptotiquement mauvais car leurs performances ($R = H_2(p_s - D)$) sont largement supérieures à l'OPTA ($R = H_2(p_s) - H_2(D)$), comme l'ont montré les figures 3.2 et 3.3. De plus, l'étiqueteur associé ne présente pas de structure particulière, ce qui peut le rendre complexe.

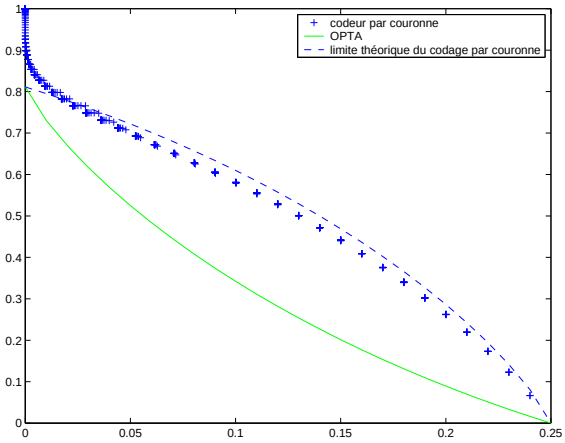


Fig. 3.3 – Performances du codeur par couronne ($p_s = 0.25$, $m = 100$).

3.4 Le codage de source par syndrome d'Ancheta

Ancheta [1] a proposé un codage linéaire de la SBA dont la distorsion est asymptotiquement nulle et dont le rendement tend vers l'entropie de la source. Sa technique repose sur le codage de source par syndrome (*Syndrome-source-coding*) et est représentée sur la figure 3.4.

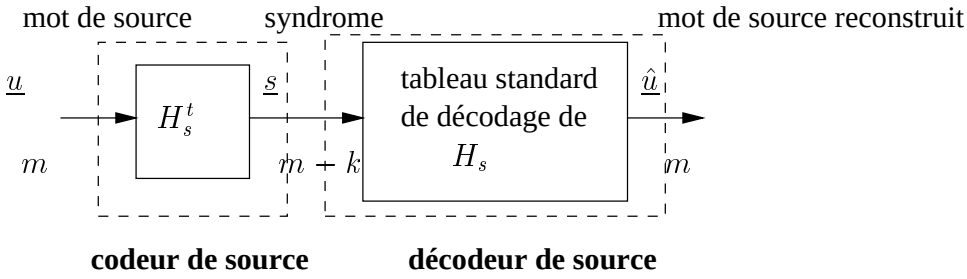


Fig. 3.4 – Codage de source par syndrome.

Le codage de source par syndrome s'appuie sur la matrice de parité H_s d'un code linéaire de longueur m et de dimension k. Le mot de source $\underline{u}$ est comprimé par son syndrome $\underline{s} = \underline{u}H_s^t$, de dimension $m-k$. Ce codeur de source est donc simple en regard des codeurs de source traditionnels.

Le décodeur de source reçoit le syndrome et doit, afin de minimiser la distorsion, trouver le mot de source le plus probable connaissant ce syndrome. Le décodeur fournit donc au destinataire le chef de classe $\underline{e}$ associé au syndrome reçu, c'est-à-dire le mot de la cellule de Voronoï du mot nul dont le syndrome est identique à celui reçu. Ce décodeur est donc plus complexe que le codeur associé.

3.4.1 Performances

Le rendement R_s du codeur par syndrome est donné par

$$R_s = \frac{m-k}{m} = 1 - \frac{k}{m}.$$

Ancheta [1] a montré que le codage de source par syndrome permet d'atteindre une distorsion aussi faible que voulue et un rendement proche de l'entropie de la source $H_2(p_s)$. L'argument utilisé par Ancheta pour prouver cela se fonde sur la dualité : l'erreur ajoutée par le CBS est équivalente à la sortie d'une SBA et il existe des codes de canal de la probabilité d'erreur aussi petite que voulue lorsque $k/m < 1 - H_2(p_s)$.

Pour établir l'expression de la distorsion dans le cas général, nous utilisons à nouveau le fait que les cellules de Voronoï définissent une partition de F_2^m et qu'elles se déduisent les unes des autres par translation :

$$\begin{aligned} D &= \frac{1}{m} E[d_H(\underline{u}, \hat{\underline{u}})] \\ &= \frac{1}{m} \sum_{c \in \mathcal{C}} \sum_{\underline{e} \in V(\underline{0})} p(\underline{u} = \underline{c} + \underline{e}) d_H(\underline{c} + \underline{e}, \underline{e}) \\ &= \frac{1}{m} \sum_{\underline{c} \in \mathcal{C}} w_H(\underline{c}) \sum_{\underline{e} \in V(\underline{0})} p(\underline{u} = \underline{c} + \underline{e}) \\ &= \frac{1}{m} \sum_{\underline{c} \in \mathcal{C}} w_H(\underline{c}) \sum_{\underline{e} \in V(\underline{0})} p_s^{w_H(\underline{c}+\underline{e})} (1-p_s)^{m - w_H(\underline{c}+\underline{e})}. \end{aligned} \tag{3.5}$$

La distorsion D du codage de source par syndrome fait donc intervenir la distribution de poids du code linéaire choisi $\mathcal{C}$ et la probabilité des cellules de Voronoï.

Le rendement R_s du codeur de source est simplement

$$R_s = 1 - \frac{k}{m}. \tag{3.6}$$

Si nous choisissons un code trivial, i.e. un code linéaire pour lequel la matrice Par est la matrice nulle, l'expression de la distorsion devient

$$D = \frac{k}{m} p_s = (1 - R_s) p_s. \tag{3.7}$$

Ces codes triviaux ont donc des performances éloignées de l'OPTA sauf lorsque leur rendement tend vers 0. Illustrons ces calculs en poursuivant l'exemple introduit au premier chapitre.

Suite de l'exemple 1 *Lorsque la SBA est codée par ses syndromes, le code systématique produit une distorsion D égale à*

$$D = \frac{1}{5}(4p_s + 12p_s^2 - 32p_s^3 + 28p_s^4 - 8p_s^5).$$

Son rendement R_s est égal à 2/5 Cette distorsion est supérieure à celle d'un code trivial de même rendement.

Une autre famille de codes peut s'étudier facilement : les codes de Hamming binaires définis en D.1. Soit A_i le nombre de mots de poids i d'un code de Hamming binaire de longueur m. Les coefficients A_i vérifient la relation suivante, tirée de [43], y étant un réel,

$$\sum_{i=0}^{m} iA_i y^{i-1} + \sum_{i=0}^{m} A_i y^i + \sum_{i=0}^{m} (m-i)A_i y^{i+1} = (1+y)^m. \tag{3.8}$$

Cette relation est due au fait que les codes de Hamming sont parfaits et corrigent une erreur. Par conséquent, la distorsion créée en utilisant les codes de Hamming est

$$\begin{aligned} D &= \frac{1}{m}\sum_{i=0}^{m} iA_i p_s^{i-1}(1-p_s)^{m-i-1}(i(1-p_s)^2 + p_s(1-p_s) + (m-i)p_s^2) \\ &= \frac{1}{m}\sum_{i=0}^{m} iA_i p_s^{i-1}(1-p_s)^{m-i-1}((m-1)p_s^2 + (1-2i)p_s + i). \end{aligned}$$

Les coefficients A_i sont facilement déterminés en résolvant l'équation (3.8) avec $y = 1$, les coefficients initiaux étant $A_0 = 1$ et $A_1 = 0$.

Longueur	Rendement R_s	Distorsion ($p_s = 0.4$)	Distorsion ($p_s = 0.1$)
7	0.4286	0.4211	0.0669
15	0.2061	0.4125	0.1039
31	0.1613	0.4062	0.1180

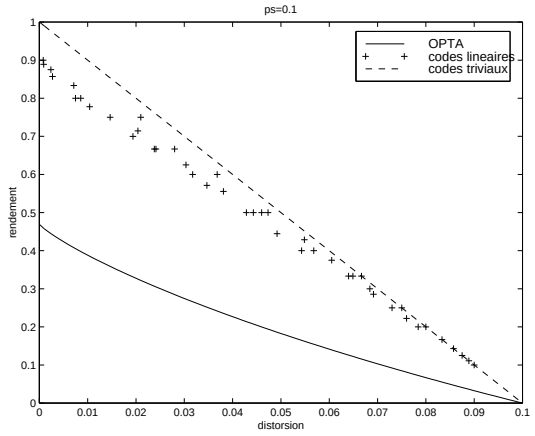
Tab. 3.1 – Distorsion d'une SBA comprimée par les syndromes des codes de Hamming.

Le tableau 3.1 montre la distorsion des codes de Hamming pour deux SBA différentes. Cet exemple montre que des codes, considérés comme bons tant pour le codage de SBS que pour le codage de canal, se révèlent mauvais puisque la plupart dépassent le niveau de distorsion p_s correspondant au code universel (dont le rendement R_s est nul). Il est par conséquent nécessaire de rechercher de bons codes pour le codage de SBA par syndrome.

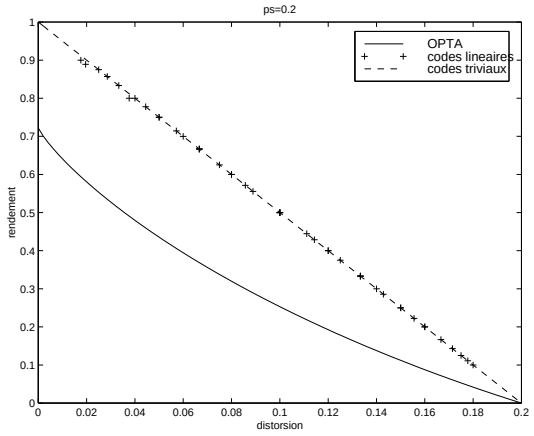
3.4.2 Recherches de bons codes de petite longueur

Nous avons mené une recherche exhaustive pour des longueurs de code m variant entre 3 et 10 pour différentes valeurs du paramètre p_s de la SBA (0.1, 0.2, 0.3, 0.4, 0.5). Nous nous sommes restreints aux codes systématiques car deux codes équivalents engendrent une même distorsion. La SBA considérée varie donc d'une source fortement asymétrique à une source symétrique. Les meilleures performances des codes examinés sont représentés par des croix sur les figures 3.5-3.7. Les performances des codes triviaux sont indiquées en pointillés et l'OPTA est tracée en trait continu. Pour de faibles

longueurs, le codage de SBA par syndrome n'est efficace que pour les sources fortement asymétriques ($p_s = 0.1$). Lorsque p_s vaut 0.2, nous avons trouvé quelques codes de haut rendement qui battent les codes triviaux. Pour les valeurs de p_s supérieures ou égales à 0.3, nous n'avons pas pas trouvé de meilleurs codes que les codes triviaux ! Le codage par syndrome est donc d'autant plus efficace que la source est asymétrique.

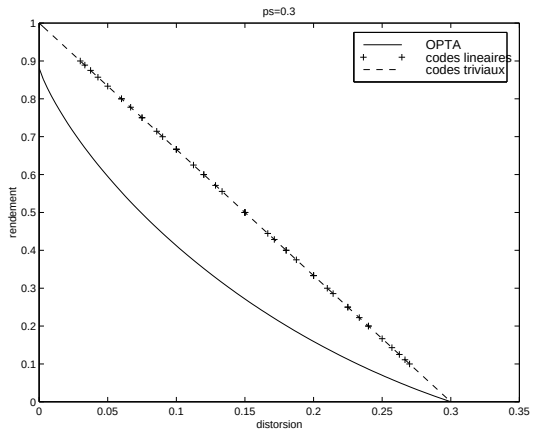


(a) : $p_s = 0.1$

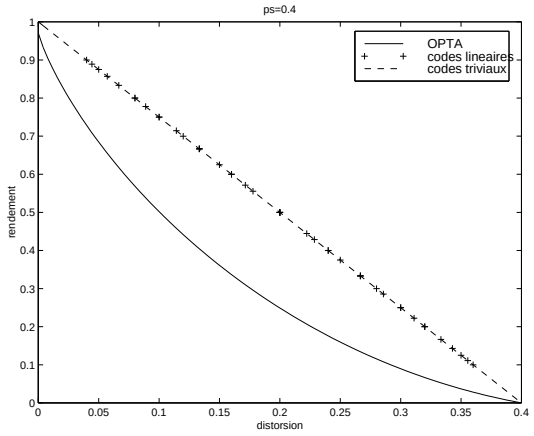


(b) : $p_s = 0.2$

Fig. 3.5 – Performances du codage par syndrome de SBA $p_s = 0.1$, 0.2.



(a) : $p_s = 0.3$



(b) : $p_s = 0.4$

Fig. 3.6 – Performances du codage par syndrome de SBA $p_s = 0.3$, 0.4.

3.5 Compression d'une SBA par un codeur SBS

Le codage de SBA par syndrome avec des codes de petite longueur permet de coder plus efficacement les sources fortement asymétriques. Nous cherchons donc un moyen efficace de comprimer avec pertes les sources binaires faiblement asymétriques. Le codeur de source exposé en 1.2.1 permet d'atteindre les performances optimales

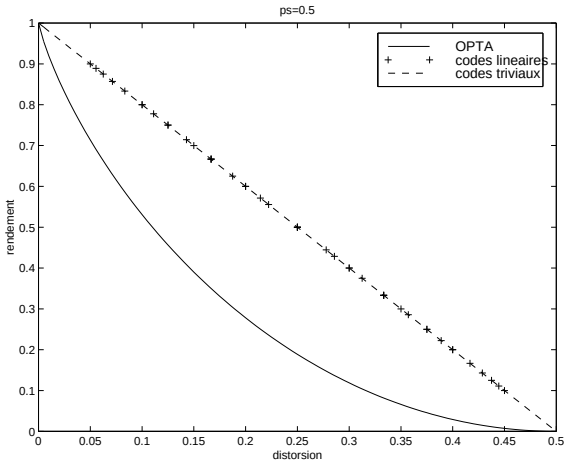


Fig. 3.7 – Performances du système d'Ancheta pour $p_s = 0.5$.

lorsque la source est symétrique. Il est donc naturel de calculer la distorsion des codes linéaires appliqués à une SBA, le codeur de source étant le décodeur de canal associé au canal binaire symétrique : ce schéma de compression est rappelé sur la figure 3.8.

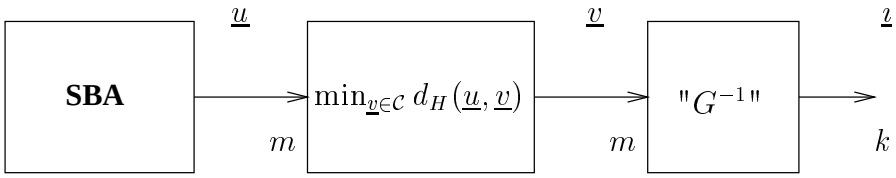


Fig. 3.8 – Compression d'une SBA par le codeur utilisé pour comprimer la SBS.

3.5.1 Performances

Le code linéaire utilisé pour la compression de la SBA est supposé être de longueur m et de dimension k. Le rendement R_s, exprimé en nombre de bits par symbole de source, est alors

$$R_s = \frac{k}{m}.$$

La distorsion moyenne par bit, toujours notée D, dépend du poids des translatés du code et du poids des chefs de classe (en utilisant une nouvelle fois les propriétés des cellules de Voronoï) :

$$\begin{aligned} D &= \frac{1}{m} E[d_H(\underline{u},\underline{v})] \\ &= \frac{1}{m} \sum_{\underline{v}\in\mathcal{C}} \sum_{\underline{u}\in V(\underline{v})} p(\underline{u}) d_H(\underline{u},\underline{v}) \end{aligned}$$

$$= \frac{1}{m} \sum_{\underline{e} \in V(\underline{0})} w_H(\underline{e}) \sum_{\underline{v} \in \mathcal{C}} p(\underline{u} = \underline{v} + \underline{e}). \tag{3.9}$$

Notons $A_{\mathcal{C}}$ le polynôme énumérateur de poids du code $\mathcal{C}$:

$$A_{\mathcal{C}}(z) = \sum_{\underline{v} \in \mathcal{C}} z^{w_H(\underline{v})},$$

et $A_{\underline{e}+\mathcal{C}}$ le polynôme énumérateur de poids du translaté du code $\mathcal{C}$ par le mot $\underline{e}$ ($A_{\underline{e}+\mathcal{C}} = \sum_{\underline{v} \in \mathcal{C}} z^{w_H(\underline{v}+\underline{e})}$). L'expression (3.9) se récrit alors

$$D = \frac{(1-p_s)^m}{m} \sum_{e \in V(\underline{0})} w_H(\underline{e}) A_{\underline{e}+\mathcal{C}} \left(\frac{p_s}{1-p_s} \right). \tag{3.10}$$

Lorsque le code choisi est un code trivial de rendement R_s, la distorsion est alors égale à $D = (1-R_s)p_s$. Dans le cas général, l'évaluation exacte de cette distorsion est difficile car elle nécessite le calcul de 2^{m-k} polynômes énumérateurs. Bien sûr, lorsque le rendement R_s est élevé, il est possible de tirer avantage de l'identité de McWilliams [43] liant le polynôme énumérateur d'un code à celui de son dual

$$A_{\mathcal{C}^\perp}(z) = \frac{(1+z)^m}{2^k} A_{\mathcal{C}} \left(\frac{1-z}{1+z} \right). \tag{3.11}$$

En effet, une fois le polynôme $A_{\mathcal{C}}$ calculé à partir du polynôme $A_{\mathcal{C}^\perp}$, le polynôme $A_{\underline{e}+\mathcal{C}}$ est déterminé par le polynôme énumérateur du dual du code engendré par les vecteurs générateurs du code $\mathcal{C}$ et le vecteur $\underline{e}$.

L'inégalité de Sullivan [61], généralisée aux codes linéaires q-aires par Kløve [28], permet de donner une borne supérieure de la distorsion ne dépendant que du polynôme $A_{\mathcal{C}}$. Cette inégalité stipule que, si $\underline{e}$ est un chef de classe non nul, alors

$$A_{\underline{e}+\mathcal{C}}(z) \leq \frac{(1+z)^{k+1} - (1-z)^{k+1}}{(1+z)^{k+1} + (1-z)^{k+1}} A_{\mathcal{C}}(z), \quad 0 \leq z \leq \frac{1}{2}. \tag{3.12}$$

Nous réinjectons cette inégalité dans l'équation (3.10) afin d'obtenir une borne nouvelle :

$$\begin{aligned} D &\leq \frac{(1-p_s)^m}{m} \frac{1-(1-2p_s)^{k+1}}{1+(1-2p_s)^{k+1}} A_{\mathcal{C}} \left(\frac{p_s}{1-p_s} \right) \sum_{\underline{e} \in V(\underline{0})} w_H(\underline{e}), \\ &\leq \frac{1}{m 2^{m-k}} \frac{1-(1-2p_s)^{k+1}}{1+(1-2p_s)^{k+1}} A_{\mathcal{C}^\perp}(1-2p_s) \sum_{\underline{e} \in V(\underline{0})} w_H(\underline{e}). \end{aligned} \tag{3.13}$$

où la dernière inégalité provient de l'identité de McWilliams. Cette borne sera appelée borne de la distorsion de Sullivan, eu égard à celui qui a formulé l'inégalité dont elle découle. Remarquons que la borne de la distorsion donnée par l'inégalité (3.13) vaut exactement la distorsion lorsque la source est symétrique. Notons aussi que pour les codes triviaux, cette borne est mauvaise car le terme prédominant est alors $(2-2p_s)^{m-k}$. Cette borne est heureusement plus facile à évaluer que les performances exactes (qui nécessitent de calculer le tableau standard de décodage) : seuls la cellule de Voronoï du mot nul et le polynôme énumérateur du code dual doivent être calculés.

3.5.2 Exemple et recherche exhaustive de bons codes de petite longueur

Éclaircissons les calculs de performance par l'exemple introduit au premier chapitre.

Suite de l'exemple 1 *Lorsque le décodeur par tableau standard code une SBA, le code systématique de matrice génératrice G_{sys}, dont le rendement R_s est égal à $3/5$, produit une distorsion D égale à*

$$D = \frac{1}{5}(5p_s - 12p_s^2 + 12p_s^3 - 4p_s^4).$$

Cette distorsion est supérieure (resp. inférieure) à celle d'un code trivial de même rendement lorsque $p_s \leq 0.37$ (resp. $p_s \geq 0.37$). La borne (3.13), obtenue grâce à l'inégalité de Sullivan, devient

$$D \leq \frac{3}{5}\frac{1-(1-2p_s)^4}{1+(1-2p_s)^4}(1 - 5p_s + 12p_s^2 - 12p_s^3 + 4p_s^4).$$

Cet exemple nous invite à penser que les codes de faibles longueurs coderont difficilement les sources fortement asymétriques alors qu'il sera possible d'obtenir de bons résultats pour les source peu asymétriques. Cette impression est confirmée par une recherche exhaustive sur les codes linéaires systématiques de longueur inférieure ou égale à 10 pour deux valeurs du paramètre p_s ($p_s = 0.1$ et $p_s = 0.4$). Le résultat de cette recherche est représentée sur les figures 3.9 et 3.10 : les performances des meilleurs codes linéaires trouvés sont indiquées par des croix alors que les performances des codes triviaux sont tracées en traits mixtes et l'OPTA en trait plein. Lorsque la source est fortement asymétrique ($p_s = 0.1$), les codes triviaux ne sont pas battus. Lorsqu'elle est faiblement asymétrique ($p_s = 0.4$), il existe des codes de faible longueur dont les performances sont supérieures à celles des codes triviaux. La borne de Sullivan relative à ces codes est tracée en pointillés. Remarquons que cette borne peut se révéler très proche de la distorsion effective du code. Sur la figure 3.10, la borne de Sullivan n'est pas représentée car les meilleurs codes trouvés sont triviaux : les valeurs alors prises par cette borne dépassent allègrement la valeur de p_s.

3.5.3 Compression par deux codes linéaires

L'un des moyens théoriques pour atteindre l'OPTA d'une SBA est de décomposer l'extension m^{e} de cette source en $m+1$ sous-sources [64]. La w^{e} sous-source émet des mots de poids w. À chaque sous-source, on associe un code en bloc qui :

– code la sous-source sans perte si le poids est faible ou proche de m ;

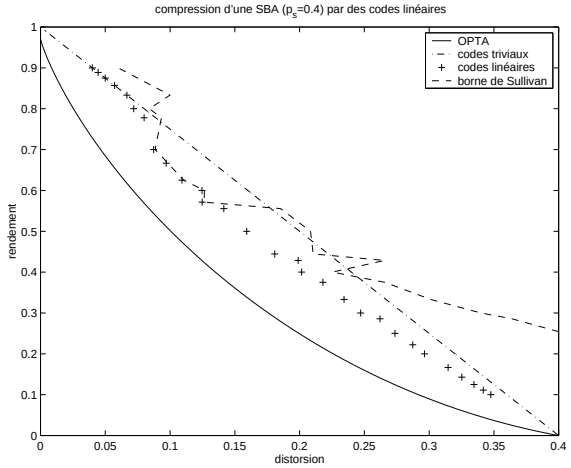


Fig. 3.9 – Meilleurs codes linéaires de dimension ≤ 10, $p_s = 0.4$ (source faiblement asymétrique), codage par région de Voronoï.

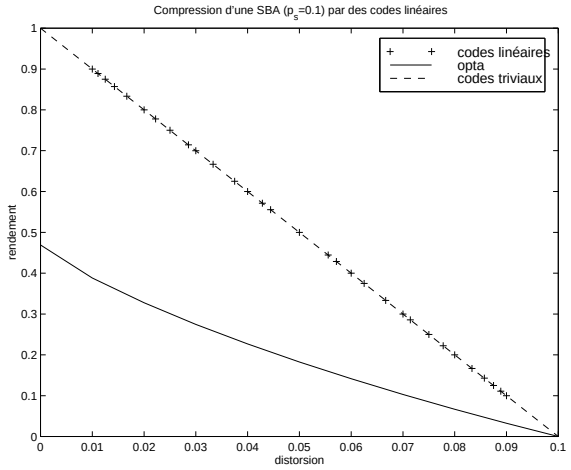


Fig. 3.10 – Meilleurs codes linéaires de dimension ≤ 10, $p_s = 0.1$ (source fortement asymétrique), codage par région de Voronoï.

– code la source avec une distorsion proche de $H_2^{-1}(H_2(w/m) - R)$ dans les autres cas [1].

Nous reprenons à notre compte l'idée de décomposition de la SBA en utilisant cette fois des codes linéaires, dont le codeur de source associé est moins complexe que celui associé à un code en bloc quelconque. Nous nous limitons à deux sous-sources, puisque nous nous restreignons à de petites longueurs, espérant ainsi ne pas trop nous éloigner du schéma asymptotiquement optimal par ces deux simplifications. Le codeur de source procède ainsi :

– si la SBA émet un mot $\underline{u}$ de poids inférieur à $m\delta$ (où δ est un réel positif), nous utilisons le code linéaire $\mathcal{C}_1$ de longueur m et de dimension k_1 pour comprimer la source ;

[1] une démonstration de l'efficacité de cette technique est donnée par Viterbi et Omura dans [64].

– si la SBA émet un mot $\underline{u}$ de poids strictement supérieur à $m\delta$, nous utilisons un code linéaire $\mathcal{C}_2$ de longueur m et de dimension k_2 pour comprimer cette sous-source.

La figure 3.11 illustre ce principe. Puisque nous souhaitons construire un code en bloc

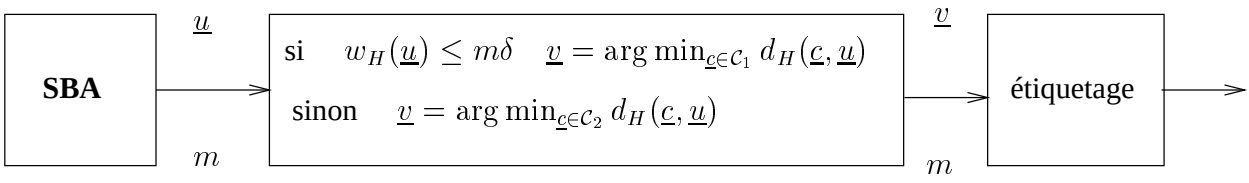


Fig. 3.11 – Compression d'une SBA par deux codes linéaires.

et non un code à longueur variable, un étiqueteur va associer une étiquette au mot de $\underline{v}$ reçu. Cette étiquette comportera au plus $k_1 + k_2$ bits : il est en effet possible que certains mots des codes $\mathcal{C}_1$ et $\mathcal{C}_2$ ne soient jamais utilisés en raison du seuil $m\delta$ introduit. Le calcul du rendement d'un tel schéma nécessite donc de compter les mots de chacun des codes réellement utilisés par le codeur. Si la recherche du mot de code le plus proche du mot de source émis demeure simple, l'étiqueteur, en revanche, est de complexité identique à celui d'un code non-linéaire de même longueur et de même taille. Ce codeur est donc une solution intermédiaire au niveau de la complexité entre le codeur associé à un code linéaire et le codeur associé à un code non-linéaire.

Nous menons une recherche sur les couples $(\mathcal{C}_1, \mathcal{C}_2)$ en nous restreignant aux codes linéaires systématiques de longueur inférieure ou égal à 8. Les figures 3.12 et 3.13 présentent les résultats de cette recherche respectivement pour une source très asymétrique ($p_s = 0.1$) et peu asymétrique ($p_s = 0.4$) : l'OPTA est tracée en trait continu, les performances des codes linéaires, qui sont un cas particulier de ce codeur ($\delta = 0$ ou $\delta = 1$), sont indiquées par des croix ($+$) et celles du codeur correspondant à deux vraies sous-sources ($0 < \delta < 1$) par des croix obliques ($\times$).

Si ce codeur n'apporte pas de véritable amélioration pour une source faiblement asymétrique, l'amélioration est notable pour la source fortement asymétrique : nous réussisons enfin à battre les codes linéaires triviaux et les codes adaptés au codage par syndrome pour des rendements élevés (supérieurs à 0.4 environ).

3.6 Algorithme de Lloyd pour la compression de SBA

L'amélioration des performances constatée lors de l'utilisation du codeur présenté au paragraphe 3.5.3 par rapport au codeur n'utilisant qu'un code linéaire peut s'expliquer par la non-linéarité introduite par le seuil : en effet le code de source résultant n'est pas *a priori* linéaire. Pour de petites longueurs, les codes non-linéaires semblent être de bons candidats pour comprimer la SBA, cependant leur utilisation entraîne une

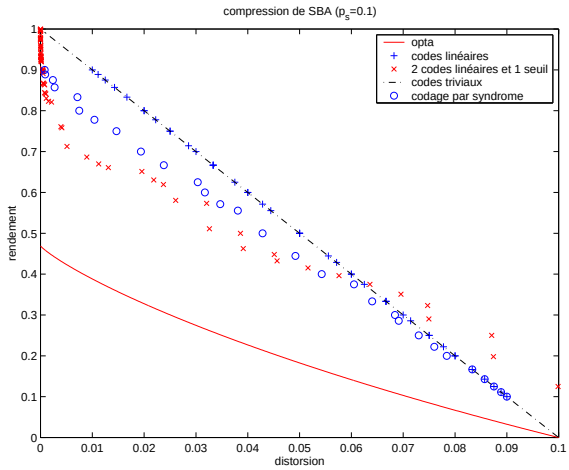


Fig. 3.12 – Comparaison des performances des codes linéaires (+), du codeur de la figure 3.11 (×) et du codeur par syndrome (o), $p_s = 0.1$ (OPTA en trait continu).

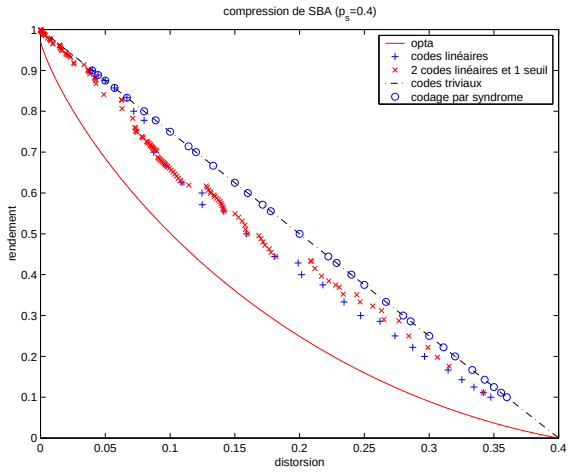


Fig. 3.13 – Comparaison des performances des codes linéaires (+), du codeur de la figure 3.11 (×) et du codeur par syndrome (o), $p_s = 0.4$ (OPTA en trait continu).

augmentation de complexité du codeur de source sauf dans le cas simple où le code non linéaire serait un code affine (i.e. un code linéaire translaté par un vecteur non nul, c'est-à-dire une classe du code linéaire distincte du code lui-même).

Nous allons donc rechercher de bons codes non linéaires. La recherche exhaustive semble prohibée car trop complexe, même pour des petites longueurs (il existe plus de dix millions de codes binaires non linéaires de longueur 5 et de taille 8 alors que la recherche portant sur les codes linéaires systématiques de mêmes caractéristiques se réduit dans ce cas à l'examen de vingt codes). L'algorithme de Lloyd tel qu'il a été développé au paragraphe 2.5.2 convient alors à notre tâche : il suffit de remplacer la densité de probabilité de la source $p(\underline{u})$ (supposée uniforme dans le paragraphe 2.5.2) par celle d'une SBA. Le codeur optimisé par l'algorithme de Lloyd est représenté sur

la figure 3.14. Il reste à préciser l'initialisation de cet algorithme.

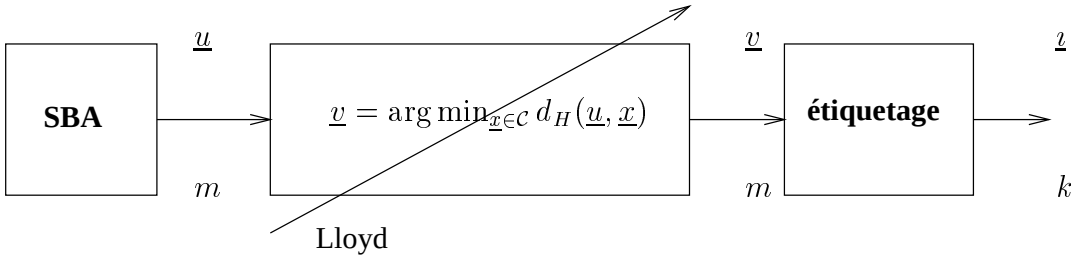


FIG. 3.14 – Compression d'une SBA par un code en bloc optimisé par l'algorithme de Lloyd.

3.6.1 Initialisation de l'algorithme de Lloyd

L'initialisation de l'algorithme de Lloyd consiste à lui fournir ou une partition de F_2^m, chaque ensemble de la partition étant assimilé à une région de Voronoï, ou un code de source. Nous choisissons la deuxième possibilité. Pour construire un code de longueur m à M éléments, nous tirons aléatoirement M vecteurs notés $\underline{v}_1, \cdots, \underline{v}_M$ de manière indépendante. Les composantes de ces vecteurs sont considérées comme des variables aléatoires indépendantes et identiquement distribuées. Le choix de leur loi de probabilité provient du calcul de la fonction rendement-distorsion.

Fixons la distorsion D tel que $R_{SBA}(D) = \frac{\log_2 M}{m}$. Rappelons que la fonction rendement-distorsion est déterminée par l'égalité (1.31)

$$R_{SBA}(D) = \inf_{m\in\mathbb{N}} \min_{p(\underline{v}|\underline{u})/E[d(\underline{U},\underline{V})]\leq mD} \frac{1}{m} I(\underline{U}, \underline{V})$$

où $\underline{u}$ est le mot de source et $\underline{v}$ le mot de source reconstruit. Puisque la SBA est une source sans mémoire et que la distance de Hamming entre deux mots est la somme des distances de Hamming sur chaque composante, la fonction rendement-distorsion peut être obtenue dans le cas monodimensionnel ($m = 1$) [41] :

$$R_{SBA}(D) = \min_{p(\underline{v}|\underline{u})/E[d(\underline{U},\underline{V})]\leq D} I(\underline{U}, \underline{V}).$$

En utilisant le lien entre entropie et information mutuelle (1.9), nous obtenons

$$I(\underline{U}, \underline{V}) = H_2(p_s) - H_2(\underline{U}|\underline{V}).$$

La fonction rendement-distorsion est donc obtenue en maximisant l'entropie conditionnelle $H_2(\underline{U}|\underline{V}) = H_2(\underline{U} - \underline{V}|\underline{V})$. Cette dernière est maximale lorsque la variable aléatoire $\underline{U} - \underline{V}$ est indépendante de $\underline{V}$ or le poids de $\underline{U} - \underline{V}$ mesure la distorsion donc :

$$p(u - v = 0) = 1 - D \quad \text{et} \quad p(u - v = 1) = D.$$

Par conséquent, la loi de probabilité de $\underline{v}$ qui permet d'atteindre l'OPTA est

$$p(v=0) = \frac{1-p_s-D}{1-2D} \quad \text{et} \quad p(v=1) = \frac{p_s-D}{1-2D}. \tag{3.14}$$

Nous utilisons cette distribution pour tirer aléatoirement les composantes des mots formant le code non linéaire servant à initialiser l'algorithme de Lloyd, espérant ainsi approcher le plus possible la distorsion minimale à rendement fixé.

3.6.2 Résultats

Nous faisons varier la longueur m des codes de 3 à 10, la taille du code est une puissance de 2 ($M = 2^k$, $1 \leq k \leq m$). À longueur et à taille fixées, nous lançons l'algorithme de Lloyd pour 100 initialisations tirées suivant la loi donnée par l'équation (3.14). Nous ne conservons que les meilleurs résultats, c'est-à-dire les codes de plus faible distorsion et de plus haut rendement : les meilleures performances sont indiquées par des losanges sur les figures 3.15 et 3.16. Sur ces figures sont aussi représentées les performances des différents systèmes reposant sur des codes linéaires exposés dans ce chapitre et l'inéluctable OPTA.

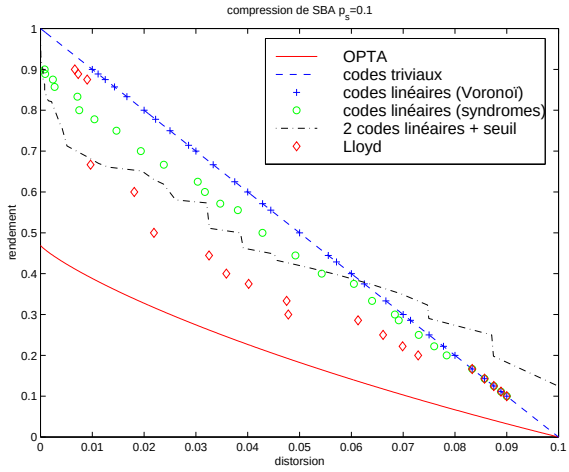


Fig. 3.15 – Comparaison des performances des codes obtenus par l'algorithme de Lloyd avec les autres systèmes, $p_s = 0.1$.

Lorsque la source est fortement asymétrique, les codes obtenus par l'algorithme de Lloyd battent très largement les codes linéaires, que l'on code par syndrome ou par région de Voronoï sauf dans le cas peu intéressant des rendements très élevés et largement supérieurs à l'entropie de la source [2]. Lorsque la source est peu asymétrique,

[2] dans le cas de la SBS, nous avions établi une borne de la distorsion, D_{inf}, valable pour les codes linéaires et non linéaires. Cette simple expérience montre que dans le cas d'une SBA, une borne inférieure de la distorsion valable pour les codes linéaires a peu de chance d'être valable pour les codes non linéaires à moins de sous-estimer largement la distorsion engendrée par les codes linéaires.

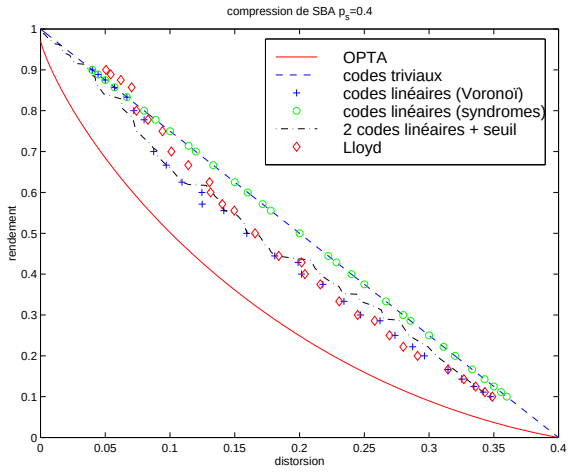


FIG. 3.16 – Comparaison des performances des codes obtenus par l'algorithme de Lloyd avec les autres systèmes, $p_s = 0.4$.

les codes obtenus ont des performances comparables à celles des codes linéaires, parfois légèrement meilleures, parfois un peu en retrait.

3.7 Remarque sur le codage conjoint SBA-CBS

Nous présentons tout d'abord dans cette section un système de codage conjoint SBA-CBS dû à Massey. Ce dernier, dans [37], a réutilisé la technique de codage par syndrome afin de réaliser un codeur conjoint SBA-CBS, représenté sur la figure 3.17.

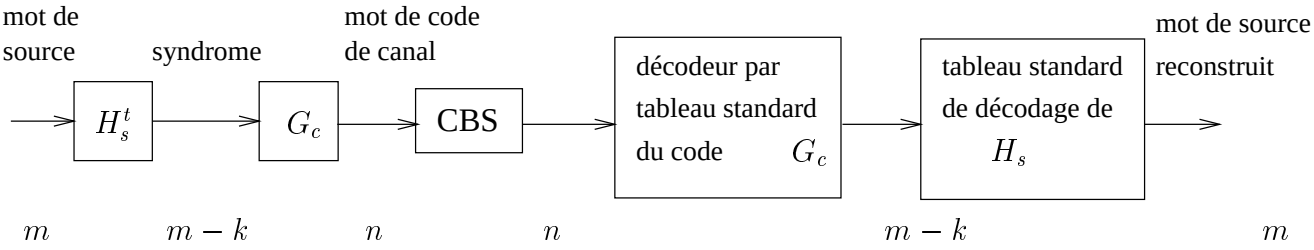


FIG. 3.17 – Schéma de codage conjoint SBA-CBS d'Ancheta-Massey.

Le système de codage conjoint d'Ancheta-Massey utilise deux codes linéaires binaires. Le mot de source $\underline{u}$, de dimension m, est comprimé par son syndrome $\underline{s} = \underline{u}H_s^t$, de dimension $m-k$ (H_s est la matrice de parité associé au codeur de source par syndrome). Ce syndrome est alors utilisé comme mot d'information par la matrice génératrice G_c, définissant le code de canal de dimension $m-k$ et de longueur n, pour produire le mot de code de canal. Ce codeur est conjoint dans la mesure où la matrice $H_s^t G_c$ peut être implémentée directement.

Le décodeur global, non présenté par Massey dans [37], utilise *a priori* la séparabilité du codeur : le décodeur de canal utilisant le tableau standard de décodage fournit une estimée du syndrome et le décodeur de source fournit au destinataire le chef de classe associé à cette estimée.

Le rendement global R de ce système est

$$R = \frac{R_s}{R_c} = \frac{\frac{m-k}{m}}{\frac{m-k}{n}} = \frac{n}{m}$$

Massey montre dans [37] qu'il est possible d'atteindre un rendement proche de $H_2(p_s)/(1-H_2(p))$ et une distorsion globale D aussi petite que nécessaire avec un tel système (en s'autorisant des codes de longueurs suffisamment grandes). Dans le cas où la distorsion autorisée serait non nulle, le schéma de codage conjoint devient sous-optimal, c'est-à-dire qu'il ne peut atteindre le rendement minimum défini par l'OPTA.

Une formule concise des performances de ce système conjoint est malheureusement difficile à obtenir en raison de la non-linéarité du décodeur de source et du décodeur de canal. D'après l'étude menée pour le codage de source par syndrome, un tel système de codage conjoint est adapté aux sources fortement asymétriques.

Nous n'avons pas mené davantage d'investigation sur le codage conjoint SBA-CBS faute de temps mais ce dernier est possible en appliquant le système source ou canal décrit au précédent chapitre, à condition que la source soit faiblement asymétrique : les performances du codage de source par région de Voronoï sont dans ce cas suffisamment bonnes pour envisager un tel système. Lorsque la source est fortement asymétrique, le recours aux codes non linéaires s'impose pour concevoir un système source ou canal efficace.

3.8 Conclusions

Après avoir rappelé l'OPTA correspondant à une source binaire asymétrique et au critère de fidélité défini par la distance de Hamming, nous avons examiné différents systèmes. Nous avons d'abord proposé le codeur par couronne ou par boule qui, en fonction du poids du mot de source, transmet le mot tel quel ou le projette (sur une couronne ou sur une boule). Ce codeur, assez simple, permet d'atteindre asymptotiquement l'entropie de la SBA presque sans perte. Dans le cas où la distorsion désirée est non nulle, ce codeur ne peut cependant pas atteindre l'OPTA.

Nous avons donné les performances du codage par syndrome, développé par Ancheta qui a montré son optimalité dans le cas d'un compression presque sans perte. Une recherche exhaustive nous a permis de montrer que le codage par syndrome est d'autant

plus performant que la source est asymétrique, lorsque la longueur est fixée. Les codes performants pour le codage par syndrome ne semblent pas être les mêmes que ceux pour le codage de SBS ou de canal binaire symétrique.

Nous avons ensuite utilisé le codage par régions de Voronoï définies par un code linéaire. Nous avons mis en évidence une borne supérieure des performances de ce système sans pour autant pouvoir juger de sa finesse. À longueur fixée, ces codes sont d'autant plus performants que la source est peu asymétrique. Dans ce cas, la distorsion obtenue est légèrement inférieure à celle observée en comprimant une SBS (par le même code).

Nous avons obtenu des codes *a priori* non linéaires en divisant la SBA en deux sous-sources et en utilisant un code linéaire pour chacune des sous-sources, ce qui améliore les performances car le codeur est plus flexible. La recherche du mot de code le plus proche du mot de source émis demeure simple (en raison de l'utilisation des codes linéaires) mais l'étiqueteur est de complexité équivalente à celle d'un code non linéaire. Ce sytème est plus flexible, à longueur de code fixée, en terme de rendement.

Enfin, l'algorithme de Lloyd, initialisé par un code tiré au hasard suivant une loi de probabilités provenant de l'OPTA, a permis d'obtenir de bons codes non linéaires. Les performances des divers systèmes étudiés montrent un compromis entre complexité et performances dans le cas de la SBA : à longueur et taille du code fixées, la recherche du mot de code le plus proche et l'étiquetage sont relativement simples, ce qui n'est pas le cas pour les codes non linéaires où une recherche exhaustive est nécessaire pour trouver le mot de code le plus proche et où l'étiquetage est une correspondance sans aucune structure *a priori*. Le codage par syndrome et celui par région de Voronoï sont complémentaires : le premier permet de coder des sources fortement asymétriques alors que le second comprime mieux des sources peu asymétriques. Il est aussi apparu qu'à petite longueur fixée, il est plus facile de coder par des codes linéaires une SBS qu'une SBA dans le sens où l'écart entre les performances des codes linéaires pour la compression de SBS et $R_{SBS}(D)$ est inférieur à l'écart entre les performances de ces codes pour la compression de SBA et $R_{SBA}(D)$.

Chapitre 4

Codage de source par code arithmétique

4.1 Introduction

Les codes arithmétiques [1] se sont développés en même temps que les codes binaires : dès 1955, Diamond les proposa, ils devinrent un sujet important de recherche jusque dans le milieu des années 1970. Brown, Peterson, Mandelbaum, Massey, Chien, Preparata, pour ne citer que quelques-uns, s'illustrèrent dans leur étude. La nécessité de construire des unités de calcul arithmétique fiables motiva leur développement [38] même si d'autres possibilités d'amélioration, comme la qualité des composants et la mise en place de calculateurs en parallèle effectuant les mêmes opérations, furent aussi étudiées. Là encore, les codes arithmétiques furent conçus dans le but de corriger des erreurs mais par principe de dualité, ils peuvent être utilisés pour comprimer des entiers : la source est alors dite arithmétique. Or pour un calculateur, qu'est-ce qu'un entier sinon des bits ? Nous verrons donc dans ce chapitre la compression de source arithmétique et de source binaire symétrique par ces codes. Nous étendons ainsi le principe de dualité exposé au premier chapitre à une source un peu moins académique que la SBS, montrant ainsi sa généralité. Auparavant, nous rappellerons la définition des codes arithmétiques et leurs propriétés.

[1] les codes arithmétiques utilisés ici ne doivent pas être confondus avec le « codage de source arithmétique » qui est à longueur variable. Ici les longueurs sont fixes.

4.2 Définition et propriétés des codes arithmétiques

4.2.1 Décomposition non-adjacente et poids arithmétique

Avant de définir un code arithmétique, nous allons introduire le **poids arithmétique** et la **décomposition non-adjacente**. Considérons une unité de calcul arithmétique (i.e. une unité effectuant des opérations sur des entiers). Cette unité travaille en pratique sur une représentation binaire des nombres. Ainsi lors d'une addition, le calculateur doit ajouter trois bits pour calculer chaque bit de la somme (un bit par entier à ajouter et la retenue) et calculer la nouvelle retenue. Ainsi, si une seule erreur se produit lors d'une somme, la différence entre le résultat erroné fourni par le calculateur et le vrai résultat sera **en valeur absolue** une puissance de deux. Les codes arithmétiques doivent pouvoir corriger sinon détecter ce type d'erreur. La forme non-adjacente permet de fournir une représentation des entiers adéquate.

Définition 4.2.1 (décomposition binaire non-adjacente, [38]) *La décomposition binaire non-adjacente d'un entier N est la suite d'entiers relatifs $(b_i)_{i\in\mathbb{N}}$ vérifiant les propriétés suivantes :*

- $N = \sum_{i\in\mathbb{N}} b_i 2^i$;
- $\forall i \in \mathbb{N} \quad |b_i| < 2$ *(b_i vaut donc -1, 0 ou 1) ;*
- $\forall i \in \mathbb{N} \quad b_i b_{i+1} = 0$ *(lorsque l'on examine deux coefficients consécutifs, l'un d'eux est nul).*

Cette décomposition est unique.

Cette décomposition, à la différence de la décomposition binaire classique, permet donc de prendre en compte les erreurs potentielles du calculateur décrites au paragraphe précédent. Signalons qu'à l'instar des décompositions habituelles qui existent dans une base quelconque, la décomposition non-adjacente est généralisable, elle aussi [10]. La décomposition binaire non-adjacente se calcule aisément à partir des décompositions binaires de $3N$ et N [8].

Exemple 2 *Prenons $N = 14$ (donc $3N = 42$). Décomposons N et $3N$ en base 2 en adoptant volontairement une notation vectorielle pour la décomposition binaire :*

$$N = (0,0,1,1,1,0)\,(= 0\times 32 + 0\times 16 + 8 + 4 + 2 + 0\times 1) \text{ et } 3N = (1,0,1,0,1,0)$$

Soustrayons N à $3N$ en effectuant les opérations bit à bit en suivant les règles de calcul suivantes (différentes de celles du corps F_2) : $1-0 = 1$, $1-1 = 0-0 = 0$ et $0-1 = -1$. Ces règles de calcul sont celles que l'on applique en considérant la décomposition binaire comme un vecteur d'entiers relatifs. Nous avons donc

$$3N - N = (1,0,0,-1,0,0)$$

Cette dernière équation signifie que $28 = 32 - 4$. *Or la décomposition binaire adjacente de* $2N$ *diffère, par définition, de celle de* N *par une multiplication par deux (un décalage) donc*

$$N = (0, 1, 0, 0, -1, 0) = 16 - 2$$

Définition 4.2.2 (poids arithmétique) *Le nombre de termes non nuls dans la décomposition binaire non-adjacente d'un entier* N *s'appelle poids arithmétique de* N, *noté* $w_A(N)$.

Le poids arithmétique peut se calculer sans passer par la décomposition non-adjacente : Chang et Reed [50] ont proposé un algorithme récursif simple réalisant ce calcul.

Si l'on supprime la dernière propriété vérifiée par les $(b_i)_{i\in\mathbb{N}}$, à savoir $b_i b_{i+1} = 0$, on obtient une décomposition binaire dite modifiée.

Propriété 1 ([50]) *Toute décomposition binaire modifiée de* N *comporte un nombre de termes non nuls supérieur ou égal au poids arithmétique de* N.

Cette propriété justifie l'étude du poids arithmétique. Remarquons que ce dernier possède toutes les propriétés exigées pour être qualifié de norme. Il est donc naturel de définir la **distance arithmétique** entre deux entiers N_1 et N_2, distance notée $d_A(N_1, N_2)$ comme le poids de la différence :

$$d_A(N_1, N_2) = w_A(N_1 - N_2).$$

4.2.2 Résumé de la théorie des codes arithmétiques

Deux grandes familles de codes arithmétiques existent : les codes dits AN et les codes à résidus. À première vue, ces deux familles sont bien distinctes, il existe cependant une relation forte entre les deux. Nous examinerons les codes AN en détail avant de définir les codes à résidus. De plus, bien des notions du codage correcteur d'erreurs classique ont leurs analogues dans le domaine des codes arithmétiques : nous essayerons de souligner ces analogies.

Les codes AN

Définition 4.2.3 (Code AN) *Soient* A *et* B *deux entiers naturels. Le code* AN *est l'ensemble des multiples de* A *positifs ou nuls strictement inférieurs à* AB *(i.e.* $\{AN, 0 \leq N < B\}$*).*

Le produit AB est noté m et B est, bien entendu, la taille du code. Au sens strict du terme, un code AN n'est pas linéaire. Cependant, pour peu que $N_1 + N_2 < B$, $A(N_1 + N_2)$ est un mot du code : un code AN est donc linéaire si on le considère comme un idéal de $\mathbb{Z}_m = \mathbb{Z}/\mathbb{Z}_m$. Le décodeur associé à un code arithmétique est supposé trouver l'élément du code le plus proche (au sens de la distance arithmétique) d'un entier $N' = AN + E$ compris entre 0 et $AB - 1$ (i.e $\arg\min_{0 \leq N < B} w_A(N' - AN)$) : *a priori*, cette recherche est menée de manière exhaustive. Par souci de clarté, nous qualifions ce décodeur de décodeur arithmétique. À l'instar des codes construits sur les corps finis, un code AN admet une distance minimale notée $d_{A,\min}$ définie par

$$d_{A,\min} = \min_{0<N<B} w_A(AN)$$

Cette distance minimale gouverne le nombre d'erreurs corrigibles et le nombre d'erreurs détectables par le décodeur : un code AN donné peut corriger t erreurs et en détecter s si et seulement si $d_{A,\min} > 2t + s$.

Le décodeur peut tirer parti de la structure d'idéal du code pour essayer de retrouver l'erreur arithmétique : ce décodeur, dit modulaire, effectue alors ses opérations dans l'anneau $\mathbb{Z}_m$. Cependant, la distance arithmétique n'est pas adaptée aux calculs sur cet anneau car $w_A(N)$ et $w_A(N - m)$ sont *a priori* distincts alors que N et $N - m$ se confondent dans $\mathbb{Z}_m$! On définit alors le poids modulaire $w_M(N)$ d'un entier N par

$$w_m(N) = \min(w_A(N), w_A(m - N)) \tag{4.1}$$

Le poids modulaire n'est une norme (donc n'induit une distance) que pour des valeurs bien précises de m [38] : m doit être une puissance de 2 à laquelle on ajoute -1, 0 ou 1. Dans ces cas, la distance modulaire minimale du code correspond à $d_{A,\min}$ [38]. Puisque le code est un idéal de cet anneau, la structure quotient associée comporte A classes, contenant chacune B éléments, ces classes formant une partition de $\mathbb{Z}_m$. La classe d'un nombre N est déterminée par son syndrome, à savoir le reste de la division euclidienne de N par A, noté $N[A]$. Le chef d'une classe est défini comme l'élément de la classe de poids modulaire minimum. Le décodeur modulaire retranche donc (dans $\mathbb{Z}_m$) à N' son chef de classe noté $C(N')$. L'erreur arithmétique E, si elle est en valeur absolue plus petite que m, est donc égale à $C(N')$ ou à $C(N') - m$. Cette indétermination est le prix à payer pour une simplification du décodeur.

Les codes arithmétiques parfaits connus sont les suivants [38] :
- $A = 1$, B quelconque (code parfait universel) ;
- $B = 1$, A quelconque (code nul) ;
- si A est un nombre premier impair tel que -2 est primitif dans $\mathbb{Z}_A$ mais pas 2, $B = (2^{(A-1)/2} - 1)/A$, alors le code est parfait et corrige une erreur ($A = 23$ et $B = 89$ par exemple) ;
- si A est un nombre premier impair tel que 2 est primitif dans $\mathbb{Z}_A$, $B = (2^{(A-1)/2} + 1)/A$, alors le code est parfait et corrige une erreur ($A = 11$ et $B = 3$, par exemple).

Ces deux dernières familles de codes sont semblables aux codes de Hamming binaires qui sont parfaits et corrigent une erreur. De même que le code simplexe est le dual d'un

code de Hamming [43], le code BN où $B = (2^{(A-1)/2}+1)/A$ avec A un nombre premier impair strictement supérieur à 3 tel que 2 soit primitif dans $\mathbb{Z}_A$ peut être vu comme le dual du code AN : tous les éléments non nuls du code ont un poids arithmétique égal à $(A-1)/3$ ou à $(A+1)/3$ (un seul de ces nombres est entier). Ce poids arithmétique se confond alors avec la distance de Hamming minimale.

À l'instar des codes construits sur les corps finis, il est possible de définir des codes arithmétiques cycliques : le décalage cyclique de la décomposition binaire d'un élément du code arithmétique doit être la décomposition binaire d'un autre élément du code. Les codes tels que $m = 2^n - 1$ sont tous des codes cycliques. Massey et Garcia [38] définissent le rendement R d'un code cyclique par

$$R = \frac{\log_2 B}{\log_2 A + \log_2 B}\,.$$

Puisque A divise m, l'ordre de 2 dans $\mathbb{Z}_A$ divise n. Si n est strictement plus grand que cet ordre, le code contient alors un élément de poids arithmétique égal à 2. On impose donc de choisir n comme l'ordre de 2 dans $\mathbb{Z}_A$ afin d'obtenir une distance arithmétique minimale suffisante (un code cyclique non nul de distance minimale égale à 1 est forcément un code pour lequel A est une puissance de 2). Une famille bien connue de codes cycliques a été découverte par Mandelbaum et Barrows [38] : B est choisi parmi les nombres premiers pour lesquels 2 est primitif dans le corps fini associé et $A = (2^{B-1}-1)/B$. La distance minimale de tels codes est alors $\lfloor (B+1)/3 \rfloor$.

D'une part, nous disposons de codes cycliques à haut rendement corrigeant une erreur, d'autre part, les codes de Mandelbaum-Barrows ont un faible rendement et une distance assez grande. Chien, Hong et Preparata [9] ont, par ailleurs, répertorié des codes de rendement intermédiaire et de distance satisfaisante. Nous ne sommes donc pas restreints dans le choix d'un bon code arithmétique.

Codes à résidus

Définition 4.2.4 (Code à résidus) *Soit m un entier naturel et $m_1, \cdots, m_k$ k diviseurs de m. Le code à résidus dans $\mathbb{Z}_m$ engendré par les $(m_i)_{1\leq i\leq k}$ est l'ensemble des $(k+1)$-uplets s'écrivant $(N, N[m_1], \cdots, N[m_k])$ où N est un élément de $\mathbb{Z}_m$ et où $N[m_i]$ désigne le reste de la division euclidienne de N par m_i. N est appelé entier d'information et les $(N[m_i])_{1\leq i\leq k}$ les entiers redondants de N.*

Définition 4.2.5 (Code AN associé, [38]) *Le code AN associé est le code tel que A est le plus petit commun multiple des $(m_i)_{1\leq i\leq k}$, B vérifiant $AB = m$.*

La capacité de correction d'un code à résidus est identique à celle de son code AN associé car les entiers redondants d'un nombre N déterminent de manière unique le syndrome $N[A]$, réciproquement ce syndrome détermine exactement tous les entiers

redondants [38]. Cette propriété se démontre aisément en utilisant le théorème du reste chinois [5]. Ainsi les deux familles de codes arithmétiques sont étroitement liées. Nous nous limiterons donc par la suite de chapitre à l'étude des codes AN sans perte de généralité.

Codes arithmétiques et correction d'erreur au sens de Hamming

Nous terminons ce bref panorama de la théorie des codes arithmétiques en précisant le rapport entre distance arithmétique et distance de Hamming. Considérons deux entiers naturels N et N'. $d_H(N, N')$ représente le nombre de composantes distinctes dans les décompositions binaires de ces deux entiers : la différence bit à bit (calculée dans $\mathbb{Z}$ et non sur F_2) de ces deux nombres est une décomposition modifiée qui n'est pas nécessairement non-adjacente. Par conséquent, les deux distances considérées sont liées par l'inégalité suivante :

$$d_H(N, N') \geq d_A(N, N') \tag{4.2}$$

Les codes arithmétiques peuvent donc servir de code de canal comme l'ont souligné Massey et Garcia [38].

4.3 Compression d'une source arithmétique selon le critère de la distance arithmétique

4.3.1 Définition

Définition 4.3.1 source arithmétique symétrique. *Une source arithmétique symétrique de paramètre m, notée SAS, est une source émettant de manière équiprobable un entier compris entre 0 et $m-1$.*

La SAS peut donc être considérée comme une généralisation de la SqS : nous avions supposé que q était une puissance d'un nombre premier dans le premier chapitre afin de pouvoir utiliser les codes en blocs q-aires linéaires. Le fait d'utiliser d'autres outils (les codes arithmétiques dans notre cas) nous permet de nous affranchir de cette contrainte. La SAS est un moyen terme entre la SBS et les sources continues : une source continue quantifiée de manière équiprobable peut être ainsi considérée comme une SAS.

Par dualité, nous allons nous intéresser à la compression de la SAS par les codes AN, illustrée sur la figure 4.1. Nous choisirons le critère de la distance arithmétique, distance sur laquelle repose la construction des codes arithmétiques, pour mesurer la

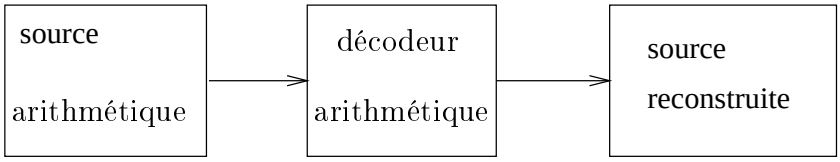


Fig. 4.1 – Compression d'une source arithmétique par un code arithmétique.

fidélité de la source reconstruite. La distorsion par bit d'information, notée $D_{SAS,A}$, s'écrit donc

$$D_{SAS,A} = \frac{1}{\log_2 m} E[d_A(N', AN)]. \tag{4.3}$$

4.3.2 OPTA

Dans le cas général, la fonction rendement-distorsion $R_{SAS,A}(D)$ associée à la SAS et au critère de distance arithmétique est difficilement calculable car la minimisation de l'information mutuelle doit prendre en compte une matrice de taille $m \times m$ dont les entrées sont les distances arithmétiques entre deux symboles de source. Elles n'ont pas *a priori* de formule exacte déterminée aisément. Par conséquent, nous pouvons recourir à l'algorithme de Blahut-Arimoto [13, 64], pour déterminer cette fonction. Cependant, la taille de la matrice des distances détermine la complexité de l'algorithme de Blahut-Arimoto : la valeur de m est donc limitée. Une autre solution est de remarquer que la distance arithmétique étant comme toute distance, au sens mathématique du terme, symétrique ($d_A(N, N') = d_A(N', N)$)[2], la fonction rendement-distorsion est alors définie par les équations paramétriques suivantes :

$$s \in \mathbb{R}^-,\ D = \frac{1}{\log_2 m} \frac{\sum_{N=0}^{m-1} w_A(N) 2^{s w_A(N)}}{\sum_{N=0}^{m-1} 2^{s w_A(N)}} \tag{4.4}$$

$$R_{SAS,A}(D) = sD + \frac{\log_2(A) - \log_2(\sum_{N=0}^{m-1} 2^{s w_A(N)})}{\log_2 m}, \tag{4.5}$$

où le rendement s'exprime en bits par symbole (ici des entiers) et où la distorsion D mesure une distorsion par bit. La fonction rendement-distorsion dépend donc du poids arithmétique des m premiers entiers naturels : cette fonction changera peu d'allure entre deux valeurs proches de m. La figure 4.2 montre la fonction rendement-distorsion obtenue pour $m = 33$ (en trait continu) et $m = 17000$ (en pointillés).

[2] nous sommes alors dans le cas d'une source symétrique avec distorsion symétrique (*symmetric source with balanced distortion measure*) comme défini dans [64].

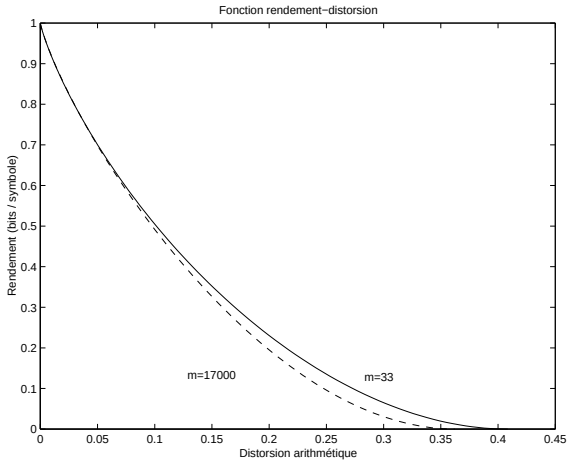


FIG. 4.2 – $R_{SAS,A}(D)$ pour $m = 33$ et $m = 17000$.

4.3.3 Distorsion et rendement

Le codeur de source (équivalent à un décodeur de canal par dualité) associé au code arithmétique AN reçoit un nombre N'. Afin de minimiser la distorsion $D_{SAS,A}$, le codeur de source doit trouver l'élément de code le plus proche, au sens de la distance arithmétique, de N', c'est-à-dire calculer la valeur suivante :

$$\arg \min_{N, 0 \leq N < B} d_A(N', AN).$$

Le codeur est donc contraint à réaliser une recherche exhaustive sur tous les éléments du code : il est possible de définir des cellules de Voronoï mais elles ne sont plus égales à une translation près et ne contiennent plus forcément le même nombre d'éléments.

La distorsion $D_{SAS,A}$ par bit d'information a donc pour expression

$$D_{SAS,A} = \frac{1}{m \log_2 m} \sum_{N'=0}^{m-1} \min_{0 \leq N < B} d_A(N', AN). \tag{4.6}$$

Le rendement R exprimé en nombre de bits « codés » par bit d'information s'écrit

$$R == \frac{\log_2 B}{\log_2 m} = \frac{\log_2 B}{\log_2 A + \log_2 B}\,.$$

4.3.4 Recherche des meilleurs codes

Nous supposons à présent que le paramètre m de la source est fixé et nous cherchons à déterminer les meilleurs couples (A, B) pour comprimer une telle source. Nous

restreignons le domaine de recherche en imposant à A et B de vérifier les conditions suivantes :

$$2 \leq A < m \tag{4.7}$$

$$A(B-1) < m \leq AB \tag{4.8}$$

Une recherche exhaustive menée pour $m = 1023$ a produit les codes indiqués dans la table 4.1. La figure 4.3 montre les performances de ces derniers (elles sont indiquées par les croix (+)) ainsi que l'OPTA (en trait continu).

A	B	$D_{SAS,A}$	R	A	B	$D_{SAS,A}$	R
2	512	4.995817e-02	9.001269e-01	53	20	1.616064e-01	4.322537e-01
3	341	6.667607e-02	8.414814e-01	55	19	1.641483e-01	4.248526e-01
4	256	7.498613e-02	8.001128e-01	59	18	1.657125e-01	4.170513e-01
5	205	7.997217e-02	7.680563e-01	61	17	1.683522e-01	4.088039e-01
7	147	8.564257e-02	7.200687e-01	67	16	1.715784e-01	4.000564e-01
11	93	9.092191e-02	6.540081e-01	71	15	1.739248e-01	3.907441e-01
13	79	9.297498e-02	6.304669e-01	77	14	1.741203e-01	3.807892e-01
19	54	9.991633e-02	5.755699e-01	83	13	1.770533e-01	3.700961e-01
23	45	1.125281e-01	5.492627e-01	91	12	1.792041e-01	3.585468e-01
25	41	1.202516e-01	5.358307e-01	101	11	1.842880e-01	3.459919e-01
27	38	1.297348e-01	5.248667e-01	107	10	1.854611e-01	3.322396e-01
29	36	1.303214e-01	5.170654e-01	115	9	1.978774e-01	3.170372e-01
35	30	1.420533e-01	4.907582e-01	141	8	2.072628e-01	3.000423e-01
37	28	1.456706e-01	4.808033e-01	155	7	2.088271e-01	2.807751e-01
39	27	1.479192e-01	4.755558e-01	179	6	2.148886e-01	2.585327e-01
41	25	1.515365e-01	4.644511e-01	205	5	2.266204e-01	2.322255e-01
45	23	1.555449e-01	4.524200e-01	299	4	2.447070e-01	2.000282e-01
47	22	1.573047e-01	4.460060e-01	341	3	2.592741e-01	1.585186e-01
49	21	1.595533e-01	4.392937e-01	683	2	2.942741e-01	1.000141e-01

Tab. 4.1 – Table des meilleurs codes arithmétiques pour la compression de SAS suivant le critère de la distance arithmétique, $m = 1023$.

4.4 Compression d'une SBS par un code arithmétique selon le critère de distance de Hamming

Nous choisissons à présent le critère de la distance de Hamming pour mesurer la fidélité de la source reconstruite : la distance de Hamming entre deux entiers est mesurée en réalité entre les deux représentations binaires classiques des nombres (évitons toute méprise due à un abus de langage). Ce choix est naturel lorsqu'il s'agit de comprimer une SBS et est adapté aux codes arithmétiques en raison de l'inégalité (4.2). Les résultats que nous allons donner sont proches de ceux établis pour la distance arithmétique.

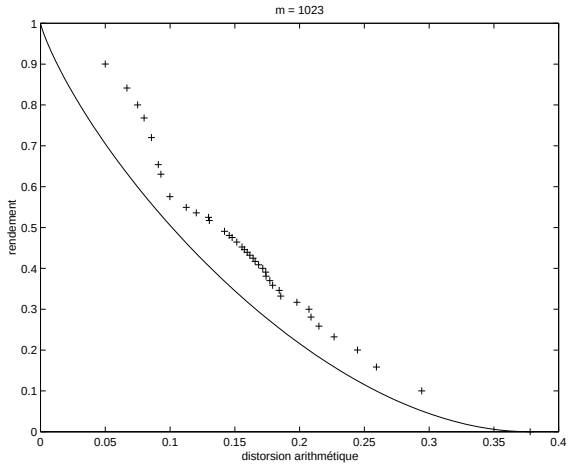


Fig. 4.3 – Meilleurs codes arithmétiques pour la compression d'une SAS de paramètre $m = 1023$.

La distorsion par bit d'information $D_{SAS,H}$ a donc pour expression

$$D_{SAS,H} = \frac{1}{\log_2 m} E[d_H(N', AN)]. \tag{4.9}$$

4.4.1 Distorsion et rendement

Le codeur de source (équivalent à un décodeur de canal par dualité) associé au code arithmétique AN reçoit un nombre N'. Afin de minimiser la distorsion $D_{SAS,H}$, le codeur de source doit trouver l'élément de code le plus proche, au sens de la distance de Hamming, de N', c'est-à-dire calculer la valeur suivante :

$$\arg \min_{N, 0 \leq N < B} d_H(N', AN).$$

Hélas, aucun lien n'apparaît entre ce codeur de source et les décodeurs arithmétique et modulaire décrits en 4.2. Une recherche exhaustive s'impose donc pour coder la source : il est possible de définir des cellules de Voronoï mais elles ne sont plus translatées les unes des autres et ne contiennent plus forcément le même nombre d'éléments.

La distorsion $D_{SAS,H}$ par bit d'information a donc pour expression

$$D_{SAS,H} = \frac{1}{m \log_2 m} \sum_{N'=0}^{m-1} \min_{0 \leq N < B} d_H(N', AN). \tag{4.10}$$

La formule du rendement, quant à elle, n'est pas modifiée :

$$R = \frac{\log_2 B}{\log_2 A + \log_2 B}.$$

Cette définition permet de comparer les performances des codes arithmétiques avec celles des codes en bloc q-aires linéaires et leurs bornes. L'OPTA est définie par $R_{SBS}(D) = 1 - H_2(D)$.

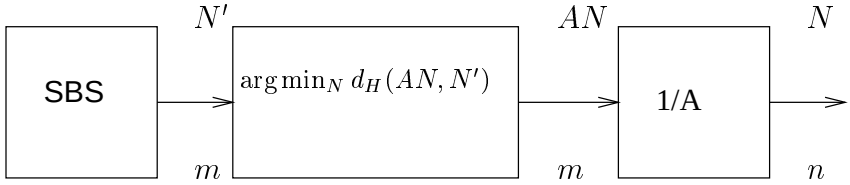


Fig. 4.4 – Compression d'une SBS par un code arithmétique.

4.4.2 Recherche de bons codes

D'après le paragraphe 4.2, les codes arithmétiques dont le paramètre m est une puissance de deux ont une distance arithmétique faible (égale à 1). La distance de Hamming minimale de tels codes est aussi égale à 1. Ces codes sont donc *a priori* mauvais : il est néanmoins possible de donner la formule exacte de leur distorsion $D_{SAS,H}$. Dans ce but, nous allons d'abord établir un lemme.

Lemme 4.4.1 *Soient a et b deux entiers naturel. Soient $A = 2^a$ et $B = 2^b$. Les cellules de Voronoï du code arithmétique AN à B éléments se déduisent par translation les unes des autres. La cellule de Voronoï $V(0)$ est composée des entiers positifs strictement inférieurs à A.*

Preuve

Introduisons la notion de support d'un entier. Le support d'un entier N de décomposition binaire $N = \sum_{i\in\mathbb{N}} b_i 2^i$ est l'ensemble des entiers i tels que b_i est non nul. Ce support est noté $\mathrm{Supp}(N)$. Le support est lié à la distance de Hamming par l'égalité suivante

$$d_H(N, N') = |\mathrm{Supp}(N)| + |\mathrm{Supp}(N')| - |\mathrm{Supp}(N) \cap \mathrm{Supp}(N')|. \tag{4.11}$$

Considérons un entier naturel N tel que $0 < N < A$. Alors $\mathrm{Supp}(N) \subset \{0, \cdots, a-1\}$ et $\mathrm{Supp}(AN) \subset \{a, \cdots, a+b-1\}$. Il est alors évident que

$$V(0) = \{0, 1, \cdots, A-1\}.$$

Considérons maintenant un entier N tel que $jA < N < (j+1)A$ où j est un entier naturel strictement positif. Alors $\mathrm{Supp}(N) = \mathrm{Supp}(N - jA) \cup \mathrm{Supp}(jA)$ et $\mathrm{Supp}(N - jA) \cap \mathrm{Supp}(jA) = \emptyset$. Calculons alors $d_H(N, lA)$ où l est un entier naturel strictement inférieur à B en appliquant l'égalité (4.11) :

$$\begin{aligned} d_H(N, lA) &= |\mathrm{Supp}(N)| + |\mathrm{Supp}(lA)| - |\mathrm{Supp}(N) \cap \mathrm{Supp}(lA)| \\ &= |\mathrm{Supp}(N - jA)| + |\mathrm{Supp}(jA)| + |\mathrm{Supp}(lA)| - |\mathrm{Supp}(jA) \cap \mathrm{Supp}(lA)| \\ &= |\mathrm{Supp}(N - jA)| + |\mathrm{Supp}(j)| + |\mathrm{Supp}(l)| - |\mathrm{Supp}(j) \cap \mathrm{Supp}(l)| \\ &\geq d_H(N, jA). \end{aligned}$$

Par conséquent, N appartient à $V(jA)$. Les cellules de Voronoï se déduisent les unes des autres par translation.

□

De ce lemme, on en déduit que la distorsion résultant de la compression d'une SBS par un code arithmétique AN avec m puissance de deux est donnée par

$$D_{SBS,H} = \frac{1-R}{2}.$$

Comme annoncé au début de paragraphe, ces codes sont mauvais du point de vue de la distorsion : ils sont le pendant des codes en bloc linéaires triviaux. D'un point de vue pratique, ces codes sont les plus adaptés à une représentation binaire des entiers puisque A et B sont des puissances de 2.

Pour trouver des codes arithmétiques convenant à la compression de SBS au sens de Hamming, nous considérons donc la n^{e} extension de la SBS et nous cherchons A et B tels que $AB \geq 2^n$ afin de « couvrir » suffisamment la source. A et B vérifient donc :

$$\begin{aligned} 2 \leq A &\leq 2^n - 1 \\ \frac{2^n}{A} \leq B &< \frac{2^n}{A} + 1. \end{aligned}$$

Nous menons cette recherche pour n variant de 3 à 17. Les performances des meilleurs codes arithmétiques trouvés sont représentées sur la figure 4.5 en pointillés. $R_{SBS}(D)$ est tracée en trait continu et la borne inférieure D_{inf} (1.34) en traits mixtes. Les valeurs exactes correspondant à ces codes sont fournies dans le tableau E.1. Nous illustrons ainsi le fait que la borne D_{inf} est valable pour les codes non linéaires. Les meilleurs codes trouvés restent proches de la borne calculée pour une longueur identique et une « dimension » $k = \log_2 B$. Les codes ainsi trouvés sont proches de la borne D_{inf}.

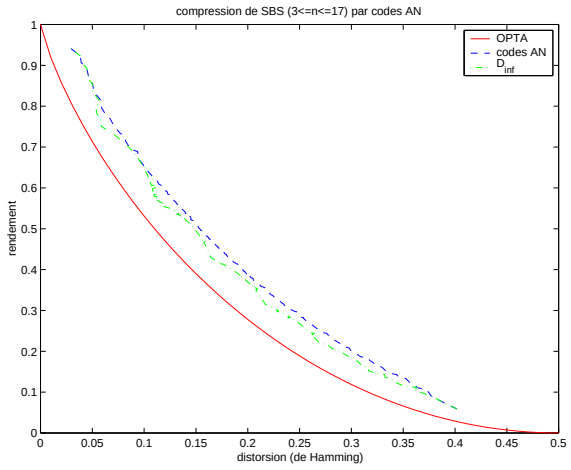


FIG. 4.5 – Compression d'une SBS par les codes arithmétiques.

R	p	Système séparé						Source ou canal				
		m	n	A_s	B	A_c	Distorsion	m	n	A	B	Distorsion
0.5	0.1	12	6	131	32	2	2.9635e-01	12	6	97	43	2.8829e-01
	0.01	12	6	131	32	2	2.1815e-01	12	6	67	62	1.8050e-01
	0.001	12	6	131	32	2	2.0942e-01	12	6	67	62	1.6769e-01
2	0.1	2	4	1	4	4	0.1	2	4	4	4	0.1
	0.01	2	4	1	4	4	0.01	2	4	4	4	0.01
	0.001	2	4	1	4	4	0.001	2	4	4	4	0.001

Tab. 4.2 – Comparaison du système séparé avec le système source ou canal lorsque les codes utilisés sont arithmétiques.

4.5 Codage conjoint SBS-CBS par des codes arithmétiques selon le critère de distance de Hamming

Nous avons étudié dans le deuxième chapitre ce codage conjoint en utilisant des codes linéaires binaires et établit que, lorsque la complexité du système global est faible, il vaut mieux choisir uniquement un codage de source ou un codage de canal suivant la valeur du rendement global R. Qu'en est-il si les codes linéaires sont remplacés par des codes arithmétiques ?

Pour répondre à cette question, nous fixons le rendement global $R = \frac{n}{m}$ et la valeur du paramètre p du CBS. Nous cherchons alors deux codes arithmétiques :

- le code A_sN de cardinal B comprimant la m^{e} extension de la SBS. A_s et B vérifient la relation $A_s(B-1) < 2^m \leq A_sB$;
- le code A_cN de cardinal B protègeant les bits d'information du code précédent des perturbations introduites par le canal. A_c et B vérifient la relation $A_c(B-1) < 2^n \leq A_cB$.

À rendement global fixé, nous examinerons donc plusieurs systèmes comprimant plus ou moins fortement la SBS en faisant varier A_s.

Parallèlement, nous cherchons le meilleur code arithmétique correspondant à la structure « source ou canal » exposée au deuxième chapitre :

- si le rendement global R est inférieur à 1, on cherche un code arithmétique AN tel que $A(B-1) < 2^m \leq AB$ et $B \leq 2^n$;
- si le rendement global R est supérieur à 1, on cherche un code arithmétique AN tel que $A(B-1) < 2^n \leq AB$ et $B \geq 2^m$.

Nous limitons les longueurs n et m à 12 et menons une recherche exhaustive pour deux valeurs du rendement global $R = \frac{1}{2}$ et $R = 2$. Les meilleurs systèmes obtenus et leurs performances sont indiqués dans la table 4.2. Ces performances sont également reportées sur les figures 4.6 et 4.7 où l'OPTA a été par ailleurs représentée.

On observe que le système source ou canal a des performances égales ou meilleures que celles du système séparé. Ce dernier se confond d'ailleurs avec le système source

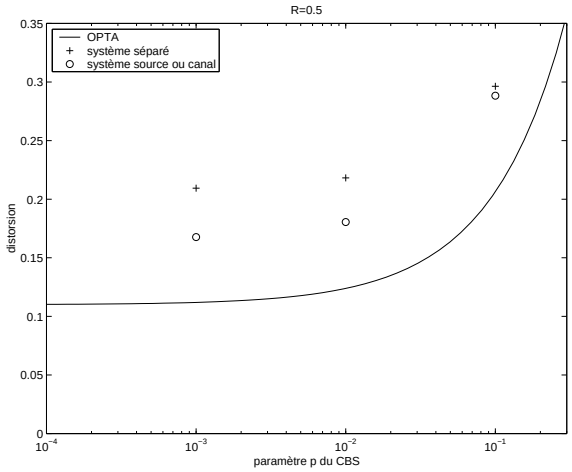


Fig. 4.6 – Comparaison du système séparé avec le système source ou canal, $R = 0.5$.

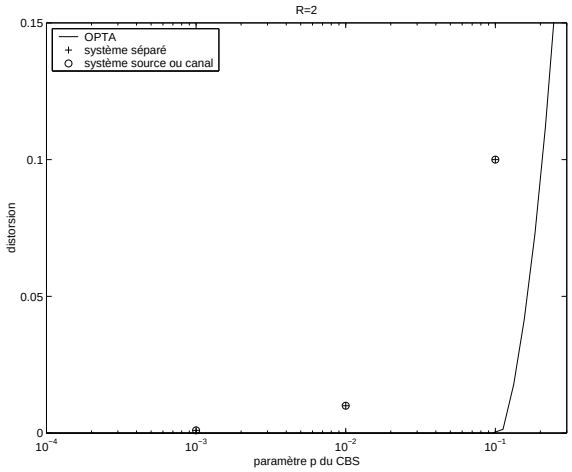


Fig. 4.7 – Comparaison du système séparé avec le système source ou canal, $R = 2$.

ou canal lorsque le rendement global est égal à 2. Le phénomène observé dans le cas des codes linéaires ne semble donc pas être spécifique à ces derniers : la structure « source ou canal » est préférable au système séparé. En raison des petites valeurs des paramètres A et B des codes arithmétiques considérés, les performances des deux systèmes restent éloignées de l'OPTA.

4.6 Conclusions

Nous avons donné dans ce chapitre un nouvel exemple de la dualité entre codage de source et codage de canal en considérant les codes arithmétiques. Nous avons rappelé brièvement la théorie sous-jacente à ces codes et le contexte dans lequel ils ont vu le jour. Les codes arithmétiques servent à combattre les erreurs pouvant se produire

dans un additionneur d'entiers, une erreur se confondant alors en valeur absolue avec une puissance de 2. La distance arithmétique permet alors de définir rigoureusement le poids de ces erreurs.

Par dualité, ces codes arithmétiques peuvent donc comprimer une source émettant des entiers suivant le critère de la distance arithmétique : nous avons donné quelques exemples d'une telle compression. Il est également possible d'utiliser ces codes pour comprimer une SBS en utilisant le critère de la distance de Hamming : la distance de Hamming entre les représentations binaires de deux entiers est supérieure ou égale à la distance arithmétique entre ces deux entiers. Le code arithmétique se comporte alors comme un code non linéaire dont les mots peuvent être énumérés simplement. Cependant le codeur de source associé à ces codes nécessite une recherche exhaustive du mot de code le plus proche, ce qui limite la valeur du cardinal du code en pratique. Nous avons alors cherché les meilleurs codes arithmétiques et comparé leur distorsion à la borne D_{inf}, obtenue au premier chapitre et valable pour les codes non linéaires.

Enfin, nous avons comparé les performances du système source ou canal et du système séparé lorsque ces deux derniers utilisent des codes arithmétiques : la distorsion du système source ou canal est inférieure à celle du système séparé à rendement global égal et à faible complexité. Ce résultat étend donc celui obtenu au deuxième chapitre où les codes utilisés étaient des codes linéaires binaires.

Chapitre 5

Dualité appliquée aux codes réels

Nous avons vu dans les chapitres précédents que la dualité qui existe entre codage de source et codage de canal a permis d'appliquer des outils de codage de canal tels que les codes linéaires pour la compression de sources discrètes. Cette même dualité nous a également permis d'employer l'algorithme de Lloyd pour créer des codes de canal. Nous nous proposons dans ce chapitre d'utiliser à des fins de codage de source les codes BCH réels, passant ainsi du domaine discret à celui du continu. La source retenue est soit une source gaussienne aux échantillons indépendants et identiquement distribués soit une image. Ces codes BCH s'avéreront difficiles à mettre en oeuvre pour une source quelconque. Nous utiliserons alors des quantificateurs vectoriels s'appuyant sur des codes linéaires réels et optimisés par algorithme de Lloyd pour comprimer une source gaussienne et la transmettre sur le canal binaire symétrique : ce système imposera une relation linéaire entre le code de source et le code de canal. Knagenhjelm [29], Hagen et Hedelin [25] ont montré qu'un tel type de relation permettait d'obtenir des codeurs de source robustes lorsque le code de canal choisi est universel.

5.1 Utilisation de la dualité : codes BCH réels et bruit impulsif (distance de Hamming)

5.1.1 État de l'art des codes réels

La définition de codes sur le corps des réels $\mathbb{R}$ ou des complexes $\mathbb{C}$ n'est pas une idée nouvelle : de nombreux outils bien connus en traitement du signal (transformée de Fourier discrète, transformée discrète en cosinus...) peuvent être considérés comme des codes réels ou complexes : Marshall [35, 36] semble avoir été le premier à introduire les codes tirés de la transformée de Fourier discrète et les codes BCH réels ou

complexes (des exemples antérieurs de codes BCH construits sur d'autres structures que les corps finis existent : Shankar [54] a construit de tels codes sur des anneaux d'entiers congruents). Wolf [67] a souligné que la transformée de Fourier permettait de supprimer le bruit impulsif du signal émis lorsque celui-ci est redondant (c'est-à-dire que certaines de ses composantes spectrales sont nulles). L'utilisation des codes réels tant en codage de canal qu'en codage de source demeure un sujet de recherche. Marshall semble être le premier à s'intéresser au double emploi de ces codes : il propose, dans [35], un codage source-canal combiné où la suppression de la redondance se fait par l'usage d'une matrice unitaire et où la transformée de Fourier est à la base de la correction d'erreurs. La correction d'erreurs à partir de transformée de Fourier ou de Hadamard est présentée plus complètement dans [36] : les concepts de correction d'erreur sur les corps finis sont transposés aux cas réel et complexe, introduisant alors les codes BCH réels. Shiu et Wu ont prolongé ce travail en définissant une nouvelle catégorie de codes réels linéaires en utilisant les propriétés de la transformée discrète en cosinus [68], codes pouvant combattre des bruits non additifs sous certaines hypothèses [56]. Ils ont ensuite proposé un nouvel exemple de la similarité entre codage sur les corps finis et codage sur $\mathbb{R}$ en construisant une classe de codes linéaires réels décodables par vote majoritaire [57]. Plus récemment, Gabay [23] a utilisé les codes BCH réels pour corriger les erreurs introduites par un canal binaire symétrique lors de transmission d'images.

5.1.2 Rappel sur les codes BCH binaires

Avant de définir à proprement parler les codes BCH réels, nous allons rappeler les propriétés essentielles des codes BCH binaires. Le lecteur désirant une étude plus complète peut se reporter aux ouvrages de McWilliams et Sloane [43], de Peterson et Weldon [50] ou à celui de Berlekamp [5].

Construction et propriétés des codes BCH

Définissons tout d'abord les codes cycliques.

Définition 5.1.1 *(***Code cyclique binaire***)* *Un code $\mathcal{C}$* ***cyclique binaire*** *de longueur n et de dimension k est un code linéaire binaire de longueur n et de dimension k invariant par décalage cyclique i.e. un idéal de l'anneau $F_2[x]/(x^n-1)$, anneau constitué des restes de la division euclidienne des polynômes à coefficients dans F_2 par le polynôme x^n-1. Autrement dit un code cyclique est un code linéaire tel que pour tout mot de code $\underline{c} = (c_0, \cdots, c_{n-1})$ le mot $(c_{n-1}, c_0, \cdots, c_{n-2})$ est encore un mot de code.*

Tout élément de F_2^n peut donc être considéré comme un polynôme à coefficients dans F_2 de degré strictement inférieur à n (ces deux espaces vectoriels sont isomorphes). La représentation polynomiale est bien adaptée à la description des codes cycliques comme le montre la propriété suivante.

Propriété 2 *Pour tout code $\mathcal{C}$ cyclique binaire, il existe un unique polynôme $g(x)$ tel que tout élément du code cyclique est le produit de g et d'un autre polynôme. Ce polynôme est appelé* ***polynôme générateur*** *et est de degré $n-k$. De plus, il divise x^n-1.*

Le code dual du code cyclique $\mathcal{C}$ est le code cyclique de polynôme générateur h tel que $g(x)h(x) = x^n - 1$.

Pour trouver les codes cycliques à longueur donnée, il faut donc connaître les factorisations possibles de $x^n - 1$. Pour les trouver toutes, il faut décomposer ce polynôme en produit de polynômes irréductibles dont les coefficients ne sont pas nécessairement binaires. Cette décomposition dépend de la parité de n, nous supposerons désormais n impair.

Propriété 3 *Soit m l'ordre de 2 modulo n (i.e. le plus petit entier i non nul tel que $2^i = 1[n]$). Le polynôme $x^n - 1$ admet n racines distinctes, éléments de F_{2^m}, appelé corps de décomposition de $x^n - 1$. Ces n racines sont appelées racines n^e de l'unité.*

Une des racines n^{es} de l'unité primitive, c'est-à-dire dont les puissances engendrent toutes les racines n^{es}, est notée α. Remarquons que tout polynôme g sur F_2 vérifie la propriété $g(x^2) = g(x)^2$. Ainsi si α^i est racine de g, $\alpha^{2i}, \alpha^{4i}, \cdots$ le seront aussi. Introduisons alors la classe cyclotomique de i :

Définition 5.1.2 *(***Classe cyclotomique***)* *La* ***classe cyclotomique*** *de i notée CT_i est définie par*

$$CT_i = \{i[n], 2i[n], 4i[n], \cdots, 2^j i[n], \cdots\}$$

Si α^i est racine de g alors $\prod_{j \in CT_i} x - \alpha^j$ divise g. De plus, $\prod_{j \in CT_i} x - \alpha^j$ est **l**e polynôme minimal de α^i, c'est-à-dire le polynôme à coefficients dans F_2, admettant α^i pour racine, de degré le plus faible possible. Ce polynôme minimal est noté $M^{(i)}(x)$.

Soient b et d deux entiers naturels non nuls. Nous pouvons maintenant définir les codes BCH.

Définition 5.1.3 *(***Code BCH binaire***)* *Le* ***code BCH binaire*** *de longueur n et de distance BCH d est le code cyclique dont le polynôme générateur admet entre autres pour racines $\alpha^b, \alpha^{b+1}, \cdots, \alpha^{b+d-2}$, i.e. $d-1$ racines qui sont des puissances consécutives de la racine n^e primitive α, et de plus grande dimension possible.*

Les polynômes minimaux permettent de trouver le polynôme générateur du code BCH.

Théorème 5.1.1 *Le* ***polynôme générateur*** *g* ***du code BCH*** *de longueur n et de distance BCH d est le plus petit polynôme commun multiple des polynômes $(M^{(i)}(x))_{b \leq i \leq b+d-2}$.*

Lorsque $b = 1$, le code est dit BCH au sens strict. Ce théorème montre que le code BCH de longueur n et de distance BCH $2d$ est identique au code BCH de même longueur et de distance BCH $2d+1$. Par conséquent, on considère des distances BCH impaires, sans perte de généralité. Enfin, justifions l'appellation de d par la proposition suivante.

Propriété 4 *La distance minimale d'un code BCH est supérieure ou égale à la distance BCH d, i.e. le nombre de zéros consécutifs du polynôme générateur.*

Le code dual d'un code BCH binaire est un code cyclique pas forcément BCH (son polynôme générateur n'a pas forcément des racines qui sont des puissances consécutives d'un élément primitif).

Propriétés spectrales

Il est possible de définir pour tout élément de F_2^n (ou de $F_2[x]/(x^n-1)$) une transformée de Fourier lorsque n est impair. Soit $\underline{c} = (c_1, \cdots, c_n)$ un élément de F_2^n. Notons $c(x)$ le polynôme canoniquement associé à cet élément. La transformée de Fourier $\underline{C} = (C_1, \cdots, C_n)$ de cet élément est défini par

$$0 \leq i \leq n \quad C_i = \sum_{j=0}^{n-1} c_j \alpha^{ij} = c(\alpha^i) \tag{5.1}$$

où α est une racine n^{e} de l'unité. Notons que les coefficients C_i obtenus par transformée de Fourier appartiennent *a priori* au corps F_{2^m}. La transformée de Fourier inverse est défini par

$$0 \leq i \leq n \quad c_i = \sum_{j=0}^{n-1} C_j \alpha^{-ij} = C(\alpha^{-i}). \tag{5.2}$$

Les classes cyclotomiques lient les composantes de $\underline{C}$ entre elles :

$$C_{i2^j} = c(\alpha^{i2^j}) = (c(\alpha^i))^{2^j} = C_i^{2^j},$$

définissant ainsi les contraintes de conjugaison. Par analogie avec la transformée de Fourier défini sur l'ensemble des distributions, $\underline{C}$ est appelé **spectre** de $\underline{c}$.

Par définition, la transformée de Fourier des mots d'un code BCH est donc nulle pour les indices $b, b+1, \cdots, b+d-1$. Les contraintes de conjugaison permettent de trouver quels autres coefficients de la transformée de Fourier seront nuls. Les codes BCH peuvent donc être définis directement à partir de leurs propriétés spectrales.

Décodage des codes BCH

Un algorithme classique pour décoder les codes BCH lorsque le poids de l'erreur est inférieur ou égal à l'entier t, où t est défini par $d = 2t+1$, est celui développé par Peterson-Gorenstein-Zierler [50]. Il ne nécessite pas de calculer le tableau standard de décodage, ce qui le rend plus efficace que ce dernier. Un autre algorithme, dû à Berlekamp et Massey [5], permet d'accomplir la même tâche en s'appuyant sur les propriétés spectrales des codes BCH.

Ces deux algorithmes sont cependant incomplets mais il existe des algorithmes complets (et différents du décodage par tableau standard) reposant sur des propriétés

spécifiques aux corps finis pour les cas $t = 2$ [5], pour une partie des codes BCH corrigeant trois erreurs [63]. Enfin, plusieurs algorithmes ont été proposés afin de décoder des erreurs de poids supérieur à t [26, 49].

Nous allons nous intéresser ici à l'algorithme de Peterson-Gorenstein-Zierler : supposons qu'un mot de code BCH a été émis sur un CBS : le bruit ajouté est donc additif. Soit $\underline{c}$ le mot de code émis, $\underline{e}$ le bruit ajouté par le canal et de poids w inférieur ou égal à t et $\underline{y}$ le mot reçu par le décodeur. L'algorithme de Peterson-Gorenstein-Zierler appliqué aux codes BCH binaires procède en quatre étapes que nous allons détailler :

- calcul des syndromes ;
- recherche du poids de l'erreur ;
- détermination des positions de l'erreur (aussi appelée localisation de l'erreur) ;
- la quatrième étape consistant en la détermination de la valeur de l'erreur aux positions déterminées par l'étape précédente est triviale sur le corps F_2 : l'erreur vaut 1.

Une fois l'erreur estimée, il ne reste plus qu'à la retrancher au mot reçu pour obtenir un mot de code.

Première étape : calcul des syndromes. Calculons la transformée de Fourier $\underline{Y}$ du mot reçu. Par linéarité de la transformée de Fourier, $\underline{Y}$ est la somme des transformées de Fourier du mot émis et du bruit, notées respectivement $\underline{C}$ et $\underline{E}$: $\underline{Y} = \underline{C} + \underline{E}$. D'après les propriétés spectrales des codes BCH, les coefficients $Y_b, Y_{b+1}, \cdots, Y_{b+d-2}$ dépendent uniquement de l'erreur $\underline{e}$. Ces coefficients constituent les syndromes $S_j = Y_{b+j-1}$, $1 \leq j \leq d-1$ de l'erreur. Notons i_j, $\quad 1 \leq j \leq w$, les positions inconnues des w erreurs et posons $X_j = \alpha^{i_j} \in F_2^m$. Notons que le syndrome S_i est la somme des puissances $(b+i-1)^{\text{e}}$ des éléments X_j. Si tous les syndromes sont nuls, l'algorithme prend fin. Les étapes suivantes de l'algorithme permettent de résoudre l'équation clé du décodage des codes BCH :

$$(1 + S(x))\Lambda(x) = \Omega(x)[x^{2t+1}] \tag{5.3}$$

où S est le polynôme formé par les syndromes ($S(x) = \sum_{i=1}^{2t} S_i x^i$), Λ est le polynôme localisateur d'erreurs défini par

$$\Lambda(x) = \prod_{j=1}^{w}(1 - X_j x) = 1 + \sum_{j=1}^{w} \Lambda_j x^j, \tag{5.4}$$

et Ω le polynôme évaluateur d'erreurs dont le degré est w. L'algorithme doit trouver Λ de degré le plus faible possible et Ω à partir de S vérifiant l'équation (5.3).

Deuxième étape : recherche du poids de l'erreur. La recherche du poids de l'erreur nécessite de calculer le déterminant d'une matrice M_j définie par

$$1 \leq j \leq t \quad M_j = \begin{pmatrix} S_b & \cdots & S_{b+j-1} \\ \vdots & & \vdots \\ S_{b+j-1} & \cdots & S_{b+2j-2} \end{pmatrix}. \tag{5.5}$$

Le déterminant de cette matrice est non nul lorsque $j \leq w$ et nul au-delà [43]. On initialise donc j en le posant égal à t puis on calcule le déterminant de M_j. Si ce déterminant est non nul alors l'erreur est de poids estimé $\nu = j$ et l'on passe à l'étape suivante sinon on décrémente j et on recommence.

Troisième étape : localisation de l'erreur. Les relations de Newton qui lient les coefficients de Λ aux sommes des puissances des éléments X_i, c'est-à-dire aux syndromes de l'erreur, permettent de déterminer le polynôme localisateur d'erreur :

$$M_\nu \begin{pmatrix} \Lambda_\nu \\ \vdots \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_{b+\nu} \\ \vdots \\ S_{b+2\nu-1} \end{pmatrix}. \tag{5.6}$$

Une fois le polynôme Λ déterminé, il faut chercher ses racines dans le corps F_{2^m} appelé, pour cette raison, corps localisateur d'erreur. L'inversion des racines trouvées donne alors les positions inconnues de l'erreur.

Quatrième étape : détermination de l'erreur. Sur le corps F_2, cette étape est *a priori* inutile puisque les symboles erronés valent obligatoirement 1 lorsque le mot reçu est à une distance inférieure ou égale à t d'un mot de code. Cette étape peut cependant servir à vérifier les calculs (surtout quand ils sont fait manuellement) ou à détecter les dépassements de capacité de correction en résolvant le système :

$$\begin{pmatrix} X_1^b & \cdots & X_\nu^b \\ \vdots & & \vdots \\ X_1^{b+\nu-1} & \cdots & X_\nu^{b+\nu-1} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ \vdots \\ e_{i_\nu} \end{pmatrix} = - \begin{pmatrix} S_b \\ \cdots \\ S_{b+\nu-1} \end{pmatrix}. \tag{5.7}$$

Srinivasan et Sarwate [59], poursuivant les travaux de Sarwate et Morrison [53], ont montré que lorsque le mot reçu est à une distance supérieure à t d'un mot de code, le décodeur pouvait ne pas détecter le dépassement de capacité de correction, ce dysfonctionnement se produit dans les conditions précisées par un théorème, dû à Srinivasan et Sarwate, rappelé dans l'annexe F. Il est possible de pallier à un tel dysfonctionnement en modifiant légèrement l'algorithme de Peterson-Gorenstein-Zierler.

Ayant brièvement rappelé la définition et les propriétés essentielles des codes BCH binaires, ainsi qu'une méthode de décodage, nous sommes prêts pour définir les codes BCH réels et montrer la forte analogie qui existe entre ces deux types de codes.

5.1.3 Codes BCH réels

Pour mesurer la distance entre deux vecteurs, nous conservons comme critère la distance de Hamming qui est toujours définie comme le nombre de composantes non nulles de la différence des deux vecteurs. La notion de distance de Hamming minimale

d'un code garde tout son sens lorsque le code est réel (ou complexe). Cette distance est adaptée à un bruit additif impulsif (alors que, face à un bruit blanc gaussien additif, on préferera étudier la distance euclidienne).

Construction des codes BCH réels

Commençons par définir les codes réels cycliques.

Définition 5.1.4 *(**code réel cyclique**) Un **code réel cyclique** de longueur n et de dimension k est un sous-espace vectoriel de $\mathbb{R}^n$ de dimension k et invariant par décalage cyclique. C'est donc un idéal de l'anneau quotient $\mathbb{R}[x]/(x^n-1)$.*

L'anneau $\mathbb{R}[x]/(x^n-1)$ est un anneau principal : tous ses idéaux sont engendrés par un seul polynôme appelé polynôme générateur et noté $g(x)$. Ce polynôme donc divise x^n-1. La factorisation de x^n-1 se fait simplement sur le corps des complexes $\mathbb{C}$

$$x^n - 1 = \prod_{j=0}^{n-1} (x - e^{\frac{2\sqrt{-1}\pi j}{n}}) \tag{5.8}$$

Le corps des complexes joue donc le rôle de corps de décomposition de x^n-1 donc également celui de corps localisateur d'erreurs. Une racine n^{e} de l'unité primitive α est, par exemple, $\alpha = e^{\frac{2\sqrt{-1}\pi}{n}}$ [1]. La contrainte de conjugaison dans le cas réel affirme que si un polynôme à coefficients réels admet une racine complexe alors le conjugué de cette racine est aussi une racine. La décomposition de x^n-1 en polynômes à coefficients réels dépend donc de la parité de n car cette parité détermine le nombre de solutions réelles

$$\begin{aligned} x^{2n} - 1 &= (x-1)(x+1) \prod_{j=1}^{n-1} (x^2 - 2x\cos\left(\frac{\pi j}{n}\right) + 1) \\ x^{2n+1} - 1 &= (x-1) \prod_{j=1}^{n} (x^2 - 2x\cos\left(\frac{2\pi j}{2n+1}\right) + 1). \end{aligned}$$

Nous pouvons à présent définir les codes BCH réels.

Définition 5.1.5 *(**Code BCH réel.**) Un **code BCH réel** de longueur n et de distance BCH d est un code cyclique réel dont le polynôme générateur g admet entre autres pour racines $\alpha^b, \alpha^{b+1}, \cdots, \alpha^{b+d-2}$ (où α est une racine n^{e}de l'unité primitive). Les ratios $\frac{b}{n}, \cdots, \frac{b+d-2}{n}$ sont appelées **fréquences de parité**.*

La contrainte de conjugaison, nécessairement vérifiée par le polynôme générateur (sinon le code obtenu est un code BCH complexe), implique que $\alpha^{-b}, \alpha^{-b-1}, \cdots, \alpha^{-b-d+2}$ sont aussi racines de ce polynôme. Le degré du polynôme générateur g est compris entre $d-1$ et $2d-2$. Rappelons que dans le cas binaire, la distance minimale d'un code BCH est supérieure à la distance d (même si, dans certains cas, davantage de renseignements sur cette distance minimale sont disponibles [43]). Dans le cas réel, nous pouvons

[1] toute puissance j de α, où j est premier est avec n, est aussi une racine n^{e} de l'unité primitive

judicieusement choisir les racines du polynôme générateur pour obtenir des **codes séparables à distance minimale maximale** (MDS, *Maximum Distance Separable*), i.e. égale à $n-k+1$, k étant la dimension du code :

- lorsque n et d sont impairs, les racines de g choisies sont les éléments de l'ensemble $\{\alpha^{\frac{n-d}{2}+1}, \cdots, \alpha^{\frac{n+d}{2}-1}\}$. Le spectre des mots de ce code sera donc nul pour les hautes fréquences ;
- lorsque n est impair et d pair, les éléments $\alpha^{-\frac{d}{2}+1}, \alpha^{-\frac{d}{2}+2}, \cdots, 1, \cdots, \alpha^{\frac{d}{2}-1}$ constituent les racines du polynôme g. Le spectre des mots de ce code sera donc nul pour les basses fréquences ;
- lorsque n et d sont pairs, deux ensembles de racines de g sont possibles : $\{\alpha^{-\frac{d}{2}+1}, \cdots, 1, \cdots, \alpha^{\frac{d}{2}-1}\}$ ou $\{\alpha^{\frac{n-d}{2}+1}, \cdots, \alpha^{\frac{n+d}{2}-1}\}$.

Ce choix est illustré sur les figures 5.1 et 5.2 suivant la parité de la longueur n. Dans ces trois cas, la distance minimale est égale à la distance BCH. Le choix des racines en vue d'obtenir un code MDS se réduit donc à choisir convenablement b : la facilité à construire des codes réels MDS est bien connue, un code réel tiré au hasard ayant de fortes chances d'être MDS [43]. L'intérêt des codes BCH réels réside donc dans leur structure qui permet un codage rapide.

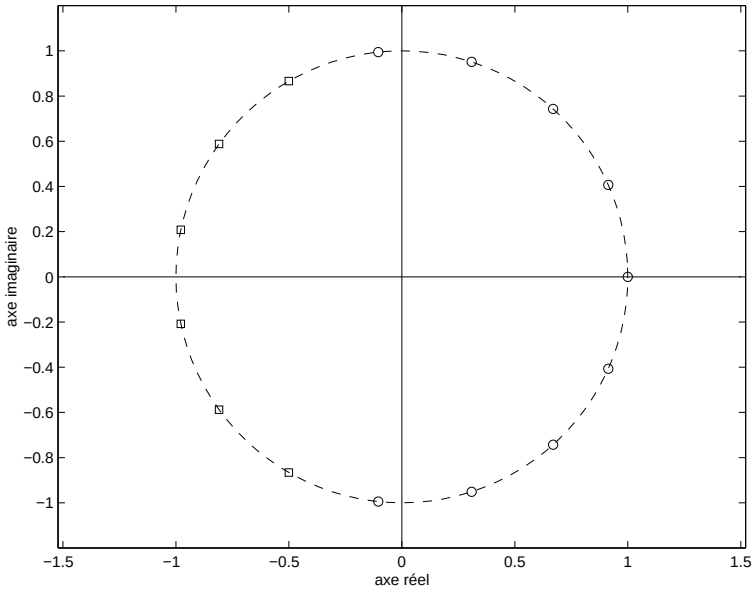


Fig. 5.1 – Choix des racines d'un code BCH réel MDS de longueur 15 : les carrés correspondent aux racines d'un code de dimension impaire (égale à 9), les cercles à celles du code dual, de dimension paire (égale à 6).

Il existe bien entendu des codes BCH réels de longueur n paire et de distance BCH d impaire : leur distance minimale n'est pas égale à $n-k+1$. Si notre but est de construire le code BCH de plus grande dimension possible avec une distance BCH supérieure à d impaire, nous construirons en fait un code BCH de longueur n et de distance BCH $d+1$ paire. Nous nous restreindrons, par la suite, à ces codes BCH réels MDS [2].

[2] Marshall [36] a montré l'existence de codes réels MDS pour n pair et d impair : ces codes sont analogues aux codes binaires négacycliques [5].

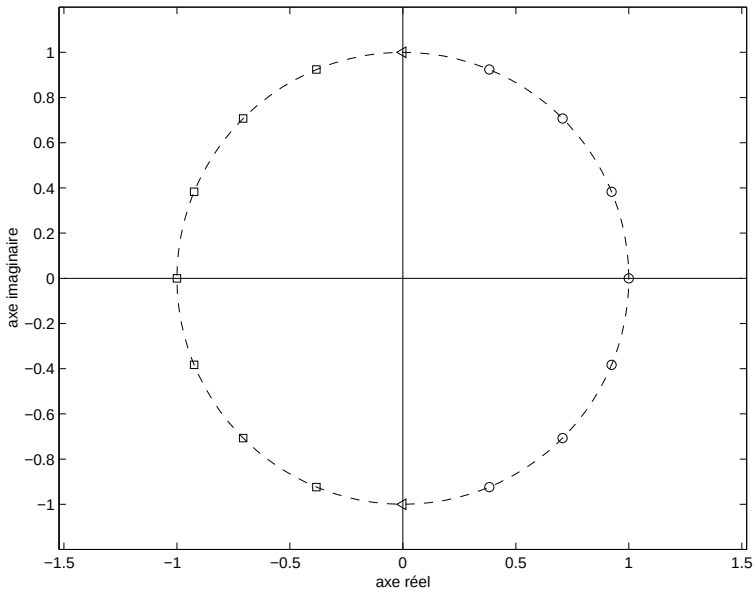


Fig. 5.2 – Choix des racines d'un code BCH réel MDS de longueur 16 : les carrés correspondent à celles d'un code de dimension impaire « passe-bas », les cercles à celles d'un code de dimension paire « passe-haut », les triangles aux racines n^{es} non utilisées.

Une autre différence avec le cas binaire est que le code dual d'un code BCH réel est un code BCH. Par conséquent, le dual d'un code BCH réel MDS est un code BCH réel MDS. Enfin remarquons que le rayon de recouvrement d'un code BCH réel (i.e. la distance de Hamming maximale entre un mot quelconque et le mot du code BCH réel le plus proche) est inférieur ou égal à $n-k$ (à l'instar des codes construits sur les corps finis).

Matrice génératrice et matrice de parité des codes BCH réels

Puisque les codes BCH réels sont des sous-espaces vectoriels, leur représentation par des matrices est naturelle. Ces codes dérivant de la transformée de Fourier discrète, rappelons l'expression de la matrice TF_n correspondant à cette transformée sur un vecteur de longueur n :

$$TF_n = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \alpha & \cdots & \alpha^{n-2} & \alpha^{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \alpha^{n-1} & \cdots & \alpha^{(n-1)(n-2)} & \alpha^{(n-1)^2} \end{pmatrix} \tag{5.9}$$

où $\alpha = e^{-\frac{2\sqrt{-1}\pi}{n}}$. La matrice TF_n admet bien sûr une inverse. La matrice de parité d'un code BCH de longueur n et de dimension k consiste donc à prendre $n-k$ lignes de la matrice TF_n de sorte qu'une quelconque de ces lignes soit la conjuguée d'une autre (ou d'elle-même). La deuxième colonne de ces lignes permet de déterminer aisément quels

coefficients du spectre des mots du code ainsi généré seront nuls : les matrices ainsi formées mettent en évidence les propriétés spectrales du code. La matrice génératrice est, quant à elle, formée des k lignes de TF_n restantes. Puisque les vecteurs lignes de TF_n forment une base orthogonale de $\mathbb{C}^n$, ces deux matrices vérifient bien $GH^t = 0$ et sont de rang plein.

Les deux matrices ainsi formées sont complexes or le code ainsi défini est réel. En additionnant ou soustrayant les lignes conjuguées deux à deux, nous obtenons des matrices réelles correspondant à un code équivalent. Ces matrices peuvent être calculées directement à partir du polynôme générateur du code en utilisant la propriété cyclique du code. La matrice génératrice systématique peut être aussi obtenue en appliquant un codage systématique, c'est-à-dire en calculant les restes de la division euclidienne des polynôme x^{n-k+i}, i étant compris entre 0 et $k-1$, par g : ces k restes forment la sous-matrice Par de la matrice génératrice. La matrice de parité systématique est alors déduite facilement de la matrice génératrice.

Puisque les codes BCH réels sont issus de la transformée de Fourier discrète, il est possible d'interpréter le mot de code comme une interpolation du mot d'information lorsque les racines du polynôme générateur correspondent à des fréquences de parité élevées.

Décodage des codes BCH réels

L'algorithme de Peterson-Gorenstein-Zierler, présenté ci-dessus dans le cas des codes BCH binaires, s'adapte aux codes BCH réels MDS. Nous montrons dans l'annexe F que l'algorithme de Peterson-Gorenstein-Zierler adapté directement au cas réel n'est pas sujet aux dysfonctionnements existant dans le cas binaire (du moins pas aux mêmes s'il y en a). Nous présentons aussi les modifications apportées pour circonvenir le problème de dépassement de capacité de correction.

L'algorithme comporte toujours quatre étapes :
- les syndromes sont calculés par un algorithme de transformée de Fourier rapide. Ces syndromes sont conjugués deux à deux ;
- le rang de la matrice M_j définie par l'équation (5.5) est égale au poids de l'erreur (lorsque le vrai poids de l'erreur, noté w, est inférieur à $\lfloor \frac{d-1}{2} \rfloor$). Le poids estimé est toujours noté ν. Au lieu de calculer plusieurs déterminants successifs, on calcule la décomposition en valeurs singulières de la matrice M_t. Le nombre d'erreurs estimé est égal au nombre de valeurs singulières non nulles ;
- l'équation (5.6) permettant de calculer le polynôme localisateur d'erreurs Λ est modifiée :

$$\begin{pmatrix} S_b & \cdots & S_{b+\nu-1} \\ \vdots & & \vdots \\ S_{b+2t-\nu-1} & \cdots & S_{b+2t-2} \end{pmatrix} \begin{pmatrix} \Lambda_\nu \\ \vdots \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_{b+\nu} \\ \vdots \\ S_{b+2t-1} \end{pmatrix}. \qquad (5.10)$$

Le système est surdéterminé : il prend ici en compte toutes les équations à notre disposition alors que l'algorithme de Peterson-Gorenstein-Zierler n'en considérait que ν. Ce système peut être résolu selon la méthode des moindres carrés. Pour détecter le dépassement de capacité de correction, nous réestimons, grâce au polynôme Λ calculé, les syndromes $S_{b+\nu-1}, \cdots, S_{b+d-2}$ afin de détecter un éventuel dépassement de capacité de correction. La recherche des racines de ce polynôme peut s'effectuer en calculant sa transformée de Fourier discrète inverse ;

– l'évaluation des erreurs $e_{i_1}, \cdots, e_{i_w}$ nécessite de résoudre le système

$$\begin{pmatrix} X_1^b & \cdots & X_w^b \\ \vdots & & \vdots \\ X_1^{b+d-2} & \cdots & X_w^{b+d-2} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ \vdots \\ e_{i_w} \end{pmatrix} = \begin{pmatrix} S_b \\ \cdots \\ S_{b+d-2} \end{pmatrix}. \tag{5.11}$$

Nous avons pris en compte dans l'évaluation des erreurs toutes les données à notre disposition, à savoir les $d-1$ syndromes. Le système (5.11), contrairement à celui présenté en (5.7) est surdéterminé. En effet, toutes les opérations se déroulant non plus dans un corps fini mais dans le corps des complexes, des erreurs d'arrondis s'accumulent, depuis le calcul des syndromes jusqu'à celui des valeurs de l'erreur. La surdétermination du système ne privilégie donc pas plus certains syndromes que d'autres *a priori*. La résolution par les moindres carrés du système (5.11) impose à l'erreur évaluée d'être réelle : en effet, il est possible de réordonner les $d-1$ équations de ce système pour faire apparaître les équations et leurs conjuguées (si d est pair, l'équation où intervient une des deux racines n^{e}de l'unité réelles est dupliquée). Le nouveau système s'écrit alors

$$\begin{pmatrix} \mathcal{M} \\ \mathcal{M}^* \end{pmatrix} \begin{pmatrix} e_{i_1} \\ \vdots \\ e_{i_w} \end{pmatrix} = - \begin{pmatrix} \mathcal{S} \\ \mathcal{S}^* \end{pmatrix}$$

où $\mathcal{M}$ (resp. $\mathcal{S}$) est constituée des $\lfloor \frac{d-1}{2} \rfloor$ premières lignes de la matrice (resp. du vecteur formé des syndromes) du système (5.11) si d est impair et des $\frac{d}{2}$ premières si d est pair. La matrice composée de $\mathcal{M}$ et de sa conjuguée est de rang plein car elle est équivalente à une matrice de Van der Monde. La résolution par les moindres carrés donne la valeur de l'erreur

$$\begin{pmatrix} e_{i_1} \\ \vdots \\ e_{i_w} \end{pmatrix} = -(\mathcal{M}^\dagger \mathcal{M} + \mathcal{M}^t \mathcal{M}^*)^{-1} (\mathcal{M}^\dagger \mathcal{S} + \mathcal{M}^t \mathcal{S}^*). \tag{5.12}$$

L'erreur calculée est donc bien réelle. Nous vérifions que les syndromes, notés $\hat{S}_i$, correspondant à l'erreur estimée se confondent avec les syndromes initialement calculés. Si c'est le cas, l'algorithme a bien fonctionné sinon un dépassement de capacité de correction est détecté.

Les modifications apportées par la transposition de l'algorithme de Peterson-Gorenstein-Zierler au cas réel se résument par la prise en compte de toutes les équations à résoudre

et la résolution de ces systèmes surdéterminés par la méthode des moindres carrés. La détection de la capacité de correction se situe à deux niveaux : la réussite ou l'échec de la recherche des ν racines du polynôme localisateur et la réestimation des syndromes à partir de l'erreur estimée.

Contrairement au cas des codes sur les corps finis, l'implémentation de cet algorithme doit prendre en compte le bruit de calcul généré par les diverses étapes. Trois seuils positifs sont donc introduits :

- le rang des matrices M_j servant à déterminer le nombre d'erreurs ν est estimé à partir de la décomposition en valeurs singulières de cette matrice. Les valeurs singulières inférieures à un premier seuil θ_1 sont considérées comme nulles ;
- le polynôme localisateur Λ est évalué sur les n racines n^{es} de l'unité : ces n valeurs sont alors classées par module croissant. S'il n'y a pas de dépassement de capacité de correction, les ν premières valeurs doivent être nulles. En raison du bruit de calcul, nous comparons le module de la ν^{e} à un seuil θ_2. Si la valeur est inférieure au seuil, nous avons identifé ν racines sinon nous détectons un dépassement de capacité de correction ;
- enfin, après réestimation de syndromes, nous calculons l'erreur quadratique relative

$$\frac{\sqrt{\sum_{i=1}^{d-1} |\hat{S}_i - S_i|^2}}{\sqrt{\sum_{i=1}^{d-1} |S_i|^2}}$$

que nous comparons à un seuil θ_3. Si l'erreur quadratique relative est supérieure à ce seuil, un dépassement de capacité de correction est détecté.

On notera que les deux premiers seuils dépendent de la variance du signal en entrée du décodeur.

En ce qui concerne l'algorithme de Berlekamp-Massey, il est directement donné dans [5] pour n'importe quel corps : il n'y a donc pas besoin en théorie de le modifier. Il convient cependant de tester les coefficients des polynômes calculés par cet algorithme afin de les mettre à zéro si leur module est trop petit.

Le décodage des codes BCH réels du point de vue du traitement du signal

Les résolutions de systèmes aux moindres carrés et la décomposition en valeurs singulières introduites dans l'algorithme de Peterson-Gorenstein-Zierler sont des outils classiques de traitement du signal. Une plus forte analogie existe entre le décodage des codes BCH réels et le problème de l'estimation de sinusoïdes. Ce dernier consiste à identifier les w sinusoïdes ou exponentielles complexes composant le signal observé bruité : les inconnues sont donc w, les fréquences, les phases et les amplitudes. Dans le cas des codes BCH réels, le signal observé est composé des $d-1$ syndromes et les phases sont toutes nulles. L'estimation du nombre d'erreurs détermine le nombre de

sinusoïdes à identifier. La localisation de l'erreur identifie leur fréquence et l'évaluation de l'erreur détermine leur amplitude. Plusieurs méthodes en traitement de signal [47] résolvent plus ou moins bien ce problème :

- la méthode de Pysarenko suppose w connu et identifie les fréquences à partir de la matrice d'autocorrélation d'ordre $2w$ du signal : le polynôme Λ appartient au noyau de cette matrice. L'estimation des amplitudes nécessite la résolution d'un système de Van der Monde ;
- la méthode de Prony, initialement décrite pour approximer une fonction par une somme d'exponentielles, estime les fréquences de la même manière que la méthode de Pysarenko. Puis elle résout aux moindres carrés le système comportant autant d'équations que d'échantillons observés liant les valeurs des échantillons aux amplitudes, utilisant l'existence d'un prédicteur d'ordre 2 pour chacune des exponentielles. Le lien entre cette méthode et le décodage des codes BCH a été souligné par Wolf [66] ;
- l'algorithme MUSIC (*MUltiple SIgnal Characterization*) travaille à un ordre supérieur à $2w$. L'espace est décomposé en deux sous-espaces orthogonaux : le sous-espace déterminé par les w sinusoïdes à identifier et le sous-espace engendré par le bruit. La dimension du sous-espace associé au signal utile est le double de w. La matrice d'autocorrélation permet de déterminer une base du sous-espace bruit. Une sinusoïde sera donc orthogonale à cette base, les fréquences composant le signal sont donc identifiées en identifiant les maxima de l'inverse de la norme du vecteur composé des produits scalaires de la sinusoïde avec chacun des vecteurs de cette base.
- l'algorithme Esprit consiste à calculer deux matrices d'autocorrélation d'ordre $2w$ et $2w+1$ puis à exploiter la structure de ces deux matrices. La matrice d'autocorrélation d'ordre $2w+1$ contient la matrice d'ordre $2w$ et une autre matrice de taille $2w \times 2w$ qui est est le produit de la première par une matrice composée de matrices de rotation de taille 2×2. Cette dernière permet alors d'estimer les fréquences des sinusoïdes en trouvant l'angle de ces rotations ;
- la méthode de Capon consiste à chercher le filtre qui aura une puissance maximale à une fréquence fixée connaissant le signal observé : ce filtre est donc adapté au signal observé. Le spectre estimé est l'inverse de la puissance du signal résultant du filtrage d'une sinusoïde par le filtre blanchissant le signal observé. Les maxima de ce spectre correspondent aux fréquences cherchées. L'amplitude estimée est égale à la sortie du filtre adapté attaqué par le signal observé.

Le point commun à toutes ces méthodes est leur approche statistique du problème. Le bruit impulsif changeant d'un mot de code BCH à un autre, cette approche statistique ne peut être exploitée à moins que le nombre de syndromes ne soit considérable et que le nombre d'erreurs introduits reste modeste, ce qui n'est pas une situation typique dans le domaine du codage correcteur d'erreurs.

Vers un décodage complet

Par principe de dualité entre codage de source et codage de canal, nous pouvons utiliser les codes BCH réels pour comprimer une source réelle. À l'instar de la compression de la source binaire symétrique par des codes linéaires binaires, il faut disposer d'un algorithme complet pour espérer coder efficacement la source. Lorsque le mot de source est à une distance inférieure ou égale à la capacité de correction t, l'algorithme de Peterson-Gorenstein-Zierler ou celui de Berlekamp-Massey permet de trouver le mot du code le plus proche. La probabilité de cet événement dépend bien entendu de la source à comprimer.

Lorsque le mot de source est à une distance strictement supérieure à la capacité de correction, Wolf a proposé d'appliquer un algorithme de vote [67], expliqué originellement pour les codes complexes MDS construits à partir de la matrice TF_n. Cet algorithme se transpose directement aux codes BCH réels MDS : puisque le code (complexe ou BCH réel) est MDS, n'importe quel k-uplet choisi parmi les n composantes peut être considéré comme un mot d'information. Wolf calcule le mot de code correspondant à chacun des C_n^k mots d'information construits à partir du mot reçu. Les C_n^k mots de code se répartiront en deux classes : un nuage de points assez dense autour du mot de code le plus proche et des mots de code assez éloignés (au sens de la distance euclidienne) de ce nuage. Wolf souligne au passage que, grâce à cet algorithme de vote, il est possible de corriger presque tout le temps $d-2$ erreurs, c'est-à-dire le double par rapport aux codes de même type construits sur les corps finis. L'inconvénient de cet algorithme réside bien sûr dans sa complexité !

Plusieurs autres stratégies, curieusement jamais mentionnées ni par Marshall ni par Wolf, et d'une simplicité parfois déconcertante, mènent à un décodage complet potentiellement sous-optimal :

- la première est de projeter le mot de source sur le code, c'est-à-dire calculer son spectre, mettre à zéro ce spectre aux fréquences de parité et utiliser la transformée inverse pour obtenir un mot de code. Ceci est équivalent à une projection du mot sur le sous-espace vectoriel défini par le code BCH réel dans un espace euclidien. Cette stratégie minimise donc l'erreur quadratique et donne en général un mot de code à une distance de Hamming de n du mot de source. Elle n'est donc pas adaptée à notre objectif (compression supprimant du bruit impulsif), malgré sa simplicité ;
- la seconde consiste à effectuer une division euclidienne du mot de source considéré comme un polynôme par le polynôme générateur. Du point de vue du traitement du signal, nous cherchons à filtrer le mot de source par un filtre dont les pôles sont sur le cercle unité. Le mot de code obtenu sera donc à une distance de Hamming inférieure ou égale à $d-1$ du mot de source et l'erreur quadratique résultante sera élevée, comparée à l'erreur quadratique minimale obtenue par projection. Cette méthode n'est pas plus adaptée que la précédente à notre objectif ;
- la troisième méthode possible est d'utiliser un algorithme de décodage algébrique

(par exemple celui de Berlekamp-Massey) gérant les effacements en plus des erreurs et l'appliquer itérativement en faisant croître le nombre d'effacements et varier la position de ceux-ci, jusqu'à ce qu'une solution convenable soit trouvée. Le mot de code obtenu sera le mot de code le plus proche du mot de source (au sens de la distance de Hamming). Cette méthode permet donc un décodage complet optimal des codes BCH réels MDS.

Justifions la complétude de cette dernière méthode : un code MDS de distance d peut corriger au plus $d-1 = n-k$ effacements, or le rayon de recouvrement ρ du code est majoré par $n-k$ en raison de la borne de la redondance. La dernière méthode appliquée à un mot de source à distance ρ du mot de code le plus proche produira donc bien le mot de code voulu. L'optimalité de cette méthode est assurée par la prise en compte progressive d'un nombre de plus en plus élevé d'effacements (de 0 à $n-k$). L'inconvénient de cette méthode consiste en sa gloutonnerie : elle nécessite de recourir dans le pire des cas à $1+\sum_{l=0}^{n-k-1} C_n^l$ algorithmes de correction d'erreurs et d'effacements. En effet, si un mot n'a pu être décodé avec succès en supposant que 0, puis 1,..., puis $n-k-1$ effacements se sont produits, il faut donc chercher à décoder le mot en supposant $n-k$ effacements. Le code étant supposé MDS, $n-k$ colonnes quelconques de la matrice de parité prise comme sous-matrice de la matrice de TF_n forment une base de $\mathbb{C}^{n-k}$. Par conséquent, quelles que soient les $n-k$ colonnes considérées, il est possible de trouver l'erreur correspondant aux syndromes (équation (5.11)). Le mot d'erreur ainsi calculé sera bien réel car la moitié des $n-k$ équations est le conjugué de l'autre moitié. La complexité de l'algorithme peut être réduite en prenant en compte la spécificité des effacements : un code de distance d peut corriger e erreurs et l effacements si $2e+l$ est strictement inférieur à d. Par conséquent le nombre d'effacements est initialisé à zéro et augmente de 2 en 2. Si la distance d est paire et que l'algorithme de décodage algorithme a échoué avec $d-2$ effacements, il est nécessaire de refaire cet algorithme en supposant la présence de $d-1$ effacements (afin de garder la propriété de complétude de cette méthode de décodage).

Cette méthode est à rapprocher de l'algorithme GMD (*Generalized Minimum Distance*) proposé par Forney [22] dans le cadre d'un décodage de canal à décisions souples. La principale différence provient du fait que, connaissant les probabilités de transition du canal, il est possible d'attribuer des valeurs de confiance à chacune des composantes du mot. Ces valeurs de confiance traduisent le niveau de bruit *a priori* perturbant la composante reçue. Ces valeurs, une fois classées par ordre croissant, définissent une position estimée des effacements et évitent de devoir effectuer une recherche exhaustive de celles-ci.

Des trois stratégies, la première et la dernière paraissent les plus intéressantes : la première a l'avantage de la simplicité au détriment de l'efficacité en terme de distance de Hamming alors que la troisième est optimale de ce point de vue mais très coûteuse en temps.

5.1.4 Application des codes BCH réels MDS au codage de source

Le choix de l'une ou l'autre des stratégies conduisant à un décodage complet dépend bien sûr des statistiques de la source, de la complexité autorisée et la distorsion visée. La troisième méthode est la meilleure vis-à-vis de la distorsion de Hamming et les statistiques décrivant la source joueront sur la complexité moyenne de cette méthode. Cependant, indépendamment de la source, la complexité du pire des cas est déterminée par la valeur du rayon de recouvrement ρ : il est donc souhaitable de trouver sa valeur afin de savoir si le codage de source par un code BCH réel MDS est réalisable.

Théorème 5.1.2 [3] *Le rayon de recouvrement ρ d'un code BCH réel MDS de longueur n et de dimension k est $n-k$.*

Preuve

La borne de la redondance, comme nous l'avons vu, montre que le rayon de recouvrement, noté ρ est inférieur à $n-k$. Il reste donc à montrer qu'il est supérieur ou égal à $n-k$. Pour cela, introduisons les codes de Reed-Solomon complexes : un code de Reed-Solomon complexe de longueur n et de dimension k est un code complexe (i.e. un sous-espace vectoriel de $\mathbb{C}^n$ de dimension k) cyclique. Par conséquent, les $n-k$ racines de son polynôme générateur sont $n-k$ racines n^{es} de l'unité distinctes. La matrice de parité d'un code de Reed-Solomon est donc une sous-matrice extraite de la matrice TF_n : les codes de Reed-Solomon sont donc MDS.
Notons $\mathcal{C}$ le code BCH réel MDS de longueur n et de dimension k et RS un code de Reed-Solomon de longueur n et de dimension $k+1$ (si $k=n$, le théorème est trivial) contenant $\mathcal{C}$: $n-k-1$ des $n-k$ racines du polynôme générateur de $\mathcal{C}$ forment les racines du polynôme générateur de RS. Minorons le rayon de recouvrement ρ de $\mathcal{C}$ grâce au code RS :

$$\begin{aligned}
\rho &= \max_{\underline{u}\in\mathbb{R}^n} \min_{\underline{v}\in\mathcal{C}} d_H(\underline{u},\underline{v}) \\
&\geq \max_{\underline{u}\in RS\cap\mathbb{R}^n} \min_{\underline{v}\in\mathcal{C}} d_H(\underline{u},\underline{v}) \\
&\geq \max_{\underline{u}\in RS\cap\mathbb{R}^n} \min_{\underline{v}\in RS\cap\mathbb{R}^n} d_H(\underline{u},\underline{v}) \\
&\geq \min_{\substack{(\underline{u},\underline{v})\in(RS\cap\mathbb{R}^n)^2 \\ \underline{u}\neq\underline{v}}} d_H(\underline{u},\underline{v}), \qquad (5.13)
\end{aligned}$$

les inégalités précédentes découlant de l'inclusion de $\mathcal{C}$ dans RS donc dans $RS\cap\mathbb{R}^n$, qui est un sous-espace vectoriel de $\mathbb{R}^n$. L'inégalité (5.13) montre que le rayon de recouvrement est supérieur ou égal à la distance de Hamming minimale de $RS\cap\mathbb{R}^n$. Puisque $RS\cap\mathbb{R}^n$ est un sous-ensemble de RS, code MDS, on en déduit :

$$\rho \geq \min_{\substack{(\underline{u},\underline{v})\in RS^2 \\ \underline{u}\neq\underline{v}}} d_H(\underline{u},\underline{v}) = n-k,$$

ce qu'il fallait démontrer. Ce résultat est à rapprocher du lemme du supercode [12].

[3] Ce résultat n'était qu'une conjecture mais P. Solé nous a mis sur la piste de la démonstration.

□

Il est amusant de remarquer que l'on a à la fois construit des codes à distance maximale et à rayon de recouvrement maximal. Dans le cas binaire, le rayon de recouvrement des meilleurs codes obtenus pour la compression de SBS était faible. On en déduit que l'utilisation des codes BCH réels MDS en codage de source est alors fortement dictée par les statistiques décrivant la source, ces statistiques déterminant à la fois les performances de ces codes et la complexité moyenne de la troisième méthode.

Corollaire 2 *Supposons que la source est constituée d'échantillons gaussiens indépendants et identiquement distribués. Le mot de source est à distance $n-k$ d'un code BCH réel MDS de longueur n et de dimension k avec une probabilité* 1.

Preuve

Nous allons démontrer ce corollaire en montrant que la probabilité qu'un mot de source soit à une distance strictement inférieure à $n-k$ du code BCH réel MDS est nulle. Soit H la matrice de parité du code BCH réel MDS. Le syndrome défini par $\underline{s} = \underline{u}H^t$ suit une loi gaussienne de dimension $n-k$ dont les composantes sont indépendantes et identiquement distribuées. Le mot de source $\underline{u}$ est à une distance $d_{\underline{u}}$ du code ($d_{\underline{u}} < n-k$) : il est alors possible de trouver $d_{\underline{u}}$ colonnes de H telles que le syndrome de $\underline{u}$ soit une combinaison linéaire réelle de ces $d_{\underline{u}}$ colonnes : le syndrome appartient alors à un sous-espace vectoriel de dimension $d_{\underline{u}}$ de $\mathbb{C}^{n-k}$. La probabilité que le syndrome appartienne à un tel sous-espace vectoriel est donc nulle. Puisque le rayon de recouvrement est égal à $n-k$, n'importe quelle combinaison de moins de $n-k$ colonnes de la matrice H engendre un sous-espace vectoriel au sens strict de $\mathbb{C}^{n-k}$. Par conséquent, la probabilité qu'un mot de source soit à une distance strictement inférieure à $n-k$ du code BCH réel MDS est nulle.

□

Ce corollaire peut se généraliser à d'autres sources (laplaciennes, gaussiennes généralisées, etc...) pourvu que la probabilité d'obtenir un syndrome appartenant à un sous-espace vectoriel de dimension strictement inférieure à $n-k$ soit nulle. La distorsion de Hamming sera par conséquent égale à $1-k/n$ pour de telles sources.

Remarquons également qu'en raison de la valeur du rayon de recouvrement, un mot à distance $\rho = n-k$ d'un mot de code est aussi à distance ρ de $C_n^{n-k}-1$ mots de code distincts du premier. Le rayon de recouvrement détermine la complexité de l'algorithme de codage de source : trouver le mot de code le plus proche (au sens de la distance de Hamming) du mot de source sera facile car ce mot de code est très certainement à une distance $n-k$ du mot de source et il n'est pas le seul à être à une telle distance du mot de source. Il suffit alors de résoudre un système d'équations pour trouver un des mots de codes les plus proches. Le rayon de recouvrement détermine également la distorsion de Hamming observée.

Compression d'une source gaussienne

Nous avons mis en œuvre la méthode de décodage complet en choisissant comme méthode de décodage algébrique l'algorithme de Berlekamp-Massey pour comprimer une source gaussienne vectorielle blanche de variance σ^2. Cette méthode de décodage complet trouve le mot de code le plus proche (au sens de la distance de Hamming) du mot de source. D'après le corollaire 2, la distorsion de Hamming par symbole sera égale à $1 - R$ où R désigne le rendement du code BCH ($R = k/n$). Pour des faibles longueurs n, il est possible de choisir le mot de code qui minimise l'erreur quadratique parmi les C_n^{n-k} mots de code les plus proches. Un tel décodeur de source cherche donc à enlever de la source un bruit impulsif d'énergie minimale.

Si la contrainte imposée par la recherche du mot de code le plus proche au sens de la distance de Hamming est supprimée, la minimisation de l'erreur quadratique moyenne (EQM) conduit, comme nous l'avons vu au paragraphe 5.1.3 à annuler $n - k$ composantes du spectre du mot de source. Or ce spectre sera une source blanche gaussienne de dimension n (par propriété de la transformée de Fourier discrète). Par conséquent, l'EQM par symbole est égale, dans ce cas, à $(1 - k/n)\sigma^2$.

Nous avons simulé pour deux longueurs distinctes n (7 et 15) ce codeur de source en tirant aléatoirement 100000 vecteurs de dimension n. La variance de la source est égale à 1. Les résultats sont reportés dans le tableau 5.1. Les trois premières colonnes donnent les paramètres du code. La quatrième colonne donne la valeur $1 - R$ qui représente à la fois la distorsion de Hamming théorique et l'erreur quadratique moyenne sans contrainte. La cinquième colonne donne la distorsion de Hamming obtenue par simulation et la dernière colonne l'EQM minimale résultant du choix du mot de code le plus proche au sens de la distance de Hamming. Les résultats sont également représentés sur la figure 5.3.

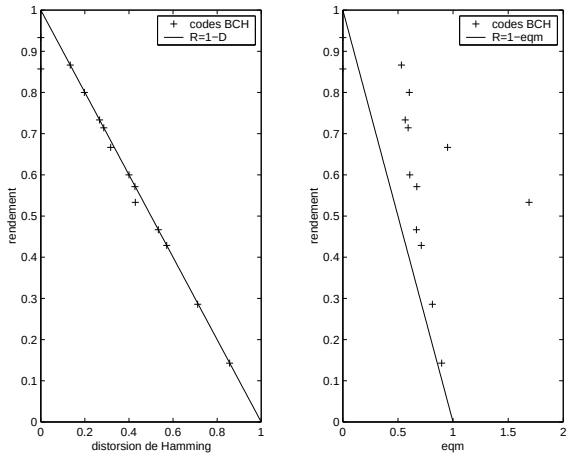


Fig. 5.3 – Performances (distorsion de Hamming et EQM) des codes BCH réels MDS pour la compression de source gaussienne i.i.d.

Longueur n	Dimension k	Rendement R	$1-R$	Distorsion de Hamming	EQM contrainte
7	1	1.428571e-01	8.571428e-01	8.567571e-01	8.966470e-01
7	2	2.857142e-01	7.142857e-01	7.117100e-01	8.129165e-01
7	3	4.285714e-01	5.714285e-01	5.712914e-01	7.116506e-01
7	4	5.714285e-01	4.285714e-01	4.270457e-01	6.710666e-01
7	5	7.142857e-01	2.857142e-01	2.857142e-01	5.924420e-01
15	8	5.333333e-01	4.666667e-01	4.289893e-01	1.689633e+00
15	9	6.000000e-01	4.000000e-01	3.999793e-01	6.075193e-01
15	10	6.666667e-01	3.333333e-01	3.171433e-01	9.507172e-01
15	11	7.333333e-01	2.666667e-01	2.666660e-01	5.662928e-01
15	12	8.000000e-01	2.000000e-01	1.977600e-01	6.028560e-01
15	13	8.666667e-01	1.333333e-01	1.333333e-01	5.313677e-01

Tab. 5.1 – Performances des codes BCH réels MDS pour la compression de source gaussienne i.i.d.

Nous remarquons que la distorsion de Hamming théorique prévue correspond avec la distorsion obtenue par simulation : le rayon de recouvrement des codes BCH réels MDS détermine bien la distorsion de Hamming générée par ces codes. L'EQM contrainte est largement supérieure à celle non contrainte, ce qui n'est guère surprenant car notre codeur de source cherche à supprimer un bruit impulsif. La compression de source gaussienne blanche par les codes BCH réels MDS se révèle donc peu performante. Nous allons donc essayer de comprimer une source courante, cette fois-ci corrélée, à savoir une image.

Compression d'images

Nous utilisons les codes BCH réels pour comprimer une image ligne par ligne. Nous utilisons l'algorithme de décodage complet des codes BCH réels avec minimisation de l'erreur quadratique en second critère pour départager les mots de codes à distance de Hamming égale du mot de source à coder. L'image de référence est Léna, quantifiée sur 256 niveaux de gris et de taille 256×256 pixels. Nous allons la comprimer par deux codes de longueur $n = 15$ et de dimension 7 ou 11. Pour cela nous supprimons la dernière ligne et la dernière colonne de l'image de référence : l'image obtenue est représentée sur la figure 5.4.

L'image comprimée est obtenue en choisissant un codage systématique. Les images reconstruites obtenues sont présentées sur les figures 5.5 et 5.6. La distorsion et l'EQM correspondante sont indiquées dans le tableau 5.2. Encore une fois, on observe que le mot de source se trouve très fréquemment à distance $n-k$ du mot de code le plus proche. La plus grande différence avec le cas de la source gaussienne réside dans la valeur de l'EQM contrainte qui reste petite. Les codes BCH réels semblent donc mieux

Fig. 5.4 – Image de référence Léna (255×255 pixels.)

Fig. 5.5 – Léna comprimée puis reconstruite par un code BCH réel, $n = 15$, $k = 11$.

adaptés à des sources corrélées qu'à des sources indépendantes. Les différences les plus notables à l'oeil nu consistent dans les effets de bloc : il est facile de voir à l'oeil nu quelle est la longueur des codes employés !

Nous avons essayé de combiner compression par ligne et compression par colonne mais sans succès car les erreurs s'accumulant, l'image reconstruite comporte plusieurs colonnes erronées par paquets de $n - k$.

Fig. 5.6 – Léna comprimée puis reconstruite par un code BCH réel, $n = 15$, $k = 7$.

Longueur n	Dimension k	Rendement R	$1 - R$	Distorsion de Hamming	EQM contrainte normalisée
15	7	4.666667e-01	5.333333e-01	5.333026e-01	8.2e-03
15	11	7.333333e-01	2.666667e-01	2.666667e-01	4.6e-03

Tab. 5.2 – Performances des codes BCH réels MDS en compression d'images.

5.2 Codage conjoint source gaussienne - canal binaire symétrique, selon le critère de l'EQM, par étiquetage linéaire

De nombreux articles [17, 18, 19, 29, 30] considérant ce modèle de communication -à savoir source réelle, canal binaire symétrique et critère de la distance euclidienne- font toujours abstraction du code de canal : celui est supprimé de la chaîne de transmission. Cette suppression permet, d'une part, de simplifier l'étude et, d'autre part de développer des quantificateurs ou des étiqueteurs conjoints. Un code de canal peut-il améliorer les performances de ces sytèmes ? Comment optimiser le quantificateur lorsque le code de canal est fixé ?

Nous ne choisissons pas comme quantificateur le codeur de source se fondant sur les codes BCH réels : l'une des difficultés liés à ce codage provient du fait que ce quantificateur possède un nombre infini de sorties. Si l'on souhaite transmettre le mot de source comprimé sur un canal binaire symétrique, il faudra encore le quantifier et, éventuellement, le coder pour le protéger du bruit tout en choisissant une bonne correspondance entre l'index représentant un mot du code de source et ce dernier. Autant d'étapes supplémentaires qui rendent la conception d'un système de codage

conjoint ardue. Un moyen de simplifier cette conception est d'imposer une relation linéaire entre les mots du code de source et les mots du code de canal et de choisir le critère objectif le plus étudié dans ce domaine, à savoir l'erreur quadratique moyenne (EQM), plus facile à traiter que la distance de Hamming.

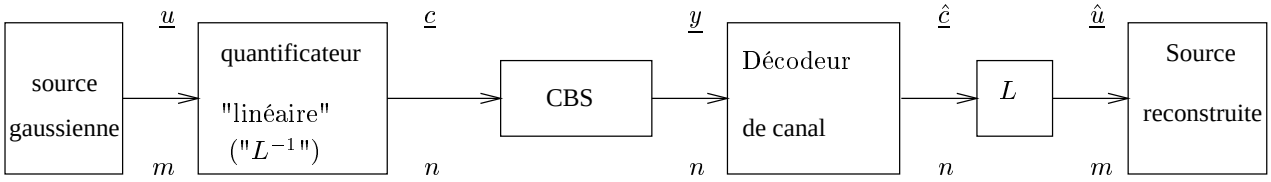


FIG. 5.7 – Quantification d'une source gaussienne par codage conjoint et étiquetage linéaire et transmission sur CBS

Un tel système est représenté sur la figure 5.7 : un code de canal binaire linéaire $\mathcal{C}_{canal}$ de longueur n et de dimension k est choisi ainsi qu'une matrice à coefficients réels L de taille $m \times n$. Le code de source $\mathcal{C}_{source}$ est alors défini par

$$\mathcal{C}_{source} = \{\underline{s} \in \mathbb{R}^m / \underline{s} = L(2\underline{c} - \underline{1}), \quad \underline{c} \in \mathcal{C}_{canal} \subset F_2^n\} \tag{5.14}$$

où $\underline{1}$ est le vecteur de dimension n composé uniquement de 1. Les mots $\underline{c}$ du code de canal sont modulés ($2\underline{c} - \underline{1}$ correspond à la sortie d'un modulateur d'une modulation de phase à deux états), ce qui permet de s'assurer que, lorsque le code de canal est universel, le code de source est à moyenne nulle, à l'instar de la source considérée. Notons $\underline{c}_m = 2\underline{c} - \underline{1}$.

Le mot de source émis $\underline{u}$ est donc quantifié par le mot de source $L\underline{c}_m$ et le mot $\underline{c}$ correspondant est émis sur le canal : le codeur est donc conjoint. Le canal binaire symétrique ajoute du bruit au mot $\underline{c}$: le décodeur reçoit le mot $\underline{y}$ et restitue le mot $\hat{\underline{c}}$, mot du code de canal le plus probable connaissant le mot reçu. Le mot de source estimé est donc $L\hat{\underline{c}}_m$. en raison de la linéarité entre code de source et code de canal. La matrice L est appelée matrice d'étiquetage.

Les performances optimales du codage conjoint sont données par le théorème de codage conjoint de Shannon [13]. Le rendement R et la distorsion D doivent vérifier

$$R \geq \frac{R(D)}{C_{CBS}(p)} \tag{5.15}$$

où p est la probabilité d'erreur du canal binaire symétrique. La source que nous étudierons dans cette section sera la source gaussienne blanche. Le critère de fidélité retenu sera l'erreur quadratique moyenne. Sa fonction rendement-distorsion est donnée par [41]

$$R(D) = \begin{cases} \frac{1}{2}\log_2 \frac{\sigma^2}{D} & \text{si} \quad 0 < D \leq \sigma^2 \\ 0 & \text{si} \quad D > \sigma^2 \end{cases} \quad \text{bits par symbole de source,} \tag{5.16}$$

où σ^2 est la variance de la source gaussienne par symbole.

Le rendement R du système de la figure 5.7 est donc

$$R = \frac{n}{m} \text{ bits par symbole de source.} \tag{5.17}$$

La distorsion totale par symbole de source, suivant le critère de l'erreur quadratique, s'écrit :

$$D = \frac{1}{m} E[||\underline{u} - L\underline{\hat{c}}_m||^2] = \frac{1}{m} \sum_{(\underline{c},\underline{\hat{c}}) \in \mathcal{C}_{canal}^2} \int_{\underline{u} \in V_s(\underline{c})} p(\underline{u}) \sum_{\underline{y} \in V_c(\underline{\hat{c}})} p(\underline{y}|\underline{c}) ||\underline{u} - L\underline{\hat{c}}_m||^2 d\underline{u} \tag{5.18}$$

avec

- $p(\underline{u})$ la densité de probabilité de l'extension m^{e} de la source ;
- $p(\underline{y}|\underline{c})$ la probabilité que la sortie du CBS soit $\underline{y}$ sachant que le mot d'entrée est $\underline{c}$;
- $V_s(\underline{c})$ est l'ensemble des mots de source pour lesquels la sortie du codeur est le mot $\underline{c}$ du code de canal ;
- $V_c(\underline{\hat{c}})$ est l'ensemble des mots reçus pour lesquels la sortie du décodeur de canal est le mot $\underline{\hat{c}}$;
- $d\underline{u} = \prod_{i=1}^{m} du_i$ ($d\underline{u}$ est l'élément différentiel).

Il reste à choisir de manière optimale $V_s(\underline{c})$ et $V_c(\underline{\hat{c}})$. Le décodeur de canal à décision dure optimale est celui à maximum de vraisemblance :

$$\begin{aligned} \underline{y} \in V_c(\underline{\hat{c}}) &\iff \underline{\hat{c}} = \arg \max_{\underline{c} \in \mathcal{C}_{canal}} p(\underline{c}|\underline{y}) \\ &\iff \underline{\hat{c}} = \arg \max_{\underline{c} \in \mathcal{C}_{canal}} p(\underline{y}|\underline{c}) p(\underline{c}) \end{aligned} \tag{5.19}$$

où $p(\underline{c})$ est la probabilité que le codeur émette le mot $\underline{c}$ sur le canal, i.e.

$$p(\underline{c}) = \int_{\underline{u} \in V_s(\underline{c})} p(\underline{u}) d\underline{u}. \tag{5.20}$$

Afin de déterminer les régions $V_s(\underline{c})$, posons

$$\mu(\underline{\hat{c}}, \underline{c}) = \sum_{\underline{y} \in V_c(\underline{\hat{c}})} p(\underline{y}|\underline{c}). \tag{5.21}$$

D'après l'expression de la distorsion (5.18), le partitionnement optimal de l'espace $\mathbb{R}^m$ décrit par les mots de source est donné par :

$$V_s(\underline{c}) = \{\underline{u} / \forall \underline{c}' \in \mathcal{C}_{canal} \sum_{\underline{\hat{c}} \in \mathcal{C}_{canal}} \mu(\underline{\hat{c}}, \underline{c}) ||\underline{u} - L\underline{\hat{c}}_m||^2 \leq \sum_{\underline{\hat{c}} \in \mathcal{C}_{canal}} \mu(\underline{\hat{c}}, \underline{c}') ||\underline{u} - L\underline{\hat{c}}_m||^2\} \tag{5.22}$$

Ce partitionnement correspond à la condition du plus proche voisin prenant en compte les perturbations introduites par le canal. Les cas d'égalité dans la définition (5.22) sont résolus au hasard (pour une source gaussienne i.i.d., la probabilité d'observer l'égalité est nulle). Remarquons également que les régions de Voronoï $V_s(\underline{c})$ associées au code

de source et celles associées au code de canal $V_c(\underline{c})$ dépendent les unes des autres par l'intermédiaire des probabilités $p(\underline{c})$.

Ainsi, lorsque le code de canal et la matrice L sont fixées, il faut optimiser conjointement le décodeur de canal et le quantificateur implicite au codeur conjoint. Remarquons que si le codeur est max-entropique, le décodeur de canal devient indépendant du codeur conjoint. En revanche, ce dernier dépend toujours du décodeur de canal.

5.2.1 Choix de la matrice d'étiquetage L

Le choix de la matrice L est un problème *a priori* crucial pour déterminer les meilleures performances de ce système conjoint, tous les autres paramètres demeurant fixes. La démarche adoptée pour choisir L consiste à minimiser la distorsion lorsque le codeur conjoint et le décodeur de canal sont fixés, ce qui est approche similaire à celle de l'algorithme de Lloyd employé pour la conception de codeur/décodeur de source. Les probabilités $p(\underline{c})$ et $\mu(\underline{\hat{c}}, \underline{c})$ sont donc supposées fixes ainsi que les régions $V_s(\underline{c})$. Autrement dit, nous allons chercher une matrice L, *a priori* différente de celle qui a permis de fixer ces paramètres, optimisant le décodeur de source. La matrice L optimale est obtenue par minimisation de la distorsion donnée par l'équation (5.18). Cette distorsion étant une somme de carrés de normes, nous sommes assurés de trouver un minimum (et non un maximum). Posons

$$s_{\underline{c}} = \sum_{\underline{\hat{c}} \in \mathcal{C}_{canal}} \mu(\underline{\hat{c}}, \underline{c}) \underline{\hat{c}}_m. \tag{5.23}$$

$s_{\underline{c}}$ peut être interprété comme la sortie souple du décodeur de canal. La minimisation de la distorsion est alors équivalente à l'équation

$$L \left(\sum_{\underline{\hat{c}} \in \mathcal{C}_{canal}} \left(\sum_{\underline{c} \in \mathcal{C}_{canal}} p(\underline{c}) \mu(\underline{\hat{c}}, \underline{c}) \right) \underline{\hat{c}}_m \underline{\hat{c}}_m^t \right) = \sum_{\underline{c} \in \mathcal{C}_{canal}} \left(\int_{\underline{u} \in V_s(\underline{c})} \underline{u} p(\underline{u}) d\underline{u} \right) s_{\underline{c}}^t. \tag{5.24}$$

Puisque le critère est quadratique, sa minimisation nous conduit à une équation de type Wiener-Hopf. Le terme de droite de cette équation mesure la corrélation entre les décisions souples du décodeur de canal et les vecteurs moyens des régions $V_s(\underline{c})$. La matrice qui multiplie L mesure l'autocorrélation des mots du code de canal modulés.

5.2.2 Algorithme d'optimisation

Lorsque la matrice L vient d'être calculée grâce à l'équation (5.24), il nous faut recalculer les probabilités $\mu(\underline{\hat{c}}, \underline{c})$ et $p(\underline{c})$. Nous proposons alors d'optimiser la distorsion à rendement et code de canal donné en appliquant itérativement une mise à jour des divers paramètres inconnus. Supposons qu'à la fin de l'étape i, nous disposions de

la matrice d'étiquetage $L^{(i)}$ et des probabiblités $\mu^{(i)}(\underline{\hat{c}}, \underline{c})$ et $p^{(i)}(\underline{c})$ et de la distorsion correspondante $D^{(i)}$. L'étape $i+1$ se déroule en quatre phases :

- calculer la matrice $L^{(i+1)}$ à partir de l'équation (5.24) faisant intervenir $\mu^{(i)}(\underline{\hat{c}}, \underline{c})$ et $p^{(i)}(\underline{c})$;
- calculer $p^{(i+1)}(\underline{c})$ en déterminant le nouveau partitionnement de l'espace décrit par la source à partir de $L^{(i+1)}$ et de $\mu^{(i)}(\underline{\hat{c}}, \underline{c})$ (équations (5.20) et (5.22)) ;
- mettre à jour le décodeur de canal défini par (5.19) et calculer $\mu^{(i+1)}(\underline{\hat{c}}, \underline{c})$ à partir de l'équation (5.21) de $p^{(i+1)}(\underline{c})$;
- calculer la distorsion $D^{(i+1)}$ du nouveau système.

L'algorithme s'arrête lorsque l'écart relatif entre $D^{(i)}$ et $D^{(i+1)}$ est inférieur à une constante donnée (0.001 par exemple). **Cet algorithme est similaire à l'algorithme de Lloyd** au sens où il fixe tour à tour les éléments variables du système (codeur conjoint, décodeur de canal, décodeur de source).

Pour initialiser l'algorithme, lorsque les longueurs m et n et la dimension k sont données, nous choisissons le code linéaire binaire systématique de plus petite probabilité d'erreur par mot (à paramètre p du CBS fixé) : le code de canal $\mathcal{C}_{canal} = \underline{c}_1, \cdots, \underline{c}_{2^k}$ est donc déterminé. La matrice $L^{(0)}$ est initialisée à partir de ce code et du quantificateur vectoriel de dimension m sur k bits obtenu par l'algorithme LBG [33], lui-même initialisé par le quantificateur de dimension m sur 1 bit composé du vecteur $\sigma\sqrt{2/\pi}(1, \cdots, 1)$ et de son opposé. Le code de source défini par ce quantificateur est noté $\{\underline{v}_1, \cdots, \underline{v}_{2^k}\}$. Pour déterminer $L^{(0)}$, nous supposons que le quantificateur est max-entropique (i.e. $p^{(0)}(\underline{c}) = 2^{-k}$) et que le canal est parfait : l'équation (5.24) devient alors

$$L^{(0)} \left(\sum_{\underline{\hat{c}} \in \mathcal{C}_{canal}} \underline{\hat{c}}_m \underline{\hat{c}}_m^t \right) = \sum_{i=1}^{2^k} \left(\int_{\underline{u} \in V_s^{(0)}(\underline{c}_i)} \underline{u} p(\underline{u}) d\underline{u} \right) \underline{c}_{im}^t. \tag{5.25}$$

où $V_s^{(0)}(\underline{c}_i) = \{\underline{u} \,/\, \forall j, \quad 1 \leq j \leq 2^k, \quad ||\underline{u} - \underline{v}_i||^2 \leq ||\underline{u} - \underline{v}_j||^2\}$.

Résultats de simulations

Nous avons appliqué l'algorithme précédent en faisant varier m et n entre 3 et 10, la dimension des codes k variant de 1 à $\min(8, n)$. Les codes de canal retenus sont ceux trouvés au premier chapitre. 100000 vecteurs de source ont servi à estimer les probabilités $p(\underline{c})$ et $\mu(\underline{\hat{c}}, \underline{c})$ dépendant de la source ainsi que les vecteurs moyens intervenant dans le calcul de L (équation (5.24)).

Les figures 5.8-5.10 montrent les performances du schéma 5.7 pour différentes valeurs du paramètre du CBS ($p = 0.1, 0.01, 0.001$) : les performances des sytèmes sont représentées par des croix, l'OPTA défini par l'équation (5.15) par un trait plein. À rendement fixé, plusieurs systèmes sont donc simulés, la dimension du code de canal n'intervenant pas dans le rendement global : l'un des systèmes simulés (à rendement fixé) ne comprend pas de codage de canal alors que tous les autres mettent en œuvre

un codage de canal non universel (c'est-à-dire un code de canal dont la matrice génératrice n'est pas l'identité). Les cercles montrent les systèmes les plus performants : ils correspondent à des schémas où le code de canal est universel ! Le système devient, dans ce cas, un quantificateur vectoriel optimisé pour le canal (*Channel Optimized Vector Quantizer* étudié, entre autres, par Farvardin [17, 18, 19]) contraints par une relation linéaire entre les sorties du quantificateur et l'index les représentant : à faible complexité, il semble donc préférable d'avoir un rendement de source le plus élevé possible quitte à ne pas protéger l'index par des bits de parité lorsque ce dernier est transmis sur le canal. Encore une fois, la stratégie de conception indiquée par la preuve du théorème du codage source-canal, c'est-à-dire la conception d'un système tandem, ne semble pas indiquée dans les cas que nous avons étudiés. Remarquons aussi que les systèmes les plus performants ne sont pas tellement éloignés de l'OPTA, malgré leur faible complexité, lorsque le canal est faiblement bruité.

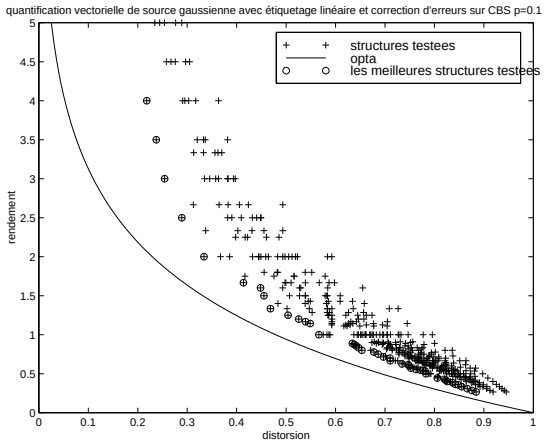


Fig. 5.8 – Performances du schéma 5.7 pour $p = 0.1$

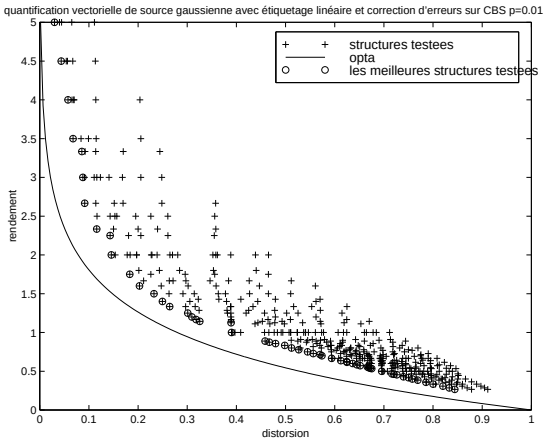


Fig. 5.9 – Performances du schéma 5.7 pour $p = 0.01$

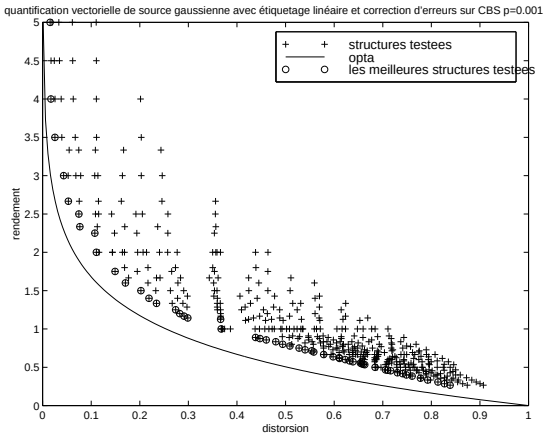


Fig. 5.10 – Performances du schéma 5.7 pour $p = 0.001$

5.3 Conclusions

Nous nous sommes intéressés dans ce chapitre à l'étude de codes réels pour la compression de source continue ou d'images et l'utilisation de ces codes dans le cadre du codage conjoint.

Dans un premier temps, nous avons présenté les codes BCH réels, qui sont la transposition des codes BCH binaires au corps des réels, le corps localisateur devenant le corps des complexes. Ils diffèrent cependant des codes BCH binaires en plusieurs points :

- il est possible de construire des codes BCH réel MDS pour n'importe quelles valeurs de la longueur n et de la dimension k, la seule exception étant le cas d'une longueur paire et d'une dimension impaire ;
- le dual d'un code BCH est un code BCH ;
- les racines du polynôme générateur des codes BCH réels MDS ne s'écrivent pas comme des puissances consécutives d'une racine n^{e} de l'unité primitive fixe (dans le cas réel, la première puissance dépend de la parité de la longueur et de la valeur de la distance BCH) ;
- la rayon de recouvrement des codes BCH réels MDS est maximal.

Les algorithmes existants dans le cas binaire s'adaptent aux codes réels : il suffit de prendre en compte le bruit de calcul en introduisant des seuils. La version la plus courante de l'algorithme de Peterson-Gorenstein-Zierler est sujette dans le cas binaire à des dysfonctionnements lorsque le mot à décoder est en dehors des sphères centrées sur les mots de code et de rayon la capacité de correction BCH. Dans le cas réel, de tels dysfonctionnements n'existent pas car, d'une part, le choix des racines du polynôme générateur du code BCH réel MDS n'est pas exactement similaire à celui effectué dans le cas binaire et d'autre part, la résolution des systèmes linéaires prend en compte toutes les équations à notre disposition et non une partie. L'algorithme de Berlekamp-Massey, tel qu'il est présenté par Berlekamp lui-même [5] n'est pas sujet à de tels dysfonctionnements et fonctionne pour n'importe quel corps, fini ou non.

Marshall [36, 35] et Wolf [66] ont suggéré l'emploi de ces codes réels pour la correction d'erreurs. Marshall a même suggéré de les employer comme codes de source : le principe de dualité entre codage de source et codage de canal nous a incité à examiner leurs performances dans le domaine du codage de source avec pertes, le critère retenu étant celui de la distance de Hamming (qui correspond aux bruits impulsifs).

Dans ce but, après avoir exposé les diverses méthodes rencontrées dans la littérature, nous avons proposé un algorithme de décodage complet et optimal des codes BCH réels MDS similaires à l'algorithme GMD de Forney [22] s'appuyant sur des méthodes de décodage algébrique corrigeant erreurs et effacements. La valeur du rayon de recouvrement de ces codes prend alors toute son importance : pour des sources à valeurs continues, les mots de source seront à une distance $n-k$ du plus proche mot de code avec une probabilité égale à 1 : le code étant MDS, C_n^{n-k} mots de code sont

alors à distance $n-k$ du mot de source. Il est alors possible de départager ces mots en choisissant celui qui minimise l'erreur quadratique, c'est-à-dire minimisant la puissance du bruit impulsif. Ce critère est secondaire par rapport au premier qui consiste à minimiser la distance de Hamming.

Nous avons alors appliqué ce décodeur de source à deux types de sources : la source gaussienne i.i.d. et des images. Il est apparu que les codes BCH réels semblent plus adaptés au codage d'images, c'est-à-dire de source corrélée, que de source blanche.

Dans la perspective d'un codage source-canal binaire symétrique conjoint, les mots d'information réels devraient être quantifiés puis codés. Nous avons alors choisis d'imposer une relation linéaire entre le code de source et le code de canal, cette relation linéaire définissant un code de source réel, et de considérer le critère de l'erreur quadratique moyenne, plus simple que la distance de Hamming. L'utilisation d'un code de canal linéaire binaire limite cependant le nombre de mots dans le dictionnaire de source. Nous avons cherché à optimiser cette relation en fonction du code de canal choisi en proposant un algorithme similaire à celui de Lloyd-Max et en nous restreignant à des codes de faibles longueurs. L'approche ainsi adoptée se rapproche du quantificateur COVQ proposé par Farvardin [17, 18, 19] à la différence que nous y insérons un code de canal et que nous imposons une relation linéaire entre les mots du code de source et les mots du code de canal. Les meilleurs systèmes trouvés correspondent à un code de canal universel ! À faible complexité, il vaut donc mieux construire un quantificateur COVQ de plus fine résolution que de chercher à protéger des effets du canal les étiquettes provenant d'un quantificateur de plus basse résolution. Les systèmes étudiés, même de faible complexité, ne sont pas très éloignés de l'OPTA.

Conclusions et perspectives

Toute thèse s'expose au rire des dieux.
Kierkegaard

Conclusions

Ce document a présenté plusieurs exemples de la dualité entre codage de source et codage de canal : cette dualité a permis d'utiliser plusieurs types de codes correcteurs d'erreurs en bloc - outil classique du codage de canal - pour comprimer plusieurs types sources avec pertes. Les codes correcteurs considérés dans nos travaux sont les codes en blocs linéaires, les codes arithmétiques et les codes BCH réels. Les sources étudiées sont la source q-aire symétrique, la source binaire asymétrique et la source gaussienne : nous avons étudié une panoplie d'outils couvrant diverses situations.

Source binaire symétrique, canal binaire symétrique et distance de Hamming

L'étude des cellules de Voronoï, qui déterminent les performances des codes linéaires tant dans le domaine du codage de canal que du codage de source, nous a permis de trouver des bornes inférieures des performances aussi bien en codage de canal, en codage de source qu'en codage conjoint. Ces bornes convergent vers les meilleures performances théoriquement possibles (OPTA). Ces bornes inférieures permettent de démontrer la réciproque des théorèmes du codage de canal et du codage de source, retrouvant ainsi des résultats bien connus de la théorie de l'information.

Nous avons aussi montré qu'il existe des codes linéaires aptes à comprimer efficacement la source q-aire symétrique et proposé un algorithme permettant de construire de bons codes linéaires pour la compression de la source symétrique. Nous avons établi également que les bons codes linéaires binaires de petite longueur pour la compression se confondent avec les bons codes de canal linéaires binaires.

La dualité nous a conduit à étudier le codage conjoint source binaire symétrique - canal binaire symétrique : ce modèle se retrouve dans nombre de réseaux de communication. Nous avons comparé les performances d'un système séparé, où codeur de source et codeur de canal sont donc conçus séparément, au système source ou canal qui ef-

fectue exclusivement du codage de source si le rendement global est inférieur à 1 et exclusivement du codage de canal si le rendement global est supérieur à 1. Ce système permet d'atteindre les meilleures performances possibles lorsque le canal est parfait ou lorsque le rendement global (alors égal à l'inverse du rendement canal) est supérieur à l'inverse de la capacité du canal. Nous avons illustré ce principe de dualité en remplaçant les codes linéaires binaires par des codes arithmétiques qui sont non linéaires mais qui possèdent néanmoins une structure permettant de simplifier leur décodage. Nous avons également utilisé un algorithme de Lloyd généralisé doublement pour améliorer ce système et étudié sa robustesse lorsque le canal varie lentement : la dualité permet à l'algorithme de Lloyd d'améliorer des codes correcteurs d'erreurs bien que cet algorithme ait été initialement développé uniquement pour le codage de source. Il permet aussi de trouver des décodeurs performants lorsque la sortie du canal est quantifiée plus finement que dans le cas du canal binaire symétrique.

Lorsque ce modèle (SBS, CBS et critère de distance de Hamming) est retenu, il est donc préférable de ne faire que du codage de source ou que du codage de canal, un mélange des deux réduisant parfois considérablement les performances, quel que soit le type de code considéré (linéaire ou non linéaire). Les outils développés dans ce document permettent de prévoir les performances et de trouver des bons codes. L'algorithme de Lloyd ne permet pas d'améliorer significativement les performances des codes linéaires dans le cas du canal binaire symétrique.

Perspectives

Le travail accompli, notamment dans l'étude du codage conjoint, s'est limité à des codes de petite longueur. La conclusion à laquelle nous sommes arrivés, à savoir que, dans les cas considérés, l'un des deux codeurs habituellement utilisés est inutile, est-elle encore valable lorsque nous augmentons la longueur des codes ?

Autrement dit, dans le cas de la source binaire symétrique et du canal binaire symétrique, existe-t-il des codes permettant à la structure source ou canal d'atteindre les meilleures performances théoriquement possibles quels que soient le rendement, la distorsion et la valeur du paramètre du canal binaire symétrique ?

Une telle étude doit s'appuyer sur des techniques de codage aléatoire. Une réponse affirmative parachèverait l'étude de ce cas simple mais inhérent à de nombreux systèmes de communications numériques tout en démontrant qu'une conception non séparée du système peut être aussi performante, tout en étant moins complexe, qu'une conception séparée.

Source binaire asymétrique, canal parfait et distance de Hamming

Les codes linéaires binaires nous ont par ailleurs servi à comprimer la source binaire asymétrique : après avoir donné les performances du codeur par syndrome développé

par Ancheta, nous avons examiné celles du codeur par région de Voronoï. Ces deux codeurs sont complémentaires dans le sens où le codeur par syndrome est efficace pour des sources fortement asymétriques alors que le codeur par région de Voronoï l'est pour des sources faiblement asymétriques. La théorie de l'information indique que cette source est bien codée par des codes non linéaires, aussi avons nous introduit de la non-linéarité en divisant cette source en deux sous-sources, chacune étant codée par un code linéaire. La recherche du mot de code le plus proche demeure simple alors que l'étiqueteur est aussi complexe que celui associé à un code non linéaire. Comme prévu, les performances de ce codeur sont meilleures que celles du codeur associé à un code linéaire. Puis nous avons recherché, grâce à l'algorithme de Lloyd initialisé par un code aléatoire tiré suivant un loi de probabilité issue de l'OPTA, des « bons » codes non linéaires. Le codeur associé est le plus complexe parmi les codeurs considérés pour cette source, néanmoins leurs performances outrepassent celles des autres codeurs.

L'algorithme de Lloyd permet donc de trouver de bons codes non linéaires pour la compression de source binaire asymétrique. Si un décodage simple est nécessaire, le codeur par syndrome d'Ancheta ou le codeur par région de Voronoï donnent des performances intéressantes. Néanmoins, les performances des codes en blocs de petite longueur pour la compression de la SBA dépendent largement du degré d'asymétrie de la source : plus la source est asymétrique, plus il est difficile d'obtenir des performances proches de l'OPTA.

Perspectives

Les meilleurs codes trouvés pour la compression de la source binaire asymétrique sont non linéaires. L'utilisation de codes linéaires pour comprimer cette source permet un gain en complexité car le décodage des codes linéaires est plus aisé que celui des codes non linéaires. Encore faut-il que les codes linéaires soient performants...

Afin de généraliser l'étude faite pour le modèle précédent aux sources binaires asymétriques, il conviendrait dans un premier temps de montrer l'optimalité ou la non optimalité des codes linéaires pour la compression avec pertes de ces sources. À notre connaissance, aucune étude théorique sur ce sujet n'a été entreprise. L'étude analogue menée pour la source symétrique a montré que l'existence de codes linéaires possédant un « bon » rayon de recouvrement permettait de déduire l'existence de bons codes linéaires pour la compression. L'asymétrie de la source modifie-t-elle ce critère ou non ?

Source arithmétique symétrique, canal parfait et distance arithmétique

La dualité a permis d'utiliser efficacement les codes arithmétiques pour comprimer la source arithmétique symétrique suivant le critère de distance arithmétique. Rappelons que la source arithmétique symétrique est équivalente à la sortie d'un codeur max-entropique : les codes arithmétiques peuvent donc comprimer efficacement la sortie d'un tel codeur.

Perspectives

Les performances des codes arithmétiques tant en codage de canal qu'en codage de source ont été peu explorées. Seule la distance arithmétique des codes a été l'objet de recherche. Existe-t-il des paramètres simples permettant de prévoir ces performances ? La question semble peu aisée car la distribution des poids arithmétiques jusqu'à un certain entier donné n'admet pas, à notre connaissance, de formule littérale simple.

Source gaussienne, canal parfait et distance de Hamming

Le dernier exemple de dualité donné dans ce document concerne les codes réels et plus particulièrement les codes BCH réels MDS. Ces codes, utilisés en codes de canal, peuvent combattre le bruit impulsif. Ils peuvent donc être utilisés en compression de source pour éliminer un bruit impulsif de la source : le critère de fidélité pertinent est alors la distance de Hamming. Un algorithme proche du GMD développé par Forney permet de coder une source continue, et non plus discrète, suivant ce critère : la différence majeure réside dans le fait que nous ne pouvons pas supputer la position des effacements à partir de l'observation mais que nous en essayons un certain nombre. La complexité du pire des cas est alors gouvernée par le rayon de recouvrement de ces codes, rayon de valeur maximale (i.e. égal à la longueur moins la dimension). Par conséquent, pour nombre de sources, le mot de source sera en probabilité 1 à distance égale de plusieurs mots de codes. Il est alors possible de choisir un de ces mots en appliquant un second critère, celui de l'erreur quadratique moyenne, par exemple. Nous avons ainsi réalisé un codeur de source à double critère.

Les codes BCH réels permettent donc de comprimer presque sans perte une source gaussienne au sens de la distance de Hamming : la compression presque sans perte correspond alors à une valeur faible du rayon de recouvrement, limitant ainsi la complexité du codeur de source.

Perspectives

Les codes BCH réels furent utilisés pour la compression de source suivant le critère de fidélité défini par la distance de Hamming : l'obtention d'une expression concise de la fonction rendement-distorsion suivant ce critère pour des sources continues serait intéressante ainsi que l'expression de cette fonction lorsque les deux critères (distance de Hamming et EQM) sont considérés.

Source gaussienne, canal binaire symétrique et erreur quadratique moyenne

Nous avons étudié ce modèle en imposant une relation linéaire entre le code de source défini par un code linéaire réel et le code de canal défini par un code linéaire binaire. Cette relation linéaire permet de simplifier le décodeur de source et donne naissance à un codeur conjoint : un algorithme, similaire à l'algorithme de Lloyd,

est utilisé pour optimiser la relation linéaire, prenant ainsi en compte le canal et le comportement du décodeur de canal. À nouveau, le système le plus simple est le plus performant : le code de canal le meilleur est le code universel !

Dans un tel modèle, il est donc préférable de construire un codeur de haute résolution prenant en compte les effets du canal à un codeur de plus faible résolution dont la sortie est protégée par un code correcteur d'erreurs. Cependant, l'élaboration de codeurs de haute résolution robustes aux perturbations introduites par le canal est une tâche difficile.

Perspectives

L'obtention de bornes de performances dépendant du code de canal utilisé permettrait de confirmer ou d'infirmer notre observation. De plus, elle suggérerait peut-être quel type de codes de plus grande dimension doit être employé lorsque la conception de quantificateurs robustes devient impossible car trop complexe.

Perspectives générales : utilisation des briques développées dans des systèmes complets

Les diverses « briques de base » étudiées dans ce document peuvent être incorporées dans des systèmes plus complets : nous allons donner quelques exemples d'utilisation de ces briques.

Transmission d'une source quantifiée en sous-bandes

Les outils développés dans le cadre d'une source binaire (symétrique ou asymétrique) et d'un canal binaire symétrique suivant le critère de la distance de Hamming peuvent s'appliquer lorsqu'une source quelconque est décomposée en sous-bandes, chacune des sous-bandes donnant lieu à une quantification différente et à un étiquetage. Ce système est représenté sur la figure 2 où la source est analysée en K bandes.

La source unique est donc à l'origine de K trames binaires (assimilables à des sources symétriques ou non) dont l'influence sur la distorsion globale n'est pas équivalente. Un système source ou canal (de rendement n_i/m_i), distinct pour chaque trame binaire, permet alors de transmettre ces trames. En raison de la simplicité du système source ou canal, il est envisageable d'optimiser le débit binaire (c'est-à-dire le rapport des sommes $\sum_{i=1}^{K} n_i$ et $\sum_{i=1}^{K} m_i$) en fonction de l'état du canal (si une voie de retour est disponible) et de la qualité de service souhaitée. Zahir-Azami [69] a employé un tel système en employant le critère de l'EQM qui est lié à celui de la distance de Hamming. En effet, lors du calcul de la fonction rendement-distorsion associée à une source binaire symétrique et au critère de l'erreur quadratique moyenne, il apparaît que :

$$\mathrm{EQM} = D(1 - D),$$

où D est la distorsion de Hamming. L'optimisation du débit binaire nécessite, pour être efficace, de disposer d'un certain nombre de systèmes source ou canal de référence : les codes ainsi considérés peuvent être les codes linéaires ou les codes arithmétiques construits grâce à la distance arithmétique.

Un tel système permet, par exemple, de transmettre une image : la décomposition en sous-bandes s'assimile alors à une transformée qui opère sur une image découpée en blocs. L'opération délicate consiste alors à mettre en adéquation le débit des quantificateurs avec les systèmes source ou canal employés pour atteindre la qualité de service voulue.

Enregistrement numérique magnétique avec pertes

Le canal discret correspondant à l'enregistrement numérique sur bande magnétique est habituellement modélisé par un filtre de réponse impulsionnelle $1 - z^{-2}$ auquel s'ajoute du bruit. Abstraction faite du bruit, ce canal se comporte donc comme un code arithmétique. Une source continue peut donc être décomposée en sous-bandes puis quantifiée comme précédemment, le système source ou canal se fondant alors sur des codes arithmétiques et sur la distance arithmétique pour combattre le bruit. L'égalisation du canal se fait donc grâce à un décodeur arithmétique. La distorsion globale dépend alors de la répartition du débit binaire total entre les sous-bandes. L'utilisation des codes arithmétiques est donc envisagée comme une alternative aux codes traditionnellement utilisés dans ce domaine (*runlength-limited codes*).

Restauration d'enregistrements

La restauration d'enregistrements peut nécessiter parfois de supprimer des *clicks*, c'est-à-dire un bruit impulsif (généré par le récepteur dans le contexte de restauration d'un signal émis par onde radio). La compression de source par les codes BCH réels peut permettre de restaurer l'enregistrement, une fois celui-ci numérisé : la distance de Hamming modélisant ce bruit, l'algorithme de compression que nous avons présenté permet d'obtenir une source reconstruite où la puissance du bruit impulsif est réduite. Bien sûr, nous ne prétendons pas reconstruire au mieux la source par ce simple moyen.

Il est tout d'abord nécessaire d'identifier à peu près la position des *clicks* dans tout l'enregistrement. Une fois ceci accompli, la source est reconstruite à partir des échantillons juste antérieurs et postérieurs aux *clicks*. La difficulté réside alors dans le choix de la longueur et de la capacité de correction du code BCH : les contraintes imposées lors de la reconstruction ne doivent pas être trop strictes afin de ne pas altérer encore plus la source qui ne satisfait pas *a priori* à ces contraintes. Une fois ce premier traitement effectué, il est possible de recourir à des méthodes plus fines pour améliorer le résultat, les codes BCH permettant de réaliser un prétraitement.

Codage de la parole à bas débit

Les codeurs de parole à bas débit sont utilisés dans le cadre de communications radio-mobiles, par exemple, en raison de la difficulté à transmettre sur ce type de canal. Les codeurs de parole à bas débit actuels exploitent la prédiction par bloc ainsi que le codage différentiel : leur principe de fonctionnement est représenté sommairement sur la figure 3. De brefs signaux de parole (d'une durée d'environ 10 à 20 ms) sont analysés : un filtre prédicteur (parfois deux, un pour le « long terme », l'autre pour le « court terme ») est calculé sur cette période, permettant alors de quantifier l'erreur de prédiction. Il faut à la fois transmettre au décodeur de parole l'erreur quantifiée et les coefficients du filtre prédicteur. Le système GSM met en œuvre une protection hiérarchique de ces données, l'erreur quantifiée étant moins protégée que les coefficients du filtre. Comment alors optimiser les différents quantificateurs en fonction du code de canal choisi (ou imposé par la norme GSM par exemple) ? Notre étude sur les quantificateurs à étiquetage linéaire montre que cette optimisation n'est pas facile. Néanmoins, c'est une piste à explorer pour améliorer les performances en changeant le moins possible un système déjà normalisé.

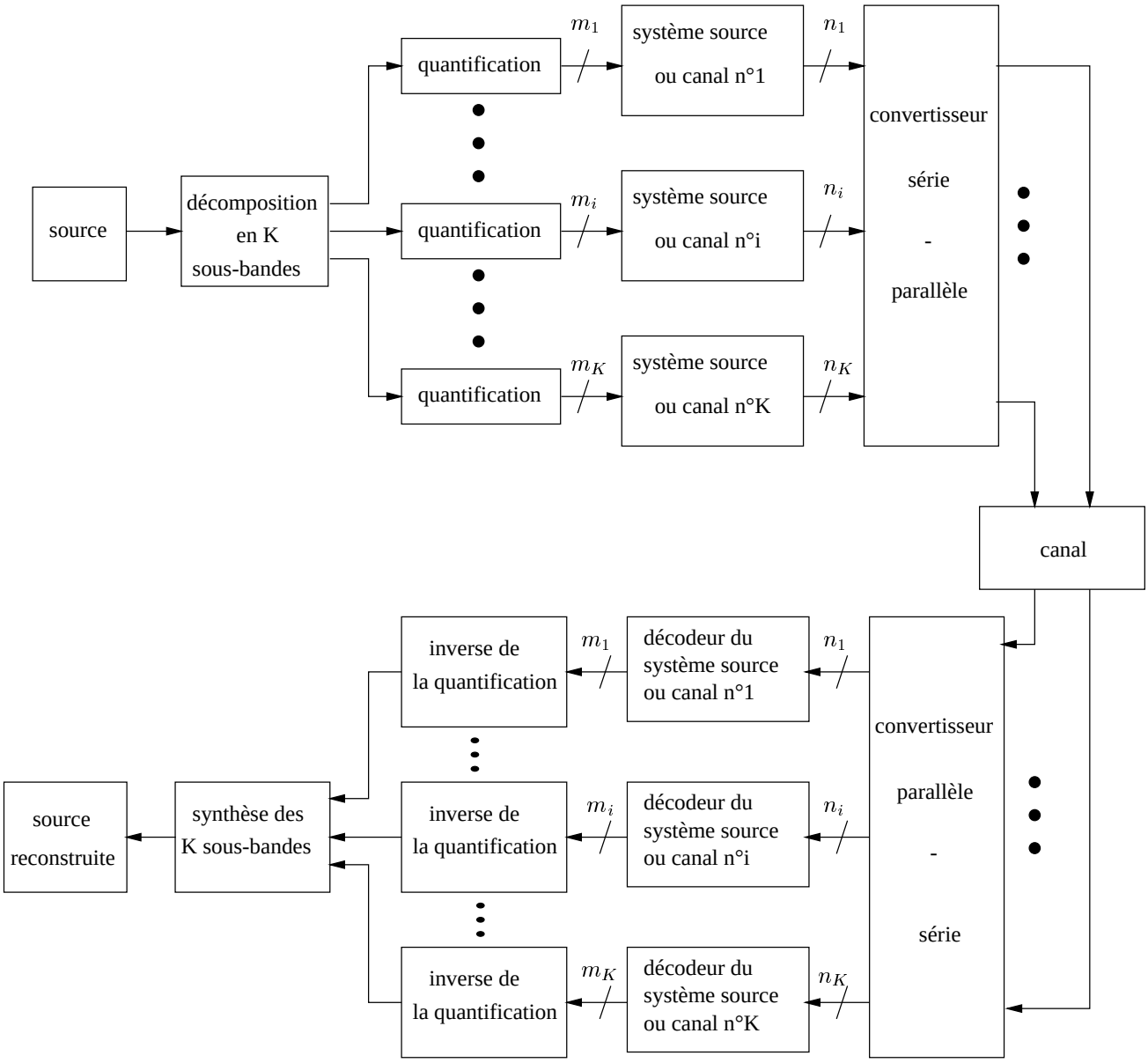


FIG. 2 – Transmission d'une source quantifiée en sous-bandes

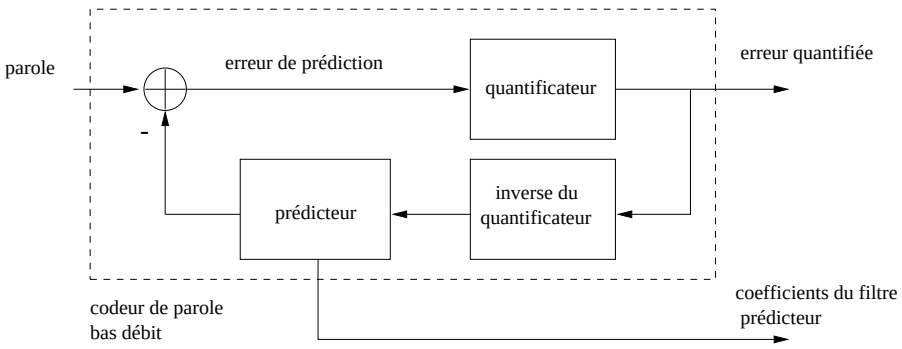


FIG. 3 – Codeur de parole bas débit

Bibliographie

[1] T.C. Ancheta, Jr. Syndrome-source-coding and its universal generalization. *IEEE Transactions on Information Theory*, 22(4) :432–436, July 1976.

[2] G. Battail. Communication en présence de bruit gaussien. *Écho des recherches*, 139 :3–12, premier trimestre 1990.

[3] E. Bedrosian. Weighted PCM. *IRE Transactions on Information Theory*, 4 :45–49, March 1958.

[4] T. Berger. *Rate Distortion Theory.* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[5] E.R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[6] J.-C. Bic, D. Duponteil, and J.-C. Imbeaux. *Éléments de communications numériques.* Collection technique et scientifique des télécommunications. Dunod, Paris, 1986.

[7] R.E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley Publishing Company, Reading, Massachusets, 1983.

[8] S.H. Chang and N. Tsao-Wu. Distance and structure of cyclic arithmetic codes. In *Proceedings Hawaii International Conference on Systems Sciences 1968*, pages 463–466, 1968.

[9] R.T. Chien. On linear residue codes for burst-error correction. *IEEE Transactions on Information Theory*, 10(2) :127–133, April 1964.

[10] W.E. Clark and J.J. Liang. On arithmetic weight for a general radix representation of integers. *IEEE Transactions on Information Theory*, 19 :823–826, November 1973.

[11] G. Cohen and P. Frankl. Good coverings of Hamming spaces with spheres. *Discrete Mathematics*, 56 :125–131, 1985.

[12] G.D. Cohen, M.G. Karpovsky, H.F. Mattson Jr, and J.R. Schatz. Covering radius - survey and recent results. *IEEE Transactions on Information Theory*, 31(3) :328–343, May 1985.

[13] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991.

[14] T.R. Crimmins, H.M. Horwitz, C.J. Palermo, and R.V Palermo. Minimization of mean-square error for data transmitted via group codes. *IEEE Transactions on Information Theory*, 15 :72–78, January 1969.

[15] P. Duhamel and O. Rioul. Codage combiné source/canal : enjeux et approches. In *Actes du Seizième Colloque sur le traitement du signal et des images (GRETSI)*, pages 699–704, Grenoble, Septembre 1997.

[16] A.A. El Gamal, L.A. Hemachandra, I. Shperling, and V.K Wei. Using simulated annealing to design good codes. *IEEE Transactions on Information Theory*, 33(1) :116–123, January 1987.

[17] N. Farvardin. A study of vector quantization for noisy channels. *IEEE Transactions on Information Theory*, 36(4) :799–809, July 1990.

[18] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels : An approach to combined source-channel coding. *IEEE Transactions on Information Theory*, 33(6) :827–838, November 1987.

[19] N. Farvardin and V. Vaishampayan. On the performance and complexity of channel-optimized vector quantizers. *IEEE Transactions on Information Theory*, 37(1) :155–160, January 1991.

[20] T.J. Ferguson and J.H. Rabinowitz. Self-synchronizing Huffman codes. *IEEE Transactions on Information Theory*, 30(4) :687–693, July 1984.

[21] T. Fine. Properties of an optimum digital system and applications. *IEEE Transactions on Information Theory*, 10 :287–296, October 1964.

[22] G.D. Forney, Jr. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2) :125–131, April 1966.

[23] A. Gabay, P. Duhamel, and O. Rioul. Exploitation de la redondance résiduelle pour combattre le bruit. Technical report, CNES, November 1998.

[24] T.J. Goblick Jr. *Coding for a Discrete Information Source with a Distortion Measure*. PhD thesis, MIT, Cambridge, Massachusets, 1962.

[25] R. Hagen and P. Hedelin. Robust vector quantization by a linear mapping of a block code. *IEEE Transactions on Information Theory*, 45(1) :200–218, January 1999.

[26] C.R.P Hartmann. Decoding beyond the BCH bound. *IEEE Transactions on Information Theory*, 18(3) :441–444, May 1972.

[27] A. Kerdock and J.K. Wolf. On the minimum distortion of block codes for a binary symmetric source. *IEEE Transactions on Information Theory*, 18(3) :433–435, May 1972.

[28] T. Kløve. The weight distributions of cosets. *IEEE Transactions on Information Theory*, 40(3) :911–913, May 1994.

[29] P. Knagenhjelm. How good is your index assignment ? In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages II.423–II.426, Minneapolis, Minnesota, April 1993.

[30] P. Knagenhjelm and E. Agrell. The Hadamard transform - a tool for index assignment. *IEEE Transactions on Information Theory*, 42(4) :1139–1151, July 1996.

[31] G.R. Lang and F.M. Longstaff. A Leech lattice modem. *IEEE Journal of Selected Areas in Communications*, 7 :968–973, August 1989.

[32] R. Laroia, N. Farvardin, and S.A. Tretter. On optimal shaping of multidimensional constellations. *IEEE Transactions on Information Theory*, 40(4) :1044–1056, July 1994.

[33] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1) :84–95, January 1980.

[34] S.P. Lloyd. Least squares quantization in PCM's, technical report. *Bell Telephone Laboratories Paper*, 1957.

[35] T.G. Marshall, Jr. Codes and algorithms for simultaneous error correction and rate reduction implementable with standard digital processors. In *Proceedings IEEE Global Telecommunications Conference*, pages 936–940, Miami, Florida, November 1982.

[36] T.G. Marshall, Jr. Coding of real-number sequences for error correction : A digital signal processing problem. *IEEE Journal on Selected Areas in Communications*, 2(2) :381–392, March 1984.

[37] J.L. Massey. Joint source and channel coding. In *Communications Systems and Random Process Theory (Amsterdam, The Netherlands)*, pages 279–293, Amsterdam, The Netherlands, 1978.

[38] J.L. Massey and O.N. Garcia. Error-correcting codes in computer arithmetic. In *Advances in Information Systems Science*, volume 4, chapter 5, pages 273–326. Plenum Press, 1972.

[39] J. Max. Quantization for minimum distortion. *IEEE Transactions on Information Theory*, 6(1) :7–12, 1960.

[40] J.C. Maxted and J.P. Robinson. Error recovery for variable length codes. *IEEE Transactions on Information Theory*, 31(6) :794–801, November 1985.

[41] R.J. McEliece. *The Theory of Information and Coding, A Mathematical Framework for Communication*, volume 3 of *Encyclopedia of mathematics and its applications*. Addison-Wesley Publishing Company, Reading, Massachusets, 1977.

[42] S.W. McLaughlin, D.L. Neuhoff, and J.J Ashley. Optimal binary index assignments for a class of equiprobable scalar and vector quantizers. *IEEE Transactions on Information Theory*, 41(6) :2031–2037, November 1995.

[43] F.J. McWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library*. North-Holland, Amsterdam, The Netherlands, 1977.

[44] A. Méhes and K. Zeger. Binary lattice vector quantization with linear block codes and affine index assignments. *IEEE Transactions on Information Theory*, 44(1) :79–94, January 1998.

[45] M.E. Monaco and J.M. Lawler. Corrections and additions to "error recovery for variable length codes". *IEEE Transactions on Information Theory*, 33(3) :454–456, May 1987.

[46] B.L. Montgomery and J. Abrahams. Synchronization of binary source codes. *IEEE Transactions on Information Theory*, 32(6) :849–854, November 1986.

[47] P.S. Naidu. *Modern Spectrum Analysis of Time Series.* CRC Press Inc., USA, 1996.

[48] P.G. Neumann. Efficient error-limiting variable length codes. *IEEE Transactions on Information Theory*, 8(4) :292–304, July 1962.

[49] J.E.M. Nilsson. An algebraic procedure for decoding beyond e_{bch}. *IEEE Transactions on Information Theory*, 42(2) :649–652, March 1996.

[50] W.W. Peterson and J.E. Weldon. *Error Correcting Codes.* MIT Press, Massachusets, 2nd edition, 1972.

[51] C. Pépin. *Quantification vectorielle et codage conjoint source-canal par les réseaux de points.* PhD thesis, E.N.S.T., Paris, décembre 1997.

[52] S. Roman. *Coding and Information Theory.* Graduate Text in Mathematics. Springer-Verlag, New York, 1992.

[53] D.V. Sarwate and R.D. Morrison. Decoder malfunction in BCH decoders. *IEEE Transactions on Information Theory*, 36(4) :884–889, July 1990.

[54] P. Shankar. On BCH codes over arbitrary integer rings. *IEEE Transactions on Information Theory*, 25(4) :480–483, July 1979.

[55] C.E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, 4 :142–163, 1959.

[56] J. Shiu and J.-L. Wu. Real-number error-control coding as a new technique for nonlinear filtering. In *IEEE Workshop on Nonlinear Signal and Image Processing*, pages 650–653, 1995.

[57] J. Shiu and J.-L. Wu. Class of majority decodable real-number-codes. *IEEE Transactions on Communications*, 44(3) :281–283, March 1996.

[58] P.H. Skinnemoen. *Robust Communication with Modulation Organized Vector Quantization.* PhD thesis, Norwegian Institute of Technology, 1994.

[59] M. Srinivasan and D.V. Sarwate. Malfunction in the Peterson-Gorenstein-Zierler decoder. *IEEE Transactions on Information Theory*, 40(5) :1649–1653, September 1994.

[60] P. Stokes. The domain of covering codes. In H. Stichtenoth and M.A. Tsfasman, editors, *Coding Theory and Algebraic Geometry*, volume 1518 of *Lecture Notes in Mathematics*, pages 170–177. Springer Verlag, New York, 1991.

[61] D.D. Sullivan. A fundamental inequality between the probabilities of binary subgroups and codes. *IEEE Transactions on Information Theory*, 13(1) :91–94, January 1967.

[62] V.A. Vaishampayan and N. Farvardin. Joint design of block source codes and modulation signal sets. *IEEE Transactions on Information Theory*, 38(4) :1230–1248, July 1992.

[63] J.A. van der Horst and T. Berger. Complete decoding of triple error-correcting binary BCH codes. *IEEE Transactions on Information Theory*, 22(2) :138–147, March 1976.

[64] A.J. VITERBI and J.O OMURA. *Principles of Digital Communication and Coding.* McGraw-Hill Book Company, New York, 1979.

[65] G.A. WOLF and G.R. REDINBO. The optimum mean-square estimate for decoding binary block codes. *IEEE Transactions on Information Theory*, 20(3) :344–351, May 1974.

[66] J.K. WOLF. Decoding of Bose-Chaudhuri-Hocquengehm codes and Prony's method for curve fitting. *IEEE Transactions on Information Theory*, 13(6) :608, October 1967.

[67] J.K. WOLF. Redundancy, the discrete Fourier transform, and impulse noise cancellation. *IEEE Transactions on Communications*, 31(3) :458–461, March 1983.

[68] J.-L. WU and SHIU J. Discrete cosine transform in error control coding. *IEEE Transactions on Communications*, 43(5) :1857–1861, May 1995.

[69] S.B. ZAHIR AZAMI. *Codage conjoint source/canal, protection hiérarchique.* PhD thesis, E.N.S.T., Paris, 1999.

Annexe A

Inégalités binomiales

Nous rappelons ici quelques inégalités portant sur les coefficients binomiaux : ces inégalités, au demeurant classiques, sont redémontrées dans une annexe de [52].

A.1 Coefficient binomial

Le coefficient binomial C_n^k, qui donne le nombre de sous-ensembles de cardinal k d'un ensemble de cardinal n, est défini par

$$0 \leq k \leq n \quad C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{k!}.$$

Si k est plus grand que n, alors $C_n^k = 0$.

Théorème A.1.1 *Soient $n \in \mathbb{N}$ et $\delta \in]0,1[$ tel que $\delta n \in \mathbb{N}$ alors*

$$\frac{2^{nH_2(\delta)}}{\sqrt{8n\delta(1-\delta)}} < C_n^{n\delta} < \frac{2^{nH_2(\delta)}}{\sqrt{2\pi n\delta(1-\delta)}} \tag{A.1}$$

où la fonction H_2 désigne l'entropie binaire et est définie par

$$\begin{array}{rccl} H_2: & [0,1] & \to & [0,1] \\ & x & \mapsto & -x\log_2 x - (1-x)\log_2(1-x) \end{array}$$

Ce théorème découle de la formule de Stirling ($n! = (n/e)^n\sqrt{2\pi n}e^{1/(12n)+o(1/n)}$).

A.2 Somme de coefficients binomiaux

L'objectif est d'encadrer une somme de coefficients binomiaux. Soient n un entier naturel et q un réel strictement plus grand que 1. Soit δ un nombre réel positif inférieur

ou égal à $1-1/q$ tel que le produit δn soit un entier naturel. Soit z un nombre réel de l'intervalle $]0,1]$. La somme de coefficients binomiaux peut être majorée par :

$$\sum_{i=0}^{n\delta}(q-1)^i C_n^i \leq \sum_{i=0}^{n\delta}(q-1)^i C_n^i z^{i-n\delta} \leq (z^{-\delta}+(q-1)z^{1-\delta})^n$$

Nous minimisons la borne supérieure en choisissant $z=\frac{\delta}{(q-1)(1-\delta)}$, qui, par hypothèse sur δ, est bien compris entre 0 et 1. La borne supérieure ainsi obtenue s'écrit alors :

$$\sum_{i=0}^{n\delta}(q-1)^i\binom{n}{i} \leq (q-1)^{n\delta}\left(\left(\frac{1-\delta}{\delta}\right)^{\delta}\left(1+\frac{\delta}{1-\delta}\right)\right)^n = (q-1)^{n\delta}2^{nH_2(\delta)} \qquad \text{(A.2)}$$

On reconnaît dans l'inégalité (A.2) la borne de Chernoff [13] qui ne peut s'appliquer que lorsque δ est inférieur ou égal à $1-1/q$. Par ailleurs, la somme de coefficients binomiaux est minorée par son dernier terme puisque lesdits termes sont tous positifs. Grâce au théorème A.1.1, on obtient le théorème suivant :

Théorème A.2.1

$$\forall n\in\mathbb{N}\quad \forall q\in\mathbb{R},\ q>1\quad \forall\delta\in]0,1-1/q]\ \textit{tel que}\ \delta n\in\mathbb{N},$$

$$\frac{2^{nH_2(\delta)}(q-1)^{n\delta}}{\sqrt{8n\delta(1-\delta)}} < \sum_{i=0}^{n\delta}(q-1)^i C_n^i \leq (q-1)^{n\delta}\,2^{nH_2(\delta)}$$

Ce théorème permet donc d'estimer facilement le nombre de mots à l'intérieur d'une boule de rayon δn défini par la distance de Hamming. Une autre minoration de ce nombre existe [60] :

$$0\leq\delta\leq 1-1/q\quad \frac{2^{nH_2(\delta)}(q-1)^{n\delta}}{n+1} < \sum_{i=0}^{n\delta}(q-1)^i C_n^i. \qquad \text{(A.3)}$$

Lorsque le poids des mots suit une loi binomiale de paramètre p, un autre théorème, qui se déduit du précédent, nous permet d'évaluer le nombre de mots à l'intérieur d'une sphère de rayon δn et centrée sur le mot nul.

Théorème A.2.2 *Soit n un entier naturel, soient p et δ deux nombres réels de l'intervalle $]0,1]$ tels que δn soit un entier naturel, alors*

– *si $\delta\leq p$, alors*

$$\frac{p^{\delta n}(1-p)^{n(1-\delta)}2^{nH_2(\delta)}}{\sqrt{8n\delta(1-\delta)}} \leq \sum_{i=0}^{\delta n} C_n^i p^i(1-p)^{n-i} \leq p^{\delta n}(1-p)^{n(1-\delta)}2^{nH_2(\delta)}$$

– *si $\delta > p$, alors*

$$\frac{p^{\delta n}(1-p)^{n(1-\delta)}2^{nH_2(\delta)}}{\sqrt{8n\delta(1-\delta)}} \leq \sum_{i=\delta n}^{n} C_n^i p^i(1-p)^{n-i} \leq p^{\delta n}(1-p)^{n(1-\delta)}2^{nH_2(\delta)}$$

Annexe B

Quelques propriétés des cellules de Voronoï

B.1 Définition et distribution des distances

Rappelons dans un premier temps la définition d'une cellule de Voronoï. Soit $\mathcal{C}$ un code q-aire de longueur m et de taille M. La cellule de Voronoï du mot de code $\underline{v}$, notée $V(\underline{v})$ est définie par :

$$V(\underline{v}) = \{\underline{u} \in F_q^m, \quad \forall \underline{v'} \in \mathcal{C} \quad d_H(\underline{u}, \underline{v}) \leq d_H(\underline{u}, \underline{v'})\} \tag{B.1}$$

$\underline{v}$ est donc le mot de code le plus proche du mot $\underline{u}$. Du point de vue du codage de source, $\underline{v}$ est interprété comme la valeur quantifiée de $\underline{u}$. La définition (B.1) est cependant imprécise : il apparaît en effet que les mots $\underline{u}$ équidistants de deux (ou plus) mots de code appartiennent à plusieurs régions ! Les frontière des cellules de Voronoï sont donc mal définies lorsque l'on utilise (B.1). Dans bien des cas, il est souhaitable que les régions de Voronoï forment une partition de F_q^m. À cette fin, lorsqu'un mot $\underline{u}$ est équidistant de plusieurs mots de code, la cellule de Voronoï à laquelle il appartient est choisie au hasard.

Intéressons-nous à la distribution des distances à l'intérieur des cellules de Voronoï en introduisant $\alpha_i(\underline{v})$ et α_i leur moyenne :

$$\begin{aligned} 0 \leq i \leq m,\ \underline{v} \in \mathcal{C} \quad \alpha_i(\underline{v}) &= |\{\underline{u} \in V(\underline{v}), d_H(\underline{u}, \underline{v}) = i\}| \\ 0 \leq i \leq m \quad \alpha_i &= \frac{1}{M} \sum_{\underline{v} \in \mathcal{C}} \alpha_i(\underline{v}) \end{aligned} \tag{B.2}$$

Lemme B.1.1 *Les coefficients $(\alpha_i)_{0 \leq i \leq m}$ ont les propriétés suivantes :*

$$0 \leq \alpha_i \leq C_m^i (q-1)^i \tag{B.3}$$

$$\sum_{i=0}^{m} \alpha_i = \frac{q^m}{M} \tag{B.4}$$

Preuve
L'inégalité (B.3) se déduit de :

$$\alpha_i(\underline{v}) \leq |\{\underline{u} \in F_q^m, d_H(\underline{u}, \underline{v}) = i\}| = (q-1)^i C_m^i.$$

(B.4) est la conséquence du partitionnement de F_q^m en régions de Voronoï :

$$\sum_{i=0}^{m} \alpha_i(\underline{v}) = |V(\underline{v})|$$
$$\sum_{i=0}^{m} \alpha_i = \frac{1}{M} \sum_{\underline{v} \in \mathcal{C}} |V(\underline{v})| = \frac{q^m}{M}$$

□

Dans le cas d'un code linéaire de dimension k ($M = q^k$), il est possible de choisir des cellules de Voronoï toutes de même cardinal q^{n-k} : elles sont alors toutes translatées les unes des autres. Ainsi, lorsqu'on a fixé la frontière d'une région de Voronoï, on fixe en même temps les frontières des autres régions. Par conséquent $\alpha_i(\underline{v})$ est indépendant de $\underline{v}$ et on peut identifier α_i avec $\alpha_i(\underline{0})$, i.e. le nombre de chefs de classe de poids i : α_i est donc un entier. La cellule de Voronoï d'un code q-aire linéaire est stable par multiplication par un scalaire, par conséquent α_i est un multiple de $q-1$ lorsque i est non nul.

B.2 Minoration de $\Sigma_i f(i)\alpha_i$

Soient $f : \mathbb{R} \to \mathbb{R}$ une fonction croissante et d un nombre réel positif. Notre but est de minorer la somme des valeurs prises par f sur l'ensemble des entiers naturels pondérés par les $(\alpha_i)_{i \in \mathbb{N}}$ définis dans la section précédente. Distinguons dans l'ensemble de sommation deux sous-ensembles : les termes tels que i est strictement inférieur à d et ceux pour lesquels i est supérieur ou égal à d. Par hypothèse sur f, nous pouvons écrire :

$$\sum_{i=0}^{m} f(i)\alpha_i \geq \sum_{i=0}^{d-1} f(i)\alpha_i + f(d) \sum_{i=d}^{m} \alpha_i.$$

Récrivons alors (B.4) en $\sum\limits_{i \geq d} \alpha_i = q^m/M - \sum\limits_{i<d} \alpha_i$ et injectons cette égalité dans l'inégalité précédente :

$$\sum_{i=0}^{m} f(i)\alpha_i \geq f(d)\frac{q^m}{M} - \sum_{i=0}^{d-1}(f(d) - f(i))\alpha_i$$

Enfin utilisons l'inégalité (B.3) pour obtenir une borne $g(d)$ indépendante des α_i :

$$\sum_{i=0}^{m} f(i)\alpha_i \geq f(d)\frac{q^m}{M} - \sum_{i=0}^{d-1}(f(d) - f(i))(q-1)^i C_m^i = g(d)$$

Afin de maximiser $g(d)$, nous calculons la différence $g(d) - g(d-1)$:

$$g(d) - g(d-1) = (f(d) - f(d-1)) \left(\frac{q^m}{M} - \sum_{i=0}^{d-1} (q-1)^i C_m^i \right)$$

Puisque f est croissante, g est maximum lorsque d est le plus grand entier naturel vérifiant

$$\sum_{i=0}^{d-1} (q-1)^i C_m^i \leq \frac{q^m}{M}, \tag{B.5}$$

ou, de manière équivalente, le plus petit entier tel que $\sum_{i \leq d} (q-1)^i C_m^i \geq \frac{q^m}{M}$ lorsque M est strictement plus grand que 1. Par la suite, $g(d)$ désignera la borne inférieure après maximisation par le bon choix de l'entier d. Cette borne est valable pour tous les codes en bloc, qu'ils soient linéaires ou non.

La valeur de d ayant été ainsi calculée, on peut interpréter $d-1$ comme la capacité de correction maximale d'un code de longueur et de taille données.

Déterminons à présent les cas pour lesquels la borne $g(d)$ se confond avec la somme $\sum_i f(i)\alpha_i$.

Lemme B.2.1 *Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction strictement croissante. La borne inférieure $g(d)$ est égal à $\sum_i f(i)\alpha_i$ si et seulement si le code $\mathcal{C}$ est parfait ou quasi-parfait.*

PREUVE

Supposons que

$$g(d) = \frac{q^m}{M} f(d) - \sum_{i<d} (f(d) - f(i))(q-1)^i C_m^i = \sum_i f(i)\alpha_i$$

En utilisant la formule (B.4) pour remplacer q^m/M par $\sum_i \alpha_i$ et en réarrangeant les termes, l'égalité précédente devient

$$\sum_{i=d}^{m} (f(i) - f(d))\alpha_i = \sum_{i=0}^{d-1} (f(d) - f(i)) \left(\alpha_i - (q-1)^i C_m^i \right) \tag{B.6}$$

Or la fonction f est supposée strictement croissante, le terme de gauche de cette équation est donc positif. Par ailleurs, le terme de droite est négatif du fait de la croissance de f et de l'inégalité (B.3). Par conséquent, les deux termes sont nuls, ce qui implique

$$\begin{aligned} i \in \{0, \cdots, d-1\} \quad & \alpha_i = (q-1)^i C_m^i, \\ i \in \{d+1, \cdots, m\} \quad & \alpha_i = 0. \end{aligned}$$

Ces valeurs des coefficients α_i correspondent à des codes parfaits ou quasi-parfaits. Réciproquement, si l'on suppose que le code est parfait ou quasi-parfait, l'égalité (B.6) est vérifiée. Le lemme est ainsi démontré.

□

B.3 Les cellules de Voronoï ne sont pas creuses

Nous nous proposons de démontrer un résultat assez intuitif et que nous n'avons pas trouvé dans la littérature.

Théorème B.3.1 *Soit $\mathcal{C}$ un code linéaire q-aire de longueur n et de rayon de recouvrement ρ (i.e. $\rho = \max_{\underline{u}\in F_q^n} \min_{\underline{v}\in\mathcal{C}} d_H(\underline{u},\underline{v})$). Les poids des chefs de classe de la cellule de Voronoï du mot nul vérifient*

$$\forall i\, 0 \leq i \leq \rho \quad \alpha_i \neq 0.$$

Ce qui peut se reformuler par : il n'y a pas de « trous » dans la distribution des poids de la cellule de Voronoï du mot nul.

Preuve

Soit S_w l'ensemble des mots de poids w et $\mathcal{C}^*$ le code $\mathcal{C}$ privé du mot nul. Par définition de ρ, α_ρ est non nul. Nous allons supposer l'existence d'un α_w nul pour w strictement compris entre 0 et ρ, c'est-à-dire

$$\exists w, \quad 0 < w < \rho \quad \alpha_w = 0 \Longrightarrow \left\{ \begin{array}{llll} \forall \underline{x} \in S_w & \exists \underline{c} \in \mathcal{C}^* & d_H(\underline{c},\underline{x}) \leq d_H(\underline{0},\underline{x}) & \text{(a)} \\ \exists \underline{y} \in S_{w+1} & \forall \underline{c}' \in \mathcal{C}^* & d_H(\underline{y},\underline{c}') \geq d_H(\underline{y},\underline{0}) & \text{(b)} \end{array} \right. \tag{B.7}$$

Nous allons montrer que nous pouvons être plus précis en transformant les inégalités précédentes en égalité.

Premier cas : supposons que l'inégalité (a) soit stricte. Il existe alors un mot $\underline{x}$ de poids w et un mot $\underline{c}$ tels que

$$\begin{array}{rcl} d_H(\underline{y},\underline{x}) & = & 1 \\ d_H(\underline{c},\underline{x}) & < & w = d_H(\underline{x},\underline{0}) \end{array}$$

En appliquant l'inégalité triangulaire, nous obtenons $d_H(\underline{y},\underline{c}) < w+1$, ce qui contredit l'inégalité (b).

Deuxième cas : supposons que l'inégalité (b) soit stricte. Il existe un vecteur $\underline{x}$ de S_w tel que $d_H(\underline{y},\underline{x}) = 1$. L'inégalité (a) nous assure l'existence d'un mot de code $\underline{c}$ tel que $d_H(\underline{c},\underline{x}) \leq d_H(\underline{0},\underline{x}) = w$. Par inégalité triangulaire, on obtient $d_H(\underline{y},\underline{c}) \leq w+1$. Or, de l'inégalité (b), on déduit que $d_H(\underline{y},\underline{c}) > w+1$. D'où contradiction.

Par conséquent, nous pouvons écrire

$$\exists w, \quad 0 < w < \rho \quad \alpha_w = 0 \Longrightarrow \left\{ \begin{array}{llll} \forall \underline{x} \in S_w & \exists \underline{c} \in \mathcal{C}^* & d_H(\underline{c},\underline{x}) = d_H(\underline{0},\underline{x}) \\ \exists \underline{y} \in S_{w+1} & \exists \underline{\hat{c}} \in \mathcal{C}^* & d_H(\underline{y},\underline{\hat{c}}) = d_H(\underline{y},\underline{0}) \\ & \forall \underline{c}' \in \mathcal{C}^* & d_H(\underline{y},\underline{c}') \geq d_H(\underline{y},\underline{0}) \end{array} \right.$$

Définissons $\mathcal{S}(\underline{y}) = \{\underline{c} \in \mathcal{C}^*, \quad d_H(\underline{y},\underline{c}) = d_H(\underline{y},\underline{0})\}$. Notons que $\mathcal{S}(\underline{y})$ n'est pas vide en raison de l'existence de $\hat{c}$. Posons

$$Z = \mathcal{S}(\underline{y}) \bigcap \left(\bigcap_{\substack{\underline{x}\in S_w \\ d_H(\underline{x},\underline{y})=1}} \mathcal{S}(\underline{x}) \right)$$

et supposons cet ensemble non vide : soit $\underline{c}$ un élément de Z. Alors pour chaque composante i non nulle de $\underline{y}$, il existe un mot $\underline{x}$ à distance w de $\underline{c}$ et à distance 1 de $\underline{y}$ tel que $\underline{y}$ et $\underline{x}$ diffèrent en leur i^{e} composante :

$$\underline{y} = \underline{x} + y_i \underline{e}_i$$

où e_i est le i^{e} vecteur canonique (le vecteur dont la seule composante non nulle est la i^{e}qui vaut 1). Supposons $y_i = c_i$. On en déduit $x_i \neq c_i$. Or

$$\begin{aligned} d_H(\underline{y},\underline{c}) &= d_H(\underline{x}+y_i\underline{e}_i,\underline{c}) = d_H^{1,\cdots,i-1}(\underline{x}+y_i\underline{e}_i,\underline{c}) + d_H^{i+1,\cdots,m}(\underline{x}+y_i\underline{e}_i,\underline{c}) + d_H(y_i,c_i) \\ &= d_H^{1,\cdots,i-1}(\underline{x},\underline{c}) + d_H^{i+1,\cdots,m}(\underline{x},\underline{c}) < w \end{aligned}$$

ce qui est contraire aux hypothèses sur $\underline{y}$. Ce raisonnement est valable pour toutes les composantes non nulles de $\underline{y}$. Donc $\underline{y}$ et $\underline{c}$ diffèrent au moins sur le support de $\underline{y}$ or $d_H(\underline{y},\underline{c}) = w+1 = w_H(\underline{y})$ par hypothèse.
On en déduit dans le cas binaire que $\underline{c} = \underline{0}$, ce qui est impossible donc l'ensemble Z est vide. Puisque $\mathcal{S}(\underline{y})$ est non vide, il existe donc un mot de poids w appartenant à $V(\underline{0})$, ce que nous voulions démontrer.
Dans le cas général q-aire, on en déduit que tous les éléments de Z ont un poids inférieur ou égal à $w+1$. Par définition de Z, $\underline{c}$ a au moins une composante non nulle. Notons-la c_{i_0}. Puisque le support de $\underline{y}$ contient celui de $\underline{c}$, y_{i_0} est non-nul. Il existe donc un élément non nul λ de F_q tel que $y_{i_0} = \lambda c_{i_0}$. Le code étant linéaire, $\lambda\underline{c}$ est un élément du code tel que $d_H(\underline{y},\lambda\underline{c}) \leq w$ ce qui contredit les hypothèses sur $\underline{y}$. Par conséquent, Z est vide, ce qui démontre l'existence d'un mot de poids w appartenant à $V(\underline{0})$.

□

Annexe C

Démonstrations de la convergence des bornes inférieures

Avant de démontrer les convergences des bornes inférieures P_{emi}, D_{inf}, P_{esi} et D_{sci}, nous allons tout d'abord étudier la convergence d'un paramètre commun à ces bornes, à savoir la capacité de correction maximale. L'ordre des démonstrations est différent de l'ordre d'apparition des bornes dans le corps du document : nous avons choisi de les présenter dans cette annexe par complexité croissante.

C.1 Convergence de la capacité de correction maximale

Supposons que l'on dispose d'une suite d'entiers (k_n) telle que $\lim_{n\to\infty} k_n/n \log_2 q = R$, R étant fixé et non nul. Par la suite, nous omettrons la dépendance de k_n par rapport à n, le notant simplement k. Introduisons une nouvelle suite d'entiers (d_n) définie par

$$\forall n \in \mathbb{N} \quad d_n = \max\{j \in \mathbb{N} \quad / \quad \sum_{i<j} (q-1)^i C_n^i \leq q^{n-k}\} \tag{C.1}$$

Puisqu'ils dépendent de k et de n, les entiers (d_n) dépendent donc aussi de R. Posons $\delta_n = d_n/n$ et $\delta'_n = (d_n - 1)/n$. Si nous étudions des codes (n, k) alors d_n peut être interprété comme le rayon de recouvrement minimal (sans pour autant pouvoir garantir qu'il existe un code de rayon de recouvrement d_n) et $d_n - 1$ comme la capacité de correction maximale (il ne nous est pas pour autant possible de garantir qu'il existe un code de capacité de correction $d_n - 1$). L'inégalité (C.1) est aussi appelée dans la littérature ([7, 43, 11]) borne de recouvrement sphérique (*sphere covering bound*), borne de Hamming ou borne d'empilement des sphères (*sphere packing bound*). La définition

des entiers d_n par (C.1) se traduit alors par deux inégalités :

$$\begin{aligned}\sum_{i=0}^{n\delta'_n}(q-1)^i C_n^i &\leq q^{n-k}\\ \sum_{i=0}^{n\delta_n}(q-1)^i C_n^i &> q^{n-k}.\end{aligned}$$

Notons que la fonction rendement-distorsion R_{SqS} d'une SqS se confond avec la capacité C_{CqS} d'un canal CqS sans perte de généralité. Nous utiliserons dans cette section la notation C_{CqS}. Remarquons aussi que cette fonction est décroissante et bijective sur l'intervalle $[0, 1-1/q]$: sa réciproque sur cet intervalle est définie et notée $C_{CqS}^{-1}()$ Nous souhaitons démontrer le théorème suivant, précédemment énoncé dans [11] où la preuve consiste en le rappel de la double inégalité du théorème A.2.1.

Théorème C.1.1 *Soit R un réel vérifiant $0 < R \leq \log_2 q$, alors*

$$\lim_{n\to\infty} \delta'_n = C_{CqS}^{-1}(R).$$

Commençons par établir le lemme suivant.

Lemme C.1.1 *Si n est assez grand alors $\delta'_n \leq 1 - 1/q$*

Preuve

L'hypothèse d'un rendement R non nul implique que $\delta'_n < 1$ pour n suffisamment grand. Supposons alors $\delta'_n > 1 - 1/q$. Utilisons par ailleurs les inégalités (C.1) et le théorème A.1.1 pour obtenir

$$\frac{q^n}{\sqrt{2n}} \leq \frac{q^n}{\sqrt{8n\frac{1}{q}(1-\frac{1}{q})}} = \frac{2^{H_2(n(1-1/q))}(q-1)^{n(1-\frac{1}{q})}}{\sqrt{8n\frac{1}{q}(1-\frac{1}{q})}} \leq \sum_{i=0}^{n\delta'_n}(q-1)^i C_n^i \leq 2^{(n-k)\log_2 q} = q^{n-k}.$$

En prenant le logarithme des membres extrêmes de cette triple inégalité et en réarrangeant les termes, nous obtenons

$$\frac{k}{n}\log_2 q \leq \frac{\log_2 2n}{2n}.$$

Puisque R, qui est la limite de $\frac{k}{n}\log_2 q$, est non nul, cette inégalité est fausse à partir d'un certain rang. Par conséquent $\delta'_n \leq 1 - 1/q$, ce que nous voulions démontrer. Nous en déduisons par ailleurs que $\delta_n < 1 - 1/q$.

□

Nous pouvons alors nous concentrer sur le théorème C.1.1. Utilisant successivement l'inégalité (C.2), la définition de δ'_n, l'inégalité sur les sommes de coefficients binomiaux (A.2.1) et le lemme précédent, nous obtenons

$$\begin{aligned}q^{n-k} &< \sum_{i=0}^{n\delta_n}(q-1)^i C_n^i\\ &< \sum_{i=0}^{n\delta'_n}(q-1)^i C_n^i + (q-1)^{n\delta_n} C_n^{n\delta_n}\\ &< (q-1)^{n\delta'_n} 2^{nH_2(\delta'_n)} + (q-1)^{n\delta_n}\tfrac{n-n\delta'_n}{n\delta_n} C_n^{n\delta'_n}.\end{aligned}$$

En réutilisant le théorème A.2.1,

$$q^{n-k} < (q-1)^{n\delta'_n} 2^{nH_2(\delta'_n)} \left(1 + (q-1)\frac{1-\delta'_n}{\delta_n}\right). \tag{C.2}$$

Divisons alors les deux membres de cette inégalité par q^{n-k} pour utiliser (1.12) et prenons-en le logarithme à base 2 et redivisons par n

$$0 < \frac{k}{n}\log_2 q - C_{CqS}(\delta'_n) + \frac{1}{n}\log_2\left(1 + (q-1)\frac{1-\delta'_n}{\delta_n}\right). \tag{C.3}$$

Puisque $\delta_n \geq 1/n$, nous pouvons affirmer

$$\lim_{n\to\infty} \frac{1}{n}\log_2\left(1 + (q-1)\frac{1-\delta'_n}{\delta_n}\right) = 0.$$

Nous allons maintenant exploiter l'inégalité (C.2) en utilisant cette fois-ci la minoration des sommes de coefficients binomiaux indiquée dans le théorème A.2.1.

$$q^{n-k} \geq \sum_{i=0}^{n\delta'_n} (q-1)^i C_n^i > \frac{(q-1)^{n\delta'_n} 2^{nH_2(\delta'_n)}}{\sqrt{8n\delta'_n(1-\delta'_n)}}.$$

Cette dernière inégalité est équivalente à

$$\frac{k}{n}\log_2 q < C_{CqS}(\delta'_n) + \frac{1}{2n}\log_2(8n\delta'_n(1-\delta'_n)). \tag{C.4}$$

Combinons alors les inégalités (C.4) et (C.3)

$$C_{CqS}(\delta'_n) - \frac{1}{n}\log_2\left(1 + (q-1)\frac{1-\delta'_n}{\delta_n}\right) < \frac{k}{n}\log_2 q < C_{CqS}(\delta'_n) + \frac{1}{2n}\log_2(8n\delta'_n(1-\delta'_n)). \tag{C.5}$$

Nous déduisons de cette dernière double inégalité que δ'_n admet une limite δ :

$$\lim_{n\to\infty} \delta'_n = \lim_{n\to\infty} \delta_n = C^{-1}_{CqS}(R) = \delta, \tag{C.6}$$

ce que l'on souhaitait prouver.

C.2 Convergence de P_{emi}

La borne P_{emi} s'appliquant dans le domaine du codage de canal, le rendement R de la section précédente est désormais noté R_c. Nous souhaitons à présent démontrer que cette borne inférieure P_{emi} de la probabilité d'erreur par mot sur CqS converge vers 0 lorsque le rendement R_c est inférieur à la capacité du canal $C_{CqS}(P)$ et vers 1 dans

le cas contraire. Rappelons le lien entre p et P : $P = (q-1)p$. Rappelons d'abord la définition de la borne P_{emi} (théorème 1.1.1) :

$$P_{emi} = 1 - \sum_{i=0}^{d-1} (q-1)^i C_n^i p^i (1-P)^{n-i} - \left(q^{n-k} - \sum_{i=0}^{d-1} (q-1)^i C_n^i \right) p^d (1-P)^{n-d}, \quad \text{(C.7)}$$

n désignant la longueur du code, k sa dimension et d étant le plus grand entier vérifiant :

$$\sum_{i<d} (q-1)^i C_n^i \leq q^{n-k}.$$

Nous avons fait abstraction dans les notations de la dépendance de P_{emi} et de d vis-à-vis de n et R_c afin de les alléger. Si le rendement R_c vaut $\log_2 q$, il est alors possible de montrer directement que P_{emi} converge vers 1 lorsque n tend vers l'infini. Introduisons comme dans le paragraphe C.1 $\delta_n = d/n$ et $\delta'_n = (d-1)/n$. Supposons que nous disposons d'une famille de codes de rendement limite R_c donné vérifiant $0 < R_c < \log_2 q$. La suite (δ_n) admet alors une limite $\delta = C_{CqS}^{-1}(R_c)$ d'après le théorème (C.1.1). Cette limite vérifie $\delta < 1 - 1/q$: il existe donc un n assez grand tel que $\delta_n < 1 - 1/q$. De même, un rendement R_c différent de $\log_2 q$ implique que δ'_n est strictement positif à partir d'un certain rang n et que δ est strictement positif.
Supposons à présent que $\delta'_n \leq P$. En appliquant les théorèmes (A.2.2) et (A.1.1), nous obtenons un encadrement de P_{emi}

$$P_{emi} \geq 1 - P^{n\delta'_n}(1-P)^{n(1-\delta'_n)} 2^{nH_2(\delta'_n)} \left(1 + \frac{P(1-\delta'_n)}{(1-P)\delta_n \sqrt{2\pi n \delta'_n (1-\delta'_n)}} \right) \quad \text{(C.8)}$$

$$P_{emi} \leq 1 - \frac{P^{n\delta'_n}(1-P)^{n(1-\delta'_n)} 2^{nH_2(\delta'_n)}}{\sqrt{8n\delta'_n(1-\delta'_n)}} \left(1 + \frac{P(1-\delta'_n)}{(1-P)\delta_n} \right) \quad \text{(C.9)}$$

De l'inégalité $1 - 1/q > \delta > 0$, nous déduisons

$$\lim_{n\to\infty} \frac{1}{\sqrt{8n\delta'_n(1-\delta'_n)}} = 0 \quad \text{(C.10)}$$

$$\lim_{n\to\infty} \frac{P(1-\delta'_n)}{(1-P)\delta_n \sqrt{2\pi n \delta'_n (1-\delta'_n)}} = 0. \quad \text{(C.11)}$$

Concentrons-nous maintenant sur le terme commun à la majoration et à la minoration de P_{emi} (double inégalité (C.9)) en le récrivant

$$P^{n\delta'_n}(1-P)^{n(1-\delta'_n)} 2^{nH_2(\delta'_n)} = 2^{n(H_2(\delta'_n) + \delta'_n \log_2 P + (1-\delta'_n)\log_2(1-P))}.$$

Or l'inégalité de Gibbs affirme que $H_2(\delta'_n) + \delta'_n \log_2 P + (1-\delta'_n)\log_2(1-P) \leq 0$ quelle que soit la valeur de P et que l'inégalité ne devient égalité que si $\delta'_n = P$ (l'inégalité de Gibbs est dans ce cas équivalente à la propriété de concavité de la fonction H_2). Par conséquent $\delta'_n < P$ implique $\lim_{n\to\infty} P_{emi} = 1$. La décroissance de la capacité en fonction de P et le théorème C.1.1 nous permettent alors d'affirmer

$$C_{CqS}(P) < R_c \quad \Longrightarrow \quad \lim_{n\to\infty} P_{emi} = 1 \quad \text{(C.12)}$$

Nous avons donc prouvé la réciproque forte du théorème de codage de canal dans le cas du CqS.

Supposons à présent que $\delta'_n > P$. Nous souhaitons montrer que P_{emi} converge vers 0.

$$\begin{array}{rcl} P_{emi} & \leq & 1 - \sum_{i=0}^{n\delta'_n} C_n^i P^i (1-P)^{n-i} \\ & \leq & \sum_{i=n\delta_n}^{n} C_n^i P^i (1-P)^{n-i} \\ & \leq & \sum_{i=0}^{n(1-\delta_n)} C_n^i P^{n-i} (1-P)^i. \end{array} \tag{C.13}$$

Puisque nous supposons que l'inégalité $\delta'_n > P$ est vraie, alors $1-\delta_n < 1-P$ et l'application du théorème A.2.2 nous donne

$$\sum_{i=0}^{n(1-\delta_n)} C_n^i P^{n-i} (1-P)^i \leq P^{n\delta_n} (1-P)^{n(1-\delta_n)} 2^{nH_2(1-\delta_n)}$$

En utilisant les mêmes arguments que dans le cas $\delta'_n < P$, nous montrons que si $\delta'_n > P$ alors $\lim_{n\to\infty} P_{emi} = 0$, ce qui se reformule en

$$C_{CqS}(P) > R_c \quad \Longrightarrow \quad \lim_{n\to\infty} P_{emi} = 0, \tag{C.14}$$

concluant ainsi notre démonstration.

P_{emi} est donc asymptotiquement une bonne borne inférieure de la probabilité d'erreur par mot.

C.3 Convergence de D_{inf}

La borne D_{inf} s'appliquant dans le domaine du codage de source, le rendement R de la première section est désormais noté R_s. Rappelons l'expression de la borne de la distorsion créée par la compression d'une SqS par un code en bloc de longueur m et de dimension k (théorème 1.2.1) :

$$D_{inf} = \frac{d}{m} - \frac{1}{mq^{m-k}} \sum_{i<d} (d-i)(q-1)^i C_m^i \tag{C.15}$$

d est analogue à la suite (d_m) introduite au paragraphe C.1. Définissons alors δ_m par $m\delta_m = d$ et $\delta'_m = \delta_m - 1/m$. Nous voulons prouver que, R_s étant non-nul, on a alors

$$\lim_{m\to\infty} \frac{k}{m} \log_2 q = R_s \quad \Longrightarrow \quad \lim_{m\to\infty} D_{inf} = R_{SqS}^{-1}(R_s). \tag{C.16}$$

En d'autres termes, D_{inf} est une borne asymptotiquement bonne de la distorsion.

Nous supposons donc comme dans la section C.1 disposer d'une suite d'entiers (k_m) telle que

$$\lim_{m\to\infty} \frac{k_m}{m} \log_2 q = R_s$$

avec R_s non nul. De nouveau, nous omettrons l'indice des entiers k_m, notés à présent k, afin d'alléger les notations. Nous allons commencer par énoncer et prouver un lemme.

Lemme C.3.1

$$\lim_{m\to\infty} \frac{1}{q^{m-k}} \sum_{i=0}^{m\delta'_m} (\delta_m - i/m)(q-1)^i C_m^i = 0.$$

Preuve

Posons

$$\Delta = \sum_{i=0}^{m\delta'_m} (\delta_m - i/m)(q-1)^i C_m^i.$$

Récrivons Δ en substituant $m\delta'_m - i$ à i

$$\Delta = \frac{(q-1)^{m\delta'_m}}{m} C_m^{m\delta'_m} \left(1 + \sum_{i=1}^{m\delta'_m} (q-1)^{-i} (i+1) \frac{(m\delta'_m)\cdots(m\delta'_m - i + 1)}{(m - m\delta'_m + 1)\cdots(m - m\delta'_m + i)} \right)$$

Introduisons $r_m = m\delta'_m/((q-1)(m - m\delta'_m + 1))$ afin d'exprimer plus facilement une majoration de Δ :

$$\Delta \leq (q-1)^{m\delta'_m} C_m^{m\delta'_m} \sum_{i=0}^{m\delta_m} \frac{i+1}{m} r_m^i \tag{C.17}$$

Nous reconnaissons dans le membre de droite de (C.17) les premiers termes d'une série convergente car $r_m < 1$ si m est assez grand (par application du lemme (C.1.1)). De plus, puisque δ'_m admet une limite δ, r_m admet lui aussi une limite r : $r = \delta/((q-1)(1-\delta))$. Or, par hypothèse, le rendement R_s est non nul, donc r est strictement plus petit que 1 Par conséquent, nous pouvons majorer Δ

$$\Delta \leq (q-1)^{m\delta'_m} C_m^{m\delta'_m} \frac{1}{m(1-r_m)^2}$$

Le théorème C.1.1 et la non-nullité de R_s nous permettent donc d'affirmer

$$\begin{aligned} \lim_{m\to\infty} \frac{1}{(1-r_m)^2} &= \frac{1}{(1-r)^2} < \infty \\ \frac{(q-1)^{m\delta'_m} C_m^{m\delta'_m}}{q^{m-k}} &\leq 1 \end{aligned}$$

d'où le résultat énoncé dans le lemme (C.3.1).

$$\lim_{m\to\infty} \frac{\Delta}{q^{m-k}} = 0. \tag{C.18}$$

□

PREUVE DU THÉORÈME 1.2.2 Nous allons d'abord prouver la convergence de la borne inférieure D_{inf} grâce au lemme précédent et aux résultats de la section C.1. Puis nous prouverons que cette borne est supérieure aux meilleures performances possibles théoriquement (i.e. $D_{inf} > R_{SqS}^{-1}(R_s)$).
Récrivons D_{inf} à partir de sa définition (équation (C.15)) et de Δ :

$$D_{inf} = \delta_m - \frac{\Delta}{q^{m-k}}$$

Les formules (C.6) et (C.18) prouvent que la borne inférieure converge vers la fonction rendement-distorsion si R_s est non nul. Si R_s est nul, il est aisé de montrer directement que la borne inférieure converge vers $1 - 1/q$, ce que nous ne ferons pas ici afin de ne pas compliquer inutilement le raisonnement.
Il nous reste donc à montrer que $D_{inf} > R_{SqS}^{-1}(R_s)$. Remarquons que D_{inf} est la distorsion correspondant à un code parfait ou quasi-parfait de longueur m, de dimension k, de capacité de correction $d-1$ sans garantie que celui-ci existe réellement. Notons ce code $\mathcal{C}$. Considérons le code $\mathcal{C}'$ de longueur et dimension doubles par rapport à celles de $\mathcal{C}$. Choisissons le code $\mathcal{C}'$ comme la concaténation de $\mathcal{C}$ avec lui-même (i.e. $\mathcal{C}' = \mathcal{C} \times \mathcal{C}$). La capacité de correction maximale pour un tel rendement et une longueur $2m$ est notée $d'-1$ et la borne inférieure de la distorsion D'_{inf}. En élevant au carré les deux membres de (C.1), on montre que $d' \geq d$. La distorsion de $\mathcal{C}'$ est égale à la distorsion de $\mathcal{C}$ (par définition de la distorsion (1.29)). Nous observons que

- le nombre α'_i de chefs de classe de poids i de $\mathcal{C}'$ est le i^{e} élement de la convolution des $(\alpha_i)_{0 \leq i \leq m-k}$ avec eux-mêmes, α_i étant le nombre de chefs de classe de poids i de $\mathcal{C}$;
- la borne inférieure D_{inf} pour la longueur m et le rendement R_s est obtenue en substituant à α_i le coefficient $\beta_i = (q-1)^i C_m^i$ lorsque $i < d$, $\beta_d = q^{m-k} - \sum_{i=0}^{d-1}(q-1)^i C_m^i$ lorsque $i = d$ et 0 sinon (β_i est le nombre de mots de poids i et de longueur m).

Une minoration de la distorsion du code concaténé $\mathcal{C}'$ est donc obtenue en calculant $D' = (\sum_i i\beta'_i)/(2mq^{2m-2k})$ avec $\beta'_i = \sum_{j=\max(0,i-d)}^{\min(i,d)} \beta_j \beta_{i-j}$. Tous calculs faits, nous trouvons que $D' = D_{inf}$, c'est-à-dire que la borne inférieure de la distorsion du code concaténé $\mathcal{C}'$ calculée par convolution est identique à la borne inférieure de la distorsion du code $\mathcal{C}$. Puisque $d' \geq d$, l'interprétation de la borne inférieure donnée pour les codes linéaires s'applique : $D'_{inf} \leq D' = D_{inf}$. Puisque D_{inf} converge vers la fonction rendement-distorsion à rendement R_s fixé et que nous pouvons extraire une sous-suite décroissante de bornes inférieures nous avons prouvé :

$$D_{inf} \geq R_{SqS}^{-1}(R_s)$$

Le théorème 1.2.2 est donc prouvé.

C.4 Convergence de P_{esi}

Rappelons tout d'abord l'expression de la borne inférieure de la probabilité d'erreur par symbole P_{esi}

$$P_{esi} = \frac{1}{k}\left(\sum_{a=1}^{k} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} (q-1)^w C_n^w p(w) + \sum_{a=0}^{k}\left(\sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w - s_a q^{n-k}\right) p(w_{s_a}+1)\right)$$

où la suite $(s_i)_{i\in\mathbb{N}}$ est définie par

$$s_i = \sum_{w=0}^{i} (q-1)^w C_k^w \tag{C.19}$$

et la suite $(w_i)_{i=1,\cdots,q^k}$ par

$$w_i = \arg\max\{j \ / \ \sum_{w=0}^{j} (q-1)^w C_n^w \leq i q^{n-k}\} \tag{C.20}$$

et $p(w) = p^w(1-P)^{n-w}$.

Nous voulons démontrer les théorèmes 1.1.4 et 1.1.5 qui donnent le comportement asymptotique de P_{esi} en fonction du rendement de canal R_c et de la capacité du canal q-aire symétrique $C_{CqS}(P)$:

$$\begin{array}{cc} R_c < C_{CqS}(P) & \lim_{n\to\infty} P_{esi} = 0 \text{ (théorème 1.1.4)} \\ C_{CqS}(P) < R_c < \log_2 q & \lim_{n\to\infty} P_{esi} = C_{CqS}^{-1}\left(\frac{\log_2 q}{R_c} C_{CqS}(P)\right) \text{ (théorème 1.1.5).} \end{array}$$

Preuve
Commençons par démontrer le théorème 1.1.4.
Décomposons la borne inférieure en deux termes :

$$\begin{aligned} P_1 &= \frac{1}{k}\sum_{a=0}^{k}\left(\sum_{w=0}^{w_{s_a}} (q-1)^w C_n^w - s_a q^{n-k},\right) p(w_{s_a}+1) \\ P_2 &= \frac{1}{k}\sum_{a=1}^{k} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} (q-1)^w C_n^w p(w). \end{aligned}$$

Par définition des w_{s_a}, P_1 est une somme de $k+1$ termes négatifs ou nuls dont le dernier terme est nul ($w_{s_k} = n$) et peut être majorée en valeur absolue par

$$|P_1| \leq \frac{1}{k}\sum_{a=0}^{k-1} C_n^{w_{s_a}+1}(q-1)^{w_{s_a}+1} p(w_{s_a}+1).$$

Chacun des ces $k-1$ termes est inférieur à $C_n^{\lfloor (n+1)P \rfloor} P^{\lfloor (n+1)P \rfloor}(1-P)^{n-\lfloor (n+1)P \rfloor}$. En utilisant le théorème A.1.1, il vient

$$\begin{aligned} |P_1| &\leq \frac{2^{nH_2(\frac{\lfloor (n+1)P \rfloor}{n})} P^{\lfloor (n+1)P \rfloor}(1-P)^{n-\lfloor (n+1)P \rfloor}}{\sqrt{2\pi \lfloor (n+1)P \rfloor (1-\frac{\lfloor (n+1)P \rfloor}{n})}} \\ &\leq \frac{2^{n(H_2(\frac{\lfloor (n+1)P \rfloor}{n}) + \frac{\lfloor (n+1)P \rfloor}{n} \log_2 P + (1-\frac{\lfloor (n+1)P \rfloor}{n}) \log_2 (1-P))}}{\sqrt{2\pi \lfloor (n+1)P \rfloor (1-\frac{\lfloor (n+1)P \rfloor}{n})}}. \end{aligned}$$

On reconnaît dans l'exposant la différence entre la fonction H_2 et sa tangente T_P définie par $T_P : x \mapsto -x \log_2 P - (1-x)\log_2(1-P)$. Cette différence est donc de l'ordre de $O((\frac{\lfloor (n+1)P \rfloor}{n} - P)^2)$, c'est-à-dire de l'ordre de $O(n^{-2})$. Par conséquent P_1 tend vers 0 lorsque n tend vers l'infini.
Contrairement à P_1, nous aurons besoin de l'hypothèse sur le rendement pour montrer la convergence de P_2. Ce dernier peut être simplement majoré par

$$P_2 \leq \sum_{w=w_{s_0}+1}^{n} C_n^w P^w (1-P)^{n-w}.$$

Si X désigne un vecteur de n variables binomiales indépendantes et de paramètre P alors son poids moyen est nP et la variance du poids $nP(1-P)$. Il vient en appliquant l'inégalité de Bienaymé-Tchebycheff au poids de X :

$$\begin{aligned} P_2 &\leq \Pr(w_H(X) \geq w_{s_0}+1) \\ &\leq \Pr(|w_H(X) - nP| \geq w_{s_0} + 1 - nP) \\ &\leq \frac{nP(1-P)}{(w_{s_0}+1-nP)^2}. \end{aligned}$$

Il découle de la dernière inégalité que le comportement de P_2 dépend de w_{s_0}. Or ce dernier est égal au d introduit dans l'annexe C.1. Puisque $R_c < C_{CqS}(P)$, on en déduit que $(w_{s_0}+1-nP)^2$ se comporte comme n^2 à l'infini donc que P_2 converge vers 0.
Ceci conclut la démonstration du théorème 1.1.4.
Démontrons à présent le théorème 1.1.5. La démonstration précédente nous a permis de voir que le terme prépondérant dans le comportement asymptotique de P_{esi} était le terme P_2 défini par

$$P_2 = \frac{1}{k} \sum_{a=1}^{k} a \sum_{w=w_{s_{a-1}}+1}^{w_{s_a}} (q-1)^w C_n^w p(w)$$

puique le terme P_1 tend vers 0 lorsque la longueur n tend vers l'infini. Il nous suffit donc de démontrer

$$\lim_{n\to\infty} P_2 = C_{CqS}^{-1}\left(\frac{\log_2 q}{R_c} C_{CqS}(P)\right).$$

L'idée de la démonstration consiste à interpréter P_2 comme une moyenne de probabilités portant sur le vecteur X formé de n variables aléatoires binomiales indépendantes et de paramètre p. Le poids de Hamming de X, $w_H(X)$, est en moyenne nP et de variance

$nP(1-P)$. Par conséquent, lorsque ce poids sera trop éloigné de nP, leur contribution à P_2 sera négligeable et la limite découlera de la contribution des poids moyens. Formalisons maintenant cette idée. Soit ϵ un réel strictement positif tel que $P+\epsilon < 1-1/q$. Introduisons b et c deux entiers tels que

$$b = \max\{a,\ 0 \leq a \leq k,\ w_{s_a} < n(P-\epsilon)\} \tag{C.21}$$
$$c = \min\{a,\ 0 \leq a \leq k,\ w_{s_{a-1}} + 1 > n(P+\epsilon)\}. \tag{C.22}$$

Il vient alors en appliquant le théroème A.2.2 et en utilisant les propriétés de convexité de la fonction H_2

$$\begin{aligned}
\frac{1}{k}\sum_{a=1}^{b} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} C_n^w P^w (1-P)^{n-w} &\leq \sum_{w=0}^{w_{s_b}} C_n^w P^w (1-P)^{n-w} \\
&\leq 2^{n(H_2(w_{s_b}/n) - T_P(w_{s_b}/n))} \\
&\leq \exp\left(-\frac{n}{2}\left(\frac{w_{s_b}}{n} - P\right)^2\right) \\
&\leq \exp\left(-\frac{n}{2}\epsilon^2\right).
\end{aligned} \tag{C.23}$$

De même, nous obtenons pour les poids élevés

$$\frac{1}{k}\sum_{a=c}^{k} a \sum_{w=w_{s_{a-1}+1}}^{w_{s_a}} C_n^w P^w (1-P)^{n-w} \leq \exp\left(-\frac{n}{2}\epsilon^2\right). \tag{C.24}$$

Le terme P_2 est donc majoré par

$$\begin{aligned}
P_2 &\leq 2\exp\left(-\frac{n}{2}\epsilon^2\right) + \frac{c}{k}\sum_{w=w_{s_b}+1}^{w_{s_c}} C_n^w P^w (1-P)^{n-w} \\
&\leq 2\exp\left(-\frac{n}{2}\epsilon^2\right) + \frac{c}{k}.
\end{aligned} \tag{C.25}$$

La limite de $\frac{c}{k}$ est obtenue à partir de la définition de c (C.22) :

$$w_{s_{c-1}}+1 > n(P+\epsilon) \geq w_{s_{c-2}}+1 \Longrightarrow C_{CqS}\left(\frac{w_{s_{c-1}}+1}{n}\right) \leq C_{CqS}(P+\epsilon) \leq C_{CqS}\left(\frac{w_{s_{c-2}}+1}{n}\right).$$

L'application de l'inégalité (A.3) à la définition des w_{s_a} (C.20) permet d'obtenir, après quelques manipulations,

$$\frac{R}{\log_2 q} C_{CqS}\left(\frac{c}{k}\right) + o(n^{-1}) \leq C_{CqS}(P+\epsilon) \leq \frac{R}{\log_2 q} C_{CqS}\left(\frac{c}{k}\right) + o(n^{-1}). \tag{C.26}$$

Minorons maintenant P_2

$$P_2 \geq \frac{b+1}{k}\sum_{w=w_{s_b}+1}^{w_{s_c}} C_n^w P^w (1-P)^{n-w} \geq \frac{b+1}{k}\sum_{w=n(P-\epsilon)}^{n(P+\epsilon)} C_n^w P^w (1-P)^{n-w}. \tag{C.27}$$

L'inégalité de Bienaymé-Tchebycheff permet de simplifier la minoration de P_2

$$P_2 \geq \frac{b+1}{k}\left(1 - \frac{P(1-P)}{n\epsilon^2}\right). \tag{C.28}$$

Comme pour c, nous obtenons pour b :

$$\frac{R}{\log_2 q} C_{CqS}\left(\frac{b+1}{k}\right) + o(n^{-1}) \leq C_{CqS}(P-\epsilon) \leq \frac{R}{\log_2 q} C_{CqS}\left(\frac{b+1}{k}\right) + o(n^{-1}). \tag{C.29}$$

En choisissant convenablement ϵ ($\epsilon = O(n^{-1/4})$ par exemple), on déduit des inégalités (C.25),(C.26), (C.28) et (C.29) que

$$\lim_{n\to\infty} P_2 = C_{CqS}^{-1}\left(\frac{\log_2 q}{R_c} C_{CqS}(P)\right).$$

La convergence de la borne P_{esi} est donc totalement prouvée.

□

C.5 Convergence de D_{sci}

Nous voulons démontrer que la borne D_{sci}, introduite au paragraphe 2.4.2 et définie par

$$D_{sci} = \frac{1}{m2^{m-n}}\left(\sum_{a=0}^{n} p^a(1-p)^{n-a}\left[\sum_{l=w_{s_{a-1}}+1}^{w_{s_a}} lC_m^l + (1+w_{s_a})\left(s_a 2^{m-n} - \sum_{l=0}^{w_{s_a}} C_m^l\right)\frac{1-2p}{1-p}\right]\right) \tag{C.30}$$

(où m, n et p désignent respectivement la longueur du code linéaire, sa dimension et le paramètre du CBS et où w_i et s_a sont les suites définies dans la section précédente), converge vers les meilleures performances théoriquement possibles, i.e. :

$$\lim_{m\to\infty} \frac{n}{m} = R \Longrightarrow \lim_{m\to\infty} D_{sci} = H_2^{-1}(1 - R(1 - H_2(p))). \tag{C.31}$$

Nous allons encadrer D_{sci} pour montrer sa convergence. Nous allons tout d'abord majorer D_{sci} et montrer la convergence de la borne supérieure de D_{sci} obtenue. Commençons par récrire la borne D_{sci} en utilisant l'égalité $(1-2p)/(1-p) = 1 - p/(1-p)$ et en regroupant les termes selon les polynômes $p^a(1-p)^{n-a}$:

$$\begin{aligned} D_{sci} \;=\; & \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a(1-p)^{n-a}\left\{\sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} lC_m^l + (1+w_{s_a})\left(s_a 2^{m-n} - \sum_{l=0}^{w_{s_a}} C_m^l\right) + \right. \\ & \left. (1+w_{s_{a-1}})\left(\sum_{l=0}^{w_{s_{a-1}}+1} C_m^l - s_{a-1}2^{m-n}\right)\right\}. \end{aligned} \tag{C.32}$$

Les définitions des entiers w_i et s_a (équations (C.20) et (C.19)) montrent que le facteur de $(1+w_{s_{a-1}})$ et celui de $(1+w_{s_a})$ sont positifs. La suite (w_i) étant croissante, D_{sci} est donc majorée par

$$D_{sci} \leq \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \left\{ \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} l C_m^l + (1+w_{s_a}) \left(C_n^a 2^{m-n} - \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} C_m^l \right) \right\}. \tag{C.33}$$

Cette dernière inégalité peut être récrite en regroupant les coefficients binomiaux :

$$\begin{aligned} D_{sci} &\leq \frac{1}{m} \sum_{a=0}^{n} p^a (1-p)^{n-a} C_n^a (1+w_{s_a}) \\ &\quad + \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l. \end{aligned} \tag{C.34}$$

Une minoration de D_{sci} est obtenue en récrivant l'égalité (C.32) :

$$\begin{aligned} D_{sci} &= \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \left\{ \sum_{l=w_{s_{a-1}}+1}^{w_{s_a}} l C_m^l + (1+w_{s_a}) \left(s_a 2^{m-n} - \sum_{l=0}^{w_{s_a}} C_m^l \right) + \right. \\ &\quad \left. (1+w_{s_{a-1}}) \left(\sum_{l=0}^{w_{s_{a-1}}} C_m^l - s_{a-1} 2^{m-n} \right) \right\}. \end{aligned} \tag{C.35}$$

En utilisant, d'une part, la définition de la suite (w_i) et, d'autre part, sa croissance, nous obtenons :

$$\begin{aligned} D_{sci} &\geq \frac{1}{m} \sum_{a=0}^{n} p^a (1-p)^{n-a} C_n^a (1+w_{s_a}) \\ &\quad + \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+1}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l. \end{aligned} \tag{C.36}$$

Les inégalités (C.36) et (C.34), qui donnent un encadrement D_{sci}, ont en commun le premier terme, noté D_1, les seconds termes étant assez proches. Étudions la convergence de D_1 :

$$D_1 = \frac{1}{m} \sum_{a=0}^{n} p^a (1-p)^{n-a} C_n^a (1+w_{s_a}) \tag{C.37}$$

Puisque l'entier w_{s_a} est majoré par m, seuls les termes pour lesquels a est proche de np contribueront de manière significative à D_1 : en reprenant des arguments similaires à ceux donnés pour montrer la convergence de P_2 (le terme prédominant de P_{esi}) dans la section précédente, nous concluons que :

$$\lim_{m\to\infty} D_1 = \lim_{m\to\infty} \frac{w_{s_{\lfloor np \rfloor}}}{m} = H_2^{-1}(1 - R(1 - H_2(p))). \tag{C.38}$$

Il reste alors à montrer que

$$\lim_{m\to\infty} \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l = 0 \qquad \text{(C.39)}$$

$$\lim_{m\to\infty} \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+1}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l = 0. \qquad \text{(C.40)}$$

Ces deux termes négatifs ne diffèrent que par la présence ou l'absence du coefficient binomial $C_m^{1+w_{s_{a-1}}}$. Nous nous contenterons donc de donner la démonstration de (C.39), la preuve de (C.40) reposant sur les mêmes arguments. Introduisons, par commodité, D_2 :

$$D_2 = \frac{1}{m2^{m-n}} \sum_{a=0}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l \qquad \text{(C.41)}$$

Soient ϵ un réel positif et b un entier naturel tels que $n(p+\epsilon) < b-1 < b < n/2$. Décomposons D_2 en une somme de deux termes D_2' et D_2'' :

$$D_2' = \frac{1}{m2^{m-n}} \sum_{a=0}^{b} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l \qquad \text{(C.42)}$$

$$D_2'' = \frac{1}{m2^{m-n}} \sum_{a=b+1}^{n} p^a (1-p)^{n-a} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (l - w_{s_a} - 1) C_m^l. \qquad \text{(C.43)}$$

La preuve de la convergence vers 0 de D_2' est identique à la convergence de Δ de la section C.3. La somme des coefficients binomiaux est majorée par une série convergente (car $a \leq b$) :

$$\begin{aligned} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (w_{s_a} + 1 - l) C_m^l &= C_m^{w_{s_a}} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (w_{s_a} + 1 - l) \prod_{j=0}^{w_{s_a}-l-1} \frac{w_{s_a} - j}{m - w_{s_a} + j} \\ &\leq C_m^{w_{s_a}} \sum_{l=0}^{w_{s_a}-w_{s_{a-1}}-2} (l+1) \left(\frac{w_{s_a}}{m - w_{s_a} + 1} \right)^l \end{aligned} \qquad \text{(C.44)}$$

où la dernière inégalité est obtenue en substituant $w_{s_a} - l$ à l. La définition de la suite (w_i) implique que l'inégalité

$$\frac{C_m^{w_{s_a}}}{2^{m-n}} \leq C_n^a \qquad \text{(C.45)}$$

est vraie. Par conséquent, le facteur $1/m$ assure la convergence de D_2' vers zéro.

La convergence de D_2'' exploite bien sûr le fait que $a > b$. La définition des entiers w_i implique que

$$\frac{1}{2^{m-n}} \sum_{l=w_{s_{a-1}}+2}^{w_{s_a}} (w_{s_a} + 1 - l) C_m^l \leq C_n^a + \frac{C_m^{1+w_{s_a}}}{2^{m-n}}. \qquad \text{(C.46)}$$

De cette inégalité, on tire une majoration de D_2'' (C.43) :

$$\begin{aligned} |D_2''| \;\leq\; & \frac{1}{m}\sum_{a=b+1}^{n} p^a(1-p)^{n-a}(w_{s_a}-w_{s_{a-1}}-1)C_n^a \\ & +\frac{1}{m2^{m-n}}\sum_{a=b+1}^{n} p^a(1-p)^{n-a}(w_{s_a}-w_{s_{a-1}}-1)C_m^{1+w_{s_a}}. \end{aligned} \tag{C.47}$$

Le terme $(w_{s_a}-w_{s_{a-1}}-1)/m$ est borné (en raison de la définition des w_i) et l'inégalité $b > n(p+\epsilon)$ implique que la somme $\sum_{a=b+1}^{n} p^a(1-p)^{n-a}C_n^a$ converge vers 0. Le second terme intervenant dans la majoration de $|D_2''|$ converge également vers 0 pour des raisons identiques (en utilisant également l'inégalité (C.45)), ce qui achève la preuve de la convergence.

Annexe D

Codes linéaires parfaits ou quasi-parfaits

Nous avons vu au premier chapitre que les performances des codes parfaits ou quasi-parfaits, tant en codage de canal qu'en codage de source, sont faciles à déterminer : il est donc souhaitable de connaître les paramètres de ces codes. Nous rappelons dans cette annexe les codes linéaires parfaits ou quasi-parfaits les plus connus en nous inspirant de [43]. Dans cette annexe, m désigne un entier naturel strictement supérieur à 1 (et non une longueur d'un code).

D.1 Codes linéaires parfaits

Les différents codes (ou familles de codes) parfaits sont données dans le tableau D.1. Nous rappelons brièvement quelques-unes de leurs caractéristiques.

Type de code	Taille de l'alphabet	Longueur n	Dimension k	d_{min}
codes de Hamming	q	$\frac{q^m-1}{q-1}$	$\frac{q^m-1}{q-1}-m$	3
code de Golay	2	23	12	7
code de Golay	3	11	6	5
codes à répétition	2	$2m+1$	1	$2m+1$

TAB. D.1 – Table des codes parfaits linéaires.

D.1.1 Les codes de Hamming

Les codes de Hamming sont des codes cycliques connus pour ne corriger qu'une seule erreur. Ils constituent une famille de codes dont le rendement tend vers $\log_2 q$. Les colonnes de leurs matrices de parité contiennent exactement tous les vecteurs non nuls de F_q^m non homothétiques entre eux. Leurs performances sont :

$$P_{em} = 1-(1-P)^n-(q-1)np(1-P)^{n-1} \quad \text{(D.1)}$$

$$D = \frac{q-1}{q^m+1}. \quad \text{(D.2)}$$

D.1.2 Les codes de Golay

Ces codes cycliques font partie de la famille des codes à résidus quadratiques. Le code de Golay binaire compresse une SBS avec une distorsion $D = 0.1240$. La probabilité d'erreur par mot de ce code sur CBS est

$$P_{em} = 1-(1-p)^{23}-23p(1-p)^{22}-253p^2(1-p)^{21}-1771p^3(1-p)^{20}. \quad \text{(D.3)}$$

Le code de Golay ternaire compresse une souce ternaire symétrique avec une distorsion $D = 0.1728$. La probabilité d'erreur par mot de ce code sur canal ternaire symétrique est

$$P_{em} = 1-(1-2p)^{11}-22p(1-2p)^{10}-220p^2(1-2p)^9. \quad \text{(D.4)}$$

D.1.3 Les codes à répétition

Ces codes se contentent de répéter le symbole d'information à émettre. Cette famille de code a un rendement qui tends vers 0. Leurs performances sont :

$$P_{em} = P_{es} = 1-\sum_{i=0}^{m} C_{2m+1}^i p^i (1-p)^{2m+1-i} = \sum_{i=m+1}^{2m+1} C_{2m+1}^i p^i (1-p)^{2m+1-i} \quad \text{(D.5)}$$

$$D = \frac{1}{2}\left(1-\frac{C_{2m+1}^m}{2^{2m}}\right). \quad \text{(D.6)}$$

D.2 Codes linéaires quasi-parfaits

Le tableau D.2 dresse une liste non exhaustive des codes quasi-parfaits linéaires (on notera qu'il existe également des codes non-linéaires quasi-parfaits tels les codes de Preparata poinçonnés [43]).

Type de code	Taille de l'alphabet	Longueur n	Dimension k	d_{min}	Capacité de correction
codes BCH primitifs	2	2^m-1	2^m-1-2m	5	2
codes à répétition	2	$2m+2$	1	$2m+2$	m
codes de Hamming raccourcis	2	2^m-2	2^m-m-2	3	1
code de Wagner	2	23	14	5	2
code de Golay binaire étendu	2	24	12	8	3
code de Golay ternaire étendu	3	12	6	6	2
codes de Hamming étendus	q	$\frac{q^m-1}{q-1}+1$	$\frac{q^m-1}{q-1}-m$	4	1

TAB. D.2 – Table des codes quasi-parfaits linéaires.

D.2.1 Codes BCH binaires primitifs corrigeant deux erreurs

Cette famille des codes BCH binaires primitifs a un rendement qui tend vers 1. Leurs performances sont données par :

$$\begin{aligned} P_{em} &= 1-(1-p)^{2^m-1}-(2^m-1)p(1-p)^{2^m-2} \\ &\quad -(2^m-1)(2^{m-1}-1)p^2(1-p)^{2^m-3}-(2^{2m-1}+2^{m-1}-1)p^3(1-p)^{2^m-4} \quad \text{(D.7)} \\ D &= \frac{1}{2^m-1}(\frac{5}{2}-2^{-m}-2^{1-2m}). \quad \text{(D.8)} \end{aligned}$$

D.2.2 Codes à répétition

Les codes à répétition de longueur paire suivent le même principe ques les codes à répétition de longueur impaire. Ils ont les performances suivantes :

$$\begin{aligned} P_{em}=P_{es} &= 1-\sum_{i=0}^{m-1} C_{2m}^i p^i(1-p)^{2m-i}-\frac{1}{2}C_{2m}^m p^m(1-p)^m \quad \text{(D.9)} \\ &= \frac{1}{2}C_{2m}^m p^m(1-p)^m+\sum_{i=m}^{2m} C_{2m}^i p^i(1-p)^{2m-i} \quad \text{(D.10)} \\ D &= \frac{1}{2}\left(1-\frac{C_{2m}^{m-1}}{2^{2m}}\right). \quad \text{(D.11)} \end{aligned}$$

D.2.3 Code de Wagner

Ce code comprime une SBS avec une distorsion $D=0.1041$. En tant que code de canal, ses performances sont

$$P_{em}=1-(1-p)^{23}-23p(1-p)^{22}-253p^2(1-p)^{21}-235p^3(1-p)^{20}. \quad \text{(D.12)}$$

D.2.4 Codes de Golay étendus

Le code de Golay binaire étendu rajoute un bit de parité au code de Golay binaire. Il comprime une source binaire avec une distorsion $D = 0.1397$. Ses performances sur le CBS s'expriment de la manière suivante :

$$P_{em} = 1-(1-p)^{24} - 24p(1-p)^{23} - 276p^2(1-p)^{22} - 2024p^3(1-p)^{21} - 1771p^4(1-p)^{20}. \tag{D.13}$$

Le code ternaire de Golay étendu est construit à partir de la matrice de parité du code de Golay ternaire : une colonne nulle est ajoutée à cette matrice puis une ligne de 1. Sa distorsion est $D = 0.2140$ et ses performances sur canal ternaire symétrique sont données par :

$$P_{em} = 1-(1-2p)^{12} - 24p(1-2p)^{11} - 264p^2(1-2p)^{10} - 440p^3(1-2p)^{9}. \tag{D.14}$$

D.2.5 Codes de Hamming raccourcis ou étendus

Les codes de Hamming binaires raccourcis sont obtenus à partir des codes de Hamming en ne gardant que les mots de code dont la première composante est nulle puis en supprimant cette composante. Les performances en codage de canal et en codage de source des codes de Hamming raccourcis sont :

$$P_{em} = 1-(1-p)^{2^m-2} - (2^m-2)p(1-p)^{2^m-3} - p^2(1-p)^{2^m-4} \tag{D.15}$$

$$D = \frac{1}{2^m-2}. \tag{D.16}$$

Les codes de Hamming étendus sont obtenus en rajoutant un bit de parité aux mots du code de Hamming initial. Leurs performances s'écrivent

$$\begin{aligned} P_{em} &= 1-(1-P)^{\sum_{i=0}^{m} q^i+1} - \frac{q^m+q-2}{q-1}p(1-P)^{\sum_{i=0}^{m} q^i} \\ &\quad -(q^{m+1}-2-\sum_{i=0}^{m} q^i)p^2(1-P)^{\sum_{i=1}^{m+1} q^i} \end{aligned} \tag{D.17}$$

$$D = \frac{q-1}{q^m-1}\left(2-\frac{4+\sum_{i=0}^{m-1} q^i}{q^{m+1}}\right). \tag{D.18}$$

Annexe E

Bons codes arithmétiques AN

E.1 Compression de SBS avec pertes

Le tableau suivant donne la liste des meilleurs codes arithmétiques pour la compression d'une source binaire symétrique. La recherche menée s'étend de l'extension 3 à l'extension 17 source de la source. Rappelons que si l'on considère l'extension n^{e} de la source, alors A et B vérifient les inégalités

$$\begin{aligned} 2 \leq A &\leq 2^n - 1 \\ \frac{2^n}{A} \leq B &< \frac{2^n}{A} + 1. \end{aligned}$$

Ces inégalités excluent le code universel ($A > 2$) et le code nul ($A \leq 2^n - 1$ et $2^n < AB$). La deuxième inégalité assure que la décomposition binaire du plus grand mot de code s'écrit sur n bits et que le code est de rendement maximal.

n	A	B	$D_{SBS,H}$	R	n	A	B	$D_{SBS,H}$	R
17	2	65536	2.9412e-02	9.4118e-01	16	2	32768	3.1250e-02	9.3750e-01
15	2	16384	3.3333e-02	9.3333e-01	14	2	8192	3.5714e-02	9.2857e-01
13	2	4096	3.8462e-02	9.2308e-01	17	3	43691	3.9216e-02	9.0677e-01
16	3	21846	4.1668e-02	9.0094e-01	15	3	10923	4.4446e-02	8.9434e-01
17	5	26215	4.7220e-02	8.6342e-01	16	5	13108	5.0266e-02	8.5489e-01
17	7	18725	5.1456e-02	8.3486e-01	16	7	9363	5.4890e-02	8.2455e-01
15	7	4682	5.8728e-02	8.1286e-01	17	11	11916	5.8829e-02	7.9651e-01
17	13	10083	6.2232e-02	7.8233e-01	16	13	5042	6.7342e-02	7.6874e-01
17	19	6899	7.1220e-02	7.5013e-01	17	23	5699	7.6738e-02	7.3391e-01
17	25	5243	8.1201e-02	7.2683e-01	17	29	4520	8.2152e-02	7.1424e-01
17	37	3543	8.7440e-02	6.9357e-01	17	39	3361	9.3776e-02	6.8910e-01
17	47	2789	9.4026e-02	6.7327e-01	17	49	2675	9.5126e-02	6.6972e-01
17	53	2474	9.7118e-02	6.6310e-01	17	55	2384	9.8011e-02	6.5995e-01
17	59	2222	9.9710e-02	6.5398e-01	17	61	2149	1.0049e-01	6.5114e-01
16	49	1338	1.0267e-01	6.4912e-01	17	67	1957	1.0311e-01	6.4320e-01
17	71	1847	1.0431e-01	6.3829e-01	17	77	1703	1.0713e-01	6.3140e-01
17	79	1660	1.0751e-01	6.2923e-01	17	83	1580	1.0871e-01	6.2504e-01
16	67	979	1.1162e-01	6.2095e-01	16	71	924	1.1297e-01	6.1573e-01
17	95	1380	1.1318e-01	6.1356e-01	17	97	1352	1.1375e-01	6.1182e-01
17	101	1298	1.1437e-01	6.0836e-01	17	103	1273	1.1607e-01	6.0671e-01
16	79	830	1.1655e-01	6.0606e-01	17	107	1225	1.1660e-01	6.0345e-01
17	109	1203	1.1680e-01	6.0191e-01	16	83	790	1.1793e-01	6.0161e-01
17	115	1140	1.1834e-01	5.9734e-01	17	119	1102	1.1988e-01	5.9447e-01
17	121	1084	1.2026e-01	5.9307e-01	17	123	1066	1.2173e-01	5.9165e-01
17	125	1049	1.2230e-01	5.9028e-01	17	133	986	1.2274e-01	5.8503e-01
16	101	649	1.2420e-01	5.8388e-01	17	135	971	1.2463e-01	5.8373e-01
17	139	943	1.2470e-01	5.8124e-01	17	143	917	1.2573e-01	5.7887e-01
16	109	602	1.2703e-01	5.7710e-01	17	149	880	1.2733e-01	5.7537e-01
17	153	857	1.2813e-01	5.7313e-01	17	157	835	1.2820e-01	5.7092e-01
17	163	805	1.2916e-01	5.6781e-01	17	167	785	1.3062e-01	5.6568e-01
17	173	758	1.3086e-01	5.6271e-01	17	175	749	1.3150e-01	5.6170e-01
17	179	733	1.3211e-01	5.5986e-01	17	185	709	1.3302e-01	5.5704e-01
17	187	701	1.3363e-01	5.5607e-01	16	139	472	1.3569e-01	5.5517e-01
17	203	646	1.3579e-01	5.4914e-01	17	211	622	1.3741e-01	5.4593e-01
17	215	610	1.3825e-01	5.4427e-01	17	221	594	1.3896e-01	5.4202e-01
17	223	588	1.3956e-01	5.4116e-01	17	233	563	1.3968e-01	5.3747e-01

Tab. E.1 – Liste des codes arithmétiques AN pour la compression de SBS ($3 \leq n \leq 17$)(début)

n	A	B	$D_{SBS,H}$	R	n	A	B	$D_{SBS,H}$	R
17	237	554	1.4119e-01	5.3610e-01	16	173	379	1.4239e-01	5.3538e-01
17	245	535	1.4246e-01	5.3314e-01	17	247	531	1.4323e-01	5.3250e-01
16	179	367	1.4357e-01	5.3248e-01	17	251	523	1.4382e-01	5.3122e-01
16	185	355	1.4460e-01	5.2948e-01	17	263	499	1.4510e-01	5.2723e-01
17	281	467	1.4518e-01	5.2160e-01	17	283	464	1.4606e-01	5.2106e-01
17	285	460	1.4704e-01	5.2032e-01	17	295	445	1.4768e-01	5.1751e-01
17	303	433	1.4866e-01	5.1519e-01	17	311	422	1.4925e-01	5.1301e-01
17	313	419	1.4946e-01	5.1240e-01	17	317	414	1.4996e-01	5.1138e-01
17	323	406	1.5049e-01	5.0973e-01	17	327	401	1.5158e-01	5.0867e-01
17	335	392	1.5179e-01	5.0675e-01	17	355	370	1.5272e-01	5.0185e-01
17	361	364	1.5326e-01	5.0046e-01	17	365	360	1.5391e-01	4.9952e-01
17	367	358	1.5407e-01	4.9905e-01	17	377	348	1.5445e-01	4.9664e-01
17	379	346	1.5552e-01	4.9615e-01	17	391	336	1.5634e-01	4.9367e-01
17	395	332	1.5643e-01	4.9265e-01	17	399	329	1.5676e-01	4.9188e-01
17	405	324	1.5799e-01	4.9058e-01	17	407	323	1.5801e-01	4.9032e-01
17	421	312	1.5825e-01	4.8738e-01	17	425	309	1.5829e-01	4.8656e-01
17	433	303	1.5857e-01	4.8489e-01	17	437	300	1.5970e-01	4.8405e-01
17	443	296	1.6003e-01	4.8291e-01	17	451	291	1.6068e-01	4.8146e-01
17	453	290	1.6131e-01	4.8117e-01	17	467	281	1.6175e-01	4.7850e-01
17	471	279	1.6205e-01	4.7789e-01	17	473	278	1.6249e-01	4.7758e-01
17	487	270	1.6267e-01	4.7511e-01	17	491	267	1.6421e-01	4.7416e-01
17	493	266	1.6462e-01	4.7384e-01	17	499	263	1.6611e-01	4.7288e-01
16	355	185	1.6665e-01	4.7071e-01	17	535	245	1.6685e-01	4.6686e-01
17	557	236	1.6730e-01	4.6368e-01	17	571	230	1.6878e-01	4.6150e-01
17	593	222	1.6966e-01	4.5850e-01	17	601	219	1.7008e-01	4.5734e-01
17	603	218	1.7096e-01	4.5695e-01	17	605	217	1.7112e-01	4.5656e-01
17	617	213	1.7114e-01	4.5498e-01	17	625	210	1.7186e-01	4.5378e-01
17	665	198	1.7261e-01	4.4879e-01	17	667	197	1.7497e-01	4.4836e-01
17	669	196	1.7541e-01	4.4792e-01	17	675	195	1.7586e-01	4.4749e-01
17	701	187	1.7593e-01	4.4394e-01	17	707	186	1.7701e-01	4.4348e-01
17	721	182	1.7712e-01	4.4164e-01	17	727	181	1.7717e-01	4.4117e-01
17	729	180	1.7744e-01	4.4070e-01	17	743	177	1.7772e-01	4.3927e-01
17	749	175	1.7791e-01	4.3831e-01	17	761	173	1.7932e-01	4.3733e-01
17	791	166	1.7970e-01	4.3383e-01	17	795	165	1.8027e-01	4.3331e-01
17	809	163	1.8038e-01	4.3228e-01	17	839	157	1.8175e-01	4.2910e-01
17	857	153	1.8235e-01	4.2691e-01	17	867	152	1.8346e-01	4.2635e-01

Tab. E.2 – Liste des codes arithmétiques AN pour la compression de SBS ($3 \leq n \leq 17$)(suite)

n	A	B	$D_{SBS,H}$	R	n	A	B	$D_{SBS,H}$	R
17	871	151	1.8392e-01	4.2579e-01	17	883	149	1.8397e-01	4.2466e-01
17	891	148	1.8493e-01	4.2409e-01	17	901	146	1.8538e-01	4.2293e-01
17	919	143	1.8570e-01	4.2117e-01	17	933	141	1.8590e-01	4.1997e-01
17	937	140	1.8603e-01	4.1937e-01	17	947	139	1.8612e-01	4.1876e-01
17	971	135	1.8784e-01	4.1628e-01	17	979	134	1.8826e-01	4.1565e-01
16	665	99	1.8894e-01	4.1433e-01	17	1001	131	1.9036e-01	4.1373e-01
17	1069	123	1.9086e-01	4.0838e-01	17	1099	120	1.9166e-01	4.0629e-01
17	1115	118	1.9219e-01	4.0486e-01	17	1125	117	1.9238e-01	4.0414e-01
17	1139	116	1.9324e-01	4.0341e-01	17	1141	115	1.9370e-01	4.0268e-01
17	1163	113	1.9428e-01	4.0119e-01	17	1187	111	1.9431e-01	3.9967e-01
17	1199	110	1.9549e-01	3.9890e-01	17	1211	109	1.9572e-01	3.9813e-01
17	1223	108	1.9608e-01	3.9735e-01	17	1245	106	1.9614e-01	3.9576e-01
17	1257	105	1.9729e-01	3.9496e-01	17	1267	104	1.9733e-01	3.9414e-01
17	1305	101	1.9809e-01	3.9166e-01	17	1319	100	1.9876e-01	3.9082e-01
17	1327	99	1.9912e-01	3.8996e-01	17	1347	98	2.0027e-01	3.8910e-01
17	1385	95	2.0055e-01	3.8646e-01	17	1417	93	2.0104e-01	3.8466e-01
17	1431	92	2.0172e-01	3.8374e-01	17	1441	91	2.0247e-01	3.8281e-01
17	1457	90	2.0278e-01	3.8187e-01	17	1497	88	2.0308e-01	3.7997e-01
17	1511	87	2.0489e-01	3.7900e-01	17	1577	84	2.0527e-01	3.7602e-01
17	1615	82	2.0548e-01	3.7397e-01	17	1623	81	2.0613e-01	3.7293e-01
17	1655	80	2.0676e-01	3.7188e-01	17	1673	79	2.0717e-01	3.7081e-01
17	1681	78	2.0790e-01	3.6973e-01	17	1713	77	2.0822e-01	3.6863e-01
17	1745	76	2.0906e-01	3.6753e-01	17	1763	75	2.1000e-01	3.6640e-01
17	1773	74	2.1054e-01	3.6526e-01	17	1811	73	2.1105e-01	3.6411e-01
17	1841	72	2.1152e-01	3.6294e-01	17	1899	70	2.1238e-01	3.6055e-01
17	1907	69	2.1249e-01	3.5933e-01	17	1939	68	2.1361e-01	3.5809e-01
16	1245	53	2.1466e-01	3.5800e-01	17	1957	67	2.1517e-01	3.5683e-01
17	2003	66	2.1665e-01	3.5555e-01	16	1305	51	2.1677e-01	3.5453e-01
16	1327	50	2.1756e-01	3.5274e-01	17	2163	61	2.1794e-01	3.4887e-01
17	2215	60	2.1799e-01	3.4746e-01	17	2291	58	2.1932e-01	3.4459e-01
17	2317	57	2.2096e-01	3.4311e-01	17	2383	56	2.2104e-01	3.4161e-01
17	2403	55	2.2193e-01	3.4008e-01	17	2473	54	2.2234e-01	3.3852e-01
17	2485	53	2.2387e-01	3.3694e-01	17	2531	52	2.2472e-01	3.3532e-01
17	2609	51	2.2513e-01	3.3367e-01	17	2671	50	2.2606e-01	3.3199e-01
17	2681	49	2.2700e-01	3.3028e-01	17	2777	48	2.2831e-01	3.2853e-01
17	2843	47	2.2890e-01	3.2674e-01	17	2857	46	2.2962e-01	3.2492e-01

Tab. E.3 – Liste des codes arithmétiques AN pour la compression de SBS ($3 \leq n \leq 17$)(suite)

n	A	B	$D_{SBS,H}$	R	n	A	B	$D_{SBS,H}$	R
17	2915	45	2.3076e-01	3.2305e-01	17	2997	44	2.3175e-01	3.2114e-01
17	3159	42	2.3363e-01	3.1720e-01	17	3257	41	2.3492e-01	3.1515e-01
17	3305	40	2.3522e-01	3.1305e-01	17	3433	39	2.3648e-01	3.1091e-01
17	3525	38	2.3845e-01	3.0870e-01	17	3547	37	2.3986e-01	3.0644e-01
17	3731	36	2.4026e-01	3.0411e-01	17	3763	35	2.4215e-01	3.0172e-01
17	3933	34	2.4387e-01	2.9926e-01	16	2473	27	2.4603e-01	2.9718e-01
17	3989	33	2.4732e-01	2.9673e-01	17	4311	31	2.4790e-01	2.9142e-01
17	4517	30	2.4848e-01	2.8864e-01	17	4525	29	2.5028e-01	2.8576e-01
17	4837	28	2.5206e-01	2.8279e-01	16	2857	23	2.5399e-01	2.8272e-01
17	4935	27	2.5404e-01	2.7970e-01	17	5235	26	2.5549e-01	2.7650e-01
17	5305	25	2.5698e-01	2.7317e-01	17	5607	24	2.5927e-01	2.6970e-01
17	5927	23	2.6125e-01	2.6609e-01	17	5991	22	2.6335e-01	2.6232e-01
17	6471	21	2.6531e-01	2.5837e-01	17	6695	20	2.6752e-01	2.5423e-01
17	7067	19	2.7019e-01	2.4988e-01	17	7575	18	2.7320e-01	2.4529e-01
16	4467	15	2.7623e-01	2.4418e-01	17	7771	17	2.7720e-01	2.4044e-01
16	4825	14	2.7986e-01	2.3796e-01	17	8615	16	2.8065e-01	2.3529e-01
17	9043	15	2.8247e-01	2.2982e-01	17	9781	14	2.8590e-01	2.2396e-01
17	10611	13	2.9001e-01	2.1767e-01	16	6471	11	2.9245e-01	2.1621e-01
17	11635	12	2.9410e-01	2.1088e-01	16	7091	10	2.9825e-01	2.0762e-01
17	12713	11	2.9890e-01	2.0350e-01	17	14163	10	3.0331e-01	1.9541e-01
17	15003	9	3.0930e-01	1.8647e-01	14	3253	6	3.1628e-01	1.8464e-01
17	18221	8	3.1789e-01	1.7647e-01	16	10649	7	3.1931e-01	1.7546e-01
15	6549	6	3.2103e-01	1.7233e-01	17	21363	7	3.2386e-01	1.6514e-01
16	12969	6	3.2927e-01	1.6156e-01	17	26037	6	3.3308e-01	1.5206e-01
16	14035	5	3.3973e-01	1.4512e-01	14	5459	4	3.4366e-01	1.4286e-01
17	27987	5	3.4599e-01	1.3658e-01	15	10919	4	3.5256e-01	1.3333e-01
16	21839	4	3.5578e-01	1.2500e-01	17	43687	4	3.6154e-01	1.1765e-01
9	511	2	3.6328e-01	1.1111e-01	15	15701	3	3.7285e-01	1.0566e-01
11	2047	2	3.7695e-01	9.0909e-02	13	8191	2	3.8721e-01	7.6923e-02
15	32767	2	3.9526e-01	6.6667e-02	17	131071	2	4.0181e-01	5.8824e-02

Tab. E.4 – Liste des codes arithmétiques pour la compression de SBS (fin)

Annexe F

Dysfonctionnements de l'algorithme de Peterson-Gorenstein-Zierler

Le but de cette annexe est d'expliquer brièvement pourquoi l'algorithme de Peterson-Gorenstein-Zierler, tel qu'il est habituellement présenté, est sujet à des dysfonctionnements lorsqu'il est appliqué à des codes BCH binaires. Ces dysfonctionnements furent mis en évidence par Sarwate, Srinivasan et Morrison [53, 59]. L'algorithme est en dysfonctionnement lorsqu'il ne détecte pas de dépassement de la capacité de correction et produit en sortie un mot qui n'appartient au code BCH choisi. Ces dysfonctionnements ne remettent pas en cause la capacité de cet algorithme à trouver le mot de code le plus proche du mot reçu lorsque ce dernier est à une distance inférieure ou égale à $\lfloor \frac{d-1}{2} \rfloor$ du mot de code (où d est la distance BCH). Le lecteur souhaitant en savoir plus à leur sujet est invité à se reporter à ces articles qui traitent aussi de l'algorithme d'Euclide étendu proposé par Sugiyama pour décoder les codes BCH. Puis nous montrerons pourquoi le même algorithme appliqué aux codes BCH réels MDS ne peut être sujet à de tels dysfonctionnements.

Par la suite, α désigne une racine n^{e} de l'unité primitive.

F.1 Cas des codes BCH binaires

Rappelons tout d'abord les conditions dans lesquelles la version « traditionnelle » de l'algorithme de Peterson-Gorenstein-Zierler entre en dysfonctionnement par le théorème formulé par Srinivasan et Sarwate.

Théorème F.1.1 *Soit $\mathcal{C}_1$ un code BCH binaire de longueur n et de distance BCH $d = 2t + 1$. Soit i un entier tel que $i \leq t - 1$ et soit le $\mathcal{C}_2$ BCH binaire de longueur n*

et de distance BCH supérieure ou égale à $t+i+1$. Si le mot reçu $\underline{y}$ est à distance i d'un mot $\underline{c} \in \mathcal{C}_2 \setminus \mathcal{C}_1$, alors l'algorithme de Peterson-Gorenstein-Zierler produira comme mot décodé le mot $\underline{c}$. Réciproquement, si la sortie du décodeur est un mot n'appartenant pas au code $\mathcal{C}_1$ alors ce mot appartient à $\mathcal{C}_2$, i étant le poids de l'erreur estimé par l'algorithme de Peterson-Gorenstein-Zierler.

Le lecteur est invité à se reporter à l'article [59] pour trouver la démonstration complète de ce théorème. Nous allons retranscrire ici uniquement la démonstration de l'implication :

$\forall\, \underline{y} \in F_2^n \,/\, \exists \underline{c} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ et $d_H(\underline{y}, \underline{c}) \leq t-1 \implies$ la sortie du décodeur associé à $\mathcal{C}_1$ est $\underline{c}$

Rappelons également la définition des matrices M_j :

$$M_j = \begin{pmatrix} S_b & \cdots & S_{b+j-1} \\ \vdots & & \vdots \\ S_{b+j-1} & \cdots & S_{b+2j-2} \end{pmatrix} \tag{F.1}$$

Considérons le décodeur associé au code $\mathcal{C}_2$. Le mot reçu $\underline{y}$ est à distance i, donc inférieure à $\lfloor \frac{t+i}{2} \rfloor$, du mot de code $\underline{c}$ de $\mathcal{C}_2$. La sortie du décodeur associée à $\mathcal{C}_2$ est donc égale à $\underline{c}$. Ce décodeur estime donc le nombre d'erreur à i et résout le système

$$M_i \begin{pmatrix} \Lambda_i \\ \vdots \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_{b+i} \\ \vdots \\ S_{b+2i-1} \end{pmatrix} \tag{F.2}$$

où M_i est consitué des syndromes calculés pour le code $\mathcal{C}_2$ Rappelons à présent l'équation clé de décodage associée au code $\mathcal{C}_2$.

$$(1+S(x))\Lambda(x) = \Omega(x) \quad [x^{2t+2i+1}]$$

où Ω est un polynôme de degré i. Par conséquent, les coefficients de degré strictement supérieur à i et inférieur ou égal à $i+t$ du polynôme $(1+S(x))\Lambda(x) \quad [x^{2t+2i+1}]$ sont nuls, ce qui matriciellement signifie

$$\begin{pmatrix} S_b & \cdots & S_{b+i-1} \\ \vdots & & \vdots \\ S_{b+t-1} & \cdots & S_{b+t+i-2} \end{pmatrix} \begin{pmatrix} \Lambda_i \\ \vdots \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_{b+i} \\ \vdots \\ S_{b+i+t-1} \end{pmatrix} \tag{F.3}$$

Remarquons que les polynômes associés aux mots du code $\mathcal{C}_2$ ont obligatoirement comme racines $\alpha^b, \cdots, \alpha^{b+t+i-1}$. Ceux associés aux mots du code $\mathcal{C}_1$ ont obligatoirement, en plus de ces $t+i$ racines, $\alpha^{b+i}, \cdots, \alpha^{b+2t-1}$ pour racines supplémentaires. Les $t+i$ syndromes associés au code $\mathcal{C}_2$ se confondent donc avec les $t+i$ premiers syndromes associés au code $\mathcal{C}_1$.

Nous en déduisons que la matrice M_t calculée par le décodeur associé au code $\mathcal{C}_1$ est non inversible car l'équation (F.3) montre que la $(i+1)^{\text{e}}$ligne de M_t est une combinaison

linéaire des i premières. Il en de même pour les matrices $M_{t-1}, \cdots, M_{i+1}$. La matrice M_i calculée par le décodeur associé au code $\mathcal{C}_1$ est inversible car le décodeur associé à $\mathcal{C}_2$ a bien fonctionné. Par conséquent, le décodeur associé à $\mathcal{C}_1$ estime le nombre d'erreurs à i et son cheminement se confond alors avec celui du décodeur associé à $\mathcal{C}_2$: la sortie du décodeur associé à $\mathcal{C}_1$ est donc $\underline{c}$, d'où le dysfonctionnement.

Exemple 3 *Illustrons ce dysfonctionnement en choisissant pour code $\mathcal{C}_1$ le code BCH au sens strict de longueur 15, de dimension 5 et de distance BCH 7 ($t = 3$). Le corps localisateur est F_{16} qui est isomorphe à $F_2[x]/(x^4 + x + 1)$. Le polynôme générateur de $\mathcal{C}_1$ est $g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$. Supposons que le mot reçu par le décodeur soit $y(x) = 1 + x^4 + x^6 + x^7 + x^8 + x^{14}$. Les 6 syndromes à calculer correspondent aux valeurs du polynôme $y(x)$ prises respectivement en $\alpha, \alpha^2, \cdots, \alpha^6$. Tout calcul fait, nous trouvons*

$$\begin{array}{lcllcl} S_1 & = & \alpha^{14} & S_2 & = & \alpha^{13} \\ S_3 & = & \alpha^{12} & S_4 & = & \alpha^{11} \\ S_5 & = & 0 & S_6 & = & \alpha^9 \end{array}$$

Le déterminant des matrices M_3 et M_2 est nul alors que celui de M_1 est égal à S_1. Le décodeur estime donc à 1 le nombre d'erreurs. Le polynôme localisateur d'erreurs se déduit donc facilement $\Lambda(x) = 1 + \alpha^{14}x$. Connaissant alors la position de l'erreur, il faut alors résoudre l'équation $X_1 e_{14} = S_1$ ce qui nous conduit à $e_{14} = 1$. Le mot de code estimé $\underline{\hat{c}}$ par le décodeur est donc $\hat{c}(x) = 1 + x^4 + x^6 + x^7 + x^8$ qui n'est autre que le polynôme générateur du code BCH de longueur 15, de dimension 7 et de distance BCH 5. Les mots du code $\mathcal{C}_1$ les plus proches du mot reçu sont les mots $1 + x^4 + x^7 + x^8 + x^{10} + x^{12} + x^{13} + x^{14}$ et $1 + x^2 + x^4 + x^5 + x^6 + x^7 + x^{11} + x^{14}$ à distance 4.

Il est possible de pallier à un tel dysfonctionnement grâce aux relations de Newton comme l'ont proposé Srinivasan, Sarwate et Morrison [53, 59] : les syndromes et les coefficients du polynôme Λ doivent vérifier ces $2t$ équations. L'algorithme de Peterson-Gorenstein-Zierler assure que les t premières équations sont vérifiées [59]. Il suffit donc de vérifier les $t - \nu$ équations

$$\begin{pmatrix} S_{b+t} & \cdots & S_{b+t+\nu-1} \\ \vdots & & \vdots \\ S_{b+2t-\nu-1} & \cdots & S_{b+2t-2} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ \vdots \\ e_{i_\nu} \end{pmatrix} = - \begin{pmatrix} S_{b+t+\nu} \\ \cdots \\ S_{b+2t-1} \end{pmatrix} \qquad \text{(F.4)}$$

sont vraies. Si elles ne le sont pas, ce décodeur modifié détecte un dépassement de capacité de correction (dans l'exemple 3 ci-dessus, $X_1 S_4 \neq S_5$ ($\alpha^{10} \neq 0$)). Lorsque d'autres cas de dépassement de correction se produisent, l'algorithme échoue à la recherche des racines de Λ ou à l'évaluation des erreurs : les racines n'appartiennent pas à F_{2^m} ou les erreurs n'appartiennent pas à F_2.

F.2 Cas des codes BCH réels MDS

L'algorithme de Peterson transposé directement au cas réel n'est pas sujet à de tels dysfonctionnements. Considérons le code BCH réel MDS $\mathcal{C}_1$ de longueur impaire n et de distance d impaire, $d = 2t+1$. Soit $\mathcal{C}_2$ le code BCH réel MDS de même longueur et de distance BCH d' avec d' le plus petit entier impair supérieur ou égal à $t+i+1$ où i est un entier tel que $i < t-1$: $\mathcal{C}_2$ contient donc $\mathcal{C}_1$ et ne peut être égal à $\mathcal{C}_1$ (car $i < t-1$).

Le mot reçu est supposé être à distance i d'un mot $\underline{c} \in \mathcal{C}_2 \setminus \mathcal{C}_1$. Calculons les syndromes du mot reçu relativement au code $\mathcal{C}_1$:

$$0 \leq j \leq 2t \quad S_j = y(\alpha^{b+j-1}) \tag{F.5}$$

Posons t' tel que $d' = 2t'+1$. Les syndromes relatifs au code $\mathcal{C}_2$ sont définis par

$$0 \leq j \leq 2t' \quad S'_j = y(\alpha^{b+t-t'+j-1}) = S_{t-t'+j} \tag{F.6}$$

On en déduit que le décodeur relatif au code $\mathcal{C}_1$ n'examine pas les mêmes matrices que celui relatif au code $\mathcal{C}_2$ pour déterminer le nombre d'erreurs. Du bon fonctionnement du décodeur relatif au code $\mathcal{C}_2$, on déduit que le décodeur relatif à $\mathcal{C}_1$ estimera un nombre d'erreurs ν inférieur ou égal à $t-t'+i$. Par conséquent, la démonstration des dysfonctionnements dans le cas binaire ne peut s'appliquer au cas réel. L'exemple qui suit montre que le théorème énoncé par Srinivasan et Sarwate ne peut s'appliquer tel quel aux codes BCH réels MDS.

Exemple 4 *Prenons $n = 15$, $t = 3$ et $t' = 2$. Le polynôme générateur g_1 du code $\mathcal{C}_1$ est donné par*

$$g_1(x) = (1 + 2x\cos\left(\frac{\pi}{15}\right) + x^2)(1 + 2x\cos\left(\frac{3\pi}{15}\right) + x^2)(1 + 2x\cos\left(\frac{5\pi}{15}\right) + x^2).$$

Le polynôme générateur g du code $\mathcal{C}_2$ est, quant à lui, donné par :

$$g_2(x) = (1 + 2x\cos\left(\frac{\pi}{15}\right) + x^2)(1 + 2x\cos\left(\frac{3\pi}{15}\right) + x^2).$$

Considérons que le décodeur relatif au code $\mathcal{C}_1$ reçoive le mot $\underline{y}$ tel que $y(x) = g_2(x) + x^{10}$. Autrement dit, nous sommes dans des conditions analogues à celles des dysfonctionnements observés dans le cas binaire. Les 6 syndromes de $\underline{y}$ ont pour expression :

$$\begin{array}{lcllcl} S_1 & = & 16e^{-\frac{2\sqrt{-1}\pi}{3}} \prod_{l=1}^{4} \sin\left(\frac{l\pi}{15}\right) + e^{\frac{2\sqrt{-1}\pi}{3}} & S_6 & = & S_1^* \\ S_2 & = & 1 & S_5 & = & 1 \\ S_3 & = & e^{-\frac{2\sqrt{-1}\pi}{3}} & S_4 & = & e^{+\frac{2\sqrt{-1}\pi}{3}} \end{array}$$

Il est alors aisé de voir que la matrice M_3 est non inversible alors que M_2 a un déterminant non nul (égal à $16e^{-\frac{4\sqrt{-1}\pi}{3}} \prod_{l=1}^{4} \sin\left(\frac{l\pi}{15}\right)$). Le décodeur estime donc le nombre d'erreurs à 2, puis résout le système

$$\begin{pmatrix} S_1 & S_2 \\ S_2 & S_3 \end{pmatrix} \begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_3 \\ S_4 \end{pmatrix} \tag{F.7}$$

La solution trouvée est alors $\Lambda_2 = 0$ et $\Lambda_1 = e^{\frac{\sqrt{-1}\pi}{3}}$. La version traditionnelle de l'algorithme ne teste pas la valeur des coefficients du polynôme localisateur. La recherche des racines de ce polynôme aboutit à un échec car on ne trouve qu'une racine au lieu de deux distinctes. Le décodeur BCH réel détecte donc le dépassement de capacité de correction.
Profitons-en pour examiner la résolution du système surdéterminé par les moindres carrés, qui est celle mise en œuvre par notre version de l'algorithme de Peterson-Gorenstein-Zierler :

$$\begin{pmatrix} S_1 & S_2 \\ S_2 & S_3 \\ S_3 & S_4 \\ S_4 & S_5 \end{pmatrix} \begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = - \begin{pmatrix} S_3 \\ S_4 \\ S_5 \\ S_6 \end{pmatrix}. \tag{F.8}$$

Après plusieurs manipulations algébriques fastidieuses, nous trouvons

$$\begin{array}{rcccl} \Lambda_2 & = & \frac{S_3}{3} & = & -\frac{1+\sqrt{-3}}{6} \\ \Lambda_1 & = & \frac{(1-\lambda)S_3 - 2S_3^*}{3} & = & \frac{1+\lambda+(3-\lambda)\sqrt{-3}}{6} \\ \lambda & = & 16 \prod_{l=1}^{4} \sin\left(\frac{l\pi}{15}\right). & & \end{array}$$

Les racines de ce polynôme localisateur ne sont pas des racines n^{es} de l'unité : le décodeur détecte donc un dépassement de capacité de correction.