N° d'ordre: 3470

THÈSE

présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention TRAITEMENT DU SIGNAL ET TÉLÉCOMMUNICATIONS

par

Khaled LAJNEF

Équipe d'accueil : Temics - IRISA
École Doctorale : Matisse
Composante universitaire : S.P.M.

Titre de la thèse :

Étude du codage de sources distribuées pour de nouveaux concepts en compression vidéo

Soutenue le 14 décembre 2006 devant la commission d'examen

M. :	Joseph	RONSIN	Président
MM. :	Ramesh	PYNDIAH	Rapporteurs
	Michel	KIEFFER	
MM. :	Béatrice	PESQUET-POPESCU	Examinateurs
	Pierre	SIOHAN	
	Christine	GUILLEMOT	

If a man will begin with certainties, he shall end in doubts ;
but if will be content to begin with doubts, he shall end in certainties.
par Francis BACON, The Advancement of Learning, Book One (1905)

Remerciements

Ce travail de thèse a été réalisé au sein de l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) à Rennes.

Je tiens à remercier tout particulièrement Christine Guillemot, Directrice de Recherche à l'INRIA et responsable du projet TEMICS de m'avoir acceuillie dans ce projet et d'avoir dirigé mes travaux de thèse. Je remercie également Pierre Siohan, Chercheur à France Télécom R&D, pour son aide, ses conseils et son regard critique en tant qu'encadrant.

Je remercie Joseph Ronsin, Professeur à l'INSA de Rennes, qui me fait l'honneur de présider ce jury.

Je remercie Ramesh Pyndiah, Professeur à l'École nationale Supérieure des Télécommunications de Bretagne et Michel Kieffer, Maître de conférences à l'Université de Paris-Sud, d'avoir bien voulu accepter la charge de rapporteur.

J'associe pleinement à ces remerciements Béatrice Pesquet-Popescu, Maitre de conférences à l'École nationale Supérieure des Télécommunications de Paris, d'avoir bien voulu juger ce travail.

Je tiens également à remercier mes collègues de l'équipe TEMICS à l'IRISA avec qui j'ai eu le plaisir de travailler.

Je remercie mes amis et ma famille pour leur soutient au cours de ces trois dernières années. Enfin j'aimerais remercier ma femme, qui a pris part malgré elle à cette aventure.

Table des matières

Introduction

Le domaine de la compression vidéo a connu, ces dernières années, de fortes évolutions menant à l'émergence d'un nombre important de standards internationaux (H.26X, MPEG-X). Pour réduire le débit, les systèmes actuels (les normes de MPEG-X ou de H.26X) utilisent la corrélation temporelle en mettant en œuvre un codage prédictif (désigné par codage d'image en mode Inter) compensé en mouvement. Ce choix se traduit par des encodeurs dont la complexité est en général grandement plus élevée (5 à 10 fois) que celle du décodeur. Cette différence est donc liée au fait que l'estimation et la compensation du mouvement sont calculées et effectuées au codeur, tandis que le décodeur va simplement utiliser les vecteurs mouvement pour reconstruire l'image décodée. Ce schéma de conception asymétrique est tout à fait adapté pour les applications actuelles du codage vidéo, que ce soit la télévision numérique ou encore le téléchargement sur des mobiles à partir de serveurs. Le développement considérable des mobiles a engendré un besoin inverse : celui de transmettre un flux vidéo vers une station de base. Pour ce type d'application, il peut être plus judicieux de rechercher un schéma de codage dual du précédent avec un codeur de complexité relativement limitée et un décodeur disposant d'une puissance de traitement nettement plus importante. Dans ce contexte, le codage vidéo distribué peut être vu comme une façon différente d'exploiter cette corrélation temporelle au décodeur en considérant, par exemple, que dans deux images successives, la deuxième constitue une version bruitée de la première (les images sont codées en mode Intra et décodées en mode Inter). La figure 1 montre l'utilisation du codage de sources distribuées dans un schéma de communication radio-mobile. Les deux terminaux mobiles dans la figure 1 présentent des complexités au codage et au décodage très faibles grâce à l'utilisation respective d'un encodeur vidéo distribué et d'un décodeur vidéo traditionnel (MPEG-X ou de H.26X). En outre, ce schéma de codage doit être robuste aux erreurs de canal de sorte que le décodeur puisse récupérer la scène avec une qualité de reconstruction élevée.

Le codage de sources distribuées a pour but de comprimer des signaux fortement corrélés que l'on code séparément et décode conjointement. Cette structure s'applique également aux réseaux de capteurs, qui transmettent des informations fortement corrélées à une unité de traitement centrale. D'un point de vue théorique, le codage de sources distribuées s'appuie sur le théorème de Slepian-Wolf [SW73] établi en 1973. Il montre la possibilité de réaliser, sans perte en efficacité de compression, un codage séparé (au lieu de conjoint) de 2 sources corrélées X et Y à valeurs discrètes avec un débit $R_X + R_Y \geq H(X,Y)$, $R_X \geq H(X|Y)$ et $R_Y \geq H(Y|X)$). Plus tard ce résultat a été

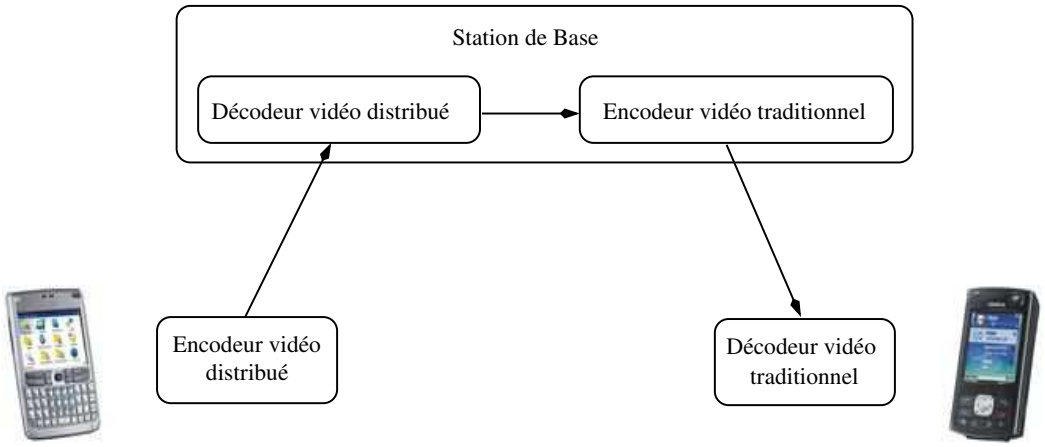

FIG. 1 – Schéma d'un lien de communication radio-mobile utilisant le principe de codage vidéo distribué.

étendu par Wyner et Ziv [WZ76] au cas de sources à valeurs continues.

Bien que la limite théorique du codage de deux sources distribuées à valeurs discrètes a été établie en 1973, ce n'est que très récemment que des mises en œuvre concrètes ont été proposées afin d'approcher au mieux cette borne. Celles-ci se basent sur des codeurs de sources, en utilisant par exemple des codes à longueur variable (CLV) [JAI97], [ZE01] d'une part, et des codes de canal comme les codes convolutifs [PR99], les codes turbo [AG02] ou les codes LDPC [LXG02b] d'autre part. L'utilisation des codeurs de canal dans le contexte du codage de sources distribuées a été motivée par le fait que le modèle de corrélation entre X et Y est similaire à celui d'un "canal de corrélation" virtuel. L'entrée et la sortie de ce "canal" sont respectivement X et Y.

Le codeur Wyner-Ziv peut être vu comme un quantificateur concaténé en série avec un codeur Slepian-Wolf. Par conséquent, beaucoup de travaux ont été réalisés pour trouver le meilleur quantificateur adapté au codage de sources distribuées comme : quantificateur de Lloyd-Max généralisé [RMZG03], quantificateur en lattice emboîtée [ZS98], [Ser00], [ZSE02], [XLCL03], [LCLX04] ou bien les quantificateurs codés en treillis [PR03a], [CPR03] et [YCXZ03].

Dans cette thèse, nous nous intéressons aux deux types de codeurs : Slepian-Wolf et Wyner-Ziv. Afin d'avoir un codeur avec une complexité réduite, le code turbo poinçonné sera utilisé dans nos schémas de codage de sources distribuées. Dans ce cadre, les contributions de cette étude portent dans un premier temps sur un ensemble d'outils de base pour le codage de Slepian-Wolf et de Wyner-Ziv :

- Nous proposons une nouvelle classe de patrons de poinçonnage du code turbo. Ce poinçonnage augmente l'efficacité de correction du code turbo dans un contexte de codage de sources distribuées et permet également de s'approcher des limites théoriques de Slepian-Wolf et de Wyner-Ziv.
- Nous introduisons un mode d'utilisation de la quantification codée par treillis (TCQ) qui permet d'améliorer les performances débit-distorsion du codeur de

Wyner-Ziv. Dans ce schéma, nous avons proposé de transmettre les bits de chemin du treillis sans compression.
- Nous établissons une nouvelle démonstration pour le calcul des valeurs limites théoriques du codage de Wyner-Ziv avec une information de bord partielle.
- Nous proposons des extensions aux théorèmes de Slepian-Wolf et de Wyner-Ziv au cas de sources multiples respectivement binaires et Gaussiennes. Nous décrivons des schémas de mise en œuvre du codage distribué de trois sources binaires et Gaussiennes basés sur un code turbo poinçonné. Cette approche nous a permis de s'approcher des bornes théoriques avec des corrélations plus faibles.

Dans un second temps, nos contributions sont consacrées au schéma de codage vidéo distribué :

- Vu la sous-optimalité du quantificateur scalaire uniforme dans un schéma de codage vidéo distribué, l'utilisation de la quantification TCQ a été proposée. Pour préserver une complexité faible à l'encodage, la TCQ a été appliquée seulement aux coefficients de bandes DC.
- Nous décrivons une technique d'amélioration de la qualité de l'information de bord dans un schéma de codage vidéo de type Wyner-Ziv utilisant le principe du codage de trois sources distribuées.
- Afin de réduire la latence du décodeur vidéo distribué, nous proposons un mécanisme hybride de contrôle de débit au codage et au décodage.

La thèse est organisée en cinq chapitres. Nous commençons dans le chapitre 1 par le cadre théorique du codage de sources distribuées. Les théorèmes de Slepian-Wolf et de Wyner-Ziv sont présentés. Le chapitre 2 est consacré à l'état de l'art des schémas de mise en œuvre des techniques de codage distribué de deux sources corrélées binaires et Gaussiennes. Au chapitre 3, nous nous intéressons en première partie, au concept du codage de sources distribuées basé sur le code turbo poinçonné. Dans un deuxième temps, nous considérons le cas du codeur de deux sources distribuées utilisant la quantification TCQ et le code turbo poinçonné. Une re-démonstration et un schéma de mise en œuvre du codage de Wyner-Ziv avec information de bord partielle sont présentés. Enfin, l'effet d'un bruit de canal sur les performances débit-distorsion d'un codeur de sources distribuées est étudié. Au chapitre 4, nous considérons le cas de la compression de trois sources corrélées. Nous étendons les théorèmes de Slepian-Wolf et de Wyner-Ziv au cas de sources multiples respectivement binaires et Gaussiennes. Des propositions de mise en œuvre du codage distribué de trois sources binaires et Gaussiennes utilisant le code turbo poinçonné sont présentées. Dans le chapitre 5, l'état de l'art des schémas de codage vidéo distribué est exposé. Nous terminons au chapitre 6 en présentant nos contributions au schéma de codage vidéo distribué. L'utilisation de la quantification TCQ dans le contexte du codage vidéo distribué est proposée. Un système de compression vidéo appliquant le principe du codage de trois sources distribuées est décrit. Dans une autre partie, des nouvelles techniques de contrôle de débit au codage et au décodage sont présentées. Et enfin, nous étudions également la robustesse du codeur vidéo distribué en présence d'un bruit de canal.

Chapitre 1

Cadre théorique

La théorie de l'information est devenue aujourd'hui incontournable dans la conception de tout système de communication. Cette notion fut introduite par Claude E. Shannon [Sha48] en 1948 afin d'établir les limites théoriques de compression de données numériques et de débit de transmission d'informations en présence des canaux bruités. En particulier, il est montré que pour un processus aléatoire discret X dont on connaît les statistiques, la limite du taux de compression est l'entropie $H(X)$. Par contre, pour deux sources corrélées X et Y qui sont codées et décodées conjointement, le taux minimal de compression n'est que l'entropie conjointe $H(X,Y)$. Mais peut-on obtenir les mêmes performances avec une très faible redondance dans le cas où les deux sources sont codées séparément et décodées conjointement ?

La réponse à cette question a été présentée par Slepian et Wolf (Slepian-Wolf) en 1973 [SW73]. Ils ont présenté un théorème basé sur la compression de sources corrélées et discrètes qui sont codées indépendamment mais décodées conjointement. Ce type de compression est connu sous le nom de codage de sources distribuées. Plus tard ce théorème a été généralisé par Wyner et Ziv (Wyner-Ziv) [WZ76] au cas des sources à valeurs continues.

Ce chapitre a pour but de présenter les outils théoriques du codage de sources distribuées qui seront utilisés par la suite. Dans un premier temps, nous décrivons quelques outils de théorie de l'information utilisés pour la compression avec et sans perte. Nous commençons par la définition des diverses mesures d'information pour ensuite poursuivre en abordant la théorie débit-distorsion. Un développement plus détaillé de la plupart de ces résultats se trouve dans [CT91].

Dans un deuxième temps, les théorèmes de Slepian-Wolf et de Wyner-Ziv pour le codage distribué de deux sources corrélées respectivement binaires et Gaussiennes sont présentés.

1.1 Théorie de l'information

Dans cette section, nous présentons les différentes mesures d'information utilisées dans la théorie de l'information.

1.1.1 Quantité d'information

Soit une variable aléatoire discrète X qui prend des valeurs dans un ensemble de possibilités Ω. Soit $p(x) = Pr\{X = x\}, \ \forall x \in \Omega$, la loi de probabilité de la variable X.

On définit la quantité d'information comme étant une mesure de la connaissance qu'un observateur acquiert lorsqu'un événement se produit, connaissant seulement la probabilité de celui-ci. Plus l'événement est probable, moins la quantité d'information (ou incertitude) correspondante est grande. Cette quantité s'exprime en bits et est définie par :

$$\mathbb{I}(X = x) = -\log_2(p(x)) \tag{1.1}$$

Particulièrement, si $p(x) = 1$ alors $\mathbb{I}(X = x) = 0$, c'est-à-dire l'occurrence d'un événement certain ne peut fournir aucune information.

1.1.2 Entropie

L'entropie d'une variable aléatoire X est une mesure quantitative de l'incertitude associée aux valeurs prises par celle-ci. Elle est le nombre de bits moyens nécessaires pour décrire X. L'entropie est la moyenne de la quantité d'information $\mathbb{I}(X = x)$, notée $H(X)$. Cette quantité, exprimée en bits/symbole, est définie par :

$$H(X) = -\sum_{x \in \Omega} p(x) \log_2(p(x)) \tag{1.2}$$

L'entropie est une quantité positive. Elle présente une borne supérieure égale à $\log_2(|\Omega|)$ pour un ensemble Ω à alphabet fini.

L'entropie conjointe de deux variables aléatoires ($X \in \Omega$ et $Y \in \Psi$) avec une distribution (conjointe) $Pr(X = x, Y = y)$ est définie par :

$$H(X,Y) = -\sum_{(x,y) \in \Omega \times \Psi} p(x,y) \log_2(p(x,y)) \tag{1.3}$$

L'entropie conjointe de deux événements est toujours supérieure ou égale à l'entropie de chaque événement ($H(X,Y) \geq H(X)$ et $H(X,Y) \geq H(Y)$). Cela est du au fait que $p(x,y) = p(y|x)p(x) \leq p(x)$. L'entropie conjointe peut aussi s'écrire sous la forme suivante :

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) \tag{1.4}$$

où $H(Y|X)$ et $H(X|Y)$ les entropies conditionnelles respectivement de Y sachant X et de X sachant Y, sont définies par :

$$H(X|Y) = -\sum_{(x,y) \in \Omega \times \Psi} p(x,y) \log_2(p(x|y)) \tag{1.5}$$

$$H(Y|X) = -\sum_{(x,y) \in \Omega \times \Psi} p(x,y) \log_2(p(y|x)) \tag{1.6}$$

Pour généraliser, on définit l'entropie conjointe de n variables $\{X_0, X_1, \ldots, X_{n-1}\}$ par :

$$H(X_0, X_1, \ldots, X_{n-1}) = H(X_0) + H(X_1|X_0) + \cdots + H(X_{n-1}|X_0, X_1, \ldots, X_{n-2}) \tag{1.7}$$

1.1.3 Information mutuelle

L'information mutuelle entre deux variables aléatoires ($X \in \Omega$ et $Y \in \Psi$) est définie par :

$$\begin{aligned} I(X;Y) &= \sum_{(x,y)\in\Omega\times\Psi} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) && (1.8)\\ &= H(X) - H(X|Y) && (1.9)\\ &= H(Y) - H(Y|X) && (1.10)\\ &= H(X) + H(Y) - H(X,Y) && (1.11) \end{aligned}$$

On en déduit que $I(X;Y)$ est égale à 0 si et seulement si les deux variables aléatoires X et Y sont indépendantes. L'information mutuelle représente la réduction de l'incertitude par rapport à une des deux variables due à l'observation de l'autre variable. Elle mesure la quantité d'information que deux variables aléatoires s'apportent mutuellement. La figure 1.1 résume les équations (1.9), (1.10) et (1.11). A partir de cette figure, on peut déduire la borne supérieure de l'information mutuelle qui est : $I(X;Y) \leq min(H(X), H(Y))$.

L'information mutuelle est très utile pour déterminer la capacité des systèmes de communications.

1.2 Technique de codage de source et de compression

Le rôle du codage de source est d'éliminer la redondance contenue dans l'information produite par la source et de représenter celle-ci numériquement. Cette section fournit une introduction au codage de source, également connu sous le nom de compression de données. Les systèmes utilisés dans le codage de source sont des systèmes de compression avec ou sans perte.

La compression consiste à réduire la taille physique de blocs d'informations. Elle est caractérisée par le facteur de compression, c'est-à-dire le rapport entre le nombre de bits dans les données compressées par celui présent dans les données originales.

Dans son article daté de 1948 [Sha48], "A Mathematical Theory of Communication", Shannon a développé la théorie de la compression de données. Il a établi qu'il y a une limite fondamentale à la compression de données sans perte. Cette limite est appelée le taux d'entropie H de la source à compresser. La valeur exacte de l'entropie H dépend de la nature statistique de la source. Il est possible de compresser une source sans perte (reconstruction parfaite des données originales), à un taux de compression près de son

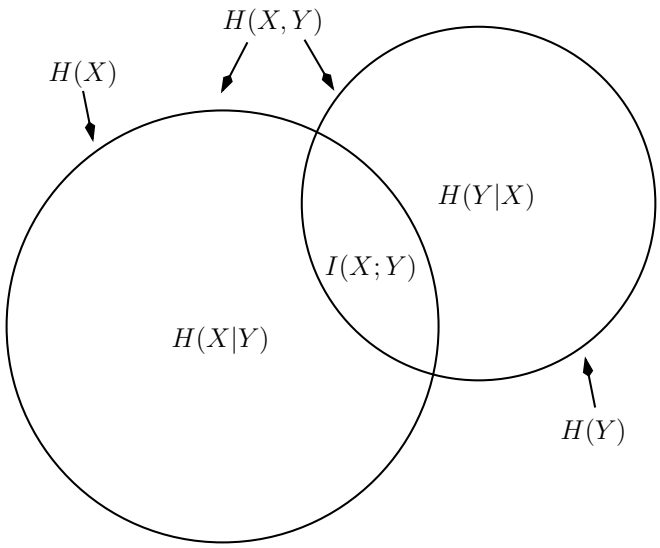

FIG. 1.1 – Relations entre entropie et information mutuelle.

entropie. Par contre, il est impossible mathématiquement de faire mieux que l'entropie H de la source.

Shannon a également développé la théorie de la compression de données avec perte, connue sous le nom de théorie débit-distorsion. Dans la compression de données avec perte, les données décompressées ne sont pas exactement identiques aux données originales. Dans ce cas, une certaine mesure de distorsion D entre les données reconstruites et les données originales est tolérée. Shannon a prouvé que pour une source (toutes ses propriétés statistiques étant connues) et pour une mesure de distorsion donnée, correspond une fonction, $R(D)$, appelée la fonction débit-distorsion. Par conséquent, le meilleur système de compression est celui qui fournit un compromis attrayant entre le taux de compression et la distorsion. Quand la compression est sans perte (aucune distorsion ou $D = 0$), le taux minimal de compression est $R(0) = H$ (pour une source à alphabet fini).

Dans ce sens, la théorie débit-distorsion est une généralisation de la théorie de la compression de données sans perte, car elle peut conduire à des systèmes ayant des mesures de distorsion nulle ($D = 0$) ou strictement positives ($D > 0$). Cette théorie fixe les limites théoriques de tous les algorithmes de compression de données mais n'indique pas exactement comment concevoir et mettre en œuvre ces algorithmes.

1.2.1 Compression sans perte

Un système de compression des données sans perte (en anglais : lossless compression) est illustré à la figure 1.2. Ce système s'applique aux sources discrètes, dont les statistiques sont connues. Les classes de codes le plus généralement utilisées dans un système de compression sans perte sont les codes de Huffman et les codes arithmétiques.

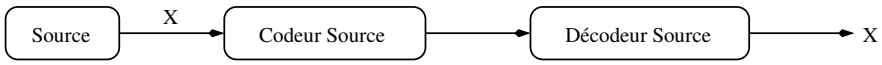

FIG. 1.2 – Système de compression des données sans perte.

Un encodeur de Huffman prend en entrée un bloc de caractères de longueur fixe et produit en sortie un bloc de longueur variable. L'encodeur arithmétique code la séquence entière pour avoir en sortie un seul mot de code associé. La conception du code de Huffman est optimale (pour une longueur de bloc fixe), en supposant que les statistiques de la source sont connues a priori. Le mot de code assigné pour chaque symbole est composé d'un nombre entier de bits. Le code arithmétique est conçu pour une grande classe de sources. Le mot de code généré est représenté sous forme d'une fraction rationnelle. Le code arithmétique permet de mieux s'approcher du taux de compression théorique que le code de Huffman.

1.2.1.1 Codage d'Huffman

David Huffman a proposé en 1952 [Huf52] une méthode statistique qui permet d'attribuer un mot de code binaire aux différents symboles à compresser. La longueur de chaque mot de code n'est pas identique pour tous les symboles : les symboles les plus fréquents (avec des probabilités élevées) sont codés avec de petits mots de code, tandis que les symboles les plus rares (avec des probabilités basses) reçoivent de plus longs codes binaires. Ce concept est semblable à celui du code Morse. On parle de codage à longueur variable CLV (en anglais VLC pour variable code length) préfixé, pour désigner ce type de codage car aucun code n'est le préfixe d'un autre. Ainsi la suite finale de mots codés à longueurs variables sera en moyenne plus petite qu'avec un code de longueur fixe.

1.2.1.2 Codage arithmétique

Le codage arithmétique [Pas76], [Ris76], [RL79] permet, à partir de la probabilité d'apparition des symboles d'une source, de créer un seul mot de code qui soit associé à une séquence de longueur arbitraire de symboles. Celui-ci diffère du code de Huffman qui attribue des mots de codes de longueurs variables à chaque symbole de la source. Le code associé à une séquence est un nombre réel de l'intervalle [0, 1[. Ce code est construit par des subdivisions récursives d'intervalles. Un intervalle est subdivisé pour chaque nouveau symbole qui appartient à la séquence. En définitive, on obtient un sous-intervalle de l'intervalle [0, 1[tel que tout nombre réel appartenant à cet intervalle représente la séquence à coder.

Le codage arithmétique code la séquence entière pour avoir le mot de code associé en sortie. Ce mot de code est un nombre entre 0 et 1. Comme le code de Huffman, le codage arithmétique est basé sur un algorithme à deux étapes. La première étape calcule la fréquence des symboles et produit une table de probabilité d'apparition de ces derniers. La deuxième étape réalise la compression. A la différence du codeur de Huffman, le

codage arithmétique permet de coder sur un nombre de bits pas nécessairement entier. Cet avantage permet de réaliser une compression plus performante. Cependant le codage arithmétique introduit une latence plus importante que le codage de Huffman car il n'est pas possible de commencer le décodage avant d'avoir obtenu la séquence entière des symboles transmis par la source. Les performances du codage arithmétique peuvent être améliorées dès lors que les tables de probabilités ne sont pas fixées, mais adaptées à la séquence courante et en fonction des séquences précédentes. On parle alors de codage arithmétique adaptatif [WNC87], [WNC98].

1.2.2 Compression avec perte

Les systèmes de compression avec perte traitent les sources réelles à valeurs continues. La compression avec perte (en anglais : lossy compression), par opposition à la compression sans perte, se permet d'éliminer quelques informations pour avoir le meilleur taux de compression possible, tout en gardant un résultat qui soit le plus proche possible des données originales. En contrepartie d'un taux de compression beaucoup plus intéressant, les données multimédias (audio, vidéo) peuvent tolérer un certain niveau de dégradation sans que les capteurs sensoriels (oeil, tympan, etc.) ne discernent une dégradation significative. Etant donné que ce type de compression supprime des informations contenues dans les données, on parle généralement de méthodes de compression irréversibles.

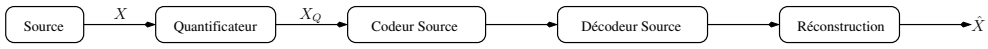

FIG. 1.3 – Système de compression des données avec perte.

La figure 1.3 illustre un système de compression avec perte d'une source X. $\hat{X}$ représente la version reconstituée de la source X. C'est la présence du quantificateur qui distingue un système de compression avec perte d'un système de compression sans perte. Le quantificateur consiste à convertir toute variable aléatoire continue X en une autre variable aléatoire discrète X_Q. Après chaque quantification, des erreurs (appelées erreurs ou bruit de quantification) considérées comme des signaux indésirables peuvent être ajoutées au signal utile. Ces signaux représentent les distorsions entre X et $\hat{X}$. L'efficacité du quantificateur se mesure alors à la qualité de restitution de la source.

Un quantificateur est optimal s'il présente des performances maximales. Celles-ci sont de plusieurs types :

1. Meilleur taux de compression,
2. Distorsion minimale. La mesure de distorsion la plus couramment utilisée est l'Erreur Quadratique Moyenne (EQM) : $D = E((X - \hat{X})^2)$,
3. Complexité minimale.

Le quantificateur utilisé dans un système de compression avec perte peut être un des trois types : un quantificateur scalaire ou un quantificateur vectoriel ou encore un quantificateur codé en treillis. Le quantificateur scalaire traite chaque variable aléatoire

X_i indépendamment des précédentes. Par contre, le quantificateur vectoriel et celui basé sur les treillis de codes tiennent compte de la corrélation qui peut exister entre les variables aléatoires successives. Si les performances des quantificateurs vectoriels sont supérieures à celles de leurs homologues scalaires lorsque des dégradations sont tolérées, il est manifeste que leur mise en œuvre conduit à des délais de calcul importants et à une complexité plus grande. Un résumé des nombreuses techniques de quantification est présenté dans [GN98].

1.2.2.1 Quantification scalaire

Considérons une source réelle X sans mémoire ($i.i.d$) prenant des valeurs dans l'intervalle $[a, b]$, avec $f_X(x)$ sa fonction de densité de probabilité. X est quantifiée en utilisant un quantificateur scalaire. La sortie de quantification X_Q prend ses valeurs dans un ensemble fini de L éléments.

Le quantificateur scalaire est défini par :

1. (L+1) niveaux de décision ou bien les extrémités des intervalles de quantification : $x_0, x_1, ..., x_L$ avec $x_0 = a \leqslant x_1 \leqslant ... \leqslant x_L = b$,
2. L valeurs quantifiées : $X_{Q_1} = y_1, X_{Q_2} = y_2, ..., X_{Q_L} = y_L$ avec $y_i \in [x_{i-1}, x_i]$ pour $i = 1, 2..., L$. L représente aussi le nombre de niveaux de quantification.

Le quantificateur scalaire associe à toute valeur X comprise dans l'intervalle $[x_{i-1}, x_i]$, une valeur quantifiée X_{Q_i} située dans cet intervalle. Ce procédé permet d'attribuer à chaque X un mot binaire unique, choisi parmi les L que contient un dictionnaire. Ainsi, par exemple, lorsque le nombre de valeurs quantifiées est une puissance de 2, on aura $L = 2^R$ et chaque valeur quantifiée peut être représentée par un mot de R bits. Dans ce cas, le taux de compression de la source X est R bits/symbole.

La distorsion introduite par un quantificateur scalaire s'exprime par :

$$D = \sum_{j=1}^{L} \int_{x_{i-1}}^{x_i} (x - y_i)^2 f_X(x) dx \tag{1.12}$$

Quantificateur scalaire uniforme :

Le quantificateur le plus simple d'un point de vue complexité de l'algorithme et difficulté de l'implémentation est le quantificateur scalaire uniforme. Dans ce cas, les niveaux de décision sont uniformément distribués et la valeur quantifiée X_{Q_i} est choisie au milieu de l'intervalle $[x_{i-1}, x_i]$: $X_{Q_i} = y_i = \frac{x_i - x_{i-1}}{2}$. La différence entre deux niveaux de quantification successifs est appelée "pas de quantification" $\Delta = \frac{x_L - x_0}{L}$. La valeur du pas de quantification Δ a une influence directe sur le débit et la distorsion (EQM) de la source.

Au lieu de chercher à minimiser la distorsion D d'un système de compression avec perte, il est parfois avantageux de chercher à maximiser le critère de SNR, (Signal to Noise Ratio) défini par :

$$SNR \triangleq 10 \log_{10}(\frac{\sigma_X^2}{D}) \tag{1.13}$$

avec σ_X^2 la variance de la source X à compresser.

Nous utilisons dans la partie codage vidéo distribué (chapitre 6) un quantificateur scalaire à zone morte (dead-zone). Ce quantificateur emploie un intervalle de décision autour de zéro (appelé zone morte) dont la taille est un multiple de la taille des autres intervalles de quantification. Généralement, la taille de la zone morte est le double de la taille des autres intervalles. Ce quantificateur quantifie à zéro toutes les valeurs les plus significatives dans une zone morte autour de l'origine, permettant ainsi d'améliorer le compromis débit-distorsion [GN98].

Quantificateur scalaire de Lloyd-Max : L'algorithme de Lloyd-Max [Llo82], [Max60] s'adapte à la source en cherchant à trouver d'une manière itérative les L valeurs quantifiées et les $L+1$ niveaux de décisions dans l'intervalle $[a,b]$ avec une distorsion donnée par (1.12) réduite au minimum.

Pour tout $i = 1, 2, ..., L$, la minimisation de la distorsion du quantificateur de Lloyd-Max (quantification optimale) relativement à y_i et x_i est obtenue par annulation des dérivées partielles de D, $\frac{\partial D}{\partial y_i} = 0$ et $\frac{\partial D}{\partial x_i} = 0$, respectivement. A partir de $\frac{\partial D}{\partial y_i} = 0$, on obtient la première égalité :

$$\begin{aligned} \frac{\partial D}{\partial y_i} &= \frac{\partial}{\partial y_i} \sum_{j=1}^{L} \int_{x_{j-1}}^{x_j} (x - y_j)^2 f_X(x) dx \\ &= \int_{x_{i-1}}^{x_i} (-2)(x - y_i) f_X(x) dx = 0, \quad i = 1, 2, ..., L. \end{aligned} \tag{1.14}$$

(1.14) peut s'écrire comme :

$$y_i = \frac{\int_{x_{i-1}}^{x_i} x f_X(x) dx}{\int_{x_{i-1}}^{x_i} f_X(x) dx} \tag{1.15}$$

Avec la dérivée partielle de D relativement à x_i, une deuxième égalité peut être déterminée :

$$\begin{aligned} \frac{\partial D}{\partial x_i} &= \frac{\partial}{\partial x_i} \sum_{j=1}^{L} \int_{x_{j-1}}^{x_j} (x - y_j)^2 f_X(x) dx \\ &= (x_i - y_i)^2 f_X(x_i) - (x_i - y_{i+1})^2 f_X(x_i) = 0, \quad i = 1, 2, ..., L. \end{aligned} \tag{1.16}$$

soit

$$x_i = \frac{y_i + y_{i+1}}{2}, \quad i = 1, 2, ..., L. \tag{1.17}$$

Initialement, l'algorithme de Lloyd-Max utilise un dictionnaire de valeurs de reconstruction y_i d'un quantificateur uniforme (le plus souvent). Puis à chaque itération, l'algorithme de Lloyd-Max génère un nouveau quantificateur en utilisant les deux équations (1.15) et (1.17). La distorsion obtenue avec le nouveau quantificateur est plus faible ou égale à celle du quantificateur de l'itération précédente. Si la distorsion ne

décroît plus, le processus d'itération sera arrêté.

La quantification scalaire a de nombreuses limites : elle ne peut pas atteindre les limites théoriques en terme de débit-distorsion, et plus généralement elle ne tient pas compte des corrélations qui peuvent exister entre les différents échantillons d'une source. Cependant, cette technique est quasi-instantanée et demande très peu de temps de calculs et d'espace de stockage. En s'autorisant un délai plus élevé pour disposer de plusieurs échantillons à la fois et opérant avec la mémoire de ces derniers, la quantification vectorielle permet de s'approcher davantage de la limite de la courbe débit-distorsion $R(D)$.

1.2.2.2 Quantification vectorielle

La quantification vectorielle [Gra84], [GG92] consiste à représenter tout vecteur $\mathbf{x} = (x_1, x_2, ..., x_n)$ de l'espace $\Re^n$ (espace de variables à valeurs continues et de dimension n) par un autre vecteur $\mathbf{y}_i$ $(i = 1, 2, ..., N)$ de même dimension appartenant à un dictionnaire qui est un ensemble fini Υ de N vecteurs codes. Mathématiquement, on parle d'un quantificateur vectoriel de dimension n et de taille N défini comme une application $\mathcal{Q}$ de $\Re^n$ vers Υ. $\mathcal{Q}$ permet de partitionner l'espace $\Re^n$ en N régions (appelées aussi régions de Voronoï) $\mathcal{R}_i$ $(i = 1, 2, ..., N))$ définies par :

$$\mathcal{R}_i = \{\mathbf{x} \in \Re^n / \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i, \text{ si } ||\mathbf{x} - \mathbf{y}_i||^2 \leq ||\mathbf{x} - \mathbf{y}_j||^2, \text{ pour tout } i \neq j\} \tag{1.18}$$

avec $||\mathbf{x} - \mathbf{y}_i||^2$ la distance euclidienne au carré entre $\mathbf{x}$ et $\mathbf{y}_i$ définie par :

$$||\mathbf{x} - \mathbf{y}_i||^2 = \sum_{j=1}^{n} (x(j) - y_i(j))^2 \tag{1.19}$$

Tous les vecteurs $\mathbf{x}$ appartenant à la région $\mathcal{R}_i$ sont représentés par le même vecteur code $\mathbf{y}_i$. Par la suite, $||\mathbf{x} - \mathbf{y}_i||^2$ représente aussi la distorsion mesurée entre les vecteurs $\mathbf{x}$ et $\mathbf{y}_i$.

Pour tout vecteur $\mathbf{x}$, un quantificateur vectoriel cherche dans le dictionnaire Υ le vecteur code le plus proche en terme de distance euclidienne. Le débit binaire R d'un quantificateur vectoriel est défini par :

$$R = \frac{1}{n} \log_2 N \tag{1.20}$$

Le quantificateur scalaire peut être vu comme une forme particulière d'un quantificateur vectoriel de dimension 1. Les performances en terme de distorsion (des vecteurs mieux reconstruits) obtenues avec une quantification vectorielle de dimension supérieure à 1 sont meilleures que celles obtenues avec une quantification scalaire. Ce gain est dû à l'exploitation de la mémoire de la source (corrélation entre les différentes composantes des vecteurs de la source). En contre partie la quantification vectorielle présente une

complexité et des délais de calculs plus importants. Les gains et les complexités des quantificateurs vectoriels augmentent avec la dimension n.

Un quantificateur dont le dictionnaire forme un réseau régulier de points dans un espace euclidien est aussi un quantificateur vectoriel. Ce réseau est désigné par le terme lattice et le quantificateur vectoriel est appelé quantificateur sur lattice et noté Λ. Dans l'espace $\mathbb{R}^n$, soit l'empilement régulier de sphères identiques de rayon r. Un quantificateur sur lattice est constitué de l'ensemble des centres de ces sphères. Tout vecteur dans $\mathbb{R}^n$ peut être représenté par la base constituée par les n vecteurs $\mathbf{c}_i = \{c_{i1}, c_{i2}, ..., c_{in}^T\}$ qui sont les centres des sphères. Un quantificateur sur lattice Λ est défini comme étant l'ensemble des points $\mathbf{l} \in \mathbb{R}^n$ tels que :

$$\Lambda = \{\mathbf{l} \in \mathbb{R}^n | \exists \mathbf{i} \in \mathbb{Z}^n, \mathbf{l} = \mathbf{G}^T \mathbf{i}\}$$

avec $\mathbf{G}$ la matrice génératrice du quantificateur sur lattice Λ définie par :

$$\mathbf{G} = (\mathbf{c}_1^T, \mathbf{c}_2^T, ..., \mathbf{c}_n^T)^T$$

Les quantificateurs sur lattice sont souhaitables [CS82, CS93], parce que :

1. il y a des algorithmes rapides pour les mettre en œuvre, et
2. pour des taux de compression élevés, la distorsion générée est minimale avec des sources sans mémoire dont laquelle les données ont une distribution uniforme.

Pour une dimension égale à 2, la lattice hexagonale est le meilleur empilement pour avoir la plus faible distorsion d'un quantificateur sur lattice (figure 1.4).

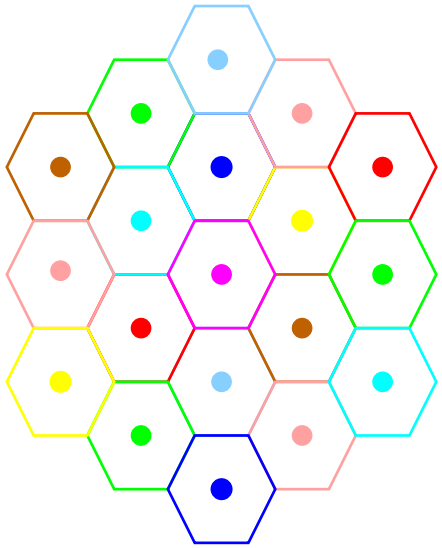

FIG. 1.4 – La lattice hexagonale.

1.2.2.3 Quantification codée en treillis

Considérons un graphe $\mathbb{G}$ satisfaisant les propriétés suivantes :

1. il y a un nombre fixe de puissance 2 (soit ce nombre égale à 2^j) de branches orientées sortantes de chaque sommet de $\mathbb{G}$.
2. chaque branche de $\mathbb{G}$ a une étiquette de symbole d'un alphabet de données A.
3. le graphe $\mathbb{G}$ est relié (pour deux sommets donnés de $\mathbb{G}$, il y a un chemin les reliant)

Soit s un sommet de $\mathbb{G}$. Soient n et m deux entiers positifs tels que n est un multiple de m. Considérons l'ensemble de chemins $\mathcal{P}$ du graphe $\mathbb{G}$ débutant du sommet s et contenant n/m branches. Chaque chemin dans $\mathcal{P}$ est caractérisé par une séquence $\mathbf{y} = (y_1, y_2, ..., y_n)$ de longueur n dont les symboles sont les données de l'alphabet A. L'ordre des symboles dans chaque séquence correspond à celui des branches qui sont visitées par un chemin donné. Soit $R = j/m$. Dans ce cas, il y a 2^{nR} chemins différents dans $\mathcal{P}$ qui peuvent être identifiés uniquement par nR symboles.

Un treillis est par définition une représentation graphique de tous les chemins de $\mathcal{P}$. Etant donnée une séquence $\mathbf{x} = (x_1, x_2, ..., x_n)$ de l'espace A^n (espace d'alphabet de données A et de dimension n), le quantificateur codé en treillis cherche à trouver le meilleur chemin dans $\mathcal{P}$ dont la séquence $(y_1, y_2, ..., y_n)$ permet de réduire au maximum la distorsion globale :

$$D = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{1.21}$$

La sortie du quantificateur est caractérisée par les mots de codes binaires de longueur nR correspondant au chemin minimal de $\mathcal{P}$ et la séquence $(y_1, y_2, ..., y_n)$. Pour retrouver le chemin minimal sélectionné au quantificateur, le treillis du graphe de tous les chemins de $\mathcal{P}$ utilise la séquence de mots de codes de longueur nR. La séquence reconstruite sera celle de $(y_1, y_2, ..., y_n)$ du chemin retrouvé. Le taux de compression de ce quantificateur est R bits par symbole. Pour la recherche du meilleur chemin, l'algorithme de Viterbi est le plus souvent utilisé.

Exemple : Soit le graphe $\mathbb{G}$ de la figure 1.5 formé par deux sommets et 4 branches. La dimension des étiquettes sur les branches est 2 ($m = 2$). Considérons la séquence $\mathbf{x} = (0, 1, 0, 0, -1, 0)$ de longueur $n = 6$ à quantifier avec un quantificateur codé en treillis en utilisant le graphe $\mathbb{G}$. Le taux de compression est $R = j/m = 0.5$ bit par symbole. Considérons le sommet s_0 comme le point de départ pour quantifier la séquence $\mathbf{x}$. L'ensemble $\mathcal{P}$ dans $\mathbb{G}$ peut avoir $2^{nR} = 8$ chemins possibles commençant du sommet s_0 et de longueur $n/m = 3$ branches. La figure 1.6 illustre le treillis des 8 chemins possibles de l'ensemble $\mathcal{P}$.

Chaque paire d'éléments de la séquence $\mathbf{x}$ ($m = 2$) est quantifiée en utilisant le treillis de la figure 1.6. En tout, il y a trois instants correspondant à $n/m = 3$. Au premier instant du treillis, on calcule la distance euclidienne au carré entre les étiquettes des branches correspondantes et le message à quantifier $(0, 1)$, (les deux premiers éléments

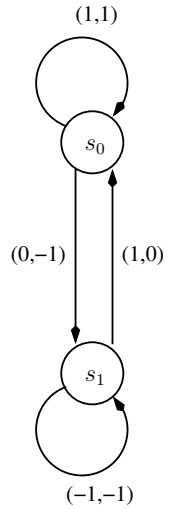

FIG. 1.5 – Graphe $\mathbb{G}$.

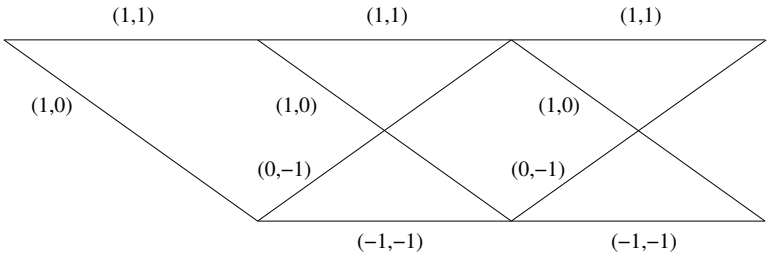

FIG. 1.6 – Treillis des chemins de l'ensemble $\mathcal{P}$.

de x). Au deuxième et troisième instants le même calcul est réalisé pour quantifier respectivement $(0,0)$ et $(-1,0)$. La figure 1.7 présente le nouveau treillis des chemins de $\mathcal{P}$ tenant compte des poids de distance euclidienne.

Le meilleur chemin dans le treillis de la figure 1.7 est celui dont la somme des poids est la plus faible. Ce chemin (marqué par des traits larges) est constitué par les branches de transition entre les sommets s_0 et s_1 comme l'indique la figure 1.8.

Sur les étiquettes de branches du treillis de la figure 1.8, on a rajouté les mots binaires correspondants (de 1 bit de longueur). Par conséquent, la séquence de mots de code résultante après la quantification de $\mathbf{x}$ est $(0,1,1)$.

Le quantificateur basé sur des treillis de codes est considéré comme un quantificateur vectoriel algébrique. Marcellin et Fischer [MF90] ont développé une classe importante de quantificateur basé sur des treillis de codes appelée quantificateur codé par treillis (en anglais TCQ pour Trellis Coded Quantization). Dans le chapitre 3, nous présentons la TCQ et nous proposons un schéma de codage de sources distribuées utilisant cette dernière.

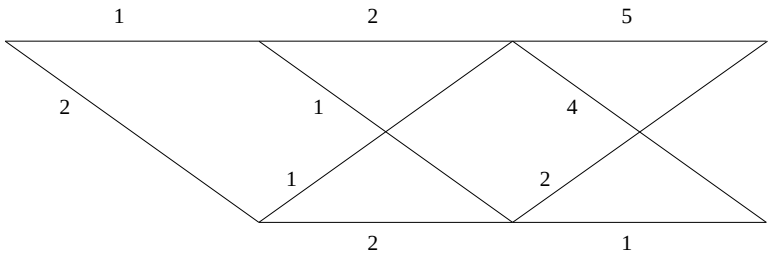

FIG. 1.7 – Poids en terme de distance euclidienne de tous les chemins de $\mathcal{P}$.

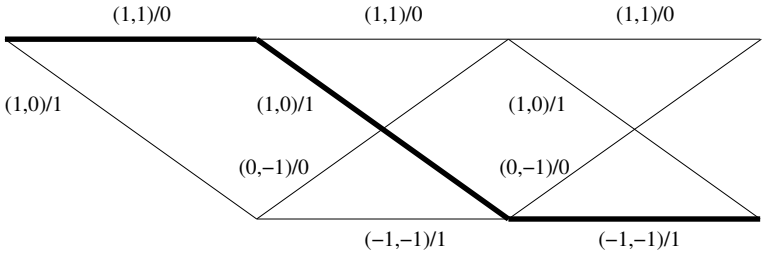

FIG. 1.8 – Le meilleur chemin de l'ensemble $\mathcal{P}$ dans le treillis.

1.2.2.4 Débit-distorsion

La théorie débit-distorsion a été développée pour des sources avec ou sans mémoire. Cette théorie peut être appliquée pour des sources à variables continues ou discrètes.

Dans la compression avec perte, les données décompressées ne sont pas forcément identiques à celles des données originales. Souvent, il suffit d'avoir une approximation raisonnablement proche. La mesure de distorsion est une entité mathématique qui indique exactement les valeurs d'approximations voulues. Généralement, c'est une fonction qui assigne à deux lettres X et $\hat{X}$ dans l'alphabet Ω un nombre non négatif dénoté par :

$$d(X, \hat{X}) \geq 0 \tag{1.22}$$

où X désigne les données originales, $\hat{X}$ son approximation, et $d(X, \hat{X})$ est la mesure de distorsion entre X et $\hat{X}$.

Les mesures de distorsion les plus connues sont :

1. la mesure de distorsion de Hamming (utilisée pour les sources discrètes) définie par :

$$d(X, \hat{X}) = \begin{cases} 0 & \text{si } X = \hat{X} \\ 1 & \text{si } X \neq \hat{X} \end{cases} \tag{1.23}$$

2. la mesure de l'erreur quadratique de la distorsion (qui est seulement applicable quand Ω est un ensemble de nombres). C'est la mesure la plus utilisée pour des alphabets à valeurs continues. Elle est définie par :

$$d(X, \hat{X}) = (X - \hat{X})^2 \tag{1.24}$$

En pratique (codage vidéo), les mesures de qualité sont généralement exprimées en fonction du rapport signal crête sur bruit (en anglais Peak Signal to Noise Ratio : $PSNR$), soit, dans le cas d'un signal représenté sur 8 bits :

$$PSNR = 10 * \log_{10} \frac{(2^8 - 1)^2}{d(X, \hat{X})} \tag{1.25}$$

La théorie débit-distorsion indique que pour une source et une mesure de distorsion donnée ($d(X, \hat{X}) \leq D$), il existe une fonction $R(D)$, appelée la fonction débit-distorsion, définie par :

$$R(D) = \min_{d(X,\hat{X}) \leq D} I(X; \hat{X}) \tag{1.26}$$

D'après le théorème de Shannon, la fonction $R(D)$ est le débit minimal d'un système de compression avec une distorsion D. La figure 1.9 illustre l'allure du débit R en fonction de la distorsion D. C'est la limite inférieure du débit. Deux régions du couple débit-distorsion (R, D) peuvent être observées. Une première région située au-dessus de la limite de R qui correspond aux systèmes de compression réalisables en pratique (performances atteignables). L'objectif est donc de s'approcher au maximum de cette limite. Quant à la deuxième région située au-dessous de la courbe de la figure 1.9, elle présente le couple (R, D) non admissible : c'est-à-dire aucun système de compression ne peut avoir une distorsion D et un débit inférieur à R.

La fonction débit-distorsion vérifie les propriétés suivantes :

1. $R(0) = +\infty$.
2. $R(D) \geq 0$ est décroissante quand la distorsion augmente.
3. $R(D)$ est une fonction continue et concave de D.

Pour une source binaire (source discrète), avec un modèle de corrélation entre X et $\hat{X}$ et un canal binaire symétrique caractérisé par une probabilité de transition p, la fonction débit-distorsion s'exprime par [CT91] :

$$R(D) = \begin{cases} h(p) - h(D), & \text{si } 0 \leq D \leq \min\{p, 1-p\}, \\ 0, & \text{si } D \geq \min\{p, 1-p\}. \end{cases} \tag{1.27}$$

avec $h(x) = -x \log_2(x) - (1-x) \log_2(1-x)$, pour toute valeur de x entre 0 et 1.

Dans le cas d'une source Gaussienne X *i.i.d* de moyenne nulle et de variance σ^2 ($X \sim \mathcal{N}(0, \sigma^2)$) avec une distorsion mesurée en terme de EQM, la fonction $R(D)$ est donnée par [CT91] :

$$R(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma^2}{D}, & \text{si } 0 \leq D \leq \sigma^2, \\ 0, & \text{si } D \geq \sigma^2. \end{cases} \tag{1.28}$$

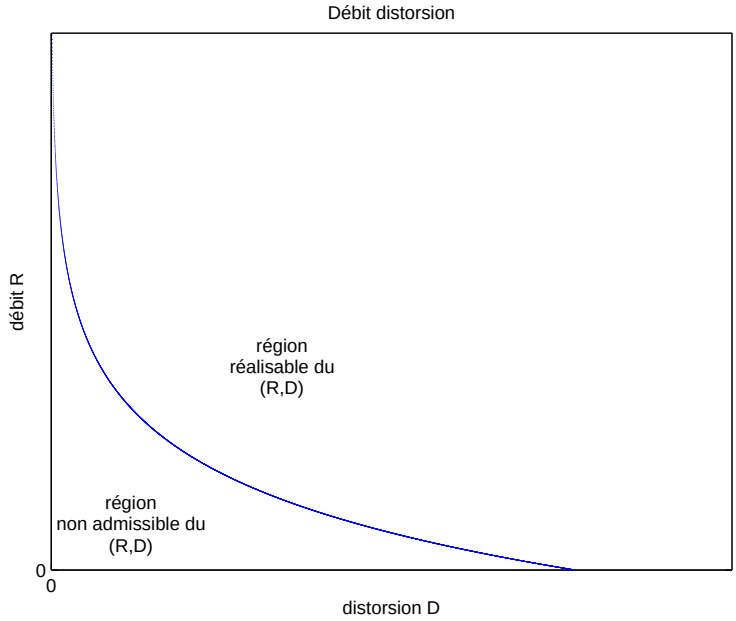

FIG. 1.9 – Région du couple débit-distorsion (R,D) admissible.

1.3 Technique de codage de canal

Le codage de canal adapte le signal, obtenu après le codage de source, au canal de transmission (figure 1.10). Ce type de codage réintroduit généralement une redondance efficace pour protéger les données contre les erreurs de transmission qui peuvent survenir sur le canal. En effet, l'algorithme de décodage de canal à la réception utilise cette redondance pour détecter, puis éventuellement, corriger les données erronées.

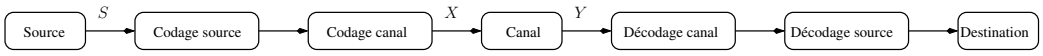

FIG. 1.10 – Schéma d'un système de communication.

1.3.1 Capacité de canal

En 1948, Shannon a montré dans un deuxième théorème qu'il est théoriquement possible de transmettre l'information à travers un canal bruité avec une probabilité d'erreur arbitrairement faible à condition que le débit de la transmission soit inférieur à la capacité du canal C [Sha48].

Pour un canal dont l'ensemble des entrées est X et l'ensemble des sorties est Y (figure 1.10), la capacité C (exprimée en bits/seconde) est donnée par le maximum de

l'information mutuelle entre X et Y pour toutes les lois possibles X, soit

$$C = \max_{p(X)} I(X;Y) \tag{1.29}$$

En particulier, pour un canal binaire symétrique caractérisé par sa probabilité de transition p, la capacité est $C = 1 - h(p)$. A titre formel, la capacité d'un canal à bruit blanc gaussien additif (AWGN) de densité de puissance $N_0/2$ s'exprime par :

$$C = W \log_2(1 + \frac{P}{N_0 W}) \tag{1.30}$$

où W est la largeur de bande du canal exprimée en Hertz et P est la puissance moyenne du signal dans un intervalle de signalisation de durée T secondes.

La fonction de correction d'erreurs (ou décodage canal) est devenue essentielle dans la conception des systèmes de télécommunication. Le but de la recherche dans le domaine du codage de canal est de trouver des codes correcteurs approchant les performances de la limite théorique de Shannon (fixée par la capacité C) avec une complexité faible. En fait, il existe deux grandes familles de codes correcteurs d'erreurs. La première est celle des codes en blocs où une séquence de k bits d'information est codée dans un bloc de n symboles ($n > k$). La deuxième est celle des codes convolutionnels utilisés pour la transmission des symboles d'information qui arrivent en série sans structures en blocs. Nous présentons les deux meilleurs codes correcteurs (permettent d'obtenir des performances proches de la limite de Shannon) à savoir les codes turbo et les codes LDPC (Low Density Parity Check).

1.3.2 Les codes turbo

Le code turbo découvert par Berrou *et al.* [BGT93], [BG96] est constitué de deux codeurs convolutionnels récursifs systématiques (CRS) concaténés en parallèle et séparés par un entrelaceur. Le décodage s'effectue d'une manière itérative. Le décodeur turbo est composé de deux décodeurs SISO (Soft-In Soft-Out) concaténés en série via un entrelaceur/désentrelaceur. Le décodage est basé sur l'échange d'information extrinsèque d'un décodeur à l'autre jusqu'à l'obtention des performances désirées. Chaque décodeur calcule les probabilités a posteriori (A Posteriori Probability : APP) pour estimer le bit d'information. L'algorithme de décodage utilisé dans chaque SISO pour calculer les APP est celui de Bahl *et al.* [BCJR74] appelé algorithme BCJR (Bahl, Cocke, Jelinek, et Raviv) ou bien MAP (Maximum A Posteriori). Nous présentons dans l'annexe A, la dérivation des équations de l'algorithme MAP.

Les codes turbo sont devenus très importants pour la protection de données transmises par blocs courts (Internet, CDMA2000, 3GPP, UMTS entre autres). Pour d'autres applications (espace lointain, télédiffusion, ...), les codes turbo offrent également les meilleures performances. Différentes propriétés des codes turbo sont fournies dans le livre [VY00].

1.3.3 Les codes LDPC

Les codes LDPC ont été découverts en 1962 par Gallager [Gal62], [Gal63]. Pour plus de détails, nous invitons le lecteur à consulter le livre [Gal69].

Les codes LDPC sont des codes en blocs (n, k), décrits par leur matrice de parité $\mathbf{H}$ et/ou le graphe associé à $\mathbf{H}$ qui est appelé en anglais "bipartite". La matrice de parité $\mathbf{H}$ d'un code LDPC binaire est caractérisée par un nombre faible de 1 (d'où le nom faible densité). La manière dont les 1 sont répartis dans la matrice $\mathbf{H}$ est décrite par les degrés des polynômes $\tilde{\lambda}(x)$ et $\tilde{\rho}(x)$. Les polynômes $\tilde{\lambda}(x)$ et $\tilde{\rho}(x)$ indiquent respectivement le pourcentage des colonnes et des lignes dans la matrice $\mathbf{H}$ avec différents poids de Hamming (nombre de 1). Quand $\tilde{\lambda}(x)$ et $\tilde{\rho}(x)$ ont seulement un terme, le code LDPC est dit régulier, sinon il est irrégulier. A partir de la matrice de parité $\mathbf{H}$, on peut aussi déduire si le code est régulier ou non. Si le nombre de 1 dans chaque ligne (ou chaque colonne) n'est pas le même alors le code est irrégulier. En général, on s'attend à ce qu'un code LDPC irrégulier optimisé soit plus performant qu'un code régulier pour les mêmes taux et longueur de mots de code. Étant donnés $\tilde{\lambda}(x)$ et $\tilde{\rho}(x)$, le taux du code LDPC peut être déterminé. Cependant, plusieurs codes peuvent être utilisés suivant la matrice $\mathbf{H}$.

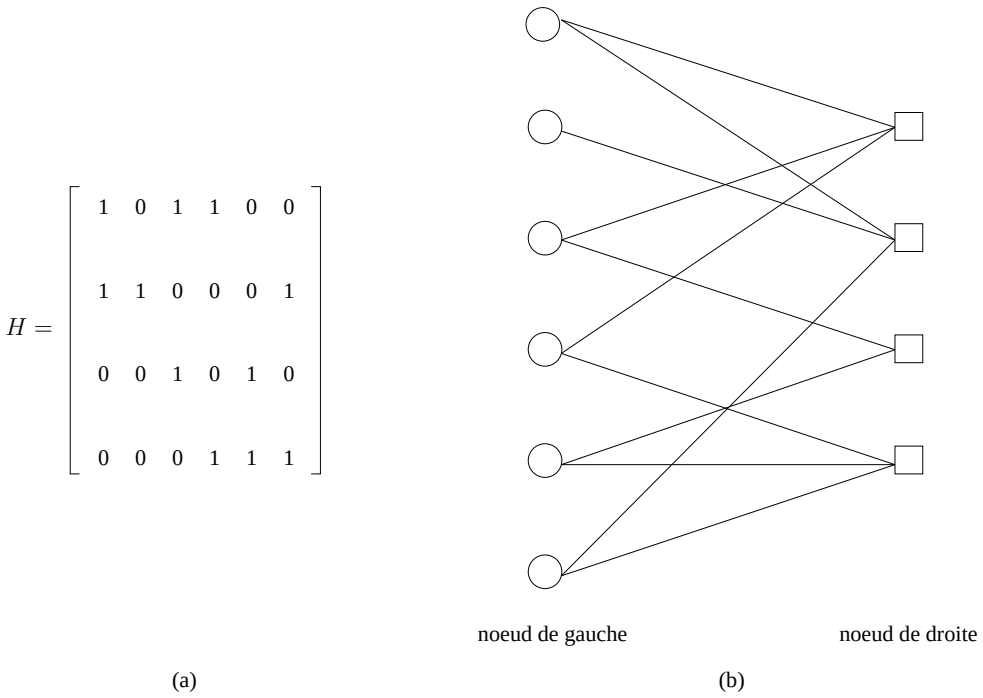

FIG. 1.11 – Code LDPC (6,2) : (a) matrice de parité $\mathbf{H}$, (b) Graphe "bipartite".

Le graphique "bipartite" d'un code LDPC est une représentation équivalente de la matrice de parité $\mathbf{H}$. Chaque colonne est représentée par un nœud à gauche (désigné par nœud de gauche ou "bit node") et chaque ligne par un nœud à droite (désigné par

nœud de droite ou "check node") dans le graphe "bipartite". Tous les nœuds de gauche sont mis dans une colonne. Quant aux nœuds de droite, ils sont mis dans une autre colonne parallèle à la première (celle de gauche). Pour chaque 1 dans la matrice de parité **H**, une branche reliera un nœud de gauche avec son correspondant de droite. On désigne par degré d'un nœud le nombre de branches connectées à ce dernier. Le graphique "bipartite" d'un code LDPC est utilisé dans le processus de décodage. Les codes LDPC sont actuellement les codes les plus proches de la borne de Shannon pour un canal gaussien [MN96], [CFRU01]. Un exemple d'un code LDPC (6,2) avec sa matrice de parité et le graphe "bipartite" correspondant est illustré à la figure 1.11.

1.4 Théorème de Slepian-Wolf

Dans la théorie de l'information introduite par Shannon, il est connu que le taux minimal pour compresser sans perte deux sources corrélées et discrètes X et Y est l'entropie conjointe $H(X,Y)$. Pour s'approcher de ce taux, les deux sources sont codées et décodées conjointement. En 1973, Slepian-Wolf [SW73] ont étendu ce résultat au cas des sources corrélées qui sont codées indépendamment mais décodées conjointement.

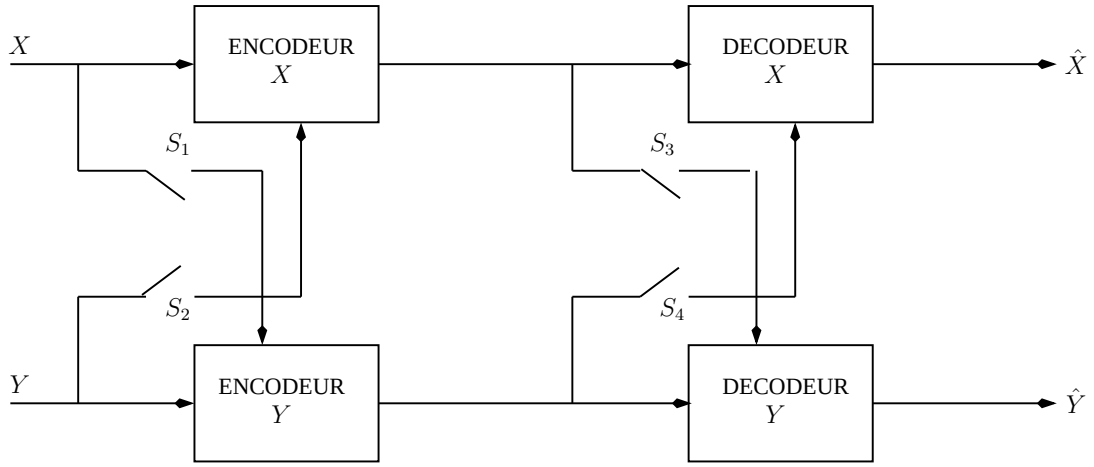

FIG. 1.12 – Codage de deux sources corrélées.

Considérons deux sources corrélées et discrètes *i.i.d.* notées X et Y. Sur la figure 1.12, on présente les 16 cas possibles pour les coder et décoder selon les états des interrupteurs S_1,S_2,S_3 et S_4 désignés par le quadruple $S_1S_2S_3S_4$. Si l'interrupteur est ouvert, le symbole S_i, $i = 1,2,3,4$, sera à 0 sinon à 1. Le tableau 1.1 présenté dans [SW73], résume les théorèmes de Slepian-Wolf correspondants à chacun des 16 cas. L'interrupteur S_i =x indique qu'il est possible d'avoir deux états : ouvert ou fermé (0 ou 1). En association avec le tableau 1.1, la figure 1.13 illustre les régions de débits minimums admissibles $\mathcal{R}$ suivant l'état des interrupteurs S_i, $i = 1,2,3,4$ et donne les limites théoriques des théorèmes de Slepian-Wolf. Sur cette figure certaines lignes et

certains points sont marqués avec les noms des théorèmes du tableau 1.1. Les états des interrupteurs correspondants sont apposés. Avec les théorèmes f et g du tableau 1.1, ces points et ces lignes peuvent être employés pour déterminer la frontière de la région $\mathcal{R}$ pour n'importe quel cas (les 16 cas).

TAB. 1.1 – *Les débits minimums admissibles selon le théorème de Slepian-Wolf*

$S_1S_2S_3S_4$ selon la figure 1.12	Point de référence selon la figure 1.13	Débits théoriques atteignables
		il est nécessaire que
0xxx	A	$R_X \geq H(X\|Y)$
x0xx	B	$R_Y \geq H(Y\|X)$
xx0x	C	$R_Y \geq H(Y)$
xxx0	D	$R_X \geq H(X)$
xxxx	E	$R_X + R_Y \geq H(X,Y)$
		il est suffisant que
1xx1	a	$R_X = 0\ R_Y = H(X,Y) + \xi_{xy}$
x11x	b	$R_X = H(X,Y) + \xi_{xy}\ R_Y = 0$
xx1x	c	$R_X = H(X) + \xi_x\ R_Y = H(Y\|X) + \xi_y$
xxx1	d	$R_X = H(X\|Y) + \xi_x\ R_Y = H(Y) + \xi_y$
xxxx	e	$R_X = H(X) + \xi_x\ R_Y = H(Y) + \xi_y$
		$\xi_x, \xi_y, \xi_{xy} > 0$
		bit de bourrage
xxxx	f	$(R_X, R_Y) \in \mathcal{R} \Rightarrow (R_X + \delta_x, R_Y + \delta_y) \in \mathcal{R}$
		$\delta_x, \delta_y \geq 0$
		partage du temps
xxxx	g	$(R_X, R_Y) \in \mathcal{R}$, $(R'_X, R'_Y) \in \mathcal{R}$ et
		$R_X + R_Y = H(X,Y)$ et $R'_X + R'_Y = H(X,Y)$
		donc $(R''_X, R''_Y) \in \mathcal{R}$,
		$R''_X = \lambda R_X + (1-\lambda)R'_X$,
		$R''_Y = \lambda R_Y + (1-\lambda)R'_Y$, $0 \leq \lambda \leq 1$

Pour le cas du quadruple 1011 et suivant les états du tableau 1.1, les théorèmes B E a c d e peuvent être appliqués. Avec B et E, la région des débits minimums admissibles $\mathcal{R}$ ne peut pas inclure les zones au-dessous des lignes marquées par B et E sur la figure 1.13. Les quatre points a, c, d et e sont dans la région $\mathcal{R}$. Le théorème f montre que les points au-dessus de a sur l'axe R_Y sont dans la région $\mathcal{R}$. Finalement, selon le théorème g, le segment [a c] de la ligne reliée par les points a et c fait partie de la région $\mathcal{R}$. La figure 1.14 illustre la région $\mathcal{R}$ correspondante au cas 1011.

Considérant maintenant, le cas 0011 qui correspond au système où les deux sources sont codées indépendamment mais décodées conjointement (les deux sources X et Y sont

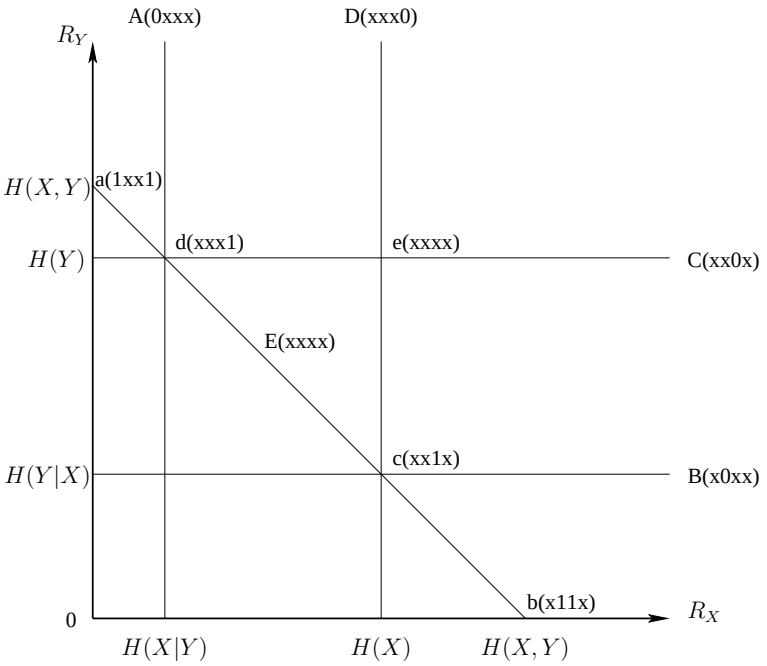

FIG. 1.13 – Les régions des débits minimums suivant le tableau 1.1.

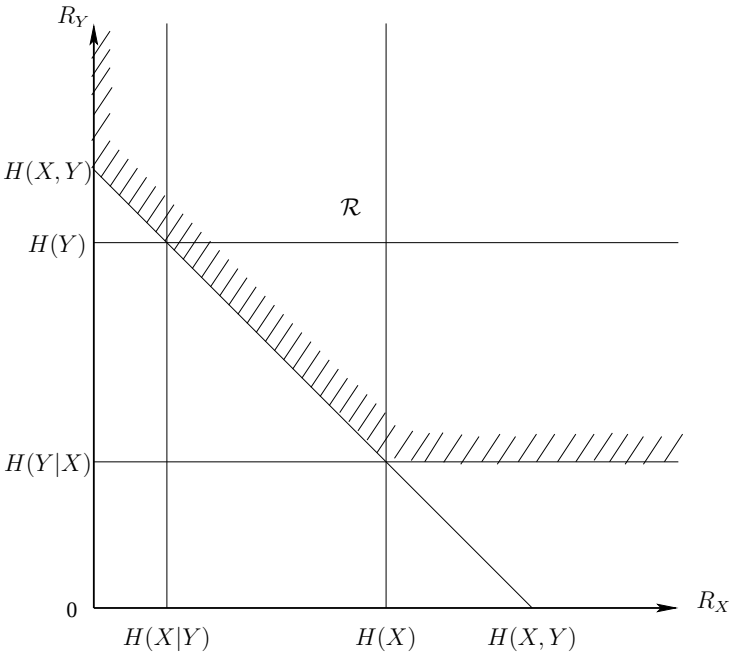

FIG. 1.14 – Région des débits minimums du cas 1011.

disponibles aux deux décodeurs). Pour reconstruire X et Y avec un taux d'erreur par bit quasi-nul, le théorème de Slepian-Wolf montre que le taux minimal ($R = R_X + R_Y$) pour compresser les deux sources est l'entropie conjointe $H(X,Y)$ avec :

$$R_X \geq H(X|Y) \tag{1.31}$$
$$R_Y \geq H(Y|X) \tag{1.32}$$
$$R_X + R_Y \geq H(X,Y) \tag{1.33}$$

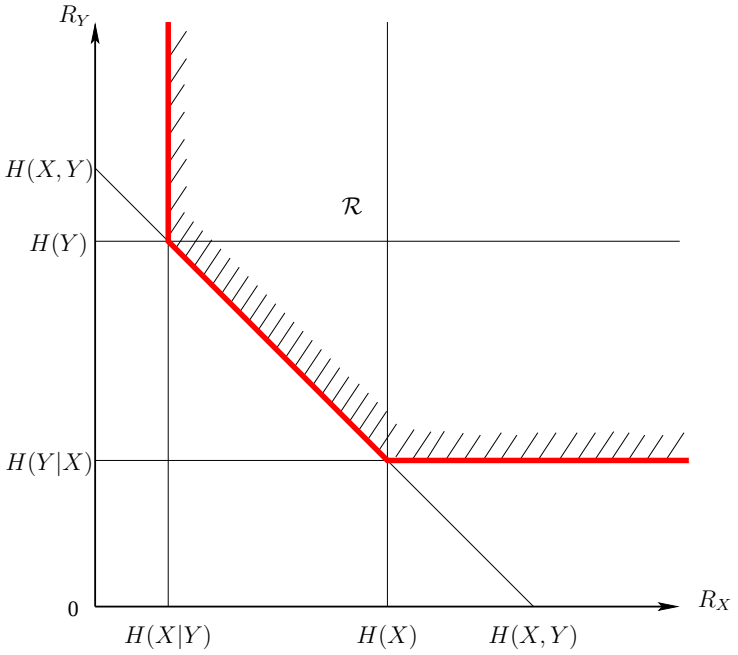

FIG. 1.15 – Région des débits minimums du cas 0011.

La région des débits minimums admissibles $\mathcal{R}$ du cas 0011 peut être déterminée selon le tableau 1.1 par les théorèmes A B E c d e. Par conséquent, la région $\mathcal{R}$ ne peut pas contenir les zones à gauche de la ligne A ni celles au-dessous des lignes B et E sur la figure 1.13. Les points c, d et e sont inclus dans $\mathcal{R}$. Selon le théorème g, le segment [a c] de la figure 1.13 fait partie de la région $\mathcal{R}$. La figure 1.15 montre la région des débits minimums admissibles $\mathcal{R}$ pour laquelle deux sources X et Y sont codées séparément et décodées conjointement. Les lignes rouges verticales, horizontales et diagonales correspondent respectivement, aux équations (1.31), (1.32) et (1.33) qui représentent les limites inférieures des débits minimums admissibles R_X et R_Y.

Prenons l'état $S_1 = 0$, $S_2 = 0$, $S_3 = 0$ et $S_4 = 1$ (c'est-à-dire le quadruple 0001) qui représente le système de codage distribué avec information de bord où les deux sources sont codées séparément mais décodées conjointement (Y est connue seulement par le décodeur X). Pour un taux d'erreur par bit quasi-nul, le théorème d de Slepian-Wolf

présenté au tableau 1.1 fixe le taux minimal $R = R_X + R_Y \geq H(X,Y)$ pour compresser d'une façon distribuée les deux sources X et Y avec :

- X doit être compressé au taux théorique de son entropie conditionnelle, c'est-à-dire $R_X = H(X|Y) + \xi_X$
- et, Y doit être transmis à $R_Y = H(Y) + \xi_Y$

où ξ_X et ξ_Y sont deux valeurs réelles positives. Suivant le tableau 1.1, les théorèmes A B C E d e déterminent la région des débits minimums atteignables $\mathcal{R}$ du cas 0001. Les zones à gauche de la ligne A et au-dessous de la ligne C de la figure 1.13 n'appartiennent pas à la région $\mathcal{R}$. Le point d correspondant à la limite théorique du taux de compression admissible est dans $\mathcal{R}$. Avec le théorème f, tous les points qui sont au-dessous de la ligne C et à droite de la ligne A appartiennent à la région $\mathcal{R}$. La figure 1.16 illustre la région des débits minimums atteignables $\mathcal{R}$ du système correspondant au quadruple 0001 (c'est-à-dire les deux sources sont codées indépendamment mais décodées conjointement avec seulement Y disponible aux deux décodeurs).

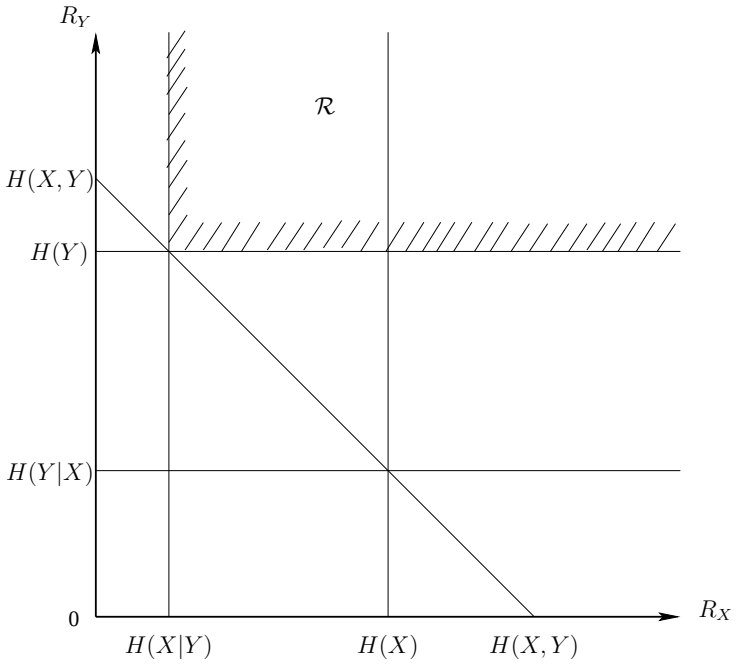

FIG. 1.16 – Région des débits minimums du cas 0001.

1.5 Théorème de Wyner-Ziv

Le théorème d de Slepian-Wolf (cas du quadruple 0001) a été généralisé au cas des sources corrélées à valeurs continues par Wyner-Ziv [WZ76]. Considérons X et Y deux sources corrélées à valeurs continues. La source Y est appelée information de bord. Soit $d(X,\hat{X})$ la mesure de distorsion entre X et la version reconstituée $\hat{X}$ (l'estimation de

X). Alors, quel est le débit minimal nécessaire pour coder X sous la contrainte que la distorsion moyenne $d(X, \hat{X})$ de X est D et que l'information de bord Y est valable seulement au décodeur ? La réponse à cette question a été fournie par Wyner-Ziv en 1976 [WZ76].

Considérons la figure 1.17 qui présente 3 différents cas de codage de la source X avec une information de bord Y suivant l'état des interrupteurs A et B. Le cas où l'interrupteur A est fermé et celui où B est ouvert n'est pas considéré.

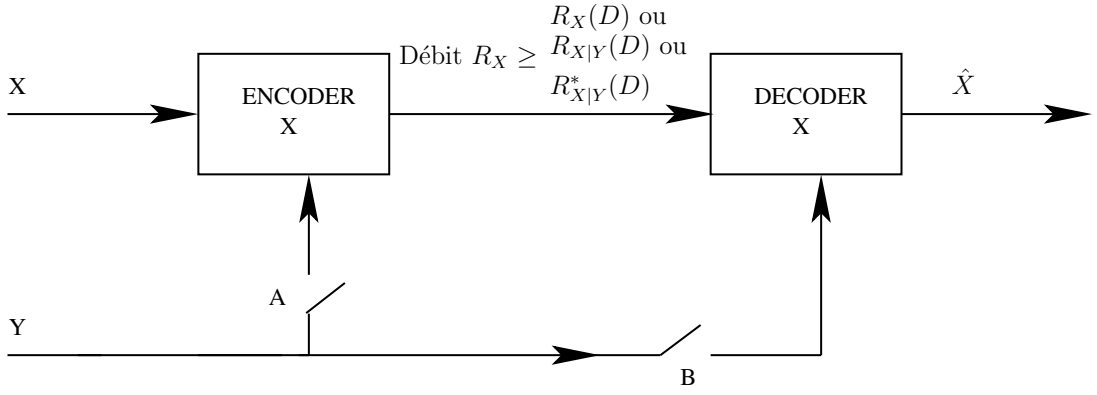

FIG. 1.17 – Codeur avec information de bord d'une source à valeurs continues.

i) les interrupteurs A et B dans 1.17 sont ouverts : Pas d'information de bord disponible ni au codeur, ni au décodeur.
Dans ce cas, le débit minimal pour coder la source X est donné par la fonction débit-distorsion suivante :

$$R_X(D) = \min_{E\{d(X,\hat{X})\} \leq D} I(X; \hat{X}) \tag{1.34}$$

Pour $d(X, \hat{X}) = 0$, il est clair que $R_X(0) = H(X)$.

ii) les interrupteurs A et B sont fermés : l'information de bord Y est connu au codeur et décodeur.
Ici, le débit de la source compressée X est réduit à la fonction conditionnelle débit-distorsion :

$$R_{X|Y}(D) = \min_{E\{d(X,\hat{X})\} \leq D} I(X; \hat{X}|Y) \tag{1.35}$$

iii) l'interrupteur A est ouvert tandis que celui de B est fermé : l'information de bord Y est valable seulement au décodeur (non au codeur).
Pour ce cas, le théorème de Wyner-Ziv donne le débit minimal $R^*_{X|Y}(D)$, appelé fonction débit-distorsion, et défini par :

$$R^*_{X|Y}(D) = \min_{\text{pour tout } Z \text{ et } E\{d(X,\hat{X})\} \leq D} I(X; Z|Y) \tag{1.36}$$

avec Z une variable aléatoire à valeurs continues telle que $Y - X - Z$ forme une chaîne de Markov (les sources Y et Z sont conditionnellement indépendantes étant donné X). $I(X;Z|Y)$ peut être écrit sous la forme :

$$\begin{aligned} I(X;Z|Y) &= H(Z|Y) - H(Z|X,Y) \\ &= H(Z|Y) - H(Z|X) \\ &= H(Z|Y) - H(Z) + H(Z) - H(Z|X) \\ &= I(X;Z) - I(Y;Z) \end{aligned} \tag{1.37}$$

Dans [WZ76], Wyner-Ziv ont montré que :

$$R_{X|Y}(D) \leq R^*_{X|Y}(D) \leq R_X(D). \tag{1.38}$$

Pour $D = 0$, le codeur de Wyner-Ziv redevient un système de compression basé sur le codage de Slepian-Wolf avec $R^*_{X|Y}(0) = R_{X|Y}(0) = H(X|Y)$.

Pour deux sources Gaussiennes corrélées et une mesure de distorsion basée sur l'erreur quadratique moyenne (EQM), nous avons $R^*_{X|Y}(D) = R_{X|Y}(D)$. Par conséquent, le codeur de Wyner-Ziv n'introduit aucune perte en débit de transmission par rapport au système de codage où l'exploitation de la corrélation entre X et Y est effectuée au codeur et au décodeur. Pradhan *et al.* [PCR03] ont récemment prolongé la condition de non perte de débit pour le codage de Wyner-Ziv de deux sources X et Y avec des distributions plus générales à condition que le modèle de corrélation entre X et Y soit une Gaussienne.

1.6 Codage multiterminal de 2 sources

Cette section rappelle les principes du codage de source multiterminal (désigné par CSM). Ce type de codage généralise le théorème de Slepian-Wolf dont le quadruple est 0011 au cas des sources à valeurs continues. Il y a deux classes de codage de source multiterminal comme l'indique la figure 1.18 : le CSM *direct* et *indirect*. Le problème du codage de source multiterminal *direct* [Ber78], [Tun78], [Ooh97] consiste à trouver les régions débit-distorsion pour deux sources continues et corrélées Y_1 et Y_2 qui sont compressées séparément et décodées conjointement sous des contraintes de distorsions respectives $E[d(Y_1, \hat{Y}_1)] \leq D_1$ et $E[d(Y_2, \hat{Y}_2)] \leq D_2$. Dans le codage de source multiterminal *indirect* [BZV96], [VB97], [Ooh98], [PTR04] les deux sources Y_1 et Y_2 sont des observations bruitées d'une autre source X. Le problème est alors de trouver les régions débit-distorsion de Y_1 et Y_2 de sorte qu'on puisse estimer la source X sous la contrainte d'une distorsion $E[d(X, \hat{X})] \leq D$. Le codage de source multiterminal *indirect* est aussi connu sous le nom de problème CEO (Chief Executive Officer).

Les limites inférieures des régions débit-distorsion du codage multiterminal *direct* de deux sources Y_1 et Y_2 transmises aux taux respectifs R_1 et R_2, ont été dérivées dans

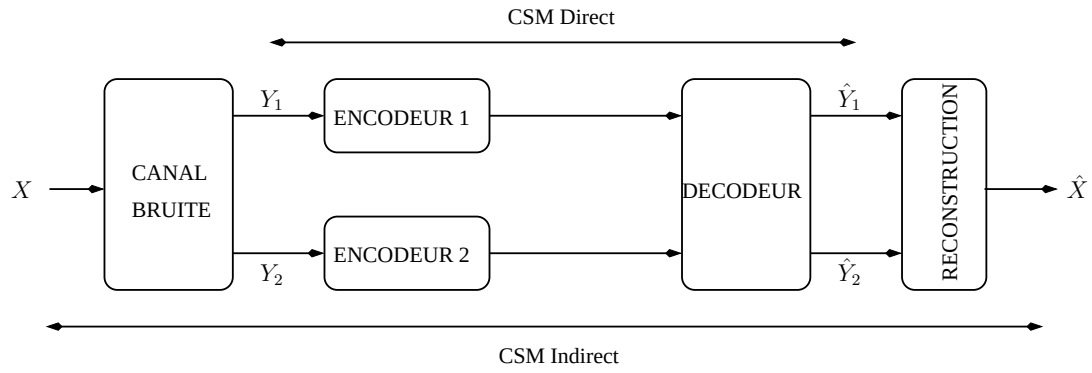

FIG. 1.18 – Codage de source multiterminal Direct et Indirect.

[Ber78], [Tun78] par :

$$R_1 \geq I(Y_1; Z_1) - I(Z_1; Z_2) \tag{1.39}$$

$$R_2 \geq I(Y_2; Z_2) - I(Z_1; Z_2) \tag{1.40}$$

$$R_1 + R_2 \geq I(Y_1 Y_2; Z_1 Z_2) \tag{1.41}$$

avec Z_1 et Z_2 deux variables aléatoires à valeurs continues telles que $Z_1 - Y_1 - Y_2 - Z_2$ forme une chaîne de Markov. Pour le cas CSM *indirect*, les limites inférieures des régions débit-distorsion sont données dans (1.39-1.41) avec $Z_1 - Y_1 - X - Y_2 - Z_2$ forme une chaîne de Markov.

Maintenant, considérons le problème du codage multiterminal *direct* de deux sources Gaussiennes et corrélées Y_1 et Y_2. Soient $\sigma^2_{y_1}$ et $\sigma^2_{y_2}$ les variances respectives de Y_1 et Y_2. Les deux sources Y_1 et Y_2 sont codées et décodées avec des distorsions moyennes respectives D_1 et D_2. Soit $\rho = \frac{cov(Y_1,Y_2)}{\sqrt{\sigma^2_{Y_1}\sigma^2_{Y_2}}}$ le coefficient de corrélation entre Y_1 et Y_2, avec $cov(Y_1, Y_2)$ la covariance de ces deux variables. Pour une mesure de distorsion basée sur l'erreur quadratique moyenne, la limite inférieure de la région débit-distorsion dérivée par [Ooh97] peut être exprimée par :

$$R_1 \geq \frac{1}{2}\log^+[\frac{\sigma^2_{Y_1}}{D_1}(1 - \rho^2 + \rho^2 2^{-2R_2})] \tag{1.42}$$

$$R_2 \geq \frac{1}{2}\log^+[\frac{\sigma^2_{Y_2}}{D_2}(1 - \rho^2 + \rho^2 2^{-2R_1})] \tag{1.43}$$

$$R_1 + R_2 \geq \frac{1}{2}\log^+[(1-\rho^2)\frac{\beta_{\max}}{2}\frac{\sigma^2_{Y_1}}{D_1}\frac{\sigma^2_{Y_2}}{D_2}] \tag{1.44}$$

avec $\log^+ x = \max\{\log x, 0\}$ et $\beta_{\max} = 1 + \sqrt{1 + \frac{4\rho^2}{(1-\rho^2)^2}\frac{D_1}{\sigma^2_{Y_1}}\frac{D_2}{\sigma^2_{Y_2}}}$. La somme minimale $(R_1 + R_2)$ des débits du problème de codage multiterminal *direct* de deux sources Gaussiennes a été dérivée dans [WTV05]. Les auteurs dans [WTV05] ont en effet prouvé qu'il y a égalité dans (1.44), la région inférieure présentée dans [Ooh97].

Pour le problème du codage de source multiterminal *indirect* avec une mesure de distorsion basée sur l'erreur quadratique moyenne, la borne inférieure de $R_1 + R_2$ est donnée par [Ooh98], [PTR04] :

$$R_1 + R_2 \geq \frac{1}{2} \log^+ [\frac{\sigma_X^2}{D}(1 - \frac{\sigma_N^2(\sigma_X^2 - D)}{2\sigma_X^2 D})^{-2}] \tag{1.45}$$

avec σ_X^2 et σ_N^2 les variances de la source X et des bruits Gaussiens N_1 et N_2 tels que $Y_1 = X + N_1$ et $Y_2 = X + N_2$ (X, N_1 et N_2 sont indépendants).

1.7 Conclusion

Nous avons rappelé dans ce chapitre quelques éléments de la théorie de l'information utiles au codage de sources distribuées. Différentes techniques de compression avec ou sans perte ont été décrites. Nous avons en particulier rappelé les principaux fondements du codage de source et du codage de canal.

Nous avons également présenté les théorèmes de Slepian-Wolf et de Wyner-Ziv établissant les débits minimums atteignables en codage distribué respectivement sans perte et avec perte de deux sources corrélées, binaires et Gaussiennes.

Dans le prochain chapitre, nous présenterons un état de l'art des schémas de mise en œuvre concrets de codage de deux sources distribuées.

Chapitre 2

Schémas de codage de sources distribuées : état de l'art

2.1 Introduction

Dans le chapitre précédent, nous avons défini le cadre théorique du codage de deux sources distribuées. Bien que la limite théorique du codage de deux sources distribuées à valeurs discrètes ait été établie en 1973 par Slepian-Wolf [SW73], ce n'est que très récemment que des mises en œuvre concrètes ont été proposées afin d'approcher au mieux cette borne. Celles-ci se basent sur des codeurs de sources (exemple des codes CLV) d'une part et plus souvent sur des techniques de codage de canal, en utilisant par exemple des codes convolutifs, des codes turbo, ou des codes LDPC, d'autre part.

Ce chapitre est consacré à l'état de l'art des schémas de mise en œuvre des techniques de codage de sources distribuées présentées dans la littérature. Tout d'abord, nous nous intéressons au cas de deux sources binaires. Dans un deuxième temps, nous présentons des schémas de codage distribué de deux sources Gaussiennes.

2.2 Codage de Slepian-Wolf de deux sources binaires

2.2.1 Codage de sources distribuées avec des codes CLV

Nous présentons les différents schémas de codage de sources distribuées basés sur des codes CLV. Nous introduisons d'abord la technique de Al Jabri et Al-Issa [JAI97] puis nous présentons le schéma de codage de sources distribuées de Zhao et Effros [ZE01].

2.2.1.1 Technique de Al Jabri et Al-Issa

Les travaux de Al Jabri et Al-Issa [JAI97] ont abouti au premier schéma de codage de sources binaires distribuées utilisant les codes CLV dans le cas des séquences binaires.

Soient X et Y deux sources binaires corrélées. La corrélation est présentée par la fonction de densité de probabilité discrète $P_{XY}(x, y)$. Comme l'indique la figure 2.1, chaque source est codée par un encodeur. L'encodeur Y est appelé codeur primaire qui

garanti à la source Y un débit de transmission proche de son entropie $H(Y)$. Le codeur secondaire est l'encodeur X. Ce dernier exploite les statistiques $P_{XY}(x, y)$ pour coder la source X avec un taux de l'ordre de $H(X|Y)$. Au niveau décodage, le décodeur restitue les sources à partir d'informations reçues.

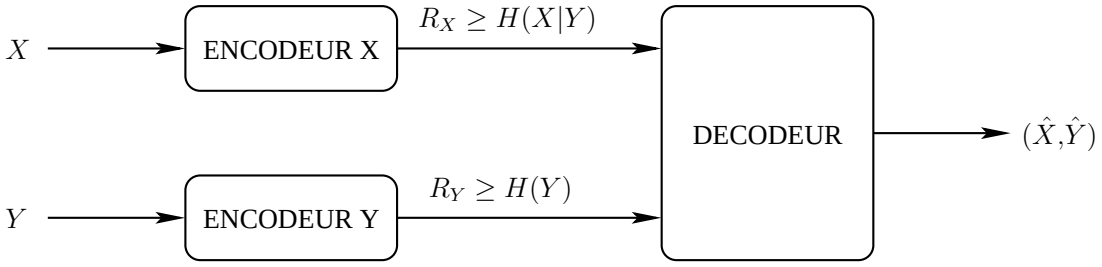

FIG. 2.1 – Structure de la technique de codage distribué de Al Jabri et Al-ISSA.

Le principe de l'algorithme du codage distribué présenté par Al Jabri et Al-Issa est simple. Il s'agit de :

1. Fixer deux séquences X^m et Y^m corrélées et de même taille m. L'alphabet de ces deux séquences est respectivement $\mathbb{X}^m$ et $\mathbb{Y}^m$.
2. Générer les mots de code $\{\underline{h}_1, \underline{h}_2, ..., \underline{h}_m\}$ par l'algorithme de Huffman à partir de la séquence $\{\underline{y}_1, \underline{y}_2, ..., \underline{y}_m\}$ ensemble de symboles de Y^m.
3. Construire pour chaque i, $i = 1, 2, ..., m$, l'ensemble $\mathbb{Y}_i$ dont les éléments appartiennent à $\mathbb{Y}^m$ tel que :

$$\mathbb{Y}_i = \{\underline{y} \ : \ \underline{y} \in \mathbb{Y}^m \text{ et } p(\underline{y}|\underline{x}_i) > 0\}$$

4. Partitionner l'ensemble $\mathbb{X}^m$ en des sous-ensembles $\mathbb{X}_1$, $\mathbb{X}_2$, ... , $\mathbb{X}_{d(m)}$ tels que les éléments de chaque $\mathbb{X}_i$, $i = 1, 2, ... , d(m)$ aient des correspondances disjointes avec les sous-ensembles de $\mathbb{Y}^m$ définis à l'étape précédente.
5. Représenter les sous-ensemble $\mathbb{X}_i$, $i = 1, 2, ...$, par les symboles s_i avec des probabilités correspondantes $p(s_i)$. Les symboles $\{s_1, s_2, ... s_{d(m)}\}$ sont codés par l'algorithme de Huffman. Soit $\{g_1, g_2, ... g_{d(m)}\}$ l'ensemble des mots de code obtenus avec l'algorithme de Huffman.
   Dans l'étape précédente, il y a plusieurs façons de partitionner l'ensemble $\mathbb{X}^m$. La meilleure est celle qui fournit la plus faible entropie, c'est-à-dire :

$$\min \sum_{i=1}^{d(m)} p(s_i) \log_2 \frac{1}{p(s_i)}$$

Au niveau du décodage, dans une première étape, le décodeur estime $\underline{y}_i$ les valeurs à l'entrée du codeur primaire. Dans une seconde étape, ce même décodeur utilise la table de CLV, l'information provenant du codeur secondaire et les valeurs estimées de $\underline{y}_i$, $i = 1, 2, ... , m$, pour résoudre l'ambiguïté sur les symboles $\underline{x}_j$, $j = 1, 2, ... , m$.

Pour bien expliquer la technique d'Al Jabri et Al-Issa, considérons l'exemple de deux sources X et Y avec la fonction de densité de probabilité conjointe qui est présentée par la matrice suivante :

$$\mathbf{P_{XY}} = \begin{pmatrix} 3/20 & 1/15 & 1/20 & 0 & 0 \\ 3/20 & 0 & 0 & 1/30 & 1/12 \\ 0 & 1/15 & 1/20 & 0 & 0 \\ 0 & 1/15 & 1/20 & 1/30 & 1/12 \\ 0 & 0 & 0 & 1/30 & 1/12 \end{pmatrix} \tag{2.1}$$

Les fonctions de densité de probabilité marginale de X et Y sont données par le tableau 2.1.

TAB. 2.1 – *Fonctions de densité de probabilité marginale de X et Y.*

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$p(x_i)$	4/15	4/15	7/60	7/30	7/60
$p(y_i)$	3/10	1/5	3/20	1/10	1/4

Pour compresser la source X, l'ensemble $\mathbb{X}$ est partionné avec $m = 1$ suivant les étapes 3 (tableau 2.2) et 4 (tableau 2.3) de l'algorithme d'Al Jabri et Al-Issa.

TAB. 2.2 – *Correspondance des symboles $\mathbb{X}$ avec les sous-ensembles de $\mathbb{Y}_i$ suivant l'étape 3.*

Symboles de $\mathbb{X}$	Correspondance avec les sous-ensembles de $\mathbb{Y}_i$
x_1	$\{y_1, y_2, y_3\}$
x_2	$\{y_1, y_4, y_5\}$
x_3	$\{y_2, y_3\}$
x_4	$\{y_2, y_3, y_4, y_5\}$
x_5	$\{y_4, y_5\}$

On présente dans le tableau 2.3 les différentes possibilités de l'union entre les partitions de $\mathbb{X}$ et leurs entropies correspondantes. A titre d'exemple, on ne peut pas mettre x_1 et x_2 dans un même ensemble puisqu'il va y avoir une ambiguïté lors du décodage. En effet, la correspondance des symboles x_1 et x_2 avec les sous-ensembles de $\mathbb{Y}_i$ est respectivement $\{y_1, y_2, y_3\}$ et $\{y_1, y_4, y_5\}$ qui ont en commun le symbole y_1.

Pour le taux le plus faible, soit 1.55044, la figure 2.2 illustre la région des débits admissibles R_o de l'exemple présenté où la probabilité d'erreur est presque nulle.

La méthode d'Al Jabri et Al-Issa ne peut être utilisée lorsque la fonction de densité de probabilité conjointe est différente de zéro. En effet, s'il n'y a pas de zéros dans la

TAB. 2.3 – *Partition de $\mathbb{X}$ et leurs entropies correspondantes : étape 4.*

Partitions	Taux
$\{x_1\}$, $\{x_2\}$, $\{x_3\}$, $\{x_4\}$, $\{x_5\}$	2.23012
$\{x_1\}$, $\{x_2\}$, $\{x_3, x_5\}$, $\{x_4\}$	1.99679
$\{x_1\}$, $\{x_2, x_3\}$, $\{x_4\}$, $\{x_5\}$	1.89028
$\{x_1, x_5\}$, $\{x_2\}$, $\{x_3\}$, $\{x_4\}$	1.89028
$\{x_1, x_5\}$, $\{x_2, x_3\}$, $\{x_4\}$	1.55044

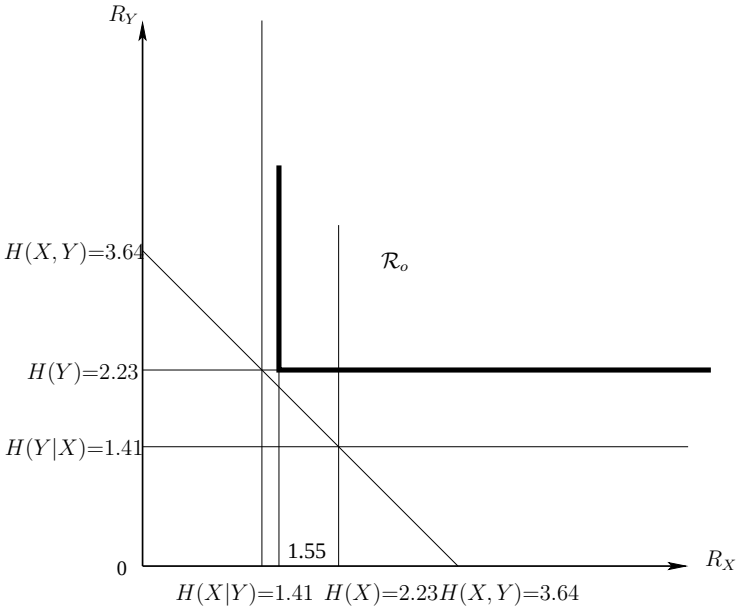

FIG. 2.2 – Région des débits admissibles pour l'exemple présenté avec $m = 1$ suivant la technique d'Al Jabri et Al-ISSA.

matrice $\mathbf{P_{XY}}$, alors il serait impossible de construire les ensembles $\mathbb{Y}_i$ et par conséquent, le partitionnement de l'ensemble $\mathbb{X}^m$ ne pourrait pas être effectué.

2.2.1.2 Travaux de Zhao et Effros

La technique de Zhao et Effros [ZE01] est basée sur un code CLV (code de Huffman ou code arithmétique) pour compresser deux sources $(X, Y) \in \mathbb{X} \times \mathbb{Y}$ sans mémoire et à alphabet fini. Soit $p(x, y)$ la fonction de densité de probabilité conjointe de ces deux sources.

Dans cette partie, la figure 2.1 est considérée comme le schéma de base pour présenter la technique de Zhao et Effros. La source Y est codée par un codeur de source traditionnel. Quant à la source X, le but est de trouver le modèle de l'encodeur correspondant puisque le décodeur a déjà accès à l'information de bord Y.

La façon de partitionner un ensemble, présentée dans l'algorithme du codage à longueur variable d'Al Jabri et Al-Issa, joue un rôle très important pour s'approcher des limites de Slepian-Wolf. Dans le cas de Zhao et Effros, le principe de la partition est un peu différent. En effet, le terme sous-ensemble est remplacé par la notion groupe et/ou groupe de niveau dans une structure d'arbre.

La solution pour coder X est alors : pour chaque x et $x\prime \in \mathcal{A}_y = \{x \in \mathbb{X}/p(x, y) > 0 \text{ et } y \in \mathbb{Y}\}$, l'ensemble d'éléments d'un même niveau de l'arbre, le mot de code de x à la sortie de l'encodeur n'est pas un préfixe à celui de $x\prime$. Dans le cas d'un chemin avec différents niveaux de l'arbre, certains mots de code peuvent être des préfixes à d'autres. Au décodage, vu qu'il faut décoder Y en premier et que tous les mots de code satisfont la condition de préfixe, alors l'estimation des symboles de X peut se faire avec une très faible probabilité d'erreur.

Avant de décrire le principe de la méthode de Zhao et Effros avec un exemple, nous allons définir quelques termes utiles pour la partition de l'ensemble $\mathbb{X}$. Ainsi, soient x_1 et $x_2 \in \mathbb{X}$.

- Si $p(x_1, y)p(x_2, y) = 0$, on dit que x_1 et x_2 sont associés sous $p(x, y)$ pour tout $y \in \mathbb{Y}$.
- L'ensemble $\mathcal{X} = (x_1, x_2, \ldots, x_m)$ est appelé groupe de niveau-1 pour $p(x, y)$, si les différentes paires x_i et x_j de ses éléments sont associés sous $p(x, y)$.
- Le groupe $\mathcal{X} = (\mathcal{R} : \mathcal{C}(\mathcal{R}))$ est nommé groupe de niveau-2 si $\mathcal{R}$, un groupe de niveau-1, est le parent du groupe de niveau-1 $\mathcal{C}(\mathcal{R})$ dans un graphe. Les différentes paires d'éléments de $\mathcal{R}$ et de $\mathcal{C}(\mathcal{R})$ sont associées sous $p(x, y)$. Dans le cas où $\mathcal{C}(\mathcal{R})$ est un groupe de niveau-$(M-1)$, alors $\mathcal{X}$ sera un groupe de niveau-M.
- La partition $\mathcal{P} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n\}$ de $\mathbb{X}$, avec $m \geq n$, satisfait les conditions : $\bigcup_{i=1}^{n} \mathcal{X}_i = \mathbb{X}$ et $\mathcal{X}_i \bigcap \mathcal{X}_j = \emptyset$ pour tout $i \neq j$.
- L'arbre de partition $T(\mathcal{P}) = T(\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n\})$ représente tous les liens entre les différents groupes.

Maintenant, prenons l'exemple du tableau 2.4. La figure 2.3(a) montre l'arbre de partition $T(\mathcal{P})$ avec $\mathcal{P} = \{(x_3, x_6), \mathcal{X}(3)\}$ et $\mathcal{X}(3) = ((x_7) : \{(x_0), (x_1), ((x_2) : \{(x_4), (x_5)\})\})$ est un groupe de niveau-3. Il est clair que x_7 et $\{x_0, x_1, x_2\}$ sont associés sous $p(x, y)$ de même pour x_2 et $\{x_4, x_5\}$.

TAB. 2.4 – *Fonction de densité de probabilité $p(x, y)$ avec $\mathbb{X}$ et $\mathbb{Y}$ de même taille*

	x\y	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
$p(x,y)$	x_0	0.04	0	0.04	0.02	0	0	0	0
	x_1	0	0.04	0	0	0.05	0.1	0	0
	x_2	0.15	0	0.05	0	0	0	0	0
	x_3	0	0.05	0	0.06	0	0	0	0
	x_4	0	0.06	0	0	0.05	0	0	0
	x_5	0	0	0	0.01	0.02	0.03	0	0
	x_6	0	0	0.01	0	0	0.06	0.02	0.01
	x_7	0	0	0	0	0	0	0.05	0.08

Pour décrire chaque chemin dans l'arbre d'une manière unique, un ordre de descente d'un nœud à un autre est fixé et un nombre est attribué (de gauche à droite). Les enfants d'un nœud n sont étiquetés par $n1$, $n2$, ..., nK, avec K le nombre d'enfants du nœud correspondant. Les étiquettes de la partition $T(\mathcal{P})$ sont illustrées dans la figure (b).

Pour trouver un code à longueur variable correspondant à la partition $T(\mathcal{P})$ selon l'arbre de la figure 2.3(a), il suffit d'appliquer les conditions suivantes :

- les groupes (x_3, x_6) et (x_7) ne doivent pas avoir des mots de code où l'un est le préfixe de l'autre. Suivant la figure 2.3(c), (x_3, x_6) est codé par 0 et (x_7) par 1. Le sens d'attribution est de la gauche vers la droite en commençant par 0 puis 1.
- le groupe (x_7) est le père de $\{x_0, x_1, x_2\}$. Donc les mots de code respectifs de (x_0), (x_1) et (x_2) ont comme préfixe le bit 1, c'est-à-dire le mot de code de (x_7).
- pareil pour les groupes $\{(x_4), (x_5)\}$ où le mot de code 11 de leur pères (x_2) est un préfixe de leurs mots de code respectifs.

La figure 2.3(c) montre le code adapté à la partition $T(\mathcal{P})$. Chaque nœud est présenté par un mot de code différent.

Dans l'article de Zhao et Effros [ZE01], on trouve une présentation de plusieurs théorèmes pour déterminer et fixer la partition optimale d'un ensemble $\mathbb{I}$ et de trouver son codeur adaptatif optimal.

Dans la solution de Zhao et Effros, si la fonction de densité de probabilité $p(x, y)$ est différente de zéro, alors la méthode proposée ne peut être appliquée.

2.2.2 Codage de sources distribuées avec des codes de canal

La raison derrière l'utilisation des codes de canal dans le contexte de codage de sources distribuées est que la corrélation entre la source X et l'information de bord Y peut être modélisée par un "canal de corrélation" virtuel (exemple le canal binaire symétrique). L'entrée de ce "canal" est X et la sortie est Y. L'information de bord Y est considérée comme étant une version bruitée de X. Le code canal qui présente les meilleures performances dans le contexte de codage canal permettra donc de mieux s'approcher de la limite de Slepian-Wolf.

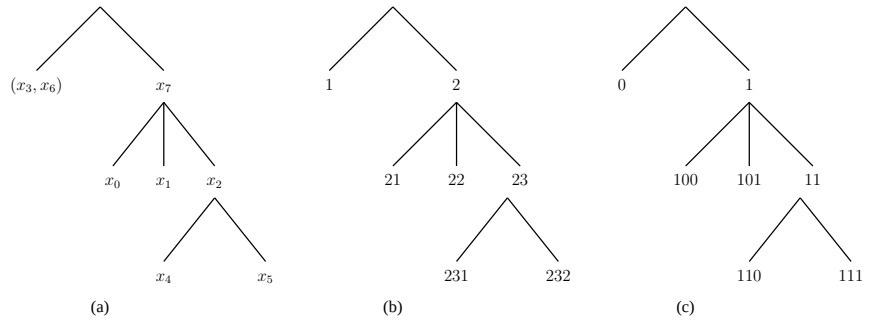

FIG. 2.3 – (a) Arbre de partition $T(\mathcal{P})$, (b) Etiquette de $T(\mathcal{P})$, (c) Code associé à $T(\mathcal{P})$.

2.2.2.1 DISCUS

Une première solution utilisant des codes de canal a été introduite en 1999 par Pradhan et Ramchandran [PR99]. Ils ont développé une solution basée sur le codage de canal. Leur approche a été nommée codage de sources distribuées en utilisant des syndromes (en anglais DIstributed Source Coding Using Syndromes : DISCUS).

Comme première tentative pour appliquer le théorème du codage de Slepian-Wolf, Pradhan et Ramchandran ont proposé un exemple simple et intéressant. Ils ont considéré deux mots binaires équiprobables X et Y à 3 bits ($H(X) = H(Y) = 3$ bits). Ces deux mots sont corrélés dans le sens suivant : la distance de Hamming entre X et Y est au plus 1 ($d_H(X, Y) \leq 1$). Si Y est disponible à l'encodeur et au décodeur, il est clair qu'il est inutile de décrire X en utilisant 3 bits, puisqu'il y a seulement 2 bits d'incertitude entre X et Y, donc $H(X|Y) = 2$ bits. Cependant, si Y est connu juste au décodeur, alors Pradhan et Ramchandran ont montré que X peut être décrite avec seulement 2 bits de données. En effet, les 8 valeurs possibles de X sont réparties dans 4 ensembles différents δ_i ($i = 0, 1, 2, 3$). Chaque ensemble δ_i contient deux éléments dont la distance de Hamming entre eux est la plus grande : $\delta_0 = \{000, 111\}$, $\delta_1 = \{001, 110\}$, $\delta_2 = \{010, 101\}$ et $\delta_3 = \{011, 100\}$. Chaque ensemble δ_i est représenté par un coset i codé sur 2 bits. Ainsi, au lieu de décrire X par 3 bits, chaque mot de code est associé au coset correspondant à deux bits. Par exemple, soient $X = 010$ et $Y = 110$. L'encodeur observe $X = 010$ et envoie le coset 10. $Y = 110$ est connu au décodage. Il y a 4 choix équiprobables pour X : 110, 100, 010 ou 111. Le décodeur reçoit 10 et décode la valeur de X dont la distance de Hamming à Y est la plus faible. Suivant la figure 2.4, la valeur de X ne peut être que 010. La connaissance de Y et la distance de Hamming 1 de X permettent de résoudre l'incertitude du décodage. Dans ce cas, le débit $H(X, Y) = H(Y) + H(X|Y) = 3 + 2 = 5$ bits est suffisant pour compresser les deux sources X et Y.

Le même exemple peut être codé en utilisant des codes en blocs (n,k, d), où n est la longueur du code, k est la longueur du message et d est la distance minimale entre deux mots de code. Considérons pour cela la matrice de parité $\mathbf{H}$ du code bloc (3,1,3) définie par :

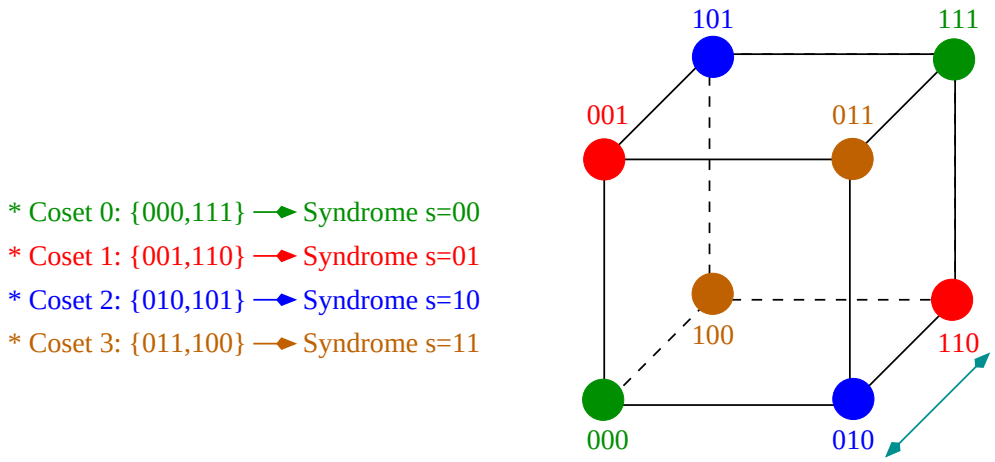

FIG. 2.4 – Exemple d'un codeur de deux sources distribuées utilisant le principe de DSCUS.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{2.2}$$

Pour chaque mot de code X à compresser, un syndrome $\mathbf{s} = \mathbf{H}X$ représenté sur deux bits est transmis au décodeur. Les syndromes 00, 01, 10 et 11 correspondent aux mots de code valides (000,111), (001,110), (010,101) et (011,100), respectivement. Au décodage la connaissance du syndrome et de Y permet de retrouver le mot de code envoyé.

D'une manière plus générale, considérons deux sources X et Y telles que $d_H(X,Y) \leq t$ et un codeur bloc $(n,\ k,\ 2t+1)$. Selon la valeur de X, l'encodeur envoie le syndrome $\mathbf{s} = \mathbf{H}X$ au décodeur. Pour faciliter le décodage, considérons la matrice $\mathbf{H} = [\mathbf{A}|\mathbf{I}]$ sous sa forme systématique. Le décodeur dispose de Y et de $\mathbf{s}$ pour calculer la valeur de $Y' = Y + Z$ (l'addition se fait modulo-2 et $Z = [0|\mathbf{s}]$ est le mot de code dont le syndrome est le même que $\mathbf{s}$). Soit X' un mot de code de syndrome nul et dont la distance de Hamming par rapport à Y' est la plus faible. Ainsi, la valeur du mot de code X peut être déduit comme $X = X' + Z$. Le taux de compression de X est $n : (n-k)$, avec $H(X|Y) = n - k$.

L'algorithme ci-dessus n'est pas limité aux codes en blocs. Il peut être étendu aux codes convolutifs.

2.2.2.2 Codage de deux sources distribuées utilisant un code turbo

Comme dans les parties précédentes, considérons X et Y deux sources corrélées (i.i.d). La mise en œuvre du codage de sources distribuées utilisant les codes turbo [BGT93], [BG96] a été entreprise pour la première fois par Garcia-Frias et Zhao [GFZ01b], [GFZ01a]. Ils ont traité plus particulièrement le problème du codage conjoint source-canal. Pour atteindre le taux de compression désiré, Garcia-Frias et Zhao ont proposé

de poinçonner les sorties des encodeurs turbo. Par la suite, on trouve d'une part les travaux de Bajcsy et Mitran [BM01] et ceux de Aaron et Girod [AG02] d'autre part. La compression dans ces deux travaux est basée sur des techniques qui n'utilisent pas le poinçonnage.

Considérons deux sources binaires et corrélées *i.i.d* X et Y. La corrélation entre ces deux sources est modélisée par le canal binaire symétrique (CBS) caractérisé par la probabilité de transition p. Soient $Pr(X_i = Y_i) = 1 - p$ et $Pr(X_i \neq Y_i) = p$. Dans cette partie, l'information de bord Y est codée séparément et supposée connue parfaitement au décodage (c'est-à-dire Y est transmise à un débit supérieur ou égal à son entropie $H(Y)$). Pour la source X, le débit de transmission est supérieur ou égal à la valeur de l'entropie conditionnelle $H(X|Y) = h(p) = -p\log_2 p - (1-p)\log_2(1-p)$.

A) Travaux de Bajcsy et Mitran

La technique présentée par Bajcsy et Mitran [BM01] est basée sur un code turbo utilisant un encodeur élémentaire FSM (Finite-State Machine : machine à nombre d'états fini) dont le taux de codage est $\frac{k}{n}$ avec $k > n$ et le nombre d'états est 2^k. Par conséquent, le taux de codage global est $\frac{k}{2n}$, c'est-à-dire un taux de compression des séquences binaires X de l'ordre de $\frac{2n}{k}$.

Un exemple de treillis d'un encodeur FSM avec $k = 2$ et $n = 1$ est illustré à la figure 2.5. Les étiquettes sur les branches du treillis indiquent le nombre de bits en entrée et celui en sortie du codeur (de gauche à droite). On remarque bien que le treillis est spécialement créé pour effectuer directement la compression. En effet, sur le treillis, le nombre de bits en entrée est plus élevé que celui en sortie.

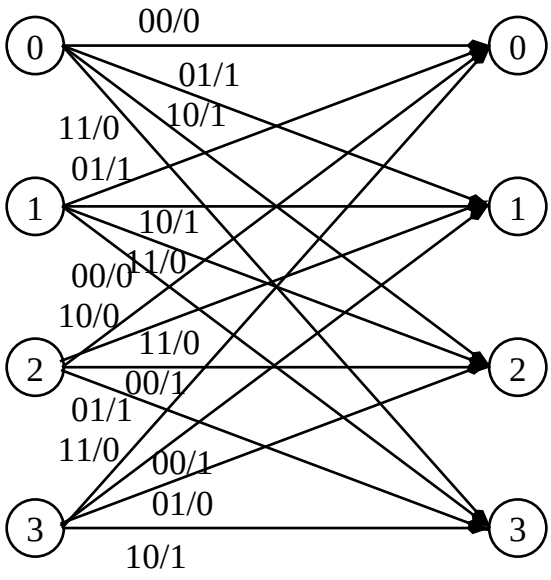

FIG. 2.5 – Treillis d'un codeur FSM avec $k = 2$ et $n = 1$.

Le décodage s'effectue de manière itérative. Chaque décodeur utilise l'algorithme

BCJR pour générer les probabilités a posteriori conditionnelle à l'information de bord Y (transmise à un taux égal à $H(Y)$) et les bits envoyés par les codeurs FSM. Dans ce cas, les informations extrinsèques sont considérées comme des Gaussiennes de variance unité et de moyenne $x - 1/2$ avec $x \in \{0, 1\}$.

B) Travaux de Aaron et Girod

Dans un premier temps Aaron et Girod [AG02] ont présenté une solution au codage distribué de deux sources, binaires et corrélées, basée sur un code turbo dont l'encodeur élémentaire est un codeur convolutionnel récursif systématique (CRS) de taux $\frac{n-1}{n}$ avec $n \geq 2$. La source Y est transmise à un taux $R_Y \geq H(Y)$ sans erreur au décodeur. Comme l'indique la figure 2.6, seules les sorties de bits de parité X_P^1 et X_P^2 (un par encodeur) sont transmises au décodeur. Les sorties systématiques sont inutiles au niveau du décodage puisqu'elles sont déjà disponibles sous une forme bruitée par les valeurs de Y. Par conséquent, le taux de compression de X est égal à $R_X = \frac{2}{n-1}$ bit par bit en entrée.

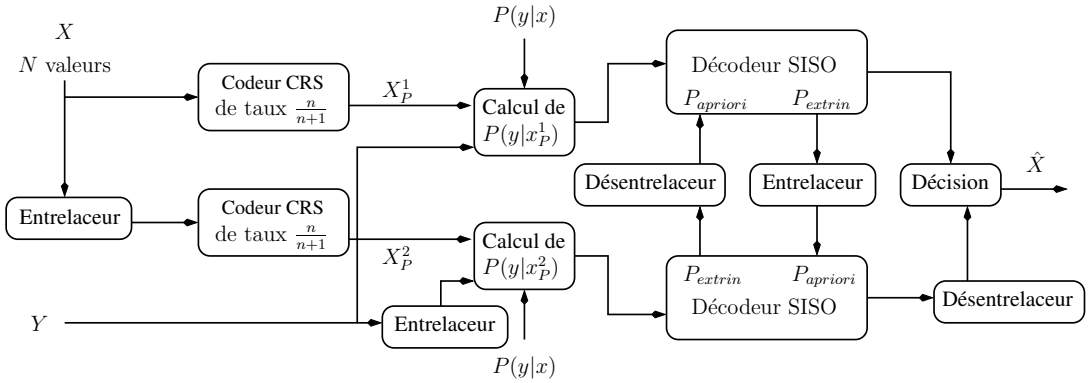

FIG. 2.6 – Schéma du codeur/décodeur pour deux sources binaires et distribuées utilisé dans les travaux de Aaron et Girod.

Le décodeur du système présenté par Aaron et Girod, est un décodeur turbo itératif composé de deux décodeurs SISO concaténés en série à travers un entrelaceur ou un désentrelaceur. La seule différence par rapport au décodeur utilisé dans le cas du décodage canal, est au niveau du calcul de la métrique de branche qui dépend des valeurs de parité X_P^1 ou X_P^2, de Y (l'information de bord) et de la corrélation entre les sources donnée par la fonction de densité de probabilité $p(y|x)$. La probabilité $p(y|x)$ représente ici le modèle de corrélation et ainsi remplace la probabilité de transition du canal dans l'algorithme du décodage MAP.

2.2.2.3 Codage de deux sources distribuées utilisant un code LDPC

Pour s'approcher au mieux de la borne théorique de Slepian-Wolf, les codes LDPC ont été utilisés dans [LXG02b], [TGFZ03].

Dans [LXG02b], Liveris *et al.* ont présenté une solution de codage de deux sources binaires distribuées X et Y basée sur des codes LDPC. La source Y, étant l'information de bord, transmise à un taux égal à son entropie $H(Y)$ et disponible au décodeur. Ils ont montré que la compression de X avec un codeur LDPC irrégulier permet de mieux s'approcher de la limite de Slepian-Wolf.

Soit un code LDPC (n, k) qui peut générer 2^{n-k} syndromes différents avec n la taille de la séquence $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ et $n : (n-k)$ le taux de compression. Soit $\mathbf{H}$ la matrice de parité du code LDPC. Au codeur, la multiplication de $\mathbf{X}$ par $\mathbf{H}$ génère le syndrome $\mathbf{Z}$ de taille $n-k$.

Le décodeur doit estimer la séquence $\mathbf{X}$ de taille n à partir de celle de $\mathbf{Y} = \{Y_1, Y_2, ..., Y_n\}$ et de $\mathbf{Z}$ de longueur n et $n-k$, respectivement. Le décodage s'effectue de manière itérative. Avant de décrire le principe du décodage LDPC adapté au codage de sources distribuées, nous allons définir quelques termes utilisés par l'algorithme de Liveris *et al.* :

- x_i, $y_i \in \{0, 1\}$, $i = 0, 1, ..., n$, sont les valeurs courantes respectives de X_i et Y_i, et qui correspondent au $i^{\text{ème}}$ nœud de gauche noté par g_i.
- $l_i \in \{2, 3, ...\}$ est le degré de g_i.
- $q_{i,m}^{out}$ (respectivement $q_{i,m}^{in}$), $i = 0, 1, ..., n$ et $m = 1, 2, ..., l_i$, est le rapport de vraisemblance (désigné par LLR) transmis sur la $m^{\text{ème}}$ branche à partir du (respectivement vers le) nœud g_i.
- $s_j \in \{0, 1\}$, $j = 0, 1, ..., n-k$, est la valeur du syndrome Z_j correspondant au $j^{\text{ème}}$ nœud de droite noté par d_j.
- $r_j \in \{2, 3, ...\}$ est le degré de d_j.
- $t_{j,m}^{out}$ (respectivement $t_{j,m}^{in}$), $j = 0, 1, ..., n-k$ et $m = 1, 2, ..., r_j$, est le rapport de vraisemblance LLR transmis sur la $m^{\text{ème}}$ branche à partir du (respectivement vers le) nœud d_j.

Soit $q_{i,0}$, $i = 0, 1, ..., n$, une variable définie par :

$$q_{i,0} = \log \frac{Pr(x_i = 0|y_i)}{Pr(x_i = 1|y_i)} = (1 - 2y_i) \log \frac{1-p}{p} \tag{2.3}$$

avec $p = Pr(x_i \neq y_i) < 0.5$ la probabilité de transition du canal CBS, modèle de corrélation entre X et Y. Le LLR transmis du nœud g_i sur la $m^{\text{ème}}$ branche peut s'écrire par :

$$q_{i,m}^{out} = q_{i,0} + \sum_{j=1, j \neq m}^{l_i} q_{i,j}^{in} \tag{2.4}$$

avec initialement $q_{i,j}^{in} = 0$.

Les valeurs $q_{i,m}^{out}$ sont assignées à celles de $t_{j,\pi(i,m,j)}^{in}$ qui leur correspondent selon les connexions dans le graphe "bipartite" du code LDPC. A partir de cette correspondance, suivant la règle de la fonction tangente hyperbolique (désignée par tanh) et le syndrome, le LLR transmis du nœud de droite d_j sur la $m$ème branche est :

$$\tanh(\frac{t_{j,m}^{out}}{2}) = (1-2s_j) \prod_{i=1, i\neq m}^{r_j} \tanh(\frac{t_{i,m}^{in}}{2}) \tag{2.5}$$

A partir de là, les valeurs de $q_{i,m}^{in}$ sont mises à jour par celles de $t_{j,\pi(i,m,j)}^{out}$ ($q_{i,m}^{in} = t_{j,\pi(i,m,j)}^{out}$) pour toutes les branches du graphe "bipartite" du code LDPC. A ce moment, une nouvelle itération peut avoir lieu. L'estimée de x_i, $\hat{x}_i$ peut être déterminée comme suit :

$$\hat{x}_i = \begin{cases} 0, & \text{si } q_{i,0} + \sum_{m=1}^{l_i} q_{i,m}^{in} \geq 0, \\ 1, & \text{sinon.} \end{cases} \tag{2.6}$$

Dans le tableau 2.5, on présente les performances des codeurs Slepian-Wolf basées sur le codeur LDPC de [LXG02b] d'une part et sur le code turbo de [AG02] d'autre part. R est le taux de compression de la source X. Vu que la source Y est transmise à un taux égal à son entropie $H(Y)$ dans les deux solutions, nous comparons leurs performances selon le taux de compression R. Il est clair que l'utilisation des codeurs LDPC permet de se rapprocher davantage de la borne théorique donnée par R.

TAB. 2.5 – *Comparaison des performances obtenues dans [LXG02b] et [AG02].*

R	0.125	0.250	0.5
$H(X\|Y)$ [AG02]	0.089	0.189	0.381
$H(X\|Y)$ [LXG02b]	0.091	0.204	0.466

2.3 Codage de Wyner-Ziv de deux sources Gaussiennes

Dans cette section, on considère deux sources corrélées X et Y à valeurs continues (*i.i.d.* Gaussiennes) telles que $Y = X + N$. X et N sont indépendantes, de moyennes nulles et de variances respectives σ_X^2 et σ_N^2.

D'un point de vue pratique, le codeur de Wyner-Ziv est formé d'un quantificateur suivi d'un codeur Slepian-Wolf comme l'indique la figure 2.7. En fait, il y a encore de la corrélation entre la version quantifiée de X et l'information de bord Y. Par conséquent, le codage de Slepian-Wolf pourrait être utilisé pour exploiter cette corrélation et ainsi réduire le débit de transmission de la source X.

Le codeur de Wyner-Ziv selon la figure 2.7 entraîne deux pertes lors du codage. Une perte dûe à la quantification (distorsion) et une autre dûe au codage de Slepian-Wolf qui est basé sur un code canal (probabilité d'erreur lors du décodage). Au décodeur, l'information de bord Y peut être employée conjointement au décodage et à la reconstruction (déquantification) de $\hat{X}$ (l'estimée de X) pour aider à réduire la distorsion de la source

X. Afin d'atteindre la limite théorique de Wyner-Ziv, les meilleurs quantificateurs et codeurs de canal doivent être utilisés. Ainsi, beaucoup d'efforts ont été consacrés à la conception des quantificateurs pour une meilleure reconstruction de la source X comme : quantificateur de Lloyd-Max généralisé [RMZG03], quantificateur en lattice emboîtée (en anglais nested lattice quantization) [ZS98], [Ser00], [ZSE02], [XLCL03], [LCLX04] ou bien les quantificateurs codés en treillis (TCQ) [PR03a], [CPR03] et [YCXZ03]. Une solution de codage basée sur un quantificateur TCQ et un codeur Slepian-Wolf sera présentée dans le chapitre 3.

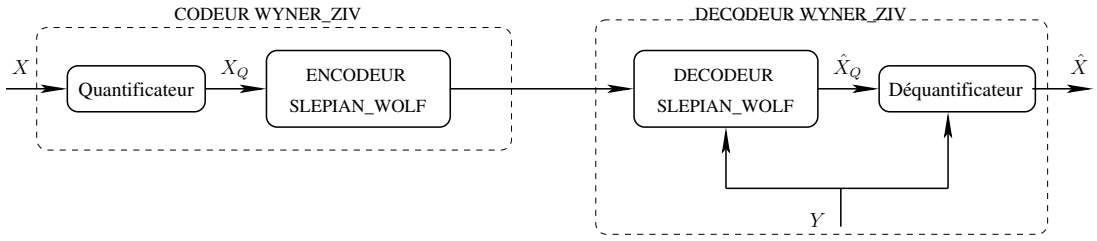

FIG. 2.7 – Codeur pratique de Wyner-Ziv.

2.3.1 DISCUS pour deux sources à valeurs continues

La source X est quantifiée en utilisant un quantificateur scalaire de Lloyd-Max.

2.3.1.1 Quantificateur scalaire et constructions de cosets sans mémoires

Soit le quantificateur scalaire de Lloyd-Max à $V = 8$ niveaux, déterminé à partir de la fonction de densité de probabilité de X. La figure 2.8 illustre la répartition de l'ensemble de ces niveaux $\nabla = \{r_0, r_1, \ldots, r_7\}$.

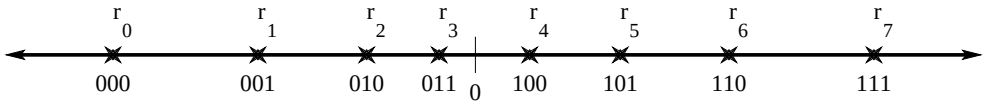

FIG. 2.8 – Quantification scalaire de Lloyd-Max à 8 niveaux.

L'ensemble ∇ est divisé en $M = 4 \leq V$ sous-ensembles $\triangle_i$ $(i = 0, 1, 2, 3)$ représentés par différents cosets tels que la distance minimale entre les deux éléments d'un $\triangle_i$ soit la plus grande. Les meilleurs sous-ensembles $\triangle_i$ dans notre cas sont :

- $\triangle_0 = \{r_0, r_4\}$ représenté par le coset 00,
- $\triangle_1 = \{r_1, r_5\}$ avec 01 comme coset,
- $\triangle_1 = \{r_2, r_6\}$ défini par le coset 10,
- et $\triangle_3 = \{r_3, r_7\}$ décrit par le coset 11.

A partir de l'échantillon X, l'encodeur détermine le mot de code de l'ensemble ∇ le plus proche (chercher l'élément r_i qui réduit au minimum la distorsion entre X et

r_i) puis il envoie le coset correspondant. Le décodeur, de son côté, déchiffre le mot de code qui est le plus proche en métrique de distance à Y ($\text{argmin}\,||r_i - Y||^2$) et dont le coset est le même que celui envoyé par l'encodeur. Une fois le mot de code déterminé, le dé-quantificateur estime $\hat{X}$ avec le minimum de distorsion $\rho(X, \hat{X}) = (X - \hat{X})^2$.

$$\hat{X} = \underset{a \in R}{\text{argmin}}\, E[(x-a)^2 | Y = y, \text{ mot de code estimé}] \tag{2.7}$$

2.3.1.2 Quantificateur scalaire et constructions de cosets basés sur des codes en treillis : Coset avec mémoire

Décrivons maintenant le cas de la quantification scalaire avec une construction de coset ayant de la mémoire. Dans ce cas, au lieu d'établir les cosets sur l'espace ∇ (échantillon par échantillon) comme pour le cas sans mémoire vu dans la section précédente, ils sont déterminés à partir de l'espace ∇^n avec un nombre de niveaux $V = 8$ (n est la longueur de la séquence X). Donc au total, il y aura 2^{3n} séquences différentes.

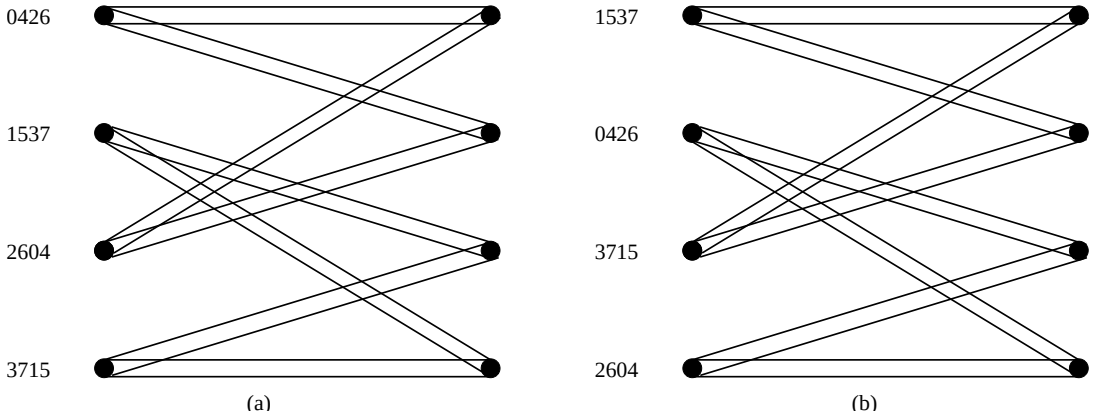

FIG. 2.9 – Treillis du codeur utilisé : (a) treillis principal, (b) treillis complémentaire.

La division de l'espace ∇^n en des cosets est réalisée sur la base des codes convolutionnels et selon la méthode de répartition de la modulation codée en treillis (Trellis-Coded-Modulation : TCM) [Ung82]. Dans le cas que l'on va présenter, l'espace ∇ est partitionné en quatre sous-ensembles comme dans la section précédente. Le codeur convolutionnel utilisé est de taux 2/3 et de longueur de contrainte 3 (treillis à 4 états). Le treillis principal de ce codeur est présenté à la figure 2.9(a) (les étiquettes des branches sont des éléments de l'espace ∇).

Au niveau du codage de sources, Pradhan et Ramchandran ont utilisé le principe du syndrome calculé à partir des codeurs convolutionnels pour compresser les informations quantifiées de X. Pour cela, soit $\mathbf{Q} : \{0,1\}^{3n} \longrightarrow \nabla^n$ une fonction telle que $\mathbf{Q}(\mathbf{x}) = \mathbf{Q}(x_1, x_2, ..., x_n) = (Q(x_1), Q(x_2), ..., Q(x_n))$ où $Q : \{0,1\}^3 \longrightarrow \nabla$ est une autre fonction d'association entre la représentation binaire en 3 bits et les éléments de l'espace ∇ ($Q(\zeta) = r_\eta$ avec $\zeta \in \{0,1\}^3$ la présentation binaire de η).

Soit $\mathbf{H}$ la matrice de parité du codeur convolutionnel. Soit aussi, $\mathbf{Z} = \mathbf{Q}^{-1}(\Theta)$ avec Θ un mot de code de l'espace ∇^n, alors la quantité $\mathbf{HZ}$ sera le syndrome $\mathbf{s}$ (appartenant à l'ensemble $\{0,1\}^n$) du vecteur Θ. Dans ce cas, on peut déduire que :

- le syndrome 0 est décrit par le treillis principal appelé aussi coset de treillis principal (figure 2.9(a)),
- le syndrome 1 est défini par le treillis complémentaire appelé aussi coset de treillis complémentaire (figure 2.9(b)).

Ainsi, le décodeur a l'accès à l'information de bord Y (n échantillons) et au syndrome $\mathbf{s}$ (vecteur de taille n) envoyé par l'encodeur. Selon les valeurs du syndrome reçues, le décodeur identifie les cosets contenant le mot de code recherché. Si le syndrome est égal à zéro, alors l'algorithme du décodage de Viterbi [Vit67], [For73] peut être utilisé en se basant sur le treillis du coset principal. Par contre, pour d'autres valeurs de syndrome, l'algorithme doit être généralisé comme suit : considérons le treillis de la figure 2.9(a). Ce dernier présente le coset avec un syndrome nul. A l'instant $k-1$, chaque nœud maintient le chemin de la métrique minimale. Dès lors que l'instant k est atteint, on calcule la métrique minimale pour chaque nœud suivant la valeur de l'échantillon de Y et la valeur du syndrome. Si le k^{me} bit du vecteur de syndrome est égal à 1 au lieu de 0, alors le coset de treillis complémentaire sera utilisé. Autrement dit, pour la séquence $\mathbf{Q}([\mathbf{0}|\mathbf{0}|\mathbf{s}])$ (qui est un mot de code dont le coset est le syndrome $\mathbf{s}$) et à l'instant k, si le k^{eme} bit de $\mathbf{s}$ est 1, alors le coset de treillis complémentaire sera la base de l'algorithme du décodage de Viterbi.

Soit $\mathbf{x\prime}$, la séquence du mot de code la plus proche en terme de métrique de distance à $\mathbf{y}$, obtenue à partir de l'algorithme que l'on vient de présenter. L'estimation de la séquence $\mathbf{x}$ peut se faire par :

$$\hat{X} = \underset{a \in R^n}{\operatorname{argmin}}[E\{(x-a)^2 | Y = y, \text{ mot de code estimé}\}] \tag{2.8}$$

Une autre méthode peut être utilisée pour déterminer le syndrome d'une séquence X. En effet, après quantification et suivant les r_n valeurs appartenant à ∇^n, le coset de treillis principal ou complémentaire peut être fixé ainsi que la valeur du syndrome. Pour mieux expliquer ce point de vue, prenons l'exemple de la figure 2.10 où la sortie de quantificateur est la séquence 7, 3, 2, 1, 5, 6. Dans ce cas, le syndrome est 101100 et il est obtenu comme suit :

- le codeur doit commencer par l'état 0 (on a 4 états), la première valeur du quantificateur est 7, donc le coset de treillis complémentaire doit être utilisé (figure 2.9(b)), résultat : le premier syndrome est 1 et l'état suivant est 1.
- de l'état 1 avec une valeur du quantificateur 3, on peut seulement utiliser le coset de treillis principal. Ainsi le syndrome est 0 et l'état suivant est 3 (figure 2.9(a)).
- de l'état 3 avec une valeur du quantificateur 2, la seule possibilité est d'utiliser le coset de treillis complémentaire. Ainsi le syndrome est 1 et l'état suivant est 2 (figure 2.9(b)).
- etc.

Pour résumer, dans l'exemple de la figure 2.10, les valeurs du quantificateur 7, 2 et 1 sont utilisées par le coset de treillis complémentaire et les autres par le coset de treillis

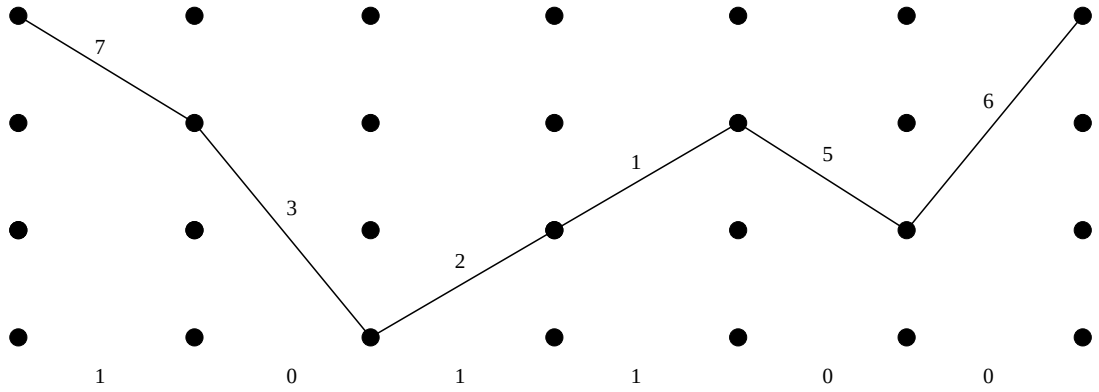

FIG. 2.10 – Exemple de détermination de syndrome à partir de la figure 2.9.

principal.

Selon les débits de transmission et de la quantité de corrélation entre les deux sources X et Y, les performances obtenues avec DISCUS sont à une distance de 2 dB de la limite de Wyner-Ziv [PR03a].

2.3.2 Codeur de Wyner-Ziv utilisant un code turbo

2.3.2.1 Codeur de Wyner-Ziv avec quantificateur de Lloyd-MAX

Dans une deuxième partie de [AG02], Aaron et Girod ont traité le cas de deux sources corrélées à valeurs continues X et Y. Le modèle de corrélation entre les deux sources est présenté par $Y = X + Z$, avec Z un bruit Gaussien indépendant de X. Avant le codage, la source X est quantifiée en utilisant un quantificateur de Lloyd-Max à 2^M niveaux. Par la suite, la séquence à la sortie du quantificateur est codée et décodée avec un code turbo de la même manière qu'avec des sources binaires à l'exception du calcul des probabilités qui utilise une fonction de densité de probabilité à valeurs continues $f(y|x)$. Le schéma du codeur/décodeur pour des sources Gaussiennes et distribuées utilisé dans les travaux de Aaron et Girod est illustré dans la figure 2.11.

L'avantage de la solution introduite par Aaron et Girod est d'avoir les mêmes performances en termes de probabilités de symboles erronés ($Pr(X_Q \neq \hat{X}_Q)$) qu'avec DISCUS pour des corrélations plus faibles entre les deux sources X et Y (3 dB de gain en corrélation).

2.3.2.2 Codeur de Wyner-Ziv utilisant un quantificateur de Lloyd-MAX avec information adjacente

Dans [RMZG03], le schéma du codeur/décodeur pour le codage distribué de deux sources Gaussiennes est le même que celui de [AG02] présenté dans la figure 2.11, sauf qu'ici la quantification est basée sur un quantificateur de Lloyd-Max avec information adjacente. Les auteurs dans [RMZG03] supposent que la fonction de densité de probabi-

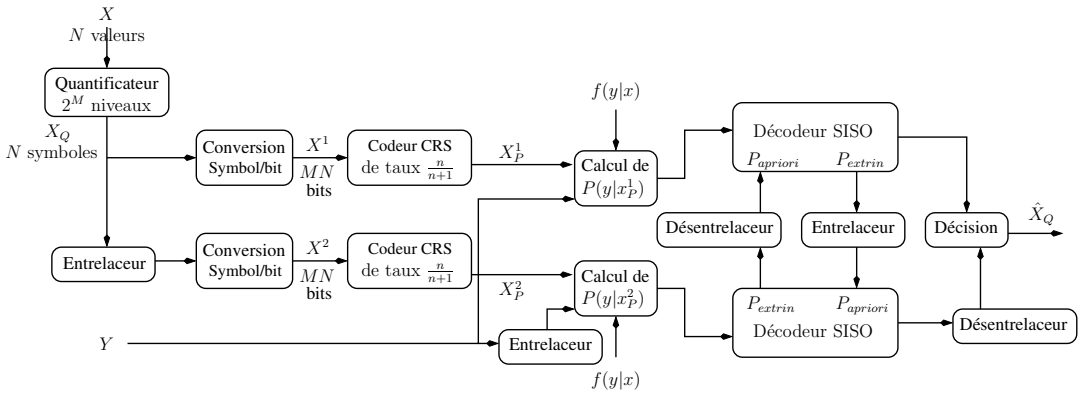

FIG. 2.11 – Schéma du codeur/décodeur pour des sources continues et distribuées utilisé dans [AG02].

lité $f(y|x)$ est connue au codeur et au décodeur. L'information de bord Y est disponible seulement au décodeur. L'idée est de trouver un quantificateur qui réduit la distorsion moyenne (EQM) $D = E[(X - \hat{X})^2]$. Autrement dit, il s'agit de trouver les meilleures partitions.

Soit X une Gaussienne qui prend ses valeurs dans un intervalle $[x_{inf}, x_{sup}]$. Il s'agit de trouver la meilleure partition de l'intervalle $[x_{inf}, x_{sup}]$ en M sous intervalles $[x_{i-1}, x_i]$, pour tout $i = 1, 2, ...M$ (avec $x_0 = x_{inf}$ et $x_M = x_{sup}$), minimisant la distorsion D. Soit N un entier positif qui représente les indices ou bien les niveaux de quantification avec $\frac{M}{N} \geq 1$. Les N premières partitions de l'intervalle $[x_{inf}, x_{sup}]$ sont représentées par les indices q, $q = 0, 1, 2, ..., N-1$. Pour les partitions au delà de N, l'indexation par $q = 0, 1, 2, ..., N-1$ se répète jusqu'à atteindre le dernier intervalle. Soit R_q $(q = 0, 1, ..., N-1)$ les régions de quantification correspondant à l'indice q comme l'indique la figure 2.12. Les auteurs dans [RMZG03] ont défini deux fonctions $d(q, x)$ et D_q (appelée distorsion partielle) telles que

$$d(q, x) = E[(x - \hat{x}(q, Y))^2 | X = x] = \int_{-\infty}^{\infty} (x - \hat{x}(q, y))^2 f(y|x) dy \tag{2.9}$$

$$D_q = \int_{R_q} d(q, x) f(x) dx \tag{2.10}$$

avec $\hat{x}(q, y) = E[X = x | Q = q, Y = y] = \frac{\int_{R_q} x f(x|y) dx}{\int_{R_q} f(x|y) dx}$. La distorsion globale est $D = \sum_{q=1}^{N} D_q$.

L'algorithme du quantificateur de Lloyd-Max généralisé est itératif et fonctionne comme suit :

1. le nombre d'itérations k et la distorsion correspondante D_0 sont initialisés respectivement à 1 et à ∞.
2. choisir un quantificateur initial à q niveaux répartis sur M partitions.
3. déterminer les fonctions $d(q, x)$ et la distorsion globale D_k.

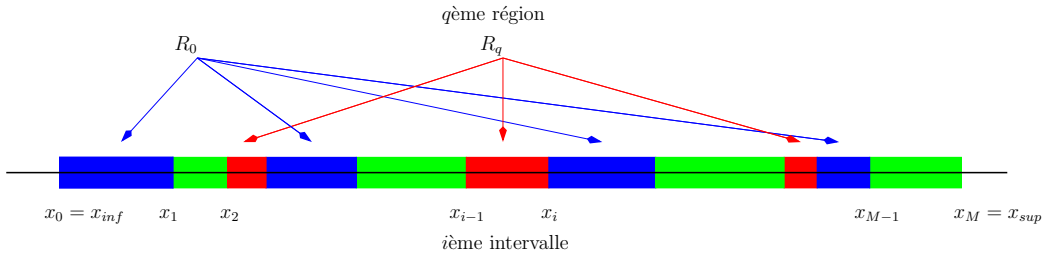

FIG. 2.12 – La répartition des q régions R_q du quantificateur de Lloyd-MAX généralisé.

4. déterminer les nouvelles extrémités x_i, $i = 1, 2, ...M$, des intervalles de quantification qui correspondent aux points d'intersections entre les fonctions $d(q = i, x)$ et $d(q = i + 1, x)$.
5. Si $\frac{D_{k-1}-D_k}{D_k} \geq$ seuil, revenir à l'étape 3.

Lorsque l'algorithme du quantificateur de Lloyd-Max généralisé converge, les centroïdes sont déterminés par les valeurs moyenne des extrémités des M intervalles. Le système de la référence [RMZG03] permet effectivement de réduire la distorsion et par la suite de s'approcher des bornes de Wyner-Ziv (entre 1.83 dB et 1.53 dB selon le débit de transmission).

2.3.3 Codeur de Wyner-Ziv avec un quantificateur en lattice emboîtée

Une paire de lattices (Λ_1,Λ_2) de dimension n avec leurs matrices génératrices correspondantes $\mathbf{G}_1$ et $\mathbf{G}_2$ est dite emboîtée, s'il existe une matrice $\mathbf{P}$ de nombres entiers et de taille $n \times n$ telle que $\mathbf{G}_2 = \mathbf{G}_1\mathbf{P}$ avec $det(\mathbf{P}) > 1$. L'emboîtage consiste à inclure dans les régions de Voronoï de lattice Λ_1, une autre lattice de résolution plus élevée.

Le codeur de Wyner-Ziv utilisant la quantification en lattice emboîtée de Zamir *et al.* [ZS98], [ZSE02] fonctionne comme suit :

- Soit un vecteur $\mathbf{u}$ des variables aléatoires connu au codeur et au décodeur de Wyner-Ziv. $\mathbf{u}$ est uniformément distribué sur les régions de Voronoï de lattice Λ_1.
- Pour une mesure moyenne de distorsion D désirée, soit $\alpha = \sqrt{1 - D/\sigma_N^2}$ comme étant un coefficient d'estimation, avec σ_N^2 la variance du bruit de corrélation entre la source X et l'information de bord Y.
- Au codage, pour une séquence $\mathbf{x}$, le vecteur $\alpha\mathbf{x} + \mathbf{u}$ est quantifié par la lattice Λ_1. Le résultat de quantification est $\mathcal{Q}_{\Lambda_1}(\alpha\mathbf{x} + \mathbf{u}) = \mathbf{x}_{Q_{\Lambda_1}}$. Le vecteur $\mathbf{x}_{Q_{\Lambda_1}}$ est à son tour quantifié en utilisant la lattice Λ_2 pour avoir en sortie $\mathcal{Q}_{\Lambda_2}(\mathbf{x}_{Q_{\Lambda_1}}) = \mathbf{x}_{Q_{\Lambda_2}}$. L'index correspondant à $\mathbf{x}_{Q_{\Lambda_1}} - \mathbf{x}_{Q_{\Lambda_2}}$ sera transmis au décodeur.
- Au décodage, le vecteur $\mathbf{w} = \mathbf{x}_{Q_{\Lambda_1}} - \mathbf{x}_{Q_{\Lambda_2}} - \mathbf{u} - \alpha\mathbf{y}$ est déterminé. La reconstruction de $\mathbf{x}$ est $\hat{\mathbf{x}} = \mathbf{y} + \alpha(\mathbf{w} - \mathcal{Q}_{\Lambda_2}(\mathbf{w}))$, avec $\mathcal{Q}_{\Lambda_2}(\mathbf{w}))$ la version quantifiée de $\mathbf{w}$ obtenue avec la lattice Λ_2.

Ce mécanisme peut être vu comme la concaténation de deux codeurs source et canal. En effet, le quantificateur de lattice Λ_1 joue le rôle d'un codeur de source. Quant à la

lattice Λ_2, elle peut être considérée comme un codeur de canal qui essaye de réduire la probabilité d'erreur de décodage.

Une autre application utilisant un quantificateur en lattice emboîtée dans un codeur de Wyner-Ziv a été réalisée par Servetto dans [Ser00]. Le vecteur $\mathbf{x}$ est codé deux fois par les lattices Λ_1 et Λ_2 pour avoir respectivement $\mathcal{Q}_{\Lambda_1}(\mathbf{x}) = \mathbf{x}_{Q_{\Lambda_1}}$ et $\mathcal{Q}_{\Lambda_2}(\mathbf{x}) = \mathbf{x}_{Q_{\Lambda_2}}$. L'indice de la différence $\mathbf{x}_{Q_{\Lambda_1}} - \mathbf{x}_{Q_{\Lambda_2}}$ est transmis au décodeur. Au décodage, étant donné $\mathbf{x}_{Q_{\Lambda_1}} - \mathbf{x}_{Q_{\Lambda_2}}$, le vecteur le plus proche de l'information de bord $\mathbf{y}$ est considéré comme la version quantifiée de $\mathbf{x}$.

Dans [XLCL03], [LCLX04] et pour des débits élevés, un quantificateur en lattice emboîtée suivi d'un codeur pratique de Slepian-Wolf, codeur LDPC, a été considéré comme étant une solution pour le problème de codage de Wyner-Ziv. Le vecteur $\mathbf{u}$ n'a pas été pris en compte et la valeur de $\alpha \approx 1$. Dans ce cas, le codeur cherche à quantifier d'abord $\mathbf{x}$ à $\mathcal{Q}_{\Lambda_1}(\mathbf{x}) = \mathbf{x}_{Q_{\Lambda_1}}$ et puis $\mathbf{x}_{Q_{\Lambda_1}}$ à $\mathcal{Q}_{\Lambda_2}(\mathbf{x}_{Q_{\Lambda_1}})$. Du fait qu'il y a encore une corrélation entre le vecteur $\mathbf{v} = \mathbf{x}_{Q_{\Lambda_1}} - \mathcal{Q}_{\Lambda_2}(\mathbf{x}_{Q_{\Lambda_1}})$ et l'information de bord $\mathbf{y}$, un codeur LDPC peut être utilisé pour compresser le vecteur $\mathbf{v}$. Au décodage et avec l'information de bord $\mathbf{y}$, le décodeur LDPC est utilisé pour décoder $\hat{\mathbf{v}}$ l'estimée de $\mathbf{v}$. Etant donnée $\hat{\mathbf{v}}$, le vecteur $\mathbf{w} = \hat{\mathbf{v}} - \mathbf{y}$ est calculé et l'estimée de $\mathbf{x}$ est déterminé par $\hat{\mathbf{x}} = \mathbf{y} + \mathbf{w} - \mathcal{Q}_{\Lambda_2}(\mathbf{w}) = \hat{\mathbf{v}} - \mathcal{Q}_{\Lambda_2}(\mathbf{w})$.

Pour un quantificateur en lattice emboîtée de dimension 1, le codeur/décodeur présenté dans [XLCL03], [LCLX04] permet d'atteindre des performances de 1.53 dB loin des limites théoriques de Wyner-Ziv pour des débits entre 4 et 5 bits/symbole. Pour les mêmes débits, les performances obtenues avec un quantificateur en lattice emboîtée de dimension 2 s'approchent davantage en étant à 1.36 dB des limites théoriques.

2.4 Codage multiterminal de deux sources Gaussiennes

En pratique, le premier schéma de mise en œuvre du codage multiterminal de deux sources Gaussiennes a été réalisé par [PR00] pour le cas indirect. Les auteurs dans ce cas utilisent le principe de DISCUS pour compresser les deux sources Y_1 et Y_2 (les observations bruitées de X) et reconstruire la source X (voir figure 1.18). Pour un même débit, Pradhan et Ramchandran montrent que les performances du codage multiterminal de deux sources Gaussiennes sont les mêmes que celles obtenues avec un codeur de Wyner-Ziv basé sur DISCUS.

D'autres applications basées sur une quantification TCQ et le codeur LDPC ont été considérées dans [YSXZ04], [YSXZ05] pour les deux cas du codage de sources multiterminal direct et indirect. Dans [YSXZ04], les performances du système sont comparées à celles obtenues avec [PR00]. Un gain de plus de 2 dB est atteint. Les performances du système présenté dans [YSXZ04] sont à 0.5 dB proches des limites théoriques.

Noté que le codage de Wyner-Ziv avec une information de bord partielle (information de bord bruitée) défini dans [Ooh97] est considéré comme un cas particulier de codage multiterminal. Une solution de codage de Wyner-Ziv avec une information de bord partielle a été présentée par Jeanne *et al.* dans [JZGS03a]. L'approche consiste à appliquer la méthode d'Al Jabri et Al-Issa au contexte de deux sources Gaussiennes.

La figure 2.13 décrit le schéma du système de codage/décodage. Les sources corrélées X et Y ($Y = X + Z$) désignent des variables aléatoires Gaussiennes (i.i.d) de moyenne nulle et de variance respectives : l'unité et $1 + \sigma_Z^2$, avec σ_Z^2 la variance de la variable Gaussienne Z (i.i.d) à moyenne nulle. Z est indépendante de X.

Après quantification des deux sources X et Y, les valeurs quantifiées Y_q sont transmises parfaitement alors que celles de X_q sont compressées avec un codeur CLV, basé sur la méthode de Al Jabri et Al-Issa, avant d'être envoyées au décodeur. Au décodage, grâce à X_b (les bits issus du codeur CLV), Y_q et la table de CLV, l'estimée $\hat{X}_q$ peut être trouvée sans ambiguïté. La reconstruction de X, $\hat{X}$, est déterminée en utilisant l'information de bord Y_q et l'estimé $\hat{X}_q$.

Les résultats de simulation du système proposé ont été comparés, au niveau du taux d'erreur des symboles quantifiés et de la distorsion globale, à ceux de Pradhan et Ramchandran obtenus avec la méthode DISCUS. Jeanne *et al.* [JZGS03a], [JZGS03b] ont montré que DISCUS donne de meilleures performances que leur méthode utilisant les CLV. Par contre, cette méthode offre une plus grande souplesse pour le paramétrage de système.

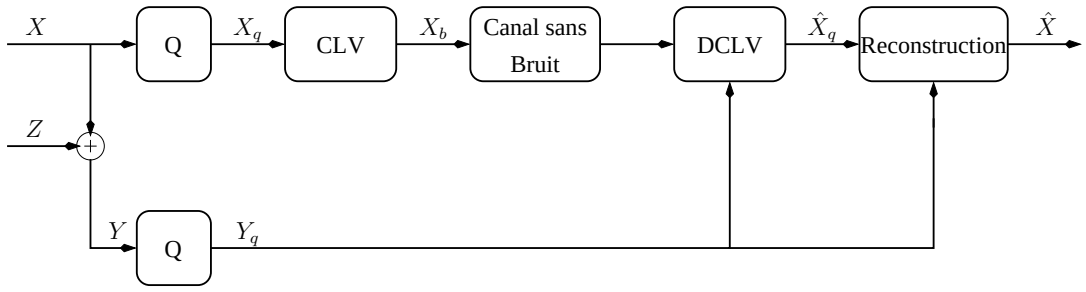

FIG. 2.13 – Schéma de principe des travaux de Jeanne *et al.*

2.5 Conclusion

Différentes solutions de codage distribué de deux sources binaires et Gaussiennes permettant d'approcher les limites théoriques à la fois de Slepian-Wolf, de Wyner-Ziv et de codage multiterminal ont été présentées. Il a été montré que les codes de canal utilisés dans le contexte de codage de sources distribuées fournissent des performances en terme de taux de compression meilleures que celles obtenues avec les codes CLV. Plus les codes de canal sont bons (permettent d'obtenir des performances proches de la limite de Shannon), meilleures sont les performances des codeurs de sources distribuées. En particulier, les codes LDPC permettent de s'approcher davantage de la limite de Slepian-Wolf. Néanmoins, cela nécessite de traiter des séquences d'information de grande longueur avec une complexité supérieure au codage. Ceci nous a amené à utiliser les codes turbo dans nos schémas de codage de sources distribuées.

Dans la plupart des travaux proposés, les schémas de codage de Wyner-Ziv sont

constitués d'un quantificateur concaténé à un code de sources ou un code de canal. Par conséquent, des efforts ont été consacrés à la conception des quantificateurs qui permettent d'améliorer les performances débit-distorsion du système. Pour cela, dans les solutions considérées, certains quantificateurs tiennent compte de la distribution de probabilité conditionnelle de deux sources Gaussiennes. Dans ces contextes, nous proposons d'utiliser la quantification TCQ dans notre schéma de codage de Wyner-Ziv.

Dans le prochain chapitre, nous présenterons les contributions que nous avons apportées dans les schémas de codage distribué de deux sources binaires et Gaussiennes.

Chapitre 3

Codage distribué de 2 sources binaires ou Gaussiennes utilisant le code turbo poinçonné

3.1 Introduction

Dans le chapitre précédent, différents schémas de mise en œuvre du codage distribué de deux sources binaires et Gaussiennes permettant d'approcher les limites théoriques ont été décrits. Dans ce chapitre, nous allons présenter nos contributions au codage de deux sources distribuées.

Nous allons nous intéresser en première partie de ce chapitre, au concept du codage de sources distribuées (CSD) basé sur le code turbo poinçonné. Nous avons évoqué dans le chapitre 2, qu'une première utilisation d'une telle technique a déjà été considérée par Garcia-Frias et Zhao dans [GFZ01b] et [GFZ01a]. La différence ici est la méthode de poinçonnage utilisée. En effet, afin d'augmenter l'efficacité de correction du code turbo et avoir des performances proches des limites théoriques, une nouvelle classe de patrons de poinçonnage est présentée. Une analyse de la propriété des codes turbo poinçonnés dans le contexte du codage de sources distribuées est proposée.

Dans un deuxième temps, nous considérons le cas du codeur de deux sources distribuées utilisant la quantification codée par treillis (TCQ) et le code turbo poinçonné. Le but est d'améliorer les performances débit-distorsion de notre système de codage de Wyner-Ziv. Dans la solution proposée, la faible corrélation entre les bits de chemins du treillis et l'information de bord, nous a conduit à compresser seulement les mots de code issus de la quantification TCQ. Par la suite, nous examinons le problème de codage de Wyner-Ziv avec une information de bord partielle. Nous re-démontrons les limites théoriques et nous proposons des schémas de mise en œuvre basés sur le code turbo poinçonné. Enfin, le codage conjoint source-canal de deux sources binaires et Gaussiennes avec information de bord au décodeur est étudié.

3.2 Schéma de codage distribué de deux sources binaires

Nous considérons dans cette partie, deux sources binaires corrélées X et Y. La corrélation est modélisée par le canal binaire symétrique (CBS) avec une probabilité de transition p. Nous avons $Pr(X_i = Y_i) = 1 - p$ et $Pr(X_i \neq Y_i) = p$. Soient R_X et R_Y les débits de transmission respectifs de X et de Y.

Considérons le schéma de codage distribué de deux sources binaires X et Y utilisant un turbo code poinçonné comme l'indique la figure 3.1. Dans un système de codage de sources distribuées basé sur un code de canal, l'information de bord Y est considérée comme une version bruitée de la source X. La source Y est codée séparément au taux de la valeur théorique de son entropie $R_Y \geq H(Y)$. Pour la source X, seuls les bits de parité X_P^1 et X_P^2 sont envoyés au décodeur à un taux proche de la valeur de l'entropie conditionnelle de $R_X \geq H(X|Y)$. Dans un premier temps, nous supposons que les bits de parité sont transmis sans bruit. Etant donnée la source Y et les bits de parité X_P^1 et X_P^2, le décodeur turbo va estimer la source X.

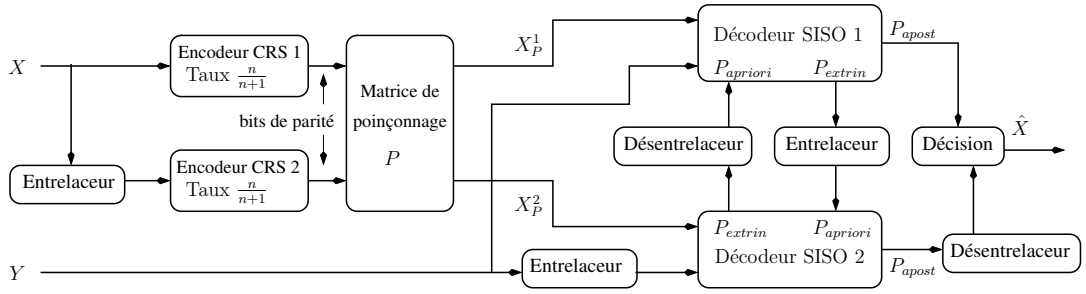

FIG. 3.1 – Système d'un codeur distribué pour 2 sources binaires corrélées en utilisant un code turbo poinçonné.

Considérons un code turbo constitué de deux codeurs convolutionnels récursifs systématiques (CRS) de taux $n/(n+1)$ concaténés en parallèle et séparés par un entrelaceur. Les sorties de chaque codeur sont poinçonnées en utilisant un patron de poinçonnage qui est défini par la matrice $\mathbf{P}$. Le poinçonnage est le processus d'élimination systématique de certains symboles/positions de la séquence codée à la sortie d'un encodeur. De cette façon, le rapport du nombre des bits d'information qui entrent dans le codeur et le nombre des symboles codés qui subsistent après poinçonnage présente un taux résultant supérieur au taux de codage de l'encodeur non poinçonné. Sans poinçonnage, le taux d'un code turbo est $n/(n+2)$. Pour n bits à l'entrée de l'encodeur turbo, seulement $(n+2)$ bits sont à la sortie qui correspondent à n bits systématiques et 2 bits de parité. Pour obtenir un taux de compression élevé, les bits systématiques des deux codeurs élémentaires ne sont pas transmis et quelques bits de parité sont éliminés.

Le poinçonnage permet d'obtenir des taux élevés en utilisant le même encodeur par un simple changement du nombre de bits à éliminer. Cependant, selon la matrice de poinçonnage (patron de poinçonnage), cette technique peut introduire une dégradation de la performance du code turbo due à la diminution des poids des mots de code quand

les bits de parité sont éliminés périodiquement.

Les patrons de poinçonnage classiques sélectionnent périodiquement les mêmes positions des bits de parité retenus. Par exemple, pour avoir 3 : 1 comme taux de compression, on peut utiliser un code turbo formé par deux CRS de taux 1/2 et de longueur de contrainte $K = 3$ suivis par un patron de poinçonnage **P1** :

$$\mathbf{P1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.1}$$

La recherche de la meilleure matrice de poinçonnage avec des taux de compression très élevés n'est pas une simple affaire [HB89]. Une des solutions pour trouver les meilleures positions des bits de parité retenus est la détermination du spectre de poids du code turbo pour différentes combinaisons du patron de poinçonnage. Cependant, pour certains codeurs CRS, les performances du code turbo restent encore trop faibles.

Pour améliorer la propriété de distance ainsi que les performances du code turbo poinçonné, un nouveau patron de poinçonnage a été utilisé (désigné par patron modifié). Dans ce cas, les positions des bits de parité à éliminer changent périodiquement suivant certaines règles (le taux de codage et la longueur de contrainte des codeurs élémentaires, le taux global souhaité ...). Plus de détails sur cette technique sont présentés dans [FKJ99], [Laj01].

Pour l'exemple avec un taux de compression 3 : 1 et un code turbo formé par deux CRS de taux 1/2 et de longueur de contrainte $K = 3$, on peut utiliser le patron suivant :

$$\mathbf{P2}=\left(\begin{array}{cccccc|cccccc|cccccc|c|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \tag{3.2}$$

Le patron de poinçonnage modifié **P2** permet de sélectionner toutes les positions de la réponse impulsionnelle du codeur CRS. La matrice **P2** permet aussi d'avoir un spectre de poids de code turbo meilleur que celui obtenu avec un patron classique **P1** comme l'indique le tableau 3.1 obtenu avec deux CRS en parallèle de taux 1/2 séparés par un entrelaceur aléatoire de longueur 36 [Laj01]. Pour une question de simplicité, la longueur de contrainte des codeurs CRS est $K = 3$. Dans ce tableau, a_d et c_d sont respectivement le nombre des chemins incorrects et le nombre total des bits erronés pour les chemins avec un poids de Hamming d. Le meilleur codeur est celui avec une distance libre d_{free} la plus grande et des valeurs de c_d faible. Si deux codeurs ont la même valeur d_{free}, alors plus la valeur de c_d est faible, meilleur est le code. Le tableau 3.1 montre que la distance libre de code turbo poinçonné utilisant le patron **P2** est plus grande que celle obtenue avec le patron **P1**. Les valeurs de c_d obtenues avec la matrice **P2** sont aussi plus faible que celles obtenues avec **P1**.

La recherche du spectre de poids de code turbo poinçonné (ou non) n'est pas une tache facile à réaliser. Cela est dû à l'existence d'un entrelaceur entre les deux codeurs CRS. Le concept de notre algorithme est extrêmement simple. Pour l'exemple du tableau 3.1, la procédure commence par explorer tous les chemins du treillis du premier codeur CRS jusqu'à une profondeur de 36 branches ou transitions. A chaque transition, nous

Encodeur élémentaire original			Code turbo		
K	Polynôme générateur	R	Patron de poinçonnage	d_{free}	a_d, $d = d_{free}, d_{free}+1, d_{free}+2...$ c_d, $d = d_{free}, d_{free}+1, d_{free}+2...$
3	$h_0 = 7$ $h_1 = 5$	1/2	**P1**	4	1,6,29,104,337,1162,3776,11338 2,16,94,409,1582,6222,22952,77145
			code turbo poinçonné		
			P2	5	1,2,32,206,1014,4183,14495,41712 3,7,128,960,5480,25617,99676,318659

TAB. 3.1 – Spectre de poids du code turbo utilisant un CRS de taux 1/2, de polynôme générateur h_0 et h_1 (en forme octal) et de longueur de contrainte $K = 3$. La longueur de l'entrelaceur aléatoire est 36 et les deux matrices de poinçonnage sont **P1** et **P2**. Le taux global est 3/4.

sauvegardons le poids des bits systématiques et de parité. Si l'état zéro est atteint, alors l'exploration du chemin courant peut être arrêtée et tous les bits systématiques et de parité restants sont mis à zéro jusqu'à une longueur de profondeur de 36. Dans ce cas, les bits systématiques sont entrelacés aléatoirement et la séquence des bits de parité issue du deuxième codeur CRS peut être déterminée. Les poids sont calculés pour les trois séquences : systématique, première parité et deuxième parité. Ce processus se répète pour chaque chemin du treillis. Il est clair que plus la profondeur est grande (taille de l'entrelaceur), plus l'exploration est complexe.

Le décodeur turbo est composé de deux décodeurs SISO concaténés en série via un entrelaceur/désentrelaceur comme l'indique la figure 3.1. L'algorithme de décodage utilisé est le MAP. Chaque décodeur prend en entrée les bits de parité générés par l'encodeur correspondant et l'information de bord Y.

Expérimentalement, les performances du codage de sources distribuées basé sur le code turbo poinçonné utilisant le patron modifié ont été améliorées. Pour un taux de compression 3 : 1, le patron de poinçonnage classique utilisé pour un codeur distribué de deux sources binaires avec un code turbo poinçonné formé par un codeur élémentaire CRS de taux 2/3 et de longueur de contrainte $K = 5$ est :

$$\mathbf{P3} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{3.3}$$

Le patron modifié peut être :

$$\mathbf{P4} = \left(\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \tag{3.4}$$

La figure 3.3 montre le taux d'erreur de la source binaire X en fonction de la probabilité de transition p du canal CBS. Les résultats de simulation sont obtenus avec des codes turbo poinçonnés formés par des codeurs CRS de taux 1/2 et 2/3 (voir figure 3.2)

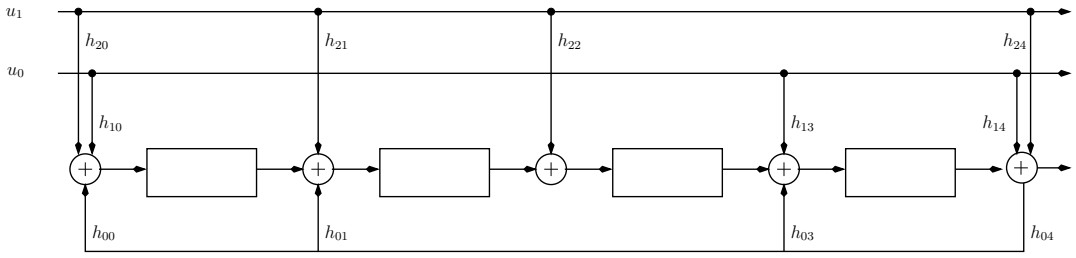

FIG. 3.2 – Codeur convolutionnel récursif systématique (CRS) de rendement 2/3, de longueur de contrainte respectivement $K = 5$ et de polynômes générateurs $(h_0 = 33, h_1 = 23, h_2 = 35)$.

et de longueur de contrainte respectivement $K = 3$ et 5. Les polynômes générateurs de ces codeurs CRS (présentés en base octale) sont respectivement $(h_0 = 7, h_1 = 5)$ et $(h_0 = 33, h_1 = 23, h_2 = 35)$. On peut remarquer sur la figure 3.3 que les performances du codage de sources distribuées utilisant un code turbo poinçonné sont meilleures avec les patrons modifiés que celles obtenues avec des patrons classiques. Cette amélioration des performances est due à une augmentation de la distance libre d_{free} et à un meilleur spectre de poids du code résultant. Par conséquent, il s'avère que les codes avec de bonnes propriétés du spectre de poids peuvent également être de grand intérêt pour le codage de sources distribuées.

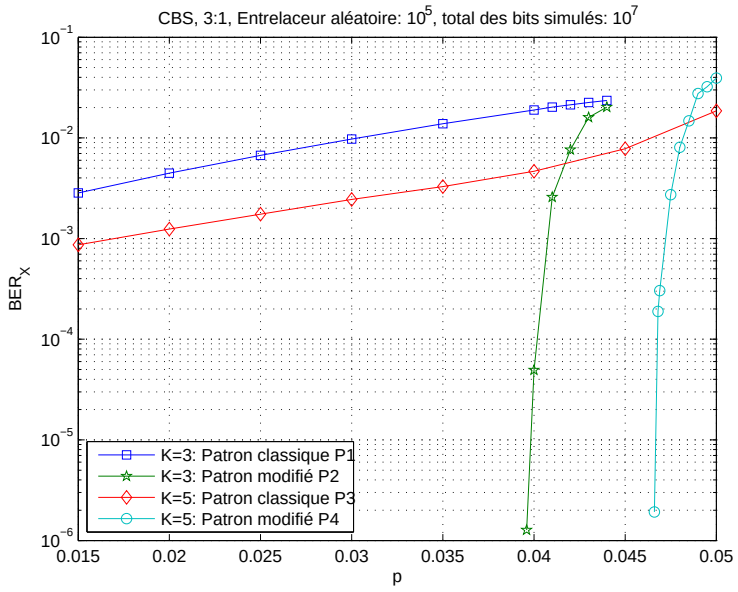

FIG. 3.3 – Performances du codeur distribué de deux sources binaires obtenues avec un code turbo poinçonné basé sur des codeurs CRS de taux 1/2 et 2/3 avec respectivement des longueurs de contrainte $K = 3$ et 5. Le taux de compression est 3 : 1. Les résultats sont obtenus avec une taille d'entrelaceur de 10^5 et après 18 itérations du décodeur turbo de 100 séquences binaires.

La figure 3.4 présente les résultats de simulation obtenus dans [AG02] et [LXG02b]. Les performances de ces travaux se situent respectivement à 0.154 et 0.034 de la limite théorique de Slepian-Wolf, égale à 0.5 pour un taux de compression 2 : 1 de la source X. La taille d'entrelaceur est de 10^5. Nos résultats illustrés sur la figure 3.4, montrent que nous nous situons à 0.0767 de la limite théorique avec un codeur CRS de longueur de contrainte $K = 5$ et un taux de codage 2/3.

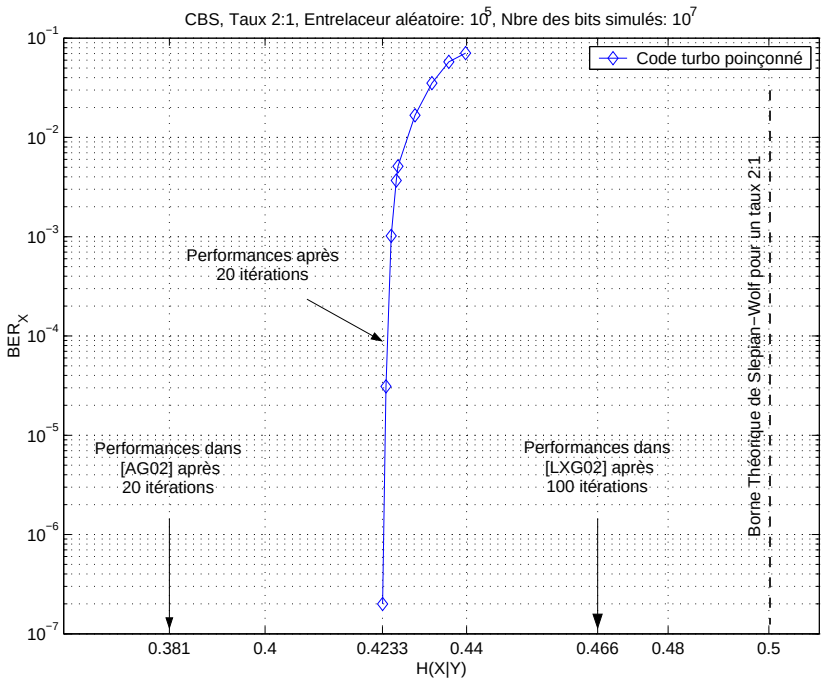

FIG. 3.4 – Performances en terme du taux d'erreur binaire (BER) du codage des sources distribuées (X,Y) où Y est non compressée et X est compressée à 2 : 1 avec un code turbo poinçonné dont le codeur élémentaire est de taux 2/3, de longueur de contrainte $K = 5$ et de taille d'entrelaceur : 10^5. Au total 10^7 bits sont simulés.

Outre un gain en performance par rapport au codeur utilisé par [AG02] (taux de codage 4/5 du codeur élémentaire), le poinçonnage des codes turbo (utilisant un codeur élémentaire initial de taux 2/3 pour avoir 4/5 après poinçonnage) diminue le temps de calcul au codage et au décodage (moins de branches dans le treillis du codeur final). Notre résultat est toutefois de moindre qualité que celui obtenu avec un code LDPC irrégulier qui est aussi, a priori, plus complexe.

Dans le reste du document, le patron modifié est utilisé avec un codeur CRS de taux 2/3 et de longueur de contrainte $K = 5$ pour atteindre des taux de compression de 3 : 1 et 6 : 1.

3.3 Schéma de codage distribué de deux sources Gaussiennes

Soient maintenant T et W deux sources Gaussiennes corrélées et sans mémoire (*i.i.d*). Le modèle de corrélation entre T et W est défini par : $T = W + N_0$ avec N_0 un bruit Gaussien *i.i.d* de moyenne nulle et de variance σ_0^2. N_0 est indépendant de $W \sim \mathcal{N}(0, \sigma_W^2 = 1)$. La variance de la source T est $\sigma_T^2 = \sigma_W^2 + \sigma_0^2$. On définit par Correlation-SNR (CSNR=$\frac{\sigma_W^2}{\sigma_0^2}$) le terme qui représente le rapport des variances de W et N_0.

3.3.1 Codeur de Wyner-Ziv basé sur un quantificateur scalaire et un code turbo poinçonné

Considérons le système de codage de Wyner-Ziv de deux sources Gaussiennes T et W comme l'indique la figure 3.5. La source W est connue au décodeur et désignée par l'information de bord. La source T est quantifiée avec un quantificateur scalaire à 2^M niveaux. Les symboles de la version quantifiée de T (U) sont compressés avec un encodeur turbo constitué de deux codeurs CRS de taux $n/(n+1)$ concaténés en parallèle et séparés par un entrelaceur. Les sorties des codeurs CRS sont poinçonnées par une matrice de poinçonnage comme pour le cas d'une source binaire dans la section 3.2. Seuls quelques bits de parité sont transmis au décodeur.

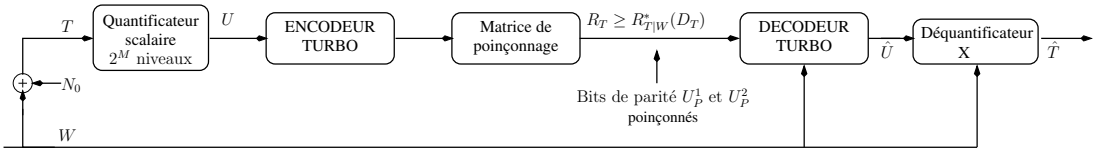

FIG. 3.5 – Codeur de Wyner-Ziv de deux sources Gaussiennes utilisant un quantificateur scalaire et un code turbo poinçonné.

La région des taux de compression admissible dans ce cas est définie par :

$$R_T \geq R^*_{T|W}(D_T) = I(T;U|W) \tag{3.5}$$

$$D_T \geq E[d(T,\hat{T})] \tag{3.6}$$

avec D_T la valeur moyenne de la distorsion théorique entre T et $\hat{T}$ (la valeur reconstruite de T) qui s'exprime par :

$$D_T = \frac{\sigma^2}{2^{2R^*_{T|W}(D_T)}} \tag{3.7}$$

La fonction débit-distorsion de la source T, sous la contrainte que la distorsion moyenne entre T et $\hat{T}$ est inférieure à D_T, est donnée par [WZ76] :

$$R^*_{T|W}(D_T) = \frac{1}{2}\log_2 \frac{\sigma_0^2}{D_T} \tag{3.8}$$

L'estimée $\hat{U}$ est obtenue par un décodage turbo utilisant l'algorithme MAP basé sur une distribution de probabilité à valeurs continues, les bits de parité reçus et l'information de bord W. La métrique de branche de l'algorithme MAP utilise la probabilité $Pr(W|U)$ définie par :

$$\begin{aligned} Pr(W = w|U = u) &= \int_t p(w, t|u)dt \\ &= \int_t p(w|t)p(t|u)dt \end{aligned} \tag{3.9}$$

avec $p(t|u) = \frac{p(u|t)p(t)}{p(u)} = \frac{p(t)}{p(u)}$ nulle en dehors de l'intervalle de quantification $[a, b]$ centré sur u. La probabilité $p(u|t)$ est égale à 1 parce que dans le cas d'une quantification scalaire la connaissance de $T = t$ permet de connaître la valeur prise par $U = u$. La probabilité $p(u)$ est définie par $p(u) = \int_a^b p(t)dt$.

La reconstruction optimale de T peut être déterminée à partir de $\hat{U} \in [a, b]$ et V comme suit :

$$\begin{aligned} \hat{T} &= E(T = t \in [a, b]|\hat{U}, W = w) \\ &= \int_a^b t\ Pr(T = t \in [a, b]|\hat{U}, W = w)dt \\ &= \int_a^b t\ \frac{Pr(\hat{U}|T = t)p(t|w)}{p(\hat{U}|w)}dt \end{aligned} \tag{3.10}$$

Les figures 3.6 et 3.7 montrent les performances en termes respectivement du taux d'erreur symbole ($Pe_T = Pr(U \neq \hat{U})$) et de la distorsion moyenne mesurée de la source T obtenus avec notre codeur de Wyner-Ziv basé sur un codeur turbo poinçonné. La source T est quantifiée en utilisant un quantificateur scalaire de Lloyd-Max à 4 niveaux ($M = 2$). Le terme Correlation-SNR (CSNR=$\frac{1}{\sigma_0^2}$) représente le rapport entre les variances de W et de N_0. Dans la figure 3.6 et pour un débit de 1 bit/symbole, les performances obtenues avec notre système sont meilleures de 1.5 dB par rapport à celles obtenus dans [AG02] pour un taux d'erreur symbole $Pe_T = 10^{-5}$. En effet, pour une même valeur du taux d'erreur symbole, notre solution peut compresser des sources avec moins de corrélation (pour $Pe_T = 10^{-5}$, le CSNR=8 dB dans notre système contre un CSNR=9.5 dB dans [AG02] pour un débit de 1 bit/symbole).

Sur la figure 3.6 on présente les performances de notre codeur de Wyner-Ziv pour un débit de $\frac{2}{3}$ bit/symbole. Il y a une perte en terme de corrélation de presque 4 dB par rapport aux performances obtenues avec un débit de 1 bit/symbole.

La figure 3.7 illustre l'écart entre la distorsion moyenne mesurée obtenue avec notre solution et la borne théorique de Wyner-Ziv. Pour des débits de 1 et de $\frac{2}{3}$ bit/symbole, la distorsion moyenne mesurée est respectivement à 2.46 et 1.84 dB de la limite de Wyner-Ziv. Plus la corrélation entre T et W augmente, plus l'écart entre les distorsions mesurées et théoriques diminue.

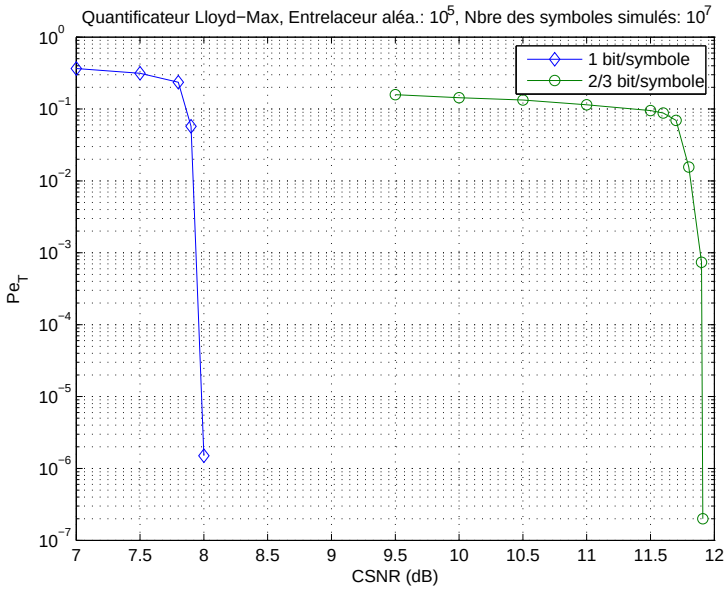

FIG. 3.6 – Taux d'erreur symbole de la source T dans un système de codage de Wyner-Ziv utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. La source T est quantifiée avec un quantificateur de Lloyd-Max à 4 niveaux. Au total 10^7 symboles sont simulés.

3.3.2 Codeur de Wyner-Ziv basé sur la TCQ et le code turbo poinçonné

Afin de s'approcher davantage de la limite théorique de Wyner-Ziv, nous allons remplacer la quantification scalaire uniforme par une TCQ.

3.3.2.1 Principe de la TCQ

La TCQ [MF90] utilise la notion de partitions d'ensemble proposée par Ungerboeck en modulation codée par treillis [Ung82]. Pour obtenir le débit désiré, un codeur convolutif de rendement $\frac{n}{n+1}$ est utilisé pour le partitionnement du dictionnaire de quantification. Pour un débit de transmission R bits/symbole, la TCQ cherche dans ce dictionnaire (comprenant 2^{R+1} mots de codes) la meilleure séquence de mots de code qui minimise la distance euclidienne (c'est-à-dire la distorsion globale) entre la séquence quantifiée et la séquence d'origine. La TCQ est la technique la plus puissante en codage de source parce qu'elle peut atteindre les limites théoriques en terme de débit-distorsion.

La TCQ consiste à partitionner un dictionnaire de quantification initial en sous-dictionnaires complémentaires associés aux transitions entre les états d'un code convolutif.

Soit une source X *i.i.d* à quantifier en utilisant la TCQ. Le taux de compression

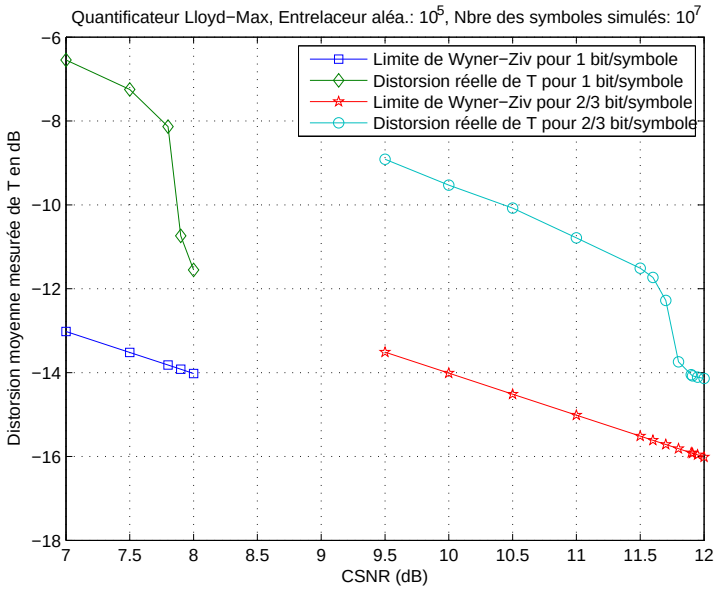

FIG. 3.7 – Distorsion moyenne mesurée pour la source T dans un système de codage de Wyner-Ziv utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux $2/3$, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. La source T est quantifiée avec un quantificateur de Lloyd-Max à 4 niveaux. Au total 10^7 symboles sont simulés.

désiré est de R bits/symbole (le quantificateur TCQ est appelé dans ce cas R-bits TCQ). Considérons un quantificateur scalaire uniforme dont le dictionnaire de quantification D est de cardinal 2^{R+1} et un code convolutif de rendement $1/2$.

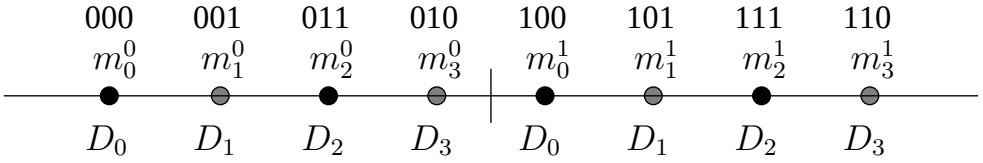

FIG. 3.8 – Exemple de partition pour une TCQ à 2 bits/symbole.

Le dictionnaire du quantificateur scalaire D est partitionné en 4 sous-dictionnaires D_0, D_1, D_2 et D_3 comprenant chacun 2^{R-1} mots de code. Chaque sous-dictionnaire représente une transition dans le treillis du code convolutif. Ces sous-dictionnaires sont désignés également sous le nom de cosets. $B_0 = D_0 \cup D_2$ et $B_1 = D_1 \cup D_3$ sont appelés des super-dictionnaires. Lors de la quantification d'une source X et à chaque instant t, seul B_0 ou B_1 est disponible suivant l'état sur le treillis du code convolutif.

Par exemple, considérons la partition de la figure 3.8 correspondant au débit $R = 2$

bits/symbole (2-bits TCQ) et le treillis d'un codeur convolutif de rendement 1/2 à la figure 3.9 (les branches en traits discontinus signifient que le bit inséré est 0 alors que les branches en traits continus correspondent à 1). Sur la figure 3.8, les mots de code sont désignés par m_i^j, avec $i = 0, 1, 2, 3$ appelé indice de coset ou du sous-dictionnaire D_i et $j = 0, 1 = 2^{R-1} - 1$ appelé indice du mot de code. Ces mots de code sont étiquetés de gauche à droite par les sous-dictionnaires D_0, D_1, D_2 et D_3. Dans la figure 3.9, on peut remarquer que pour chaque état du treillis il y a deux branches qui entrent ou sortent, assignées par des sous-dictionnaires d'un même super-dictionnaire, B_0 ou B_1.

Etant donnée la source X, le décodeur de Viterbi est utilisé pour chercher la version quantifiée X_Q située à une distance euclidienne minimale de la source X, c'est-à-dire avec une erreur quadratique la plus faible. Pour cela, à chaque instant t, une métrique de branche $d_i(t) = \min[(x(t) - m_i^0)^2, (x(t) - m_i^1)^2]$ est assignée à chaque branche du treillis étiquetée par D_i, avec $x(t)$ la valeur de la source X prise à l'instant t. L'algorithme de Viterbi va chercher à sélectionner le chemin optimal à travers le treillis minimisant la métrique globale de toutes les branches.

A la sortie du quantificateur TCQ, la séquence X_Q est représentée par deux séquences. La première séquence, désignée par Ct_X (appelée chemin du treillis), est constituée de bits spécifiant le chemin optimal du treillis correspondant à la séquence de sous-dictionnaires à la fin du décodage de Viterbi. Quant à la deuxième séquence (appelée mots de code et désignée par Mc_X), elle est composée, à chaque instant, des indices j des mots de code de longueur $R-1$ bits appartenant aux sous-dictionnaires indexés par le bit de transition.

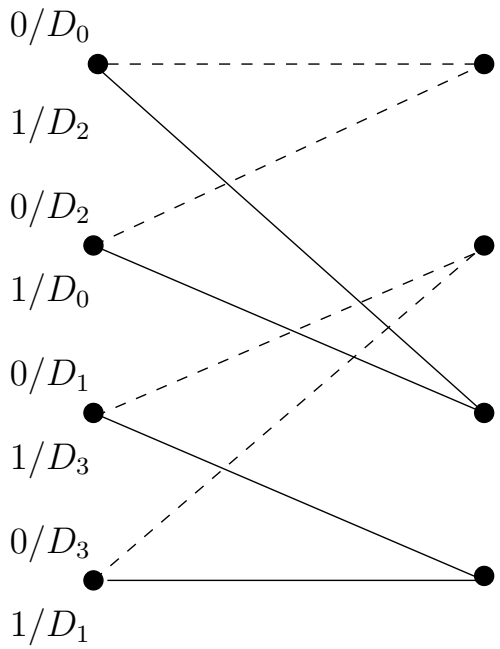

FIG. 3.9 – Treillis d'un code convolutif de rendement 1/2 associé à la partition de la figure 3.8.

La reconstruction de la source X s'effectue comme l'indique la figure 3.10 en deux étapes. D'abord, la première séquence Ct_X (chemin du treillis) est codée en utilisant le codeur convolutif de rendement $1/2$ pour récupérer la séquence de sous-dictionnaires appropriés D_i (désignée par Cc_X). Chaque groupe de deux bits en sortie du codeur convolutif correspond à un indice de coset i. Ensuite, les bits de la deuxième séquence Mc_X sont utilisés pour sélectionner le mot de code m_i^j, c'est-à-dire $\hat{X} = X_Q$ comme l'indique la figure 3.10.

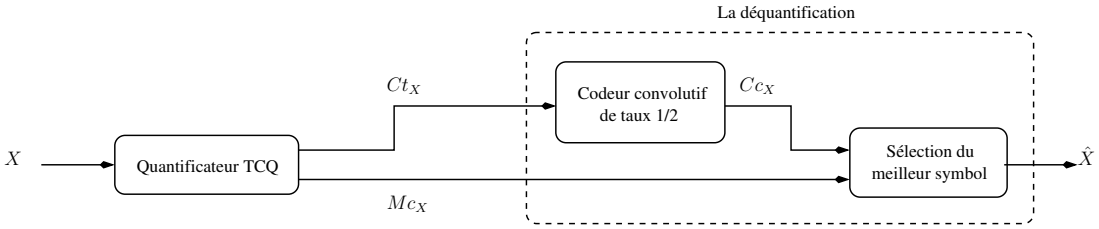

FIG. 3.10 – Quantification et déquantification d'une source X en utilisant la TCQ.

La quantification TCQ peut être généralisée au cas vectoriel. On parle alors de quantification vectorielle codée par treillis (TCVQ) [FMW91], [WM92]. La structure d'un codeur TCVQ est similaire à celle d'un système TCQ, avec une augmentation de la complexité des calculs qui est due à la recherche de vecteurs.

Dans notre codeur de Wyner-Ziv qui est basé sur un code turbo poinçonné, nous utilisons seulement un quantificateur TCQ pour préserver une complexité faible à l'encodage.

3.3.2.2 Schéma de codage de Wyner-Ziv avec une quantification TCQ

Dans un codeur de Wyner-Ziv, afin de diminuer la distorsion de la source T et de se rapprocher des bornes théoriques de Wyner-Ziv, nous avons remplacé la quantification scalaire uniforme par une quantification TCQ. On suppose que les valeurs prises par la source W (information de bord) sont connues au décodeur [WZ76]. Nous proposons d'utiliser un quantificateur TCQ (R-bits TCQ) comme indiqué sur la figure 3.11.

Les symboles de la séquence des mots de code Mc_T sont codés avec un codeur turbo. Les sorties de chaque codeur sont poinçonnées. Pour obtenir un taux de compression élevé, les bits systématiques des deux codeurs élémentaires sont éliminés et seulement quelques bits de parité sont transmis au décodeur.

L'entropie conditionnelle $H(Ct_T|W)$ des bits de la séquence Ct_T s'approche de 1 quand le débit R augmente. Par conséquent, et à l'inverse des méthodes proposées dans [PR03a], [CPR03] et [YCXZ03], les bits de la séquence Ct_T sont envoyés sans compression au décodeur.

Au décodage, la séquence des bit Ct_T indiquant le chemin dans le treillis est codée en utilisant le codeur convolutif de rendement $1/2$ pour re-générer la séquence de sous-dictionnaires(Cc_T). Pour décoder les symboles des mots de code et ainsi estimer la

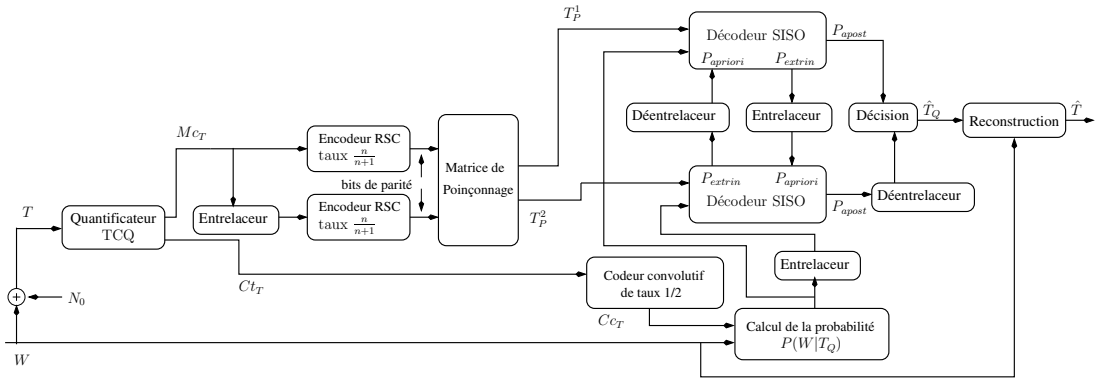

FIG. 3.11 – Schéma d'un codeur de Wyner-Ziv utilisant la TCQ et le code turbo poinçonné.

valeur de $\hat{T}_Q$, un décodeur turbo composé de deux décodeurs SISO est utilisé. Chaque décodeur prend en entrée les bits de parité générés par l'encodeur correspondant et la probabilité $Pr(W|T_Q)$ qui dépend de l'information de bord W et de la séquence Cc_T.

En fait, la probabilité $Pr(W|T_Q)$ s'exprime par :

$$\begin{aligned} Pr(W = w|T_Q = t_q) &= \int_t p(w,t|t_q)dt \\ &= \int_t p(w|t)p(t|t_q)dt \end{aligned} \tag{3.11}$$

La probabilité $Pr(T|T_Q) = \frac{Pr(T_Q|T)Pr(T)}{Pr(T_Q)}$ est facile à déterminer pour un quantificateur scalaire. Dans le cas d'une quantification TCQ, la probabilité $Pr(T_Q|T)$ est différente de 1 car pour une valeur donnée prise par X dans un intervalle de quantification $[a,b]$ d'un sous-dictionnaire de la figure 3.8, la version quantifiée m_i^j, qui dépend du chemin optimal trouvé par l'algorithme Viterbi, peut appartenir à un autre intervalle (différent de $[a,b]$). Pour calculer la probabilité désirée, un grand nombre de séquences d'entraînement de $T \sim \mathcal{N}(0,\sigma_T^2)$ sont quantifiées avec un quantificateur TCQ. Pour chaque mot de code m_i^j, on détermine un histogramme sur les valeurs prises par T comme l'indique la figure 3.12. On remarque sur cette figure (3.12) qu'il y a chevauchement entre les intervalles de quantification. Pour un mot de code m_i^j, les nouveaux intervalles de quantification désignés par $[c,d]$ (déduits à partir de la figure 3.12) sont plus larges que ceux donnés par les sous-dictionnaires initiaux. La figure 3.12 montre aussi que la probabilité $Pr(T|T_Q)$ est non Gaussienne surtout pour les mots de code m_0^0 et m_3^1.

Lors du décodage et à la différence de [YCXZ03], les régions d'intégration $[c,d]$ dans notre cas ne sont que celles qui contiennent les mots de code m_i^j où les valeurs de i sont déterminées par la séquence Cc_T obtenue après le codeur convolutif. En effet, pour améliorer les performances du décodage turbo et en présence de l'information de bord W, nous profitons de la disponibilité de la séquence Cc_T qui a été compressée sans

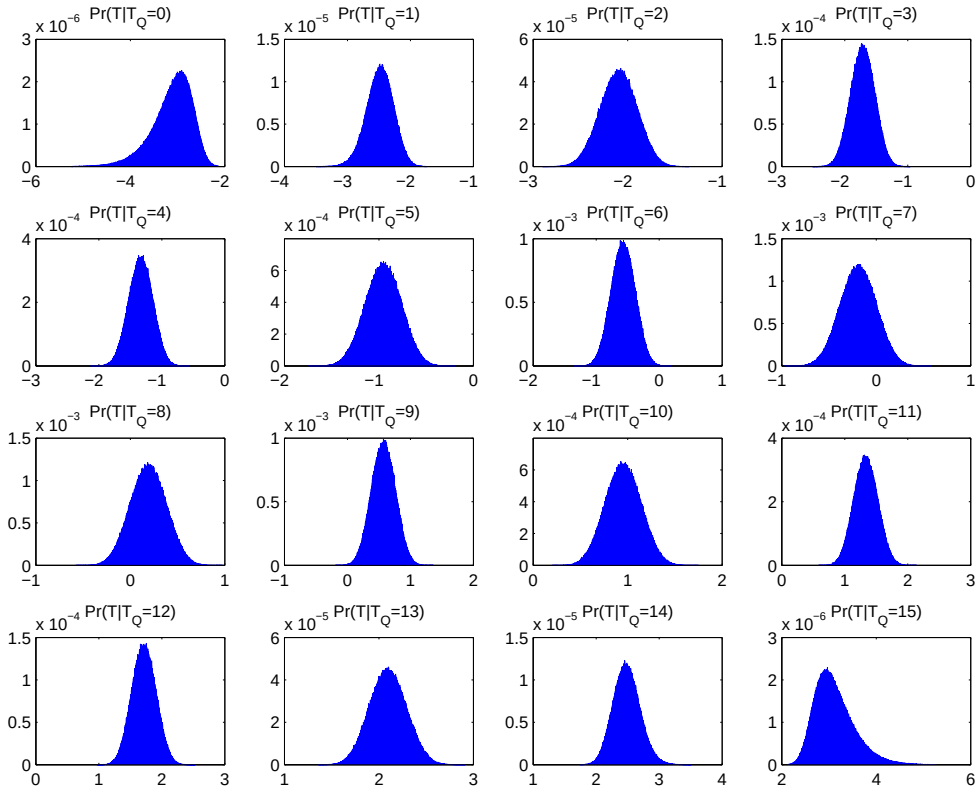

FIG. 3.12 – Les probabilités $Pr(T|T_Q = m_i^j)$ obtenues après quantification d'une source Gaussienne T avec 3-bits TCQ utilisant le treillis d'un code convolutif de rendement $1/2$ et de longueur de contrainte $K = 9$. La variance de T est $\sigma_T^2 = 0.28$.

perte. Par exemple, pour une source T quantifiée avec 3-bits TCQ utilisant le treillis d'un code convolutif de rendement $1/2$, si à l'instant t le symbole constitué par les deux bits de la séquence Cc_T est l'indice du coset $i = 0$, alors le décodeur turbo va concentrer la recherche du mot de code seulement parmi tous les m_0^j (il y a quatre possibilités selon la valeur de $j = 0, 1, 2, 3$).

La reconstruction optimale de T peut être déterminée à partir de $\hat{T}_Q \in [c, d]$ et W comme suit :

$$\begin{aligned} \hat{T} &= E(T = t|\hat{T}_Q, W = w) \\ &= \int_a^b t\ Pr(T = t \in [a, b]|\hat{T}_Q, W = w)dt \\ &= \int_a^b t\ \frac{p(\hat{T}_Q|t)p(t|w)}{p(\hat{T}_Q|w)}dt \end{aligned} \tag{3.12}$$

Pour calculer la probabilité $Pr(\hat{T}_Q|T)$, nous utilisons les lois $Pr(T|T_Q = m_i^j)$ illustrées à la figure 3.12.

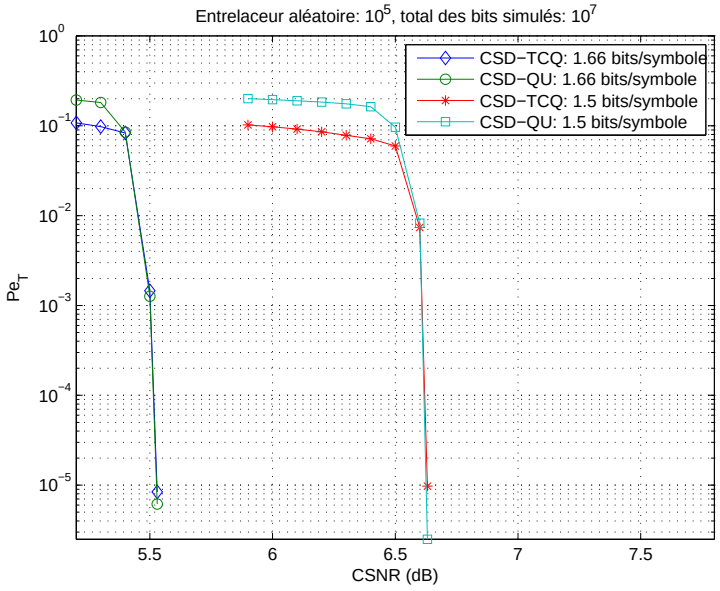

FIG. 3.13 – Taux d'erreur symbole obtenu en sortie du codeur de Wyner-Ziv basé sur la TCQ et le code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux $2/3$, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. La source T est quantifiée avec un quantificateur 3-bits TCQ . Au total 10^7 symboles sont simulés.

Sur la figure 3.13, le taux d'erreur symbole $Pe_T = Pr(T_Q \neq \hat{T}_Q)$ est tracé en fonction de la Correlation-SNR (CSNR) pour deux codeurs de Wyner-Ziv utilisant respectivement un quantificateur scalaire uniforme (QU) (le codeur est désigné par CSD-QU) et un quantificateur 3-bits TCQ (le codeur de Wyner-Ziv est désigné par CSD-TCQ). Les

performances obtenues avec des débits de 1.5 et 1.6 bits/symbole peuvent être observées sur la figure 3.13. Le code turbo poinçonné utilise un codeur CRS de taux de codage 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$. L'entrelaceur utilisé est aléatoire et de taille 10^5. Enfin, le nombre de symboles simulés est de 10^7.

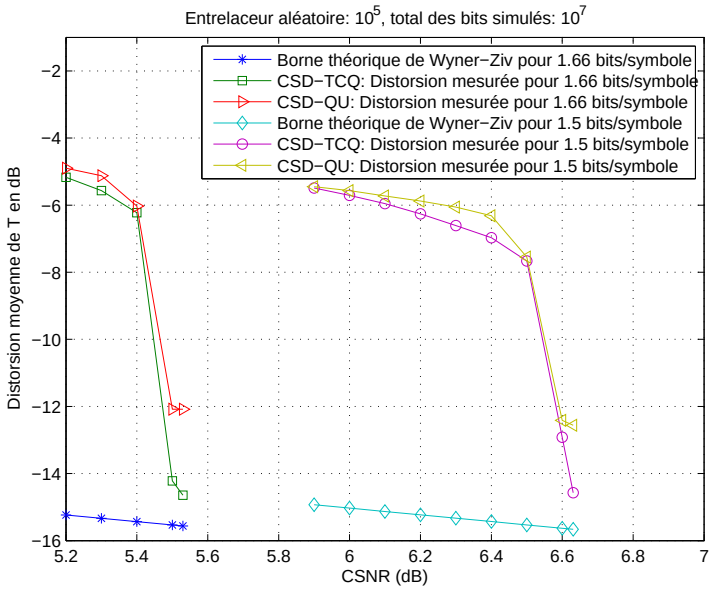

FIG. 3.14 – Distorsion moyenne mesurée d'un codeur de Wyner-Ziv basé sur la TCQ et le code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. La source T est quantifiée avec un quantificateur 3-bits TCQ . Au total 10^7 symboles sont simulés.

Pour des débits de 1.5 et 1.66 bits/symbole, la figure 3.14 montre les valeurs des distorsions moyennes mesurées de la source X obtenues avec les deux codeurs CSD-QU et CSD-TCQ. Pour un débit de 1.66 bits/symbole, on peut voir que la distorsion moyenne de T dans un CSD-TCQ basé sur la TCQ se situe à une distance de 0.916 dB de la borne théorique de Wyner-Ziv. Cependant, pour le CSD-QU la distance entre les distorsions théorique et mesurée est de l'ordre de 3.48 dB. Pour des débits plus faibles (1.5 bits/symbole), la différence entre la distorsion mesurée et la borne théorique diminue avec le CSD-TCQ. Alors que pour le CSD-QU, une petite amélioration de la distorsion peut être observée. Néanmoins, le CSD-TCQ s'approche mieux de la borne théorique que le CSD-QU (pour 1.5 bits/symbole, les performances de CSD-TCQ sont à une distance de 1.092 dB de la limite de Wyner-Ziv alors qu'elles sont à une distance de 3.09 dB avec le CSD-QU).

A un débit égal à 1.72 bits/symbole, la distorsion mesurée d'un codeur de Wyner-Ziv utilisant la TCQ et un codeur LDPC est à une distance de 0.92 dB de la borne théorique [YCXZ03]. Pour un débit plus faible (1.66 bits/symbole), notre solution pré-

sente une meilleure performance débit-distorsion. Par contre, pour des débits supérieurs à 2 bits/symboles, nous avons remarqué que le schéma de codage dans [YCXZ03] permet d'avoir des performances débit-distorsion meilleures que celles obtenues avec notre système.

3.3.3 Schéma de codage distribué de deux sources Gaussiennes quantifiées

Dans des systèmes pratiques (par exemple un schéma de communication radiomobile) l'information de bord W ne peut pas être connue parfaitement au décodeur. Une quantification est nécessaire pour pouvoir représenter en un nombre fini de bits la source W. Par conséquent, la version de l'information de bord reçue par le décodeur présente une distorsion non nulle par rapport à la source W.

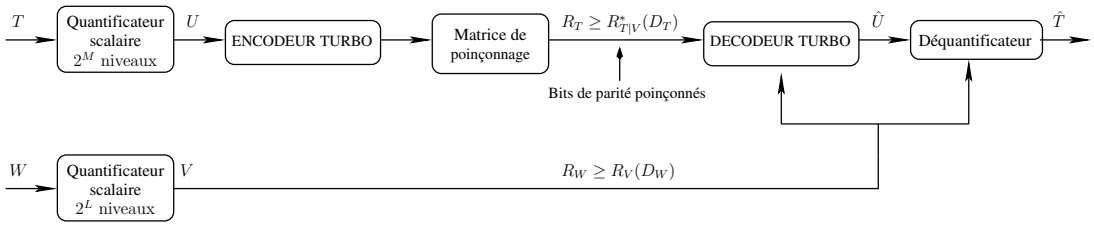

FIG. 3.15 – Système de codage distribué de deux sources quantifiées utilisant un code turbo poinçonné.

Dans cette section, nous allons étudier et analyser le problème du codage distribué de deux sources Gaussiennes corrélées T et W quantifiées par deux quantificateurs de Lloyd-Max différents comme l'indique la figure 3.15. Ce problème correspond au codage de Wyner-Ziv avec une information de bord partielle dont les bornes débit-distorsion ont été dérivées par Oohama dans [Ooh97]. Notre contribution est d'une part de dériver d'une autre manière ces bornes théoriques et d'autre part de proposer une mise en œuvre d'un schéma de codage de Wyner-Ziv avec une information de bord partielle.

La structure de l'encodeur/décodeur turbo est la même que celle utilisée dans la partie 3.2. La source W est quantifiée en utilisant un quantificateur de Lloyd-Max à 2^L niveaux. Tous les bits à la sortie du quantificateur sont transmis sans compression. Par conséquent, le débit de transmission de W est L bits/symbole. Dans ce cas, la séquence de symboles V (la version quantifiée de W) est considérée comme l'information de bord partielle qui est différente de W. La source T est quantifiée avec un quantificateur de Lloyd-Max à 2^M niveaux. Par la suite, les symboles quantifiés U (la version quantifiée de T) sont codés avec un codeur turbo et seulement les bits de parité sont transmis au décodeur. Au décodeur turbo, la séquence des symboles $\hat{U}$ est estimée en utilisant les bits de parité ainsi que l'information de bord partielle V. La reconstruction optimale

de T peut être déterminée à partir de $\hat{U} \in [e, f]$ et de V par :

$$\begin{aligned} \hat{T} &= E(T = t \in [e, f] | \hat{U}, V = v) \\ &= \int_e^f t \, Pr(T = t \in [e, f] | \hat{U}, V = v) dt \\ &= \int_e^f t \, \frac{p(\hat{U}|t)p(t|v)}{p(\hat{U}|v)} dt \end{aligned} \tag{3.13}$$

La région des taux de compression admissible, dans ce cas, est définie par :

$$R_T \geq R^*_{T|V}(D_T) = I(T; U|V) \tag{3.14}$$
$$R_W \geq R_V(D_W) = I(W; V) \tag{3.15}$$
$$D_T \geq E[d(T, \hat{T})] \tag{3.16}$$
$$D_W \geq E[d(W, V)] \tag{3.17}$$

avec D_T et D_W les valeurs moyennes de la distorsion théorique respectivement entre T et $\hat{T}$, et W et V.

Pour déterminer la fonction débit-distorsion $R^*_{T|V}(D_T)$, le système de codage de Wyner-Ziv avec l'information de bord partielle peut être modélisé par une chaîne de Markov comme l'indique la figure 3.16. Des représentations en chaîne de Markov du codeur de Wyner-Ziv avec toute l'information de bord ont été utilisées dans [PR03a], [CX04].

On suppose que le modèle de la figure 3.16 est le suivant : les sources W et $T = W + N_0$ sont deux sources Gaussiennes corrélées sans mémoires ($i.i.d$), avec N_0 un bruit Gaussien $i.i.d$ de moyenne nulle et de variance σ_0^2. N_0 est indépendant de $W \sim \mathcal{N}(0, \sigma_W^2 = 1)$. Soient V et U deux variables aléatoires Gaussiennes telles que : $V \rightarrow W \rightarrow T \rightarrow U$ forme une chaîne de Markov. Soient $U = T + Q_T$ et $W = V + Q_W$, avec $Q_T \sim \mathcal{N}(0, \sigma_{Q_T}^2)$ et $Q_W \sim \mathcal{N}(0, \sigma_{Q_W}^2)$ deux variables aléatoires Gaussiennes $i.i.d$ supposées indépendantes entre elles et indépendantes de T, V et N_0 comme l'indique la figure 3.16. Nous avons $\sigma_{Q_W}^2 = D_W$. Par conséquent, la distorsion théorique D_W peut s'exprimer par :

$$D_W = \sigma_W^2 / 2^{2R_V(D_W)} = 1/2^{2R_V(D_W)} \tag{3.18}$$

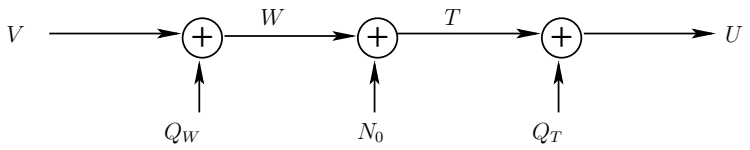

FIG. 3.16 – Schéma du codage de source distribuée avec une information de bord partielle.

L'expression de la fonction débit-distorsion de la source T peut s'écrire sous la forme

suivante :

$$\begin{aligned} R^*_{T|V}(D_T) &= I(T;U|V) = I(T;U) - I(V;U) \\ &= H(U) - H(U|T) + H(U|V) - H(U) \\ &= -H(T+Q_T|T) + H(T+Q_T|V) \\ &= -H(Q_T) + H(V+Q_W+N_0+Q_T|V) \\ &\simeq \frac{1}{2}\log_2(2\pi e(\sigma_0^2+\sigma_{Q_T}^2+\sigma_{Q_W}^2)) - \frac{1}{2}\log_2(2\pi e(\sigma_{Q_T}^2)) \\ &\simeq \frac{1}{2}\log_2(\frac{\sigma_0^2+D_W+\sigma_{Q_T}^2}{\sigma_{Q_T}^2}) \end{aligned} \tag{3.19}$$

Le terme $\sigma_{Q_T}^2$ peut être dérivé de la façon suivante : l'estimation sous optimale de la source T peut s'exprimer par des combinaisons convexes de U et V par $\hat{T} = aV + bU$. Vu que $E[(T-\hat{T})V] = 0$, $E[(T-\hat{T})U] = 0$, il vient :

$$1 - a - b = 0 \tag{3.20}$$

$$(1-b)D_W + (1-b)\sigma_0^2 - b\sigma_{Q_T}^2 = 0 \tag{3.21}$$

Par la suite, nous obtenons :

$$a = 1 - b = \frac{\sigma_{Q_T}^2}{D_W+\sigma_0^2+\sigma_{Q_T}^2} \qquad b = \frac{D_W+\sigma_0^2}{D_W+\sigma_0^2+\sigma_{Q_T}^2}$$

Par conséquent, $\hat{T}$ peut s'écrire :

$$\hat{T} = \frac{\sigma_{Q_T}^2}{D_W+\sigma_0^2+\sigma_{Q_T}^2}V + \frac{D_W+\sigma_0^2}{D_W+\sigma_0^2+\sigma_{Q_T}^2}U \tag{3.22}$$

La distorsion D_T peut être calculée comme suit :

$$\begin{aligned} D_T &= E[(T-\hat{T})^2] = E[((1+b-a)V + (1-b)(Q_W+N_0) - bQ_T)^2] \\ &= (1-b)^2(\sigma_0^2+D_W) + b^2\sigma_{Q_T}^2 \\ &= \frac{(D_W+\sigma_0^2)\sigma_{Q_T}^2}{D_W+\sigma_0^2+\sigma_{Q_T}^2} \end{aligned} \tag{3.23}$$

Dans ce cas, $\sigma_{Q_T}^2$ peut s'exprimer par :

$$\sigma_{Q_T}^2 = \frac{D_T(D_W+\sigma_0^2)}{D_W+\sigma_0^2-D_T} \tag{3.24}$$

Ainsi, la fonction théorique débit-distorsion pour la source T peut s'écrire par :

$$R^*_{T|V}(D_T) = \frac{1}{2}\log_2(\frac{\sigma_0^2+D_W}{D_T}) \tag{3.25}$$

Dans ce cas, les valeurs de a et b deviennent :

$$a = \frac{1}{2^{2R^*_{T|V}(D_T)}}, \qquad b = 1 - a$$

Finalement, la distorsion théorique D_T est donnée par :

$$D_T = \frac{D_W + \sigma_0^2}{2^{2R^*_{T|V}(D_T)}} \tag{3.26}$$

L'équation (3.25) obtenue avec une façon de dérivation différente, s'avère être identique à celle donnée par Oohama dans [Ooh97]. En effet, soit $\rho = \frac{cov(T,W)}{\sqrt{\sigma_T^2 \sigma_W^2}} = \sqrt{\frac{\sigma_W^2}{\sigma_0^2+\sigma_W^2}}$ la corrélation entre T et W, avec σ_T^2 la variance de la source T. Dans ce cas, la fonction débit-distorsion de la source T donnée par (3.25) peut être exprimée, comme il apparaît dans le théorème 1 de [Ooh97], par :

$$R^*_{T|V}(D_T) = \frac{1}{2}\log_2[\frac{\sigma_T^2}{D_T}(1 - \rho^2 + \rho^2 2^{-2R_V(D_W)})] \tag{3.27}$$

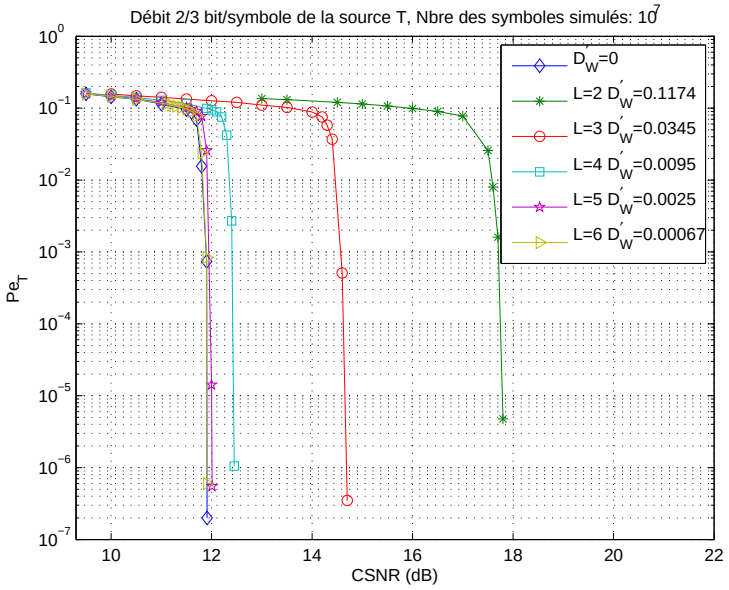

FIG. 3.17 – Taux d'erreur symbole de la source T dans un système de codage de Wyner-Ziv avec une information de bord partielle V utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. Les débits sont de $\frac{2}{3}$ bit/symbole pour T et de L bit/symbole pour W. Au total 10^7 symboles sont simulés.

Les figures 3.17 et 3.18 montrent l'impact de la distorsion mesurée (désignée par D'_W), provoquée par la quantification de W, sur les performances de décodage de la

source T. Les résultats de simulation sont comparés à ceux obtenus avec un système de codage de Wyner-Ziv où W n'est pas quantifiée et disponible au décodeur. La source T est quantifiée en utilisant un quantificateur de Lloyd-Max à 4 niveaux ($M = 2$). Le terme Correlation-SNR (CSNR=$\frac{1}{\sigma_0^2}$) représente le rapport de variances de W et N_0.

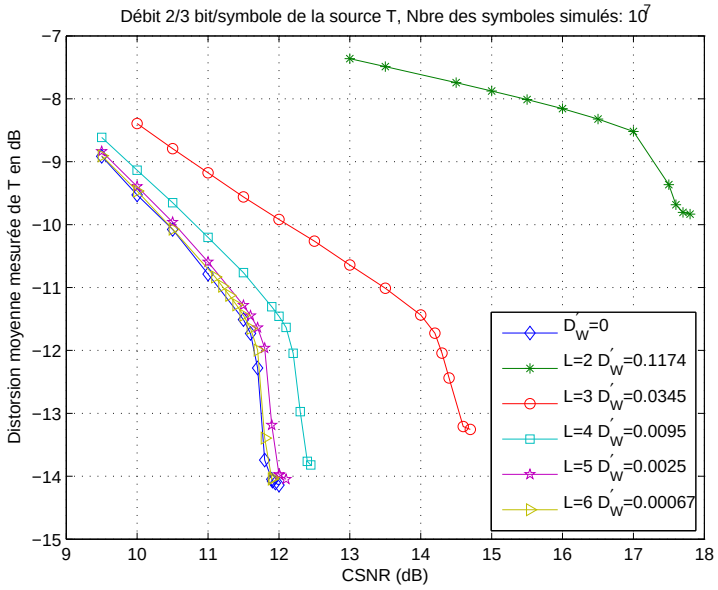

FIG. 3.18 – Distorsion moyenne mesurée pour la source T dans un système de codage de Wyner-Ziv avec une information de bord partielle V utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. Les débits sont de 2/3 bit/symbole pour T et de L bit/symbole pour W. Au total 10^7 symboles sont simulés.

On peut observer, à partir des figures 3.17 et 3.18, qu'il y a une perte de performance de décodage pour la source T en termes respectivement du taux d'erreur symbole (Pe_T) et de la distorsion moyenne mesurée. Plus la distorsion D'_W est faible, meilleure est la performance de décodage de la source T avec moins de corrélation. Pour des distorsions D'_W inférieures à 0.001 (-30 dB), une performance identique à celle obtenue avec un système de codage de Wyner-Ziv où W n'est pas quantifiée, peut être réalisée. Le tableau 3.2 résume les performances débit-distorsion de la source T. Les valeurs du CSNR dans ce tableau correspondent aux points sur les différentes courbes de la figure 3.17 pour lesquelles le taux d'erreur symbole $Pe_T \sim 10^{-5}$. La différence entre les distorsions théoriques et mesurées de la source T diminue avec un débit de W élevé.

Cette étude, nous a permis d'évaluer la distorsion maximale sur l'information de bord au-delà de laquelle les performances d'un système de codage de sources distribuées chutent.

Source W			Source T : débit de $\frac{2}{3}$ bits/symboles et $\text{Pe}_T \sim 10^{-5}$			
Débit (bits/symbole)	Distorsion théorique	Distorsion mesurée	CSNR en dB	Distorsion théorique	Distorsion mesurée	Différence
2	-12.04	-9.3	17.8	-15.03	-9.83	5.2
3	-18.06	-14.62	14.6	-16.99	-13.21	3.78
4	-24.08	-20.22	12.45	-16.18	-13.82	2.36
5	-30.10	-26.02	12.00	-15.95	-13.97	1.98
6	-36.12	-31.74	11.91	-15.90	-14.06	1.84

TAB. 3.2 – Impact de la distorsion de W sur la distorsion de T (les valeurs sont en dB).

3.4 Codage conjoint source-canal pour deux sources corrélées binaires et gaussiennes

Dans cette section, nous considérons le cas où les bits de parité sont transmis au décodeur via un canal bruité modélisé par un canal binaire symétrique sans mémoire. Le cas du codage distribué avec deux sources binaires en présence d'un canal AWGN a été traité par Garcia-Frias dans [GF01].

3.4.1 Cas des sources binaires

Considérons le schéma de codage distribué de deux sources binaires corrélées X et Y de la figure 3.19. La corrélation est modélisée par le canal CBS avec une probabilité de transition p. Nous avons $Pr(X_i = Y_i) = 1 - p$ et $Pr(X_i \neq Y_i) = p$. Soient R_X et R_Y les débits de transmission respectifs de X et de Y.

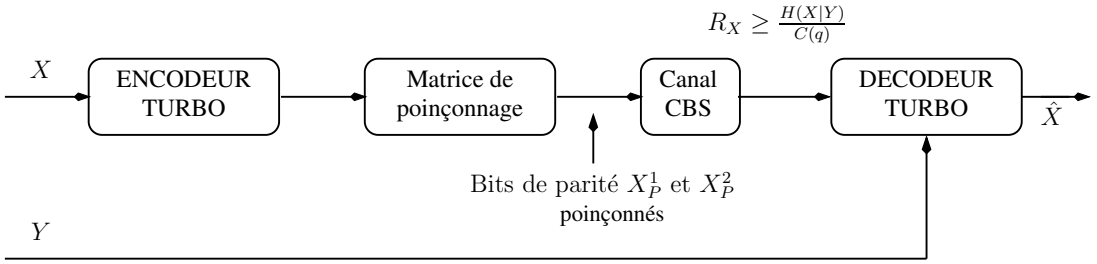

FIG. 3.19 – schéma de codage conjoint source-canal de deux sources binaires corrélées.

Nous essayons d'étendre notre solution au codage conjoint source-canal d'une source binaire X avec une information de bord Y (binaire) connue seulement au décodeur. Supposons que les bits de parité poinçonnés X_P^1 et X_P^2 de la figure 3.19 passent dans un canal binaire symétrique sans mémoire (de probabilité de transition q et de capacité $C(q) = 1 - h(q)$ avec $h(q) = -q\log_2(q) - (1-q)\log_2(1-q)$) et arrivent au décodeur

respectivement sous la forme de mesures bruitées V_P^1 et V_P^2. Le processus de décodage est le même que celui présenté dans la section 3.2, sauf que la métrique de branche de l'algorithme MAP doit inclure la probabilité $Pr(V_P^i|X_P^i)$ pour $i \in \{1,2\}$.

Pour une probabilité de transition $q = 0$, le taux minimal de compression du signal principal X dans un schéma de codage de sources distribuées est $H(X|Y)$. Cependant, en présence du canal bruité de capacité $C(q)$ avec $q \neq 0$, la limite de compression de la source X devient $H(X|Y) \leq C(q)R_X$ [AG02] au lieu de $H(X|Y)$. Par conséquent, en se basant sur le théorème de Slepian-Wolf, la source X peut être transmise à un débit $R_X \geq \frac{H(X|Y)}{C(q)}$ avec un taux d'erreur binaire quasi nul.

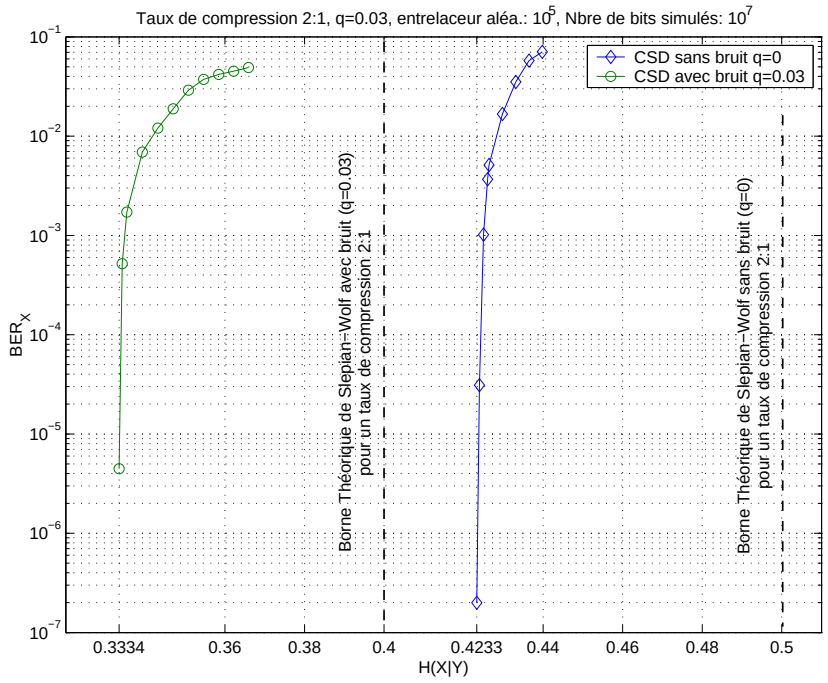

FIG. 3.20 – Performances en terme du taux d'erreur binaire (BER) du codeur conjoint source-canal de sources distribuées (X,Y) où Y est non compressée et X est compressée à $2:1$ avec un code turbo poinçonné dont le codeur élémentaire est de taux $2/3$, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 33, h_1 = 23, h_2 = 35)$. La taille d'entrelaceur est 10^5. Les bits de parité passent dans un canal CBS avec une probabilité de transition $q = 0.03$ et de capacité $C(q) = 0.805$. Les résultats de la figure 3.4 d'un codeur CSD sans bruit $(C(q = 0) = 1)$ sont illustrés.

Pour une valeur de $q = 0.03$, les résultats de simulation pour un codage conjoint source-canal d'une source binaire X avec une information de bord Y (binaire) connue au décodeur sont illustrés sur la figure 3.20. Le taux de compression de l'encodeur turbo est $2:1$. On trouve dans la figure 3.20 les résultats de simulation (figure 3.4) d'un système de codage distribué de deux sources binaires X et Y (X est compressée à $2:1$ et Y à $1:1$) avec $q = 0$ (les bits de parité arrivent au décodeur non bruités). On remarque sur la figure 3.20 que la limite théorique du codage conjoint source-canal (pour $q \neq 0$) est décalée vers la gauche par rapport à la borne classique de Slepian-Wolf. Notre système qui est

basé sur un code turbo poinçonné s'approche mieux de la limite théorique que celui de [AG02]. Cependant, les performances dans [LXG02a] utilisant un codeur conjoint source-canal basé sur des codes systématiques IRA (Irregular Repeat Accumulate) qui sont de la famille des codes LDPC, atteignent cette limite théorique mieux que notre système et celui de [AG02]. La figure 3.20 montre que les performances en terme de débit de notre codeur conjoint source-canal est à une distance de 0.069 bits de la borne théorique qui est égale, pour un $q = 0.03$, à $0.5 \times C(0.03) = 0.5 \times 0.805 = 0.403$ bits. Dans [AG02] et [LXG02a], les résultats de simulation indiquent que les performances en terme de débit sont respectivement à 0.113 bits et 0.059 bits de la limite théorique (égale à 0.403 bits).

3.4.2 Cas des sources Gaussiennes

Considérons maintenant, le schéma de codage conjoint source-canal de deux sources Gaussiennes corrélées et sans mémoire (*i.i.d*), T et W, illustré sur la figure 3.21. La source W est l'information de bord connue seulement au décodeur. Le modèle de corrélation entre T et W est défini par : $T = W + N_0$ avec N_0 un bruit Gaussien *i.i.d* de moyenne nulle et de variance σ_0^2. N_0 est indépendant de $W \sim \mathcal{N}(0, \sigma_W^2 = 1)$. La variance de la source T est $\sigma_T^2 = \sigma_W^2 + \sigma_0^2$. On définit par Correlation-SNR (CSNR=$\frac{\sigma_W^2}{\sigma_0^2}$) le terme qui représente le rapport des variances de W et N_0.

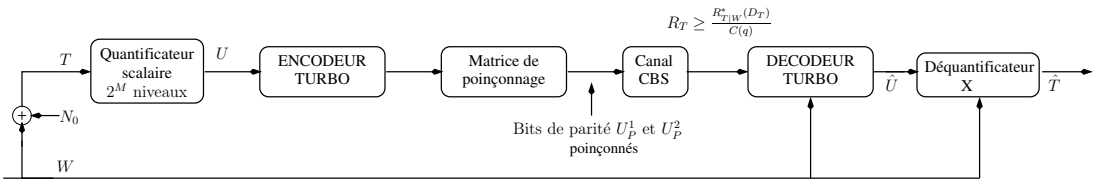

FIG. 3.21 – schéma de codage conjoint source-canal de deux sources Gaussiennes corrélées.

Après quantification en utilisant un Lloyd-Max à 2^M niveaux, la version quantifiée U est compressée par un codeur turbo poinçonné. Les bits de parité retenus, U_P^1 et U_P^2 de la figure 3.21, sont bruités par un canal binaire symétrique sans mémoire de probabilité de transition q et de capacité $C(q) = 1 - h(q)$. Les bits reçus au décodeur sont notés V_P^1 et V_P^2 et correspondent respectivement à U_P^1 et U_P^2. En plus de la probabilité $Pr(W|U)$, la métrique de branche de l'algorithme MAP va utiliser $Pr(V_P^i|U_P^i)$ $(i \in \{1, 2\})$ pour estimer $\hat{U}$.

Compte tenu de la présence du bruit et pour avoir un taux d'erreur symbole faible, le débit de transmission de la source T doit être $R_T \geq \frac{R^*_{T|W}(D_T)}{C(q)}$. Par conséquent, la valeur minimale de la distorsion théorique D_T entre T et $\hat{T}$ peut s'exprimer par :

$$D_T = \frac{\sigma^2}{2^{2R_T C(q)}} \tag{3.28}$$

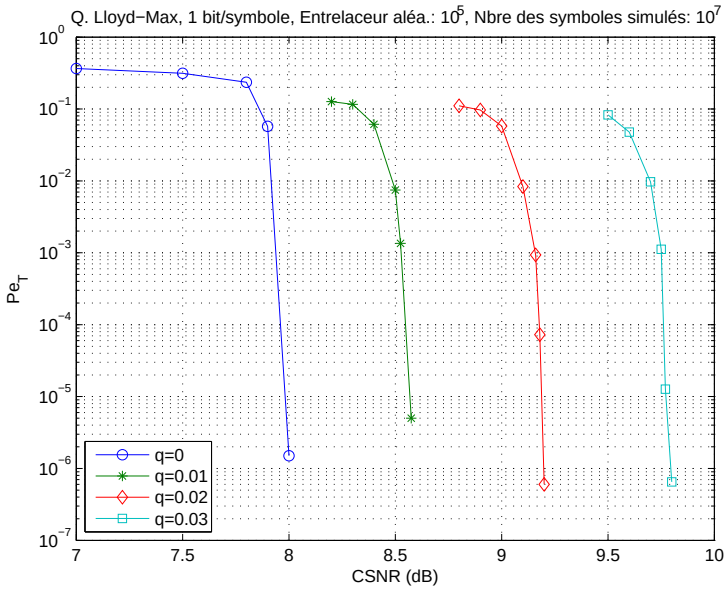

FIG. 3.22 – Taux d'erreur symbole obtenu avec un codeur conjoint source-canal de deux sources distribuées (T,W) en utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux $2/3$, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. Au total 10^7 symboles sont simulés. La source T est quantifiée avec un quantificateur de Lloyd-Max à 4 niveaux. Les bits de parité passent dans un canal CBS avec une probabilité de transition q. W est l'information de bord disponible au décodage.

Les figures 3.22 et 3.23 illustrent les résultats de simulation obtenus avec un codeur de Wyner-Ziv en présence d'un bruit CBS et pour un débit de transmission de 1 bit/symbole. Le codeur turbo poinçonné est basé sur un encodeur CRS de taux $2/3$, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$. Dans la figure 3.22 et pour différentes valeurs de la probabilité de transition q, on montre le taux d'erreur symbole (Pe_T) en fonction de la corrélation CSNR. Pour une valeur de $Pe_T = 10^{-5}$, on observe que la corrélation entre T et W doit augmenter pour contrer l'influence du bruit du CBS. Plus la valeur de q augmente, plus la perte en terme de corrélation est significative.

La figure 3.23 présente les performances en terme de distorsion moyenne mesurée. Les bornes théoriques en présence de bruits sont montrées. Pour une même valeur de distorsion, la perte de corrélation augmente en fonction de q. On remarque que plus la corrélation augmente moins il y a d'écart entre les distorsions théoriques et mesurées (de 2.46 dB pour $q = 0$ jusqu'à 1.99 dB avec $q = 0.03$). C'est le même comportement obtenu avec un codeur distribué de deux sources Gaussiennes pour $q = 0$ et différentes valeurs de débit (figure 3.7).

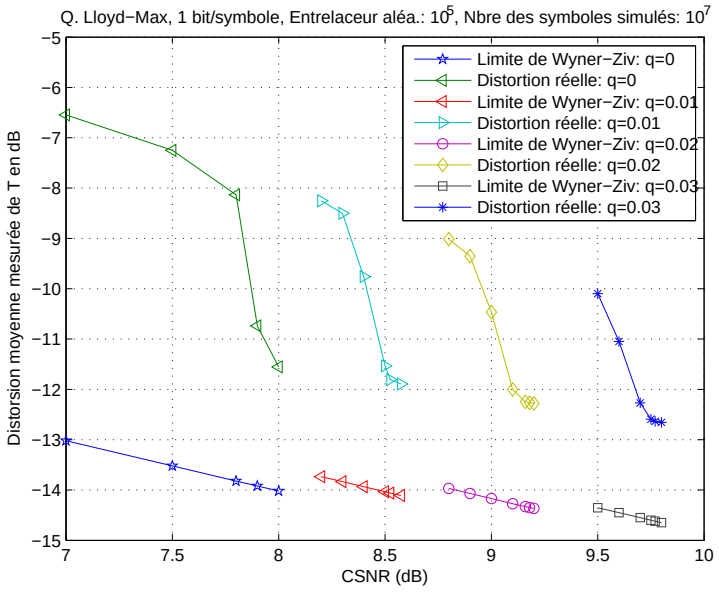

FIG. 3.23 – Distorsion moyenne mesurée d'un codeur conjoint source-canal de deux sources distribuées (T,W) en utilisant un code turbo poinçonné. Un entrelaceur aléatoire de longueur $\sim 10^5$ et un codeur CRS de taux 2/3, de longueur de contrainte $K = 5$ et un polynôme générateur $(h_0 = 27, h_1 = 23, h_2 = 35)$ sont utilisés. Au total 10^7 symboles sont simulés. La source T est quantifiée avec un quantificateur de Lloyd-Max à 4 niveaux. Les bits de parité passent dans un canal CBS avec une probabilité de transition q. W est l'information de bord disponible au décodage.

3.5 Conclusion

Nous avons présenté dans ce chapitre le codage distribué de deux sources binaires et Gaussiennes utilisant un code turbo poinçonné. La matrice de poinçonnage utilisée a permis d'améliorer l'efficacité de décodage turbo. Nous avons montré que notre solution donne de meilleures performances que celle décrite dans [AG02]. En outre, le poinçonnage des codes turbo a contribué à diminuer le temps de calcul au codage et au décodage par rapport à celui de la solution [AG02].

Par la suite, afin de s'approcher davantage des limites théoriques de Wyner-Ziv, nous avons décrit un codeur de deux sources distribuées combinant une quantification TCQ et un code turbo poinçonné. Dans le schéma de codage considéré, nous avons proposé de transmettre les bits de chemin du treillis sans compression. Nous avons observé que la distorsion mesurée du système proposé peut s'approcher de 0.916 dB de la borne théorique de Wyner-Ziv avec un débit de 1.66 bits/symbole. Pour des débits plus élevés, les performances débit-distorsion d'un codeur de Wyner-Ziv utilisant la TCQ et un code LDPC sont meilleures que celles obtenues avec notre solution.

Dans une autre partie de ce chapitre, nous avons considéré le problème du codage de Wyner-Ziv avec une information de bord partielle. Une re-démonstration des limites théoriques a été proposée. Les performances débit-distorsion en fonction du nombre de

niveaux de quantification de la source W ont été présentées. Nous avons remarqué que plus la distorsion D'_W (distorsion mesurée de W) est faible, meilleure est la performance du décodage de la source T avec moins de corrélation.

L'effet du bruit de canal binaire symétrique sur d'une part, les bornes théoriques et d'autre part, les performances débit-distorsion d'un codeur distribué de deux sources binaire et Gaussienne a été étudié. Les résultats de simulation ont montré une perte en corrélation en fonction de la probabilité de transition de canal.

Chapitre 4

Codage distribué de 3 sources binaires ou Gaussiennes utilisant le code turbo poinçonné

4.1 Introduction

Dans le chapitre précédent et comme dans la plupart des travaux actuels en matière de codage de sources distribuées, le cas de deux sources est en général considéré. Il serait intéressant d'étendre ce modèle au cas de plusieurs sources. Le but est d'offrir de nouvelles perspectives d'application du codage de sources distribuées.

Dans ce chapitre, nous proposons un schéma de codage distribué de 3 sources binaires et Gaussiennes utilisant le code turbo poinçonné. Nous étendons les théorèmes de Slepian-Wolf et de Wyner-Ziv au cas de sources multiples respectivement binaires et Gaussiennes.

Des propositions de mise en œuvre du codage distribué de trois sources binaires et Gaussiennes utilisant le code turbo poinçonné sont également proposées. Enfin, nous présentons des comparaisons de performances débit-distorsion entre les systèmes avec 3 sources et ceux avec 2 sources.

4.2 Schéma de codage distribué de trois sources binaires

Nous étondons tout d'abord le théorème de Slepian-Wolf dans le cas de 3 sources corrélées à valeurs discrètes X, Y et Z. Nous présentons ensuite un système de compression de trois sources discrètes basé sur un code turbo poinçonné.

Considérons le système de la figure 4.1 représentant une architecture de codage distribué de trois sources binaires corrélées *i.i.d* X, Y et Z avec des probabilités conditionnelles $Pr(Y \neq Z|Z) = p_1$, $Pr(X \neq Y|Y) = p_2$ et $Pr(X \neq Z|Z) = p_3$. On suppose que $Pr(X) = Pr(Y) = Pr(Z) = 1/2$. Le problème du codage distribué de trois sources binaires corrélées a d'abord été étudié dans [LLN+03], en considérant des modèles de corrélation symétriques donnés par $Pr(Y \neq Z|Z) = Pr(X \neq Y|Y) = Pr(X \neq Z|Z)$ et

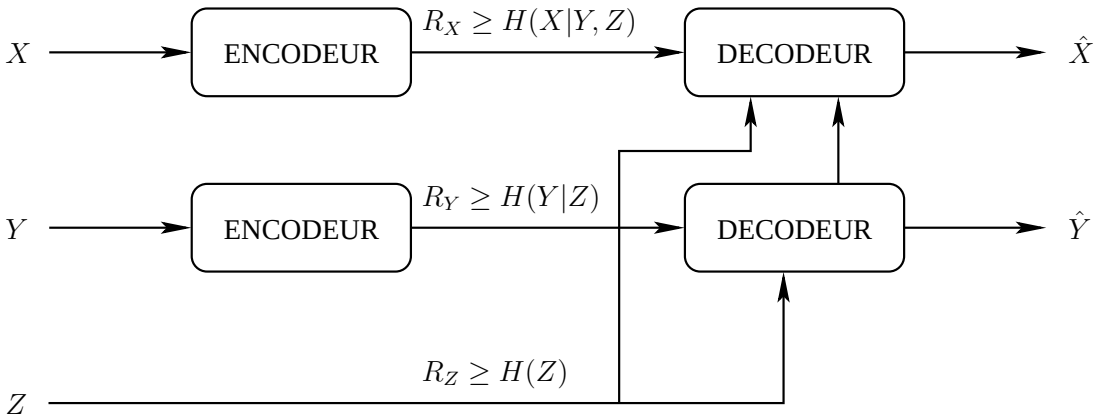

FIG. 4.1 – Schéma du principe de codage/décodage pour trois sources binaires distribuées.

en utilisant des codes LDPC. Nous étendons ici les limites théoriques de Slepian-Wolf pour des modèles de corrélation plus généraux. Afin d'avoir un codeur avec une complexité réduite, nous utilisons des codes turbo poinçonnés à la place des codes LDPC. Nous concentrons également notre étude sur des taux de compression plus élevés que ceux traités dans [LLN+03].

4.2.1 Modèle de corrélation et limites théoriques

Le théorème de Slepian-Wolf peut facilement être généralisé pour plusieurs sources. La région de débit admissible pour le codage distribué de trois sources binaires corrélées X, Y et Z est donnée par :

$$\begin{aligned} R_X &\geq H(X|Y,Z) && (4.1)\\ R_Y &\geq H(Y|Z) && (4.2)\\ R_Z &\geq H(Z) && (4.3)\\ R_X + R_Y + R_Z &\geq H(X,Y,Z) && (4.4) \end{aligned}$$

avec

$$\begin{aligned} H(X|Y,Z) &= -\sum_X \sum_Y \sum_Z Pr(X,Y,Z)\log_2(Pr(X|Y,Z)) \\ &= -\sum_X \sum_Y \sum_Z Pr(Y,Z|X)Pr(X)\log_2(Pr(X|Y,Z)) \qquad (4.5) \end{aligned}$$

L'expression de la limite théorique du débit de la source Y est donnée par $H(Y|Z) = -p_1\log_2(p_1) - (1-p_1)\log_2(1-p_1)$. Nous allons déterminer maintenant, l'expression de $H(X|Y,Z)$ suivant le modèle considéré dans la suite.

Entre les sources Z et Y, on a un canal binaire symétrique caractérisé par la probabilité de transition p_1. La corrélation entre les sources X et (Y,Z) est modélisée par

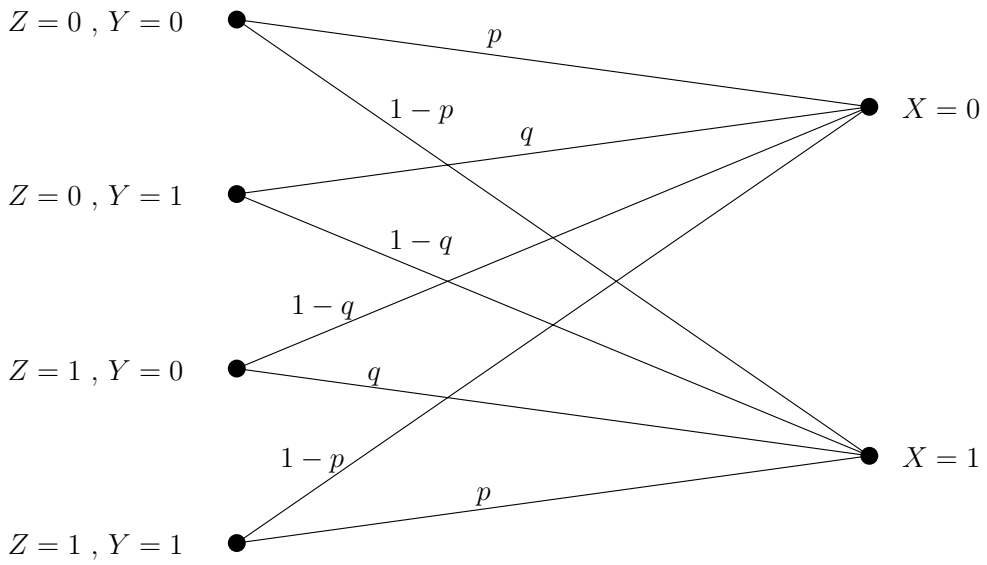

FIG. 4.2 – Canal binaire symétrique pour trois sources.

le canal illustré sur la figure 4.2 avec $p, q \in [0, 1]$ les probabilités de transition. Dans ce modèle et pour simplifier les calculs, nous avons imposé une condition de symétrie telle que : $Pr(X = i|Y = j, Z = k) = Pr(X \neq i|Y \neq j, Z \neq k)$ pour $i, j, k \in \{0, 1\}$. Ainsi, nous avons les expressions suivantes :

$$\begin{aligned} Pr(X = Z|Y = Z) &= p & (4.6)\\ Pr(X \neq Z|Y = Z) &= 1 - p & (4.7)\\ Pr(X = Z|Y \neq Z) &= q & (4.8)\\ Pr(X = Y|Y \neq Z) &= 1 - q & (4.9) \end{aligned}$$

Les valeurs de p et q peuvent être exprimées en fonction de p_1, p_2 et p_3. En effet,

$$\begin{aligned} Pr(X = 0|Z = 0) &= \frac{\sum_Y Pr(X = 0, Y, Z = 0)}{Pr(Z = 0)} \\ &= \frac{Pr(X = 0|Y = 0, Z = 0).Pr(Y = 0|Z = 0).Pr(Z = 0)}{Pr(Z = 0)} \\ &\quad + \frac{Pr(X = 0|Y = 1, Z = 0).Pr(Y = 1|Z = 0).Pr(Z = 1)}{Pr(Z = 0)} \\ &= p(1 - p_1) + qp_1 = 1 - p_3 \qquad (4.10) \end{aligned}$$

De même on a :

$$Pr(X = 0|Y = 1) = qp_1 + (1 - p)(1 - p_1) = p_2 \qquad (4.11)$$

D'où :

$$p = \frac{2-(p_1+p_2+p_3)}{2(1-p_1)} \quad (4.12)$$

$$q = \frac{p_1+p_2-p_3}{2p_1} \quad (4.13)$$

Les différentes cas possibles des probabilités conditionnelles $Pr(Y,Z|X)$ peuvent être exprimées par :

$$\begin{aligned} Pr(Y=Z=X|X) &= \frac{Pr(X,Y,Z)}{Pr(X)} = \frac{Pr(X|Y,Z).Pr(Z|Y).Pr(Y)}{Pr(X)} \\ &= Pr(X|Y,Z).Pr(Z|Y) = p(1-p_1) \quad (4.14) \\ Pr(Y\neq Z=X|X) &= qp_1 \quad (4.15) \\ Pr(Y=X\neq Z|X) &= (1-q)p_1 \quad (4.16) \\ Pr(Y=Z\neq X|X) &= (1-p)(1-p_1) \quad (4.17) \end{aligned}$$

Etant données les expressions de probabilité des équations (4.14), (4.15), (4.16) et (4.17), l'entropie conditionnelle $H(X|Y,Z)$ de (4.5) donnant la limite théorique de débit de la troisième source X peut être réécrite sous la forme suivante :

$$\begin{aligned} H(X|Y,Z) = \quad & p_1\log_2(p_1) + (1-p_1)\log_2(1-p_1) \\ - \quad & (1-\frac{p_1+p_2+p_3}{2})\log_2(1-\frac{p_1+p_2+p_3}{2}) \\ - \quad & [\frac{p_1+p_2-p_3}{2}\log_2(\frac{p_1+p_2-p_3}{2}) \\ + \quad & \frac{p_1-p_2+p_3}{2}\log_2(\frac{p_1-p_2+p_3}{2}) \\ + \quad & \frac{-p_1+p_2+p_3}{2}\log_2(\frac{-p_1+p_2+p_3}{2})] \quad (4.18) \end{aligned}$$

4.2.2 Codeur distribué de trois sources binaires basé sur un code turbo poinçonné

Dans la figure 4.3, nous proposons une architecture de codage étendant le principe turbo au cas de trois sources distribuées. Les trois sources sont codées séparément et décodées conjointement. Les deux sources X et Y sont codées par deux encodeurs turbo différents. Chaque encodeur est constitué de deux codeurs élémentaires CRS montés en parallèle et séparés par un entrelaceur comme l'indique la partie codage de la figure 4.3. Les taux de codage des codeurs CRS pour les sources X et Y sont respectivement $\frac{n}{n+1}$ et $\frac{m}{m+1}$. Pour chaque symbole codé par un encodeur CRS, seulement des bits de parité poinçonnés (X_P^i et Y_P^i, pour tout $i \in \{1,2\}$) sont transmis au décodeur. Par conséquent, les taux R_X et R_Y de compression dépendent du rendement de chaque encodeur CRS et des matrices de poinçonnage respectives P_X et P_Y.

Vu que l'information de bord Z (transmise à un taux égal à son entropie théorique $H(Z)$) est disponible au décodage, les deux sources X et Y sont compressées à des taux

plus faibles que leurs entropies respectives. Nous cherchons à compresser la source Y à un taux R_Y approchant la limite de Slepian-Wolf donnée par $H(Y|Z)$. Quant à la source X, puisque au décodage on dispose de deux informations de bord qui sont $\hat{Y}$ et Z, elle sera compressée à $H(X|Y,Z)$.

Au décodage, deux décodeurs turbo itératifs sont utilisés pour estimer $\hat{Y}$ et $\hat{X}$. Les décodeurs turbo sont composés de deux décodeurs SISO concaténés en série à travers un entrelaceur ou un désentrelaceur et utilisant l'algorithme MAP. Après un certain nombre d'itérations, le décodeur turbo de la source Y estime $\hat{Y}$ en utilisant la probabilité conditionnelle $P(Y|Z)$, l'information de bord Z et les bits de parité correspondants. L'estimée $\hat{Y}$ est transmise au décodeur turbo de la source X. Ce dernier dispose en plus de l'information de bord Z et les bits de parité correspondants pour décoder $\hat{X}$ (l'estimée de la source X) en utilisant la probabilité conditionnelle $P(X|Y,Z)$. Tant que les performances, en terme de taux d'erreur binaire, de la source Y sont bonnes (entre 10^{-4} et 10^{-5}), il n'y a pas une grande erreur sur l'estimation de la source X.

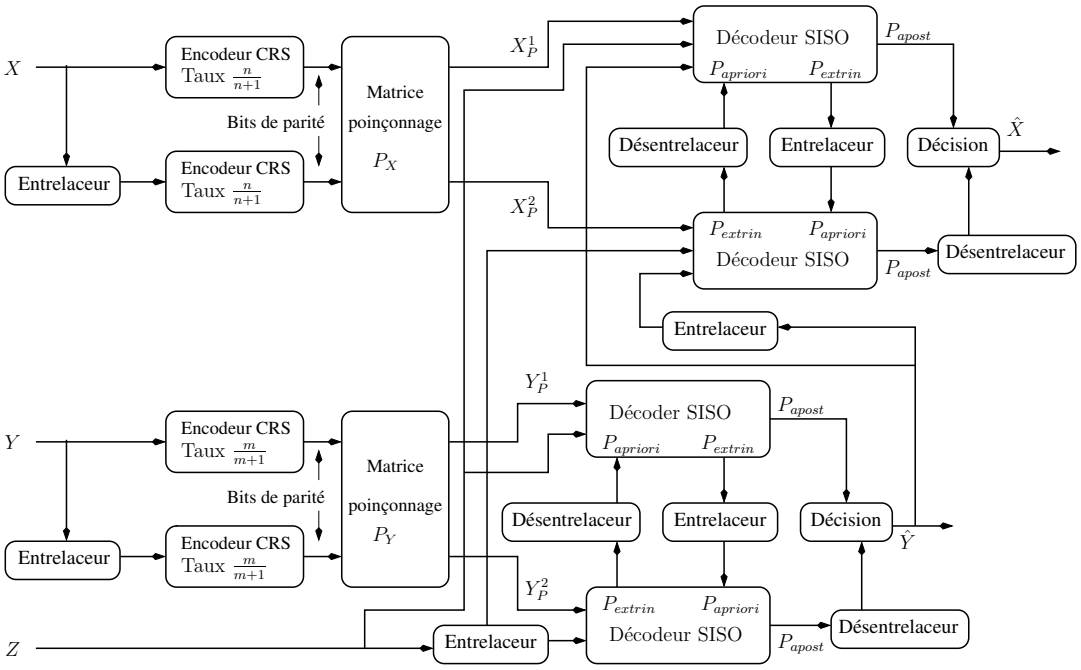

FIG. 4.3 – Schéma du codeur/décodeur pour trois sources binaires corrélées X, Y et Z.

4.3 Schéma de codage distribué de trois sources Gaussiennes

Considérons maintenant trois sources Gaussiennes corrélées et sans mémoires X, Y et Z (*i.i.d*) de moyennes nulles et de variances respectives σ_X^2, σ_Y^2 et σ_Z^2.

4.3.1 Modèle de corrélation

La corrélation entre les trois sources est modélisée par : $Y = Z + N_1$ et $X = Y + Z + N$, avec $N_1 \sim \mathcal{N}(0, \sigma_1^2)$ et $N \sim \mathcal{N}(0, \sigma^2)$ des bruits Gaussiens $(i.i.d)$ supposés indépendants entre eux et de Z. Par conséquent, la source Y est indépendante de N. Afin de pouvoir comparer les performances débit-distorsion de codage de X dans un système de codage distribué de trois sources (désigné par CSD-3) à celles obtenues avec un système de codage distribué à 2 sources (désigné par CSD-2), le signal $(Y + Z)$ doit avoir une énergie égale à l'unité ($\sigma^2_{Y+Z} = 1$). Par la suite, nous imposons que $\sigma_Z^2 = \frac{1-\sigma_1^2}{4}$.

Nous allons nous intéresser à deux schémas de codage distribué de trois sources Gaussiennes. Avec le premier schéma, les valeurs prises par la source Z sont seulement connues par les décodeurs, tandis que les sources X et Y sont quantifiées avant d'être codées avec un codeur de Slepian-Wolf. Quant au deuxième schéma qui correspond à une solution réelle, les trois sources X, Y et Z sont quantifiées. Mais, seules les sources X et Y sont codées.

4.3.2 Premier schéma : Z est non quantifiée

4.3.2.1 Structure du codeur/décodeur

La figure 4.4 montre la structure du codeur/décodeur pour le codage distribué de trois sources Gaussiennes. Avant codage, les sources Y et X sont quantifiées en utilisant un quantificateur de Lloyd-Max scalaire à 2^M niveaux. Les séquences de symboles quantifiées Y_Q et X_Q sont introduites dans différents encodeurs turbo et seulement les bits de parité poinçonnés sont transmis aux décodeurs. La source Z est considérée comme l'information de bord et transmise sans compression.

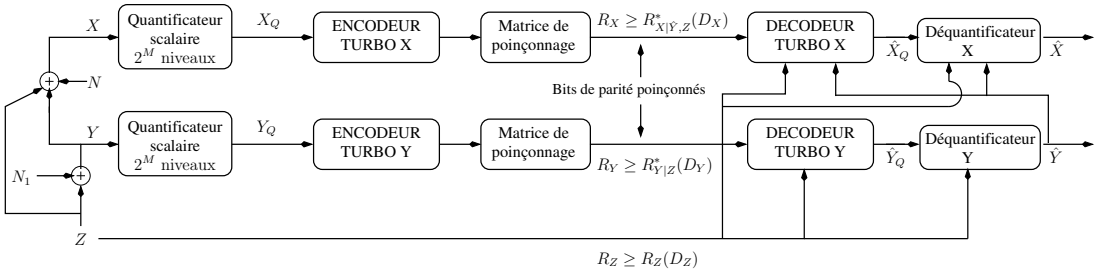

FIG. 4.4 – Schéma du codeur-décodeur pour trois sources Gaussiennes corrélées X, Y et Z.

La structure du décodeur est similaire à celle considérée dans un système de codage distribué pour trois sources binaires corrélées à l'exception de l'utilisation de la distribution de probabilité à valeurs continues et de l'existence des déquantificateurs à la sortie de chaque décodeur turbo. La métrique de branche du décodeur turbo X dépend de la probabilité $Pr(\hat{Y} = \hat{y}, Z = z | X_Q = x_q)$ déterminée dans l'annexe B. Les déquantificateurs après le décodage turbo, définis comme suit, permettent d'estimer $\hat{Y}$

et $\hat{X}$. L'estimateur optimal de Y étant donnés $\hat{Y}_Q \in [c,d]$ et Z est :

$$\begin{aligned}\hat{Y} &= E(Y = y \in [c,d] | \hat{Y}_Q, Z = z) \\ &= \int_c^d y \; Pr(Y = y \in [c,d] | \hat{Y}_Q, Z = z) dy \\ &= \int_c^d y \; \frac{p(\hat{Y}_Q|y)p(y|z)}{p(\hat{Y}_Q|z)} dy \end{aligned} \tag{4.19}$$

De la même manière, afin de minimiser la distorsion $E[(X - \hat{X})^2]$, l'estimateur optimal de X étant donnés $\hat{X}_Q \in [a,b]$, $\hat{Y}$ et Z peut être exprimé par :

$$\begin{aligned}\hat{X} &= E(X = x \in [a,b] | \hat{X}_Q, \hat{Y}, Z = z) \\ &= \int_a^b x \; Pr(X = x \in [a,b] | \hat{X}_Q, \hat{Y}, Z = z) dx \\ &= \int_a^b x \; \frac{p(\hat{X}_Q|x)p(x|\hat{Y},z)}{p(\hat{X}_Q|\hat{Y},z)} dx \end{aligned} \tag{4.20}$$

4.3.2.2 Bornes théoriques

Maintenant, nous présentons les régions débit- distorsion (R_X, R_Y, D_X, D_Y) qui peuvent être obtenues par un système de codage distribué de trois sources Gaussiennes. D_X et D_Y sont respectivement les distorsions théoriques moyennes entre X et $\hat{X}$, et Y et $\hat{Y}$. La région débit-distorsion donnée par (R_Y,D_Y) est similaire à celle d'un système de codage Wyner-Ziv pour deux sources Gaussiennes corrélées. Par conséquent, les régions (R_X,R_Y,D_X,D_Y) sont définies comme suit :

$$R_X \geq R^*_{X|\hat{Y},Z_Q}(D_X) = I(X; X_Q|\hat{Y}, Z_Q) \tag{4.21}$$

$$R_Y \geq R^*_{Y|Z_Q}(D_Y) = I(Y; Y_Q|Z_Q) \tag{4.22}$$

$$D_X \geq E[d(X,\hat{X})] \tag{4.23}$$

$$D_Y \geq E[d(Y,\hat{Y})] \tag{4.24}$$

Dans [WZ76], il a été montré que la fonction débit-distorsion de la source Y étant donnée l'information de bord Z peut être exprimée par :

$$R^*_{Y|Z}(D_Y) = \frac{1}{2}\log_2(\frac{(\sigma_1^2)}{D_Y}) \tag{4.25}$$

Maintenant, supposons que $X_Q = X + Q_X$, $Y_Q = Y + Q_Y$ et $\hat{Y} = Y + N_{D_Y}$ où $Q_X \sim \mathcal{N}(0, \sigma^2_{Q_X})$, $Q_Y \sim \mathcal{N}(0, \sigma^2_{Q_Y})$ et $N_{D_Y} \sim \mathcal{N}(0, D_Y)$ sont des bruits Gaussiens (*i.i.d*) supposés indépendants entre eux et de X, Y, Z et N. La fonction débit-distorsion théorique de la source X, sous la contrainte que la distorsion moyenne entre X et $\hat{X}$

soit inférieure à D_X, peut être écrite de la façon suivante :

$$\begin{aligned} R^*_{X|\hat{Y},Z}(D_X) &= I(X;X_Q|\hat{Y},Z) = I(X;X_Q) - I(\hat{Y},Z;X_Q) \\ &= H(X_Q|\hat{Y},Z) - H(X_Q) + H(X_Q) - H(X_Q|X) \\ &= H(Y+Z+N+Q_X|\hat{Y},Z) - H(X+Q_X|X) \\ &= H(\hat{Y}-N_{D_Y}+Z+N+Q_X|\hat{Y},Z) - H(Q_X) \\ &\simeq \frac{1}{2}\log_2(2\pi e(\sigma^2+D_Y+\sigma^2_{Q_X})) - \frac{1}{2}\log_2(2\pi e(\sigma^2_{Q_X})) \\ &\simeq \frac{1}{2}\log_2(\frac{\sigma^2+D_Y+\sigma^2_{Q_X}}{\sigma^2_{Q_X}}) \end{aligned} \tag{4.26}$$

Dans [PR03a], l'estimée de la source Y est $\hat{Y} = a_0 Z + b_0 Y_Q$ avec $a_0 = \frac{1}{2^{2R^*_{Y|Z}(D_Y)}}$, $b_0 = 1 - a_0$ et $D_Y = \frac{\sigma_1^2}{2^{2R^*_{Y|Z}(D_Y)}}$. Soit $\hat{X} = a\hat{Y} + bZ + cX_Q$ l'estimée optimale de X exprimée par des combinaisons convexes de $\hat{Y}$, Z et X_Q. Etant donné que $E[(X-\hat{X})\hat{Y}] = 0$, $E[(X-\hat{X})Z] = 0$ et $E[(X-\hat{X})X_Q] = 0$, nous avons :

$$2 - aa_0 - ab_0 - b - 2c = 0 \tag{4.27}$$

$$\sigma_1^2(1 - b_0 a - c) - b_0 a\sigma^2_{Q_Y} = 0 \tag{4.28}$$

$$\sigma_1^2(1 - b_0 a - c) + (1-c)\sigma^2 - c\sigma^2_{Q_X} = 0 \tag{4.29}$$

On en déduit,

$$a = b = 1 - c = \frac{\sigma^2_{Q_X}}{\sigma^2 + \sigma^2_{Q_X} + D_Y} \tag{4.30}$$

$$c = \frac{\sigma^2 + D_Y}{\sigma^2 + \sigma^2_{Q_X} + D_Y} \tag{4.31}$$

La distorsion D_X peut être calculée comme suit :

$$\begin{aligned} D_X &= E[(X-\hat{X})^2] \\ &= c\sigma^2_{Q_X} \end{aligned} \tag{4.32}$$

Par conséquent, de (4.26) et (4.32), nous avons :

$$\sigma^2_{Q_X} = \frac{(\sigma^2 + D_Y)}{2^{2R^*_{X|\hat{Y},Z}(D_X)} - 1} \tag{4.33}$$

c peut être exprimé par :

$$c = \frac{2^{2R^*_{X|\hat{Y},Z}(D_X)} - 1}{2^{2R^*_{X|\hat{Y},Z}(D_X)}} \tag{4.34}$$

Ainsi, la distorsion D_X devient :

$$D_X = \frac{(\sigma^2 + D_Y)}{2^{2R^*_{X|\hat{Y},Z}(D_X)}} \tag{4.35}$$

Et finalement, la fonction débit-distorsion théorique de la source X peut s'écrire par :

$$R^*_{X|\hat{Y},Z}(D_X) = \frac{1}{2}\log_2(\frac{\sigma^2 + D_Y}{D_X}) \tag{4.36}$$

Dans (4.36), on peut voir que la distorsion D_Y a un impact sur les performances de décodage de X dans le sens débit-distorsion. Les signaux utilisés pour décoder X sont en fait $\hat{Y} = Y + N_{D_Y}$ et Z, et non Y et Z, avec $N_{D_Y} \sim \mathcal{N}(0, D_Y)$ une variable aléatoire Gaussienne ($i.i.d$) supposée indépendante de tous les autres bruits, de Z et de Y. Cependant, les performances du système en terme de débit-distorsion doivent être évaluées considérant l'entière région (R_X, R_Y, D_X, D_Y). Il s'avère que le système CSD-3 permet d'avoir une valeur de distorsion de X plus faible pour un même débit et un même taux de corrélation.

4.3.3 Deuxième schéma : Z est quantifiée

4.3.3.1 Structure du codeur/décodeur

Les sources X et Y sont quantifiées en utilisant un quantificateur Lloyd-Max à 2^M niveaux. Les séquences de symboles quantifiées Y_Q et X_Q sont introduites dans différents encodeurs turbo et seulement les bits de parité poinçonnés sont envoyés aux décodeurs. La source Z est quantifiée en utilisant un Lloyd-Max à 2^L niveaux. Le débit de transmission de Z est L bits/s (bits/symbole). Z_Q, la version quantifiée de Z, est considérée comme l'information de bord partielle.

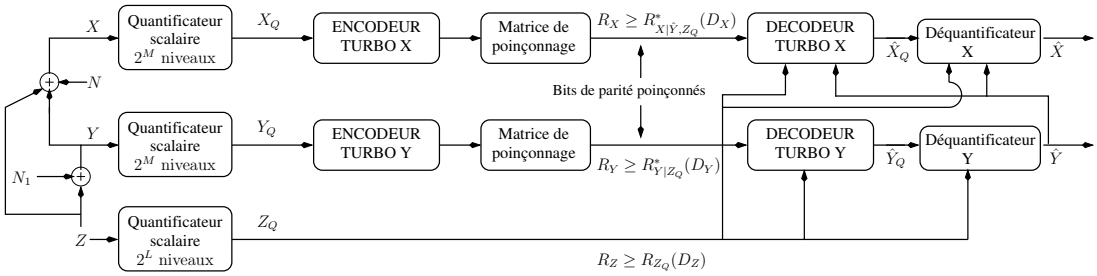

FIG. 4.5 – Schéma d'un codeur distribué de trois sources Gaussiennes corrélées et quantifiées X, Y et Z.

La partie décodage est la même que celle présentée dans la section 4.3.2.1, sauf qu'ici les décodeurs turbo et les déquantificateurs de X et de Y utilisent l'information de bord partielle Z_Q.

4.3.3.2 Bornes théoriques

Les régions de débits et de distorsions moyennes d'un système de codage distribué de trois sources Gaussiennes quantifiées sont données par (R_X, R_Y, R_Z, D_X, D_Y, D_Z) . D_X, D_Y et D_Z sont respectivement les distorsions théoriques moyennes entre X et $\hat{X}$, Y et $\hat{Y}$, et Z et Z_Q. La région débit-distorsion donnée par (R_Y,R_Z,D_Y,D_Z) est similaire à celle d'un système de codage distribué de deux sources Gaussiennes quantifiées déjà présenté au chapitre 3. Par conséquent, les régions (R_X,R_Y,R_Z,D_X,D_Y,D_Z) sont définies comme suit :

$$\begin{aligned}
R_X &\geq R^*_{X|\hat{Y},Z_Q}(D_X) = I(X;X_Q|\hat{Y},Z_Q) && (4.37)\\
R_Y &\geq R^*_{Y|Z_Q}(D_Y) = I(Y;Y_Q|Z_Q) && (4.38)\\
R_Z &\geq R_{Z_Q}(D_Z) = I(Z;Z_Q) && (4.39)\\
D_X &\geq E[d(X,\hat{X})] && (4.40)\\
D_Y &\geq E[d(Y,\hat{Y})] && (4.41)\\
D_Z &\geq E[d(Z,Z_Q)] && (4.42)
\end{aligned}$$

La fonction débit-distorsion théorique de Z s'écrit sous la forme :

$$R_{Z_Q}(D_Z) = \frac{1}{2}\log_2(\frac{\sigma_Z^2}{D_Z}) \tag{4.43}$$

Dans (3.25), nous avons vu que la fonction débit-distorsion de la source Y étant donnée l'information de bord Z_Q présentant une distorsion D_Z par rapport à Z, peut être exprimée par :

$$R^*_{Y|Z_Q}(D_Y) = \frac{1}{2}\log_2(\frac{(\sigma_1^2 + D_Z)}{D_Y}) \tag{4.44}$$

Il reste à déterminer la fonction débit-distorsion de la source X. Soient $X = X_Q + Q_X$, $Y = Y_Q + Q_Y = \hat{Y} + N_{D_Y}$ et $Z = \hat{Z} + N_{D_Z}$ avec $Q_X \sim \mathcal{N}(0,\sigma^2_{Q_X})$, $Q_Y \sim \mathcal{N}(0,\sigma^2_{Q_Y})$, $N_{D_Y} \sim \mathcal{N}(0,D_Y)$ et $N_{D_Z} \sim \mathcal{N}(0,D_Z)$ indépendantes de X_Q, Y_Q et $\hat{Z}$. Dans le système présenté à la figure 4.5, la version quantifiée de Z, Z_Q, est la même que $\hat{Z}$.

Pour le cas de trois sources, la fonction débit-distorsion de la source X s'écrit :

$$\begin{aligned}
R^*_{X|\hat{Y},\hat{Z}}(D_X) &= I(X;X_Q|\hat{Y},\hat{Z})\\
&= I(X;X_Q) - I(\hat{Y},\hat{Z};X_Q)\\
&= H(X_Q|\hat{Y},\hat{Z}) - H(X_Q|X)\\
&= H(\hat{Y} + N_{D_Y} + \hat{Z} + N_{D_Z} + N - Q_X|\hat{Y},\hat{Z})\\
&\quad -H(-Q_X|X)\\
&\simeq \frac{1}{2}\log_2(\frac{\sigma^2 + D_Y + D_Z + \sigma^2_{Q_X}}{\sigma^2_{Q_X}}) \qquad (4.45)
\end{aligned}$$

On a vu dans au chapitre 3, que $\hat{Y} = a_0\hat{Z} + b_0 Y_Q$ avec $a_0 = \frac{1}{2^{2R^*_{Y|Z_Q}(D_Y)}}$, $b_0 = 1 - a_0$ et $D_Y = \frac{\sigma_1^2 + D_Z}{2^{2R^*_{Y|Z_Q}(D_Y)}}$.

Soit $\hat{X} = a\hat{Y} + b\hat{Z} + cX_Q$ l'estimation (MMSE) de X étant données $\hat{Y}$, $\hat{Z}$ et X_Q. Puisque $E[(X - \hat{X})\hat{Y}] = 0$, $E[(X - \hat{X})\hat{Z}] = 0$ et $E[(X - \hat{X})X_Q] = 0$, alors on aura :

$$a = \frac{2D_Z + \sigma_1^2}{D_Z + \sigma_1^2}(1 - c) \tag{4.46}$$

$$b = \frac{\sigma_1^2}{D_Z + \sigma_1^2}(1 - c) \tag{4.47}$$

$$c = \frac{1}{1 + \frac{(D_Z + \sigma_1^2)\sigma_{Q_X}^2}{(D_Z + \sigma_1^2)(4D_Z + \sigma_1^2 + \sigma^2) - b_0(2D_Z + \sigma_1^2)^2}} \tag{4.48}$$

La distorsion D_X peut être déterminée de la façon suivante :

$$D_X = E[(X - \hat{X})^2] = c\sigma_{Q_X}^2 \tag{4.49}$$

Par la suite, de (4.45) et (4.49), nous avons :

$$\sigma_{Q_X}^2 = \frac{(\sigma^2 + D_Y + D_Z)}{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - 1} \tag{4.50}$$

et la composante c peut être exprimée par :

$$c = \frac{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - 1}{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - \frac{D_Z(2D_Y - b_0 D_Z)}{D_Z(2D_Y - b_0 D_Z) + (1/a_0)D_Y(\sigma^2 + D_Y + D_Z)}} \tag{4.51}$$

Supposons que :

$$\epsilon = \frac{D_Z(2D_Y - b_0 D_Z)}{D_Z(2D_Y - b_0 D_Z) + (1/a_0)D_Y(\sigma^2 + D_Y + D_Z)} \tag{4.52}$$

Donc, (4.51) devient :

$$c = \frac{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - 1}{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - \epsilon} \tag{4.53}$$

Il peut être vérifié que le dénominateur de c, $(2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - \epsilon)$ est toujours positif et différent de zéro.

Par conséquent, la distorsion théorique D_X devient :

$$D_X = \frac{(\sigma^2 + D_Y + D_Z)}{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - \epsilon} \tag{4.54}$$

Et finalement, la fonction débit-distorsion de la source X peut s'écrire :

$$R^*_{X|\hat{Y},\hat{Z}}(D_X) = \frac{1}{2}\log_2(\frac{\sigma^2 + D_Y + D_Z}{D_X} + \epsilon) \tag{4.55}$$

Pour des valeurs de D_Z très faibles, on a :

$$\begin{aligned} \epsilon &\ll 2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} \\ c &\simeq \frac{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)} - 1}{2^{2R^*_{X|\hat{Y},\hat{Z}}(D_X)}} \\ R^*_{X|\hat{Y},\hat{Z}}(D_X) &= \frac{1}{2}\log_2(\frac{\sigma^2 + D_Y}{D_X}) \end{aligned}$$

4.4 Résultats de simulation

Dans toutes les simulations réalisées dans cette partie, nous avons utilisé un codeur CRS de taux 2/3 avec une longueur de contrainte $K = 5$ et des polynômes générateurs $(h_0 = 33, h_1 = 23, h_2 = 35)$ et $(h_0 = 27, h_1 = 23, h_2 = 35)$ respectivement pour le cas binaire et le cas Gaussien. La longueur de l'entrelaceur aléatoire est de $\sim 10^5$. Les résultats de simulation sont obtenus avec 100 trames, c'est-à-dire au total 10^7 symboles simulés. L'algorithme de décodage utilisé est le MAP (Maximum A Posteriori).

On désigne par X, Y et Z les trois sources corrélées (binaires ou Gaussiennes) pour le système de codage CSD-3. Soient aussi T et W les deux sources corrélées pour le système CSD-2, avec T le signal principal. On compare ici, les performances obtenues avec un système CSD-3 et celles obtenues avec CSD-2.

4.4.1 Cas des sources binaires

Pour le système de codage CSD-2 de deux sources binaires, la corrélation entre T et W est modélisée par la probabilité de transition q_1. La source T est compressée et W est transmise à son taux entropique (soit 1 : 1). Dans les figures 4.6, 4.7 et 4.8 et pour le système de codage CSD-3 de trois sources binaires corrélées, le taux de compression pour Y est mis à 2 : 1. La source Z est transmise à un taux égal à son entropie, soit 1 : 1. La corrélation entre Y et Z est fixée à $p_1 = 0.0861$. Cette valeur correspond à la limite de performance, en terme de taux d'erreur binaire, du système CSD-2 avec un taux de compression 2 : 1 ($H(Y|Z) = 0.4233$ bits). Dans les figures 4.9 et 4.10, les taux de compression des deux sources X et Y sont les mêmes et Z est transmise à son taux entropique.

Pour un même taux de compression 2 : 1 (la limite théorique est 0.5) des sources X et Y, la figure 4.6 montre que CSD-3 permet d'obtenir les mêmes performances que CSD-2 (le débit $H(X|Y,Z)$ est égal à 0.4242 bits, soit meilleur de 0.0009 bits que l'entropie conditionnelle $H(T|W)$), avec une corrélation plus faible ($p_1 = 0.0861$, $p_2 = 0.09$ et $p_3 = 0.136$ pour CSD-3 et $q_1 = 0.0861$ pour CSD-2). Cependant, les débits globaux de CSD-3 et de CSD-2 sont respectivement $R_3 = R_X + R_Y + R_Z = 0.5 + 0.5 + 1 = 2$ et

$R_2 = R_T + R_W = 0.5 + 1 = 1.5$ bits. Ce qui correspond à un débit moyen par source de $\frac{2}{3}$ bit/source pour CSD-3 et de 0.75 bit/source pour CSD-2. On peut déduire que le débit moyen par source est meilleur avec le système CSD-3 que celui obtenu avec CSD-2.

Pour des taux de compression plus élevés, la figure 4.7 compare les performances entre le cas 2 sources (CSD-2 avec T compressée à $3:1$ et W à $1:1$) et le cas 3 sources où X est compressée à $3:1$, Y à $2:1$ et Z à $1:1$. On peut voir que le système CSD-3 permet de se rapprocher davantage de la borne théorique 0.33 pour une corrélation plus faible ($p_1 = 0.0861$ et $p_2 = p_3 = 0.073$ dans CSD-3 au lieu de $q_1 = 0.0466$ dans CSD-2). CSD-3 a un gain de 0.0045 bits par rapport à CSD-2 (le débit $H(X|Y,Z)$ est égal à 0.2762 bits alors que l'entropie conditionnelle $H(T|W) = 0.2717$ bits). Le débit moyen par source est de 0.611 et de 0.666 bit/source respectivement avec CSD-3 et CSD-2.

La différence de performance entre CSD-3 et CSD-2 augmente avec le taux de compression comme l'indique la figure 4.8 (taux de compression $6:1$ pour les sources X et T, c'est-à-dire une limite théorique de 0.1667). La figure 4.8 montre un gain de 0.0164 bits, en terme de distance avec la limite théorique, de CSD-3 par rapport à CSD-2 ($H(X|Y,Z) = 0.1371$ bits dans CSD-3 et $H(T|W) = 0.1207$ bits dans CSD-2) avec une faible corrélation entre les différentes sources ($p_1 = 0.0861$ et $p_2 = p_3 = 0.0489$ dans CSD-3 contre $q_1 = 0.0164$ dans CSD-2). Le débit moyen par source obtenu avec CSD-3 reste inférieur à celui de CSD-2 (0.55 bit/source pour CSD-3 et 0.583 bit/source pour CSD-2).

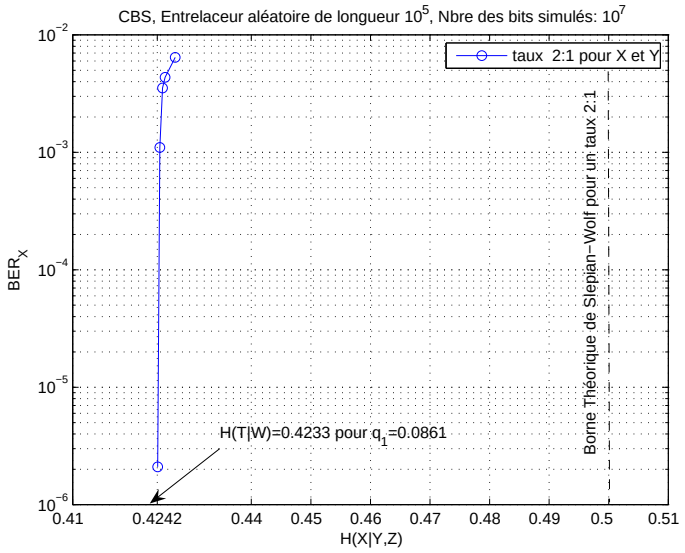

FIG. 4.6 – Taux d'erreur binaire (BER) des systèmes de codage de sources distribuées CSD-2 et CSD-3 : le taux de compression est de $2:1$ pour X et Y dans CSD-3 et dans CSD-2 le taux est de $2:1$ pour T. Un code turbo poinçonné dont le codeur élémentaire est de taux 2/3, de longueur de contrainte $K = 5$ et taille d'entrelaceur : 10^5 a été utilisé. Au total 10^7 bits sont simulés.

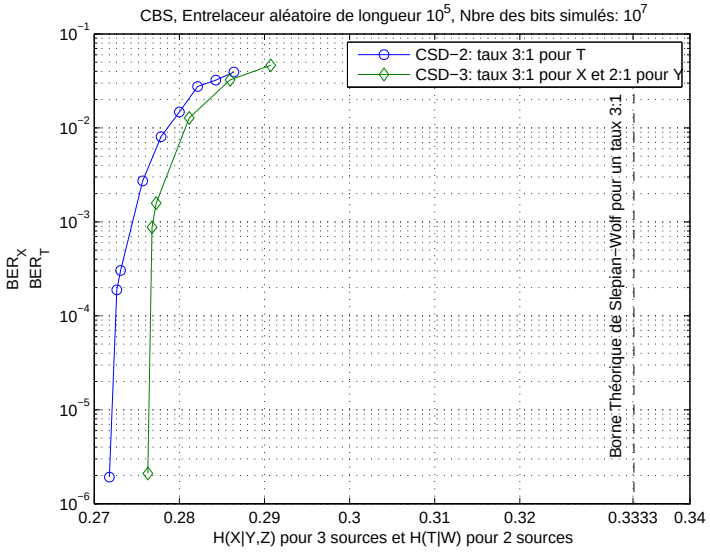

FIG. 4.7 – Taux d'erreur binaire (BER) des systèmes de codage de sources distribuées CSD-2 et CSD-3 : le taux de compression est de $3:1$ pour X et $2:1$ pour Y dans CSD-3 et dans CSD-2 le taux est de $3:1$ pour T. Un code turbo poinçonné dont le codeur élémentaire est de taux $2/3$, de longueur de contrainte $K=5$ et taille d'entrelaceur : 10^5 a été utilisé. Au total 10^7 bits sont simulés.

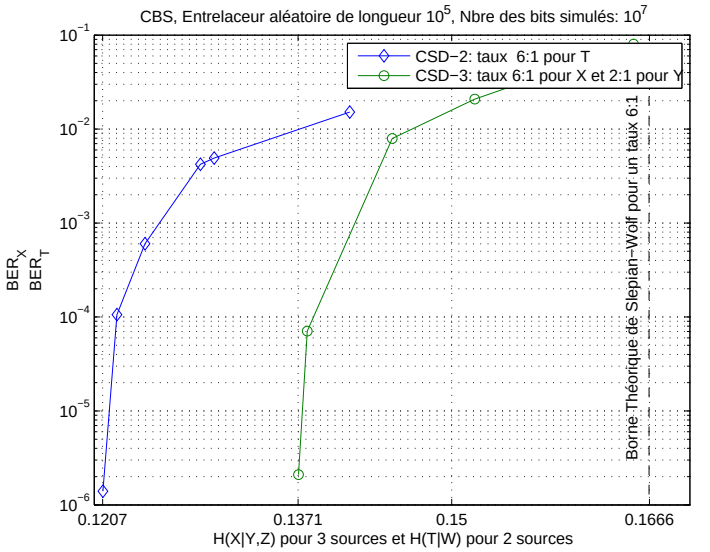

FIG. 4.8 – Taux d'erreur binaire (BER) des systèmes de codage de sources distribuées CSD-2 et CSD-3 : le taux de compression est de $6:1$ pour X et $2:1$ pour Y dans CSD-3 et dans CSD-2 le taux est de $6:1$ pour T. Un code turbo poinçonné dont le codeur élémentaire est de taux $2/3$, de longueur de contrainte $K=5$ et taille d'entrelaceur : 10^5 a été utilisé. Au total 10^7 bits sont simulés.

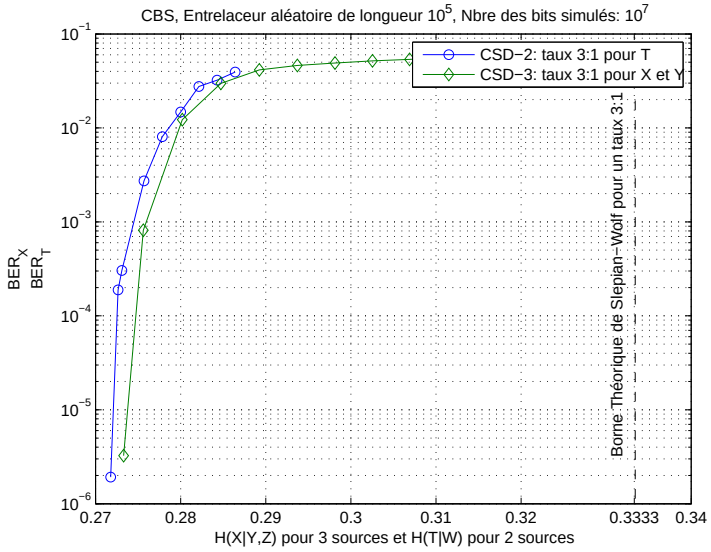

FIG. 4.9 – Taux d'erreur binaire (BER) des systèmes de codage de sources distribuées CSD-2 et CSD-3 : le taux de compression est de $3:1$ pour X et Y dans CSD-3 et dans CSD-2 le taux est de $3:1$ pour T. Un code turbo poinçonné dont le codeur élémentaire est de taux $2/3$, de longueur de contrainte $K=5$ et taille d'entrelaceur : 10^5 a été utilisé. Au total 10^7 bits sont simulés.

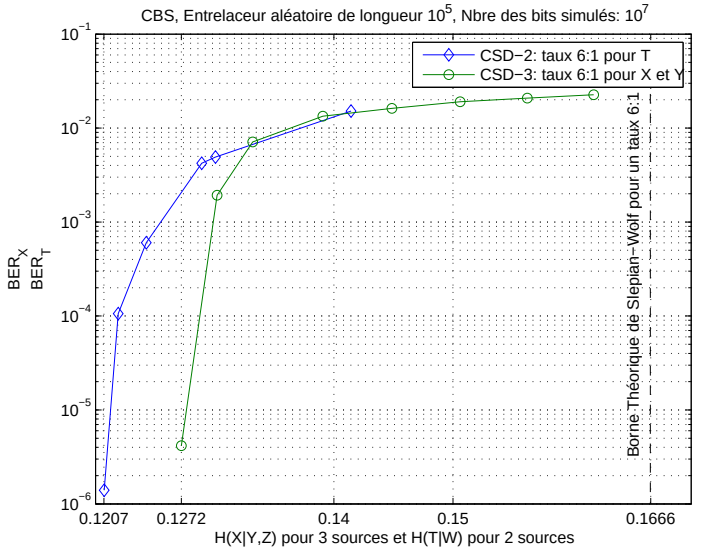

FIG. 4.10 – Taux d'erreur binaire (BER) des systèmes de codage de sources distribuées CSD-2 et CSD-3 : le taux de compression est de $6:1$ pour X et Y dans CSD-3 et dans CSD-2 le taux est de $6:1$ pour T. Un code turbo poinçonné dont le codeur élémentaire est de taux $2/3$, de longueur de contrainte $K=5$ et taille d'entrelaceur : 10^5 a été utilisé. Au total 10^7 bits sont simulés.

Pour des taux de compression plus élevés de la source Y dans DSC-3, les figures 4.9 et 4.10 montrent que les performances de décodage du système CSD-3 sont meilleures que celles obtenues avec CSD-2, et ceci avec moins de corrélation entre les sources et pour des débits moyens par source plus faibles. Dans la figure 4.9, le taux de compression des sources X, Y et T est de 3 : 1. Les débits obtenus sont $H(X|Y,Z) = 0.2733$ et $H(Y|Z) = 0.2718$ bits dans CSD-3 et $H(T|W) = 0.2718$ bits dans DSC-2 (avec $p_1 = 0.0466$ et $p_2 = p_3 = 0.0605$ dans CSD-3 contre $q_1 = 0.0466$ dans CSD-2). Pour la figure 4.10, même pour un taux de compression de 6 : 1 pour Y, les performances du système CSD-3 sont plus proches de la limite du Slepian-Wolf que celles obtenues avec CSD-2. En effet, $H(X|Y,Z) = 0.1272$ et $H(Y|Z) = 0.1207$ bits dans CSD-3 alors que $H(T|W) = 0.1207$ bits dans CSD-2 (ceci pour $p_1 = 0.0164$ et $p_2 = p_3 = 0.023$ dans CSD-3 contre $q_1 = 0.0164$ dans CSD-2).

4.4.2 Cas des sources Gaussiennes

4.4.2.1 La source Z n'est pas quantifiée

Les performances débit-distorsion d'un système CSD-3 de trois sources Gaussiennes corrélées sont comparées avec celles d'un CSD-2 de deux sources Gaussiennes W et T (où W et T sont quantifiées). Le modèle de corrélation entre W et T est défini par : $T = W + N_0$ avec $W \sim \mathcal{N}(0,1)$ l'information de bord qui est indépendante de $N_0 \sim \mathcal{N}(0,\sigma_0^2)$. Les sources X, Y et T sont quantifiées en utilisant un quantificateur de Lloyd-Max à 4 niveaux. Afin de pouvoir comparer les performances de la source X dans CSD-3 avec celles de T dans un système CSD-2, les distorsions mesurées de W (D'_W) et Y (D'_Y) doivent avoir la même valeur. Pour cela, le débit de Y est fixé à $\frac{2}{3}$ bit/symbole (désigné par bit/s). La Correlation-SNR entre Y et Z qui permet d'avoir les meilleures performances à ces débits est $CSNR(Y,Z) = 11.91$ dB. Par conséquent, la variance du bruit N_1 est fixée à $\sigma_1^2 = 0.01584$ et la distorsion mesurée sera pour $D'_Y = 0.0095$. Comme l'indique le tableau 3.2, pour avoir $D'_W = D'_Y$, la source W doit être quantifiée avec un quantificateur de Lloyd -Max à 16 niveaux.

Sur les figures 4.11 et 4.13, les taux d'erreur symbole Pe_X (pour CSD-3 : $Pr(X_Q \neq \hat{X}_Q)$) et Pe_T (pour CSD-2 : $Pr(T_Q \neq \hat{T}_Q)$) sont tracés en fonction de la Correlation-SNR respectivement $CSNR(X,Y,Z) = \frac{1}{\sigma^2}$ pour CSD-3 et de $CSNR(T,W) = \frac{1}{\sigma_0^2}$ pour CSD-2.

A partir de la figure 4.11, on peut observer que pour des signaux moins corrélés (ou bien pour un débit élevé, ici 1 bit/s), les performances obtenues avec le système CSD-2 sont meilleures que celles obtenues avec le CSD-3. La distorsion D'_Y provenant de l'information de bord $\hat{Y}$, pénalise les performances de décodage de la source X. Par conséquent, pour le même taux d'erreur symbole, CSD-2 fournit des performances meilleures que CSD-3. Les valeurs de distorsions moyennes tracées en fonction de la Corrélation-SNR pour le débit 1 bit/s sont montrées sur la figure 4.12. Il y a une perte de performance de la distorsion mesurée pour la source X dans un système CSD-3 (pour la même valeur de distorsion, $CSNR(X,Y,Z) = 8.25$ dB avec CSD-3 contre $CSNR(T,W) = 8.23$ dB pour CSD-2).

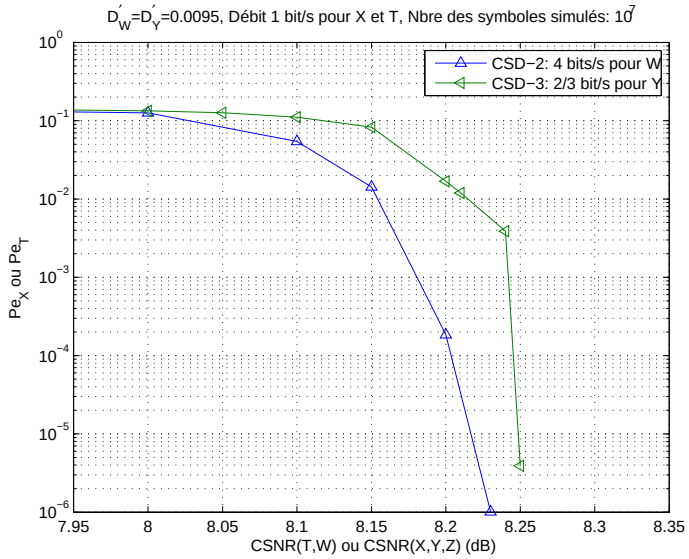

FIG. 4.11 – Taux d'erreur symbole des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes pour un débit de 1 bit/s (les débits de T et W dans CSD-2 sont respectivement 1 bit/s et 4 bits/s et les débits de X et Y dans CSD-3 sont respectivement de 1 bit/s et de $\frac{2}{3}$ bit/s). $D'_Y = D'_W = 0.0095$.

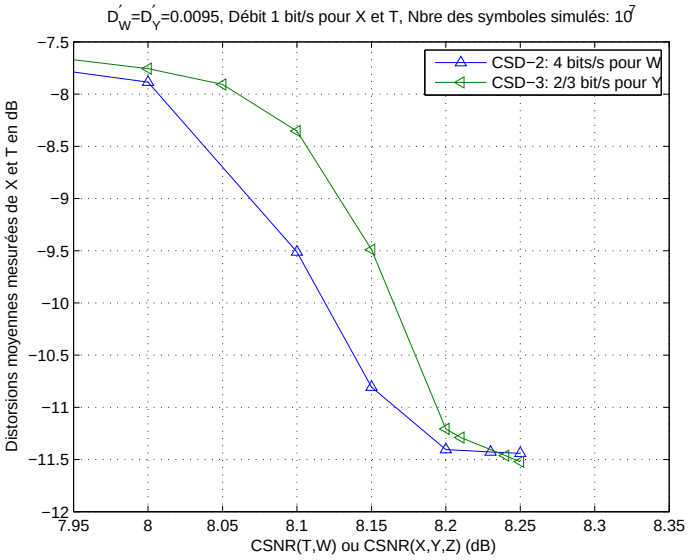

FIG. 4.12 – Distorsion moyenne mesurée des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes pour un débit de 1 bit/s (les débits de T et W dans CSD-2 sont respectivement 1 bit/s et 4 bits/s et les débits de X et Y dans CSD-3 sont respectivement de 1 bit/s et de $\frac{2}{3}$ bit/s). $D'_Y = D'_W = 0.0095$.

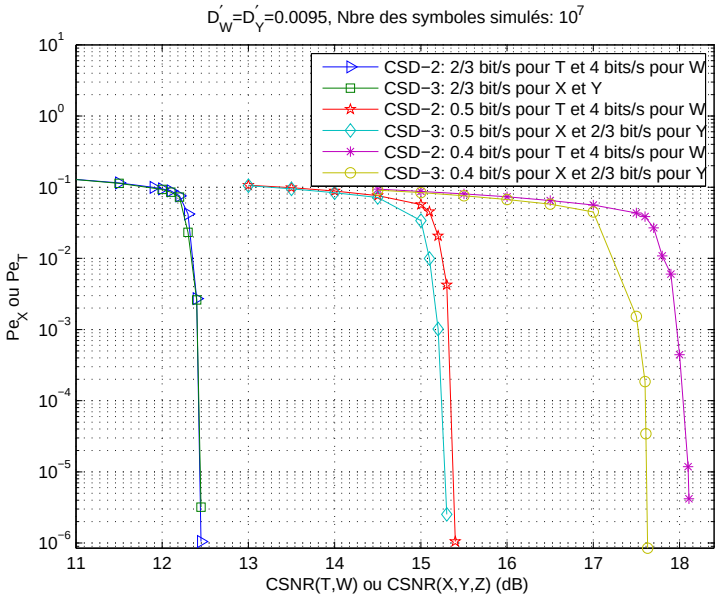

FIG. 4.13 – Taux d'erreur symbole des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes : les débits de Y dans CSD-3 et W dans CSD-2 sont respectivement de $\frac{2}{3}$ bit/s et de 4 bits/s, $D'_Y = D'_W = 0.0095$.

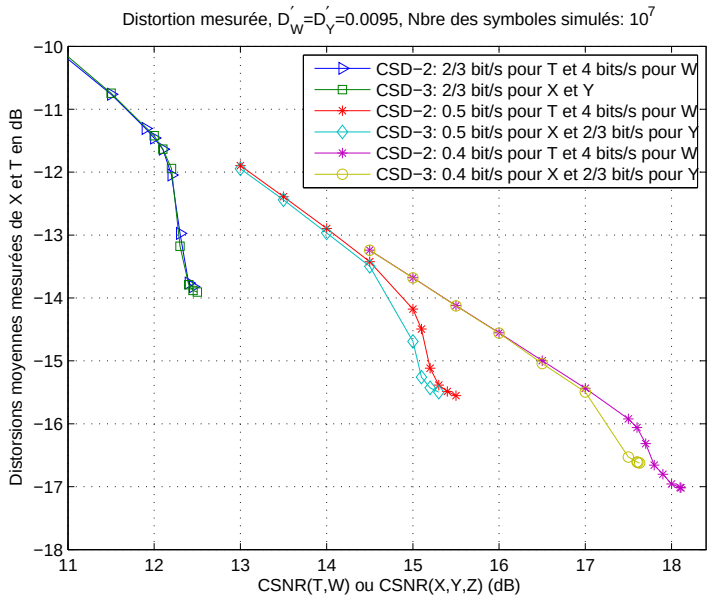

FIG. 4.14 – Distorsion moyenne mesurée des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes : les débits de Y dans CSD-3 et W dans CSD-2 sont respectivement de $\frac{2}{3}$ bit/s et de 4 bits/s, $D'_Y = D'_W = 0.0095$.

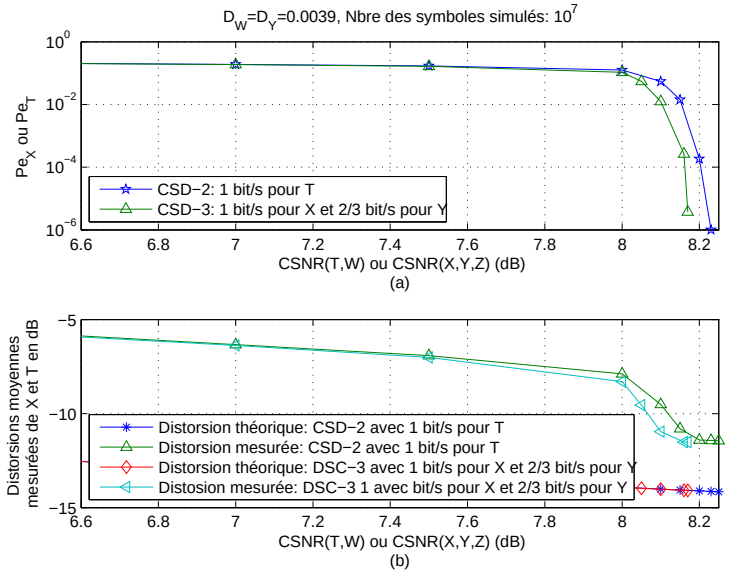

FIG. 4.15 – Résultats de simulation des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes ; les débits de T et W dans CSD-2 sont respectivement 1 bit/s et 4 bits/s, les débits de X et Y dans CSD-3 sont respectivement de 1 bit/s et $\frac{2}{3}$ bit/s : (a) Taux d'erreur symbole, (b) Distorsion moyenne mesurée.

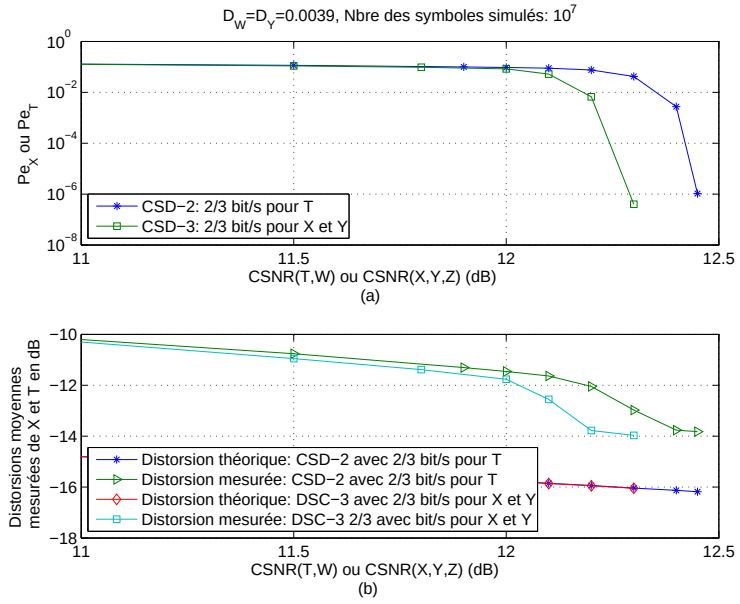

FIG. 4.16 – Résultats de simulation des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes ; les débits de T et W dans CSD-2 sont respectivement $\frac{2}{3}$ bit/s et 4 bits/s, les débits de X et Y dans CSD-3 sont de $\frac{2}{3}$ bit/s : (a) Taux d'erreur symbole, (b) Distorsion moyenne mesurée.

Pour des taux de compressions élevés (débit de transmission plus faible) et des sources très corrélées, les performances obtenues avec le CSD-3 sont meilleures que celles obtenues avec CSD-2. Dans la figure 4.13 et pour un même débit ($\frac{2}{3}$ bit/s), CSD-3 et CSD-2 fournissent le même taux d'erreur symbole $Pe_X = Pe_T = 10^{-5}$ pour des Correlation-SNR quasiment identiques. Le gain de Correlation-SNR avec un système CSD-3 augmente avec le taux de compression (une différence de l'ordre de 0.11 et de 0.487 dB pour respectivement des débits de 0.5 et de 0.4 bit/s par rapport à CSD-2). CSD-3 fournit une meilleure performance que CSD-2 pour des corrélations et des débits globaux plus faibles. La figure 4.14 montre les valeurs des distorsions moyennes mesurées obtenues avec CSD-3 et CSD-2 pour différentes valeurs de débits, 2/3, 0.5 et 0.4 bit/s. Nous pouvons observer que la distorsion entre Y et $\hat{Y}$ a un effet nocif sur la distorsion moyenne entre X et $\hat{X}$ pour un taux plus élevé. Dans ce cas aussi, le système CSD-3 présente des gains de corrélation par rapport au CSD-2 de 0, 0.11 et 0.487 dB pour des débits respectivement de 2/3 , 0.5 et 0.4 bit/s et des taux d'erreur symbole $Pe_X = Pe_T = 10^{-5}$.

Pour des débits élevés, les performances d'un système CSD-3 augmentent quand la distorsion mesurée de Y diminue (par l'augmentation de débit de Y ou pour une corrélation plus grande entre Y et Z). Dans les figures 4.15 et 4.16, les résultats de simulation pour un CSD-3 sont obtenus avec $CSNR(Y,Z) = 14.2$ dB ($\sigma_1^2 = 0.0094$). Ici, les distorsions théoriques de W (D_W) et de Y (D_Y) ont la même valeur 0.0039. On peut remarquer sur les figures 4.15(a) et 4.16(a) qu'avec une grande corrélation entre Y et Z, CSD-3 permet de fournir un gain de 0.06 et 0.15 dB en terme de Correlation-SNR par rapport à CSD-2 et pour des débits respectivement de 1 et de $\frac{2}{3}$ bit/s ($Pe_X = Pe_T = 10^{-5}$). A partir des figures 4.15(b) et 4.16(b), on peut observer qu'avec moins de corrélation, la distorsion moyenne mesurée obtenue avec CSD-3 se rapproche plus de la borne théorique que celle obtenue avec CSD-2.

4.4.2.2 La source Z est quantifiée

Maintenant, nous allons comparer les performances débit-distorsion du nouveau système CSD-3 de trois sources Gaussiennes corrélées et quantifiées avec celles d'un CSD-2 de deux sources Gaussiennes W et T (où W et T sont quantifiées). Les sources X et Y sont quantifiées avec un Lloyd-Max à 4 niveaux. La source Z est quantifiée avec un Lloyd-Max à 32 niveaux. La corrélation entre Y et Z est fixé à $CSNR(Y,Z) = 12$ dB pour avoir la distorsion mesurée de Y, $D'_Y = 0.0095$. Les distorsions, théorique et mesurée, de Z sont très faibles, respectivement 0.00024 et 0.00062. Par conséquent, la fonction de débit distorsion de X est $R^*_{X|\hat{Y},\hat{Z}}(D_X) = \frac{1}{2}\log_2 \frac{\sigma^2+D_Y}{D_X}$ (la même que celle du système présenté dans la section 4.3.2.2).

Les figures 4.17 et 4.18 montrent les performances obtenues en termes respectivement de taux d'erreur symbole et de distorsion moyenne mesurée des systèmes CSD-3 et CSD-2. Les résultats de simulation sont presque identiques à ceux présentés dans 4.4.2.1.

Les débits globaux du système CSD-3 et CSD-2 sont respectivement $R_{3S} = R_{X|\hat{Y},\hat{Z}} + R_{Y|\hat{Z}} + R_{\hat{Z}}$ et $R_{2S} = R_{T|W_Q} + R_{W_Q}$. Par conséquent, les débits moyens par source de CSD-3 et CSD-2 sont respectivement $R_{3S}/3$ et $R_{2S}/2$. Le tableau 4.1 montre les valeurs

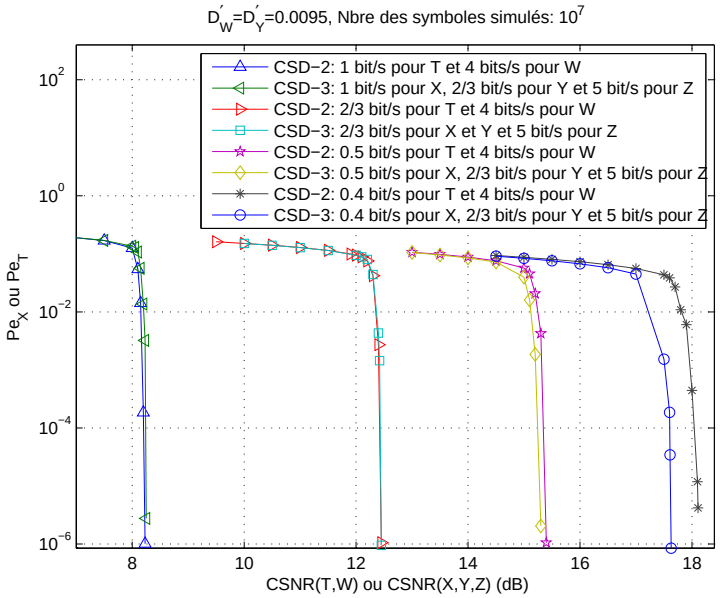

FIG. 4.17 – Taux d'erreur symbole des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes : les débits de Y et de Z dans CSD-3 et de W dans CSD-2 sont respectivement $2/3$ bit/s, 5 bits/s et 4 bits/s, $D'_Y = D'_W = 0.0095$.

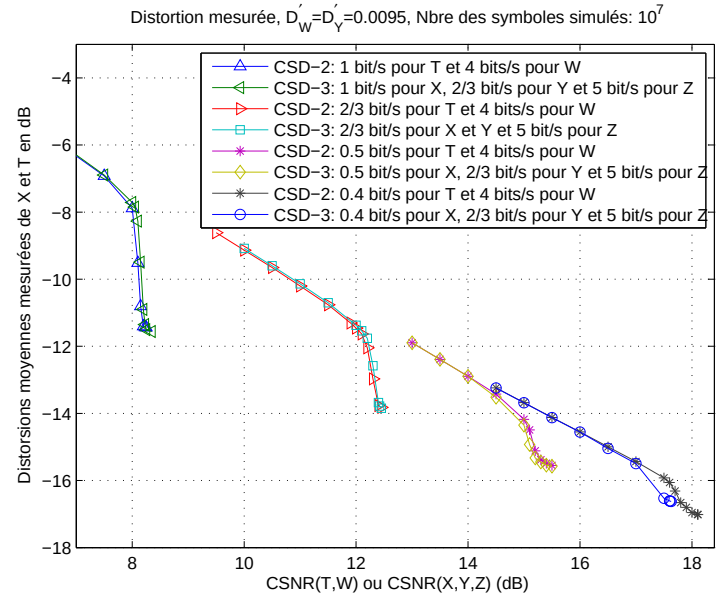

FIG. 4.18 – Distorsion moyenne mesurée des systèmes de codage distribué CSD-3 et CSD-2 avec des sources Gaussiennes : les débits de Y et de Z dans CSD-3 et de W dans CSD-2 sont respectivement $2/3$ bit/s, 5 bits/s et 4 bits/s, $D'_Y = D'_W = 0.0095$.

des débits moyens par source des systèmes CSD-3 et CSD-2 pour différents taux de compression des sources X et T (les débits de Y et Z dans CSD-3 et W dans CSD-2 sont respectivement $\frac{2}{3}$ bit/s, 5 bits/s et 4 bits/s). On remarque que le système CSD-3 présente des débits moyens par source meilleurs que ceux obtenus avec CSD-2. Par conséquent, pour des débits de X et T inférieurs à $\frac{2}{3}$ bit/s, non seulement on a un gain en corrélation mais aussi au niveau du taux de compression.

Pour un taux de compression de 1 bit/s de X et de T (cas où il y a perte de performances en termes du taux d'erreur symbole et de distorsions mesurée pour CSD-3 par rapport à CSD-2), les débits moyens par source obtenus avec le système CSD-3 sont meilleurs que ceux dans CSD-2.

TAB. 4.1 – *Comparaison des débits moyens par source obtenus avec CSD-3 et CSD-2. Les débits de Y et Z dans CSD-3 et de W dans CSD-2 sont respectivement $\frac{2}{3}$ bits/s, 5 bits/s et 4 bits/s*

Débits de X et de T en bit/s	débit moyen par source dans CSD-3 en bit/source	débit moyen par source dans CSD-2 en bit/source
1	2.222	2.5
2/3	2.111	2.333
0.5	2.055	2.25
0.4	2.022	2.2

4.5 Conclusion

Nous avons présenté dans ce chapitre les schémas de codage distribué de trois sources corrélées binaires ou Gaussiennes utilisant un code turbo poinçonné. L'extension au cas de trois sources corrélées offre de nouvelles perspectives d'application du codage de sources distribuées.

Dans un premier temps, une généralisation du théorème de Slepian-Wolf au cas de trois sources corrélées binaires a été présentée. A partir des résultats de simulation de la technique du codage distribué de trois sources binaires, nous avons montré que le gain apporté par celle-ci est de permettre la compression des sources avec des corrélations plus faibles que dans le cas de deux sources. Le débit moyen par source obtenu avec un système à trois sources est meilleur que celui obtenu dans le cas de deux sources.

Dans un deuxième temps, des extensions du théorème de Wyner-Ziv au cas de trois sources corrélées et Gaussiennes avec une information de bord parfaite (Z est non quantifiée) ou partielle (Z est quantifiée) ont été présentées. Pour des faibles distorsions de Z dans un schéma de codage distribué avec 3 sources quantifiées, la fonction débit-distorsion est égale à celle du schéma de codage où seulement X et Y sont quantifiées.

Pour des sources Gaussiennes avec Z non quantifiée, l'impact de la distorsion introduite par la quantification de la deuxième source (Y) sur les performances débit-

distorsion du signal principal (X) a été évalué. Pour les mêmes débits et la même Correlation-SNR, nous avons montré que des performances élevées peuvent être atteintes (des valeurs de distorsion plus faibles pour le signal principal X) avec un codeur distribué de trois sources Gaussiennes meilleures que celles dans le cas de codage de Wyner-Ziv de deux sources avec information de bord partielle. La "Qualité", dans le sens de débit-distorsion, de l'information de bord utilisée pour décoder le signal principal s'avère être meilleure dans un système de codage distribué avec trois sources.

Un schéma de codage distribué avec trois sources Gaussiennes concret est obtenu lorsque celles-ci sont quantifiées. Les informations de bord dans ce schéma sont partielles. Toutefois, pour des faibles distorsions mesurées de Z, nous avons obtenu les mêmes performances débit-distorsion de la source X que celles obtenues avec un codeur de Wyner à trois sources où Z est non quantifiée. Le schéma de codage distribué avec trois sources Gaussiennes quantifiées, nous a permis d'évaluer le débit moyen par source. Dans un système à trois sources Gaussiennes, le débit moyen par source reste meilleur que celui obtenu dans le cas de deux sources.

Dans les schémas de codage distribué de trois sources corrélées binaires ou Gaussiennes, nous avons remarqué que plus le taux de compression est élevé, meilleures seront les performances par rapport au cas de deux sources.

Chapitre 5

Etat de l'art du codage vidéo distribué

5.1 Introduction

Bien que l'étude du codage de source distribuée remonte au début des années 70, ce n'est que très récemment que les efforts vers des réalisations pratiques (solutions faisables) du codage vidéo distribué (désigné aussi par codage vidéo de Wyner-Ziv) ont été proposés. Le but est de développer des schémas de codage vidéo adaptés aux applications de communication radio-mobile dont la complexité de l'encodeur est très faible.

Une première approche connue sous le nom de PRISM (Power-efficient, Robust, hIghcompression, Syndrome-based Multimedia coding) a été proposée par Puri et Ramchandran [PR02], [PR03c], [PR03b] pour la transmission de multimédia sur les réseaux sans fil en utilisant des syndromes. Le but principal de cette approche est de combiner la faible complexité et la robustesse du codage des images en mode Intra avec l'efficacité de compression des images en mode Inter.

En 2002, se servant des codes turbo, Aaron, Zhang et Girod [AZG02] ont présenté les résultats d'un codeur vidéo utilisant un schéma de codage en mode Intra et de décodage en mode Inter où différentes images sont indépendamment codées mais sont conjointement décodées. En 2003, Zhu, Aaron et Girod [ZAG03] ont proposé une approche du codage vidéo distribué à basse complexité sous le nom de "distributed compression for large cameras arrays". Dans cette solution, des vues multiples et corrélées d'une scène sont indépendamment codées avec un codeur de Wyner-Ziv dans le domaine pixel mais sont conjointement décodées à un nœud central. Zhu *et al.* ont effectué dans [ZAG03] une comparaison de performances entre le codeur de Wyner-Ziv dans le domaine pixel et un système de codage utilisant la norme JPEG-2000. Les résultats démontrent que pour des débits faibles, la solution présentée par Zhu *et al.* atteint des valeurs de $PSNR$ meilleures que JPEG-2000 avec une complexité de codage inférieure. Pour plus de détails, le lecteur peut se référer au [ZAG03].

En 2004, Aaron, Rane, Setton et Girod [ARSG04] ont proposé une architecture

semblable à celle de [AZG02] en utilisant la transformée en cosinus discrète (DCT : Discrete Cosine Transform). Les résultats obtenus avec la nouvelle solution de codage sont meilleurs en terme d'efficacité de codage par rapport à ceux de [AZG02] (au coût d'une complexité légèrement élevée au codage associé à la transformée DCT).

Une autre solution de codage vidéo distribué avec une faible complexité à l'encodeur a été proposée par Aaron, Rane et Girod [ARG04], [AG04]. Dans cette solution, mis à part le bitstream résultant du processus de codage des images, l'encodeur transmet également des informations supplémentaires sur l'image courante pour aider le décodeur lors de la phase d'estimation de mouvement.

En 2004, Rane, Aaron et Girod [RAG04] ont présenté une autre approche visant la robustesse de la transmission des images vidéo. Spécifiquement, le but de cette approche est de rendre le bitstream obtenu après le codage séparé plus résistant aux erreurs de transmission d'un canal bruité. Dans ce cas, les mêmes images sont codées à la fois avec un codeur vidéo distribué et un codeur conventionnel (codeur MPEG dans [RAG04]). Les bits générés avec les deux codeurs passent dans un canal bruité. Le décodage s'effectue d'une manière conjointe pour reconstruire les images codées. Cette approche peut être vue comme une application de codage systématique source-canal.

Fondamentalement, il y a deux principaux secteurs d'intérêt au codage vidéo distribué, la faible complexité de l'encodeur (comme dans [ARSG04] et [ARG04]), et la robustesse de la transmission (comme dans [PR02], [PR03c], [PR03b] et [RAG04]). En pratique, les codeurs vidéo distribués sont utilisés dans deux domaines différents : domaine pixel ([AZG02], [ARZG03], [ASG03], [ARRMG03]) et domaine transformé ([PR02], [RAG04], [ARSG04], [AG04] et [SA03]). Pour améliorer les performances de décodage, certains schémas de codage vidéo distribué utilisent une voie de retour. Par conséquent, nous classons les solutions proposées en deux classes : solution avec voie de retour [AZG02] et solution sans voie de retour [PR02]. Dans ce chapitre, nous présentons ces différentes approches.

5.2 Codage vidéo : techniques usuelles

Une séquence vidéo est une succession d'images sur l'axe temporel. Ces images successives présentent de fortes corrélations appelées corrélations temporelles. Au sein d'une image, les pixels voisins sont corrélés. Cette corrélation est appelée corrélation spatiale. Le principe du codage vidéo se base sur l'élimination de la redondance spatio-temporelle existant entre les images successives d'une séquence vidéo. Pour le codage, chaque image est décomposée en blocs de pixels appelés macro-blocs de taille 16×16. Il y a deux modes de codage des images d'une séquence vidéo :

1. Codage d'image en mode Intra (I) : le codage, dans ce cas, exploite seulement la corrélation spatiale présente dans l'image à coder. Une transformée DCT, qui permet de décorréler spatialement les données à coder, est appliquée à chaque bloc de l'image. Les coefficients issus de la transformée sont quantifiés et codés en utilisant un codeur entropique (par exemple un code de Huffman). Vu que ce mode de codage n'exploite pas la corrélation temporelle dans la séquence vidéo,

le taux de compression ne peut pas atteindre des valeurs très élevées. Cependant, l'avantage de ce mode de codage est d'une part la faible complexité de l'encodeur et d'autre part la robustesse contre les erreurs de transmission (Chaque image est codée indépendamment des autres images de la séquence vidéo).

2. Codage d'image en mode Inter ou prédictive (P) : à l'inverse du codage en mode Intra, la prédiction exploite les corrélations spatiales et temporelles présentent dans une séquence vidéo. L'exploitation de la corrélation temporelle est réalisée avec l'opération d'estimation de mouvement. En effet, pour chaque bloc b_i de l'image courante, une recherche exhaustive est effectuée dans la précédente image déjà codée (en mode Intra ou Inter) afin de trouver le meilleur bloc proche (le plus ressemblant) en terme de mesure d'erreur quadratique à b_i. Ce plus proche bloc est utilisé pour prédire le bloc courant. Une erreur de prédiction entre les deux blocs (courant et le plus proche) est déterminée et codée en utilisant une DCT suivie par un quantificateur et un codeur entropique. Les coordonnées du bloc le plus proche dans l'image précédente, appelées vecteurs de mouvement, sont transmises au décodeur. Au décodage, une fois la reconstruction de l'image précédente est réalisée, l'estimée de l'image courante est déterminée grâce aux vecteurs de mouvement et l'erreur de prédiction codée. Ce mode de codage peut atteindre des taux de compression élevés. Par contre, il présente deux inconvénients majeurs. D'une part, la complexité qui est due principalement à l'estimation de mouvement de l'encodeur est grande, et d'autre part, la perte des vecteurs de mouvement fragilise la robustesse et rend le processus de reconstruction des images P difficile.

Un autre mode de codage d'image est appelé mode bi-directionnel (B). Les images codées en B sont déterminées par prédiction bi-directionnelle des images adjacentes I ou P.

La séquence vidéo est découpée en groupes d'images appelés GOP (Group Of Pictures). Chaque GOP est composé d'images codées en mode I, P et/ou B. La première image d'un GOP est codée en mode Intra pour permettre l'accès aléatoire aux flux, et la robustesse en terme de codage indépendant de toutes les autres images du même GOP.

5.3 Codage vidéo distribué sans voie de retour : PRISM

L'approche présentée par le groupe du Berkeley [PR02], sous le nom de PRISM, vise à combiner le codage d'images en mode Intra (faible complexité de codage et robustesse aux erreurs de transmission) avec l'efficacité de compression en mode Inter. La première image d'un GOP est codée en mode Intra (codage classique). Les autres images du GOP utilisent le système de codage de Wyner-Ziv. Le processus d'estimation de mouvement pour générer l'information de bord se réalise au décodage. Il est basé sur la précédente image déjà décodée.

Cependant, l'approche PRISM utilise le concept d'information de bord d'une façon différente de la description donnée au chapitre 2. L'innovation de la solution PRISM est d'avoir plusieurs candidats d'information de bord de telle façon que lors de la procédure de décodage, le meilleur élément est sélectionné pour la reconstruction de l'image

courante. En effet, une information supplémentaire obtenue grâce au code de contrôle de redondance cyclique (noté CRC, ou en anglais Cyclic Redundancy Check) est envoyée de l'encodeur vers le décodeur. Le code CRC permet d'aider le décodeur à décider quel est le meilleur candidat pour l'information de bord à prendre en compte pour la reconstruction. Ainsi, l'approche présentée par le groupe du Berkeley ne demande pas l'utilisation de la voie de retour pour améliorer la reconstruction des images codées.

Nous allons brièvement présenter l'architecture de l'encodeur et du décodeur de la solution PRISM pour le codage vidéo distribué. Plus de détails se trouvent dans [PR02], [PR03c], [PR03b], [Pur02] et [PR].

5.3.1 Codage

Considérons l'architecture de codage du système PRISM illustrée sur la figure 5.1. Chaque image de la séquence vidéo est d'abord divisée en blocs de pixels. La taille des blocs est 8×8 ou 16×16. Avant de présenter les différentes opérations effectuées par l'encodeur de PRISM pour coder une image, une étape appelée classification est réalisée.

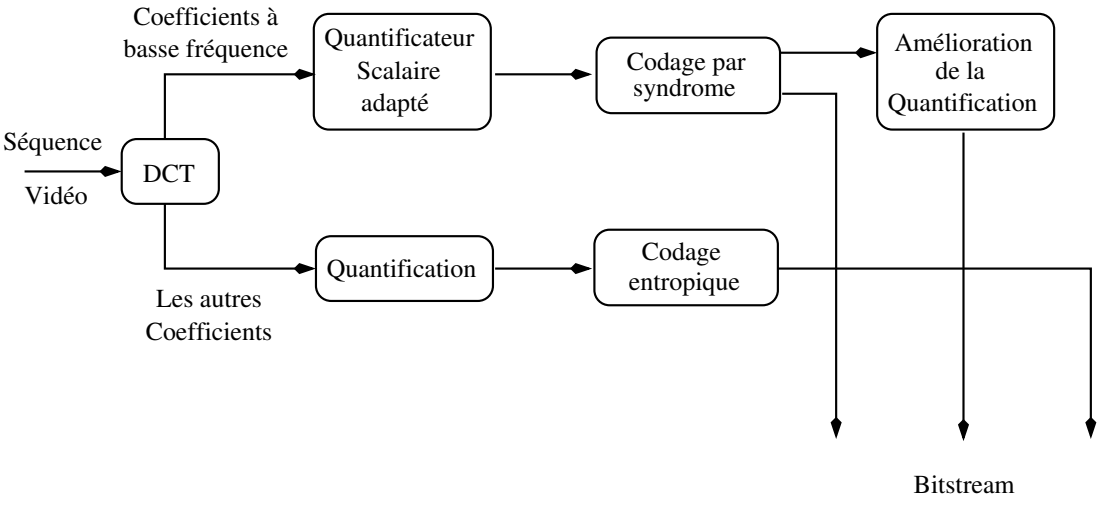

FIG. 5.1 – Schéma de codage du système PRISM [PR].

Classification

Dans l'étape de classification, chaque bloc 8×8 ou 16×16 est classifié dans une des classes prédéfinies. Cette classification est réalisée selon la corrélation (dépendance statistique ou bien bruit de corrélation) entre chaque bloc d'image courante et son correspondant (même position) dans l'image précédente. Dans [PR], la valeur de la variance du bruit de corrélation est déterminée par l'erreur quadratique entre chaque bloc d'image courante et son correspondant dans l'image précédente. L'étape de classification permet de décider quel codage est adapté pour chaque bloc de l'image courante : aucun codage (classe aucun codage), codage source traditionnel (classe de codage en mode Intra) ou codage par syndrome (classe de codage par syndrome ou Wyner-Ziv).

Ainsi, les blocs d'images classifiés dans la classe aucun codage (corrélation temporelle grande) ne sont ni codés ni transmis. Les blocs classifiés dans la classe de codage en mode Intra (corrélation très faible) sont traditionnellement codés. Par contre, les blocs classifiés dans la classe de codage par syndrome sont codés avec un codeur de Wyner-Ziv basé sur la technique DISCUS. Une attention particulière sera donnée aux procédures d'encodage/décodage de ces blocs. Les classes de mode de codage choisies pour les blocs d'une image courante sont alors transmises au décodeur comme information d'en-tête.

1) Dé-corrélation spatiale par transformée DCT

Une transformée DCT est appliquée à chaque bloc de l'image courante. Les coefficients de la transformée de chaque bloc sont par la suite scrutés en zig-zag.

Habituellement, la majeure partie de l'énergie d'un bloc d'image est concentrée dans les coefficients de la transformée représentés par de faibles fréquences. Cette caractéristique est utilisée par l'approche PRISM. En effet, pour les blocs classifiés dans la classe de codage par syndrome, la solution PRISM, comme l'indique la figure 5.2, encode les coefficients de basse fréquence en utilisant le codeur de Wyner-Ziv, tandis que ceux de haute fréquence sont codés classiquement (c'est-à-dire quantifiés puis codés avec un codeur entropique). Typiquement, plusieurs coefficients de haute fréquence ont des valeurs proches de zéro et donc le codage entropique va utiliser peu de bits pour les compresser. Par contre, les coefficients de basse fréquence ont des valeurs élevées et le codage par syndrome permettra de réduire le taux binaire requis pour les transmettre (seul le syndrome est transmis au décodeur comme dans DISCUS).

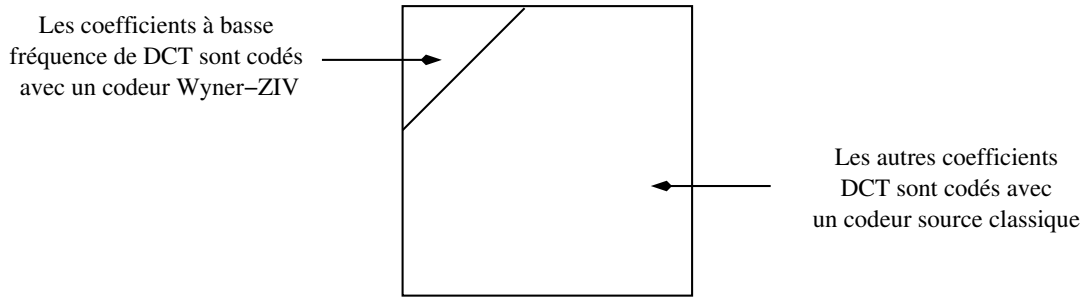

FIG. 5.2 – Sélection des coefficients DCT qui seront codés avec un codeur de Wyner-Ziv dans PRISM. Les coefficients restants sont codés avec un codeur de source classique [PR].

2) Quantification

Les coefficients de la transformée, qui sont scrutés en zig-zag, sont quantifiés. Le coefficient DC (le coefficient de la transformée avec la plus basse fréquence) et un nombre restreint de coefficients AC (présentant la basse fréquence) sont quantifiés en utilisant le quantificateur scalaire adapté. Le choix des pas de quantification de ces coefficients est dépendant du niveau de corrélation de chaque bloc 8×8 ou 16×16 de l'image courante déterminé dans l'étape de classification. En effet, si l'amplitude du bruit de corrélation est supérieure au pas de quantification, alors le décodeur pourrait ne pas estimer correctement la version quantifiée même en utilisant l'information de bord. Les coefficients restants de la transformée (coefficients à haute fréquence qui représentent

80 % du nombre total de tous les coefficients) sont quantifiés avec un quantificateur scalaire dont le pas dépend de la qualité de reconstruction (distorsion) désirée. Par conséquent, la différence entre les deux quantificateurs de la figure 5.1 est au niveau du pas de quantification.

3) Codage par syndrome

La compression des coefficients de basse fréquence (environ 20 % du nombre total des coefficients) quantifiés avec un quantificateur adapté est réalisée grâce au codage par syndrome. L'approche PRISM utilise le principe du codeur DISCUS. Un treillis à 128 états d'un codeur convolutionnel de taux 1/2 est utilisé et seulement le syndrome obtenu à la fin du codage est transmis au décodeur. Les bits de syndrome sont incorporés dans le bitstream comme l'indique la figure 5.3.

FIG. 5.3 – Contenu du bitstream après codage vidéo utilisant la solution PRISM [PR].

4) Amélioration de la quantification

La compression d'un signal avec différents pas de quantification permet d'atteindre une meilleure qualité de reconstruction (distorsion désirée) au décodeur. Comme il a été mentionné ci-dessus, pour les coefficients de basse fréquence qui sont codés par syndrome, le choix du pas de quantification est limité par la corrélation déterminée à l'étape de classification afin de réduire au minimum la probabilité d'erreur de décodage. Pour atteindre une qualité souhaitable de reconstruction, le pas de quantification du quantificateur adapté est raffiné (amélioration de la quantification). Le processus d'amélioration de quantification est juste une subdivision progressive de l'intervalle de base du quantificateur, en des intervalles de taille égale correspondants au pas de quantification cible. L'indice associé à l'intervalle amélioré (ou raffiné), faisant partie de l'intervalle de base, est transmis au décodeur. Les bits d'amélioration (indice associé aux intervalles raffinés) forment une autre composante du bitstream de la figure 5.3.

5) Codage entropique

Les coefficients de haute fréquence quantifiés avec un quantificateur scalaire sont traditionnellement codés en utilisant un codeur à longueur variable (CLV) : codeur de Huffman. Les bits générés, appelés bits du codeur de source, sont incorporés dans le bitstream comme l'indique la figure 5.3.

5) Code CRC

Ici et contrairement au codage vidéo conventionnel, c'est le décodeur qui effectue l'estimation de mouvement. Cependant, pour chaque bloc de l'image courante codé, le décodeur ne connaît pas la "meilleure" information de bord générée par estimation de mouvement de l'image précédente. Pour remédier à ce problème et améliorer la tâche de l'estimation de mouvement, l'encodeur transmet non seulement le syndrome mais également un contrôle par redondance cyclique (CRC de force suffisante) de la séquence des mots de code obtenus après quantification. Ce contrôle sert comme une "signature"

de la séquence des mots de code. Les bits de contrôle de CRC sont introduits dans le bitstream (figure 5.3) sous le nom de bits de CRC.

Pour résumer, la complexité du codeur dépend essentiellement des opérations de la transformée et de codage entropique [PR03c]. Par conséquent, cette complexité est semblable à celle d'une solution traditionnelle de codage en mode Intra.

5.3.2 Décodage

L'architecture du décodeur de PRISM est présentée à la figure 5.4. Pour la classe aucun codage, les blocs reconstruits sont égaux à ceux ayant la même position dans l'image précédante.Pour les blocs classifiés dans la classe de codage en mode Intra, le décodage traditionnel à savoir décodage entropique et déquantification (opérations inverses à celles effectuées à l'encodeur) est appliqué. Quant aux blocs classifiés de la classe de codage de syndrome, le procédé de décodage est décrit dans les cinq étapes suivantes :

1) Estimation de mouvement

La tâche la plus importante exécutée au décodeur de PRISM est l'estimation de mouvement qui fournit les informations de bord nécessaires pour décoder efficacement le mot de code associé au syndrome reçu. Les informations de bord (le meilleur bloc parmi tous ses voisins) sont obtenues à partir de l'estimation de mouvement à $\frac{1}{2}$-pixel de l'images reconstruite précédemment.

2) Décodage par syndrome

Chaque information de bord générée par l'opération d'estimation de mouvement est utilisée en association avec les bits de syndrome reçus pour décoder les mots de code correspondant à la version quantifiée des coefficients à basse fréquence. Comme dans DISCUS, vu qu'un codeur convolutionnel (avec un treillis à 128 états) est utilisé lors du codage, le décodage peut être réalisé en utilisant l'algorithme de Viterbi [Vit67], [For73]. Cet algorithme est alors employé pour identifier la séquence des mots de code la plus proche de l'information de bord utilisée. Un contrôle de détection d'erreur est effectué sur la séquence décodée en utilisant le code CRC. Si le CRC est valide, alors le décodage est déclaré bien réussi. Autrement, une autre information de bord est utilisée et le processus entier de décodage par syndrome est répété. Le CRC est employé comme une signature fiable et unique pour chaque bloc et peut identifier facilement la meilleure information de bord. Dans PRISM, le décodeur n'a pas besoin d'une voie de retour pour améliorer les performances de décodage. Cependant, un décodage multi-hypothèse qui entraîne l'utilisation de décodage canal multiple est utilisé.

3) Amélioration de la déquantification

Après l'estimation de la séquence des mots de code, la reconstruction des coefficients de basse fréquence est effectuée en utilisant un déquantificateur. Afin d'avoir une meilleure qualité de reconstruction (faible distorsion), la déquantification utilise la séquence des mots de code estimée et les bits d'amélioration transmis par l'encodeur. À cette étape, un algorithme d'estimation linéaire est exécuté pour retrouver les coefficients de Wyner-Ziv estimés.

4) Décodage entropique et déquantification associée

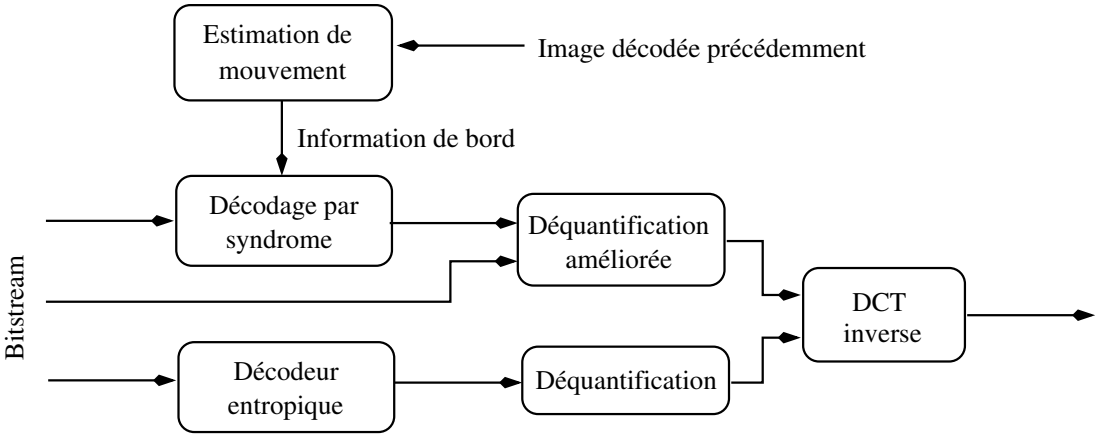

FIG. 5.4 – Schéma de décodage du système PRISM [PR].

Les bits reçus du codeur de source (correspondants au codage en mode Intra des versions quantifiées des coefficients à haute fréquence) sont décodés en utilisant des opérations inverses à celles effectuées à l'encodeur à savoir un décodage entropique et une reconstruction utilisant un déquantificateur.

5) Transformée DCT inverse

Après le balayage en zig-zag inverse des coefficients estimés, la transformée DCT inverse est alors appliquée pour retrouver les pixels des blocs reconstruits. Ainsi, le processus de décodage de PRISM est terminé.

5.3.3 Performances

Afin d'évaluer les performances du système PRISM, les auteurs dans [PR] ont codé les 15 premières images des séquences Mother and Daughter (352×288), Carphone (176×144) et football (352×240). La première image de chaque séquence vidéo est codée en mode Intra. Les performances de PRISM ont été comparées à celles d'un système de codage vidéo H.263+ (codage en mode Intra et Inter). Pour les trois séquences, les performances de PRISM sont situées entre celles du codeur de H.263+ avec un codage en modes Inter et Intra. Pour un même débit, le gain en performances de PRISM par rapport au codeur de H.263+ avec un codage en mode Intra est de 1.7, 2.5 et plus de 8 dB, respectivement pour les séquences football, Carphone et Mother and Daughter.

Les auteurs dans [PR] ont testé la robustesse de la solution PRISM proposée. Ils ont montré l'impact de la perte d'une image de la séquence football lors du codage avec les systèmes PRISM et H.263+. La robustesse de PRISM est beaucoup plus élevée que celle de H.263+. En effet, étant donné qu'il n'y a pas de propagation d'erreur entre les images (due à l'absence de la prédiction à l'encodeur), la perte d'une image en utilisant PRISM a un effet négligeable sur la qualité de la séquence vidéo décodée.

5.4 Codage vidéo distribué avec voie de retour

Une architecture avec voie de retour a été proposée par l'université de Stanford avec une mise en œuvre dans le domaine pixel ou dans le domaine transformé. Plus de détails de cette architecture peuvent être trouvés dans [GARRM05].

5.4.1 Domaine pixel

5.4.1.1 Codage

La figure 5.5 représente l'architecture du codeur vidéo distribué utilisée dans le domaine pixel [AZG02], [ASG03].

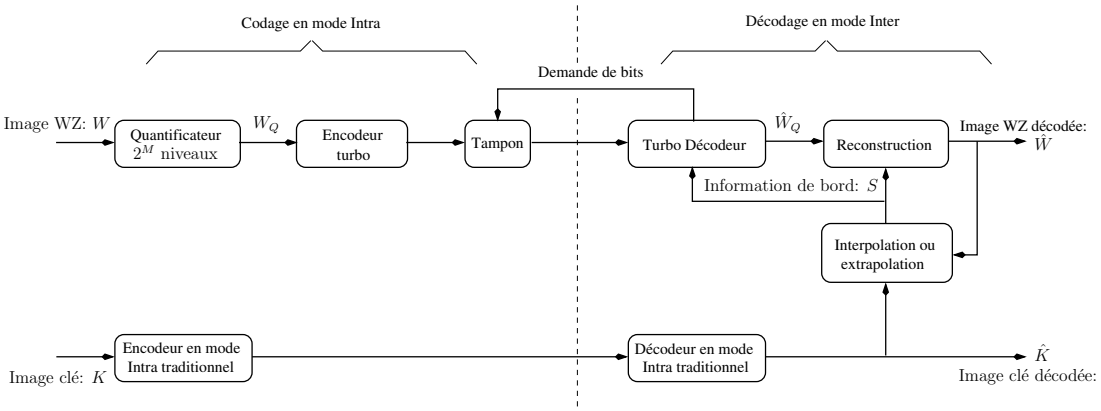

FIG. 5.5 – Schéma de codage/décodage de la solution d'un codeur vidéo distribué dans le domaine pixels de l'université de Stanford.

Pour une séquence vidéo donnée avec un $GOP = 2$, l'ensemble des images est divisé en deux sous-ensembles comme l'indique la figure 5.5. Un premier sous-ensemble d'images impaires, régulièrement espacé dans la séquence, va être codé en mode Intra et constitué des images clés ("Key Frames") désignées par K. Ces images sont codées et décodées en utilisant un codeur en mode Intra traditionnel. Le deuxième sous-ensemble d'images (l'ensemble des images paires qui se trouvent entre les Key Frames) est codé en utilisant le codeur de Wyner-Ziv. Ces images, désignées par images W (images Wyner-Ziv) sont codées en mode Intra et décodées en mode Inter.

Les pixels de chaque image Wyner-Ziv, W, sont quantifiés avec un quantificateur scalaire uniforme à 2^M niveaux. Les symboles quantifiés, W_Q, sont codés avec un codeur turbo poinçonné. Seuls les bits de parité sont stockés dans une mémoire tampon. Le nombre de bits de parité transmis au décodeur dépend de la qualité de l'information de bord S.

5.4.1.2 Décodage

Les images K sont décodées avec un décodeur en mode Intra traditionnel. Pour chaque image Wyner-Ziv W codée, le décodeur produit une information de bord, image S, par interpolation moyennée à partir des images Key Frames adjacentes décodées [ASG03]. Dans [AZG02], la première technique d'interpolation utilisée était une simple moyenne des pixels de Key Frames K se trouvant dans la même position que le pixel de l'image W. Pour utiliser l'information de bord S au décodeur turbo, le modèle de corrélation considéré entre les pixels de l'images W et leurs correspondants des images S (l'information de bord) est une distribution Laplacienne. L'estimation des paramètres de la distribution Laplacienne est déterminée en observant les statistiques des images précédemment décodées.

Le décodeur turbo utilise l'information de bord S et les bits de parité reçus pour récupérer les symboles quantifiés $\hat{W}_Q$. Si le décodeur n'arrive pas à décoder les symboles originaux, alors une demande de bits de parité additionnels de la mémoire tampon du codeur de Wyner-Ziv sera effectuée via le canal de rétroaction "feed-back". Ainsi, le canal de rétroaction est nécessaire pour adapter le débit de transmission aux statistiques de dépendance entre l'information de bord et l'image à coder. Les processus de décodage et de demande de bits de parité sont répétés jusqu'à ce qu'une probabilité d'erreur symbole acceptable soit atteinte.

Le débit binaire d'une image Wyner-Ziv est déterminé en fonction de la corrélation entre celle-ci et l'information de bord S correspondante. Puisque l'information de bord est disponible seulement au décodage, la décision sur le nombre des bits de parité à transmettre ne peut être prise que par le décodeur turbo et non par l'encodeur. Ainsi, le mécanisme de contrôle de débit est réalisé au niveau du décodeur. L'avantage du contrôle de débit au décodeur est de réduire au minimum le fardeau de cette tâche à l'encodeur. Cependant, ce mécanisme exige une voie de retour, et par conséquent une latence plus élevée. Ainsi, ce type de contrôle de débit n'est pas approprié aux applications telles que les systèmes vidéo d'acquisition et de stockage où les images compressées sont transférées et décodées en temps différé.

Après le décodage turbo (les pixels quantifiés sont estimés), chaque pixel de l'image W est reconstruit selon les valeurs du pixel de l'information de bord S et le symbole décodé $\hat{W}_Q$ tel que $\hat{W} = E[W|\hat{W}_Q, S]$ ($\hat{W}$ est l'image reconstruite). En effet, le pixel reconstruit a trois valeurs possibles :

1. une valeur proche ou égale au pixel de l'information de bord, si le pixel de l'information de bord S et le symbole décodé $\hat{W}_Q$ sont dans la même région de quantification (fixée par le symbole décodé $\hat{W}_Q$).
2. la valeur de l'extrémité inférieure de la région de quantification, si la valeur du pixel de l'information de bord S est inférieure à l'extrémité inférieure de l'intervalle de quantification du symbole estimé $\hat{W}_Q$ (le pixel de l'information de bord S est en dehors de la région de quantification).
3. la valeur de l'extrémité supérieure de l'intervalle de quantification du symbole estimé $\hat{W}_Q$, si la valeur du pixel de l'information de bord S est supérieure à cette extrémité.

Ce type de fonction de reconstruction a l'avantage de limiter la distorsion du codeur vidéo distribué. Plus le nombre de niveaux de quantification 2^M est élevé, moins il y a d'erreur lors de la reconstruction.

5.4.1.3 Performances de la solution proposée

Les performances débit-distorsion du codeur vidéo de Wyner-Ziv dans le domaine pixel de la solution de Stanford se situent entre celles des codeurs vidéo conventionnels en modes Inter et Intra. Dans [GARRM05], au même débit et pour les séquences Hall Monitor et Salesman, la solution proposée présente un gain de performances en terme de $PSNR$ de 2 à 5 dB par rapport au codeur vidéo traditionnel en mode Intra. Cependant, les performances du codeur H.263+ en mode Inter restent meilleures que celles de la solution proposée.

5.4.2 Domaine transformé

5.4.2.1 Codage

L'extension du codeur vidéo de Wyner-Ziv du domaine pixel, décrit dans [AZG02] et [ASG03], au domaine transformé a été réalisée dans [ARSG04]. Une transformée DCT qui permet de décorréler spatialement les données à coder est appliquée à l'image Wyner-Ziv W comme l'indique la figure 5.6. La transformée DCT permet au codeur de Wyner-Ziv d'atteindre une meilleure performance en termes de débit-distorsion.

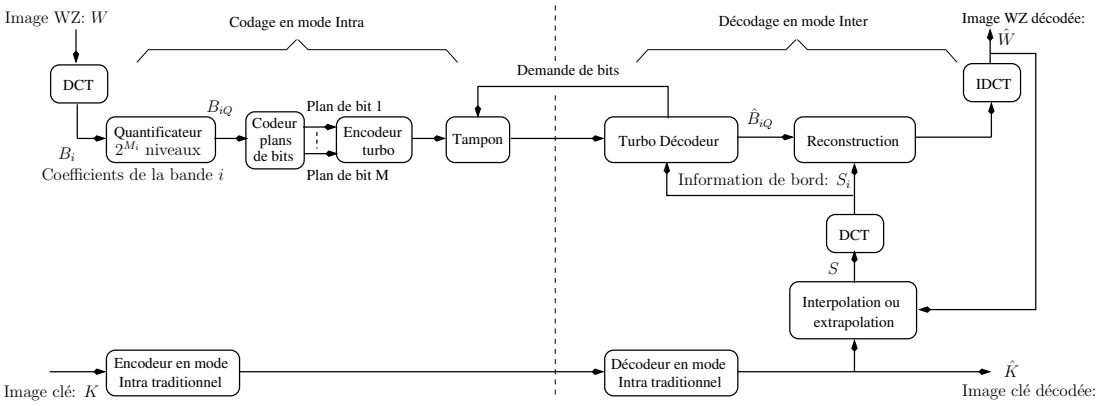

FIG. 5.6 – Schéma de codage/décodage de la solution d'un codeur vidéo distribué dans le domaine transformé de l'université de Stanford.

Dans [ARSG04], les blocs de 4×4 pixels des images Wyner-Ziv W sont transformés en blocs de 4×4 coefficients grâce à la DCT. Les coefficients de la transformée de chaque bloc sont groupés en sous-bandes (bande i contient les coefficients B_i, $i = 1, 2, ..., 16$). La bande 1 contient les coefficients les plus significatifs alors que la bande 16 est composée de coefficients à valeur très faibles. Chaque bande est codée indépendamment des autres.

Pour chaque bande i, les coefficients B_i sont quantifiés en utilisant un quantificateur scalaire uniforme à 2^{M_i} niveaux. Les symboles quantifiés, B_{iQ}, sont convertis en des plans de bits (les bits de même poids sont regroupés ensemble, exemple : le plan de bit 1 est formé par les bits de poids fort, MSB pour Most Significant Bit) de longueur constante. En tout, il y a M_i plans de bits. Chaque plan de bits est codé avec un codeur turbo poinçonné. Les bits de parité générés sont mis dans une mémoire tampon et ne seront transmis au décodeur que sur une demande décidée après le décodage (selon la corrélation de l'information de bord et l'image Wyner-Ziv).

5.4.2.2 Décodage

Comme dans le domaine pixel, les images Key Frames K sont codées et décodées avec un codeur en mode Intra traditionnel. Les images Key Frames décodées sont utilisées pour former les images S. En effet, pour chaque image Wyner-Ziv W codée, une interpolation compensée en mouvement des images Key Frames, adjacentes à W et qui sont déjà décodées, est effectuée pour générer l'image S correspondante. Une transformée DCT est appliquée aux images S. Les coefficients DCT obtenus sont regroupés ensemble pour former les bandes. Chaque bande i issue de l'image S est composée des coefficients S_i, $i = 1, 2, ..., 16$, qui sont les informations de bord correspondants aux B_i. Le modèle de corrélation assumé par le décodeur turbo entre les coefficients DCT des images W et leurs correspondants des images S est une distribution Laplacienne.

Le symbole décodé $\hat{B}_{iQ}$, associé aux coefficients B_i de la bande i de l'image W, est obtenu par le processus de décodage turbo. Pour chaque bande, le décodeur turbo commence d'abord à décoder le plan de bits 1 (les bits les plus significatifs) suivi par un décodage séquentiel des plans de bits restants. Chaque plan de bits est décodé en utilisant les bits de parité reçus et l'information de bord S_i correspondantes. Si la probabilité d'erreur estimée après décodage est supérieure à 10^{-3}, alors d'autres bits de parité seront demandés par l'intermédiaire du canal de rétroaction par le décodeur. Comme dans le cas du domaine pixel, le processus de décodage et de la demande de bits additionnels sont répétés jusqu'à atteindre la probabilité d'erreur désirée.

Etant donné le symbole décodé (tous les plans de bits sont décodés) et l'information de bord S_i, la reconstruction des coefficients de chaque bande i de l'image codée W est déterminée par $E[B_i|\hat{B}_{iQ}, S_i]$. Par la suite, une transformée inverse de la DCT (IDCT) est appliquée sur les coefficients reconstruits pour générer l'image décodée $\hat{W}$.

La solution de Stanford proposée dans le domaine transformé présente une complexité à l'encodeur semblable à celle d'un codeur en mode Intra traditionnel. Cependant, le décodeur vidéo du système proposé est plus complexe que celui d'un système conventionnel. En effet, le décodage itératif et l'estimation de mouvement dans la solution de Stanford engendrent une complexité plus élevée que celle des techniques conventionnelles telles que le décodage entropique.

5.4.2.3 Performances de la solution proposée

Dans [GARRM05], les performances débit-distorsion du codeur vidéo de Wyner-Ziv dans le domaine transformé ont été comparées à celles obtenues avec des codeurs vidéo traditionnels (H.263+) en modes Intra et Inter. Pour les séquences Hall Monitor et Salesman, les performances de la solution proposée restent entre celles des deux codeurs traditionnels. Cependant, un gain en performance jusqu'à 2 dB du domaine transformé est observé par rapport à la solution utilisant le domaine pixel. En effet, la transformée DCT exploite les dépendances statistiques dans une image et permet d'atteindre une meilleure performance débit-distorsion par rapport au domaine pixel. Par contre, la complexité de la solution basée sur la DCT est plus élevée que celle dans le domaine pixel.

5.5 Diverses améliorations de l'architecture avec voie de retour

5.5.1 Détermination de l'information de bord par fonction de hashage

Dans [ARG04] et pour un GOP supérieur à 2, mis à part les bits de parité de l'encodeur de Wyner-Ziv, des bits additionnels, désignés par bits du codeur "Hash", sont générés et envoyés au décodeur pour améliorer l'estimation de mouvement (avoir une meilleure information de bord) et la reconstruction des images W. En effet, pour chaque bloc 4×4 de coefficients quantifiés de l'image Wyner-Ziv W, un certain nombre de bits de "Hash" correspondants est envoyé. Les bits de "Hash" se composent d'un petit nombre de coefficients quantifiés. Leur nombre n'est pas bien indiqué dans [ARG04]. Les bits de "Hash" de l'image Wyner-Ziv courante sont stockés dans une petite mémoire. Ces bits sont employés pour décider sur l'éventualité d'envoyer les bits de "Hash" des blocs 4×4 de la prochaine image Wyner-Ziv.

En effet, pour chaque bloc 4×4 de l'image W courante, la décision d'envoyer ou pas des bits de "Hash" dépend d'une certaine distance. Cette distance est calculée entre les bits de "Hash" du bloc courant et ceux du bloc correspondant (même position) dans l'image précédente. Si la distance est inférieure à un certain seuil, alors les bits de "Hash" ne seront pas envoyés ; autrement, les bits seront transmis au décodeur. Puisque les procédures de génération et de stockage des bits de "Hash" exigent un faible calcul et une petite mémoire, les auteurs dans [ARG04] déclarent que la complexité de la nouvelle solution proposée reste similaire à celle d'un codeur en mode Intra traditionnel.

Au décodage, une extrapolation d'image en utilisant les bits de "Hash" reçus et l'image reconstruite précédemment (image Wyner-Ziv ou image Key Frames) est réalisée pour produire l'information de bord de l'image à décoder. Chaque bloc de 4×4 pixel de l'image S est déterminé selon deux cas :

- Si aucun bit de "Hash" n'est reçu, alors le bloc de 4×4 pixel prend son correspondant dans l'image précédemment reconstruite.
- Si le décodeur reçoit les bits de "Hash", alors une recherche de mouvement, basée sur ces bits dans l'image précédente, est réalisée pour déterminer le bloc de l'image

S.

La solution de [ARG04] permet d'améliorer le décodage turbo et de réduire le taux requis des bits de parité. Les bits de "Hash" sont également utilisés pour raffiner les coefficients de transformée pendant la reconstruction. Pour différentes tailles de GOP (de 4 à 16) des séquences Salesman et Hall Monitor, un gain en performance de presque 9 dB de la solution de [ARG04] par rapport à celles d'un codeur H.263+ en mode Intra peut être atteint. Par contre, les performances de [ARG04] sont à une distance de 1 à 4 dB de celles d'un codeur H.263+ en mode Inter.

5.5.2 Codage selon les fréquences des bandes DCT

Dans [AG04], une amélioration au système présenté dans [ARG04] a été proposée. Les coefficients de basse fréquence de la DCT (8×8) de l'image W sont codés avec l'encodeur de Wyner-Ziv. Par contre, pour les coefficients de haute fréquence, le codage et l'envoi de bits correspondants dépend d'une condition. En effet, les versions quantifiées des coefficients de haute fréquence de chaque image sont stockées dans une mémoire. Une distance entre les coefficients de haute fréquence quantifiés de l'image courante et leurs correspondants dans l'image précédente est calculée. Si la distance trouvée est inférieure à un seuil, alors les coefficients ne seront ni codés ni envoyés au décodeur. Autrement, si la distance est supérieure au seuil, alors les coefficients de haute fréquence quantifiés seront codés avec un codeur de Huffman et les bits résultants seront envoyés au décodeur. Ces bits sont utilisés au décodeur pour améliorer l'estimation de mouvement et la reconstruction de l'image Wyner-Ziv W. Comme dans [ARG04] avec les bits de "Hash", s'il n'y a pas des bits correspondants aux coefficients de haute fréquence, alors l'information de bord est déterminée à partir de l'image précédemment reconstruite. Dans l'autre cas, lorsque le décodeur reçoit des bits de l'encodeur de Huffman, un décodage entropique et un déquantificateur sont appliqués pour récupérer les coefficients de haute fréquence. Ces coefficients reconstruits sont utilisés par l'estimateur de mouvement pour générer la meilleure information de bord de l'image précédente. Il est clair que la complexité de la solution dans [AG04] est plus faible que celle de [ARG04]. Les performances obtenues dans [AG04] sont meilleures de 6 à 8 dB que celles obtenues avec un codeur vidéo classique en mode Intra.

5.5.3 Amélioration de l'interpolation d'images

La figure 5.7 montre le schéma d'interpolation d'images proposé dans [ABP05]. Dans ce cas, le codeur vidéo de Wyner-Ziv utilise le domaine pixel basé sur l'architecture proposée dans [AZG02], [ASG03]. Certaines modifications ont été apportées pour améliorer les performances débit-distorsion du codeur vidéo de Wyner-Ziv. Parmi les modifications proposées, on trouve l'amélioration de la génération de l'information de bord par interpolation compensée en mouvement des images Key Frames.

Le schéma proposé dans [ABP05] contient cinq modules : les filtres passe-bas, l'estimation de mouvement en avant, l'estimation de mouvement bi-directionnelle, le lissage spatial et la compensation de mouvement bi-directionnelle. Pour générer l'information

de bord S_t à l'instant t correspondant à l'image Wyner-Ziv codée W_t, les deux images Key Frames adjacentes K_{t-1} et K_{t+1} ($GOP = 2$) sont utilisées dans l'ordre par les cinq modules cités.

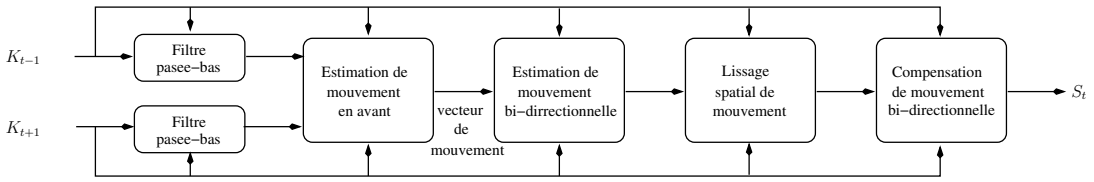

FIG. 5.7 – Schéma de l'interpolation d'images proposé dans [ABP05].

1) Estimation de mouvement en avant

Tout d'abord, les deux images Key Frames K_{t-1} et K_{t+1} sont filtrées pour améliorer la fiabilité des vecteurs de mouvement. Cette opération permet d'estimer des vecteurs de mouvement proches du champ de mouvement. Par la suite, un algorithme d'estimation de mouvement basé bloc est utilisé pour estimer le mouvement entre les images K_{t-1} et K_{t+1}. L'algorithme d'estimation de mouvement basé bloc a été choisi parmi d'autres à cause de sa faible complexité. Cet algorithme d'estimation de mouvement est caractérisé par sa taille de fenêtre, sa région de recherche et sa taille de pas. La taille de fenêtre est la dimension du bloc carré employé en tant qu'unité de base pour effectuer l'estimation de mouvement. Cette taille est fixe pour l'image entière. Le paramètre de la région de recherche est défini comme étant la dimension du secteur dans laquelle on recherche un bloc de l'image K_{t-1} semblable à son correspondant dans K_{t+1}. La taille du pas n'est que la distance entre les pixels dans l'image K_{t-1} où le vecteur de mouvement est recherché. Ce paramètre permet de réduire la complexité de l'estimation de mouvement (par augmentation de la taille).

L'algorithme d'estimation de mouvement basé bloc peut échouer à déterminer le champ de mouvement. Ainsi des régions se chevauchant peuvent apparaître dans l'image interpolée. Cela est du aux vecteurs de mouvement obtenus qui n'interceptent pas le centre de chaque bloc non chevauché de l'image interpolée. Afin de résoudre ce problème, les auteurs dans [ABP05] ont proposé que tous les vecteurs de mouvement obtenus dans les étapes précédentes servent comme des candidats potentiels pour chaque bloc non chevauché dans l'image interpolée S_t. La figure 5.8 illustre le choix du vecteur de mouvement pour un bloc donné de l'image S_t. Pour chaque bloc de l'image interpolée S_t et parmi tous les vecteurs candidats, le vecteur de mouvement choisi est celui qui intersecte au plus près le centre du bloc de l'image S_t. Par conséquent, chaque bloc de l'image interpolée S_t a un vecteur de mouvement associé.

2) Estimation de mouvement bi-directionnelle

Une fois que les vecteurs de mouvement sont déterminés, c'est l'estimation de mouvement bi-directionnelle qui est appliquée comme l'indique la figure 5.7.

L'algorithme d'estimation de mouvement bi-directionnelle, semblable à celui utilisé dans le codage d'image en mode bi-directionnel, est employé pour raffiner (améliorer) les

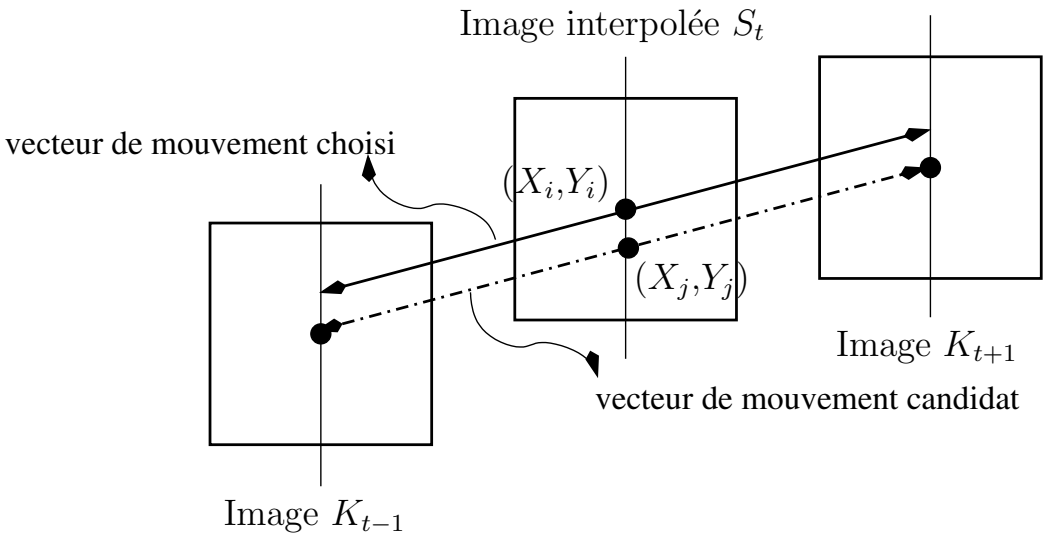

FIG. 5.8 – Sélection du vecteur de mouvement proposé dans [ABP05].

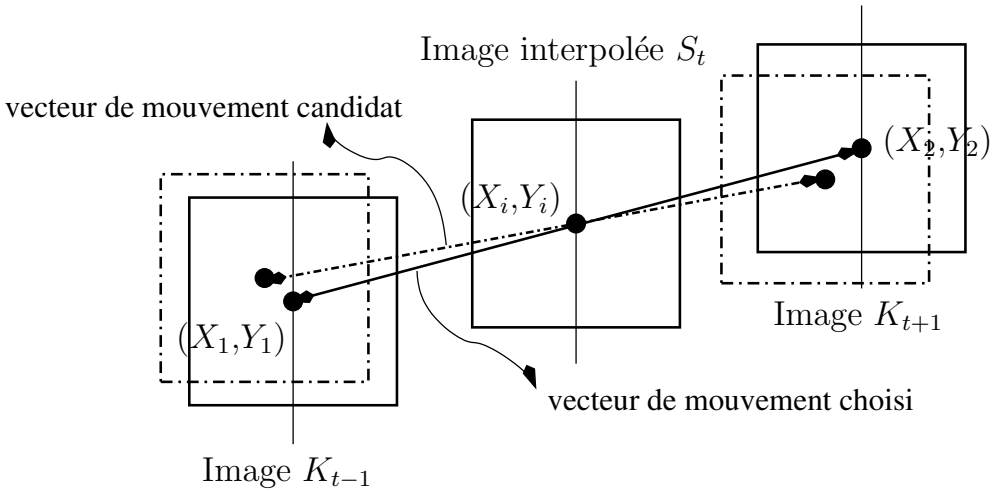

FIG. 5.9 – Estimation de mouvement bi-directionnelle proposé dans [ABP05].

vecteurs de mouvement obtenus après le module d'estimation de mouvement en avant. Cependant, puisqu'au décodeur de Wyner-Ziv les pixels interpolés ne sont pas connus, une technique d'estimation de mouvement différente est utilisée. Cette technique choisit une trajectoire linéaire entre les images Key Frames, K_{t-1} et K_{t+1}, passant par le centre des blocs de l'image interpolée S_t (figure 5.9). La région de recherche est confinée à un petit déplacement autour de la position initiale du bloc de l'image interpolée. Les vecteurs de mouvement entre l'image interpolée S_t et les images K_{t-1} et K_{t+1} sont symétriques. Par conséquent :

$$(x_1, y_1) = (x_i, y_i) + EM(B_i) \quad (5.1)$$
$$(x_2, y_2) = (x_i, y_i) - EM(B_i) \quad (5.2)$$

avec (x_1, y_1) et (x_2, y_2) respectivement les coordonnées des blocs dans les images K_{t-1} et K_{t+1}. $EM(B_i)$ représente le vecteur de mouvement obtenu à l'étape d'estimation de mouvement en avant divisée par la moitié (puisque l'image interpolée S_t est équidistante aux deux images K_{t-1} et K_{t+1}). Ainsi, les vecteurs de mouvement raffinés représentent une trajectoire de mouvement plus précise. A la fin de l'estimation de mouvement bi-directionnelle, les erreurs introduites lors du choix initial des vecteurs de mouvement sont corrigées.

3) Lissage spatial de mouvement

Les vecteurs de mouvement obtenus après l'estimation de mouvement bi-directionnelle peuvent parfois présenter une faible cohérence spatiale. Pour remédier à ce problème, des algorithmes de lissage spatial permettant la réduction du nombre de faux vecteurs de mouvement (vecteurs de mouvement incorrects une fois comparés avec les vrais champs de mouvement) ont été utilisés.

L'algorithme considéré dans [ABP05] utilise le principe WVMF (en anglais Weighted Vector Median Filters) proposé dans [ABBC96]. Le filtrage WVMF est une technique utilisée pour enlever les bruits dans les images multi-composantes. Le WVMF maintient la cohérence spatiale du champ de mouvement en cherchant pour chaque bloc les vecteurs de mouvement candidats parmi les blocs voisins. L'algorithme de lissage spatial du mouvement proposé est efficace à la limite (extrémité) d'image où les changements brusques de la direction des vecteurs de mouvement se produisent, même pour des régions homogènes (mouvement semblable).

4) Compensation de mouvement bi-directionnelle

Une fois que le champ final de vecteurs de mouvement est obtenu, l'image interpolée S_t peut être déterminée par une compensation de mouvement bi-directionnelle comme dans le codage vidéo standard.

Dans le domaine pixel, les performances obtenues en terme de $PSNR$ avec un codeur vidéo distribué utilisant la technique d'estimation de mouvement proposée dans [ABP05], sont meilleures que celles obtenues avec un système utilisant seulement les deux premiers modules à savoir l'estimation de mouvement en avant et l'estimation de mouvement bi-directionnelle.

5.5.4 Taille de GOP variable

Dans le codage vidéo traditionnel, le mécanisme pour commander la taille de GOP existe déjà. Les algorithmes utilisés pour réaliser cette tache se servent de l'information de mouvement (vecteurs de mouvement ou image résiduelle) qui est disponible à l'encodeur. Cependant, dans un codeur vidéo de Wyner-Ziv, il est difficile d'appliquer une telle solution parce que l'encodeur ne dispose pas de l'information de mouvement.

Un mécanisme pour ajuster la taille du GOP dans un codeur vidéo de Wyner-Ziv a été ajouté dans [ABP06]. Pour classifier chaque image comme Wyner-Ziv W ou Key Frames K, l'ajustement est réalisé à l'encodeur en se basant sur l'image courante et l'image précédente, en exploitant la redondance et en mesurant l'activité le long de la séquence vidéo. La technique présentée dans [ABP06] n'augmente pas la complexité de l'encodeur. Un gain de performances en terme de $PSNR$ de 0.2 à 0.8 dB peut être atteint avec un codeur vidéo de Wyner-Ziv utilisant le mécanisme qui ajuste la taille du GOP.

5.5.5 Estimation du modèle de corrélation entre l'information de bord et l'image à décoder

Dans les systèmes de codage de sources distribuées, le décodeur (décodeur turbo) a besoin de connaître la corrélation (dépendance statistique) entre l'information de bord et les données originales codées. Comme déjà mentionné ci-dessus, la plupart des travaux de codage vidéo distribué, par exemple dans le domaine pixel, considère un modèle de corrélation entre les pixels de l'images Wyner-Ziv W (désignés par $pixel_W$) et leurs correspondants des images S (notés par $pixel_S$ pixels de l'information de bord) basé sur une distribution Laplacienne définie par :

$$f(pixel_W - pixel_S) = \frac{\alpha}{2} exp(-\alpha|pixel_W - pixel_S|) \tag{5.3}$$

avec $\alpha^2 = \frac{2}{\sigma^2}$ un paramètre de la distribution Laplacienne tel que σ^2 représente la variance entre les pixels de l'images Wyner-Ziv W et leurs correspondants des images S. Dans [BAP06b], une solution a été proposée pour estimer en ligne ("ONLINE") la variance σ^2. Avant cette solution, la variance σ^2 est calculée hors-ligne ("OFFLINE") en utilisant toutes les images W et S.

L'approche de la solution proposée dans [BAP06b] est d'estimer la variance σ^2 à partir des images Key Frames K adjacentes à W. Exemple, pour un GOP=2 et à l'instant t, l'image S_t (information de bord à W_t) est obtenue par interpolation compensée en mouvement à partir des images K_{t-1} et K_{t+1} :

$$S_t(x,y) = \frac{1}{2}(K_{t-1}(x + dx_{t-1}, y + dy_{t-1}) + K_{t+1}(x + dx_{t+1}, y + dy_{t+1})) \tag{5.4}$$

avec $K_{t-1}(x+dx_{t-1}, y+dy_{t-1})$ et $K_{t+1}(x+dx_{t+1}, y+dy_{t+1}))$ représentant respectivement les images compensées en mouvement aux instants $(t-1)$ et $(t+1)$. (x,y) correspond aux coordonnées du pixel dans l'image interpolée S_t. (dx_{t-1}, dy_{t-1}) et (dx_{t+1}, dy_{t+1}) représentent respectivement les vecteurs de mouvement pour K_{t-1} et K_{t+1}.

Quand le résidu, qui est égal à la différence entre les images compensées en mouvement aux instants $(t-1)$ et $(t+1)$ ($K_{t-1}(x+dx_{t-1}, y+dy_{t-1})$ et $K_{t+1}(x+dx_{t+1}, y+dy_{t+1})$), est élevé, alors la corrélation entre les pixels des images Wyner-Ziv et interpolée, W_t et S_t, est faible. Par conséquent, la variance σ^2 a une grande valeur. Par contre, pour un faible résidu, la corrélation entre W_t et S_t est élevée (les valeurs de la variance σ^2 sont faibles). Par conséquent, les auteurs dans [BAP06b] ont déduit que la variance σ^2 calculée en ligne ("ONLINE") peut être déterminée comme suit :

$$\sigma^2 = (\frac{1}{2})^2 \frac{\sum_{(x,y)\in S_t}(K_{t-1}(x+dx_{t-1}, y+dy_{t-1}) - K_{t+1}(x+dx_{t+1}, y+dy_{t+1}))^2}{L} \quad (5.5)$$

avec L la taille de l'image.

Pour un GOP=2, les performances du codeur vidéo de Wyner-Ziv utilisant la solution d'estimation de σ^2 proposée dans [BAP06b] sont proches de celles obtenues avec un système de codage vidéo distribué connaissant les valeurs réelles de la corrélation entre W_t et S_t. Cependant, pour des GOP plus élevés (4 et 8 dans [BAP06b]), il y a des pertes de performances avec la nouvelle solution qui sont dues à la non connaissance de l'image originale au décodeur. Néanmoins, la méthode d'estimation de la corrélation entre W_t et S_t proposée dans [BAP06b] correspond à un scénario réel du codage vidéo distribué.

Dans [BAP06a], les auteurs proposent une estimation des valeurs de α (ou bien σ^2) par pixel et par bloc de pixels entre les images W_t et S_t. L'estimation de α par pixel permet de conduire à de meilleures performances.

5.6 Conclusion

Nous avons présenté les solutions de codage vidéo distribué existantes. Les architectures proposées sont classées en deux types : sans voie de retour ou avec voie de retour. Les solutions avec voie de retour permettent de bien contrôler le débit de transmission et d'avoir un décodeur simple. Cependant, ces solutions introduisent une latence et une utilisation de la bande passante plus élevées. Par conséquent, ces solutions ne sont pas appropriées aux applications telles que les systèmes vidéo d'acquisition et de stockage où les images compressées sont transférées et décodées à un temps postérieur.

En contrepartie de la non utilisation de la voie de retour dans un codeur vidéo PRISM, un décodage multi-hypothèse qui entraîne l'utilisation du décodage multiple est réalisé. En plus, l'envoi de l'information supplémentaire sous forme des bits de CRC introduit une augmentation du débit de transmission.

Les performances débit-distorsion des codeurs vidéo distribués sont situées entre celles du codeur classique (exemple H.263+) avec un codage en modes Inter et Intra. Toutefois, la complexité du codage de ces trois solutions reste comparable à celle d'un codeur classique en mode Intra. Cependant, vu d'une part la présence de la classification et d'autre part la grande complexité du décodeur dans l'architecture de codage vidéo distribué sans voie de retour, nous avons privilégié l'approche avec voie de retour dans nos travaux.

Chapitre 6

Algorithme de compression vidéo distribuée

6.1 Introduction

Nous avons présenté dans le chapitre précédent les différents travaux et approches dans le domaine du codage vidéo distribué. Les solutions proposées présentent des performances débit-distorsion légèrement supérieures au codage vidéo en mode Intra, mais celles-ci restent très inférieures aux performances du codage prédictif (mode Inter). Dans ce chapitre, nous cherchons à développer un ensemble d'outils algorithmiques, afin d'améliorer les performances débit-distorsion de l'architecture avec voie de retour pour se rapprocher de celles du codage prédictif. Le premier problème qui pénalise les performances débit-distorsion du codeur vidéo distribué est la fréquence des images clés qui sont codées en mode Intra. Pour réduire cette fréquence et améliorer les performances, la taille du *GOP* peut être augmentée. Or, une telle augmentation nécessite de développer des outils d'interpolation compensée en mouvement performants. Avec l'algorithme d'estimation de mouvement basé bloc et pour une taille du *GOP* égale à 3, une amélioration de la qualité de l'information de bord en utilisant le principe du codage de trois sources distribuées est décrite. Vu la sous-optimalité du quantificateur scalaire uniforme, des améliorations des performances débit-distorsion du codeur vidéo de Wyner-Ziv par l'utilisation de la quantification TCQ sont ensuite proposées.

Dans un troisième temps, nous nous sommes intéressés au contrôle de débit dans un codeur vidéo distribué. Le contrôle de débit dans les solutions avec voie de retour s'effectue au décodage en se basant sur le taux d'erreur vrai, c'est-à-dire en utilisant les plans de bits originaux. Pour lever cette limitation architecturale, nous introduisons un mécanisme de contrôle de débit des bits de parité basé sur une mesure de confiance calculée à la sortie du décodeur turbo.

Un mécanisme hybride de contrôle de débit au codage et au décodage basé sur une estimation du débit minimal théorique pour une corrélation donnée a été proposé. Celui-ci permet de réduire l'utilisation de la voie de retour et par conséquent de diminuer la latence du décodeur.

Enfin, l'effet d'un bruit de canal sur les performances débit-distorsion d'un codeur vidéo distribué a été étudié. Dans ce cas, seuls les bits de parité issus du codeur vidéo Wyner-Ziv ont été affectés par le bruit de canal CBS.

6.2 Schéma de codage vidéo distribué considéré

Un codeur vidéo distribué (Distributed Video Coding) est un système basé sur un codeur de Wyner-Ziv, où les images d'une séquence vidéo donnée sont codées indépendamment (codées en mode Intra) mais décodées conjointement (décodées en mode Inter).

Les images d'une séquence vidéo sont divisées en deux sous-ensembles comme l'indique la figure 6.1. Un premier sous-ensemble d'images, régulièrement espacées dans la séquence, va être codé en mode Intra (appelées images clés "Key Frames" et désignées par K). Ces images sont codées avec un codeur H.264 dont le paramètre de quantification, désigné par QP, contrôle le pas de quantificateur scalaire uniforme utilisé. Le deuxième sous-ensemble d'images est codé en utilisant le codeur de Wyner-Ziv. Ces images sont désignées par W (images Wyner-Ziv). Une séquence vidéo est divisée en GOP contenant un certain nombre d'images. La première image de chaque GOP est codée en mode Intra et les autres sont codées avec un codeur de Wyner-Ziv.

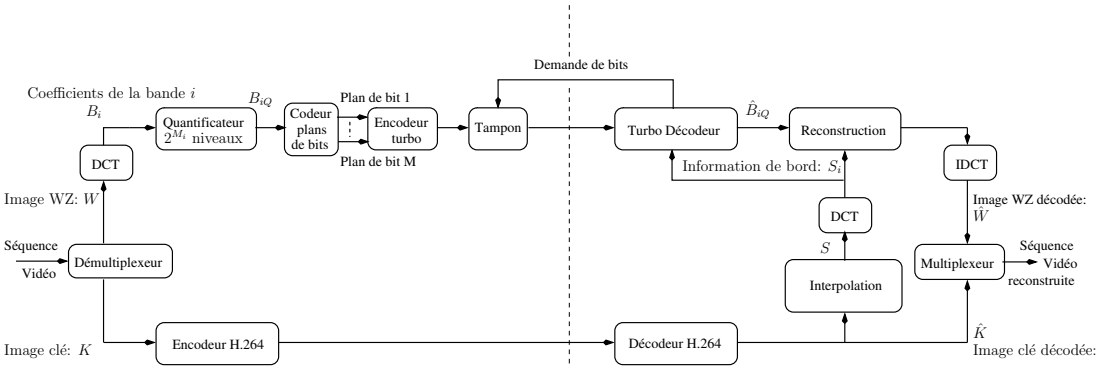

FIG. 6.1 – Schéma de codage/décodage du codeur vidéo distribué dans le domaine transformé.

Comme il a été mentionné dans le chapitre 5, le codage par transformée est un outil qui peut être utilisé dans le codage vidéo distribué pour exploiter la corrélation spatiale dans une image. Ainsi, l'utilisation du domaine transformé permet au codeur vidéo distribué d'atteindre de meilleures performances débit-distorsion.

Le schéma du codeur vidéo distribué utilisé dans ce chapitre est illustré sur la figure 6.1. Ce schéma est semblable à celui présenté dans [ARSG04]. Cependant, les modules tels que le quantificateur et la technique d'interpolation d'image utilisés ici sont ceux de [BAP06c]. Dans toutes les simulations réalisées dans ce chapitre, nous ne nous intéressons qu'aux blocs de luminance (Y) des images W.

Pour chaque bloc de 4×4 pixels de l'image Wyner-Ziv W, une transformée DCT est appliquée. Les 16 pixels corrélés à l'intérieur du bloc 4×4 sont convertis en 16 coefficients DCT. Ces coefficients DCT sont arrangés dans un bloc 4×4 appelé bloc de coefficients DCT. La figure 6.2 illustre un bloc de coefficients DCT de taille 4×4. Le premier coefficient DCT s'appelle le coefficient DC et correspond à la fréquence spatiale la plus faible. Les 15 coefficients restants sont connus comme coefficients AC. Le coefficient AC situé à la position 16 correspond en général à la fréquence spatiale la plus élevée.

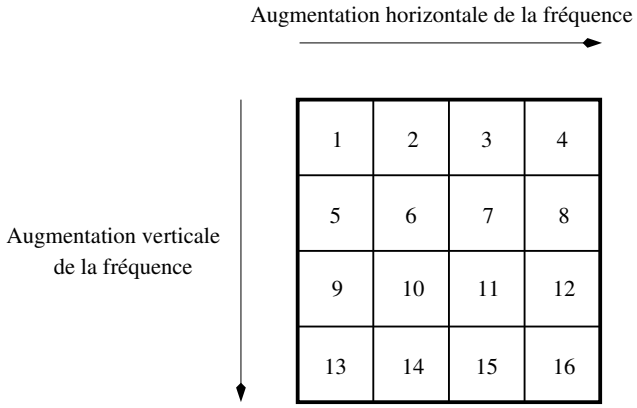

FIG. 6.2 – L'ordre des positions dans un bloc de coefficients DCT de taille 4×4.

Les coefficients DCT sont groupés pour former 16 bandes de coefficients selon leurs positions occupées dans les blocs 4×4 (figure 6.2).

Après l'opération de transformation, les coefficients de chaque bande i ($i = 1, 2, ..., 16$) sont indépendamment quantifiés avec un quantificateur scalaire uniforme à 2^{M_i} niveaux. Les valeurs de M_i correspondent au nombre de bits nécessaires pour représenter un coefficient DCT donné. Différentes valeurs de niveaux de quantification sont utilisées pour atteindre différentes performances débit-distorsion du codeur vidéo distribué. La figure 6.3 montre huit matrices 4×4 dont les éléments i correspondent aux différentes valeurs de niveaux de quantification utilisées pour chaque bande i. Chaque matrice est indexée par un coefficient appelé Qindexe (Qindexe $\in \{1, 2, ..., 8\}$). Ainsi, pour une séquence vidéo, un GOP=2 et une valeur de QP donnée du codeur H.264, les images Wyner-Ziv W sont quantifiées avec un quantificateur dont le Qindexe associé permet d'avoir les mêmes valeurs de distorsion des images paires et impaires. La valeur 0 dans ces matrices signifie que les bandes correspondantes ne sont ni quantifiées ni transmises au décodeur. Pour ces bandes et lors de la reconstruction, le décodeur utilise les coefficients DCT de l'information de bord.

Le quantificateur scalaire uniforme utilisé n'est pas le même pour chaque bande i. En effet, pour la bande 1 les coefficients DC sont caractérisés par des valeurs positives. Par conséquent, les intervalles de quantification correspondant ont seulement des valeurs positives. La figure 6.4 illustre le quantificateur scalaire uniforme utilisé pour la bande

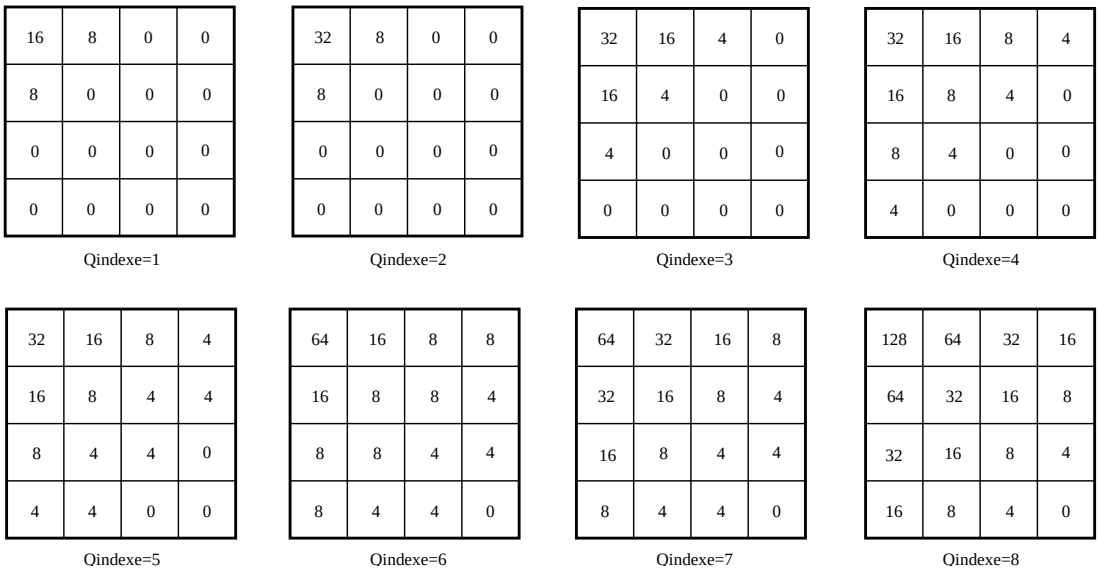

FIG. 6.3 – Les 8 matrices 4×4 indiquant les différentes valeurs de niveaux de quantification utilisées.

1, avec $0, 1, 2, ...$ indiquant les indices de quantification et T_1 représentant la largeur de l'intervalle de quantification ou bien le pas de quantification. Ce pas dépend du nombre de niveaux de quantification et de la valeur maximale des coefficients DC. Cette valeur maximale $V_{1\max}$ est déterminée dans [HPN97] par :

$$V_{1\max} = \sqrt{n^2} I_{\max} \tag{6.1}$$

avec n^2 le nombre de pixels dans un bloc $n \times n$ et $I_{\max}$ l'intensité maximale du pixel. Ainsi, pour un bloc de pixels de taille 4×4 et un signal représenté sur 8 bits, la valeur maximale $V_{1\max}$ est égale à 1024. Ainsi, le pas de quantification T_1 s'exprime par :

$$T_1 = \frac{V_{1\max}}{2^{M_i}} \tag{6.2}$$

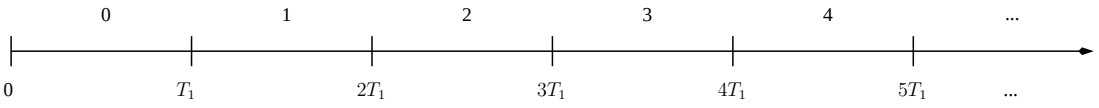

FIG. 6.4 – Quantificateur scalaire uniforme utilisé pour la bande 1.

Quant aux autres bandes ($i \neq 1$), les coefficients AC peuvent avoir des valeurs négatives ou positives. Par conséquent, un quantificateur, dont l'intervalle central autour du zéro, est utilisé comme l'indique la figure 6.5. Le pas de quantification T_i pour chaque

bande i ($i \neq 1$) est déterminé selon le nombre de niveaux de quantification et la valeur absolue maximale $|V_{i\max}|$ des coefficients AC par :

$$T_i = \frac{2|V_{i\max}|}{2^{M_i} - 1}, \quad i \neq 1 \tag{6.3}$$

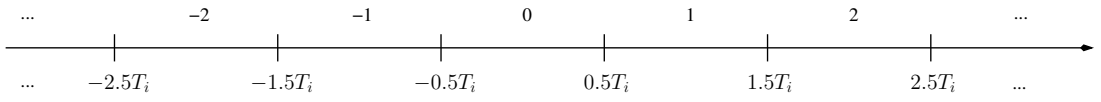

FIG. 6.5 – Quantificateur scalaire uniforme utilisé pour les bandes i avec $i \neq 1$.

Après la quantification et comme dans la section 5.4.2.1, les symboles quantifiés B_{iQ} sont convertis en plans de bits. Les bits de même poids (regroupés dans un même plan de bits) sont codés avec un encodeur turbo poinçonné dont les deux codeurs CRS sont de taux de codage 1/2, de longueur de contrainte $K = 5$ et de polynôme générateur ($h_0 = 23, h_1 = 33$). L'entrelaceur utilisé est de type aléatoire et de taille 1584 ($176 \times 144/16$). Pour chaque plan de bits, seuls les bits de parité retenus sont transmis au décodeur. La matrice de poinçonnage utilisée est créée d'une manière aléatoire. La période de poinçonnage est égale à 48.

Au décodeur, un algorithme de décodage MAP est utilisé pour estimer les plans de bits en utilisant les bits de parité transmis et les coefficients DCT de l'image interpolée S. Par la suite, les symboles quantifiés des coefficients DCT de l'image W sont déterminés.

Après le décodage turbo, les coefficients DCT reconstruits de W peuvent avoir trois valeurs possibles :

- une valeur égale au coefficients DCT de l'information de bord S, si le coefficient DCT de S et le symbole décodé sont dans la même région de quantification (fixée par le symbole décodé).
- la valeur de l'extrémité supérieure de la région de quantification, si le coefficient DCT de S est supérieur à cette extrémité supérieure.
- la valeur de l'extrémité inférieure de la région de quantification, si le coefficient DCT de S est inférieur à cette extrémité inférieure.

6.3 Codage vidéo distribué utilisant deux informations de bord

Dans cette section, nous nous intéressons à une proposition de mise en œuvre du codage de trois sources distribuées à une séquence vidéo. Le but est d'améliorer la qualité de l'information de bord d'un codeur vidéo distribué pour des GOP de taille supérieure à 2. Dans cette section, nous allons utiliser un $GOP = 3$.

Le processus de décodage classique avec un $GOP = 3$ et à l'instant t d'un codeur vidéo distribué s'effectue de la façon suivante :

1. une image S_t est créée à partir d'une interpolation compensée en mouvement appliquée sur les deux images Key Frames K_{t-1} et K_{t+2}. Par la suite, S_t est utilisée pour décoder l'image Wyner-Ziv $\hat{W}_t$.
2. une fois que la reconstruction de W_t ($\hat{W}_t$) est réalisée, l'image S_{t+1} est générée après interpolation (compensée en mouvement) de deux images $\hat{W}_t$ et K_{t+2} pour estimer l'image Wyner-Ziv $\hat{W}_{t+1}$.

Cependant, lors de l'étape de la génération de S_t, une autre image S^*_{t+1} peut être créée par une interpolation compensée en mouvement appliquée sur les deux images Key Frames K_{t-1} et K_{t+2} comme l'indique la figure 6.6. Cette nouvelle image peut être considérée comme une autre information de bord pour décoder l'image Wyner-Ziv $\hat{W}_{t+1}$.

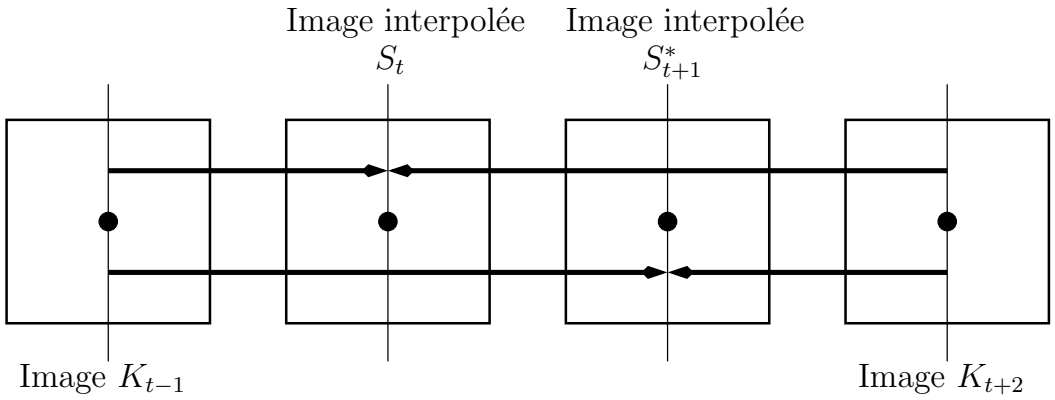

FIG. 6.6 – Génération de deux informations de bord à partir de deux images Key Frames pour un $GOP = 3$.

La reconstruction de l'image W_t se réalise avec la connaissance des coefficients DCT de l'image S_t. Par contre, pour décoder les plans de bits de l'image W_{t+1}, nous utilisons les coefficients DCT de deux images S_{t+1} et S^*_{t+1}. En effet, une moyenne des deux coefficients DCT de S_{t+1} et S^*_{t+1} est prise comme information de bord pour estimer le coefficient correspondant dans l'image W_{t+1}. Le modèle de corrélation considéré entre les coefficients moyennés et leurs correspondants de l'image W_{t+1} est une distribution Laplacienne. Si on désigne les coefficients DCT de W_{t+1}, S_{t+1} et S^*_{t+1} par des variables aléatoires respectives X, Y et Z, alors notre modèle de corrélation sera $X - \frac{Y+Z}{2}$. Ceci ressemble au modèle $X - (Y + Z)$ considéré dans le chapitre 4 où l'énergie de $(Y + Z)$ est égale à l'unité.

Les résultats de simulation illustrés à la figure 6.7 sont obtenus avec un $GOP = 3$ du codage distribué de la séquence vidéo Foreman à 30 Hz et une variance de la distribution Laplacienne calculée hors-ligne ("OFFLINE"). Deux codeurs CVD sont utilisés. Un premier codeur, désigné par CVD-2S, utilise seulement une information de bord lors de décodage. Quant au deuxième codeur CVD, désigné par CVD-3S, il utilise deux informations de bord pour décoder la deuxième image Wyner-Ziv d'un GOP donné. Des

gains de performances en terme de $PSNR$ de l'ordre de 0.23 dB du codeur CVD-3S par rapport à CVD-2S peuvent être remarqués pour des débits faibles de l'image W. Cependant, des dégradations de performances du codeur CVD-3S par rapport à celles de CVD-2S sont observées pour des débits élevés de l'image W. Cela confirme nos résultats théoriques du chapitre 4.

Nous avons aussi remarqué que plus la qualité des images K codées en mode Intra est faible (des valeurs de QP élevées), moins le gain en performances du codeur CVD-3S par rapport à CVD-2S est important.

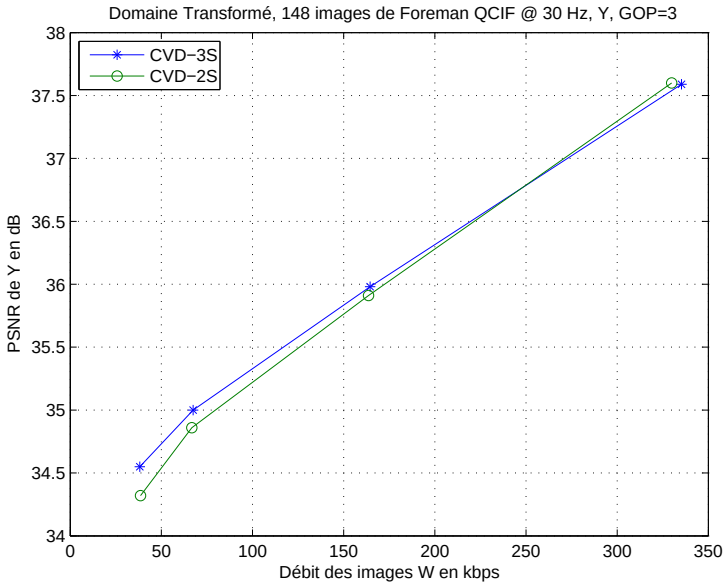

FIG. 6.7 – Performances du codeur vidéo distribué utilisant 1 ou 2 informations de bord avec une séquence Foreman QCIF, 30 Hz, un $GOP = 3$ et un $QP = 06$.

Afin d'améliorer les performances du codeur CVD-3S, il serait intéressant de chercher un autre modèle de corrélation entre les coefficients DCT de W_{t+1} d'une part et ceux de S_{t+1} et de S^*_{t+1} d'autre part. En effet, il est plus approprié de trouver une loi de probabilité adéquate modélisant la corrélation entre les coefficients DCT de trois images W_{t+1}, S_{t+1} et S^*_{t+1}.

Dans la suite du chapitre, une seule information de bord est utilisée pour la reconstruction des images Wyner-Ziv.

6.4 Codeur vidéo distribué utilisant la TCQ et le code turbo

6.4.1 Approche

Dans cette partie, nous présentons une architecture de codage vidéo distribué utilisant la quantification TCQ pour les coefficients de bandes 1 (composantes DC) à la place d'un quantificateur scalaire uniforme comme l'indique la figure 6.8. Vu la sous-optimalité du quantificateur scalaire uniforme, l'objectif est d'améliorer les performances débit-distorsion du système CVD.

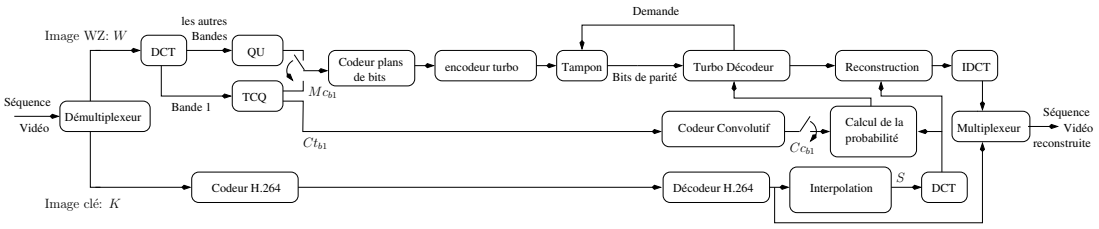

FIG. 6.8 – Schéma d'un codeur vidéo distribué utilisant la TCQ et le code turbo poinçonné.

Après l'application de la transformée DCT sur les images Wyner-Ziv W, un quantificateur TCQ est utilisé seulement pour les coefficients de bandes 1 pour préserver une complexité faible à l'encodage. La séquence du chemin du treillis Ct_{b1} est envoyée directement au décodeur vidéo. Alors que la séquence de mots de code Mc_{b1} est codée par plans de bits. Quant aux coefficients des autres bandes (autres que un), le quantificateur scalaire uniforme (QU) est utilisé. Les coefficients quantifiés sont codés par plans de bits. Chaque plan de bits est ensuite compressé en utilisant un codeur turbo poinçonné et seulement les bits de parité retenus sont envoyés au décodeur.

Les images K sont décodées avec un décodeur H.264. Ainsi, pour chaque image Wyner-Ziv, une information de bord S est générée par interpolation, compensée en mouvement à partir des images K. Comme pour les images W au codage, une transformée DCT est appliquée aux images S et des bandes de coefficients sont formées. Le modèle de corrélation considéré entre les coefficients DCT des images W et leurs correspondants des images S (l'information de bord) est une distribution Laplacienne.

Pour chaque bande AC des images W, le décodeur turbo utilise les coefficients DCT de l'information de bord S et les bits de parité reçus pour estimer les plans de bits. Si le nombre d'erreurs estimé est supérieur à 10^{-3}, alors une demande de bits de parité additionnels est envoyée vers le tampon du codeur turbo par la voie de retour. Le processus est répété jusqu'à ce qu'une probabilité d'erreur binaire acceptable (10^{-3}) soit atteinte.

Pour les bandes 1, le même processus sera répété pour les plans de bits sauf qu'avant le décodage turbo, la séquence Cc_{b1} doit être récupérée par codage convolutif de la séquence Ct_{b1}. Puis, les probabilités se basant sur des distributions Laplaciennes à

l'entrée du turbo décodeur sont calculées seulement dans les régions de quantification où leurs indices contiennent les symboles de Cc_{b1}.

Après le décodage turbo, les indices des coefficients DCT quantifiés sont estimés. Chaque bande de l'image W est reconstruite. Pour ne pas introduire une complexité supplémentaire au décodage, la méthode de reconstruction de la section 3.3.2.2 n'est pas appliquée aux coefficients des bandes DC. Ici, l'algorithme de reconstruction des coefficients DCT de toutes les bandes est similaire à celui décrit dans la section 6.2. Si le coefficient DCT de l'information de bord et l'indice décodé sont dans la même région de quantification (fixée par l'indice décodé) alors le coefficient DCT reconstruit prendra une valeur égale à celle de l'information de bord. Cependant, si le coefficient DCT de l'information de bord est inférieur à l'extrémité inférieure de la région de quantification, alors le coefficient DCT reconstruit prendra la valeur de cette extrémité. Par contre, si le coefficient DCT de l'information de bord est supérieur à l'extrémité supérieure de la région de quantification, alors la valeur reconstruite prendra la valeur de cette extrémité supérieure. Enfin, une transformée inverse de la DCT (IDCT) sera appliquée sur les coefficients reconstruits.

6.4.2 Résultats de simulation

Tous les résultats de simulation présentés sont réalisés avec un codeur convolutif de la quantification TCQ de longueur de contrainte $K_{TCQ} = 9$. La séquence vidéo à coder est Foreman en format QCIF et de fréquence d'image égale à 15 Hz.

Nous comparons les performance débit-distorsion de deux codeurs vidéo distribués. Le premier codeur (désigné par CVD-1) utilise différents quantificateurs scalaires uniformes pour chaque bande de la DCT. Quant au deuxième codeur (Désigné par CVD-2), les bandes 1 sont quantifiées avec la TCQ et pour les autres bandes on utilise différents quantificateurs scalaires uniformes. Les résultats de simulation obtenus avec la TCQ sont réalisés avec différents débits : 3-bits TCQ avec Qindexe=1 (voir figure 6.3), 4-bits TCQ avec Qindexe=3, 5-bits TCQ avec Qindexe=6, et 6-bits TCQ avec Qindexe=8. Les variances du bruit de corrélation entre les coefficients DCT de W et de S sont calculées hors-ligne (OFFLINE).

Pour un GOP=2 (c'est-à-dire entre deux images Intra, il y a une seule image W), la figure 6.9 illustre le gain de performance obtenu avec la quantification TCQ au niveau de la bande 1 par rapport à la quantification uniforme. A un débit (débit global des images W et K) de 500 kbps, on peut voir que le $PSNR$ de Y (des images W et K) obtenu avec le CVD-2 utilisant la TCQ est meilleur de 0.29 dB que celui obtenu avec un CVD-1 à base d'un quantificateur scalaire uniforme. Ce gain de performance augmente en fonction de la taille du GOP et du débit. Sur la figure 6.10 pour un GOP=4 (entre deux images Intra, il y a 3 images Wyner-Ziv), la différence de $PSNR$ entre les deux approches passe de 0.39 dB à 0.435 dB pour des débits respectifs de 500 à 600 kbps. La figure 6.11 illustre un gain obtenu avec le CVD-2 de l'ordre de 0.568 dB par rapport à CVD-1 pour un GOP=8 et un débit de 700 kbps.

Le CVD-2 est pénalisé par les performances de la TCQ à faible débit. On observe que plus le débit est petit, moins le gain de performances du CVD-2 par rapport à

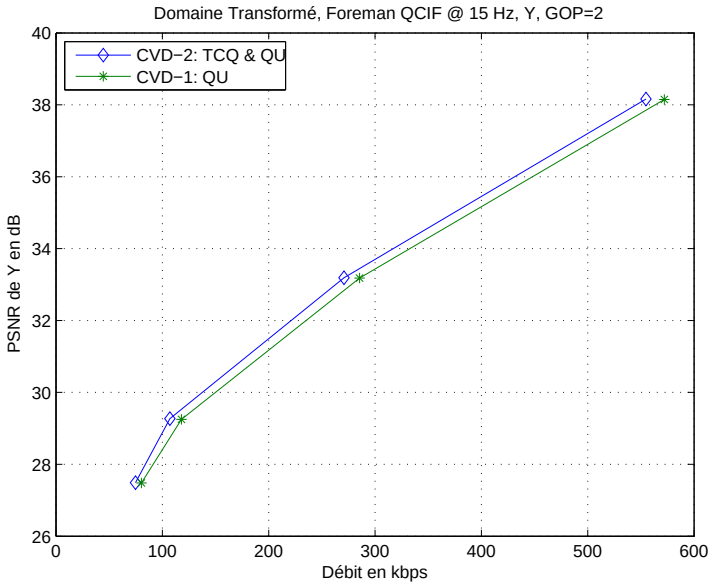

FIG. 6.9 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$.

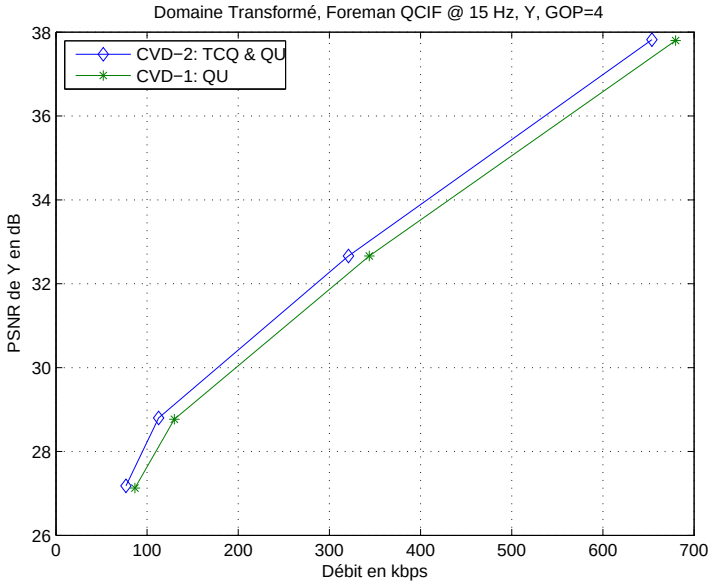

FIG. 6.10 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 4$.

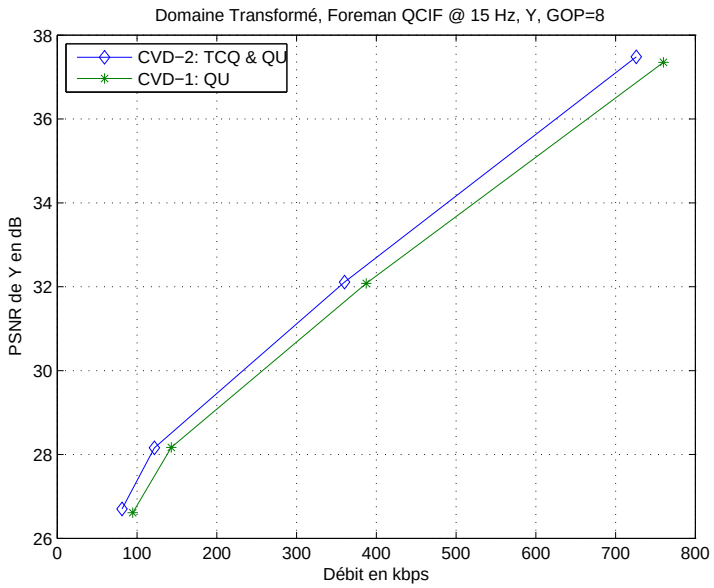

FIG. 6.11 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 8$.

CVD-1 est important.

Afin d'avoir une idée sur la complexité engendrée par l'utilisation de la quantification TCQ dans un codeur vidéo distribué, le tableau 6.1 montre pour un $PSNR = 27.48$ dB, les temps de codage (désigné par t_e) et de décodage (désigné par t_d) de la séquence Foreman (149 images) obtenus avec les systèmes CVD-1 et CVD-2. Les temps de codage et décodage sont déterminés en utilisant un PC équipé d'un pentium 4 avec une fréquence de 3000 Mhz. On observe une légère augmentation de la complexité au codage du codeur CVD-2 par rapport à CVD-1 (t_e=31 secondes dans CVD-2 au lieu de 20 sec dans CVD-1). Néanmoins, la complexité du codeur CVD-2 reste quand même très faible par rapport à celle d'un codeur classique à savoir H.264 en mode Intra (t_e=39 sec) ou Inter (t_e=496 sec pour IPB avec un $GOP = 16$).

Dans le tableau 6.1, on remarque que la complexité du décodeur de CVD-2 diminue de près de 40 % par rapport à celle de CVD-1. Cela est du à la connaissance parfaite de la séquence Cc_{b1} au décodeur. En effet, les bits de la séquence Cc_{b1} correspondent aux plans de bits avec des poids les plus faibles. Ces plans de bits sont faiblement corrélés avec ceux de l'information de bord. Si ces plans de bits sont issus d'un quantificateur scalaire uniforme, alors le décodeur va utiliser plus de bits de parités pour les décoder et par conséquent, le temps de décodage va croître.

Pour mieux illustrer l'effet de la TCQ sur les bandes 1, la figure 6.12 montre les résultats de simulation des deux codeurs CVD codant et décodant seulement les bandes 1 (les autres bandes ne sont ni codées ni décodées). Pour un GOP=4, le gain de performance du CVD-2 par rapport à CVD-1 est de 0.717 dB pour un débit égal à 100

TAB. 6.1 – *Temps de codage et décodage obtenus avec les systèmes CVD-1 et CVD2 avec un GOP=2 (figure 6.9).*

$PSNR$ en dB	CVD-1		CVD-2	
	t_e en sec	t_d en min	t_e en sec	t_d en min
27.48	20	97	31	59

kbps.

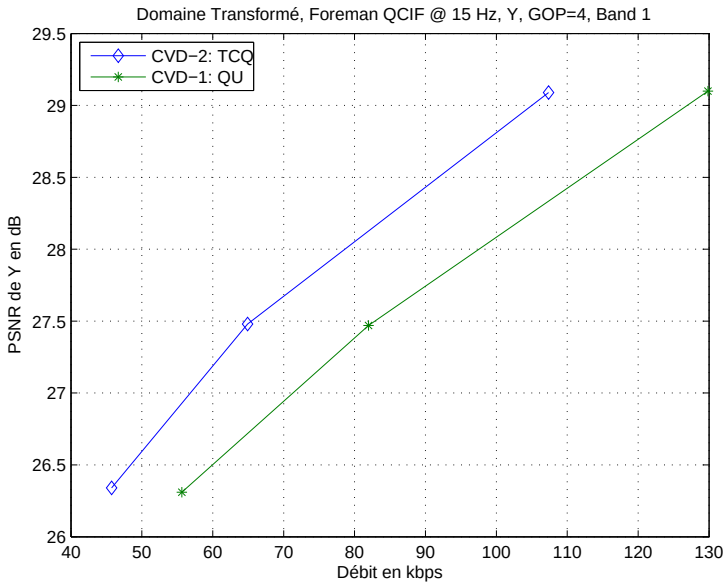

FIG. 6.12 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 4$: seule la bande 1 est codée et décodée.

Dans la suite du chapitre, à l'exception de la section 6.6, seul le quantificateur scalaire uniforme est utilisé.

6.5 Contrôle de débit en codage vidéo distribué

6.5.1 Contrôle de débit au décodeur

Les techniques de contrôle de débit au décodeur sont basées sur un canal de retour. La demande des bits de parité additionnels dépend du taux d'erreur estimé à la fin du décodage turbo. En effet, le taux d'erreur est comparé à un seuil (souvent fixé à 10^{-3}). Si le taux d'erreur est plus élevé que ce seuil, alors le décodeur va demander plus de bits de parité.

En fait, le codeur turbo présenté sur la figure 3.1 génère deux séquences de bits de parité. Une première séquence est fournie par le codeur CRS 1 et correspond au décodeur SISO 1. Alors que la deuxième séquence est générée par le codeur CRS 2 pour le processus de décodage au niveau du décodeur SISO 2.

Dans la plupart des schémas de codage vidéo distribué, au début, seulement deux bits de parité par période de poinçonnage (1 bit de chaque codeur CRS) sont envoyés au décodeur. Par la suite, à chaque fois que la probabilité d'erreur binaire obtenue après un décodage turbo est supérieure à 10^{-3}, une demande d'un seul bit de parité additionnel par période de poinçonnage est effectuée par les décodeurs SISO d'une façon alternée. En effet, lors de la première demande, le bit de parité additionnel transmis par période de poinçonnage provient du codeur CRS 1. Pour une deuxième demande, le bit de parité est fourni par le codeur CRS 2 et correspond au décodeur SISO 2. Ce processus est répété d'une façon alternée jusqu'à avoir une probabilité d'erreur inférieure à 10^{-3}.

6.5.1.1 Mesure de confiance en codage vidéo distribué

A ce jour, dans la plupart des travaux réalisés dans le codage vidéo distribué, le taux d'erreur estimé à la fin du décodage turbo est calculé en mesurant la distance de Hamming entre le plan de bit de l'image décodée et celui de l'image originale. Par conséquent, cette mesure calculée d'une manière parfaite correspond bien au vrai taux d'erreur binaire. Cependant, l'utilisation de l'image originale au décodeur n'est pas réaliste. Ainsi, une solution plus adéquate est proposée dans cette partie.

La méthode alternative pour estimer le taux d'erreur du plan de bit est d'utiliser la mesure de confiance qui permet de donner une mesure de fiabilité des performances du décodage turbo. Les mesures de confiance les plus généralement utilisées sont basées sur les probabilités a posteriori obtenues à la sortie du décodeur turbo. Nous considérons la mesure de confiance comme critère de choix pour déterminer si le décodeur turbo a besoin de plus de bits de parité ou pas. Dans ce cas-ci, la mesure de confiance est basée sur le rapport de vraisemblance défini par :

$$\Lambda_i = \log \frac{Pr(X_i = 1/S_i)}{Pr(X_i = 0/S_i)} \tag{6.4}$$

avec X_i le plan de bit à décoder à l'instant i et S_i est l'information de bord.

Pour chaque instant i, si la valeur absolue du rapport de vraisemblance ($|\Lambda_i|$) est inférieure à un seuil de confiance (fixé à 4.6 dans les résultats de simulation), alors le bit X_i sera considéré incorrectement décodé. Le seuil de confiance 4.6 correspond à la valeur $\log 99$. C'est-à-dire, si la probabilité $Pr(X_i = 1/S_i)$ ou $Pr(X_i = 0/S_i)$ est inférieure à 0.99, alors l'incertitude sur la valeur du bit X_i serait encore grande. Dans ce cas, un compteur *count* qui mesure le nombre d'erreur est incrémenté de 1. A la fin du décodage, la probabilité d'erreur binaire (désignée par BER_{MC}) est estimée par :

$$BER_{MC} = \frac{count}{N} \tag{6.5}$$

avec N la longueur de la séquence de plan de bits. Si la probabilité d'erreur BER_{MC} est supérieure au seuil 10^{-3}, des bits de parité additionnels seront demandés par le décodeur turbo comme l'indique la figure 6.13.

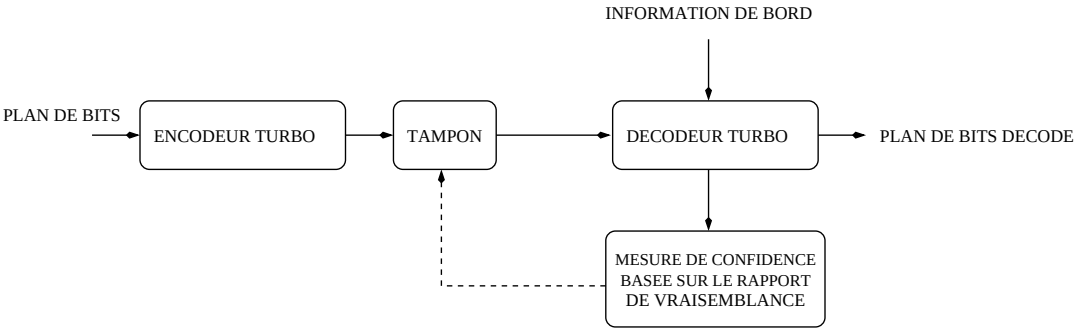

FIG. 6.13 – Codeur/décodeur turbo utilisant la mesure de confiance dans un schéma de codage vidéo distribué.

A chaque demande du décodeur, le nombre de bits de parité additionnels envoyé par le codeur est d'un bit par période de poinçonnage. Le taux d'erreur déterminé à partir de la mesure de confiance est considéré comme une approximation de la valeur réelle de ce dernier. Ainsi l'impact de cette approximation sur les performances débit-distorsion du codeur vidéo distribué a été évalué pour plusieurs séquences.

Les figures 6.14, 6.15, 6.16 et 6.17 montrent les résultats de simulation du codeur vidéo distribué dont le taux d'erreur est déterminé à partir, d'une part, de la mesure de confiance (le codeur est désigné par CVD-MC) et, d'autre part, du calcul basé sur l'image originale (le codeur est désigné par CVD-OR). Au décodeur turbo, les variances du bruit de corrélation entre les coefficients DCT de W et de S sont calculées hors-ligne. Pour un GOP=2, et à l'exception du point correspondant à débit élevé (350 kbps) de la séquence Hall Monitor pour lequel il y a une perte en $PSNR$ de 0.26 dB, les performances débit-distorsion obtenues avec un taux d'erreur estimé à partir de la mesure de confiance sont comparables à celles obtenues avec le vrai taux d'erreur. La conclusion qui peut être tirée est que la méthode basée sur la mesure de confiance est fiable pour le contrôle de débit dans un schéma de codage vidéo distribué puisque aucune perte majeure en terme de débit-distorsion n'est remarquée.

Une autre méthode d'estimation de la probabilité de bits erronés (désignée par P_b) à la sortie d'un décodeur turbo a été présentée dans [HLS00]. Cette probabilité est définie par :

$$P_b = E[\frac{1}{1+e^{|\Lambda|}}] = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{1+e^{|\Lambda_i|}} \tag{6.6}$$

La méthode de [HLS00] peut être utilisée dans le contexte du codage vidéo distribué pour estimer si le décodeur turbo a besoin de plus de bits de parité ou pas. Cependant, nous avons constaté que cette méthode (de [HLS00]) n'est pas fiable. En effet, les résultats de simulation obtenus avec le codage vidéo distribué d'une séquence Hall Monitor

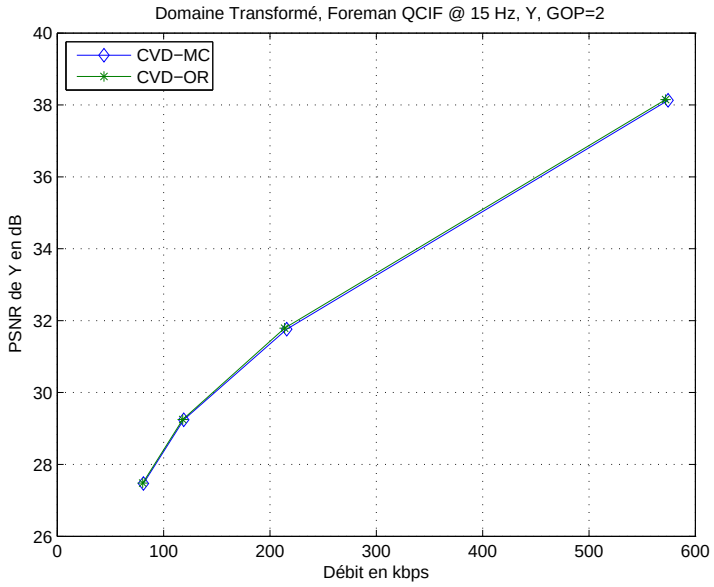

FIG. 6.14 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$. Dans CVD-MC, le taux d'erreur est estimé par la mesure de confiance. CVD-OR utilise le vrai taux d'erreur.

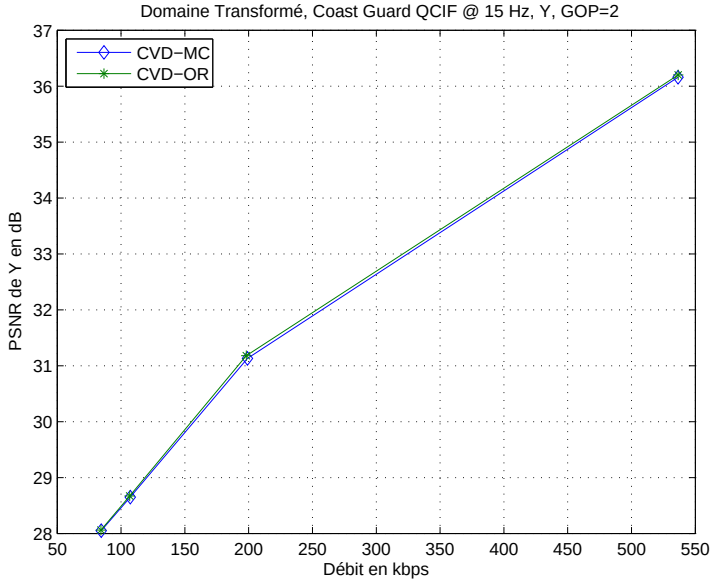

FIG. 6.15 – Résultats de simulation pour CVD de la séquence Coast Guard QCIF, 15 Hz avec un $GOP = 2$. Dans CVD-MC, le taux d'erreur est estimé par la mesure de confiance. CVD-OR utilise le vrai taux d'erreur.

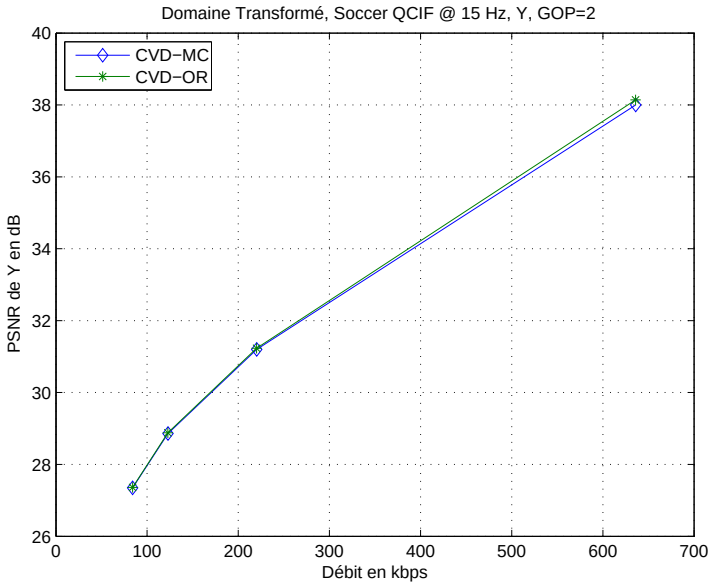

FIG. 6.16 – Résultats de simulation pour CVD de la séquence Soccer QCIF, 15 Hz avec un $GOP = 2$. Dans CVD-MC, le taux d'erreur est estimé par la mesure de confiance. CVD-OR utilise le vrai taux d'erreur.

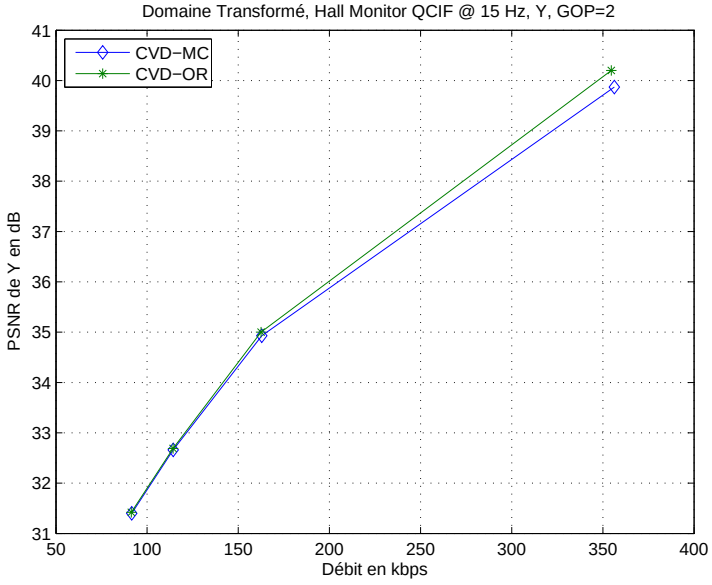

FIG. 6.17 – Résultats de simulation pour CVD de la séquence Hall Monitor QCIF, 15 Hz avec un $GOP = 2$. Dans CVD-MC, le taux d'erreur est estimé par la mesure de confiance. CVD-OR utilise le vrai taux d'erreur.

utilisant la méthode de [HLS00] présentent des pertes de performances de l'ordre de 2.71 dB par rapport à celles obtenues avec un CVD-OR pour un débit de 350 kbps.

Dans la suite de cette section, pour tous les résultats de simulation du codeur vidéo distribué, le taux d'erreur est estimé par la mesure de confiance.

6.5.1.2 Demande de bits multiples

Le but de demander un seul bit de parité additionnel par période de poinçonnage est donc d'utiliser le moins de bits possible pour avoir un débit le plus faible (granularité de débit la plus fine possible) avec une même qualité de reconstruction (même valeur de distorsion). Pour chaque demande d'un bit additionnel par période de poinçonnage, tout le processus de décodage turbo est répété. Toutefois, une telle solution peut mener à un nombre élevé de demandes de bits de parité selon la corrélation entre l'image Wyner-Ziv et l'information de bord. En effet, une séquence vidéo formée par des images très actives (des scènes avec beaucoup de mouvement) demande un débit global élevé pour assurer une qualité de reconstruction acceptable.

Par conséquent, une grande complexité et un délai élevé du décodeur vidéo distribué peuvent avoir lieu suite au nombre élevé de bits de parité additionnels demandés. Ceci va entraîner un autre inconvénient du contrôle de débit au décodage qui est le nombre élevé de l'utilisation de la voie de retour.

Afin de réduire la complexité et le délai du décodeur vidéo distribué, une autre solution a été implémentée. En effet, lorsque le taux d'erreur binaire à la sortie du décodeur turbo est supérieur à 10^{-3}, une demande de deux bits de parité additionnels par période de poinçonnage (un bit pour chaque décodeur SISO) est effectuée. Cette solution permet de diminuer la complexité totale du décodage turbo par un facteur égal à 2. Cependant, ceci entraîne un surcoût de débit global égal à ΔR qui s'exprime en kbps par :

$$\Delta R = \frac{\#planBits \times Eb \times Freq \times \#Period \times FractWZParGop}{1000} \tag{6.7}$$

avec $\#planBits$ le nombre total de plans de bits des bandes DCT quantifiées, Eb la moyenne du nombre prévu de bits de parité supplémentaires envoyés par période de poinçonnage (pour 2 bits par période, il y a deux possibilités équiprobables 0 ou 1, donc $Eb = 0.5(0+1)$), $Freq$ la fréquence de la séquence, $\#Period$ le nombre de périodes de poinçonnage et $FractWZParGop$ la fraction des images Wyner-Ziv W par GOP.

Par exemple, pour une séquence QCIF à 15 Hz avec un total de plans de bits $\#planBits = 10$ (Qindexe=1), une période de poinçonnage de taille 48 ($\#Period = \frac{1584}{48} = 33$) et un GOP=2 (1 image W par GOP, $FractWZParGop = 0.5$), ΔR est égal à 1.23 kbps.

L'impact sur le débit global du mécanisme basé sur la demande de deux bits de parité par période de poinçonnage a été évalué expérimentalement. On peut voir sur les figures 6.18, 6.19, 6.20 et 6.21 que la demande de 2 bits de parité au lieu de 1 par période de poinçonnage engendre une perte négligeable en terme de débit-distorsion.

Pour la figure 6.18, la perte en débit pour le même $PSNR$ est de 0.82 kbps (près de la valeur estimée $\Delta R = 1.23$ kbps) avec Qindexe=1 (qui correspond au $PSNR = 27.47$ dB), et autour de 7.30 kbps (la perte ΔR estimée est de 7.79 kbps) avec un Qindexe=8 (qui correspond au $PSNR = 38.15$ dB).

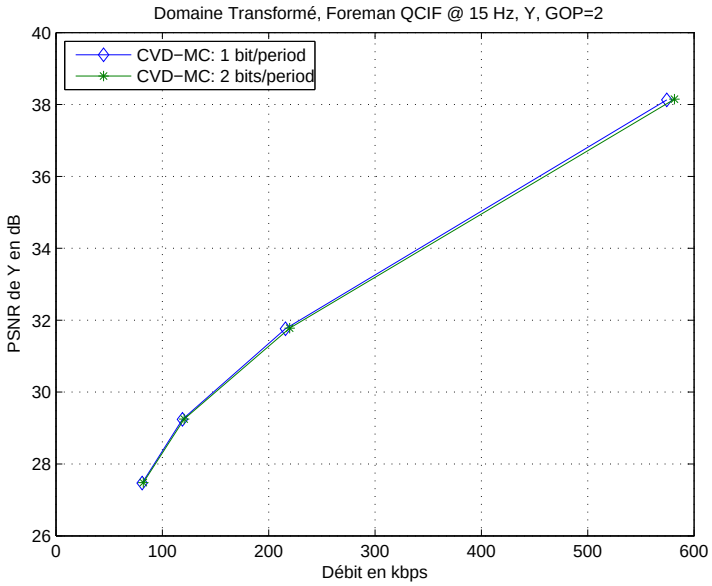

FIG. 6.18 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance

Pour diminuer davantage le délai et la complexité du décodeur vidéo distribué, une demande de 3, 4 ou plus de bits de parité supplémentaires par période de poinçonnage peut être envisagée. Cependant, faire cette demande d'une manière constante (c'est-à-dire demander le même nombre de bits de parité additionnels pour tous les plans de bits de chaque bande i) entraîne une grande perte en débit du codeur vidéo distribué. Pour la séquence Foreman, les figures 6.22 et 6.23 montrent la dégradation des performances en terme de débit après des demandes respectives de 3 et de 4 bits par rapport à celles obtenues avec une demande de 1 bit par période. Pour le même $PSNR$ (autour de 38.15 dB) et un Qindexe=8, on peut observer dans les figures 6.22 et 6.23 des pertes de débits de 13.38 et de 19.47 kbps pour des demandes respectives de 3 et de 4 bits par période de poinçonnage. Ces valeurs sont proches des pertes estimées par l'équation (6.7). En effet, pour un Qindexe=8 ($\#planBits = 63$ dans (6.7)), les pertes ΔR sont égales à 15.59 ($Eb = (0 + 1 + 2)/3$ dans (6.7)) et 23.38 ($Eb = (0 + 1 + 2 + 3)/4$) kbps pour des demandes respectives de 3 et de 4 bits par période de poinçonnage.

Le tableau 6.2 montre les temps globaux du décodage de la séquence Foreman avec différentes valeurs du nombre de bits de parité demandés par période de poinçonnage. Les surcoûts en débit d'un CVD avec des demandes de 2, 3 ou 4 bits de parité par rapport

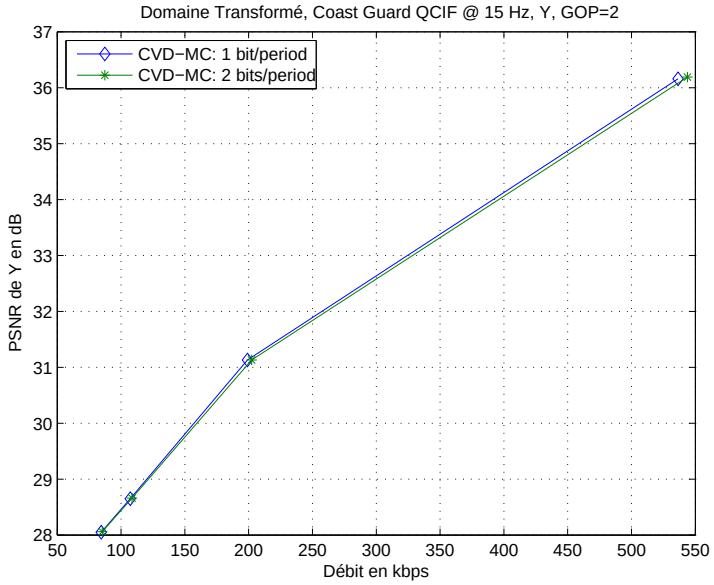

FIG. 6.19 – Résultats de simulation pour CVD de la séquence Coast Guard QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance.

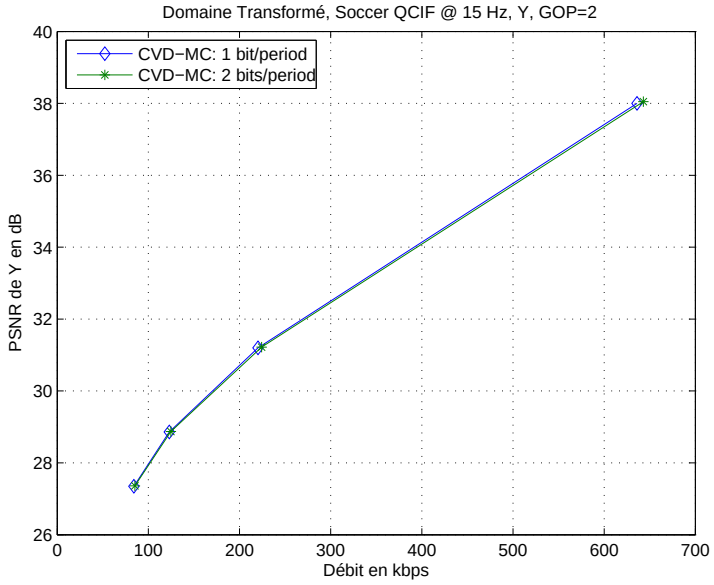

FIG. 6.20 – Résultats de simulation pour CVD de la séquence Soccer QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance.

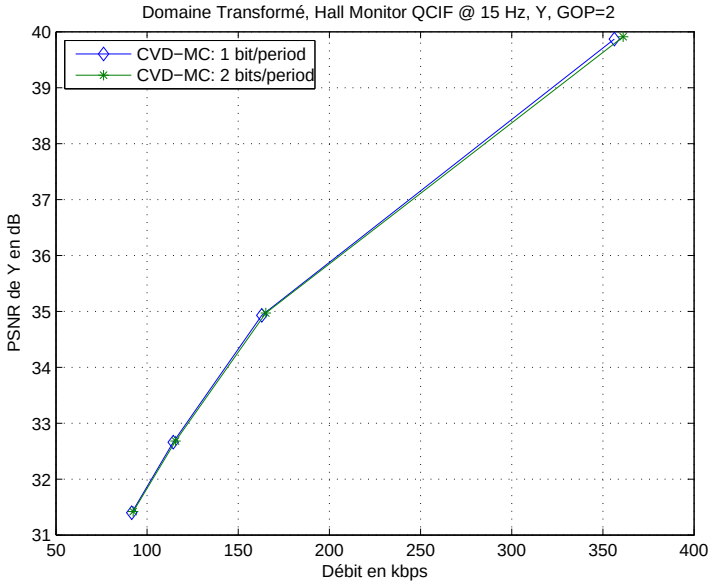

FIG. 6.21 – Résultats de simulation pour CVD de la séquence Hall Monitor QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance.

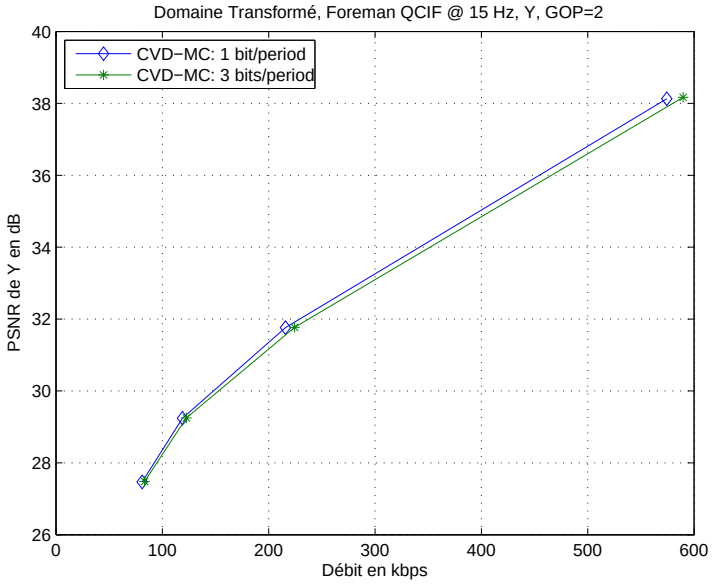

FIG. 6.22 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 3 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance.

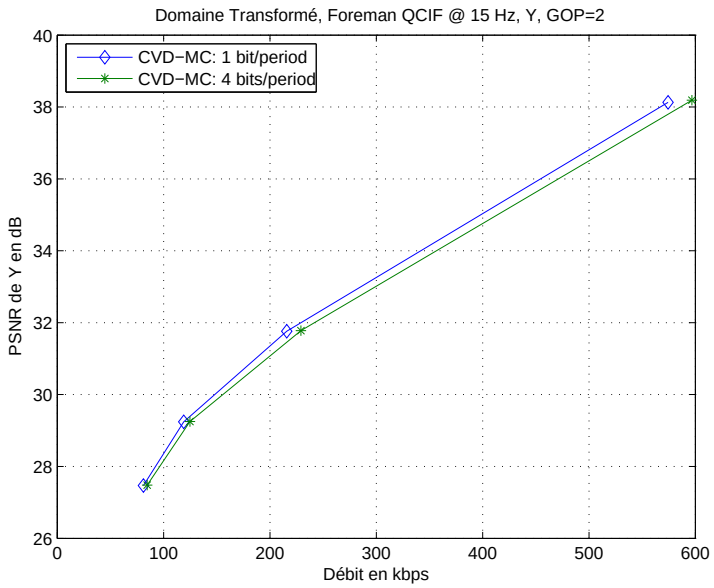

FIG. 6.23 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$ et 1 bit/période de poinçonnage ou 4 bits/période de poinçonnage. Le taux d'erreur est estimé par la mesure de confiance.

à celles d'un CVD avec une demande de 1 bit de parité par période de poinçonnage sont illustrées. Il est clair que pour une demande avec un grand nombre de bits de parité additionnels, le temps de décodage (désigné par t_d) diminue mais la perte en débit (désignée par p_d), pour un même $PSNR$, augmente.

TAB. 6.2 – *Comparaison des performances des différentes valeurs du nombre de bits de parité demandés par période de poinçonnage pour un codeur vidéo distribué utilisant la mesure de confiance.*

$PSNR$ en dB	1 bit/période		2 bits/période		3 bits/période		4 bits/période	
	t_d en min	p_d en kbps	t_d en min	p_d en kbps	t_d en min	p_d en kbps	t_d en min	p_d en kbps
31.76	236	0	143	3.65	108	8.52	92	12.77
38.15	720	0	407	7.30	295	13.38	237	19.47

6.5.2 Contrôle de débit hybride encodeur/décodeur

Un algorithme d'allocation de débit dans un codeur vidéo distribué doit préserver une complexité faible à l'encodeur et permettre une adaptation fine du débit pour tous les plans de bits de chaque bande DCT.

Classiquement, une fois que la fonction débit-distorsion $D(R)$ est déterminée, le problème d'allocation de débit se résout en utilisant une technique basée sur les multiplicateurs de Lagrange, avec R le débit et D la distorsion associée. Cela revient à minimiser un critère du type $D(R) + \lambda R$, où λ est un coefficient réel positif. Dans ce cas, une recherche sur λ est effectuée jusqu'à atteindre le débit ou la distorsion désirés. Notre méthode n'est pas directement basée sur les multiplicateurs de Lagrange et évite ainsi la recherche sur les valeurs de λ. Nous mettons en œuvre un algorithme adapté au codage vidéo distribué.

Cette section présente l'algorithme de contrôle de débit où le débit de chaque plan de bits est variable selon la bande à coder et la variance du bruit de corrélation entre les images Wyner-Ziv W et l'information de bord. Le but de l'algorithme est d'assurer une bonne qualité de reconstruction au décodeur tout en optimisant l'allocation du débit en fonction du besoin de la scène. L'idée est de trouver une approche basée sur le théorème de Wyner-Ziv pour effectuer une répartition optimale du débit entre les plans de bits de chaque bande. Nous allons proposer une solution pour, d'une part, fixer la qualité de reconstruction désirée et, d'autre part, présenter l'allocation de débit pour tous les plans de bits de chaque bande DCT des images Wyner-Ziv W.

6.5.2.1 Qualité de reconstruction désirée

Dans tous les travaux réalisés dans le codage vidéo distribué, les niveaux de quantification des images Wyner-Ziv W sont fixés de telle sorte que leurs distorsions moyennes soient égales à celles des images Key Frames K d'une séquence donnée. Le but dans ce cas est d'avoir une qualité de reconstruction homogène entre les deux types d'images (W et K) de la séquence vidéo. Ainsi, l'oeil humain ne peut pas percevoir ou détecter la différence entre les deux types d'images. Cependant, pour satisfaire cette contrainte, une recherche hors-ligne est réalisée pour déterminer le meilleur quantificateur des images W en fonction des valeurs du pas de quantification $Qstep_K$ (indexé par le paramètre de quantification QP du codeur H.264) des images K codées en mode Intra avec H.264. En effet, pour une valeur QP donnée, les bandes i de l'image W de chaque séquence sont d'abord quantifiées avec tous les nombres de niveaux de quantifications 2^{M_i} possibles ($2^{M_i} \in \{0, 2, 4, 8, 16, 32, 64, 128, 256\}$ avec $2^{M_i} = 0$ signifie que les bandes correspondantes ne sont ni quantifiées ni transmises au décodeur). Ensuite, les symboles quantifiés sont codés en plans de bits puis compressés avec un codeur turbo. Au décodage, les coefficients DCT de l'information de bord sont générés par application de la transformée DCT sur les images S interpolées par compensation de mouvement. Par la suite, après le décodage turbo et la reconstruction des bandes DCT de l'image W correspondantes à chaque niveau de quantification, toutes les distorsions globales associées D^W de la séquence vidéo sont déterminées. Toutefois, la présence de la transformée DCT dans notre système de codage vidéo distribué permet d'avoir une propriété d'orthogonalité

entre les bandes DCT et ainsi de supposer la somme des distorsions D_i^W de toutes les bandes i égale à la distorsion globale $D^W = \sum_{i=1}^{16} D_i^W$ (pour une transformée DCT 4×4) d'une image W donnée. Cela revient à dire que la contribution de chaque bande au débit et à la distorsion est indépendante.

La distorsion D_i^W d'une bande i correspondant à un niveau de quantification donné est mémorisée lors du processus de décodage. La distorsion globale D^W qui est égale à celle des images Key Frames K (D^K) permet de trouver la meilleure valeur du niveau de quantification de chaque bande i.

Pour un même quantificateur utilisé à chaque bande i des images W, la procédure qu'on vient de décrire peut être appliquée avec l'utilisation seulement du premier GOP plus l'image K codée en Intra du GOP suivant. Il est clair que cette procédure n'est pas adaptée pour des applications pratiques. Même le fait d'utiliser le même quantificateur, cela demande un grand délai de codage et de décodage. Alors la question qui se pose est : quelle est la solution la plus adaptée pour un codeur vidéo distribué pour choisir le meilleur niveau de quantification par bande i des images Wyner-Ziv W ?

Pour choisir le meilleur niveau de quantification, la procédure décrite ci-dessus est basée sur le codage et le décodage de toutes les bandes i d'une image W. Le choix final dépend essentiellement de trois événements :

- la distorsion globale des images Key Frames K désignée par D^K.
- la connaissance des valeurs possibles du niveau de quantification.
- la valeur de la distorsion D_i^W par bande i dépendante du niveau de quantification.

Le premier paramètre qui est aussi le $PSNR$ des images K codées avec un codeur H.264 dépend du choix du pas de quantification $Qstep_K$. Au total, 52 valeurs de $Qstep_K$ sont acceptées par le codeur H.264. Le pas $Qstep_K$ est indexé par le paramètre de quantification QP (voir tableau 6.3). $Qstep_K$ double sa valeur pour chaque augmentation de 6 de la valeur de QP.

TAB. 6.3 – *Les valeurs du pas de quantification dans un codeur H.264*

QP	0	1	2	3	4	5	6	7	8
$Qstep_K$	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625

Les six premières valeurs (de QP=0 à 5) du tableau 6.3 sont connues par notre codeur vidéo distribué. Une fois que la valeur de QP est fixée, le pas de quantification $Qstep_K$ peut être déterminé d'une manière itérative suivant le tableau 6.3. Après chaque codage d'une image K avec le codeur H.264, la distorsion $D^K(QP)$ en fonction du QP peut être connue. Même la distorsion par bande DCT $D_i^K(QP)$ de l'image K peut aussi être déterminée. L'idée est de baser le choix du niveau de quantification des bandes DCT de l'image W sur les distorsions $D_i^K(QP)$. Le nombre de niveaux de quantifications possibles des bandes i de l'image W est $2^{M_i} \in \{0, 2, 4, 8, 16, 32, 64, 128, 256\}$. Nous cherchons à trouver le meilleur niveau de quantification tel que la distorsion par bande DCT D_i^W de l'image W soit égale à $D_i^K(QP)$.

Comme point de départ et pour chaque bande i de l'image W, nous allons chercher le niveau de quantification le plus proche de la fraction suivante :

$$2^{M_i} \approxeq \frac{|MaxDCT_i - MinDCT_i|}{Qstep_K} \tag{6.8}$$

avec $MaxDCT_i$ et $MinDCT_i$ respectivement les valeurs maximale et minimale des coefficients DCT d'une bande i. Exemple, si $\frac{|MaxDCT_i - MinDCT_i|}{Qstep_K}$ est entre 16 et 32 alors le niveau de quantification sera une de ces deux valeurs. Le but dans ce cas est de diminuer le temps de recherche.

Une fois que la valeur $\frac{|MaxDCT_i - MinDCT_i|}{Qstep_K}$ est connue, la distorsion D_i^W par bande i entre les coefficients DCT de l'image originale W et leurs versions reconstruites (inconnus au codeur) doit être estimée au codage pour savoir quelle valeur du niveau de quantification prendre. Les coefficients DCT reconstruits sont inconnus lors du processus du codage. Ils sont déterminés seulement après le décodage turbo et la phase de reconstruction suivant la méthode présentée à la section 6.2.

Par conséquent, la distorsion D_i^W dépend de la région de quantification et du bruit de corrélation entre les images W et S. Mathématiquement et pour un niveau de quantification donné, si on connaît la variance par bande σ_i^2 du bruit de corrélation (distribution Laplacienne) entre les coefficients DCT de W et ceux de S alors la distorsion D_i^W pourrait être estimée par :

$$D_i^W = \int_{-\infty}^{+\infty} (x - f(n))^2 p(n) dn \tag{6.9}$$

avec x le coefficient DCT de W connu au codage, $p(n)$ la densité de probabilité du bruit de corrélation qui est une distribution Laplacienne de variance σ_i^2 et $f(n)$ la fonction définie par :

$$f(n) = \begin{cases} LB & \text{si } n < LB - x \\ x + n & \text{si } LB - x < n < UB - x \\ UB & \text{si } n > UB - x \end{cases} \tag{6.10}$$

avec LB et UB les deux extrémités respectives inférieure et supérieure de la région de quantification contenant le coefficient x. Il est clair que la fonction $f(n)$ représente la version reconstruite des coefficients DCT de W estimée au codage. Par conséquent, la distorsion par bande D_i^W peut être exprimée comme suit :

$$\begin{aligned} D_i^W &= \int_{-\infty}^{+\infty} (x - f(n))^2 p(n) dn \\ &= \int_{-\infty}^{LB-x} (x - LB)^2 p(n) dn + \int_{LB-x}^{UB-x} n^2 p(n) dn + \int_{UB-x}^{+\infty} (x - UB)^2 p(n) dn \end{aligned} \tag{6.11}$$

La variance par bande i, σ_i^2, du bruit de corrélation peut être déterminée au décodage ou bien estimée au codeur.

Si nous utilisons la variance calculée au décodage (appelée variance "ONLINE"), alors il est clair qu'il faut :

1. d'abord, coder les images K aux instants $(t-1)$ et $(t+1)$ avec le codeur H.264,
2. déterminer suivant la méthode de [BAP06b] vue au chapitre 5, le résidu qui est égal à la différence entre les images compensées en mouvement aux instants $(t-1)$ et $(t+1)$.
3. appliquer une transformée DCT sur le résidu.
4. estimer la variance σ_i^2 par :

$$\sigma_i^2 = E[R_i^2] \tag{6.12}$$

avec R_i les coefficient DCT du résidu.

Pour toutes les images W, nous utilisons ici le même quantificateur à chaque bande i. Par conséquent, les variances calculées au décodage sont déterminées après le codage et le décodage de seulement deux images K.

Au codeur, la méthode qui ne demande pas beaucoup de temps de calcul ni une grande complexité est celle qui détermine la variance σ_i^2 (appelée variance par moyenne) de la façon suivante :

1. Calculer la moyenne entre les pixels des images K aux instants $(t-1)$ et $(t+1)$ pour générer une image M. Soient P_{t-1}^K et P_{t+1}^K les pixels respectivement des images K_{t-1} et K_{t+1}. Les pixels de l'image M, P_{t-1}^M, sont égales à $P_{t-1}^M = \frac{1}{2}(P_{t-1}^K + P_{t+1}^K)$.
2. appliquer une transformée DCT sur l'image M.
3. estimer la variance σ_i^2 par :

$$\sigma_i^2 = \frac{1}{L}\sum_{j=1}^{L}(Coef_i^M - Coef_i^W)^2 \tag{6.13}$$

avec L la taille de chaque bande i (égale à 1584 pour une transformée de 4×4 d'une séquence QCIF) et $Coef_i^M$ et $Coef_i^W$ les coefficients DCT de la bande i respectivement des images M et W. La technique décrite ici pour estimer la variance σ_i^2 est désignée dans la suite par interpolation moyennée.

Connaissant les valeurs de variance σ_i^2, les distorsions par bande i sont déterminées suivant les valeurs du niveau de quantification. Au maximum, nous utilisons seulement les deux niveaux qui sont les plus proches de $\frac{|MaxDCT_i - MinDCT_i|}{Qstep_K}$. Par conséquent, nous estimons deux valeurs de distorsion par bande i. Le choix du niveau de quantification par bande i se fait de telle sorte que la distorsion D_i^W soit égale ou proche de $D_i^K(QP)$. Si les deux valeurs de distorsion ne correspondent pas à celle de $D_i^K(QP)$, alors d'autres niveaux de quantification autour de $\frac{|MaxDCT_i - MinDCT_i|}{Qstep_K}$ seront utilisés.

6.5.2.2 Allocation de débit

Comme nous l'avons évoqué précédemment, le contrôle de débit au décodage utilise beaucoup la voie de retour. Ici, nous présentons un algorithme d'allocation de débit au

codage qui génère un flux minimal et variable du nombre de bits de parité à transmettre au décodeur selon les plans de bits de chaque bande i à coder.

D'après le théorème de Wyner-Ziv [WZ76], le débit minimal R_{min} nécessaire pour compresser une source Gaussienne X sachant que la source Y (Gaussienne) est connue seulement au décodeur est :

$$R_{min} = \frac{1}{2}\log_2 \frac{\sigma^2}{D_X} \tag{6.14}$$

avec σ^2 la variance du bruit de corrélation entre X et Y et D_X (la distorsion) l'erreur moyenne quadratique entre X et sa valeur reconstruite $\hat{X}$.

Or, dans un schéma de codage vidéo distribué, on connaît au codeur :

1. la variance par bande σ_i^2 du bruit de corrélation entre les coefficients DCT de l'image Wyner-Ziv et leur correspondants de l'information de bord S. Cette variance est envoyée par le décodeur ou bien estimée par interpolation moyenne.
2. la distorsion estimée D_i^W (suivant l'équation (6.11)) par bande i entre les coefficients DCT de l'image originale W et leurs versions reconstruites (inconnus au codeur).

Par conséquent, le débit minimal R_{min}^i par bande i peut être estimé comme suit :

$$R_{min}^i = \frac{1}{2}\log_2 \frac{\sigma_i^2}{D_i^W} \tag{6.15}$$

Alors, comment peut-on répartir le débit minimal R_{min}^i sur les plans de bits de chaque bande i ? Il faut rappeler que les variances du bruit de corrélation sont inconnues par plan de bits.

Pour résoudre ce problème, prenons l'exemple de la figure 6.24 qui indique les huit régions de quantification obtenues avec un quantificateur scalaire uniforme à 8 niveaux pour les bandes AC et DC (3 bits par symbole quantifié). Pour chaque bande i et comme l'indique la figure 6.24, le premier plan de bits (constitué des bits de poids fort : MSB) divise les régions de quantification en deux parties égales, P_0 et P_1. La première partie P_0 est réservée pour les bits à valeur 0. Quant à l'autre partie P_1, elle contient tous les bits à 1. Par conséquent, le débit minimal $R1_{min}^i$ du premier plan de bits peut être estimé par :

$$R1_{min}^i = \frac{1}{2}\log_2 \frac{\sigma_i^2}{D1_i^W} \tag{6.16}$$

avec $D1_i^W$ la distorsion calculée suivant l'équation (6.11) en utilisant un quantificateur scalaire uniforme à deux niveaux dont les régions de quantification correspondent aux deux parties P_0 et P_1.

Les symboles formés par les deux bits issus chacun du premier et du deuxième plan de bits divisent les régions de quantification en quatre parties, P_{00}, P_{01}, P_{10} et P_{11}.

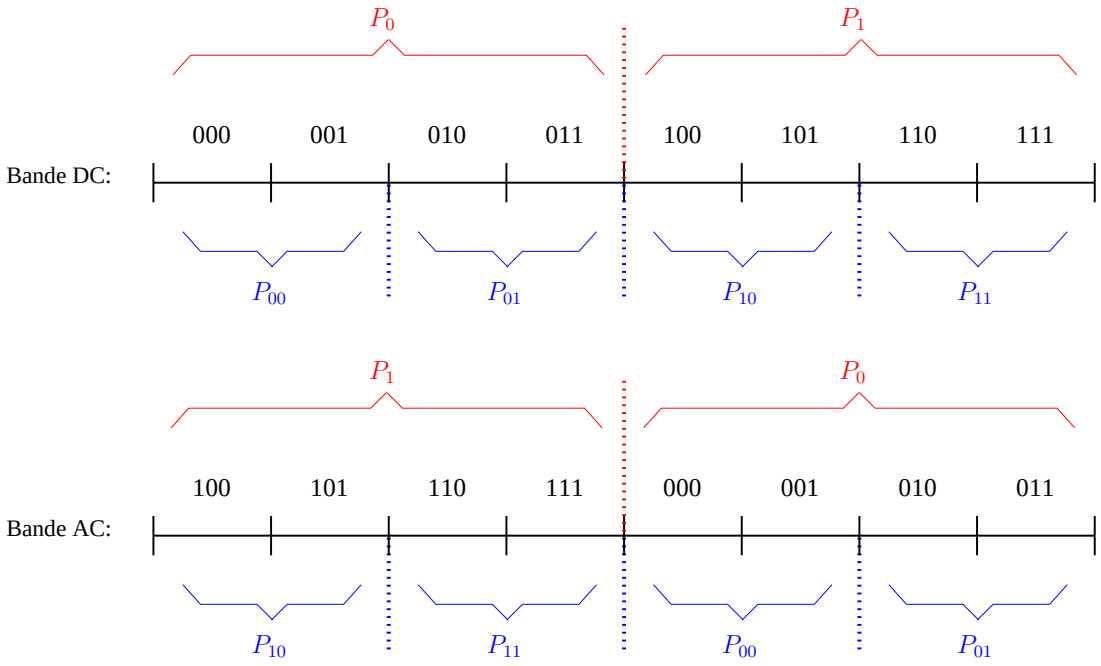

FIG. 6.24 – Quantificateur scalaire uniforme utilisé avec 8 niveaux de quantification.

Chaque partie P_j ($j = 00, 01, 10$ ou 11) contient le symbole j. Le débit minimal RS^i_{min} de ces symboles est :

$$RS^i_{min} = \frac{1}{2}\log_2 \frac{\sigma_i^2}{D2_i^W} \tag{6.17}$$

avec $D2_i^W$ la distorsion calculée suivant l'équation (6.11) en utilisant un quantificateur à quatre niveaux de quantification correspondant aux quatre parties P_j. Ainsi, le débit minimal $R2^i_{min}$ du deuxième plan de bits peut être estimé de la façon suivante :

$$R2^i_{min} = RS^i_{min} - R1^i_{min} \tag{6.18}$$

Enfin, le débit minimal $R3^i_{min}$ du troisième plan de bits est estimé à partir de l'équation (6.15) par :

$$R3^i_{min} = R^i_{min} - RS^i_{min} \tag{6.19}$$

Après l'envoi d'un nombre minimal de bits de parité par plan de bits, le processus de décodage peut être exécuté. Si la probabilité d'erreur binaire obtenue après le décodage turbo reste supérieure à 10^{-3}, alors une demande de bits de parité additionnels sera effectuée via la voie de retour. Il est clair que l'utilisation de cette voie va être diminuée grâce à l'allocation de débit à l'encodeur.

6.5.2.3 Résultats de simulation

A) Allocation de débit avec utilisation des niveaux de quantification de la figure 6.3

On désigne par CVD-CDE et CVD-CDD, les systèmes de codage vidéo distribué utilisant le contrôle de débit respectivement à l'encodeur et au décodeur. Les figures 6.25, 6.26, 6.27 et 6.28 montrent les résultats de simulation des deux codeurs CVD-CDE et CVD-CDD pour respectivement les séquences Foreman, Coast Guard, Soccer et Hall Monitor. Afin de pouvoir comparer les performances débit-distorsion de CVD-CDE à celles obtenues avec un CVD-CDD dans les sections 6.5.1.1 et 6.5.1.2, d'une part les variances du bruit de corrélation sont calculées en hors ligne et, d'autre part, les Qindexes de la figure 6.3 sont utilisés (c'est-à-dire l'algorithme de la recherche des niveaux de quantification n'est pas utilisé). On remarque qu'il n'y a aucune perte de performances de CVD-CDE par rapport à CVD-CDD que se soit avec une demande de 1 ou de 2 bits de parité par période de poinçonnage. Cependant, comme l'indique le tableau 6.4 pour la séquence Foreman, le temps de décodage t_d du codeur CVD-CDE est plus faible que celui avec le codeur CVD-CDD.

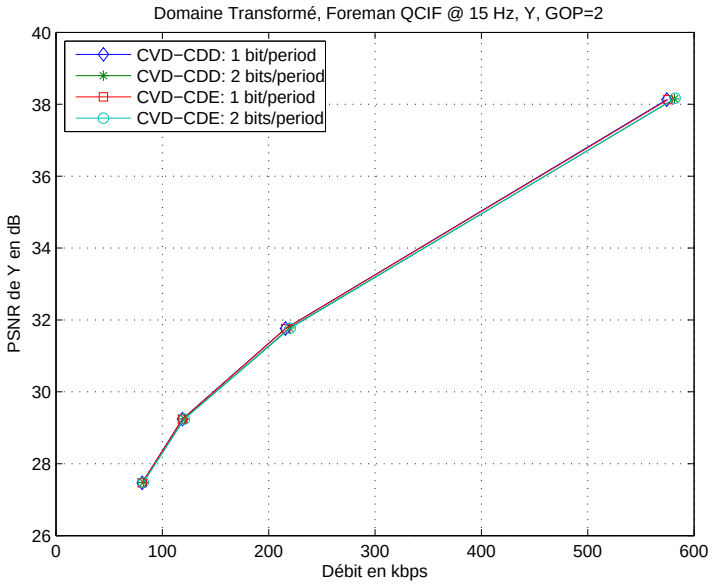

FIG. 6.25 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$, 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage et un contrôle de débit au codage ou au décodage.

Dans la suite de cette section, une fois que la probabilité d'erreur binaire après le décodeur turbo est supérieure à 10^{-3}, une demande de deux bits de parité par période de poinçonnage est effectuée. On désigne par CVD-VE et CVD-VD, les systèmes de codage vidéo distribué utilisant le contrôle de débit au codage dont les variances σ_i^2

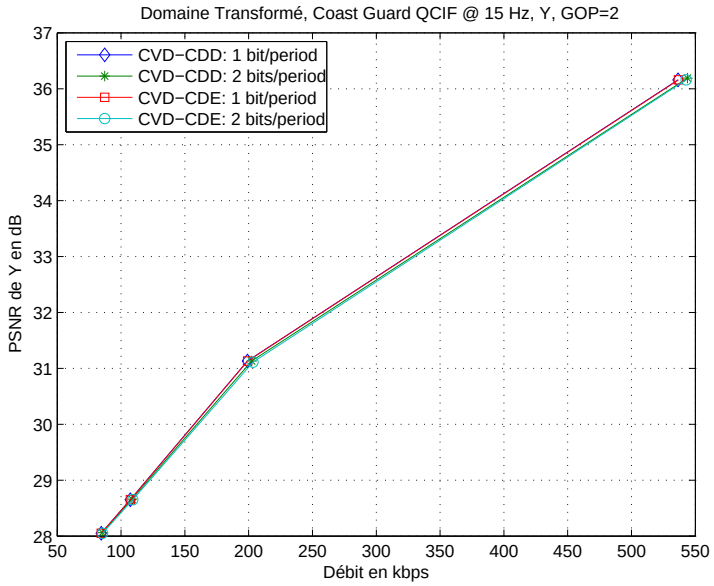

FIG. 6.26 – Résultats de simulation pour CVD de la séquence Coast Guard QCIF, 15 Hz avec un $GOP = 2$, 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage et un contrôle de débit au codage ou au décodage.

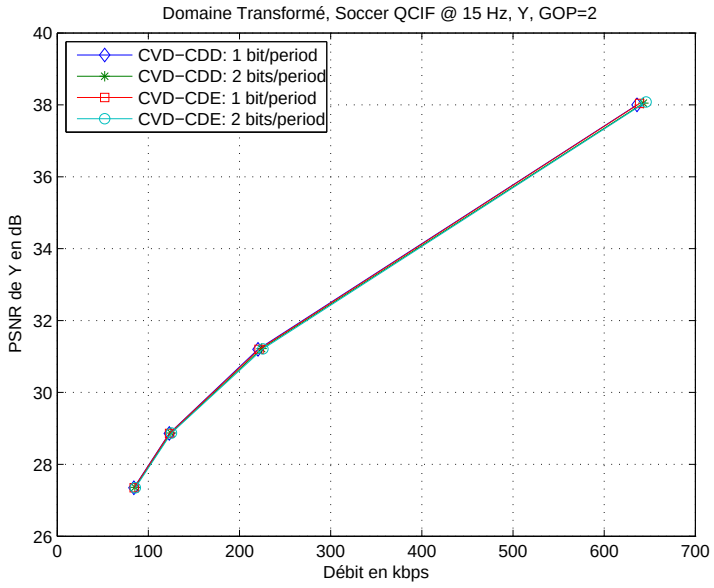

FIG. 6.27 – Résultats de simulation pour CVD de la séquence Soccer QCIF, 15 Hz avec un $GOP = 2$, 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage et un contrôle de débit au codage ou au décodage.

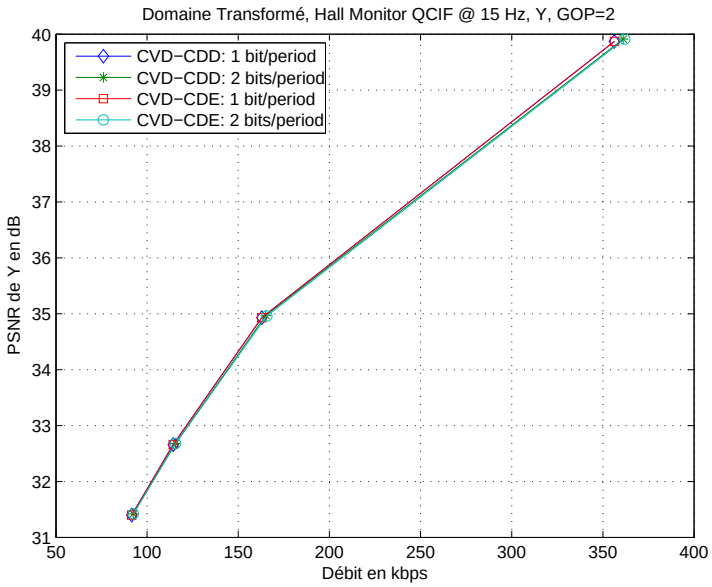

FIG. 6.28 – Résultats de simulation pour CVD de la séquence Hall Monitor QCIF, 15 Hz avec un $GOP = 2$, 1 bit/période de poinçonnage ou 2 bits/période de poinçonnage et un contrôle de débit au codage ou au décodage.

TAB. 6.4 – *Comparaison des temps de décodage des codeurs CVD-CDE et CVD-CDD avec la séquence Foreman.*

$PSNR$ en dB	t_d en min			
	CVD-CDD		CVD-CDE	
	1 bit/période	2 bits/période	1 bit/période	2 bits/période
31.76	236	143	101	66
38.15	720	407	303	165

sont calculées respectivement à l'encodeur (par une interpolation moyennée des images adjacentes K) et au décodeur.

La figure 6.29 illustre les résultats de simulation obtenus avec les codeurs CVD-VE et CVD-VD. Au décodeur turbo, les variances utilisées sont déterminées hors-ligne. On peut observer sur la figure 6.29 qu'il n'y a aucune perte de performances entre les codeurs CVD-VE et CVD-VD. Cependant, le temps de décodage avec le codeur CVD-VE est plus élevé que celui avec CVD-VD. En effet, les valeurs de variances utilisées par le codeur CVD-VE (calculées par une interpolation moyennée des images adjacentes K) ne sont qu'une approximation de celles calculées réellement entre les coefficients DCT des images W et S. A un $PSNR = 31.76$ dB, t_d du codeur CVD-VE est égale à 73 min au lieu de 66 min (tableau 6.4) avec un CVD-VD. Pour un $PSNR = 38.15$ dB, t_d est égale à 206 min avec CVD-VE contre 165 avec CVD-VD.

Le délai du codage dans un codeur CVD-VD dépend de nombreux paramètres et il est donc difficile à l'estimer. Néanmoins, ce délai est très élevé par rapport à celui obtenu avec un codeur CVD-VE.

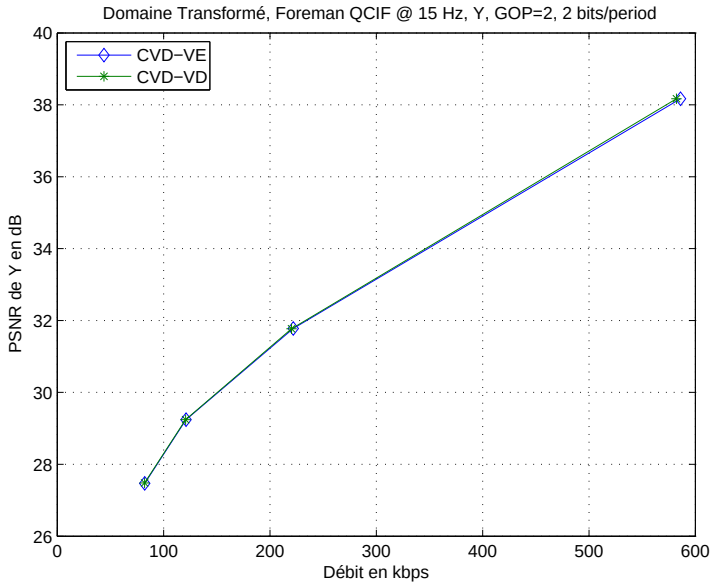

FIG. 6.29 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$, 2 bits/période de poinçonnage et un contrôle de débit au codage.

B) Allocation de débit et recherche des niveaux de quantification en ligne

Nous utilisons ici les mêmes quantificateurs pour toutes les images W. Par conséquent, pour avoir les valeurs de niveaux de quantification selon la valeur de QP du codeur H.264, les variances σ_i^2 calculées au codage (par interpolation moyennée) ou au décodage (en ligne suivant l'approche de [BAP06b]) sont déterminées en utilisant uniquement les deux premières images Key Frames K adjacentes à W. La figure 6.30 montre les valeurs de niveaux de quantification de la séquence Foreman obtenues avec

l'algorithme décrit dans 6.5.2.1 pour différentes valeurs de QP et des variances σ_i^2 calculées au codage et au décodage. Quatre valeurs de QP sont utilisées (les mêmes QP utilisées pour tous les résultats de simulation de la séquences Foreman dans la section 6.5) : 42, 39, 35 et 26 qui correspondent aux matrices respectives Qindexe=1, Qindexe=3, Qindexe=5 et Qindexe=8 de la figure 6.3. On remarque qu'il y a certaines différences entre les valeurs des niveaux de quantification obtenues avec l'algorithme de 6.5.2.1 et celles obtenues après des tests sur de nombreux séquences (figure 6.3).

σ_i^2 estimée en ligne au décodeur :

16	4	0	0
4	0	0	0
0	0	0	0
0	0	0	0

$QP = 42$

32	16	4	0
8	4	0	0
0	0	0	0
0	0	0	0

$QP = 39$

64	16	8	4
16	8	0	0
8	4	0	0
0	0	0	0

$QP = 35$

128	64	32	16
64	16	16	4
32	16	16	4
16	8	4	4

$QP = 26$

σ_i^2 estimée à l'encodeur :

16	4	0	0
4	0	0	0
0	0	0	0
0	0	0	0

$QP = 42$

32	8	4	0
8	0	0	0
0	0	0	0
0	0	0	0

$QP = 39$

64	16	8	4
16	8	0	0
8	4	4	0
0	0	0	0

$QP = 35$

128	64	32	8
64	16	16	4
32	16	16	4
16	8	4	0

$QP = 26$

FIG. 6.30 – Valeurs des niveaux de quantification de la séquence Foreman QCIF, 15 Hz, obtenues avec différentes valeurs de QP et des variances σ_i^2 calculées au codage et au décodage.

Sur la figure 6.31, les résultats de simulation utilisant les niveaux de quantification des figures 6.30 et 6.3 sont illustrés. Afin de pouvoir comparer les performances débit-distorsion des différents codeurs CVD-CDD, CVD-VE et CVD-VD, la variance σ_i^2 utilisée au décodeur turbo est déterminée en hors-ligne (OFFLINE). Dans la figure 6.31, on remarque que les performances du codeur CVD-VD sont plus proches de celles de CVD-CDD. Par contre, les performances du codeur CVD-VE ne sont pas les mêmes que celles du CVD-CDD. Pour certains débits (110 ou 567 kbps), les résultats de simulation, obtenus avec un codeur CVD-VD, ont montré que les valeurs de $PSNR$ des images Wyner-Ziv sont plus proches de celles des images Key Frames. Cependant, avec le codeur CVD-VE et pour les débits 110 et 567 kbps, les valeurs de $PSNR$ des images W ne sont pas égales à celles des images S.

La figure 6.32 montre les performances des codeurs CVD-VE et CVD-VD utilisant les niveaux de quantification de la figure 6.30 avec la variance σ_i^2, utilisée au décodeur turbo, déterminée en ligne (ONLINE) suivant l'approche de [BAP06b]. Dans ce cas

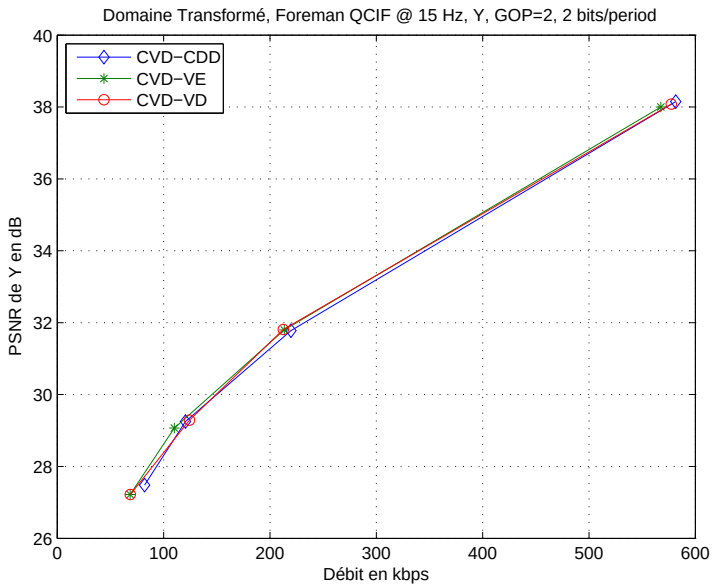

FIG. 6.31 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$, 2 bits/période de poinçonnage et un contrôle de débit au codage. La variance σ_i^2 utilisée au décodeur turbo est déterminée en hors-ligne.

aussi et avec un codeur CVD-VD, les résultats de simulation ont montré que les valeurs de $PSNR$ des images W sont plus proches de celles des images K.

Vu la faible complexité (moins de latence au codage et au décodage) du codeur CVD-VE par rapport à celle de CVD-VD, l'algorithme décrit dans 6.5.2.1 et l'allocation de débit basés sur une variance estimée par une interpolation moyennée présente une solution fiable pour l'optimisation débit-distorsion du codage vidéo distribué.

6.6 Performances débit-distorsion de notre schéma de codage vidéo distribué

Dans cette section et pour différentes séquences vidéo (avec des formats QCIF et CIF et des fréquences de 15 à 30 HZ), nous comparons les performances débit-distorsion obtenues avec notre codeur vidéo distribué (désigné par CVD) et celles obtenues avec les codeurs vidéo H.264 en mode Intra et Inter ($GOP = 16$ constitué des images I, P et B). Dans le schéma de codage vidéo distribué, les quantifications TCQ et scalaire uniforme sont utilisées respectivement pour les coefficients DCT des bandes DC et AC. Le mécanisme de contrôle de débit au codage basé sur une estimation de débit minimal théorique est considéré. Si le taux d'erreur, estimé en fonction de la mesure de confiance calculée à la sortie du décodeur turbo, est supèrieur à 10^{-3}, alors des demandes de deux bits de parité supplémentaire sont effectuées. Les variance σ_i^2 utilisées au décodeur turbo

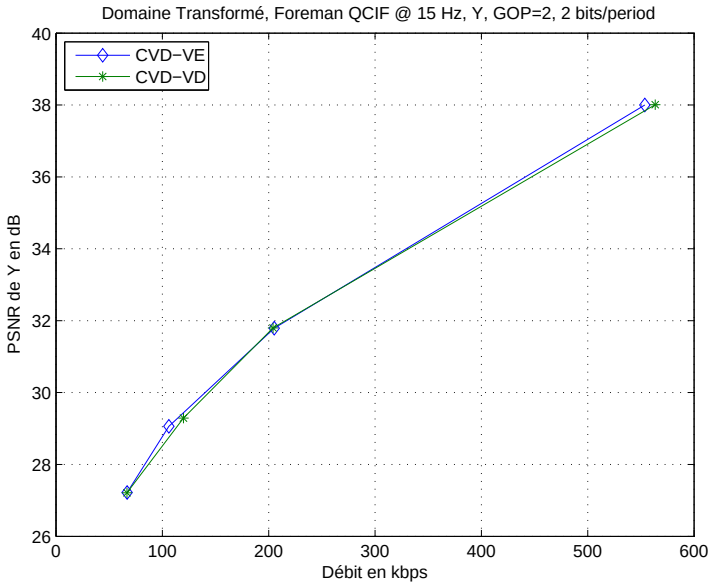

FIG. 6.32 – Résultats de simulation pour CVD de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$, 2 bits/période de poinçonnage et un contrôle de débit au codage. La variance σ_i^2 utilisée au décodeur turbo est déterminée en ligne.

sont déterminées en ligne.

Pour un GOP=2, format de séquences vidéo QCIF et à l'exception des figures 6.33-a et 6.34-d, les performances débit-distorsion de notre codeur vidéo distribué sont situées entre celles du codeur de H.264 avec un codage en modes Inter et Intra. Pour les figures 6.33-a et 6.34-d correspondantes respectivement aux résultats de simulation des séquences vidéo Foreman QCIF à 15 Hz et Carphone QCIF 30 Hz, la dégradation des performances par rapport à celles d'un codeur H.264 en Intra est due au grand débit nécessaire pour décoder les plans de bits à faible poids. En effet, la corrélation entre ces derniers et les coefficients DCT de l'information de bord est très faible. Pour remédier à ce problème, il serait intéressant de coder les plans de bits à poids faible avec un codeur entropique traditionnel.

Sur les figure 6.35-a et 6.35-b, nous montrons les performances débit-distorsion de CVD et des codeurs H.264 en modes Intra et Inter obtenues avec des séquences CIF respectivement Hall Monitor et Flower. Nous observons que les performances de CVD restent entre celles des codeurs H.264 en modes Intra et Inter.

6.7 Codage vidéo distribué en présence de bruit de canal

Nous considérons maintenant, un codeur distribué d'une séquence vidéo dont les bits de parité retenus après codage turbo des images Wyner-Ziv passent dans un canal binaire symétrique sans mémoire et de probabilité de transition q. Les images Key

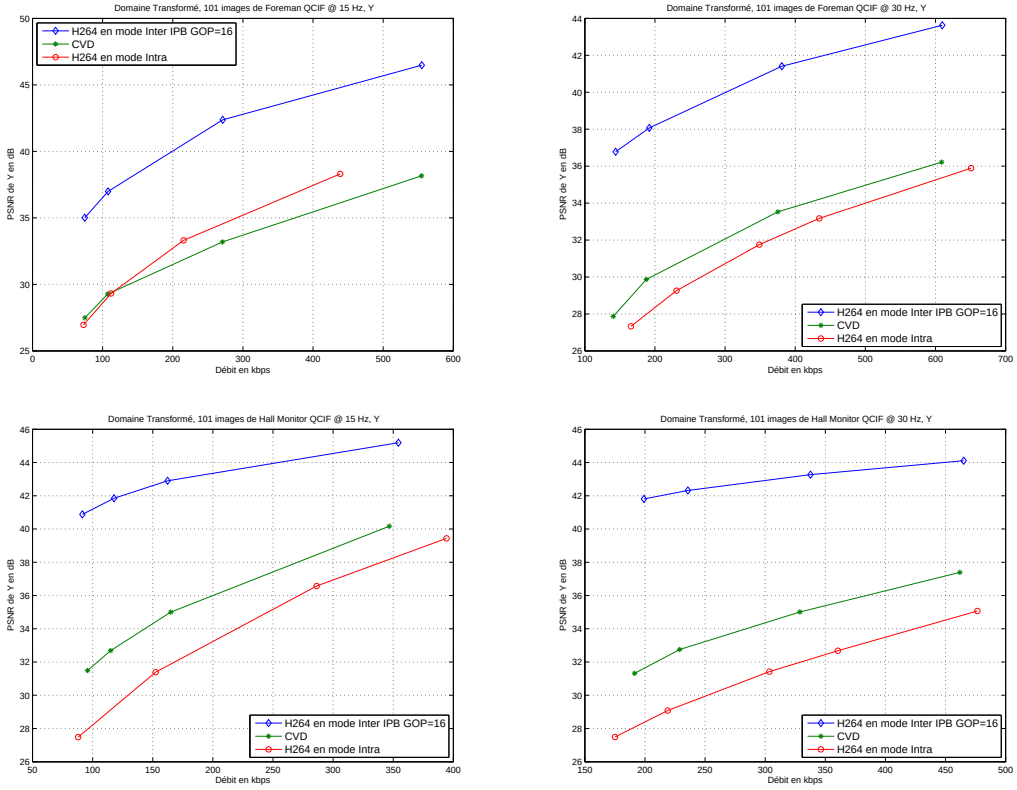

FIG. 6.33 – Performnces débit-distorsion de CVD avec $GOP = 2$ et des codeur H264 en mode Intra et Inter (IPB, $GOP = 16$) : (a) séquence Foreman QCIF 15 Hz, (b) séquence Foreman QCIF, 30 Hz, (c) séquence Hall Monitor QCIF 15 Hz et (d) séquence Hall Monitor QCIF 30 Hz.

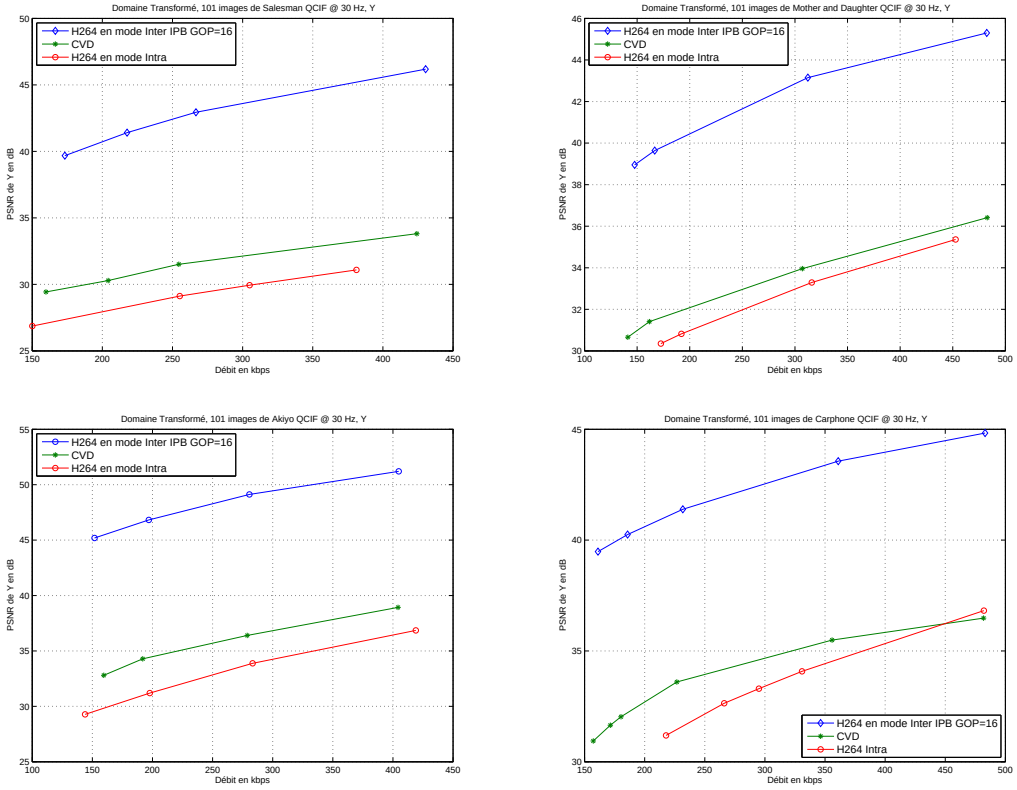

FIG. 6.34 – Performnces débit-distorsion de CVD avec $GOP = 2$ et des codeur H264 en mode Intra et Inter (IPB, $GOP = 16$) : (a) séquence Salesman QCIF 30 Hz, (b) séquence Mother and Daughter QCIF, 30 Hz, (c) séquence Akiyo QCIF 30 Hz et (d) séquence Carphone QCIF 30 Hz.

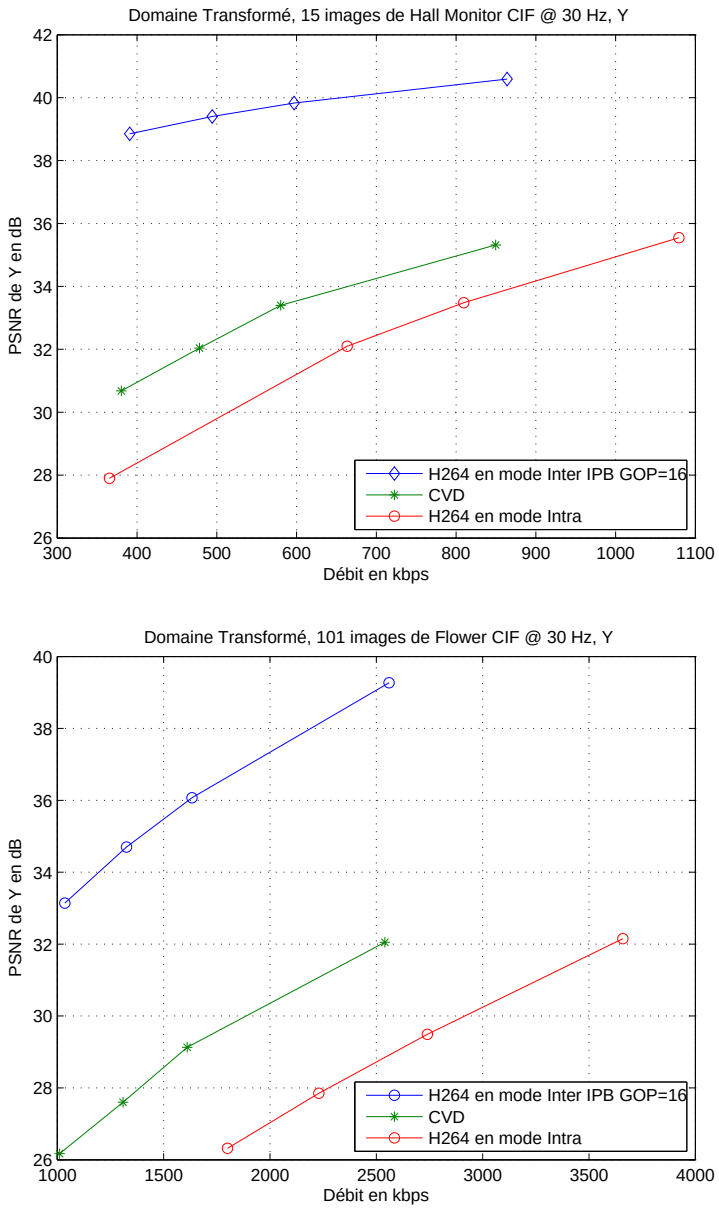

FIG. 6.35 – Performnces débit-distorsion de CVD avec $GOP = 2$ et des codeur H264 en mode Intra et Inter (IPB, $GOP = 16$) : (a) séquence Hall Monitor CIF 30 Hz, (b) séquence Flower CIF, 30 Hz.

Frames K sont codées et décodées avec un codeur H.264 sans aucun bruit de canal.

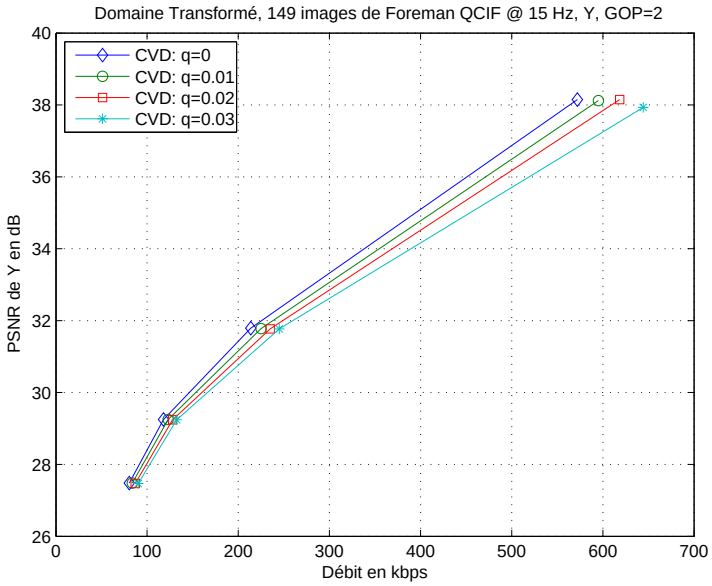

FIG. 6.36 – Performances du codeur vidéo distribué de la séquence Foreman QCIF, 15 Hz avec un $GOP = 2$ en présence d'un bruit. Les bits de parité retenus passent dans un canal CBS avec une probabilité de transition q.

Les performances du codeur vidéo distribué (CVD) en présence d'un bruit de CBS sont montrées dans la figure 6.36 pour différentes valeurs de q. Pour un débit élevé (autour de 571 kbps), on remarque que plus q augmente plus les performances en terme de $PSNR$ chutent par rapport à un système CVD sans bruit de canal. Les pertes en $PSNR$ sont de l'ordre de 0.41, 0.76 et 1.34 dB respectivement pour des valeurs de q de 0.01, 0.02 et 0.03.

La perturbation du CBS a moins d'effet sur les débits faibles d'un codeur vidéo distribué. En effet, pour un débit égal à 91 kbps, les pertes en $PSNR$ d'un codeur CVD avec des valeurs de q de 0.01, 0.02 et 0.03 par rapport à celui où $q = 0$ sont respectivement 0.17, 0.31 et 0.48 dB. Ceci se traduit par les petites valeurs des niveaux de quantification (Qindexe=1 pour un débit autour de 91 kbps) et par conséquent un nombre faible de plans de bits. En effet, à haut débit (571 kbps pour un Qindexe=8), le nombre des bits de parité des plans de bits affectés par le bruit de CBS est élevé. C'est surtout les plans de bits les moins significatifs qui contribuent le plus à la dégradation des performances.

6.8 Conclusion

Plusieurs approches ont été décrites afin d'améliorer les performances débit-distorsion du codeur vidéo distribué. Dans un premier temps et pour un GOP de taille 3, nous

nous sommes intéressés à une amélioration de la qualité de l'information de bord en utilisant le principe du codage de trois sources distribuées. Pour des faibles débits, le codeur vidéo distribué utilisant deux informations de bord présente des performances débit-distorsion meilleures que celui avec une seule information. Par contre, la solution basée sur la technique du codage de trois sources distribuées entraîne des pertes en performances à haut débit.

Dans un deuxième temps, nous avons proposé de combiner la quantification TCQ et le code turbo poinçonné dans un schéma de codage vidéo distribué. Afin de préserver la faible complexité à l'encodeur, la quantification TCQ a été appliquée seulement aux bandes DC. Quant aux autres bandes, une quantification scalaire uniforme a été utilisée. L'utilisation de la TCQ dans un système CVD a permis d'améliorer les performances débit-distorsion par rapport à d'autres solutions dont le quantificateur est scalaire uniforme. Selon la taille du GOP, nous avons montré que les gains en $PSNR$ de l'utilisation de la TCQ par rapport à la quantification scalaire uniforme dans un CVD sont au moins de 0.29 dB. La complexité d'un CVD utilisant la TCQ reste faible en comparaison de celle d'un codeur H.264 en mode Intra.

Par la suite et afin de permettre l'optimisation débit-distorsion dans un codeur vidéo distribué, nous avons présenté des solutions pour contrôler le débit au codage et au décodage. L'utilisation de la mesure de confiance à la sortie du décodeur turbo pour décider de la nécessité de faire une demande de bits de parités supplémentaire, a permis d'avoir des performances semblables à celles obtenues avec un décodeur utilisant les images originales.

Le contrôle de débit au décodage par des demandes de bits de parité multiples par période de poinçonnage au lieu d'un seul a été étudié. L'idée est de diminuer d'une part l'utilisation de la voie de retour lors de chaque demande des bits additionnels et d'autre part le temps de décodage turbo. Toutefois, l'utilisation d'une demande de deux bits de parité par période de poinçonnage réalise un compromis entre le temps de décodage global et les pertes de performances débit-distorsion. Il serait intéressant de poursuivre une étude similaire pour inclure une demande adaptative de bits de parités additionnels selon les variances de bruit de corrélation entre les coefficients DCT de W et S.

L'algorithme utilisé à l'encodeur pour optimiser le couple débit-distorsion a permis de diminuer le temps de décodage dans un codeur vidéo distribué. Deux solutions fiables pour fixer la qualité de reconstruction désirée à l'encodeur ont été présentées. La première utilise les variances du bruit de corrélation entre les coefficients DCT des images W et S envoyées par le décodeur. Quant à la deuxième solution, elle est basée sur des variances estimées par une interpolation moyennée au codage. Cette dernière présente un compromis entre la latence et la qualité de reconstruction désirée.

L'allocation de débit basée sur les variances calculées à l'encodeur a permis de moins utiliser la voie de retour dans un codeur vidéo distribué. Cette allocation de débit présente aussi une latence globale au codage et au décodage plus faible que celle basée sur des variances envoyées par le décodeur. Néanmoins, une solution combinant les deux techniques pourrait avoir une complexité plus faible au codage et au décodage. Cependant, il serait plus intéressant de trouver une solution de contrôle de débit sans l'utilisation de la voie de retour et avec une complexité à l'encodeur comparable à celle

d'un codeur H.264 en mode Intra.

Enfin, dans le cadre d'un schéma de mise en œuvre concret du codage vidéo distribué en présence d'un bruit de canal, nous avons présenté les performances débit-distorsion d'un CVD dont les bits de parité passent dans un CBS. Les résultats de simulation montrent que c'est surtout les plans de bits les moins significatifs qui contribuent le plus à la dégradation des performances en comparaison à un codeur CVD sans la présence du bruit.

Conclusion

Au cours de cette thèse, nous nous sommes intéressés à des schémas de codage de sources distribuées. Ce type de codage concerne le cas de signaux fortement corrélés que l'on code séparément et décode conjointement. Le codage de sources distribuées a été considéré comme solution potentielle pour la compression de l'information dans des applications exigeant des encodeurs avec une faible complexité. En particulier, nous avons présenté l'intérêt de ce type de codage dans le domaine de la compression vidéo distribuée.

Dans un premier temps, nous avons rappelé les notions fondamentales du codage de sources distribuées. Des rappels des théorèmes de Slepian-Wolf et de Wyner-Ziv pour des sources à valeurs respectivement discrètes et continues ont été effectués. Dans le chapitre 3 de cette thèse, nous avons proposé d'utiliser un code turbo poinçonné pour compresser les sources dans un codeur distribué. Pour certains taux de compression, une nouvelle matrice de poinçonnage a été utilisée pour améliorer les performances de décodage turbo dans le contexte de codage de sources distribuées. Outre un gain de codage, le codeur turbo poinçonné nous a permis de diminuer le temps de calcul au codage et au décodage. Par la suite, dans le but d'améliorer les performances des quantificateurs utilisés dans un schéma de codage de Wyner-Ziv, nous nous sommes intéressés à l'utilisation d'une quantification TCQ combinée à un code turbo poinçonné. En se basant sur l'argument que l'entropie conditionnelle des bits du chemin du treillis sachant l'information de bord, est égale à 1, seuls les symboles de la séquence des mots de code sont compressés. Les bits du chemin du treillis sont transmis au décodeur sans compression. Pour des débits inférieurs à 1.66 bits/symbole, nous avons observé que les performances en terme de la distorsion mesurée du codeur de sources distribuées utilisant la TCQ et le code turbo poinçonné sont à une distance de 0.916 dB de la borne théorique de Wyner-Ziv. Cependant, pour des débits plus élevés, l'utilisation des codes LDPC avec un quantificateur TCQ dans un codeur de sources distribuées donne de meilleures performances débit-distorsion. Dans une autre partie du chapitre 3, nous avons étudié le problème de codage de Wyner-Ziv avec une information de bord partielle. Une re-démonstration des limites théoriques a été proposée. Les performances débit-distorsion en fonction du nombre de niveaux de quantification de la source W ont été présentées. A la fin du chapitre 3, l'effet du bruit de canal binaire symétrique sur les performances débit-distorsion d'un schéma de codage de deux sources distribuées a été analysé. Les résultats de simulation ont montré qu'une perte en corrélation peut être observée selon la probabilité de transition de canal CBS.

Le chapitre 4 a été consacré à l'étude du codage distribué de trois sources corrélées. L'extension des théorèmes de Slepian-Wolf et de Wyner-Ziv au cas de trois sources respectivement binaires et Gaussiennes a été effectuée. Cette approche offre des nouvelles perspectives d'application du codage de sources distribuées. Nous avons démontré que le gain apporté par le codage de trois sources binaires distribuées est de permettre la compression des sources avec des corrélations plus faibles que dans le cas de deux sources. Dans un schéma de codage distribué de trois sources Gaussiennes, l'impact de la distorsion introduite par la quantification de la deuxième source (Y) sur les performances débit-distorsion du signal principal (X) a été étudié. Nous avons montré qu'un codeur distribué de trois sources Gaussiennes pouvait atteindre des performances supérieures à celles d'un codeur à deux sources. Toutefois, nous avons remarqué que plus le taux de compression est élevé, meilleures sont les performances du cas de trois sources (binaires ou Gaussiennes) par rapport au cas de deux sources. Bien qu'il y ait une perte de performances à des taux de compression faible, le débit moyen par source dans un codeur distribué à trois sources reste meilleur que celui obtenu avec le cas de deux sources.

Après un rapide rappel dans le chapitre 5 des méthodes existantes de compression vidéo distribuée, nous avons présenté dans le chapitre 6 nos contributions dans un tel schéma. Dans le contexte du codage de trois sources distribuées, un schéma de mise en œuvre utilisant des séquences vidéo a été proposée. Pour des faibles débits, le codeur vidéo distribué utilisant deux informations de bord présente des performances débit-distorsion meilleures que celles avec une seule information. Par contre, à haut débit, la solution basée sur la technique du codage de trois sources distribuées entraîne des pertes en performances. Ce qui conforte notre analyse théorique.

Par ailleurs, vu la sous-optimalité du quantificateur scalaire uniforme, nous avons proposé d'utiliser la quantification TCQ et le code turbo poinçonné dans un codeur vidéo distribué. Afin de préserver une faible complexité à l'encodeur, la quantification TCQ a été appliquée seulement sur les coefficients des bandes DC. Les coefficients des bandes AC ont été quantifiés en utilisant une quantification scalaire uniforme. La quantification TCQ a permis d'améliorer les performances débit-distorsion d'un schéma de codage vidéo distribué. Selon la taille du GOP, nous avons montré que les gains en $PSNR$ apportés par la TCQ par rapport à la quantification scalaire uniforme dans un codeur vidéo distribué sont au moins de 0.29 dB. Cependant, le temps de calcul demandé par l'utilisation de la quantification TCQ a introduit une légère complexité au niveau de l'encodeur. Toutefois, nous avons montré que la complexité totale du codeur vidéo distribué avec la TCQ reste très faible par rapport à celle d'un codeur H.264 en mode Inter ou Intra.

Afin de permettre l'optimisation débit-distorsion dans un codeur vidéo distribué, nous avons présenté des solutions pour contrôler le débit au codage et au décodage. Dans la plupart des travaux antérieurs réalisés dans le cadre du codage vidéo distribué, le contrôle du débit au décodage est basé sur le taux d'erreur binaire calculé à la fin du décodage turbo en mesurant la distance de Hamming entre le plan de bit de l'image décodée et celui de l'image originale. Ainsi, cette mesure calcule d'une manière parfaite un vrai taux d'erreur binaire. Toutefois, l'utilisation de l'image originale au décodeur

vidéo distribué n'est pas réaliste. Par conséquent, une solution plus adéquate a été proposée. En effet, une mesure de confiance à la sortie de décodeur turbo pour estimer le taux d'erreur binaire et décider de l'éventualité de faire une demande de bits de parité supplémentaire a été utilisée. Cette mesure a permis d'avoir des performances débit-distorsion d'un codeur vidéo distribué semblables à celles obtenues avec un décodeur utilisant des images originales. Le contrôle de débit au décodage basé sur des demandes de bits de parité multiples par période de poinçonnage au lieu d'un seul a été étudié. L'objectif est de diminuer, d'une part, l'utilisation de la voie de retour lors de chaque demande des bits additionnels et, d'autre part, le temps de décodage turbo.

Pour optimiser le couple débit-distorsion au codage, nous avons proposé un algorithme de contrôle de débit. Dans une première partie, l'algorithme a été utilisé pour fixer le nombre de niveaux de quantification nécessaire à chaque bande DCT de l'image Wyner-Ziv à coder et ainsi avoir la qualité de reconstruction désirée. Dans une seconde partie, l'algorithme utilisé a permis d'allouer le débit binaire nécessaire au codeur vidéo distribué. En contrepartie d'une légère augmentation de la complexité à l'encodeur, l'algorithme de contrôle de débit a permis de diminuer le temps de décodage. Pour le bon fonctionnement de l'algorithme proposé, les variances du bruit de corrélation entre les coefficients DCT des images Wyner-Ziv et interpolée sont utilisées. Par conséquent, deux solutions fiables et possibles pour déterminer les valeurs de ces variances ont été utilisées. La première solution, qui demande l'utilisation de la voie de retour dans le codeur vidéo distribué, utilise les variances envoyées par le décodeur. Quant à la deuxième solution, elle est basée sur des variances estimées par une interpolation moyennée au codage.

L'allocation de débit basée sur les variances calculées à l'encodeur a permis de moins utiliser la voie de retour dans un codeur vidéo distribué. Cette allocation de débit présente aussi une latence globale au codage et au décodage plus faible que celle basée sur des variances envoyées par le décodeur. L'algorithme qui fixe la qualité de reconstruction désirée donne de meilleures performances avec des variances calculées au décodeur que celles déterminées à l'encodeur. Cependant, nous avons remarqué que l'utilisation des variances, estimées par une interpolation moyennée au codage, conduit à un compromis entre la latence et la qualité de reconstruction désirée.

Dans le cadre d'un schéma de codage conjoint source-canal appliqué à des séquences vidéo, l'effet d'un bruit de canal sur les performances débit-distorsion d'un codeur vidéo distribué a été étudié. Seuls les bits de parité issus du codeur vidéo Wyner-Ziv ont été affectés par le bruit de canal CBS. Pour différentes valeurs de probabilité de transition du canal CBS, les résultats de simulation ont montré que c'est surtout les plans de bits les moins significatifs qui contribuent le plus à la dégradation des performances débit-distorsion d'un codeur vidéo distribué. Nous avons observé que plus le débit est faible, moins les pertes en performances du codeur vidéo distribuée sont importantes.

Ce travail peut encore être poursuivi dans le domaine théorique du codage distribué afin de déterminer les bornes pour les sources Laplaçiennes ainsi que les sources avec mémoire. Une étude pour chercher des quantificateurs plus adaptés au contexte du codage de sources distribuées serait très intéressante. Il serait bon également de :

– Appliquer d'autres familles de codes plus appropriés au contexte de codage vidéo

distribué pour s'approcher des performances du codage prédictif.

- Améliorer la qualité de l'information de bord générée par interpolation ou extrapolation dans un schéma de codage vidéo distribué surtout pour des valeurs de GOP plus élevées. Dans ce sens, on cite la méthode proposée par Maitre *et al.* [MGM06] basée sur l'utilisation de la notion de la reconstruction 3D, qui a permis d'améliorer la qualité du champ de mouvement et par la suite avoir des meilleures performances débit-distorsion du codeur vidéo distribué pour des GOP de taille élevée.
- Adapter la demande de bits de parités additionnels au décodeur selon les variances du bruit de corrélation entre les coefficients DCT des images Wyner-Ziv et interpolée.
- Trouver une solution de contrôle de débit sans l'utilisation de la voie de retour et avec une complexité à l'encodeur comparable à celle d'un codeur H.264 en mode Intra.
- Etudier le schéma de codage vidéo distribué dans une application réelle de communication radio-mobile et voir l'impact de l'interférence et du canal de Rayleigh sur les performances débit-distorsion.

Annexe A

Algorithme MAP

On fera constamment usage au cours de cette annexe de résultats théoriques ou pratiques dont la source peut être retracée dans plusieurs ouvrages sur les algorithmes BCJR et MAP [BCJR74], [BGT93], [BG96], [Rob94].

L'algorithme de MAP est un algorithme optimal. Nous allons présenter plus particulièrement l'algorithme MAP qui calcule la probabilité a posteriori du symbole d'information encodé par un codeur convolutionnel récursif systématique (CRS) de taux de codage $R = 1/2$.

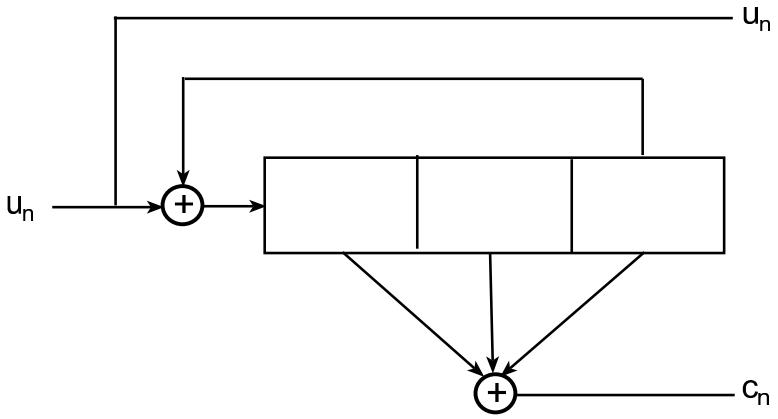

FIG. A.1 – Exemple d'un codeur convolutionnel récursif systématique $R = 1/2$, $K = 3$.

On désigne respectivement par $\mathbf{U}^N = (u_1, u_2, \ldots, u_n, \ldots, u_N)$ et $\mathbf{C}^N = (c_1, c_2, \ldots, c_n, \ldots, c_N)$ les séquences binaires systématique et parité du codeur convolutionnel CRS de taux $R = 1/2$, longueur de contrainte $K = 3$ et de polynôme générateur $h_0 = 5$ et $h_1 = 7$ ($G = [1, 7/5]$) illustré par la figure A.1.

Soient $\mathbf{X}^N = (x_1, x_2, \ldots, x_n, \ldots, x_N)$ et $\mathbf{Y}^N = (y_1, y_2, \ldots, y_n, \ldots, y_N)$ les séquences reçues au décodeur et qui sont associées respectivement à $\mathbf{U}^N$ et $\mathbf{C}^N$.

A part les séquences $\mathbf{X}^N$ et $\mathbf{Y}^N$, le décodeur a l'accès à une autre ressource qui

est la séquence d'information a priori $\tilde{\Lambda}^N$. Par conséquent, le décodeur dispose de trois séquences qu'on peut les noter par un simple vecteur $\mathbf{R}^N$ tel que :

$$\mathbf{R}^N = (\mathbf{X}^N, \mathbf{Y}^N, \tilde{\Lambda}^N) = (\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n, \dots, \mathbf{R}_N) \text{ avec } \mathbf{R}_n = (x_n, y_n, \tilde{\Lambda}_n) \quad \text{(A.1)}$$

On désigne par $\mathbf{R}_i^N$ avec $i \leq N$, un sous-vecteur de $\mathbf{R}^N$ défini par : $\mathbf{R}_i^N = (\mathbf{R}_i, \mathbf{R}_{i+1}, \dots, \mathbf{R}_N)$. Le vecteur $\mathbf{R}^N$ peut s'écrire sous la forme de $\mathbf{R}^N = (\mathbf{R}_1^n, \mathbf{R}_{n+1}^N) = \mathbf{R}_1^N$.

L'information a posteriori (rapport de vraisemblance) à l'instant n est définie par :

$$\tilde{\Lambda}_n = \ln \frac{Pr(u_n = 1|\mathbf{R}^N)}{Pr(u_n = 0|\mathbf{R}^N)} \quad \text{(A.2)}$$

L'objectif principal d'un algorithme MAP est le calcul des probabilités a posteriori du symbole $u_n = 0$ ou 1, conditionnellement sur $\mathbf{R}^N$, c'est-à-dire $Pr(u_n = 1|\mathbf{R}^N)$ et $Pr(u_n = 0|\mathbf{R}^N)$. Ces probabilités sont en fonction des probabilités de transition des états. On note par $S_{n-1} \in \{0, 1, \dots, 2^M - 1\}$ l'état du codeur après $n-1$ symboles binaires (instant $n-1$). Par la suite, l'introduction d'un symbole binaire u_n entraîne la transition de l'état du codeur de S_{n-1} vers S_n. Dans notre exemple, nous avons 4 états possibles comme l'indique la figure A.2 qui montre les différentes transitions entre l'instant $n-1$ et n.

La probabilité a posteriori peut s'exprimer par :

$$Pr(u_n = i|\mathbf{R}^N) = \sum_{s'=0}^{2^M-1} \sum_{s=0}^{2^M-1} Pr(u_n = i, S_{n-1} = s', S_n = s|\mathbf{R}^N), \qquad i = 0, 1. \quad \text{(A.3)}$$

Pour faciliter les calculs, il est bien d'introduire la probabilité conjointe $\sigma_n^i(s', s)$ définie par :

$$\sigma_n^i(s', s) = Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}^N) \quad \text{(A.4)}$$

(A.3) devient alors :

$$Pr(u_n = i|\mathbf{R}^N) = \sum_{s'=0}^{2^M-1} \sum_{s=0}^{2^M-1} \frac{\sigma_n^i(s', s)}{Pr(\mathbf{R}^N)}, \qquad i = 0, 1. \quad \text{(A.5)}$$

On peut encore simplifier $\sigma_n^i(s', s)$ par :

$$\begin{aligned}
\sigma_n^i(s', s) &= Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n, \mathbf{R}_{n+1}^N) \\
&= Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) \\
&\quad \times Pr(\mathbf{R}_{n+1}^N|u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) \\
&= Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^{n-1}, \mathbf{R}_n) \\
&\quad \times Pr(\mathbf{R}_{n+1}^N|u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) \\
&= Pr(S_{n-1} = s', \mathbf{R}_1^{n-1}) Pr(u_n = i, S_n = s, \mathbf{R}_n|S_{n-1} = s', \mathbf{R}_1^{n-1}) \\
&\quad \times Pr(\mathbf{R}_{n+1}^N|u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) \quad \text{(A.6)}
\end{aligned}$$

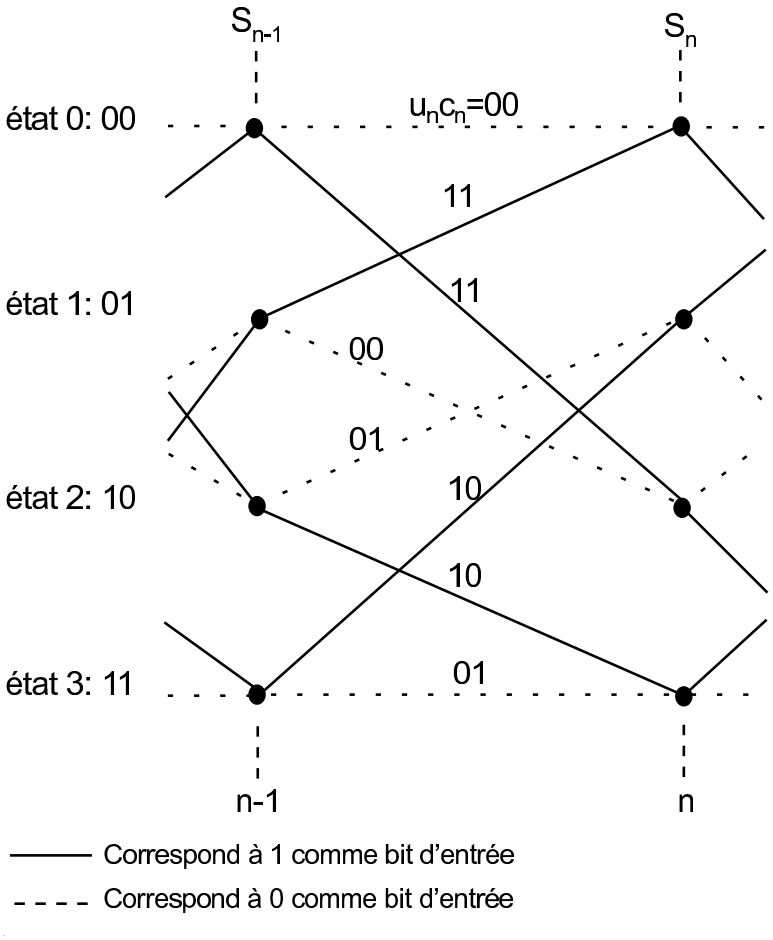

FIG. A.2 – Diagramme en treillis de la figure A.1.

Si on connaît l'état S_{n-1} alors le bit u_n, l'état S_n et le vecteur $\mathbf{R}_n$ seront indépendants de $\mathbf{R}_1^{n-1}$. Par conséquent,

$$Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s', \mathbf{R}_1^{n-1}) = Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s')$$

Par analogie, on trouve aussi :

$$Pr(\mathbf{R}_{n+1}^N | u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) = Pr(\mathbf{R}_{n+1}^N | S_n = s)$$

Ainsi, $\sigma_n^i(s', s)$ peut s'écrire par :

$$\begin{aligned} \sigma_n^i(s', s) &= Pr(S_{n-1} = s', \mathbf{R}_1^{n-1}) Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s') \\ &\quad \times Pr(\mathbf{R}_{n+1}^N | S_n = s) \\ &= \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^i(\mathbf{R}_n, s', s) \tilde{\beta}_n(s) \end{aligned} \tag{A.7}$$

où les quantités $\tilde{\alpha}_{n-1}(s')$, $\tilde{\gamma}_n^i(\mathbf{R}_n, s', s)$ et $\tilde{\beta}_n(s)$ sont définies par :

$$\tilde{\alpha}_{n-1}(s') = Pr(S_{n-1} = s', \mathbf{R}_1^{n-1}) \tag{A.8}$$

$$\tilde{\gamma}_n^i(\mathbf{R}_n, s', s) = Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s') \tag{A.9}$$

$$\tilde{\beta}_n(s) = Pr(\mathbf{R}_{n+1}^N | S_n = s) \tag{A.10}$$

Par conséquent, l'information à posteriori peut être exprimée par :

$$\begin{aligned} \Lambda_n &= \ln \frac{Pr(u_n = 1 | \mathbf{R}^N)}{Pr(u_n = 0 | \mathbf{R}^N)} \\ &= \ln \frac{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^1(\mathbf{R}_n, s', s) \tilde{\beta}_n(s)}{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^0(\mathbf{R}_n, s', s) \tilde{\beta}_n(s)} \end{aligned} \tag{A.11}$$

Les quantités $\tilde{\alpha}$, $\tilde{\gamma}$ et $\tilde{\beta}$ sont des probabilités appelées respectivement métrique d'état en avant, métrique de branche et métrique d'état en arrière. Elles sont particulièrement reliées à la probabilité de transition des états du codeur et à la probabilité de transition du canal (probabilité de corrélation entre les sources dans le cas de codage distribué). Dans la suite on va détailler les expressions de ces différentes quantités afin de simplifier le calcul du rapport de vraisemblance associé à chaque symbole d'information u_n. On

obtient :

$$\begin{aligned}
\tilde{\alpha}_n(s) &= Pr(S_n = s, \mathbf{R}_1^n) \\
&= \sum_{s'=0}^{2^M-1} \sum_{i=0}^{1} Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^n) \\
&= \sum_{s'=0}^{2^M-1} \sum_{i=0}^{1} Pr(u_n = i, S_{n-1} = s', S_n = s, \mathbf{R}_1^{n-1}, \mathbf{R}_n) \\
&= \sum_{s'=0}^{2^M-1} \sum_{i=0}^{1} Pr(S_{n-1} = s', \mathbf{R}_1^{n-1}) Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s', \mathbf{R}_1^{n-1}) \\
&= \sum_{s'=0}^{2^M-1} \sum_{i=0}^{1} Pr(S_{n-1} = s', \mathbf{R}_1^{n-1}) Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s') \\
&= \sum_{s'=0}^{2^M-1} \sum_{i=0}^{1} \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^i(\mathbf{R}_n, s', s) \qquad \text{(A.12)}
\end{aligned}$$

et

$$\begin{aligned}
\tilde{\beta}_n(s') &= Pr(\mathbf{R}_{n+1}^N | S_n = s') \\
&= \sum_{s=0}^{2^M-1} \sum_{i=0}^{1} Pr(u_{n+1} = i, S_{n+1} = s, \mathbf{R}_{n+1}^N | S_n = s') \\
&= \sum_{s=0}^{2^M-1} \sum_{i=0}^{1} Pr(u_{n+1} = i, S_{n+1} = s, \mathbf{R}_{n+1}, \mathbf{R}_{n+2}^N | S_n = s') \\
&= \sum_{s=0}^{2^M-1} \sum_{i=0}^{1} Pr(\mathbf{R}_{n+2}^N | u_{n+1} = i, S_{n+1} = s, \mathbf{R}_{n+1}, S_n = s') \\
&\quad \times \frac{Pr(u_{n+1} = i, S_{n+1} = s, \mathbf{R}_{n+1}, S_n = s')}{Pr(S_n = s')} \\
&= \sum_{s=0}^{2^M-1} \sum_{i=0}^{1} Pr(\mathbf{R}_{n+2}^N | S_{n+1} = s) \\
&\quad \times Pr(u_{n+1} = i, S_{n+1} = s, \mathbf{R}_{n+1} | S_n = s') \\
&= \sum_{s=0}^{2^M-1} \sum_{i=0}^{1} \tilde{\beta}_{n+1}(s) \tilde{\gamma}_{n+1}^i(\mathbf{R}_{n+1}, s', s) \qquad \text{(A.13)}
\end{aligned}$$

Il reste maintenant à simplifier d'avantage la fonction de probabilité de $\tilde{\gamma}_n^i(\mathbf{R}_{n+1}, s', s)$. A partir de (A.9), on peut écrire :

$$\begin{aligned}
\tilde{\gamma}_n^i(\mathbf{R}_n, s', s) &= Pr(u_n = i, S_n = s, \mathbf{R}_n | S_{n-1} = s') \\
&= Pr(\mathbf{R}_n | u_n = i, S_n = s, S_{n-1} = s') \\
&\quad \times Pr(S_n = s | u_n = i, S_{n-1} = s') Pr(u_n = i) \qquad \text{(A.14)}
\end{aligned}$$

Changeant $\mathbf{R}_n$ par $(x_n, y_n, \tilde{\Lambda}_n)$ et supposant que les entrées de décodeur sont indépendantes entre elles conditionnellement sur $u_n = i$, $S_n = s$ et $S_{n-1} = s'$, alors :

$$\begin{aligned}
\tilde{\gamma}_n^i(\mathbf{R}_n, s', s) &= Pr\big(x_n, y_n, \tilde{\Lambda}_n | u_n = i, S_n = s, S_{n-1} = s'\big) \ . \\
&\quad Pr(S_n = s | u_n = i, S_{n-1} = s') Pr(u_n = i) \\
&= Pr(x_n | u_n = i, S_n = s, S_{n-1} = s') \\
&\quad \times Pr(y_n | u_n = i, S_n = s, S_{n-1} = s') \\
&\quad \times Pr(\tilde{\Lambda}_n | u_n = i, S_n = s, S_{n-1} = s') \\
&\quad \times Pr(S_n = s | u_n = i, S_{n-1} = s') Pr(u_n = i) \\
&= Pr(x_n | u_n = i) Pr(y_n | u_n = i, S_{n-1} = s') \\
&\quad \times Pr(\tilde{\Lambda}_n | u_n = i) Pr(u_n = i) \\
&\quad \times Pr(S_n = s | u_n = i, S_{n-1} = s') \\
&= Pr(x_n | u_n = i) Pr(y_n | u_n = i, S_{n-1} = s') \\
&\quad \times Pr(u_n = i | \tilde{\Lambda}_n) Pr(\tilde{\Lambda}_n) \\
&\quad \times Pr(S_n = s | u_n = i, S_{n-1} = s') \qquad \text{(A.15)}
\end{aligned}$$

Par la suite, l'information à posteriori peut être composé en trois termes :

$$\begin{aligned}
\Lambda_n &= \ln \frac{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^1(\mathbf{R}_n, s', s) \tilde{\beta}_n(s)}{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') \tilde{\gamma}_n^0(\mathbf{R}_n, s', s) \tilde{\beta}_n(s)} \\
&= \ln \frac{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') Pr(x_n | u_n = 1) Pr(y_n | u_n = 1, S_{n-1} = s')}{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') Pr(x_n | u_n = 0) Pr(y_n | u_n = 0, S_{n-1} = s')} \\
&\quad \times \frac{Pr(u_n = 1 | \tilde{\Lambda}_n) Pr(\tilde{\Lambda}_n) Pr(S_n = s | u_n = 1, S_{n-1} = s') \tilde{\beta}_n(s)}{Pr(u_n = 0 | \tilde{\Lambda}_n) Pr(\tilde{\Lambda}_n) Pr(S_n = s | u_n = 0, S_{n-1} = s') \tilde{\beta}_n(s)} \\
&= \ln \frac{Pr(x_n | u_n = 1) Pr(u_n = 1 | \tilde{\Lambda}_n) \sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s')}{Pr(x_n | u_n = 0) Pr(u_n = 0 | \tilde{\Lambda}_n) \sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s')} \\
&\quad \times \frac{Pr(y_n | u_n = 1, S_{n-1} = s') Pr(S_n = s | u_n = 1, S_{n-1} = s') \tilde{\beta}_n(s)}{Pr(y_n | u_n = 0, S_{n-1} = s') Pr(S_n = s | u_n = 0, S_{n-1} = s') \tilde{\beta}_n(s)} \\
&= \ln \frac{Pr(x_n | u_n = 1)}{Pr(x_n | u_n = 0)} + \ln \frac{Pr(u_n = 1 | \tilde{\Lambda}_n)}{Pr(u_n = 0 | \tilde{\Lambda}_n)} \\
&\quad + \ln \frac{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') Pr(y_n | u_n = 1, S_{n-1} = s') Pr(S_n = s | u_n = 1, S_{n-1} = s') \tilde{\beta}_n(s)}{\sum_{s'} \sum_s \tilde{\alpha}_{n-1}(s') Pr(y_n | u_n = 0, S_{n-1} = s') Pr(S_n = s | u_n = 0, S_{n-1} = s') \tilde{\beta}_n(s)}
\end{aligned} \qquad \text{(A.16)}$$

Le terme $\ln \frac{Pr(u_n=1|\tilde{\Lambda}_n)}{Pr(u_n=0|\tilde{\Lambda}_n)}$ est appelé information a priori. Lors de décodage turbo et à chaque itération, c'est le dernier terme de l'équation (A.16) désigné par information extrinsèque qui est échangé entre les décodeurs SISO.

Annexe B

Calcul de la probabilité $p(\hat{y}, z|x_q)$ pour le décodage turbo de 3 sources Gaussiennes

Soient :
- $N_1 = N(0, \sigma_1^2)$, $Z \sim \mathcal{N}(0, \sigma_Z^2 = \frac{1-\sigma_1^2}{4})$ et $N = N(0, \sigma^2)$ trois variables aléatoires gaussiennes et indépendantes,
- $Y = Z + N_1 \sim \mathcal{N}(0, \sigma_Z^2 + \sigma_1^2)$ une variable aléatoire gaussienne et indépendante de N, et
- $X = Y + Z + N \sim \mathcal{N}(0, 1 + \sigma^2)$ une autre variable gaussienne.

On cherche à déterminer la probabilité $p(\hat{y}, z|x_q)$. Soit $x_q \in [a, b]$. Pour une quantification scalaire la quantité $p(x|x_q)$ est nulle en dehors de l'intervalle $[a, b]$.

$$\begin{aligned} p(\hat{y}, z|x_q) &= \frac{p(x_q, \hat{y}, z)}{p(x_q)} = \frac{\int_x p(x_q, x, \hat{y}, z)dx}{p(x_q)} \\ &= \frac{\int_x p(x, x_q)p(\hat{y}, z|x, x_q)dx}{p(x_q)} \\ &= \int_a^b p(x|x_q)p(\hat{y}, z|x)dx \\ &= \int_a^b p(x|x_q)\frac{p(x, \hat{y}, z)}{p(x)}dx \end{aligned} \tag{B.1}$$

or

$$p(x|x_q) = \frac{p(x_q|x)p(x)}{p(x_q)} = \frac{p(x)}{\int_a^b p(x)dx} \tag{B.2}$$

La probabilité $p(x_q|x)$ est égale à 1 parce que la connaissance de la réalisation x permet de déterminer la valeur quantifiée x_q.

L'équation (B.1) devient :

$$p(\hat{y}, z|x_q) = \frac{1}{\int_a^b p(x)dx} \int_a^b p(x, \hat{y}, z)dx \tag{B.3}$$

La valeur à déterminer est la probabilité $p(x, \hat{y}, z)$. Soit $\hat{Y} = Y + N_Y = Z + N_1 + N_Y$, avec N_Y un bruit Gaussien *i.i.d* de moyenne nulle et de variance $D_Y = \frac{\sigma_1^2}{2^{R^*_{Y|Z}(D_Y)}}$. On suppose que N_Y est indépendant de Z, de N_1 et de N.

Soit le vecteur colonne $\mathbf{m} = (x, \hat{y}, z)$. Soit aussi la matrice de covariance $\mathbf{M}$ telle que :

$$\mathbf{M} = \begin{pmatrix} 1+\sigma^2 & \mu_{x\hat{y}} & \mu_{xz} \\ \mu_{x\hat{y}} & \sigma_Z^2 + \sigma_1^2 + D_Y & \mu_{\hat{y}z} \\ \mu_{xz} & \mu_{\hat{y}z} & \sigma_Z^2 \end{pmatrix} \tag{B.4}$$

avec

- $\mu_{x\hat{y}} = E((2Z + N_1 + N)(Z + N_1 + N_Y)) = E(2Z^2) + E(N_1^2) = 2\sigma_Z^2 + \sigma_1^2$
- $\mu_{xz} = E((2Z + N_1 + N)Z) = E(2Z^2) = 2\sigma_Z^2\sigma_1^2$
- $\mu_{\hat{y}z} = E((Z + N_1 + N_Y)Z) = E(Z^2) = \sigma_Z^2$

La probabilité $p(x, \hat{y}, z)$ peut s'exprimer par :

$$p(x, \hat{y}, z) \quad = \quad \frac{1}{(2\pi)^{3/2}[det(\mathbf{M})]^{1/2}} \; exp[-\frac{1}{2}(\mathbf{m}' \, \mathbf{M}^{-1} \, \mathbf{m})] \tag{B.5}$$

avec $\mathbf{M}^{-1}$ la matrice inverse de $\mathbf{M}$, $det(\mathbf{M})$ le déterminant de $\mathbf{M}$ et $\mathbf{m}'$ le transposé de $\mathbf{m}$. Tout calcul fait on aura :

$$det(\mathbf{M}) \quad = \quad \sigma_Z^2(\sigma_1^2\sigma^2 + D_Y(\sigma_1^2 + \sigma^2)) \tag{B.6}$$

et

$$\mathbf{M}^{-1} = \frac{1}{det(\mathbf{M})} \begin{pmatrix} \sigma_Z^2(\sigma_1^2 + D_Y) & -\sigma_Z^2\sigma_1^2 & -\sigma_Z^2(\sigma_1^2 + 2D_Y) \\ -\sigma_Z^2\sigma_1^2 & \sigma_Z^2(\sigma_1^2 + \sigma^2) & \sigma_Z^2(\sigma_1^2 - \sigma^2) \\ -\sigma_Z^2(\sigma_1^2 + 2D_Y) & \sigma_Z^2(\sigma_1^2 - \sigma^2) & \sigma_Z^2(\sigma_1^2 + \sigma^2) + \sigma_1^2\sigma^2 \\ & & +D_Y(4\sigma_Z^2 + \sigma_1^2 + \sigma^2) \end{pmatrix} \tag{B.7}$$

Posons, $w = \sigma_Z^2(\sigma_1^2 + \sigma^2) + \sigma_1^2\sigma^2 + D_Y(4\sigma_Z^2 + \sigma_1^2 + \sigma^2)$. D'où (B.5) devient :

$$\begin{aligned} p(x, \hat{y}, z) \quad = \quad & \frac{1}{(2\pi)^{3/2}\sqrt{\sigma_Z^2(\sigma_1^2\sigma^2 + D_Y(\sigma_1^2 + \sigma^2))}} \\ & exp[-\frac{1/2}{\sigma_1^2\sigma^2 + D_Y(\sigma_1^2 + \sigma^2)}((\sigma_1^2 + D_Y)x^2 - 2((\hat{y} + z)\sigma_1^2 + 2zD_Y)x \\ & +(\sigma_1^2 + \sigma^2)\hat{y}^2 + 2(\sigma_1^2 - \sigma^2)z\hat{y} + \frac{w}{\sigma_Z^2}z^2] \end{aligned} \tag{B.8}$$

Il peut être vérifié que $p(x, \hat{y}, z)$ s'exprime par :

$$\begin{aligned} p(x,\hat{y},z) &= \frac{1}{2\pi}\sqrt{\frac{\sigma_1^2+D_Y}{\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2)}} \\ & exp[-\frac{\sigma_1^2+D_Y}{2(\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2))}(x-\frac{(\hat{y}+z)\sigma_1^2+2zD_Y}{\sigma_1^2+D_Y})^2] \\ & \times p(\hat{y},z) \end{aligned} \tag{B.9}$$

avec $p(\hat{y}, z)$ la probabilité conjointe entre $\hat{Y}$ et Z définie par :

$$p(\hat{y},z) = \frac{1}{\sqrt{2\pi\sigma_Z^2(\sigma_1^2+D_Y)}} exp[-\frac{1}{2(\sigma_1^2+D_Y)}(\hat{y}^2-2\hat{y}z+z^2\frac{\sigma_Z^2+\sigma_1^2+D_Y}{\sigma_Z^2})] \tag{B.10}$$

Donc, (B.3) peut s'écrire :

$$\begin{aligned} p(\hat{y},z|x_q) &= \frac{1}{\int_a^b p(x)dx}\int_a^b p(x,\hat{y},z)dx \\ &= \frac{p(\hat{y},z)}{\int_a^b p(x)dx}\int_a^b \frac{1}{2\pi}\sqrt{\frac{\sigma_1^2+D_Y}{\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2)}} \\ & exp[-\frac{\sigma_1^2+D_Y}{2(\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2))}(x-\frac{(\hat{y}+z)\sigma_1^2+2zD_Y}{\sigma_1^2+D_Y})^2] \end{aligned} \tag{B.11}$$

En utilisant la fonction erf définie par :

$$erf(t) = \frac{2}{\sqrt{\pi}}\int_0^t exp(-v^2)dv \tag{B.12}$$

l'équation (B.11) peut être simplifiée par :

$$\begin{aligned} p(\hat{y},z|x_q) &= \frac{p(\hat{y},z)}{2\sqrt{(2\pi)}\int_a^b p(x)dx}[erf((b-\frac{(\hat{y}+z)\sigma_1^2+2zD_Y}{\sigma_1^2+D_Y})\sqrt{\frac{\sigma_1^2+D_Y}{\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2)}}) \\ & erf((a-\frac{(\hat{y}+z)\sigma_1^2+2zD_Y}{\sigma_1^2+D_Y})\sqrt{\frac{\sigma_1^2+D_Y}{\sigma_1^2\sigma^2+D_Y(\sigma_1^2+\sigma^2)}})] \end{aligned} \tag{B.13}$$

Glossaire

AC : *Alternating current.*
APP : *A Posteriori Probability.*
AWGN : *Additive White Gaussian Noise.*
BCJR : *Bahl, Cocke, Jelinek, and Raviv.*
BER : *Bit Error Rate.*
CBS : *Canal Binaire Symétrique.*
CLV : *Codeur à Longueur Variable.*
CRC : *Cyclic Redundancy Check.*
CRS : *Convolutionnel Récursif Systématique.*
CSD : *Codage de Sources Distribuées.*
CSM : *Codage de Source Multiterminal.*
CSNR : *Correlation-Signal to Noise Ratio.*
CVD : *Codage Vidéo Distribuée.*
dB : *Décibel.*
DC : *Direct Current.*
DCT : *Discrete Cosine Transform.*
DISCOVER : *DIStributed COding for Video sERvices.*
DISCUS : *DIstributed Source Coding Using Syndromes.*
EQM : *Erreur Quadratique Moyenne.*
FSM : *Finite-State Machine.*
GOP : *Group Of Pictures.*
H.26x : *Norme de compression vidéo.*
IDCT : *Inverse Discrete Cosine Transform.*
Inter : *mode de codage avec utilisation de prédiction temporelle.*
Intra : *mode de codage sans utilisation de prédiction temporelle.*
IRA : *Irregular Repeat Accumulate.*
JPEG2000 : *Joint Photographic Experts Group.* Norme de compression d'image par ondelettes.
LDPC : *Low Density Parity Check.*
LLR : *Logarithm Likelihood Ratio.*
MAP : *Maximum A Posteriori.*
MMSE : *Minimum of Mean Square Error .*
MPEG : *Motion Picture Expert Group.*
MSB : *Most Significant Bit.*

PRISM : *Power-efficient, Robust, hIghcompression, Syndrome-based Multimedia coding.*
PSNR : *Peak Signal to Noise Ratio.*
QCIF : *Quarter Common Intermediate Format* for images.
QP : *Quantization Parameter.*
QU : *Quantificateur scalaire Uniforme.*
SISO : *Soft Input Soft Output.*
SNR : *Signal to Noise Ratio.*
TCM : *Trellis Coded Modulation.*
TCQ : *Trellis Coded Quantization.*
VLC : *Variable Code Length.*
WVMF : *Weighted Vector Median Filters.*

Publications

Journaux Internationaux

[1] K. Lajnef, C. Guillemot and P. Siohan
"Distributed Coding of Three Binary and Gaussian Correlated Sources using Punctured Turbo Codes,"
EURASIP Signal Processing Journal, Special Issue on Distributed Source Coding, Vol. 86, Issue 11, pp. 3131–3149, Nov. 2006.

Conférences Internationales

[2] K. Lajnef, C. Guillemot and P. Siohan
"Distributed Coding of Three sources using Punctured Turbo Codes,"
IEEE Intl. workshop on Multimedia Signal Processing, MMSP'04, Sept. 2004.

[3] K. Lajnef, C. Guillemot and P. Siohan
"Wyner-Ziv Coding of Three Correlated Gaussian Sources Using Punctured Turbo Codes,"
IEEE Symposium on Signal Processing and Information Technology, ISSPIT2005, Dec. 2005.

Conférences Nationales

[4] K. Lajnef, C. Guillemot and P. Siohan
"Codage de sources distribuées : analyse de performances et extension à trois sources,"
CORESA, May 2004.

[5] K. Lajnef, C. Guillemot and P. Siohan
"Compression Vidéo Distribuée utilisant la TCQ et un Turbo Code,"
CORESA, Nov. 2006.

Rapports

Etude et application du codage de sources distribuées aux communications vidéo. LIVRABLE 1 : Rapport de recherche sur les différentes techniques de CSD à 2 ou 3 sources. *Numéro du CRE : 175160*, juin 2005.

Bibliographie

[ABBC96] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini. Adaptively weighted vector-median filters for motion fields smoothing. *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Vol. 4, pp. 2267–2270, Georgia, USA, May 1996.

[ABP05] J. Ascenso, C. Brites, and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Slovak Republic, Jul. 2005.

[ABP06] J. Ascenso, C. Brites, and F. Pereira. Content adaptive Wyner-Ziv video coding driven by motion activity. *Proc. IEEE Intl. conference on Image Processing (ICIP)*, Atlanta, USA, Oct. 2006.

[AG02] A. Aaron and B. Girod. Compression with side information using turbo codes. *Proc. IEEE Data Compression Conference (DCC)*, pp. 252–261, Jan. 2002.

[AG04] A. Aaron and B. Girod. Wyner-Ziv video coding with low-encoder complexity. *Proc. Picture Coding Symposium (PCS)*, Invited paper, San Francisco, CA, Dec. 2004.

[ARG04] A. Aaron, S. Rane, and B. Girod. Wyner-Ziv video coding with hash-based motion compensation at the receiver. *Proc. IEEE International Conference on Image Processing*, Vol. 5, pp. 3097-3100, Singapore, Oct. 2004.

[ARRMG03] A. Aaron, S. Rane, D. Rebollo-Monedero, and B. Girod. Systematic lossy forward error protection for video waveforms. *Proc. IEEE International Conference on Image Processing*, Vol. 1, pp. 609-612, Barcelona, Spain, Sept. 2003.

[ARSG04] A. Aaron, S. Rane, E. Setton, and B. Girod. Transform-domain Wyner-Ziv codec for video. *Proc. SPIE Conference on Visual Communication and Image Processing*, San Jose, California, USA,Jan. 2004.

[ARZG03] A. Aaron, S. Rane, R. Zhang, and B. Girod. Wyner-Ziv coding for video : Applications to compression and error resilience. *Proc. IEEE Data Compression Conference (DCC)*, pp. 93-102, Snowbird, USA, Mar. 2003.

[ASG03] A. Aaron, E. Setton, and B. Girod. Towards practical Wyner-Ziv coding of video. *Proc. IEEE International Conference on Image Processing*, Vol. 3, pp. 869-872, Barcelona, Spain, Sept. 2003.

[AZG02] A. Aaron, R. Zhang, and B. Girod. Wyner-Ziv coding of motion video. *Proc. 36th Asilomar Conference on Signals, Systems and Computer*, Pacific Grove, USA, Nov. 2002.

[BAP06a] C. Brites, J. Ascenso, and F. Pereira. Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. *Proc. IEEE Intl. conference on Image Processing (ICIP)*, Atlanta, USA, Oct. 2006.

[BAP06b] C. Brites, J. Ascenso, and F. Pereira. Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. *Picture Coding Symposium (PCS)*, Beijing, China, Apr. 2006.

[BAP06c] C. Brites, J. Ascenso, and F. Pereira. Improving transform domain Wyner-Ziv video coding performance. *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 14–19, Toulouse, France, May 2006.

[BCJR74] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[Ber78] T. Berger. Multiterminal source coding. *The Information Theory Approach to Communications*, G. Longo, Ed. Vol. 229 of CISM Courses and Lectures, pp. 171–231, Springer-Verlag, 1978.

[BG96] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding : Turbo codes. *IEEE Trans. Communications*, vol. 44, pp. 1261–1271, Oct. 1996.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error correcting coding and decoding : Turbo codes. *Proceeding of ICC*, Geneva, Switzerland, pp. 1064–1070, May 1993.

[BM01] J. Bacjsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. *Global Communication Symposium*, pp. 1400–1404, Nov. 2001.

[BZV96] T. Berger, Z. Zhang, and H. Viswanathan. The CEO problem. *IEEE Trans. Inform. Theory*, vol. 42, no. 3, pp. 887–902, May 1996.

[CFRU01] S. Chung, G. D. Forney, T. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 db of the Shannon limit. *IEEE Communications Letters*, vol. 5, pp. 58–60, Feb. 2001.

[CPR03] J. Chou, S. S. Pradhan, and K. Ramchandran. Turbo and trellis-based constructions for source coding with side information. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2003.

[CS82] J. H. Conway and N. J. A. Sloane. A lower bound on the average error of vector quantizers. *IEEE Trans. Inform. Theory*, vol. 28, pp. 227–232, 1982.

[CS93] J. H. Conway and N. J. Sloane. *Sphere Packings, Lattices, and Groups, 2nd ed. New York*. Springer-Verlag, 1993.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons , Inc., New York, 1991.

[CX04] S. Cheng and Z. Xiong. Successive refinement for the Wyner-Ziv problem and layered code design. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2004.

[FKJ99] M. Fan, S.C. Kwatra, and K. Junghwan. Analysis of puncturing pattern for high rate turbo codes. *IEEE Military Com.*, 1999.

[FMW91] T. R. Fischer, M. W. Marcellin, and M. Wang. Trellis-coded vector quantization. *IEEE Trans. Inform. Theory*, vol. 37, pp. 1551–1566, Nov. 1991.

[For73] G. D. Forney. The Viterbi algorithm. *IEEE Proceedings*, Vol. 61, pp. 268–278, Mar. 1973.

[Gal62] R. G. Gallager. Low density parity check codes. *Trans. IRE Prof. Group on Inform. Theory*, vol. 8, pp.2l–28, Jan. 1962.

[Gal63] R. G. Gallager. Low density parity check codes. *no. 21 in Research Monograph Series*, Cambridge, MA : MIT Press, 1963.

[Gal69] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons , Inc., New York, Jan. 1969.

[GARRM05] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proc. IEEE, Special issue on advances in video coding and delivery*, vol. 93, No. 1, pp. 71–83, Invited paper, Jan. 2005.

[GF01] J. Garcia-Frias. Joint source-channel decoding of correlated sources over noisy channels. *Proc. IEEE Data Compression Conference (DCC)*, pp. 283–292, Mar. 2001.

[GFZ01a] J. Garcia-Frias and Y. Zhao. Data compression of unknown single and correlated binary sources using punctured turbo codes. *Proc. of the 39th Allerton Conference on Communication, Control and Computing*, Allerton, Illinois, Oct. 2001.

[GFZ01b] J. Garcia-Frias and Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, pp. 417–419, Oct. 2001.

[GG92] A. Gerscho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publisher, 1992.

[GN98] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2384,Oct. 1998.

[Gra84] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, pp. 4-29, Avril, 1984.

[HB89] D. Haccoun and G. Begin. High-rate punctured convolutional codes for viterbi and sequential decoding. *IEEE Trans. Comm.*, vol. 37, No. 11, pp. 1113–1125, Nov. 1989.

[HLS00] P. Hoeher, I. Land, and U. Sorger. Log-likelihood values and Monte Carlo simulation - some fundamental results. *Proc. 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, pp. 43–46, Sept. 2000.

[HPN97] B. Haskell, A. Puri, and A. Netravali. *Digital Video : An Introduction to MPEG-2*. Chapman & Hall, New York, USA, 1997.

[Huf52] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40, pp. 1098–1101, 1952.

[JAI97] A. Kh. Al Jabri and S. Al-Issa. Zerro-error codes for correlated information sources. *Proc. of Cryptography*, pp. 17–22, Dec. 1997.

[JZGS03a] M. Jeanne, R. Zhang, B. Girod, and P. Siohan. Codage de sources distribuées : comparaison de 2 systèmes approchant la borne de Wyner-Ziv. *Actes de CORESA*, Jan. 2003.

[JZGS03b] M. Jeanne, R. Zhang, B. Girod, and P. Siohan. Distributed source coding : comparaison of two methods close to the Wyner-Ziv bound. *Proc. IEEE International Symposium on Information Theory (ISIT)*, Yokohama, Japan, Jun./Jul. 2003.

[Laj01] K. Lajnef. Etude des performances des codes turbo. *Ecole Polytechnique de Montréal*, Master report, Jun. 2001.

[LCLX04] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong. Slepian-Wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding : Performance analysis and code design. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2004.

[LLN+03] A. D. Liveris, C. Lan, K. R. Narayanan, Z. Xiong, and C. N. Georghiades. Slepian-Wolf coding of three binary sources using LDPC codes. *Proc. International Symposium on Turbo Codes and related Topics, Brest, France*, pp. 63–66, Sept. 2003.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, Vol. 28, pp. 129–137, 1982.

[LXG02a] A. Liveris, Z. Xiong, and C. Georghiades. Joint source-channel coding of binary sources with side information at the decoder using IRA codes. *Proceedings of IEEE Multimedia Signal Processing Workshop*, pp. 53-56, St. Thomas, US Virgin Islands, Dec. 2002.

[LXG02b] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, vol. 6, pp. 440–442, Oct. 2002.

[Max60] J. Max. Quantizing for minimum distortion. *IRE Trans. Inform. Theory*, vol. 6, pp. 7–12, 1960.

[MF90] M. Marcellin and T. Fischer. Trellis coded quantization of memoryless and Gauss-Markov sources. *IEEE Trans. Commun.*, vol. 38, pp. 82–93, 1990.

[MGM06] M. Maitre, C. Guillemot, and L. Morin. 3d scene modeling for distributed video coding. *Proc. IEEE Intl. conference on Image Processing (ICIP)*, Atlanta, USA, pp. 8–11, Oct. 2006.

[MN96] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[Ooh97] Y. Oohama. Gaussian multiterminal source coding. *IEEE Trans. Inform. Theory*, vol. 43, No. 6, pp. 1912–1923, Nov. 1997.

[Ooh98] Y. Oohama. The rate-distortion function for the quadratic gaussian CEO problem. *IEEE Trans. Inform. Theory*, vol. 44, no. 3,pp. 1057–1070, May 1998.

[Pas76] R. C. Pasco. *Source Coding Algorithms for Fast Data Compression.* PhD thesis, Stanford University, CA, May 1976.

[PCR03] S. S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding with side information. *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 1181–1203, May 2003.

[PR] R. Puri and K. Ramchandran. PRISM : A video coding paradigm based on motion-compensated prediction at the decoder. *submitted IEEE Transactions on Image Processing.*

[PR99] S. S. Pradhan and K. Ramchandran. Distributed Source Coding Using Syndromes (DISCUS) : Design and construction. *Proc. IEEE Data Compression Conference (DCC)*, pp. 158–167, Mar. 1999.

[PR00] S. S. Pradhan and K. Ramchandran. Distributed source coding : Symmetric rates and applications to sensor networks. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, pp. 363–372, Mar. 2000.

[PR02] R. Puri and K. Ramchandran. PRISM : A new robust video coding architecture based on distributed compression principles. *Proc. 40th Allerton Conference on Communication, Control and Computing*, Allerton, Illinois, USA, Oct. 2002.

[PR03a] S. S. Pradhan and K. Ramchandran. Distributed Source Coding Using Syndromes (DISCUS) : Design and construction. *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[PR03b] R. Puri and K. Ramchandran. PRISM : A new "reversed" multimedia coding paradigm. *Proc. IEEE Intl. conference on Image Processing (ICIP)*, Sept. 2003.

[PR03c] R. Puri and K. Ramchandran. PRISM : A video coding architecture based on distributed compression principles. *ERL Technical Report*, University of California, Berkeley, USA, Mar. 2003.

[PTR04] V. Prabhakaran, D. Tse, and K. Ramchandran. Rate region of the quadratic gaussian CEO problem. *Proc. IEEE Intl. Symp. on Inf. Theory*, Jun. 2004.

[Pur02] R. Puri. *Robust Multimedia Coding : Information Theory and Practical Architectures*. PhD thesis, University of California, Berkeley, Fall 2002.

[RAG04] S. Rane, A. Aaron, and B. Girod. Systematic lossy forward error protection for error-resilient digital video broadcasting. *Proc. SPIE Visual Communications and Image Processing Conference*, San Jose, California, USA, Jan. 2004.

[Ris76] J. J. Rissanen. Generalized Kraft inequality and arithmetic coding. Technical report, IBM J. Res. Develop., vol. 20, pp. 198–203, 1976.

[RL79] J. J. Rissanen and G. Langdom. Arithmetic coding. Technical report, IBM J. Res. Develop., vol. 23, pp. 149–162, 1979.

[RMZG03] D. Rebollo-Monedero, R. Zhang, and B. Girod. Design of optimal quantizers for distributed source coding. *Proc. IEEE Data Compression Conference (DCC)*, pp. 13–22, Mar. 2003.

[Rob94] P. Robertson. Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes. *IEEE Int Conf. Universal Personal Commun.*, San Diego, USA, pp. 183–187, 1994.

[SA03] A. Sehgal and N. Ahuja. Robust predictive coding and the Wyner-Ziv problem. *Proc. IEEE Data Compression Conference (DCC)*, pp. 103-112, Mar. 2003.

[Ser00] S. Servetto. Lattice quantization with side information. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2000.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, vol .27, pp. 379–423 and 623–656, Jul. and Oct. 1948.

[SW73] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, Mar. 1973.

[TGFZ03] T. Tian, J. Garcia-Frias, and W. Zhong. Compression of correlated sources using LDPC codes. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, Utah, Mar. 2003.

[Tun78] S. Y. Tung. Multiterminal source coding. *Ph.D. Thesis, School of Electrical Engineering*, Cornell University, Ithaca, NY, May 1978.

[Ung82] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[VB97] H. Viswanathan and T. Berger. The quadratic gaussian CEO problem. *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1549–1559, Sept. 1997.

[Vit67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, Vol. IT-13, pp. 260–269, Apr. 1967.

[VY00] B. Vucetic and J. Yuan. *Turbo Codes : Principles and applications*. Kluwer Academic Publishers, Boston, May 2000.

[WM92] H. S. Wang and N. Moayeri. Trellis coded vector quantization. *IEEE Trans. on Commun.*, vol. 40, pp. 1273–1276, Aug. 1992.

[WNC87] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, vol. 30, no. 6, Jun. 1987.

[WNC98] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding revisited. *ACM Trans. on Information Systems*, vol. 16, no. 3, pp. 256–294, Jul. 1998.

[WTV05] A. B. Wagner, S. Tavildar, and P. Viswanath. The rate region of the quadratic gaussian two-terminal source-coding problem. *eprint arXiv :cs/0510095*, Oct. 2005.

[WZ76] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[XLCL03] Z. Xiong, A. Liveris, S. Cheng, and Z. Liu. Nested quantization and Slepian-Wolf coding : A Wyner-Ziv coding paradigm for *i.i.d.* sources. *Proc. IEEE Workshop on Statistical Signal Processing*, St. Louis, MO, Sept. 2003.

[YCXZ03] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao. Wyner-Ziv coding based on TCQ and LDPC codes. *Proc. Asilomar Conference on Signals, Systems and Computer*, Pacific Grove, CA, Nov. 2003.

[YSXZ04] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao. Asymmetric code design for remote multiterminal source coding. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2004.

[YSXZ05] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao. On multiterminal source code design. *Proc. IEEE Data Compression Conference (DCC)*, Snowbird, UT, Mar. 2005.

[ZAG03] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. *IEEE Workshop on Statistical Signal Processing*, pp. 30-33, St. Louis, Missouri, USA, Sept. 2003.

[ZE01] Q. Zhao and M. Effros. Optimal code design for lossless and near lossless source coding in multiple access networks. *Proc. IEEE Data Compression Conference (DCC)*, pp. 263–272, Mar. 2001.

[ZS98] R. Zamir and S. Shamai. Nested linear/lattice codes for Wyner-Ziv encoding. *Proc. IEEE Information Theory Workshop*, Killarney, Ireland, pp. 92–93, Jun. 1998.

[ZSE02] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–1276, Jun. 2002.

Table des figures

Résumé

Le codage de sources distribuées concerne le cas de signaux fortement corrélés que l'on code séparément et décode conjointement. Ce genre de technique peut s'appliquer à des réseaux de capteurs mais également au codage vidéo. En particulier, le codage de sources distribuées a été récemment étudié comme solution potentielle pour la compression de l'information dans des applications exigeant des encodeurs simples. D'un point de vue théorique, le codage de sources distribuées s'appuie principalement sur le théorème de Slepian-Wolf établi en 1973.

Dans cette thèse, nous nous intéressons au schéma de codage de sources distribuées utilisant les codes turbo poinçonnés. Dans ce contexte, pour améliorer les performances débit-distorsion, nous proposons d'utiliser la quantification codée par treillis. Nous considérons également les problèmes de codage avec information de bord de trois sources binaires ou Gaussiennes. Les limites théoriques de ces problèmes sont dérivées. Des schémas de mise en œuvre sont proposés et des résultats de simulation sont présentés et analysés pour différentes valeurs de corrélation. Enfin, nous développons et implémentons un schéma de codage vidéo distribué dans le domaine transformé. En particulier, une approche d'optimisation débit-distorsion est proposée.

Abstract

Distributed source coding is a general framework which applies to highly correlated signals that are coded separately and decoded jointly. This framework applies to sensor networks but also to video compression. In the latter application, the motivation is to reduce the complexity of the encoder, at the expense of an increase of complexity of the decoder. From a theoretical point of view, distributed source coding finds its foundations principally in the Slepian-Wolf theorem established in 1973.

In this thesis, we focus on the design of distributed source coding set-up using punctured turbo codes. In this context, to improve the rate-distortion performance, we propose to use the trellis-coded quantization. We also consider the problems of coding with side information of three binary or Gaussian sources. The rate bounds for these problems are derived. Practical solutions are provided and simulation results are presented and analyzed for different amounts of correlation. Finally, we develop and implement distributed video coding solutions in the transform domain. In particular, a rate-distortion optimization approach is proposed.