Interprétation et amélioration d'une procédure de démodulation itérative

Ziad Naja

N° d'ordre: 10204

THÈSE

Présentée pour obtenir

LE GRADE DE DOCTEUR EN SCIENCES

DE L'UNIVERSITÉ PARIS-SUD XI

Spécialité: Traitement du signal pour les télécommunications

par

Ziad NAJA

Interprétation et amélioration d'une procédure de démodulation itérative

Soutenue le 1^{er} Avril 2011 devant la Commission d'examen:

M.	Bernard FLEURY	(Rapporteur)
M.	Charly POULLIAT	(Rapporteur)
M.	Gerald MATZ	(Examinateur)
M.	Phillip REGALIA	(Président du jury)
Mme	Florence ALBERGE	(co-Directeur de thèse)
M.	Pierre DUHAMEL	(Directeur de thèse)

Thèse préparée au

Laboratoire des Signaux et Systèmes

Supelec

3, rue Joliot Curie

91190, Gif-sur-Yvette

Résumé

La géométrie de l'information est la théorie mathématique qui applique les méthodes de la géométrie différentielle dans le domaine des statistiques et de la théorie de l'information. C'est une technique très prometteuse pour l'analyse et l'illustration des algorithmes itératifs utilisés en communications numériques. Cette thèse porte sur l'application de cette technique ainsi que d'autre technique d'optimisation bien connue, l'algorithme itératif du point proximal, sur les algorithmes itératifs en général. Nous avons ainsi trouvé des interprétations géométriques (basée sur la géométrie de l'information) et proximales (basée sur l'algorithme du point proximal) intéressantes dans le cas d'un algorithme itératif de calcul de la capacité des canaux discrets sans mémoire, l'algorithme de Blahut-Arimoto. L'idée étant d'étendre cette application sur une classe d'algorithmes itératifs plus complexes. Nous avons ainsi choisi d'analyser l'algorithme de décodage itératif des modulations codées à bits entrelacés afin de trouver quelques interprétations et essayer de proposer des liens existant avec le critère optimal de maximum de vraisemblance et d'autres algorithmes bien connus dans le but d'apporter certaines améliorations par rapport au cas classique de cet algorithme, en particulier l'étude de la convergence.

Mots-clefs : Géométrie de l'information, algorithme du point proximal, algorithme de Blahut-Arimoto, décodage itératif, Modulations codées à bits entrelacés, maximum de vraisemblance.

Interpretation and amelioration of an iterative demodulation procedure

Abstract

Information geometry is a mathematical theory that applies methods of differential geometry in the fields of statistics and information theory. It is a very promising technique for analyzing iterative algorithms used in digital communications. In this thesis, we apply this technique, in addition to the proximal point algorithm, to iterative algorithms. First, we have found some geometrical and proximal point interpretations in the case of an iterative algorithm for computing the capacity of discrete and memoryless channel, the Blahut-Arimoto algorithm. Interesting results obtained motivated us to extend this application to a larger class of iterative algorithms. Then, we have studied in details iterative decoding algorithm of Bit Interleaved Coded Modulation (BICM) in order to analyse and propose some ameliorations of the classical decoding case. We propose a proximal point interpretation of this iterative process and find the link with some well known decoding algorithms, the Maximum likelihood decoding.

Keywords : Information geometry, Proximal point algorithm, Blahut-Arimoto algorithm, BICM-ID, ML.

Remerciements

Mes premiers hommages vont à celui pour lequel cette page est bien loin de suffire à exprimer toute ma reconnaissance. Pierre, je te remercie pour m'avoir accueilli il y a maintenant plus de trois ans, pour m'avoir accordé ta confiance et patiemment écouté surtout lorsque j'ai commencé à avancer dans le long chemin de la recherche avec des pas incertains et hésitants. Je te suis reconnaissant de m'avoir transmis ton goût de la recherche, ta passion de l'enseignement et ta manière de diriger une équipe et une division et de gérer des contrats. Bien plus qu'un directeur de thèse, tu as dépassé le rôle d'un tuteur pour être un ami sincère. Chercheur et pédagogue exemplaire, j'ai beaucoup appris à ton contact, tout en prenant un très grand plaisir à être ton cinquantième thésard. Je te témoigne donc ici ma profonde gratitude.

Je tiens ensuite à exprimer ma profonde reconnaissance à Florence pour m'avoir assuré les bonnes conditions pour le meilleur déroulement de la thèse. Ses connaissances à la fois vastes et pointues, combinées avec son raisonnement rationnel, m'ont garanti un support solide tout au long de cette thèse. A chaque fois je me sentais désorienté, elle était toujours présente pour me mettre sur les rails.

J'adresse également mes remerciements aux membres de Jury qui m'ont fait l'honneur de valider ce travail. Merci à Prof. Bernard FLEURY et Prof. Charly POULLIAT pour avoir eu la volonté de relire et de commenter mon manuscrit. Je tiens à remercier également Prof. Gerald MATZ d'avoir accepté de faire partie de mon Jury de thèse en tant qu'examinateur. Merci également à Prof. Phillip REGALIA qui m'a fait l'honneur d'être examinateur et président du Jury et de se déplacer de très loin spécialement pour ma soutenance.

Je tiens à remercier l'Université Paris-Sud 11, le laboratoire des signaux et systèmes (L2S) et Supélec qui ont mis à ma disposition tous les moyens indispensables pour mener ce travail avec succès.

Je tiens également à remercier mes professeurs de l'université libanaise, faculté de génie, branche 1 qui étaient et resteront une source continue de conseil et de support.

Cette thèse fut une expérience très enrichissante aussi bien sur le plan professionnel que personnel. Elle m'a permis en particulier de faire la connaissance de plusieurs amis avec qui j'ai passé des moments très agréables. A vous toutes et tous, ainsi qu'à mes anciens amis, j'adresse les expressions de mes vifs remerciements et de ma gratitude pour votre encouragement et votre support moral en vous souhaitant beaucoup de succès. J'adresse également mes remerciements à tous mes compatriotes pour les bons moments passés ensemble. Un grand merci à Houmam et sa famille pour les agréables discussions qu'on a eues ensemble autour des plats délicieux préparés par sa femme. Enfin, je ne peux pas oublier Hacheme avec qui j'ai passé ces années de thèse en confrontant nos idées sur la science et le monde en général. J'apprécierai pour toujours son inestimable et incomparable amitié.

Pendant ce travail, j'ai collaboré avec plusieurs professeurs et collègues auxquels j'ai un grand respect pour leurs connaissances, expertises et professionnalisme surtout dans le cadre du réseau d'excellence Européen Newcom++ (Network of Excellence in Wireless Communications).

Mes remerciements s'adressent également aux membres du laboratoire L2S dans lequel j'ai vécu une expérience très enrichissante grâce à sa diversité culturelle et à l'ambiance de convivialité et de respect qui y règne. Je remercie Silviu-Iulian NICULESCU, directeur du laboratoire, et tout le personnel administratif et de gestion pour leur gentillesse et serviabilité. Un grand merci également aux permanents du laboratoire qui ont dépassé leur rôle de permanants pour être des vrais amis. Je n'oublierai jamais mes discussions avec Aurélia, Alex, Claude, Patrice, Rémy, Thomas...

Mes remerciements, ma gratitude, ma reconnaissance, tous associés et c'est encore peu pour mes chers parents Hanan et Salem, à qui je dois tout après Dieu, pour les sacrifices et le dévouement qu'ils ont fait pour moi ainsi que pour mes soeurs et frère pour nous fournir un bon niveau d'instruction et d'éducation. Je leur suis infiniment reconnaissant pour leur amour, leur soutien et leur encouragement à être le meilleur. Qu'ils trouvent aussi le fruit de leur travail ... A vous deux, je dédie ce mémoire !

Je tiens à exprimer également mes remerciements et gratitude à mes chères soeurs, Hana et son mari Fadi, Sana et son mari Nazih, Dania et son mari Hayssam pour leur

encouragement et support moral en leur souhaitant plein de succès dans leur vie. Un remerciement particulier est dédié à mon frère Adnan et sa femme Farah pour m'avoir vivement encouragé durant ma thèse et pour m'avoir souvent gâté par leurs petites attentions particulières. Grâce à son soutien salvateur dans certains moments, il m'a fait oublier que le Liban est si loin ! !

Je réserve ma dernière mais **spéciale** mention à ma future femme Sara, qui était et restera pour toujours *mon rayon de soleil* et *ma source d'espoir* ! ! !

Table des matières

1

Introduction générale

La géométrie de l'information est la structure géométrique naturelle différentiable que possèdent les familles de distributions de probabilité : chaque distribution de probabilité est considérée comme étant un point dans un espace. Cet espace n'a pas les propriétés d'un espace euclidien, cependant il est possible de définir des pseudo-distances en termes de divergence de Kullback-Leibler. Cette technique s'est avérée assez pertinente pour l'analyse et l'illustration des algorithmes itératifs. Dans cette thèse, la technique géométrique ainsi que celle de l'algorithme itératif du point proximal sont appliquées sur les algorithmes itératifs en général. L'optimum exact d'une fonction est en général impossible à calculer. L'algorithme du point proximal permet de calculer une valeur approchée de cet optimum d'une manière itérative. Depuis son introduction en 1970, divers travaux ont été entrepris concernant l'étude de la convergence de cet algorithme. L'algorithme de Blahut Arimoto est un algorithme itératif permettant le calcul de la capacité des canaux discrets sans mémoire. Notre travail apporte des interprétations géo-

métriques (basée sur la géométrie de l'information) et proximales (basée sur l'algorithme du point proximal) de cet algorithme. Ces interprétations se sont révélées assez intéressantes. Une seconde partie de ce travail est consacré à l'extension de ces techniques sur une classe d'algorithmes itératifs plus compliqués tels l'algorithme de décodage itératif des modulations codées à bits entrelacés dans le but d'apporter certaines améliorations par rapport au cas classique de ces algorithmes tout en essayant de généraliser ici les résultats obtenus dans le cas de l'algorithme de Blahut-Arimoto.

La présentation de ce mémoire s'articule de la façon suivante :

- Dans le chapitre 2, nous présentons les notions de base de la géométrie de l'information, une des techniques les plus efficaces pour l'analyse et l'illustration des algorithmes itératifs tout en montrant les outils mathématiques utilisés. La notion de projection d'une loi de probabilité sur une famille de distribution de probabilité est mise en évidence. Une famille de probabilité est un espace dans lequel une distribution de probabilité spécifique est un point caractérisé par ses coordonnées dans cet espace. Cette projection est présentée comme étant une minimisation d'une divergence de Kullback-Leibler bien définie. L'inégalité triangulaire est ensuite décrite (égalité (inégalité) de Pythagore). Enfin, l'accent est mis sur un algorithme de projections alternées (algorithme de Csizar) tout en étudiant les conditions de sa convergence.
- Le chapitre 3 est basé sur l'algorithme du point proximal. Pour certaines fonctions, il n'est généralement pas possible de calculer le minimum exact. L'algorithme du point proximal permet de calculer itérativement cet optimum. Notons que cet algorithme est initialement introduit avec comme terme de pénalité la norme euclidienne. Ce terme de pénalité a pour rôle d'assurer que la nouvelle valeur du paramètre reste dans le voisinage de la valeur obtenue à l'itération précédente. Dans ce chapitre, une extension de cet algorithme est proposée généralisant ainsi une approche existante qui considère le cas où le terme de pénalité est une divergence de Kullback. Nous montrons que la propriété de convergence superlinéaire peut être étendue à ce cas particulier du terme de pénalité.
- Dans le chapitre 4, nous proposons deux classes d'algorithmes itératifs pour le calcul de la capacité d'un canal arbitraire discret sans mémoire. Nous montrons ainsi que l'algorithme classique de Blahut-Arimoto (BA) est un cas particulier de

notre étude. La formulation de ces algorithmes est basée sur une approche gradient naturel et la méthode du point proximal. En plus, des interprétations basées sur la géométrie de l'information sont proposées mettant en évidence des projections de lois de probabilités sur des familles linéaires et exponentielles de probabilité bien précises. Enfin, une analyse théorique de la convergence ainsi que des résultats de simulation montrent que nos deux nouveaux algorithmes apportent un bonus par rapport au cas classique notamment en ce qui concerne l'accélération de la vitesse de convergence.

Ce chapitre est un résumé des résultats obtenus et soumis à IEEE Transactions on Information Theory présentés en annexe A.

– Dans le chapitre 5, nous entrons plus en détails dans l'application de la méthode du point proximal dans l'étude de décodage itératif des BICM. La structure classique des BICM est rappelée et les notations sont introduites. Plusieurs reformulations et interprétations (basées sur la géométrie de l'information) du problème de décodage itératif sont proposées dans le but de trouver une interprétation "point proximal" et profiter de quelques propriétés importantes de cette méthode quant à la nature de convergence (convergence super linéaire). Une interprétation point proximal par blocs séparés est proposée (interprétation point proximale de chacun des blocs de démodulateur et de décodeur pris séparément) assurant la diminution d'un critère bien précis au fil des itérations mais ne garantissant pas la convergence. Des conclusions sont tirées quant à la différence entre cette approche proximale et le cas itératif classique.
– Le chapitre 6 porte sur le lien entre le décodage itératif des BICM et le décodage par maximum de vraisemblance afin de compléter les résultats déjà existant dans la littérature et proposer certaines applications intéressantes de ces algorithmes itératifs. Nous montrons qu'en partant du critère de maximum de vraisemblance nous pouvons retrouver les équations classiques du décodage itératif. Une approximation est cependant nécessaire pour établir le lien et obtenir une solution qui pourra être calculée analytiquement. Nous montrons aussi que le décodage turbo peut être obtenu à partir d'une implémentation hybride, de type Jacobian/Gauss-Seidel, du processus de maximisation. La propagation des extrinsèques est naturellement introduite, c'est une conséquence directe de la mise à jour. Enfin, une partie de

simulation montre une application possible des résultats obtenus qui sont publiés dans la conférence ICASSP 2011. L'article correspondant est mis en annexe B.

– Le chapitre 7 est dédié à la conclusion générale de nos travaux tout en posant les perspectives.

2

Géométrie de l'information

2.1 Définition et historique

2.1.1 Définition

La géométrie de l'information concerne l'application de la géométrie différentielle à des familles de distributions de probabilité et donc à des modèles statistiques. C'est la structure géométrique naturelle différentiable que possèdent les familles de distributions de probabilité [AN00].

Deux types d'information jouent un rôle important dans ce domaine :

– La divergence de Kullback Leibler ou entropie relative

– L'information de Fisher qui prend en compte la courbure

2.1.2 Intérêts

La géométrie de l'information a été appliquée dans l'étude des turbo codes et des codes LDPC [ITA04]. L'idée principale de la géométrie de l'information est que les distributions de probabilité sont considérées comme des points dans un espace. Cet espace n'a pas les propriétés d'un espace euclidien, cependant il est possible de définir des distances, en termes de divergence de Kullback-Leibler (KL).
Parmi les concepts de la géométrie de l'information, on peut citer la projection, qui cherche le point le plus proche dans un sous-espace par rapport à un point fixe dans l'espace. Il existe dans la littérature un algorithme itératif simple appelé processus de minimisation alternée [CT84] utilisé pour résoudre ce problème. Nous verrons, dans les chapitres suivants, certaines applications de la géométrie de l'information dans l'analyse de certains algorithmes itératifs.

2.2 Outils de base

2.2.1 Divergence de Kullback-Leibler

Définition

La divergence de Kullback-Leibler ou entropie relative mesure "la distance" entre deux distributions de probabilité définies sur le même domaine. C'est un outil fondamental utilisé notamment en statistique et en apprentissage.

Considérons deux distributions de probabilité $p = \{p(x), x \in \boldsymbol{X}\}$ et $q = \{q(x), x \in \boldsymbol{X}\}$ d'une variable aléatoire $\mathbf{X}$ prenant ses valeurs x dans un ensemble discret $\boldsymbol{X}$. La divergence de Kullback-Leibler entre ces deux distributions de probabilité p et q peut être définie comme suit : [CT91, Gal68]

$$D(p||q) = \sum_{\mathbf{x}\in\boldsymbol{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{2.1}$$

Propriétés

La divergence de Kullback, appelée aussi entropie relative possède quelques propriétés d'une métrique :

- $D(p||q)$ est toujours non négative : $D(p||q) \geq 0$
- $D(p||q) = 0$ si et seulement si $p \equiv q$

Cependant, cette divergence n'est pas une vraie distance entre distributions car elle ne possède pas la propriété de symétrie. En effet $D(p||q) \neq D(q||p)$ et ne satisfait pas en général l'inégalité triangulaire.
Par abus de langage, l'entropie relative est nommée distance de Kullback-Leibler entre distributions de probabilité.

2.2.2 Familles de distribution de probabilité

Une famille de probabilité est un espace dans lequel une distribution de probabilité spécifique est un point caractérisé par ses coordonnées dans cet espace. On distingue plusieurs types de familles dont les plus utiles dans la suite sont :

- Famille linéaire de probabilité
- Famille exponentielle de probabilité

Famille linéaire de probabilité

- **Définition** :
  Une famille linéaire de probabilité est définie comme [CT84] :
  $\forall f_1, f_2, \ldots, f_K \in \mathbf{X}$ et $\forall \alpha_1, \alpha_2, \ldots, \alpha_K$

$$\mathcal{L} = \{p : \mathbb{E}_p(f_i(x)) = \alpha_i, 1 \leq i \leq K\} \tag{2.2}$$

  La valeur de l'espérance $\mathbb{E}_p(f_i(x))$ de la variable aléatoire x par rapport à la distribution $p(x)$ est restreinte à α_i.
  Une famille linéaire de probabilité est complètement définie par $\{f_i(x)\}_{1 \leq i \leq K}$ et $\{\alpha_i\}_{1 \leq i \leq K}$.

Le vecteur $\alpha = [\alpha_1, \ldots, \alpha_k]$ sert de système de coordonnées dans l'espace de la famille linéaire, ces coordonnées sont appelées "coordonnées mixtes".

- **Propriétés** : Etant donné $p_1(x)$,$p_2(x) \in \mathcal{L}$, et $0 \leq t \leq 1$,

$$p(x;t) = (1-t)p_1(x) + tp_2(x) \in \mathcal{L} \tag{2.3}$$

En effet

- $p_1(x) \in \mathcal{L}$, il en suit que $\mathbb{E}_{p_1}(f(x)) = \alpha$
- $p_2(x) \in \mathcal{L}$, il en suit que $\mathbb{E}_{p_2}(f(x)) = \alpha$
- $\mathbb{E}_p(f(x)) = (1-t)\mathbb{E}_{p_1}(f(x)) + t\mathbb{E}_{p_2}(f(x)) = (1-t)\alpha + t\alpha = \alpha$. D'où $p(x;t) \in \mathcal{L}$.

D'autres propriétés liées à la projection sur ces familles linéaires seront mises en évidence dans la suite.

- **Exemples** :
  - Une famille linéaire importante dans le contexte du décodage souple est la famille de distributions compatibles avec le code :

$$\mathcal{L}_C = \{p : \mathbb{E}_p((f(x))) = 0\}$$

Ici, les fonctions f représentent la structure du code et $f_i(x) = 0$ les équations de parité de code

  - Un autre exemple est la famille des distributions discrètes de probabilité avec $f_i(x) = \delta_i(x)$, on a ainsi :

$$\mathbb{E}_p(f_i(x)) = \sum_x p(x)\delta_i(x) = p(i) = p_i = \alpha_i \tag{2.4}$$

Famille exponentielle de probabilité

- **Définition** :

Une famille exponentielle de distributions discrètes de probabilité $p(x)$ dans un alphabet X est l'ensemble

$$\mathcal{E} = \{p : p(x) = \frac{Q(x)exp\sum_{i=1}^{K}(\theta_i f_i(x))}{\sum_x (Q(x)exp\sum_{i=1}^{K}(\theta_i f_i(x)))}\} \tag{2.5}$$

$\mathcal{E}$ est complètement définie par $f_i(x)$ et $Q(x)$ et paramétrée par θ_i

– **Propriétés** : Etant donné $p_1(x)$, $p_2(x) \in \mathcal{E}$, et 0 ≤ t ≤ 1, la combinaison log-convexe normalisée

$$p(x;t) = Cp_1^{(1-t)}(x)p_2^t(x) \in \mathcal{E} \tag{2.6}$$

où C est une constante de normalisation.

D'autres propriétés liées à la projection sur ces familles exponentielles seront citées dans la suite.

– **Exemples** :

– Un exemple de famille exponentielle jouant un rôle important dans le contexte du décodage souple est la famille des densités factorisables :

$$\mathcal{E}_F = \{p(x) : p(x) = \prod_{i=1}^{K} p_i(x_i)\} \tag{2.7}$$

– On pourra aussi montrer que l'ensemble des distributions discrètes de probabilité $p(x)$ où x ∈ $X = \{0, 1, \ldots, 2^K - 1\}$, est lui-même une famille exponentielle de probabilité.

Notion de projection sur une famille de probabilité

– **Définition** :

Etant donné un a priori p et un espace $\mathcal{L}$, la I-projection de p sur l'espace $\mathcal{L}$ [Csi75] est la distribution p^* appartenant à $\mathcal{L}$ et minimisant la distance de Kullback-Leibler entre p et n'importe quelle autre distribution de probabilité q appartenant à $\mathcal{L}$ et pourra être reformulée par le problème d'optimisation suivant :

$$p^* = argmin_{q \in \mathcal{L}} D(q||p) \tag{2.8}$$

On pourra définir aussi une rI-projection de p sur une famille $\mathbb{Q}$ de la manière suivante :

$$q^* = argmin_{q \in \mathbb{Q}} D(p||q) \tag{2.9}$$

q^* est la rI-projection de p sur la famille $\mathbb{Q}$

Considérons l'exemple de projection d'un a priori p sur une famille linéaire de probabilité $\mathcal{L}$ définie comme suit :

$$\mathcal{L} = \{q(x) : \mathbb{E}_q(T_i(x)) = b_i \qquad i = 1, \ldots, K\} \tag{2.10}$$

où $q \in \mathcal{P}$: ensemble des distributions normalisées de probabilité.

La projection de la loi p sur cette famille linéaire $\mathcal{L}$ pourra être reformulée comme suit :

$$\min_{q \in \mathcal{L}} D(q||p) \tag{2.11}$$

et aura comme solution :

$$p^*(x) = Cp(x) \exp(\sum_i \lambda_i T_i(x)) \tag{2.12}$$

où C est une constante de normalisation de p^*

Notons que p^* appartient aussi à une famille exponentielle de probabilité $\mathcal{E}_p$ générée par p, on pourra alors dire que p^* appartient à l'intersection des deux familles de probabilité : la famille exponentielle $\mathcal{E}_p$ générée par p et T_i et celle linéaire $\mathcal{L}$ définie par les fonctions $T_i(x)$ et les constantes b_i ($p^* \in \mathcal{L} \bigcap \mathcal{E}_p$)

Un exemple de rI-projection sera la reverse I-projection sur la famille exponentielles des densités factorisables de probabilité $\mathcal{E}_F$

$$\min_{q \in \mathcal{E}_F} D(p||q)$$

où

$$\mathcal{E}_F = \{q(x) = \textstyle\prod_{i=1}^{K} q_i(x_i)\}$$

et

$$q \in \mathcal{P} \text{ : ensemble des distributions normalisées de probabilité}$$

La solution de ce problème de minimisation aura l'expression suivante :

$$p^*(x) = \textstyle\prod_i p_i(x_i) = \prod_i (\sum_{x \sim x_i} p(x))$$

qui pourra être interprétée comme étant la marginalisation de p(x)

– **Egalité de Pythagore**

Vu que la projection a lieu sur une famille linéaire de probabilité, on pourra montrer qu'elle vérifie l'égalité de Pythagore suivante :

$$D(q||p) = D(q||p^*) + D(p^*||p) \tag{2.13}$$

En effet, on a vu que la solution p^* du problème de projection a comme expression : $p^*(x) = Cp(x)\exp(\sum_i \lambda_i T_i(x))$, et d'autre part la divergence de Kullback entre les deux distributions p^* et p pourra s'écrire comme suit :

$$\begin{aligned} D(p^*||p) &= \textstyle\sum_x (Cp(x)\exp(\sum_i \lambda_i T_i(x)))\log(C\exp(\sum_i \lambda_i T_i(x))) \\ &= \log C + \textstyle\sum_i \lambda_i T_i(x) \end{aligned} \tag{2.14}$$

En se basant sur ces expressions, on pourra developper la divergence de Kullback entre les deux distributions de probabilité q et p de la manière suivante :

$$\begin{aligned} D(q||p) = \textstyle\sum_x q(x)\log\frac{q(x)}{p(x)} = \sum_x q(x)\log\frac{q(x)p^*(x)}{p(x)p^*(x)} &= D(q||p^*) + \log C + \textstyle\sum_i \lambda_i T_i(x) \\ &= D(q||p^*) + D(p^*||p) \end{aligned} \tag{2.15}$$

On retrouvera donc l'égalité de Pythagore dans (2.13)

La notion de projection étant définie, nous pouvons donc citer les propriétés suivantes :

1. La I-projection p^* de q sur une famille linéaire $\mathcal{L}$ est unique et satisfait l'égalité de Pythagore

$$D(p||q) = D(p||p^*) + D(p^*||q) \qquad \forall p \in \mathcal{L} \tag{2.16}$$

2. La I-projection p^* de q sur une famille exponentielle de probabilité $\mathcal{E}$ est unique et satisfait l'inégalité de Pythagore

$$D(p||q) \geq D(p||p^*) + D(p^*||q) \qquad \forall p \in \mathcal{E} \tag{2.17}$$

3. La "reverse" I-projection q^* de p sur une famille exponentielle de probabilité

$\mathcal{E}$ est unique et satisfait l'égalité de Pythagore :

$$D(p||q) = D(p||p^*) + D(p^*||q) \qquad \forall q \in \mathcal{E} \tag{2.18}$$

– **Algorithme de projections alternées de Csiszar** :

Le but de cet algorithme est de projeter un a priori p sur un espace affine vérifiant un ensemble de contrainte et définit comme suit :

$$L = \{q : \mathbb{E}_q(T_i(x)) = b_i, i = 1, 2, \cdots, k\} \tag{2.19}$$

Pour cela, l'idée étant de considérer une de ces k contraintes (soit L_i la ième contrainte) :

$$L_i = \{q : \mathbb{E}_q(T_i(x)) = b_i\} \tag{2.20}$$

et l'algorithme itératif de Csiszar sera définit comme suit :

– Initialisation : p_0=p

– Itérer p_{t+1}=I-projection de p_t sur $L_{\text{t mod k}}$ jusqu'à la convergence

Pour illustrer cet algorithme, on peut considérer le cas d'un espace correspondant à la dimension 1 dans lequel on a deux familles L_1 et L_2 et un a priori p, nous verrons dans ce cas que $p^* = \{L_1 \cap L_2\}$. Cela est illustré dans la figure (2.1) : Notons qu'une projection alternée pourra être vu, dans ce cas particulier, comme

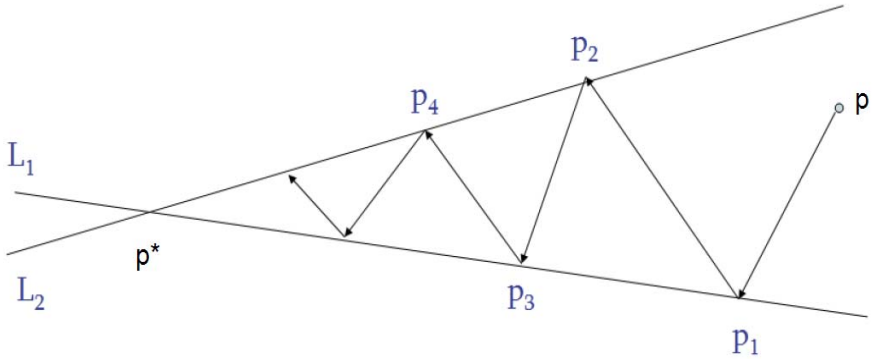

FIG. 2.1 – Algorithme de projections alternées de Csizar.

étant une série de I-projection.

– **Preuve de convergence** :

Utilisant le théorème de Pythagore, on s'approche de plus en plus de p^* d'une

itération à une autre et cela selon l'égalité suivante :

$$D(p^*||p_{t+1}) = D(p^*||p_t) - D(p_{t+1}||p_t) \tag{2.21}$$

Cela peut être illustré dans la figure (2.2) :

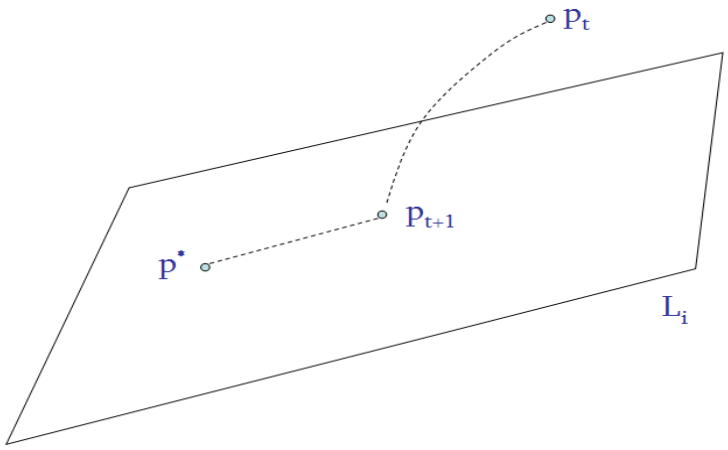

FIG. 2.2 – Preuve de convergence : théorème de Pythagore

Quant au cas général de l'algorithme de projections alternées de Csiszar, on pourra se servir du théorème suivant [CT84] :

$$Si \bigcap_{i=1}^{k} L_i = L \neq \emptyset, alors \ \ p_n \rightarrow p^* quand \ \ n \rightarrow \infty \tag{2.22}$$

où p^* est la I-projection de p sur L

Un cas encore plus général qui ne s'applique pas seulement aux I-projections consiste à résoudre le problème fondamental de faisabilité convexe suivant [BBC04] :

$$Trouver \ \ x \in \bigcap_{i=1}^{k} C_i \tag{2.23}$$

où $(C_i)_i$ sont des ensembles finis **convexes** vérifiant $\bigcap_{i=1}^{k} C_i \neq \emptyset$

Des résultats liés à la convergence de cet algorithme de projections alternées figurent dans [LM08] où les auteurs montrent la convergence de cet algorithme vis à vis de l'angle entre les espaces sur lesquels on projette les distributions de probabilité.

Plusieurs études ont été basées sur ces techniques de la géométrie de l'information afin de proposer certaines interprétations de quelques algorithmes itératifs connus dans le but de proposer des améliorations considérables vis à vis de ces algorithmes.
Parmi ces études, certaines ont considéré comme application les algorithmes de turbo décodage itératif et essayé grâce à ces techniques géométriques d'améliorer leur fonctionnement classique et trouver des liens intéressants avec d'autres algorithmes de décodage bien connus dans la littérature, notamment le décodage par Maximum a Posteriori et celui par Maximum de Vraisemblance [Ric00, ITA04, WRJ06, MDdC02, Muq01, AND11]. D'autres ont traité d'autres catégories d'algorithmes itératifs notamment l'algorithme de Blahut-Arimoto pour le calcul de la capacité des canaux discrets sans mémoire et ont utilisé ces techniques dans le but de trouver des interprétations géométriques pouvant aider à améliorer le fonctionnement classique de ces algorithmes [MD04, NAD09]. Beaucoup d'autres applications de cette technique ont été étudiées, cependant nous nous sommes concentrés dans cette thèse sur les applications concernant quelques algorithmes itératifs simples utilisés en communications numériques. Plus de détails concernant ces interprétations et ces divers applications seront donnés dans les chapitres qui suivent.

3

Algorithme du point proximal

3.1 Algorithme du point proximal

L'algorithme du point proximal a été introduit par Martinet en 1970 [Mar70] et d'importants résultats ont été publiés par Rockafellar en 1976 [Roc76]. Il s'agit d'un algorithme général pour trouver un zéro d'un opérateur monotone maximal T comme limite de la suite z_k produite par l'itération

$$z_{k+1} = (I + c_k T)^{-1}(z_k) \tag{3.1}$$

Un exemple d'opérateur monotone maximal étant le sous-différentielle d'une fonction f, convexe, propre, semi-continue inférieurement. Pour minimiser une fonction en trouvant un zéro de son gradient, on peut donc utiliser l'algorithme de point proximal. On peut

alors montrer que

$$z_{k+1} = Argmin_{z \in R^n}\{f(z) + \frac{1}{2c_k} \parallel z - z_k \parallel^2\} \tag{3.2}$$

En effet, à partir de (3.2) nous pouvons écrire :

$$\frac{\partial f(z_{k+1})}{\partial z} + \frac{1}{c_k}(z_{k+1} - z_k) = 0 \tag{3.3}$$

Nous pouvons donc conclure que la solution z_{k+1} de ce problème de minimisation vérifie l'équation suivante :

$$(I + c_k \frac{\partial f}{\partial z})(z_{k+1}) = z_k \tag{3.4}$$

Nous retrouvons ainsi la solution (3.1) dans ce cas particulier où l'opérateur T n'est autre que le sous-différentielle (∂) de la fonction f.

Pour une fonction f quelconque, il n'est généralement pas possible de calculer le minimum exact dans (3.2). La méthode du point proximal est donc utilisée afin de trouver ce minimum d'une manière itérative. Rockafellar [Roc76], Auslender [Aus87] et Correa-Lemaréchal [CL93] ont étudié la convergence de l'algorithme du point proximal lorsque ce minimum est seulement calculé d'une manière itérative.

Pour cela, nous allons voir ici quelques définitions liées à la nature de convergence d'une suite numérique

3.1.1 Etude de convergence

Convergence super-linéaire

Considérons une suite numérique x_n qui converge vers une valeur x^*, on a donc :

$$\lim_{n \to \infty} x_n = x^* \tag{3.5}$$

$$\lim_{n \to \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = \mu \tag{3.6}$$

- S'il existe μ telle que : $0 < \mu < 1$, la convergence est linéaire. μ étant la vitesse de convergence.
- Si $\mu = 0$, dans ce cas la convergence est dite super-linéaire.
- On dit que la suite de limite x^* est convergente d'ordre q ($q > 1$), s'il existe $\mu > 0$ telle que :

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^q} = \mu \tag{3.7}$$

  - $q = 2$: convergence quadratique
  - $q = 3$: convergence cubique
  - $q = 4$: convergence quartique
  - ...

En se basant sur ces définitions générales, Rockafellar [Roc76] a établi les conditions que doit vérifier la fonction f ainsi que l'opérateur T afin d'avoir une convergence super linéaire de l'algorithme du point proximal. En effet, à partir de l'équation (3.2), on pourra écrire que

$$z_{k+1} = P_k(z_k) \tag{3.8}$$

où $P_k = (I + c_k T)^{-1}$ et T étant l'opérateur de différentiel de la fonction f dans ce cas.

T et P_k doivent vérifier les deux conditions suivantes :

- T est fortement monotone avec comme module $\alpha > 0$:

$$< z - z', w - w' > \geq \alpha \parallel z - z' \parallel^2 \forall w \in T(z), w' \in T(z') \tag{3.9}$$

- P_k doit être un opérateur pseudo-contractant :

$$\parallel P(z) - P(z') \parallel \leq \parallel z - z' \parallel \tag{3.10}$$

A partir de ces deux propriétés, Rockafellar a pu montrer que l'opérateur $P'_k(z) = (1 + \alpha c_k)P_k(z)$ est aussi pseudo-contractant et on aura donc

$$\parallel P(z) - P(z') \parallel \leq (1 + \alpha c_k)^{-1} \parallel z - z' \parallel \forall z, z' \tag{3.11}$$

En particulier, cela implique que P_k a un point fixe unique qui devra être le point unique z^∞ telle que $0 \in T(z^\infty)$, cela se traduit par

$$\| z_{k+1} - z^\infty \| = \| P_k(z_k) - P_k(z^\infty) \| \leq (1 + \alpha c_k)^{-1} \| z_k - z^\infty \| \tag{3.12}$$

D'où, si $c_k \geq c > 0$, la séquence z_k converge vers la solution z^∞ d'une manière linéaire avec un coefficient $(1 + \alpha c)^{-1} < 1$.

Si en plus, $c_k \to \infty$, la convergence sera super-linéaire et on aura :

$$\lim_{k \to \infty} \frac{\|z_{k+1} - z^\infty\|}{\|z_k - z^\infty\|} = 0 \tag{3.13}$$

L'algorithme du point proximal peut être généralisé selon :

$$z_{k+1} = Argmin_{z \in R^n} \{ f(z) + \frac{1}{2c_k} P(z, z_k) \} \tag{3.14}$$

où $P(z, z_k)$ est toujours non négative et $P(z, z_k) = 0$ si et seulement si $z = z_k$. Dans la suite, nous utiliserons cette formulation en considérant pour P la divergence de Kullback, tout en sachant que le terme de pénalité assure que la nouvelle valeur du paramètre reste dans le voisinage de la valeur obtenue à l'itération précédente (pour que l'algorithme ne diverge pas).

3.1.2 Cas où le terme de pénalité est une divergence de Kullback-Leibler

Dans le cas où le terme de pénalité de l'algorithme de point proximal est une divergence de Kullback, on pourra formuler notre problème comme suit :

$$z^{k+1} = argmin_z \{ f(z) + \beta_k D(z, z^k) \} \tag{3.15}$$

Ici, nous allons essayer de généraliser les résultats de A. Hero [CH99].

On suppose que, pour n'importe quelle séquence de paramètres de relaxation β_k, la séquence z^k converge vers la solution z^*.

On suppose en plus que f(z) et $D(z, z^k)$ sont deux fois continuellement différentiables en (z, z^k).

On pourra alors écrire le développement de Taylor du gradient au voisinage de z^* comme suit :

$$\begin{aligned}\frac{\partial f(z)}{\partial z} + \beta_k \frac{\partial D(z,z^*)}{\partial z} = \frac{\partial f(z^*)}{\partial z} + \beta_k \frac{\partial D(z^*,z^*)}{\partial z} + \frac{\partial^2 f(z^*)}{\partial z^2}(z - z^*) + \beta_k \frac{\partial^2 D(z^*,z^*)}{\partial z^2}(z - z^*) \\ + R(z - z^*)\end{aligned} \tag{3.16}$$

Avec $\lim_{z \to z^*} \frac{\|R(z - z^*)\|}{\|z - z^*\|} = 0$

D'autre part, nous avons :

$$\frac{\partial f(z^*)}{\partial z} = 0 \tag{3.17}$$

et, d'après la définition de la divergence de Kullback :

$$D(z^*, z^*) = 0 \tag{3.18}$$

On aura donc

$$\frac{\partial f(z)}{\partial z} + \beta_k \frac{\partial D(z, z^*)}{\partial z} = \frac{\partial^2 f(z^*)}{\partial z^2}(z - z^*) + \beta_k \frac{\partial^2 D(z^*, z^*)}{\partial z^2}(z - z^*) + R(z - z^*) \tag{3.19}$$

or

$$\frac{\partial f(z^{(k+1)})}{\partial z} + \beta_k \frac{\partial D(z^{(k+1)}, z^*}{\partial z} = 0 \tag{3.20}$$

alors

$$\begin{aligned}\beta_k\left(\frac{\partial D(z^{(k+1)},z^*)}{\partial z} - \frac{\partial D(z^{(k+1)},z^k)}{\partial z}\right) = \frac{\partial^2 f(z^*)}{\partial z^2}(z^{(k+1)} - z^*) + \beta_k \frac{\partial^2 D(z^*,z^*)}{\partial z^2}(z^{(k+1)} - z^*) \\ + R(z^{(k+1)} - z^*)\end{aligned} \tag{3.21}$$

qui pourra se mettre sous la forme suivante :

$$\begin{aligned}\left\|\beta_k\left(\frac{\partial D(z^{(k+1)},z^*)}{\partial z} - \frac{\partial D(z^{(k+1)},z^k)}{\partial z}\right) - R(z^{(k+1)} - z^*)\right\| = \\ \left\|\frac{\partial^2 f(z^*)}{\partial z^2}(z^{(k+1)} - z^*) + \beta_k \frac{\partial^2 D(z^*,z^*)}{\partial z^2}(z^{(k+1)} - z^*)\right\|\end{aligned} \tag{3.22}$$

D'autre part, vu que $D(z, \overline{z})$ est continuellement différentiable, $\frac{\partial D(z,\overline{z})}{\partial z})$ est Lipschitz locale en ses variables z et $\overline{z}$.

Donc, et puisque z^k est bornée, il existe un ensemble borné B contenant z^k et une constante finie L positive telle que :

$$\|\frac{\partial D(z,\overline{z})}{\partial z} - \frac{\partial D(z',\overline{z'})}{\partial z}\| \leq L(\|z-z'\|^2 + \|\overline{z}-\overline{z'}\|^2)^{1/2} \tag{3.23}$$

En utilisant maintenant l'inégalité triangulaire et ce dernier résultat, (3.22) devient :

$$\|\frac{\partial^2 f(z^*)}{\partial z^2}(z^{(k+1)}-z^*) + \beta_k \frac{\partial^2 D(z^*,z^*)}{\partial z^2}(z^{(k+1)}-z^*)\| \leq \beta_k L\|z^k - z^*\| + \|R(z^{(k+1)}-z^*)\| \tag{3.24}$$

utilisons maintenant les propriétés de convexité des deux fonctions f(z) et D(z,z') :

$$(\lambda_{f(z)} + \beta_k \lambda_D)\|z^{k+1}-z^*\| \leq \beta_k L\|z^k - z^*\| + \|R(z^{(k+1)}-z^*)\| \tag{3.25}$$

avec $\lambda_{f(z)} = \min_z \lambda_{\nabla^2 f(z)}$ et $\lambda_D = \min_{z,\overline{z}} \lambda_{\nabla^2 D(z,\overline{z})}$

Cette dernière équation se simplifie comme suit :

$$\beta_k L \geq (\lambda_{f(z)} + \beta_k \lambda_D - \frac{\|R(z^{(k+1)}-z^*)\|}{\|z^{(k+1)}-z^*\|})\frac{z^{(k+1)}-z^*}{z^{(k)}-z^*} \tag{3.26}$$

Rappelons maintenant que z^k est convergente : $\lim_{k\to\infty}\|z^k - z^*\| = 0$, et que d'après la définition du reste : $\lim_{k\to\infty}\frac{\|R(z^{k+1}-z^*)\|}{\|z^{k+1}-z^*\|} = 0$, ayant en plus que $\lambda_{f(z)} > 0$ (car f(z) est convexe), nous pouvons conclure que si, dans (3.26), $\lim_{k\to\infty}\beta_k = 0$, on aura aussi une convergence super-linéaire du fait que :

$$\lim_{k\to\infty}\frac{\|z^{k+1}-z^*\|}{\|z^k - z^*\|} = 0 \tag{3.27}$$

Nous pouvons donc conclure que la convergence super linéaire est conservée avec une distance de Kullback-Leibler dans le terme de pénalité à la place de la norme euclidienne avec les mêmes conditions :

- f est une fonction convexe
- $\lim_{k\to\infty}\beta_k = 0$

Dans la suite, toutes les reformulations point proximal apportées seront à la base d'une divergence de Kullback-Leibler bien définie comme terme de pénalité (voire une différence de distances de Kullback-Leibler).

Parmi les divers études utilisant l'algorithme de point proximal dans l'analyse des algorithmes itératifs, on peut citer les travaux d'Alfred Hero [CH98, CH08]. Dans ces travaux, les auteurs proposent une version accélérée de l'algorithme EM (Expectation Maximization) suite à une interprétation de type point proximal de cet algorithme utilisant aussi la distance de Kullback dans le terme de pénalité et profitant de la propriété de la convergence super linéaire. D'autres travaux viennent dans la suite compléter ces résultats préliminaires [Tse04].

Dans cette thèse, nous proposerons des interprétations basées sur cette approche proximale pour d'autres algorithmes itératifs bien connus dans la littérature. Ainsi, et dans le chapitre suivant, nous mettons en évidence une approche proximale pour l'algorithme itératif de Blahut-Arimoto pour le calcul de la capacité des canaux discrets sans mémoire qui nous a aidé à améliorer cet algorithme (de point de vue accélération de la vitesse de convergence). Nous proposons ensuite une interprétation point proximal de l'algorithme de décodage itératif des modulations codées à bits entrelacés tout en montrant le bonus qu'apporte cette méthode par rapport à la version classique.

4

Algorithme de Blahut-Arimoto : interprétations géométriques et analyse point proximale

4.1 Introduction

En 1972, R. Blahut et S. Arimoto [Ari72, Bla72] ont proposé simultanément un algorithme itératif de calcul de la capacité des canaux sans mémoire avec des entrées et des sorties à alphabets finis ainsi que les fonctions de taux de distorsion. Depuis, plusieurs extensions ont été proposées citons notamment [DYW04] qui a étendu l'algorithme de Blahut-Arimoto aux canaux avec mémoire et entrées à alphabets finis et [Dau06] qui a considéré des canaux sans mémoire avec des entrées et/ou des sorties continues.

En parallèle, d'autres travaux se sont concentrés sur l'interprétation géométrique de l'algorithme de Blahut-Arimoto en termes de projections alternées [CT84].

Dans ce chapitre, nous reconsidérons le problème du calcul de la capacité des canaux discrets et sans mémoire d'un point de vue géométrique. Nous reprenons ainsi les outils de base de la géométrie de l'information utilisés dans cette dernière approche [CT84] et essayons de proposer d'autres interprétations géométriques. Nous proposons ensuite un algorithme de gradient naturel et une version accélérée de BA pour le calcul de la capacité tout en montrant l'équivalence de ces deux algorithmes au voisinage de la solution optimale. Une approche point proximal est proposée pour chacun des algorithmes avec comme terme de pénalité la divergence de Kullback-Leibler dans le cas de BA accélérée et une divergence chi-2 dans le cas gradient naturel. Nous présentons une analyse de convergence de ces deux algorithmes en utilisant certains résultats de [CH99]. Des résultats de simulation viennent illustrer cette analyse et montrer l'intérêt de nos deux algorithmes par rapport au cas classique (accélération de la vitesse de convergence).
Ce chapitre sera un bref résumé des résultats obtenus et soumis à IEEE Transactions on Information Theory présentés en annexe A. Cependant les interprétations géométriques apportées dans la section 2 de ce chapitre ne figurent pas dans notre article.

4.2 Algorithme classique de Blahut-Arimoto et ses interprétations géométriques

4.2.1 Algorithme classique de Blahut-Arimoto

Considérons un canal discret sans mémoire avec pour entrée X prenant ses valeurs dans l'ensemble $\{x_0, \ldots, x_M\}$ et en sortie Y prenant ses valeurs dans l'ensemble $\{y_0, \ldots, y_N\}$. Ce canal est défini par sa matrice de transition $\mathbf{Q}$ telle que $[Q]_{ij} = Q_{i|j} = Pr(Y = y_i|X = x_j)$. Les distributions des symboles d'entrée et de sortie sont caractérisées respectivement par les vecteurs $\mathbf{p} = [p_0 \cdots p_M]^T$ et $\mathbf{q} = [q_0 \cdots q_N]^T = \mathbf{Qp}$ avec $p_j = \Pr(X = x_j)$ et $q_i = \Pr(Y = y_i) = \sum_{j=0}^{M} Q_{i|j}\, p_j$.

L'information mutuelle $H(Y) - H(Y|X)$ de X et Y est égale à [Gal68, CT91] :

$$I(\mathbf{Q}, \mathbf{p}) = \sum_{j=0}^{M} \sum_{i=0}^{N} p_j \, Q_{i|j} \log \frac{Q_{i|j}}{q_i} \, . \tag{4.1}$$

Avec $\mathbf{Q}_j = [Q_{0|j} \dots Q_{N|j}]^T$ représente la $j$ème colonne de $\mathbf{Q}$. La capacité de canal est :

$$C(\mathbf{Q}) = \max_{\mathbf{p}} I(\mathbf{Q}, \mathbf{p}). \tag{4.2}$$

En résolvant ce problème de maximisation et en prenant en compte la condition de normalisation, nous obtenons le processus itératif :

$$p_j^{k+1} \; = \; p_j^k \, \frac{\exp(D_j^k)}{\sum_{j=0}^{M} p_j^k \exp(D_j^k)}. \tag{4.3}$$

où $D_j^k \triangleq D(\mathbf{Q}_j \| \mathbf{q}^k)$ avec $\mathbf{q}^k = \mathbf{Q}\mathbf{p}^k$. C'est la Divergence de Kullback-Leibler entre la distribution courante de sortie $\mathbf{q}^k$ et la $j$ème colonne de $\mathbf{Q}$.

4.2.2 Interprétations géométriques de l'algorithme de Blahut-Arimoto

L'algorithme de Blahut-Arimoto dans (4.3) peut être recalculé comme un problème de minimisation :

$$\begin{cases} \min_p & D(p(x)||p^{(k)}(x)) \\ s.c & I^{(k)}(p(x)) = \alpha \\ s.c & \sum_x p(x) = 1 \end{cases} \tag{4.4}$$

où $I^{(k)}(p(x)) = \mathbb{E}_p\{D(p(y|x)||p^k(y))\}$ est l'estimé courante de la capacité à l'itération k et α est lié au multiplieur de Lagrange de ce problème d'optimisation.
Le Lagrangien correspondant à ce problème de minimisation pourra être écrit comme suit :

$$\mathfrak{L} = D(p(x)||p^{(k)}(x)) - \lambda_1(I^{(k)}(p(x)) - \alpha) - \lambda_2(\sum_x p(x) - 1) \tag{4.5}$$

$\frac{\partial \mathfrak{L}}{\partial p(x)} = 0 \Rightarrow \log(p(x)) + 1 - \log(p^{(k)}(x)) - \lambda_1 D_j^k - \lambda_2 = 0$ et $p(x) = p^{(k)}(x) \exp(\lambda_2 - 1) \exp(\lambda_1 D_j^k)$

En prenant en compte des contraintes de normalisation, nous obtenons $\exp(\lambda_2 - 1) = \frac{1}{\sum_x p^{(k)}(x)\exp(\lambda_1 D_j^k)}$ et $p^{(k+1)}(x) = \frac{p^{(k)}(x)\exp(\lambda_1 D_j^k)}{\sum_x p^{(k)}(x)\exp(\lambda_1 D_x^k)}$
Nous verrons dans la suite que ce paramètre λ_1 est un paramètre de pas qui, pour des valeurs convenables, pourra accélérer la vitesse de convergence de l'algorithme classique de Blahut-Arimoto dans lequel $\lambda_1 = 1$.
D'où l'algorithme de Blahut-Arimoto peut être interprété comme étant la projection de $p^{(k)}(x)$ sur une famille linéaire de probabilité $\mathcal{L}$ au point $p^{(k+1)}(x)$ où $\mathcal{L}$ est définie par $f_1(x) = D_x^k = D(p(y/x)||p^{(k)}(y))$ et α_1^k est telle que $\mathbb{E}_p(D_j^k) = \alpha_1^k$.

En choisissant α_1^k croissante au fil des itérations, on pourra garantir que l'information mutuelle augmente d'une itération courante à une autre ($I^{(k+1)}(p(x)) \geq I^{(k)}(p(x))$). Cependant, cette quantité n'est qu'implicitement définie dans l'algorithme et un choix approprié n'est pas évident. Nous montrerons dans la suite que ce problème pourra être résolu en se basant sur une interprétation de type point proximal qui assure que l'information mutuelle croît au fil des itérations. Notons que cette famille linéaire de probabilité change d'une itération à une autre.

D'autre part, l'algorithme de Blahut-Arimoto peut être vu comme une projection d'une densité de probabilité sur une famille exponentielle de probabilité $\mathcal{E}$ définie par $Q(x) = p^{(k)}(x)$, $f_1^{(k)}(x) = D_j^k$ et paramétrée par $\theta_1^{(k)}$ au point $p^{(k+1)}(x)$.

Pour faire cela, nous devrons résoudre ce problème :

$$\begin{cases} \min\limits_{\theta} D(R(x)||p(x,\theta)) \\ p(x,\theta) = \frac{Q(x)exp(\theta f_1(x)}{\sum_x Q(x)exp(\theta f_1(x))} \end{cases} \tag{4.6}$$

où $R(x)$ est une densité de probabilité quelconque. Nous essayons maintenant de trouver quelques caractéristiques intéressantes de $R(x)$. Pour cela, il faut résoudre le problème de minimisation précédent.

$$\sum_x \frac{\partial(R(x)\log p(x))}{\partial\theta} = 0 \tag{4.7}$$

avec

$$\log p(x,\theta) = \log Q(x) + \theta f_1(x) - \log(\sum_x Q(x)\exp(\theta f_1(x))) \tag{4.8}$$

D'où

$$\sum_x R(x) f_1(x) - \frac{\sum_x R(x) \sum_x Q(x) f_1(x) \exp(\theta f_1(x))}{\sum_x Q(x) \exp(\theta f_1(x))} = 0 \tag{4.9}$$

On a donc

$$\sum_x R(x) f_1(x) - \frac{\sum_x Q(x) f_1(x) \exp(\theta f_1(x))}{\sum_x Q(x) \exp(\theta f_1(x))} \sum_x R(x) = 0 \tag{4.10}$$

qui nous mène à

$$\sum_x (R(x) - p(x,\theta)) f_1(x) = 0 \tag{4.11}$$

en prenant en considération que

$$\sum_x R(x) = 1 \tag{4.12}$$

et

$$p(x,\theta) = \frac{Q(x) \exp(\theta f_1(x))}{\sum_x Q(x) \exp(\theta f_1(x))} \tag{4.13}$$

Nous obtenons donc

$$\sum_x (R(x) - p^{(k+1)}(x)) D_x^k = 0 \tag{4.14}$$

qui pourra être reformulé comme suit :

$$I(R,Q) = \mathbb{E}_R(D_j^k) = \mathbb{E}_p^{(k+1)}(D_j^k) = I(p^{(k+1)}(x), Q) \geq I(p^{(k)}(x)) \tag{4.15}$$

L'algorithme itératif de Blahut-Arimoto peut donc être interprété comme étant la projection des densités de probabilité $R(x)$ ayant une information mutuelle supérieure à $I(p^{(k)}(x))$ sur une famille exponentielle de probabilité $\mathcal{E}$ définie par $Q(x) = p^{(k)}(x)$, $f_1^{(k)}(x) = D_j^k$ et paramétrée par $\theta_1^{(k)} = 1/\lambda_k$ au point $p^{(k+1)}(x)$. Notons que cette famille exponentielle change aussi d'une itération à une autre vu que $Q(x)$ et $f_1^{(k)}(x)$ dépendent des itérations. Ici, un choix convenable du paramètre pour augmenter la vitesse de la convergence est aussi difficile du à la définition implicite de cette famille. Pour cela, une interprétation point proximal maximisant explicitement l'information mutuelle sera considérée dans la suite avec un terme de pénalité bien défini.

4.3 Algorithme de gradient naturel

Dans cette section, nous proposons un nouvel algorithme pour le calcul de la capacité basé sur le gradient naturel [AD98]. Nous allons exploiter le fait que les vecteurs de probabilité d'entrée $\mathbf{p}$ constituent un ensemble de Riemann de dimension M avec comme métrique associée la matrice d'information de Fisher. L'avantage de l'approche gradient naturel réside dans le fait que la courbure de cet ensemble est bien prise en considération [AD98].

Cet algorithme de gradient naturel pour le calcul de la capacité pourra être représenté (voir annexe A), de la manière itérative suivante :

$$p_j^{k+1} = p_j^k \left[1 + \mu_k (D_j^k - I^k)\right]. \tag{4.16}$$

avec $I^k \triangleq I(\mathbf{p}^k)$ est l'estimée courante de la capacité. Notons que, comme dans le cas de BA, (4.16) correspond à une mise à jour multiplicative. Cependant, la complexité de calcul dans cet algorithme de gradient naturel a diminué par rapport au cas BA car il n'y a pas d'exponentielle dans l'équation itérative de mise à jour.

Ayant $\sum_{j=0}^{M} p_j \, D_j^k = I^k$, (4.16) garantit que $\sum_{j=0}^{M} p_j^{k+1} = 1$. Cependant, il faut choisir un pas μ_k assez petit pour assurer que $p_j^{k+1} \geq 0$. Cette contrainte est en conflit avec notre but d'accélérer la vitesse de la convergence de ce processus itératif (choisir un pas assez grand). En effet, la non négativité de p_j^{k+1} est assurée pour $\mu_k \leq -\frac{1}{\min_j D_j^k - I^k}$. Des simulations numériques indiquent qu'un choix de μ_k proche de $-\frac{1}{\min_j D_j^k - I^k}$ mène à un comportement instable dans les itérations de l'approche gradient naturel. Une convergence stable est cependant observée avec $\mu_k = \frac{1}{I^k} \leq -\frac{1}{\min_j D_j^k - I^k}$. Avec cette valeur du pas, ainsi qu'avec des valeurs fixes de pas $\mu_k = \mu > 1$, l'algorithme de gradient naturel converge plus vite que BA.

4.4 Algorithme de Blahut-Arimoto accéléré

Une comparaison numérique entre BA et NG montre que, pour des pas fixes $\mu_k = \mu$, l'algorithme NG converge μ fois plus vite : les propriétés de convergence sont presque

les même pour $\mu = 1$.

Pour démontrer cela, nous divisons le numérateur et le dénominateur de (4.3) par $\exp(I^k)$ et utilisons ensuite le développement limité en série de Taylor de premier ordre des exponentielles :

$$p_j^{k+1} = p_j^k \frac{\exp(D_j^k - I^k)}{\sum_{j=0}^{M} p_j^k \exp(D_j^k - I^k)} \approx p_j^k \left[1 + (D_j^k - I^k)\right]. \tag{4.17}$$

Le terme de droite n'est autre que l'équation de mise à jour de NG pour $\mu_k = 1$. Cette approximation en série de Taylor est valable pour $D_j^k - I^k \approx 0$. Cela est vérifiée au voisinage de la solution optimale. Nous pouvons ainsi conclure que BA et NG sont asymptotiquement équivalents pour $\mu_k = 1$. Cette relation entre ces deux algorithmes nous conduit à considérer un algorithme BA plus général :

$$p_j^{k+1} = p_j^k \frac{\exp(\mu_k D_j^k)}{\sum_{j=0}^{M} p_j^k \exp(\mu_k D_j^k)}. \tag{4.18}$$

En se servant des même arguments, nous pouvons montrer que cet algorithme est asymptotiquement équivalent à celui de gradient naturel. En effet, en utilisant le même pas $\mu_k = \mu$, les deux algorithmes possèdent la même vitesse de convergence qui est μ fois plus rapide que celle de l'algorithme BA classique. Nous appelerons l'algorithme (4.18) algorithme de BA accéléré.

4.5 Interprétations point proximal

On peut montrer sans difficulté que l'algorithme classique de Blahut-Arimoto est équivalent à :

$$\mathbf{p}^{(k+1)} = \arg\max_{\mathbf{p}} \{I^{(k)}(\mathbf{p}, \mathbf{Q}) - D(\mathbf{p}||\mathbf{p}^{(k)})\} \tag{4.19}$$

où $I^{(k)}(\mathbf{p}, \mathbf{Q}) = \mathbb{E}_{\mathbf{p}}\{D_j^k\}$. Cet algorithme n'est pas un algorithme du point proximal puisque la fonction de coût $I^{(k)}(\mathbf{p}, \mathbf{Q})$ dépend des itérations. Il est toutefois possible d'exprimer l'information mutuelle comme suit :

$$I(\mathbf{p}, \mathbf{Q}) = I^{(k)}(\mathbf{p}, \mathbf{Q}) - D(\mathbf{q}||\mathbf{q}^{(k)}) \tag{4.20}$$

En introduisant (4.20) dans (4.19), nous obtenons :

$$\mathbf{p}^{(k+1)} = \arg\max_{\mathbf{p}}\{I(\mathbf{p},\mathbf{Q}) - \Big(D(\mathbf{p}||\mathbf{p}^{(k)}) - D(\mathbf{q}||\mathbf{q}^{(k)})\Big)\} \tag{4.21}$$

D'après l'inégalité de Jensen, nous pouvons montrer que le terme de pénalité

$$\begin{aligned} D(\mathbf{p}||\mathbf{p}^{(k)}) - D(\mathbf{q}||\mathbf{q}^{(k)}) \quad &= \\ \mathbb{E}_{p(x,y)}[\log \frac{p(x)\sum_{\tilde{x}} p(y|\tilde{x})p^{(k)}(\tilde{x})}{p^{(k)}(x)\sum_{\tilde{x}} p(y|\tilde{x})p(\tilde{x})}] \end{aligned} \tag{4.22}$$

est toujours positif et qu'il est nul si et seulement si
$p(x) = p^{(k)}(x)$ et $q(y) = q^{(k)}(y)$.
D'où, nous pouvons conclure que l'algorithme classique de BA est un point proximal vu que le terme de pénalité est une distance.

Essayons d'introduire maintenant dans la reformulation point proximal un pas γ_k dans le but d'avoir un degré de liberté supplémentaire permettant d'agir sur la vitesse de convergence. Le problème sera ainsi :

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}}\left\{I(\mathbf{p},\mathbf{Q}) - \gamma_k\Big(D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k)\Big)\right\} \tag{4.23}$$

Pour $\gamma_k = 1$, nous retrouvons ainsi le cas classique. La solution de ce problème d'optimisation, après normalisation, est donnée par :

$$p_j^{k+1} = C^{k+1} p_j^k exp\left(\sum_i Q_{i|j} \log \frac{q_i^{k+1}}{q_i^k} + \frac{1}{\gamma_k}\sum_i Q_{i|j} \log \frac{Q_{i|j}}{q_i^{k+1}}\right) \tag{4.24}$$

où C^{k+1} est une constante de normalisation.
Afin d'obtenir une expression analytique, le terme de droite doit être indépendant de $\mathbf{p}^{k+1}$ et $\mathbf{q}^{k+1}$. Cela peut être assuré soit en fixant $\gamma_k = 1$, soit en remplaçant q_i^{k+1} par sa valeur q_i^k de l'itération précédente. Le premier cas correspond exactement au cas classique de BA, et le deuxième cas est connu comme étant l'algorithme de One Step Late [Gre90].
Appliquée à (4.24), la technique de One Step Late mène à cette équation itérative de

mise à jour :

$$p_j^{k+1} = p_j^k \frac{\exp(\frac{1}{\gamma_k} D_j^k)}{\sum_{j=0}^{M} p_j^k \exp(\frac{1}{\gamma_k} D_j^k)}. \tag{4.25}$$

C'est exactement l'algorithme BA accéléré obtenu dans (4.18).Nous pouvons montrer que cet algorithme n'est autre que la solution du problème d'optimisation suivant :

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \left\{ I(\mathbf{p}, \mathbf{Q}) - \left(\gamma_k D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k) \right) \right\} \tag{4.26}$$

C'est un algorithme de point proximal à condition que le terme de pénalité soit assimilable à une distance. Si le pas γ_k est choisi tel que $\gamma_k D(\mathbf{p} \parallel \mathbf{p}^k) \geq D(\mathbf{q} \parallel \mathbf{q}^k)$, la convergence de la méthode est garantie et l'information mutuelle augmente toujours au fil des itérations.

D'autre part, nous pouvons aussi montrer que l'algorithme gradient naturel n'est autre que la solution de ce problème de maximisation :

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}} \left\{ \tilde{I}^k(\mathbf{p}, \mathbf{Q}) - \gamma_k \, \chi^2(\mathbf{p} \| \mathbf{p}^k) \right\} \tag{4.27}$$

obtenu pour $\mu_k = 1/\gamma_k$.

Nous pouvons donc conclure que les deux algorithmes proposés dans ce chapitre peuvent être interprétés comme étant des algorithmes du point proximal utilisant la même fonction coût mais différents termes de pénalité. Leur équivalence asymptotique vient du fait que $\chi^2(\mathbf{p}, \mathbf{p}') \approx D(\mathbf{p} \| \mathbf{p}')$ pour tout $\mathbf{p}$ proche de $\mathbf{p}'$.

Concernant le choix du pas, γ_k doit être choisi de telle manière à ce que l'information mutuelle ne fasse que croitre au fil des itérations. Une condition suffisante est $\gamma_k \, D(\mathbf{p}^{k+1} \| \mathbf{p}^k) - D(\mathbf{q}^{k+1} \| \mathbf{q}^k) \geq 0$. Donc le choix du pas doit vérifier l'inégalité suivante :

$$\mu_k \leq \frac{D(\mathbf{p}^{k+1} \| \mathbf{p}^k)}{D(\mathbf{q}^{k+1} \| \mathbf{q}^k)} = \frac{D(\mathbf{p}^{k+1} \| \mathbf{p}^k)}{D(\mathbf{Q}\mathbf{p}^{k+1} \| \mathbf{Q}\mathbf{p}^k)} \, . \tag{4.28}$$

Cependant il est impossible de calculer, à l'itération courante, cette borne supérieure puisqu'elle dépend des quantités de l'itération suivante. Inspirés par la technique de One Step Late, nous avons choisi, dans les simulations, $\mu_k = D(\mathbf{Q}\mathbf{p}^k \| \mathbf{Q}\mathbf{p}^{k-1}) / D(\mathbf{p}^k \| \mathbf{p}^{k-1})$ qui consiste à remplacer toutes les quantités par leurs valeurs de l'itération précédente.

Nous avons enfin analysé explicitement la convergence des deux algorithmes proposés. Des résultats explicites concernant la convergence de l'algorithme de BA accéléré sont présentés. Bien que ces résultats explicites ne soient pas évidents dans le cas de l'algorithme du gradient naturel, l'équivalence asymptotique avec l'algorithme de BA accéléré les rend accessibles.

4.6 Exemple numérique

Pour illustrer nos résultats, considérons le cas d'un canal binaire symétrique avec $\mathbf{Q} = \begin{pmatrix} 0.7 & 0.1 \\ 0.2 & 0.2 \\ 0.1 & 0.7 \end{pmatrix}$. Pour calculer la capacité de ce canal, nous utilisons l'algorithme classique de BA, l'algorithme BA accéléré et l'algorithme de gradient naturel avec la même initialisation $\mathbf{p}^0$. Le pas de nos algorithmes est choisi d'une manière adaptative selon $\mu_k = D(\mathbf{Q}\mathbf{p}^k \| \mathbf{Q}\mathbf{p}^{k-1}) / D(\mathbf{p}^k \| \mathbf{p}^{k-1})$ pour $k > 1$ et $\mu_1 = 1$, avec comme critère d'arrêt $E^k = \max_j D_j^k - I^k$. Les résultats de convergence à 12 décimaux près sont montrés sur la figure (4.1) (tous les algorithmes délivrent $C = 0.365148445440$ bit)).

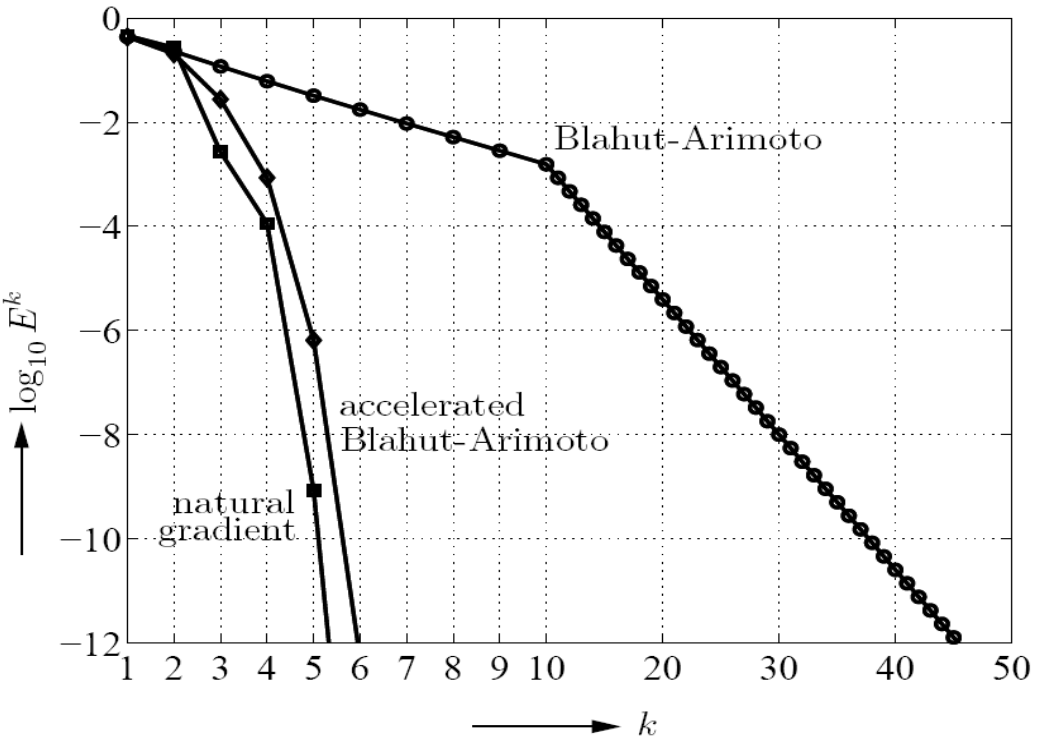

FIG. 4.1 – Convergence des algorithmes BA, BA accéléré et NG avec un pas adaptatif

Nous pouvons donc remarquer que la convergence de nos deux algorithmes (6 itérations) est beaucoup plus rapide que celle de BA classique (46 itérations). Cette figure montre clairement que nos deux algorithmes utilisés avec des pas adaptatifs réalisent une convergence super linéaire.

4.7 Conclusions

Dans ce chapitre nous avons proposé des améliorations de l'algorithme classique de Blahut-Arimoto (BA) pour le calcul de la capacité d'un canal discret sans mémoire. Des interprétations basées sur la géométrie de l'information sont présentées. Un algorithme BA accéléré et un autre algorithme gradient naturel sont introduits ayant une vitesse de convergence beaucoup plus grande que dans le cas classique (convergence super linéaire en général). Une approche point proximale est proposée pour chacun des deux algorithmes avec comme terme de pénalité la divergence de Kullback-Leibler dans le cas de BA accélérée et une divergence chi-2 dans le cas gradient naturel. Cela nous donne des pistes théoriques quant à l'analyse de la convergence de ces algorithmes.
Bien que notre étude soit concentrée sur les canaux discrets sans mémoire, nos résultats peuvent être généralisés pour les canaux à accès multiples [RG04], canaux quantiques [Nag98], canaux à interférences [Kav01, Von01] et les canaux avec information adjacente [DYW04].

5

Décodage itératif des BICM : approche point proximal

5.1 Introduction

Les modulations codées à Treillis ("**T**rellis **C**oded **M**odulations") [Ung82][Ung87] ont été longtemps considérées comme étant le meilleur choix pour une bonne performance de transmission. En effet, la modulation et le codage sont considérés simultanément. Cependant, les TCM souffrent de deux inconvénients majeurs empêchant leur utilisation dans les communications sans fil : **i**) les systèmes de base de ces modulations codées sont compatibles uniquement avec l'entrelacement symbole menant à une mauvaise performance dans le cas des canaux à Rayleigh par comparaison aux systèmes utilisant l'entrelacement bit ; **ii**) les TCMs sont conçues pour des codages à taux fixes mais peu

flexibles.

Il en suit, dans la plupart des systèmes sans fil récents, que l'entrelacement bit est plus utilisé que l'entrelacement symbole. Par conséquent, le codage canal, l'entrelacement bit et le mapping bit-symbole sont séparément réalisés en suivant l'idée originalement introduite dans [VWZP89]. Cela est connu dans la littérature comme modulation codée à bits entrelacés (**B**it-**I**nterleaved **C**oded **M**odulation) [CGB98] où l'entrelacement est effectué avant la modulation bit-symboles complexes.

La modulation codée à bits entrelacés est une approche pragmatique de la modulation codée. Elle a été tout d'abord proposée par Zehavi [Zeh92] pour améliorer les performances des modulations codées à Treillis dans le cas des canaux de Rayleigh à évanouissement. Une analyse des taux atteints et de la probabilité d'erreur est donnée par Caire [CGB98]. Les BICMs ont été récemment utilisées dans quelques standards comme DVB-S2, wireless LANs, DSL et WiMax grâce à leur flexibilité et leur simplicité. Les BICMs combinent les codes correcteurs d'erreur avec des schémas de modulation d'ordre élevé. Bien qu'elles aient étées à l'origine développées pour des canaux à évanouissement (SISO) [Zeh92][CGB98], les modulations codées à bits entrelacés ont été rapidement étendues pour les systèmes multi-Antennes (MIMO systems) [BBL00].

C'est maintenant un sérieux concurrent par rapport aux codes espace-temps (Space-time (ST) codes), qui exploitent la diversité spatiale dans les environnements MIMO contre des faibles taux de transmission. D'autre part, les BICM peuvent assurer des taux élevés de transmission tout en maintenant une grande diversité [FG98]. Dans les BICM, l'ordre de diversité est augmenté par l'utilisation d'entrelaceurs bits au lieu d'entrelaceurs symboles. Cela est réalisé en dépit d'une réduction de la distance Euclidienne minimale menant à une dégradation de performance sur les canaux Gaussiens sans évanouissement [Zeh92]. Cela peut être résolu par l'usage d'un décodage itératif (BICM-ID) au niveau du récepteur qui consiste à échanger des informations extrinsèques entre le décodeur canal et le démodulateur selon un processus de type turbo jusqu'à arriver à la convergence [LCR02].

Les BICM-ID donnent d'excellentes performances pour les canaux Gaussiens et à évanouissement. Le schéma de décodage itératif utilisé dans les BICM-ID est très simi-

laire aux turbo décodeurs série. Dans les BICM-ID, le décodeur interne est remplacé par un démodulateur qui nécessite moins de complexité que l'étape de décodage. C'est pour cela nous avons considéré dans cette thèse l'étude du décodage itératif des BICM.

Pour une constellation, un entrelaceur et un code correcteur d'erreur fixés, le mapping signal joue un rôle important dans la détermination de la performance d'erreur d'un système de BICM-ID. Bien que ce chapitre étudie le décodage itératif des BICM, les résultats peuvent être appliqués sur la large classe des décodeurs itératifs incluant les turbo décodeurs série ou parallèle ainsi que les décodeurs "Low-Density Parity-Check" (LDPC).

Aucun de ces décodeurs turbo n'a été à l'origine introduit comme solution d'un problème d'optimisation ce qui rend leur structure précise adhoc (l'échange des informations extrinsèques entre les constituants d'un récepteur itératif à la place des probabilités a posteriori était initialement intuitif) et l'analyse de leur convergence et stabilité très difficile.

Parmi les différentes approches pour l'analyse du décodage itératif, les analyses par EXIT chart et évolution de densité ont permis de faire un progrès important [GH01][tB01] mais les résultats développés selon ces approches peuvent être appliqués uniquement dans le cas de blocs de grande taille. Un autre outil d'analyse est la connexion du décodage itératif à la théorie des graphes [KFL01] et à la propagation des croyances (Belief propagation) [Pea88]. Des résultats de convergence pour la propagation des croyances existent mais sont limités au cas où le graphe correspondant est un arbre ce qui exclue les turbo codes et les codes LDPC. Un lien entre le décodage itératif et les algorithmes classiques d'optimisation a été récemment établi dans [WRJ06] où le décodage turbo est interprété comme étant la solution d'un problème d'optimisation avec contraintes. Vu que l'ensemble des contraintes n'est pas fixe, les outils classiques ne sont pas efficaces pour analyser le comportement d'une procédure itérative à la convergence.

Une approche géométrique a été considérée dans [WJR05], elle fournit une interprétation intéressante en termes de projections. Le cas particulier des BICM-ID a été étudié dans [MDdC02] menant à une bonne caractérisation du démodulateur et de décodeur.

Cependant, une caractérisation du processus entier reste à exploiter.

Dans ce chapitre, nous reformulons le décodage itératif des BICM comme étant une minimisation d'une distance de Kullback dans le but d'essayer de trouver une interprétation géométrique et une autre de type point proximal caractérisant le processus itératif en entier. Ces deux approches ont été introduites dans les chapitres précédents.

5.2 BICM-ID

Dans cette section, nous présentons la structure classique des BICM-ID et nous introduisons les notations qui seront utilisées tout au long de ce chapitre. Un résultat peu connu obtenu dans [Muq01] et résumé dans [MDdC02] est aussi utilisé. Il nous permet de caractériser le décodeur canal d'une manière très compacte. Un système

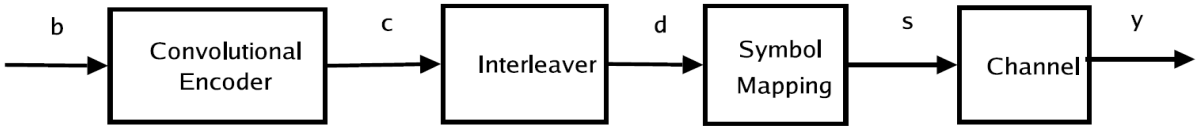

FIG. 5.1 – Modèle de transmission

conventionnel de BICM [CGB98] est construit à partir d'une concaténation série d'un codeur convolutif, d'un entrelaceur bit et d'un mapping bits-symbole de taille M (où $M = 2^m$) comme le montre la figure (5.1). Une séquence de bits d'information $\underline{b}$ est d'abord codée par le codeur convolutif afin de produire une séquence de bits codés $\underline{c}$ de longueur L_c. Cette séquence $\underline{c}$ est ensuite entrelacée par un entrelaceur bit opérant sur les indices des bits. On note $\underline{d}$ la séquence des bits codés et entrelacés. Ensuite, m bits consécutifs de $\underline{d}$ sont groupés pour former un symbole canal $d_k = (d_{km+1}, \cdots, d_{(k+1)m})$. Le signal complexe transmis $s_k = \varepsilon(d_k)$, $1 \leq k \leq M$, est ensuite choisi à partir d'une constellation M-aire ψ où ε désigne le schéma du mapping. Pour simplifier les choses, nous considérons la transmission sur un canal Gaussien à bruit blanc additif. Les signaux reçus peuvent être écrits comme suit :

$$y_k = s_k + n_k \quad 1 \leq k \leq M \tag{5.1}$$

où n_k est un bruit blanc complexe Gaussien. Le décodage par maximum de vraisem-

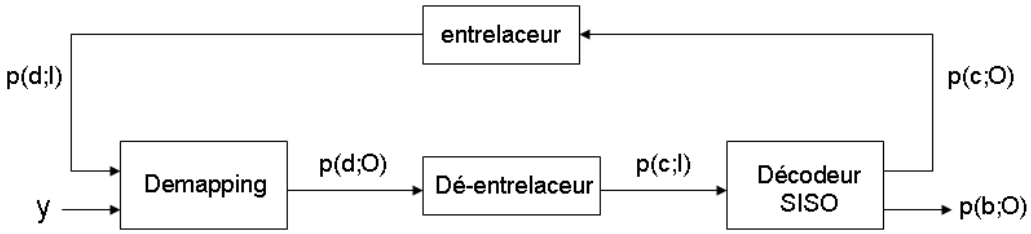

FIG. 5.2 – Receiver for a BICM-ID with soft-decision feedback

blance des BICM est très compliqué à implémenter en pratique due à la présence d'un entrelaceur bit aléatoire. La figure (5.2) montre le schéma bloc du récepteur d'un système BICM-ID avec un retour à décision souple. Le "retour" ici signifie que chaque bloc produit des informations qui seront utilisées par l'autre bloc comme des probabilités *a priori* sur chaque bit. Notons que, dans ce chapitre, toutes les quantités sont supposées normalisées, ce qui nous permet d'écrire l'a posteriori, l'a priori, ainsi que les quantités extrinsèques comme des densités de probabilité.

Le démodulateur consiste à évaluer les probabilités *a posteriori* (APP) des bits codés sans prendre en considération la structure du code, cela se traduit par :

$$p_{APP}(d_{km+i} = b) = p(d_{km+i} = b|y_k) \sim \sum_{s_k \in \psi_b^i} p(y_k|s_k)p(s_k) \tag{5.2}$$

où ψ_b^i, $b \in \{0, 1\}$, représente le sous ensemble de ψ qui contient tous les symboles ayant la valeur b à la position i. Dans les processus de turbo décodage, les quantités échangés entre les blocs ne sont pas les probabilités *a posteriori* mais plutôt l'information extrinsèque [BGT93]. Notons que la relation générale entre la probabilité a posteriori et l'information extrinsèque est donnée par :

$$APP = (\text{Extrinsèque}) * (\text{A priori}) \tag{5.3}$$

L'information extrinsèque évaluée à la sortie du démodulateur $p(d_{km+i}; O)$ est calculée par $\frac{p_{APP}(d_{km+i})}{p(d_{km+i};I)}$ où $p(d_{km+i}; I)$ est la probabilité *a priori* pour le bloc de démodulateur. Vu que l'entrelaceur bit rend les bits toujours indépendants, la probabilité *a priori* du symbole peut être approximée par le produit des probabilités *a priori* des bits composant

ce symbole et l'APP sera :

$$p_{APP}(d_{km+i} = b) = K_m \sum_{s_k \in \psi_b^i} p(y_k|s_k) \prod_j p(d_{km+j}; I) \tag{5.4}$$

et l'information extrinsèque correspondant $p(d_{km+i}; O)$ sera :

$$p(d_{km+i} = b; O) = K'_m \sum_{s_k \in \psi_b^i} p(y_k|s_k) \prod_{j \neq i} p(d_{km+j}; I) \tag{5.5}$$

où K_m et K'_m sont des facteurs de normalisation, qui assure que la somme des 2 quantités est égale à 1. Notons que $p(d_{km+i}; O)$ est calculée à partir des probabilités *a priori* $p(d_{km+j}; I)$ des autres bits constituant le même symbole.

Comme le montre la figure (5.2), l'information extrinsèque $p(d_{km+i}; O)$ est désentrelacée et délivrée au décodeur SISO [BDMP97] comme une information *a priori* sur les bits codés. Soit $c_l = d_{\sigma^{-1}(km+i)}$ où σ^{-1} est la permutation sur les indices due à la présence de désentrelaceur ; $p(c_l; I)$ est l'entrée mise à jour du décodeur SISO, qui sera utilisée comme des probabilités *a priori* sur les bits.

Le bloc de décodeur prend en entrée toutes les probabilités extrinsèques des marginales après désentrelacement $p(c_l; I)$ et applique l'algorithme somme-produit [BCJR74] pour produire des nouvelles probabilités a posteriori et extrinsèques sur les bits codés : $p_{APP}(c_l)$ et $p(c_l; O)$. Ces quantités prennent en considération toutes les séquences des observations ainsi que la structure du code.

L'algorithme qui sera utilisé dans les BICM-ID est une version spécifique dans laquelle il n'est pas nécessaire de calculer les probabilités *a posteriori* des bits d'information car cette quantité ne sera pas utilisée dans les récursions (juste à la fin, pour la décision). A la place, il sera nécessaire d'évaluer les probabilités *a posteriori* et les probabilités extrinsèques des bits codés (c'est équivalent puisqu'il existe une correspondance un à un entre les bits d'information et les bits codés). Il opère sur les probabilités extrinsèques désentrelacées $p(c_l; I)$ qui sont utilisées comme des probabilités *a priori* sur les bits codés.

L'évaluation de l'APP correspondante des bits codés peut être écrite d'une manière

très compacte [MDdC02, Muq01] comme suit :

$$p_{APP}(c_l = b) = K_c \sum_{\underline{c} \in \phi_b^l} I_{\mathcal{C}}(\underline{c}) \prod_j p(c_j; I) \tag{5.6}$$

où $I_{\mathcal{C}}(\underline{c})$ représente la fonction indicatrice du code et a l'expression suivante [MDdC02] :

$$I_{\mathcal{C}}(\underline{c}) = \begin{cases} 1 & si \;\; \underline{c} \in \mathcal{C} \\ 0 & sinon \end{cases} \tag{5.7}$$

et ϕ_b^l dénote l'ensemble des mots binaires de longueur L_c à valeur b dans la position l. K_c est une constante de normalisation. La démonstration se trouve dans l'annexe 5.1. En d'autres termes, un algorithme somme produit opère comme suit :

- A partir des quantités a priori disponibles pour chaque bit, il calcule la probabilité de tous les mots possibles ayant la même taille que les mots de code : L_c (un simple produit des quantités *a priori* individuelles).
- La probabilité de chaque mot qui n'est pas un mot de code est remise à zéro.
- L'APP de chaque bit codé est évaluée comme la probabilité marginale sur tous les mots de code restant.

Cela reste juste une caractérisation de l'algorithme d'une manière compacte, et c'est utilisé dans tout le reste de ce chapitre et de la thèse.

La probabilité extrinsèque correspondante est obtenue par une simple application de (5.3), correspondant à la formule générale de passage :

$$p(c_l = b; O) = K_c' \sum_{\underline{c} \in \phi_b^l} I_{\mathcal{C}}(\underline{c}) \prod_{j \neq l} p(c_j; I) \tag{5.8}$$

où K_c' est un facteur de normalisation.

L'information extrinsèque $p(c_l; O)$ est entrelacée et fournie au bloc de démodulateur comme étant une nouvelle information *a priori*. Le processus s'arrête (la convergence est établie) quand les APPs à la sortie des 2 blocs (bloc de démodulateur et celui de décodeur) sont égales (voir annexe 5.2) ou quand un nombre maximal d'itérations est atteint.

Notons que le résultat ci-dessus peut être appliqué sur une large classe de décodeurs itératifs, notamment les turbo décodeurs série ou parallèle et les décodeurs LDPC. La seule condition est que ce processus itératif alterne 2 étapes dans le but de calculer des quantités similaires à (5.5) et (5.8) où $p(d_{km+i} = b|y_k)$ et $I_{\mathcal{C}}$ seront remplacés par des fonctions densité de probabilité compatibles avec le système en considération. Les expressions de ces fonctions densité de probabilité sont données dans [WRJ06] dans les cas particuliers de turbo codes série et parallèle.

Dans ce qui suit, nous allons d'abord proposer une interprétation des blocs de démodulateur et de décodeur basée sur la géométrie de l'information. Nous allons ensuite formuler un critère général basé sur le schéma des BICM (pour sa simplicité). Ce critère est défini pour une séquence, un bonus par rapport à d'autres critères basés sur des évaluations bits [WRJ06].

5.3 BICM et lien avec la géométrie de l'information

Parmi les différentes approches pour l'analyse des modulations codées à bits entrelacés, une approche géométrique a été considérée, elle fournit une interprétation intéressante en termes de projections. Le cas particulier des BICM-ID a été étudié dans [MDdC02] menant à une bonne caractérisation du démodulateur et du décodeur. En effet, Muquet [MDdC02] a proposé une interprétation géométrique pour chacun des 2 blocs de démodulateur et de décodeur pris séparément, cependant une caractérisation du processus entier reste à exploiter. D'autres études considèrent, en se basant sur la géométrie de l'information, le lien entre le turbo décodage itératif et le critère optimal du Maximum a Posteriori (MAP) [Ric00, ITA04].

5.3.1 Interprétation géométrique du bloc de décodeur

Nous avons déjà vu que l'APP d'un bit codé évaluée par le décodeur peut être obtenue par (5.6). D'une manière itérative, nous pourrons ainsi écrire :

$$p_{APP}^{(n)}(c_l = b) = K_c \sum_{\underline{c} \in \phi_b^l} I_{\mathcal{C}}(\underline{c}) \prod_j p^{(n)}(c_j; I) \tag{5.9}$$

Soit $q_{dem}^{(n)}(c) = \prod_j p^{(n)}(c_j; I)$, alors et en résolvant le problème de minimisation suivant :

$$q_{dem}^{*(n)}(c) = argmin_{q \in \mathcal{L}_C} D(q||q_{dem}^{(n)}(c)) \tag{5.10}$$

où $\mathcal{L}_C$ est la famille des distributions compatibles avec le code, nous obtenons la solution suivante :

$$q_{dem}^{*(n)}(c) = K I_C(c) q_{dem}^{(n)}(c) \tag{5.11}$$

En se basant sur les définitions des projections rapportées dans le chapitre 2, nous pouvons conclure que $q_{dem}^{*}(c)$ est la I-projection de $q_{dem}(c)$ sur la famille $\mathcal{L}_C$.
Notons que le résultat du bloc de décodeur n'est autre que la marginalisation de $q_{dem}^{*}(c)$ qui pourra être interprétée comme la projection de $q_{dem}^{*}(c)$ sur la famille $\mathcal{E}_F$ des densités séparables :

$$p_{APP}^{(n)}(c_l = b) = argmin_{q \in \mathcal{E}_F} D(q||q_{dem}^{*(n)}(c)) \tag{5.12}$$

Nous pouvons ainsi conclure que l'APP d'un bit codé évaluée par le bloc de décodeur n'est autre que le résultat de deux blocs de projection, le premier étant la projection de l'extrinsèque fourni par le bloc de démodulateur sur la famille linéaire des distributions compatibles avec le code $\mathcal{L}_C$, et le deuxième étant la projection du résultat de la première projection sur la famille exponentielle des densités séparables $\mathcal{E}_F$ (Voir chapitre 2 pour plus de détails).

5.3.2 Interprétation géométrique du bloc de démodulateur

L'APP d'un bit codé et entrelacé fournie par le démodulateur peut être obtenue par (5.4) dont la version itérative est la suivante :

$$p_{APP}^{(n)}(d_{km+i}=b) = K_m \sum_{s_k \in \psi_b^i} p(y_k|s_k) \prod_j p^{(n-1)}(d_{km+j};I) \tag{5.13}$$

Soit $q_{dec}^{(n-1)}(d) = \prod_j p^{(n-1)}(d_{km+j};I)$. Muquet évalue ainsi $p(y_k|s_k)q_{dec}^{(n-1)}(d)$ à partir des observations du canal, et projette le résultat sur la famille des densités séparables $\mathcal{E}_F$ pour une marginalisation. Cependant l'interprétation de la première étape n'est pas claire dans ses travaux [MDdC02].

Une étude importante [ITA04] vient ainsi pour compléter l'approche de Muquet quant à l'analyse du processus de turbo décodage de point de vue géométrie de l'information. Dans [ITA04], les propriétés du turbo décodage sont étudiées en se basant sur les concepts et les outils de base de la géométrie de l'information, notamment sur les notions de projections sur des familles de distributions de probabilité particulières. Ces études fournissent une compréhension intuitive de l'aspect théorique et donnent une nouvelle approche d'analyse de tels systèmes itératifs.
En effet, dans ces travaux, Ikeda donne une interprétation géométrique du processus de décodage itératif des turbo codes et des codes LDPC, ce qu'on ne retrouve pas dans les approches proposées par Muquet qui se concentre quant à lui sur le décodage itératif des BICM. Il décrit aussi, en se basant sur les interprétations géométriques qu'il a apportées, une analyse de convergence de turbo décodage et un éventuel lien possible avec le critère optimal de Maximum a Posteriori (MAP). Dans ce sens, il fournit l'expression du terme d'erreur entre le turbo décodage itératif et le critère MAP. Vu la complexité de ce terme, Ikeda donne juste l'expression sans rentrer en détails dans les éventuels cas particuliers et le comportement de ce terme [ITA04]. Notons que, dans ces travaux, Ikeda développe la théorie géométrique de Richardson [Ric00] afin d'analyser et d'interpréter le turbo décodage et le décodage LDPC. Il utilise pour cela les même notations que dans [Ric00] basées sur les coordonnées logarithmiques.

En effet, le but essentiel du décodage est de marginaliser la fonction globale de vraisemblance. Cependant, la taille énorme de la séquence d'entrée, ainsi que l'existence de deux codes dans le schéma de turbo décodage rendent la marginalisation de cette fonction assez difficile voire non faisable en pratique. D'où l'idée de considérer séparément chacun de ces deux codes et réaliser le décodage d'une manière séparée via l'algorithme de BCJR [BCJR74]. Dans [ITA04], les auteurs construisent, pour chacun de ces deux blocs, une fonction de vraisemblance à partir des extrinsèques fournis par l'autre bloc (ces extrinsèques servent comme des a priori). Une marginalisation de ces deux fonctions de vraisemblance est ensuite réalisé par l'algorithme BCJR. Cette marginalisation peut être vue comme étant une projection sur la famille exponentielle des densités séparables. Le processus continue ensuite d'une manière itérative, et les deux blocs de décodeurs échangent les informations extrinsèques entre eux pour permettre à chacun des blocs de prendre en compte l'information venant de l'autre et cela jusqu'à la convergence caractérisée par une égalité des deux projections mises en jeu dans ce processus itératif. Ikeda a ensuite utilisé cette interprétation géométrique dans le but de donner une analyse de convergence du turbo décodage et un lien possible avec le critère optimal de maximum a posteriori.
Bien que les deux applications soient différentes, ces travaux proposent une interprétation géométrique du processus de décodage itératif basée sur des projections sur des familles de distributions de probabilité (avec une application portant sur les BICM dans [MDdC02] et une autre portant sur les turbo codes et les codes LDPC dans [ITA04, Ric00]).

Dans la suite, et après avoir formulé le critère d'optimisation, et posé les notations, nous proposerons une interprétation géométrique du processus itératif entier.

5.4 Formulation du critère

Nous avons vu que le turbo décodage n'a pas été à l'origine introduit comme solution d'un problème d'optimisation. Cela rend sa structure précise adhoc et l'analyse de sa convergence et stabilité difficile. Dans ce qui suit, on va essayer de reformuler le turbo décodage itératif comme étant la solution d'un problème d'optimisation bien défini.

5.4.1 Justification de la forme factorisée : la structure de décodeur

Considérons d'abord le problème le plus général de décodage d'une séquence $\underline{d} = (d_1, d_2, \cdots, d_n)$ qui a été fournie par un ensemble de 2 blocs en série. Dans le cas des BICM, le premier bloc est celui de l'encodeur canal et le deuxième bloc est celui du mapping bits-symbole, suivis par une transmission sur un canal, ici considérée comme sans mémoire.
Considérons maintenant le calcul de l'APP séquence.
En particulier, nous considérons les optimisations basées sur les critères de maximum de vraisemblance (ML) et de maximum a posteriori (MAP) qui, théoriquement, peuvent être obtenus tous les deux en réalisant une évaluation exhaustive de probabilité sur l'ensemble de tous les mots de code possibles. Cependant cette évaluation exhaustive possède une complexité exponentielle de point de vue taille de la séquence d'information, ce qui empêche son implémentation en pratique. La plupart des algorithmes de décodage profite de quelques propriétés de l'encodeur (e.g., le fait que l'encodeur est dans la plupart des cas une source de Markov) pour éviter cette recherche exhaustive. Malheureusement, cela est généralement impossible dans le cas des BICM (en effet la source de Markov correspondant à l'encodeur n'est plus observée à travers un canal sans mémoire) [Muq01].

Un décodage optimal dans le sens de ML et du MAP peut cependant être obtenu à partir d'une procédure en trois étapes au lieu d'appliquer directement la recherche exhaustive. En première étape, les APP des symboles connaissant les observations sont évaluées sans prendre en considération la structure du code. Cela assure une complexité linéaire pour l'évaluation des APPs due au fait que seules les probabilités marginales des symboles complexes transmis seront à calculer au lieu de calculer les probabilités de tous les mots de code. La deuxième étape consiste à prendre en considération la structure du code en projetant les probabilités obtenues à partir de la première étape sur l'ensemble des distributions de probabilité compatible avec la structure du code afin d'éliminer tous les mots n'appartenant pas au dictionnaire des mots de code. Une maximisation finale constitue la dernière étape de cette procédure [MDdC02].

Notons que cette procédure demeure non réalisable en pratique vu que la complexité exponentielle est juste déplacée à la procédure de projection mais elle permet de nous guider vers l'obtention d'une implémentation pratique réalisable.
Cela justifie la nécessité de quelques suppositions et approximations pour faciliter cette procédure.

Travailler sur les mots de code étant difficile (complexité de calcul), nous gardons la même approche mais cette fois ci en travaillant sur les marginales (notons qu'à la fin les 2 blocs échangent les marginales). Nous supposons donc que toutes les quantités évaluées ou propagées d'un bloc à un autre sont des quantités "bits".

Maintenant, puisque nous avons choisi de travailler uniquement sur des quantités "bits" (marginales), le résultat attendu reste sous-optimal sur toute la séquence. Cependant, nous pouvons supposer que les APP bits sont évaluées d'une manière optimale par le bloc correspondant (le démodulateur et le décodeur respectivement). Les a priori à l'entrée d'un bloc seront des quantités fournies par l'autre bloc, et sont supposées aussi factorisables.

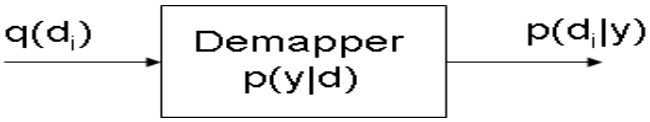

FIG. 5.3 – Bloc de démodulateur pris séparément

Nous savons que, étant donné un certain a priori $q(\underline{d})$ et les mesures du canal, un démodulateur optimal (dans le sens du MAP) comme représenté sur la figure (5.3) évalue la probabilité a posteriori suivante :

$$p(d_i|y) = \sum_{\underline{d}:d_i} p(\underline{d}|y) = \sum_{\underline{d}:d_i} \frac{p(y|\underline{d})q(\underline{d})}{p(y)} \sim \sum_{\underline{d}:d_i} p(y|\underline{d})q(\underline{d}) \tag{5.14}$$

où $q(\underline{d})$ est l'a priori sur la séquence.

La structure du décodeur étant choisie d'une manière à ce que les a priori sont des quantités factorisables, nous aurons :

$$q(\underline{d}) = \prod_i q(d_i) \tag{5.15}$$

Finalement, la probabilité a posteriori à la sortie du bloc de démodulateur sera :

$$p(d_i|y) \sim \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_i q(d_i) = q(d_i) \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j \neq i} q(d_j) \tag{5.16}$$

Avec comme notation :

$$g_{d_i}(q) = \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j \neq i} q(d_j) \tag{5.17}$$

Notons que $g_{d_i}(q)$ est indépendante de $q(d_i)$

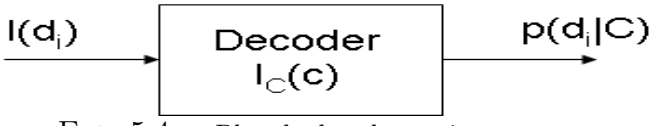

FIG. 5.4 – Bloc de décodeur pris séparément

En ce qui concerne le bloc du décodeur, nous avons vu dans la section précédente que, étant donné un certain a priori $l(\underline{d})$ et la structure du code, un décodeur optimal (dans le sens du MAP) comme représenté dans la figure (5.4) évalue la probabilité a postériori suivante :

$$p(d_i|C) = \sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c}) l(\underline{d}) \tag{5.18}$$

qui pourra être interprétée comme suit :

- Evaluation de la probabilité de tous les mots possibles : la structure de démodulateur étant choisie afin que les a priori soient des quantités factorisables, nous aurons :

$$l(\underline{d}) = \prod_i l(d_i) \tag{5.19}$$

- Garder uniquement les mots de code, en éliminant les mots n'appartenant pas dans le dictionnaire du code. Cela consiste à multiplier la probabilité obtenue précédemment par la fonction indicatrice du code.
- Prendre la marginale sur le bit i puisque nous avons supposé que nous propageons uniquement les marginales entre les 2 blocs.

Avec comme notation :

$$f_{d_i}(l) = \sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c}) \prod_{j \neq i} l(d_j) \tag{5.20}$$

Notons que $f_{d_i}(l)$ est indépendante de $l(d_i)$

Le bloc de démodulateur dispose des observations du canal, cependant, le bloc de décodeur dispose des informations concernant la structure du code.

Nous pourrons dire que chaque structure prise séparément réalise le MAP sur les marginales.

Ouvrons ici une parenthèse quant à l'interprétation géométrique du processus de décodage itératif des BICM. Nous pouvons ainsi réécrire les 2 blocs de démodulateur et de décodeur comme étant la solution des problèmes d'optimisation suivants :
Pour le bloc de démodulateur :

$$\begin{cases} \min\limits_{p(d_k)} \quad D(p(d_k)||q^{(n)}(d_k)) \\ \\ s.c. \quad E_{p(d_k)}(\log g_{d_k}(q^{(n)})) = K_1^{(n)} \end{cases} \tag{5.21}$$

Pour le bloc de décodeur :

$$\begin{cases} \min\limits_{p(d_k)} \quad D(p(d_k)||l^{(n+1)}(d_k)) \\ \\ s.c. \quad E_{p(d_k)}(\log f_{d_k}(l^{(n+1)})) = K_2^{(n)} \end{cases} \tag{5.22}$$

Nous pouvons ainsi conclure que l'APP de chacun des 2 blocs pourra être considéré comme le résultat de projection de l'extrinsèque correspondant sur une famille linéaire bien définie (Voir chapitre 2 pour plus de détails sur la structure d'une projection sur une famille linéaire). Cependant et comme le montrent les équations, ces familles linéaires varient avec les itérations. Donc, nous n'aurons pas une définition explicite de ces familles, problème que nous avons rencontré avec l'interprétation géométrique de l'algorithme de Blahut-Arimoto dans le chapitre précédent et que nous avons résolue en se basant sur une approche point proximal. C'est pour cela que nous étudierons une approche point proximal du processus de décodage itératif des BICM dans la suite de ce chapitre.

En se basant sur ce qui précède, nous voulons maintenant que les 2 probabilités a posteriori évaluées séparément par chacun des 2 blocs soient les mêmes à la convergence :

$$q(d_i)\sum_{\underline{d}:d_i} p(y|\underline{d})\prod_{j\neq i} q(d_j) = l(d_i)\sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c})\prod_{j\neq i} l(d_j) \tag{5.23}$$

Il en suit que le critère de notre problème revient à trouver les deux a priori $l(\underline{d})$ et $q(\underline{d})$ de telle manière à ce que la probabilité a posteriori séquence à la sortie du bloc de démodulateur soit égale à celle à la sortie du bloc de décodeur.

Posons notre problème d'optimisation comme suit :

$$\begin{cases} \min\limits_{l(d_k)_{1\leq k\leq n}\in\mathbb{R}^+, q(d_k)_{1\leq k\leq n}\in\mathbb{R}^+} \sum\limits_{k=1}^{n} D(\alpha_k l(d_k) f_{d_k}(l)||\beta_k q(d_k) g_{d_k}(q)) \\ \sum\limits_{d_k} \alpha_k l(d_k) f_{d_k}(l) = 1 \quad 1\leq k\leq n \\ \sum\limits_{d_k} \beta_k q(d_k) g_{d_k}(q) = 1 \quad 1\leq k\leq n \end{cases} \tag{5.24}$$

Nous voulons essayer de trouver le processus itératif que nous devons utiliser dans le but de minimiser le critère (5.24).
Une étude détaillée de cette approche se trouve en annexe 5.3 expliquant les démarches que nous avons suivies pour résoudre ce problème d'optimisation, ainsi que les limites de cette approche.

La difficulté de l'approche précédente réside dans la recherche des équations de mise à jour des variables $l(d_i)$ et $q(d_i)$: en effet, une expression analytique est très difficile à trouver (il en est de même pour le choix de μ) vu que dans la même équation on a $l(d_i)$ et $f_{(d_k)}(l)$ qui dépend de $l(d_i)$. Ainsi, pour chaque $l(d_i)$ évalué par le bloc de démodulateur, il nous faut itérer sur le bloc du décodeur. Ce qui augmentera la complexité du problème surtout que le bloc du décodeur fait appel à l'algorithme de BCJR.
L'idée étant alors de partir d'un autre critère, qui à la convergence, sera équivalent au critère demandé.

En effet, nous avons déjà vu que les APP à la sortie des 2 blocs (démodulateur et décodeur) doivent être égales à la convergence :

$$q(d_i)\sum_{\underline{d}:d_i} p(y|\underline{d})\prod_{j\neq i} q(d_j) = l(d_i)\sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c})\prod_{j\neq i} l(d_j) \tag{5.25}$$

Notons qu'à la convergence, le critère dans (5.24) tend vers 0. Dans ce cas, nous aurons que $q(d_i)g_{d_i}(q) \rightarrow l(d_i)f_{d_i}(l)$.

Cela pourra être assuré en choisissant :

$$l(d_i) = \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j \neq i} q(d_j) \tag{5.26}$$

et

$$q(d_i) = \sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c}) \prod_{j \neq i} l(d_j) \tag{5.27}$$

Ce qu'on veut donc avoir est, qu'à la convergence, les égalités suivantes seront vérifiées pour tous les bits i, $1 \leq i \leq n$:

$$\begin{array}{c} l(d_i)q(d_i) \sim q(d_i) \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j \neq i} q(d_j) \sim l(d_i) \sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{c}) \prod_{j \neq i} l(d_j) \\ \sim \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j \neq i} q(d_j) \sum_{\underline{d}':d_i'} I_{\mathcal{C}}(\underline{c}) \prod_{j \neq i} l(d_j) \end{array} \tag{5.28}$$

C'est pour cette raison qu'on a choisi à définir un critère basé sur la minimisation de cette distance de Kullback-Leibler :

$$\sum_{i=1}^{n} D(\alpha_i q(d_i) l(d_i) || \alpha_i' g_{d_i}(q) f_{d_i}(l)) \tag{5.29}$$

qui est équivalent à :

$$D(\alpha l(\underline{d}) q(\underline{d}) || \alpha' f(\underline{l}) g(\underline{q})) \tag{5.30}$$

où $f(\underline{l})g(\underline{q})$ est une densité séparable dont les marginales sont $f_{d_i}(l)g_{d_i}(q)$; α et α' sont des facteurs de normalisation.

Sachant que le critère dans (5.24) tend vers 0 à la convergence, il en suit que le critère qu'on a choisi est équivalent au critère initial et qu'on pourra continuer à travailler sur ce critère reformulé sans aucun problème.
Le bonus qu'apporte ce critère reformulé par rapport à (5.24) est qu'il est symétrique par rapport aux deux variables l et q (ce qui nétait pas le cas avec le critère initial).

Notre critère reformulé pourra être donc défini selon le problème d'optimisation

suivant :

$$\begin{cases} \min\limits_{l(d_k)_{1\leq k\leq n}\in\mathbb{R}^+, q(d_k)_{1\leq k\leq n}\in\mathbb{R}^+} \sum\limits_{k=1}^{n} D(\alpha_k q(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q) f_{d_k}(l)) \\ \sum\limits_{d_k} \alpha_k q(d_k) l(d_k) = 1 \quad 1 \leq k \leq n \\ \sum\limits_{d_k} \alpha_k' g_{d_k}(q) f_{d_k}(l) = 1 \quad 1 \leq k \leq n \end{cases} \tag{5.31}$$

Essayons donc de trouver le processus itératif que nous devons adopter dans le but de minimiser le critère dans (5.31).

Le Lagrangien de ce problème est :

$$\begin{aligned} \mathcal{L} = &\sum_k \sum_{d_k} \alpha_k q(d_k) l(d_k) \log(\alpha_k q(d_k) l(d_k)) - \sum_k \sum_{d_k} \alpha_k q(d_k) l(d_k) \log(\alpha_k^{'} g_{d_k}(q) f_{d_k}(l)) \\ &- \sum_k \lambda_k (\sum_{d_k} \alpha_k q(d_k) l(d_k) - 1) - \sum_k \mu_k (\sum_{d_k} \alpha_k' g_{d_k}(q) f_{d_k}(l) - 1) \end{aligned} \tag{5.32}$$

où λ_k et μ_k sont les multiplieurs de Lagrange correspondant. Vu que nous voulons évaluer le gradient du Lagrangien par rapport à $l(d_i)$ (ce calcul sera suffisant pour connaitre le gradient par rapport à $q(d_i)$ grâce à la symétrie de l'expression du Lagrangien) nous allons utiliser cette expression équivalente du Lagrangien :

$$\mathcal{L} = A - B_i - \sum_{k \neq i} B_k - C - D \tag{5.33}$$

où

$$A = \sum_k \sum_{d_k} \alpha_k q(d_k) l(d_k) \log(\alpha_k q(d_k) l(d_k)) \tag{5.34}$$

$$B_k = \sum_{d_k} \alpha_k q(d_k) l(d_k) \log(\alpha_k^{'} g_{d_k}(q) f_{d_k}(l)) \tag{5.35}$$

$$C = \sum_k \lambda_k (\sum_{d_k} \alpha_k q(d_k) l(d_k) - 1) \tag{5.36}$$

$$D = \sum_k \mu_k (\sum_{d_k} \alpha_k' g_{d_k}(q) f_{d_k}(l) - 1) \tag{5.37}$$

Le gradient de chaque terme par rapport à $l(d_i)$ est donné par :

$$\frac{\partial A}{\partial l(d_i)} = \alpha_i q(d_i) \log(\alpha_i q(d_i) l(d_i)) + \alpha_i q(d_i) \tag{5.38}$$

$$\frac{\partial B_i}{\partial l(d_i)} = \alpha_i q(d_i) \log\left(\alpha_i^{'} f_{d_i}(l) g_{d_i}(q)\right) \tag{5.39}$$

$$\frac{\partial C}{\partial l(d_i)} = \lambda_i \alpha_i q(d_i) \tag{5.40}$$

En montrant la dépendance de B_k en $l(d_i)$ d'une manière explicite, nous obtenons :

$$B_k = \sum_{d_k} \alpha_k q(d_k) l(d_k) \log\left(\alpha_k^{'} \sum_{\underline{d}:d_k} I_{\mathcal{C}}(\underline{d}) l(d_i) \prod_{j \neq k,i} l(d_j) \sum_{\underline{d}^{'}:d_k^{'}} p(y|\underline{d}) \prod_{j \neq k} q(d_j^{'})\right) \tag{5.41}$$

Nous déduisons alors l'expression de son gradient par rapport à $l(d_i)$:

$$\frac{\partial B_k}{\partial l(d_i)} = \sum_{d_k} \alpha_k q(d_k) l(d_k) \frac{\alpha_k^{'} \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)}{\alpha_k^{'} \sum_{\underline{d}:d_k} I_{\mathcal{C}}(\underline{d}) l(d_i) \prod_{j \neq k,i} l(d_j)} \tag{5.42}$$

Le gradient de D est donné par :

$$\frac{\partial D}{\partial l(d_i)} = \sum_{k \neq i} \mu_k \sum_{d_k} \alpha_k^{'} g_{d_k}(q) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j) \tag{5.43}$$

Le but étant de trouver $l(d_i)$ telle que $\frac{\partial \mathcal{L}}{\partial l(d_i)} = 0$. La variable $l(d_i)$ apparait dans le dénominateur de 2 fractions dans (5.41) et aussi dans la fonction dans (5.34). Pour cela une expression analytique de la variable optimale $l(d_i)$ reste difficile à trouver. Résoudre ce problème d'une manière itérative nécessite l'évaluation de $n-1$ expressions du type $\sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)$. Cette évaluation demande une modification de l'algorithme classique de BCJR. Ensuite, un algorithme de point fixe pourra être utilisé pour obtenir la valeur de $l(d_i)$. Une nouvelle approximation sera nécessaire pour résoudre ce problème d'optimisation sans augmenter la complexité de calcul : utilisons donc la technique de

One Step Late. L'algorithme de One Step Late proposé par Green [Gre90] est une technique où l'on remplace la variable par sa valeur calculée à l'itération précédente. Cette substitution sera effectuée dans l'expression du gradient, où on remplace uniquement les termes qui nous empêchent de trouver une expression analytique de notre variable optimale.

Nous remplaçons alors dans le dénominateur de cette dernière expression $l(d_i) = l^{(n)}(d_i)$ où $(n+1)$ est l'indice de l'itération courante. Pour évaluer $l(d_i)$, toutes les autres variables de l'itération courante sont supposés égales à leurs valeurs correspondantes de l'itération précédente. Nous savons que, à la convergence, le point obtenu est un point stationnaire du critère initial. Cependant, en utilisant la technique de One Step Late la convergence n'est pas garantie puisqu'on utilise quelque part les valeurs des variables de l'itération précédente. Cela reste compatible avec ce qu'on observe dans le cas des Turbo codes [KLM+06].

Nous supposons aussi que : $q^{(n)}(d_k) \sim \sum_{\underline{d}:d_k} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k} l^{(n)}(d_j)$ pour $1 \leq k \leq n$

Si nous obtenons, pour $l(d_i)$, la même expression (avec la probabilité canal à la place de la fonction indicatrice du code, et $q^{(n)}(d_j)$ à la place de $l^{(n)}(d_j)$), notre récurrence sera correcte et nous obtenons exactement l'expression du décodage itératif classique des BICM.

En se basant sur ces suppositions, nous pouvons écrire le gradient de B_k comme suit :

$$\frac{\partial B_k}{\partial l(d_i)}(\underline{l}^{(n)}, \underline{q}^{(n)}) = \sum_{d_k} \alpha_k q^{(n)}(d_k) l^{(n)}(d_k) \frac{\sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l^{(n)}(d_j)}{\sum_{\underline{d}:d_k} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k} l^{(n)}(d_j)} \tag{5.44}$$

Au dénominateur de cette expression, nous avons $q^{(n)}(d_k)$ et le gradient devient :

$$\frac{\partial B_k}{\partial l(d_i)}(\underline{l}^{(n)}, \underline{q}^{(n)}) = \sum_{d_k} \alpha_k l^{(n)}(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l^{(n)}(d_j) \tag{5.45}$$

Finalement nous obtenons :

$$\frac{\partial B_k}{\partial l(d_i)}(\underline{l}^{(n)}, \underline{q}^{(n)}) = \alpha_k q^{(n)}(d_i) \tag{5.46}$$

En suivant les même démarches, nous obtenons pour D,

$$\frac{\partial D}{\partial l(d_i)}(\underline{l}^{(n)}, \underline{q}^{(n)}) = q^{(n)}(d_i) \sum_{k \neq i} \mu_k \alpha'_k \tag{5.47}$$

Chacun des termes du gradient de lagrangien étant évalué, nous pouvons alors écrire l'expression complète du gradient comme suit :

$$\frac{\partial \mathcal{L}}{\partial l(d_i)} = \alpha_i q^{(n)}(d_i) \Big(\log(\alpha q^{(n)}(d_i) l(d_i)) - \log(\alpha^{'} f_{d_i}(l^{(n)}) g_{d_i}(q^{(n)})) - Z \Big) \tag{5.48}$$

où Z est une constante. Nous voulons trouver la probabilité $l(d_i)$ telle que $\frac{\partial \mathcal{L}}{\partial l(d_i)} = 0$, nous obtenons ainsi :

$$\alpha_i q^{(n)}(d_i) \Big(\log(\alpha q^{(n)}(d_i) l(d_i)) - \log(\alpha^{'} f_{d_i}(l^{(n)})) g_{d_i}(q^{(n)}) - Z \Big) = 0 \tag{5.49}$$

Cela mène à :

$$l^{(n+1)}(d_i) = K \sum_{\underline{d}: d_i} p(y|\underline{d}) \prod_{j \neq i} q^{(n)}(d_j) \tag{5.50}$$

où K est une constante de normalisation.

Nous remarquons que la même expression que celle du décodage itératif du schéma des BICM est obtenue comme le montre (5.5) où $l(d_i)$ est équivalente à l'extrinsèque fournie par le bloc de démodulateur ($p(d_{km+i=b}; O)$) et $q(d_i)$ à l'extrinsèque évaluée par le bloc de décodeur ($p(c_l = b; O)$).
Nous avons montré aussi que la solution classique du décodage itératif vérifie bien notre critère reformulé après utilisation de la technique de One Step late.

5.5 approche proximale : critère modifié

L'idée principale de la section précédente était l'usage de la technique One Step Late, où on remplace, dans l'expression du gradient, les termes qui nous empêchent d'obtenir une expression analytique de notre variable optimale par leur valeur de l'itération précédente. Nous avons ainsi montré que, en utilisant cette approche, nous obtenons exactement la même expression pour l'extrinsèque que celle du décodage itératif clas-

sique des BICM.

Notre idée maintenant consiste à introduire convenablement les itérations dans notre critère dans le sens du One Step Late tout en vérifiant toujours les solutions classiques des BICM-ID.

Utilisons les mêmes notations que celles de la section précédente.
Nous supposons que (n+1) correspond à l'indice de l'itération courante, et que nous disposons déjà des quantités suivantes : $l^{(n)}(d_k)$, $q^{(n)}(d_k)$, $f_{d_k}(l^{(n)})$ et $g_{d_k}(q^{(n)})$. Nous allons ensuite essayer de calculer alternativement $l^{(n+1)}(d_k)$ et $q^{(n+1)}(d_k)$ selon le critère suivant qu'on veut minimiser :

$$\begin{cases} \min\limits_{l(d_k)_{1\leq k\leq n}\in\mathbb{R}^+, q(d_k)_{1\leq k\leq n}\in\mathbb{R}^+} \sum\limits_{k=1}^{n} D(\alpha_k q(d_k) l(d_k) || \alpha'_{n,k} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})) \\ \sum\limits_{d_k} \alpha_k q(d_k) l(d_k) = 1 \quad 1 \leq k \leq n \\ \sum\limits_{d_k} \alpha'_{n,k} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)}) = 1 \quad 1 \leq k \leq n \end{cases} \tag{5.51}$$

Sans entrer dans les détails de calculs, nous pouvons montrer que la solution de ce problème d'optimisation est :

$$l^{(n+1)}(d_i) = g_{d_i}(q^{(n)}) = K_l \sum_{\underline{d}:d_i} p(y|\underline{d}) \prod_{j\neq i} q^{(n)}(d_j) \tag{5.52}$$

et

$$q^{(n+1)}(d_i) = f_{d_i}(l^{(n+1)}) = K_q \sum_{\underline{d}:d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j\neq i} l^{(n+1)}(d_j) \tag{5.53}$$

qui est exactement la solution du décodage itératif classique des BICM.

Ensuite l'idée étant de modifier l'algorithme dans le but de minimiser la distance de Kullback que nous voulons bien minimiser. En effet, nous pouvons écrire cette équation de passage entre le critère qu'on veut minimiser et le critère initial :

$$\begin{gathered} D(\alpha_k q(d_k) l(d_k) || \alpha'_{n,k} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})) = \\ D(\alpha_k q(d_k) l(d_k) || \alpha'_k g_{d_k}(q) f_{d_k}(l)) + \textstyle\sum_{d_k} \alpha_k q(d_k) l(d_k) \log \frac{\alpha'_k g_{d_k}(q) f_{d_k}(l)}{\alpha'_{n,k} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})} \end{gathered} \tag{5.54}$$

Un lien avec la méthode du point proximal semble être intéressant. En effet, pour que cette approche ressemble à un algorithme du point proximal, il faut que le terme de compensation
$\sum_{d_k} \alpha_k q(d_k) l(d_k) \log \frac{\alpha_k^{'} g_{d_k}(q) f_{d_k}(l)}{\alpha_{n,k}^{'} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})}$ soit > 0 et nul si et seulement si $l(d_k) = l^{(n)}(d_k)$ et $q(d_k) = q^{(n)}(d_k)$ afin de garantir que cette divergence de Kullback-Leibler $D(\alpha_k q(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q) f_{d_k}(l))$ décroit au fil des itérations.

Notons que ce terme vérifie bien la deuxième condition, cependant la condition de non négativité n'est pas toujours vérifiée. D'où l'idée de rajouter un terme dans le sens de One Step Late qui va assurer que la nouvelle solution reste toujours au voisinage de l'ancienne solution. Ce terme aura l'expression suivante :

$$\mu_n \sum_{d_k} \alpha_k q(d_k) l(d_k) \log \frac{\alpha_k q(d_k) l(d_k)}{\alpha_{n,k} q^{(n)}(d_k) l^{(n)}(d_k)} \tag{5.55}$$

où μ_n est le pas qui va nous donner un degrés de liberté supplémentaire pour assurer la condition de non négativité. Le choix de μ_n consiste à rendre le terme proximal toujours non négatif.

Ce terme proximal a l'expression suivante :

$$\begin{array}{c} P(\mu_n, l(d_k), q^{(n)}(d_k)) = \mu_n \sum_k D(\alpha_k^{(n)} l(d_k) q^{(n)}(d_k) || \alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k)) + \\ \sum_k \sum_{d_k} \alpha_k^{(n)} l(d_k) q^{(n)}(d_k) \log \frac{\alpha_k^{'(n)} f_{d_k}(l) g_{d_k}(q^{(n)})}{\alpha_{n,k}^{'} f_{d_k}(l^{(n)}) g_{d_k}(q^{(n)})} \end{array} \tag{5.56}$$

C'est important de noter que $P(\mu_n, l(d_k) = l^{(n)}(d_k), q(d_k) = q^{(n)}(d_k)) = 0$ et qu'à la convergence on minimise exactement ce qu'on voulait minimiser.

Basant sur cette approche, nous pouvons reformuler notre problème de minimisation comme suit :

$$\begin{cases} \min\limits_{l(d_k)_{1\leq k\leq n}\in\mathbb{R}^+, q(d_k)_{1\leq k\leq n}\in\mathbb{R}^+} \sum\limits_{k=1}^{n} D(\alpha_k q(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q) f_{d_k}(l)) + P(\mu_n, l(d_k), q(d_k)) \\ \sum\limits_{d_k} \alpha_k q(d_k) l(d_k) = 1 \quad 1 \leq k \leq n \\ \sum\limits_{d_k} \alpha_k^{'} g_{d_k}(q) f_{d_k}(l) = 1 \quad 1 \leq k \leq n \end{cases} \tag{5.57}$$

Afin de résoudre ce problème d'optimisation, nous considérons d'abord le cas où $q(d_k) = q^{(n)}(d_k)$ est donné et essayons de calculer $l(d_k) = l^{(n+1)}(d_k)$, nous obtenons ainsi :

$$(\alpha_k^{(n)} l(d_k) q^{(n)}(d_k))^{\mu_n+1} = (K_l^{(n)})^{(\mu_n+1)} (\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k))^{\mu_n} (\alpha_{n,k}^{'} f_{d_k}(l^{(n)}) g_{d_k}(q^{(n)})) \tag{5.58}$$

où $K_l^{(n)}$ est une constante de normalisation.

Ensuite, nous utilisons cette nouvelle valeur de $l(d_k)$ dans le but de calculer la nouvelle valeur $q^{(n+1)}(d_k)$ de $q(d_k)$. Nous obtenons ainsi l'expression suivante :

$$\begin{gathered} (\alpha_k^{(n+1)} l^{(n+1)}(d_k) q^{(n+1)}(d_k))^{\mu_n+1} = \\ (K_q^{(n+1)})^{(\mu_n+1)} (\alpha_k^{(n)} l^{(n+1)}(d_k) q^{(n)}(d_k))^{\mu_n} (\alpha_k^{'(n)} f_{d_k}(l^{(n+1)}) g_{d_k}(q^{(n)})) \end{gathered} \tag{5.59}$$

où $K_q^{(n+1)}$ est une constante de normalisation.

Notons qu'avec ce critère modifié, nous avons réussi à obtenir des expressions analytiques de mise à jour de nos variables d'optimisation. Un bonus par rapport à l'approche basée sur le critère dans (5.24) présentée dans la section précédente.

Notons que, pour $\mu_n = 0$, nous obtenons à nouveau les solutions du décodage itératif classique des BICM :

$$l^{(n+1)}(d_k) = g_{d_k}(q^{(n)}) \tag{5.60}$$

et

$$q^{(n+1)}(d_k) = f_{d_k}(l^{(n+1)}) \tag{5.61}$$

Une fois que nous avons trouvé des expressions analytiques pour la mise à jour des variables, l'idée maintenant est d'analyser le terme proximal dans le but de trouver une valeur de μ_n telle que $P(\mu_n, l(d_k), q^{(n)}(d_k)) > 0$ pour assurer que notre distance de Kullback $D(\alpha_k q^{(n)}(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l))$ décroit bien au fil des itérations :

En effet, nous avons l'inégalité suivante :

$$\begin{aligned} &D(\alpha_k q^{(n)}(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l)) + P(\mu_n, l(d_k), q(d_k)) \leq \\ &D(\alpha_k q^{(n)}(d_k) l^{(n)}(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})) + P(\mu_n, l(d_k) = l^{(n)}(d_k), q(d_k) = q^{(n)}(d_k)) \end{aligned} \tag{5.62}$$

Ayant $P(\mu_n, l(d_k) = l^{(n)}(d_k), q(d_k) = q^{(n)}(d_k)) = 0$ alors

$$\begin{aligned} &D(\alpha_k q^{(n)}(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l)) + P(\mu_n, l(d_k), q(d_k)) \leq \\ &D(\alpha_k q^{(n)}(d_k) l^{(n)}(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})) \end{aligned} \tag{5.63}$$

Enfin, si nous choisissons μ_n de manière à ce que le terme proximal $P(\mu_n, l(d_k), q(d_k))$ soit toujours non négatif, nous garantissons que

$$D(\alpha_k q^{(n)}(d_k) l(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l)) \leq D(\alpha_k q^{(n)}(d_k) l^{(n)}(d_k) || \alpha_k^{'} g_{d_k}(q^{(n)}) f_{d_k}(l^{(n)})) \tag{5.64}$$

Cependant la difficulté de cette approche réside dans le choix de μ_n qui va rendre notre terme proximal > 0. En effet, si on se place dans le cas du bloc de démodulateur (calcul de l) et que l'on essaie d'écrire la condition que doit vérifier μ_n pour rendre le terme proximal correspondant > 0, on remarque la présence à la fois de $l^{(n+1)}(d_k)$ et de $f_{d_k}(l^{(n+1)})$. Cela consiste donc à itérer sur le décodeur SISO à chaque fois où on veut trouver un μ_n, ce qui augmente énormément la complexité de calcul.

D'où l'idée de travailler séparément sur chacun des 2 blocs dans le but de trouver pour chacun une interprétation point proximal au lieu de trouver une approche point proximal globale pour les 2 blocs (ce que nous avons essayé de faire jusqu'à présent mais nous n'avons pas réussi)

5.6 Approche point proximal : blocs traités séparément

Nous pouvons résumer le processus du décodage itératif des BICM comme étant la résolution du problème de minimisation suivant :
Au niveau du démodulateur

$$\min_{l(d_k)} \sum_k D(\alpha_k l(d_k) q(d_k) || \beta_k q(d_k) g_{d_k}(q)) \tag{5.65}$$

Au niveau du décodeur

$$\min_{q(d_k)} \sum_k D(\alpha_k l(d_k) q(d_k) || \gamma_k l(d_k) f_{d_k}(l)) \tag{5.66}$$

Une solution est satisfaisante si elle répond aux deux critères simultanément.

Cependant la minimisation de l'un de ces critères n'entraine pas forcément la diminution de l'autre critère à l'itération suivante. La méthode du point proximal permet de faire le lien entre les deux critères via le terme de pénalité qu'elle introduit. Pour cela nous introduisons un terme de pénalité qui fait le lien avec les valeurs obtenues à l'itération précédente. Nous obtenons alors un nouveau processus de minimisation :
Au niveau du démodulateur :

$$\begin{aligned} l^{(n+1)}(d_k) &= \min_{l(d_k)} C_{dem}(l, q^{(n)}, l^{(n)}) \\ = \min_{l(d_k)} &\sum_k D(\alpha_k l(d_k) q^{(n)}(d_k) || \beta_k q^{(n)}(d_k) g_{d_k}(q^{(n)})) + \\ \mu_m^{(n)} &\sum_k D(\alpha_k l(d_k) q^{(n)}(d_k) || \alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k)) \end{aligned} \tag{5.67}$$

Au niveau du décodeur :

$$\begin{aligned} q^{(n+1)}(d_k) &= \min_{q(d_k)} C_{dec}(q, q^{(n)}, l) \\ = \min_{q(d_k)} &\sum_k D(\alpha'_k l(d_k) q(d_k) || \gamma_k l(d_k) f_{d_k}(l)) + \\ \mu_c^{(n)} &\sum_k D(\alpha'_k l(d_k) q(d_k) || \alpha'_{n,k} l(d_k) q^{(n)}(d_k)) \end{aligned} \tag{5.68}$$

Les équations de mise à jour des variables optimales de ces 2 problèmes de minimisation sont données par :

Bloc du démodulateur :

$$(\alpha_i l^{(n+1)}(d_i)q^{(n)}(d_i))^{\mu_m^{(n)}+1} = K_i(\alpha_{n,i}l^{(n)}(d_i)q^{(n)}(d_i))^{\mu_m^{(n)}}\beta_i q^{(n)}(d_i)g_{d_i}(q^{(n)}) \tag{5.69}$$

Notons que, pour $\mu_m^{(n)} = 0$, on retrouve la solution classique du décodage itératif : $l^{(n+1)}(d_i) = g_{d_i}(q^{(n)})$

Bloc du décodeur :

$$(\alpha_i l^{(n+1)}(d_i)q^{(n+1)}(d_i))^{\mu_c+1} = K_i^{'}(\alpha_{n,i}l^{(n+1)}(d_i)q^{(n)}(d_i))^{\mu_c}\gamma_i l^{(n+1)}(d_i)f_{d_i}(l^{(n+1)}) \tag{5.70}$$

Notons aussi que, pour $\mu_c^{(n)} = 0$, on retrouve la solution classique du décodage itératif : $q^{(n+1)}(d_i) = f_{d_i}(l^{(n+1)})$.

A la convergence, on retrouve les mêmes points stationnaires que pour le processus de décodage itératif classique. Pour assurer la décroissance des fonctions de coût, nous avons choisis $\mu_m^{(n)}$ et $\mu_c^{(n)}$ en respectant les conditions suivantes :

$$\begin{aligned} C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) &\leq C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) \\ C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) &\leq C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \end{aligned} \tag{5.71}$$

Considérons les notations suivantes afin de simplifier la représentation des équations. Posons

$$E = \sum_k D(\alpha_k l^{(n+1)}(d_k)q^{(n)}(d_k)||\alpha_{n,k}l^{(n)}(d_k)q^{(n)}(d_k)) \tag{5.72}$$

et

$$F = \sum_k D(\alpha_k l^{(n+1)}(d_k)q^{(n)}(d_k)||\beta_k q^{(n)}(d_k)g_{d_k}(q^{(n)})) \tag{5.73}$$

Ayant $C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \leq C_{dec}(q^{(n)}, q^{(n)}, l^{(n)})$, nous pouvons ainsi écrire :

$$F + \mu_m^{(n)}E \leq \sum_k D(\alpha_{n,k}l^{(n)}(d_k)q^{(n)}(d_k)||\gamma_k l^{(n)}(d_k)f_{d_k}l^{(n)}) \tag{5.74}$$

Nous obtenons ainsi une borne supérieure du pas $\mu_m^{(n)}$ selon :

$$\mu_m^{(n)} \leq \frac{\sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) - F}{E} \tag{5.75}$$

Notons, que dans cette expression, tous les termes peuvent être évalués à partir des quantités obtenues à l'itération (n). Ce qui rend facile le choix de $\mu_m^{(n)}$.

Dans les simulations, $\mu_m^{(n)}$ prend comme valeur cette borne supérieure :

$$\mu_m^{(n)} = \frac{\sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) - F}{E} \tag{5.76}$$

D'autre part, en travaillant sur la deuxième inégalité $C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)})$, nous pourrons trouver une borne supérieure pour $\mu_c^{(n)}$ de la même manière que pour $\mu_m^{(n)}$. En effet, à partir de cette inégalité nous pouvons écrire :

$$H + \mu_c^{(n)} G \leq F + \mu_m^{(n)} E \tag{5.77}$$

avec

$$G = \sum_k D(\alpha'_k l^{(n+1)}(d_k) q^{(n+1)}(d_k) || \alpha'_{n,k} l^{(n+1)}(d_k) q^{(n)}(d_k)) \tag{5.78}$$

et

$$H = \sum_k D(\alpha'_k l^{(n+1)}(d_k) q^{(n+1)}(d_k) || \gamma_k l^{(n+1)}(d_k) f_{d_k}(l^{(n+1)})) \tag{5.79}$$

Nous pourrons ainsi trouver une borne supérieure de $\mu_c^{(n)}$ selon :

$$\mu_c^{(n)} \leq \frac{F - H + \mu_m^{(n)} E}{G} \tag{5.80}$$

Dans les simulations, $\mu_c^{(n)}$ prend comme valeur cette borne supérieure :

$$\mu_c^{(n)} = \frac{F - H + \mu_m^{(n)} E}{G} \tag{5.81}$$

En itérant (5.69) et (5.70) avec $\mu_c^{(n)}$ et $\mu_m^{(n)}$ choisis correctement, une modulation 16QAM,

un code convolutif [5 7], et un nombre de bits d'information de 400, nous obtenons un algorithme qui converge vers les mêmes points que le décodage itératif classique : des résultats de simulation (fig. 5.5) montrent que l'algorithme de décodage itératif classique et celui basé sur une interprétation point proximal ont presque la même performance. Notons que dans cette dernière approche, l'algorithme s'arrête quand un compromis est atteint entre les APP évaluées par le démodulateur et les APP calculées par le décodeur ou bien quand un certain nombre maximal d'itérations est atteint (ici 30 itérations). Ceci n'est pas surprenant en ce qui concerne le taux d'erreur binaire (BER) puisque les deux méthodes convergent vers les même points. Nous pouvons aussi noter que ces résultats sont obtenus avec le même nombre d'itérations : cela veut dire, dans ce cas, que la technique de point proximal ne réduit pas la vitesse de convergence. Les deux méthodes ont la même complexité de calcul avec un bonus pour l'approche proximal qui minimise un critère désiré au fil des itérations, cela est assuré par le choix des pas $\mu_c^{(n)}$ et $\mu_m^{(n)}$.

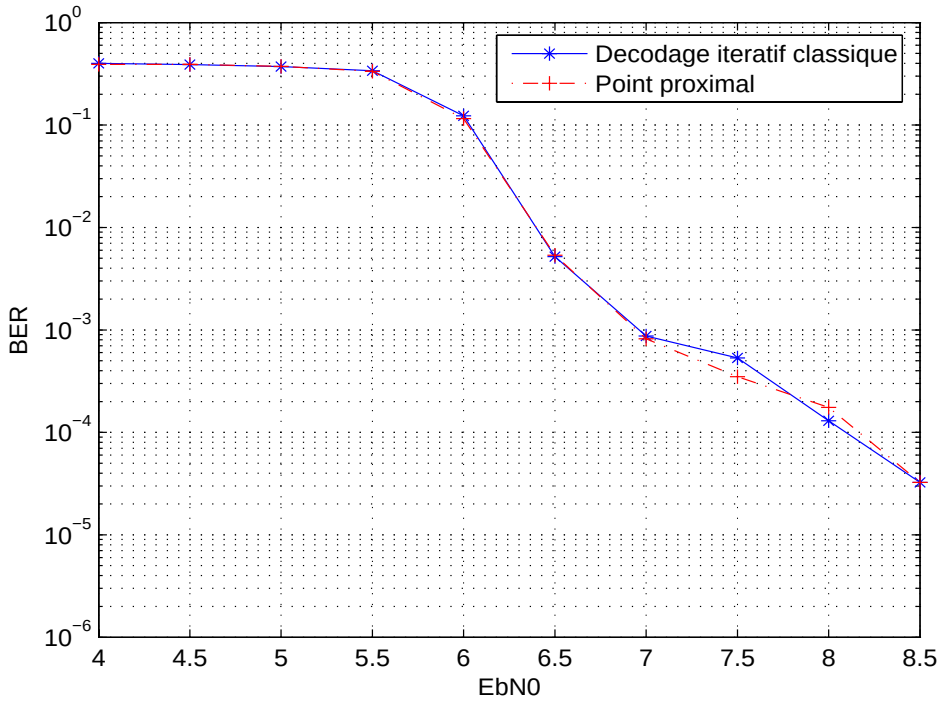

FIG. 5.5 – BER en fonction de EbN0 : comparaison des deux approches classique et point proximal

D'après le choix de $\mu_m^{(n)}$, on a :

$$C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \leq C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) \tag{5.82}$$

D'après le choix de $\mu_c^{(n)}$, on a :

$$C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \tag{5.83}$$

Et comme inégalités pouvant nous servir, on pourra écrire :

$$C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \leq C_{dem}(l^{(n)}, q^{(n)}, l^{(n)}) \tag{5.84}$$

$$C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dec}(q^{(n)}, q^{(n)}, l^{(n+1)}) \tag{5.85}$$

En effet, notre but étant de voir si, grâce à cette approche proximale, nous pourrons garantir la convergence du processus itératif. Cela serait un bonus par rapport au cas classique où la convergence n'est pas guarantie.

Pour que la convergence soit assurée, il faut garantir avoir diminué un critère bien défini au fil des itérations. Essayons de montrer, si jamais, avec ce choix, nous pourrons assurer cela.
D'après les inégalités précédentes, nous avons :

$$C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) \tag{5.86}$$

Or, nous pouvons écrire que :

$$\begin{gathered} C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) = \\ \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) + \\ \mu_c^{(n)} \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k)) \end{gathered} \tag{5.87}$$

Donc

$$C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) = \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) \tag{5.88}$$

D'autre part, on a :

$$\begin{gathered} C_{dec}(q^{(n)}, q^{(n-1)}, l^{(n)}) = \\ \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) + \\ \mu_c^{(n-1)} \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \alpha_{n-1,k} l^{(n)}(d_k) q^{(n-1)}(d_k)) \end{gathered} \tag{5.89}$$

D'où

$$C_{dec}(q^{(n)}, q^{(n)}, l^{(n)}) \leq C_{dec}(q^{(n)}, q^{(n-1)}, l^{(n)}) \tag{5.90}$$

car

$$\mu_c^{(n-1)} \sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \alpha_{n-1,k} l^{(n)}(d_k) q^{(n-1)}(d_k)) \geq 0 \tag{5.91}$$

Il en suit

$$C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dec}(q^{(n)}, q^{(n-1)}, l^{(n)}) \tag{5.92}$$

Cependant, le critère que nous désirons décroitre au fil des itérations est le critère global correspondant à la somme de ces deux critères de démodulateur et de décodeur pris séparément :

$$C_{global}(l, q, l^{(n)}, q^{(n)}) = C_{dem}(l, q^{(n)}, l^{(n)}) + C_{dec}(q, q^{(n)}, l) \tag{5.93}$$

Le but sera donc de comparer deux évaluations successives de ce critère, i.e., $C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) + C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)})$ et $C_{dem}(l^{(n+2)}, q^{(n+1)}, l^{(n+1)}) + C_{dec}(q^{(n+2)}, q^{(n+1)}, l^{(n+2)})$.
Or, d'après le choix de μ_m et μ_c, nous pouvons écrire les inégalités suivantes :

$$\begin{aligned} &C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \leq C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) \\ &C_{dem}(l^{(n+2)}, q^{(n+1)}, l^{(n+1)}) \leq C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \\ &C_{dec}(q^{(n+2)}, q^{(n+1)}, l^{(n+2)}) \leq C_{dem}(l^{(n+2)}, q^{(n+1)}, l^{(n+1)}) \end{aligned} \tag{5.94}$$

Il en suit l'inégalité désirée suivante :

$$\begin{gathered} C_{dem}(l^{(n+2)}, q^{(n+1)}, l^{(n+1)}) + C_{dec}(q^{(n+2)}, q^{(n+1)}, l^{(n+2)}) \leq \\ C_{dem}(l^{(n+1)}, q^{(n)}, l^{(n)}) + C_{dec}(q^{(n+1)}, q^{(n)}, l^{(n+1)}) \end{gathered} \tag{5.95}$$

Mais pour le moment nous pouvons pas dire que la diminution de ce critère global

garantira la convergence. En effet, il faut que la somme des deux critères évalués au même point diminue au fil des itérations ce qui n'est pas le cas ici. La raison pour laquelle nous sommes revenus vers les simulations qui montrent toujours l'existence des pas proches de zéro qui vérifient les conditions (5.82) et (5.83).

Cela nous a poussés à étudier le comportement des pas au voisinage de zéro. Travaillons maintenant sur μ_m (le cas de μ_c se traite d'une manière similaire). D'après l'inégalité (5.75), nous pouvons écrire :

$$\mu_m^{(n)} \leq Q(\mu_m^{(n)}) \tag{5.96}$$

avec

$$Q(\mu_m^{(n)}) = \frac{\sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)})) - F}{E} \tag{5.97}$$

Evaluons maintenant le développement limité de $Q(\mu_m^{(n)})$ en $\mu_m^{(n)}$ au voisinage de 0.
En effet, nous avons remarqué que

$$Q(0) = \frac{\sum_k D(\alpha_{n,k} l^{(n)}(d_k) q^{(n)}(d_k) || \gamma_k l^{(n)}(d_k) f_{d_k}(l^{(n)}))}{E} \tag{5.98}$$

suite au fait que dans le cas classique où $\mu_m^{(n)} = 0$, $l^{(n+1)}(d_k) = g_{d_k}(q^{(n)})$. Notons que $Q(0) > 0$, il en suit que n'importe quelle valeur de $\mu_m^{(n)}$ vérifiant $\mu_m^{(n)} \leq Q(0)$ remplit les conditions. En particulier, $\mu_m^{(n)} = 0$ fait l'affaire.

Cette étude montre que nous ne pouvons pas faire mieux que le cas classique à part garantir une diminution d'un critère bien défini au fils des itérations. Cependant, cela ne garantira pas la convergence, et une étude de convergence reste à analyser dans le but de pouvoir prouver que le nouveau processus proximal converge toujours. Nous pouvons ainsi se servir des interprétations géométriques apportées ainsi que des propriétés de projection afin de prouver la convergence du processus proximal.

5.7 conclusion

Dans ce chapitre, nous avons d'abord présenté le décodage itératif classique des modulations codées à bits entrelacés. Nous avons ensuite montré plusieurs interprétations de cet algorithme en se basant sur la méthode du point proximal : plusieurs reformulations et interprétations (basées sur la géométrie de l'information) du problème de décodage itératif sont proposées dans le but de trouver une interprétation de type point proximal et profiter de quelques propriétés importantes de cette méthode quant à la nature de convergence. Enfin, une interprétation point proximal par blocs séparés est adoptée (interprétation point proximal de chacun des blocs de démodulateur et de décodeur pris séparément) assurant la diminution d'un critère bien précis au fil des itérations. Les résultats de simulation montrent bien que l'algorithme de décodage itératif classique et celui basé sur cette interprétation point proximal ont presque la même performance. Ceci n'est pas surprenant en ce qui concerne le taux d'erreur binaire (BER) puisque les deux méthodes convergent vers les même points. Nous pouvons aussi noter que ces résultats sont obtenus avec le même nombre d'itérations : cela veut dire, dans ce cas, que la technique du point proximal ne réduit pas la vitesse de convergence. Les deux méthodes ont la même complexité de calcul. Cependant une étude de convergence reste à effectuer permettant de mettre en évidence le bonus apporté par la nouvelle reformulation point proximal de l'algorithme de décodage classique. Nous avons commencé quelques essais en montrant que le nouveau processus diminue toujours un critère bien précis au fil des itérations, cependant il reste à établir le lien entre la diminution de ce critère et la convergence du processus itératif.

5.8 Annexe

5.8.1 Annexe 5.1 : Notation compacte pour l'algorithme de BCJR

Dans le cas général, l'algorithme de somme-produit opère sur les probabilités *a priori* (ici, les probabilités extrinsèques $p(c_l; I)$) et calcule récursivement les quantités suivantes :

$$\alpha_k(m) = \sum_{c_1^k \in C_1^k(m)} \prod_{j=1}^{k} p(c_j; I) \tag{5.99}$$

et

$$\beta_k(m) = \sum_{c_{k+1}^N \in C_{k+1}^N(m)} \prod_{j=k+1}^{N} p(c_j; I) \tag{5.100}$$

où $C_1^k(m)$ désigne l'ensemble de $2k$ bits pris en considération dans la structure du treillis et qui terminent dans l'état m. D'une manière similaire, $C_k^N(m)$ désigne l'ensemble de $2(N-k)$ bits pris en considération dans la structure du treillis et qui commencent à l'état m.

Si on note $I_{C_1^k(m)}(.)$ la fonction indicatrice correspondante à ces mots de code, $\alpha_k(m)$ peut être réécrit comme suit :

$$\alpha_k(m) = \sum_{c_1^k \in R_1^k(m)} (\prod_{j=1}^{k} p(c_j; I)) I_{C_1^k(m)}(c_1^k) \tag{5.101}$$

où $R_1^k(m)$ l'ensemble général de $2k$ bits qui terminent dans l'état m et $I_{C_1^k(m)}(c_1^k)$ est la fonction indicatrice définie par :

$$I_{C_1^k(m)}(c_1^k) = \begin{cases} 1 \quad if \quad c_1^k \in C_1^k(m) \\ \\ 0 \quad if \quad c_1^k \in \{R_1^k(m) - C_1^k(m)\} \end{cases} \tag{5.102}$$

Considérons maintenant la somme sur tous les états du treillis, nous pouvons écrire :

$$\sum_{m} \alpha_k(m) = \sum_{c_1^k \in R_1^k(m)} (\prod_{j=1}^{k} p(c_j; I)) I_{C_1^k}(c_1^k) \tag{5.103}$$

où $I_{C_1^k}(c_1^k) = \sum_m I_{C_1^k(m)}(c_1^k)$

Les coefficients *beta* peuvent être interprétés d'une manière similaire en considérant les ensembles $C_k^N(m)$. L'algorithme somme-produit évalue la probabilité d'une transition d'un état m à un autre état $m^{'}$ effectuée à l'étape k du treillis :

$$\sigma_k(m, m^{'}) = \alpha_{k-1}(m) p(c_k = c_{m \to m'}; I) \beta_k(m^{'}) I_C(m, m^{'}) \tag{5.104}$$

qui pourra être réécrit après plusieurs simplifications comme suit :

$$\sigma_k(m, m^{'}) = \sum_{c \in R} I_{C:m \to m'}(c) (\prod_{j=1}^{N} p(c_j; I)) \tag{5.105}$$

En se basant sur les équations précédentes, nous pouvons maintenant évaluer les probabilités a posteriori $p_{APP}(c_k^l = b)$ en sommant $\sigma_k(m, m^{'})$ sur toutes les valeurs (m,m') correspondantes à la valeur b du bit correspondant :

$$p_{APP}(c_k^l = b) \sim \sum_{(m,m'):c_k^l} \sigma_k(m, m^{'}) \sim \sum_{c \in \phi_b^l} I_{\mathcal{C}}(c) \prod_j p(c_j; I) \tag{5.106}$$

où ϕ_b^l désigne l'ensemble des mots binaires de taille L_c à valeur b dans la position l.

L'information extrinsèque à la sortie du bloc du décodeur SISO est donnée par [MDdC02][MG98] :

$$p(c_l = b; O) = K_c \sum_{c \in \phi_b^l} I_{\mathcal{C}}(c) \prod_{j \neq l} p(c_j; I) \tag{5.107}$$

et l'APP correspondant par :

$$p_{APP}(c_l = b) = K_c^{'} \sum_{c \in \phi_b^l} I_{\mathcal{C}}(c) \prod_j p(c_j; I) \tag{5.108}$$

où $I_{\mathcal{C}}(\underline{c})$ représente la fonction indicatrice du code et prend la forme suivante [MDdC02] :

$$I_{\mathcal{C}}(\underline{c}) = \begin{cases} 1 & if \ \ \underline{c} \in \mathcal{C} \\ \\ 0 & sinon \end{cases} \tag{5.109}$$

K_c et $K_c^{'}$ sont des facteurs de normalisation.

5.8.2 Annexe 5.2 : Conditions de convergence du décodage itératif des BICM

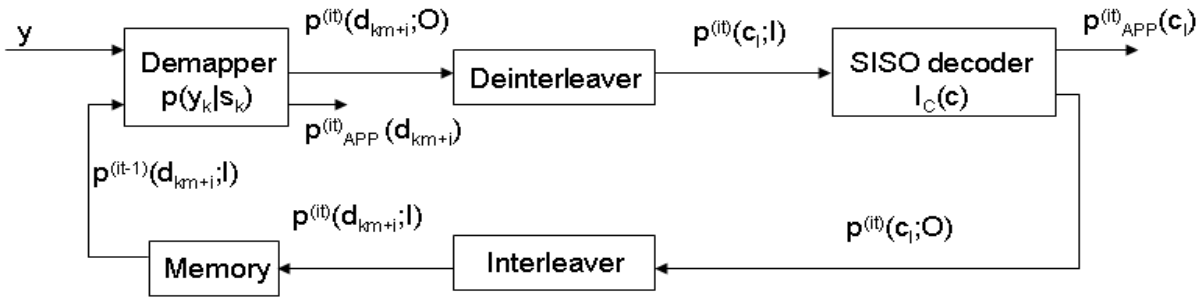

FIG. 5.6 – Iterative receiver for a BICM-ID with soft-decision feedback

Concernant le bloc de démodulateur, nous avons :

$$p^{(n+1)}(d_{km+i}=b;O)=\frac{p_{APP}^{(n+1)}(d_{km+i}=b)}{p^{(n)}(d_{km+i}=b;I)} \tag{5.110}$$

Concernant le bloc de décodeur, nous pouvons écrire :

$$p^{(n+1)}(c_l=b;O)=\frac{p_{APP}^{(n+1)}(c_l=b)}{p^{(n+1)}(c_l=b;I)} \tag{5.111}$$

A la convergence, nous avons d'une part :

$$\begin{gathered} p^{(n+1)}(c_l=b;O)=p^{(n)}(c_l=b;O)=p(c_l=b;O) \\ \updownarrow \\ p(d_{km+i}=b;I)=p^{(n)}(d_{km+i}=b;I)=p^{(n+1)}(d_{km+i}=b;I) \end{gathered} \tag{5.112}$$

et d'autre part :

$$\begin{gathered} p^{(n)}(c_l=b;I)=p^{(n+1)}(c_l=b;I)=p(c_l=b;I) \\ \updownarrow \\ p(d_{km+i}=b;O)=p^{(n)}(d_{km+i}=b;O)=p^{(n+1)}(d_{km+i}=b;O) \end{gathered} \tag{5.113}$$

Il en suit que les APP à la sortie des 2 blocs (le démodulateur et le décodeur) sont les même à la convergence.

5.8.3 Annexe 5.3 : Résolution du problème d'optimisation

Le Lagrangien du problème d'optimisation (5.24) est :

$$\mathcal{L} = \sum_k \sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log(\alpha_k l(d_k) f_{d_k}(l)) - \sum_k \sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log(\beta_k q(d_k) g_{d_k}(q)) \\ -\sum_k \lambda_k (\sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) - 1) - \sum_k \mu_k (\sum_{d_k} \beta_k q(d_k) g_{d_k}(q) - 1) \tag{5.114}$$

où λ_k et μ_k sont des multiplieurs de Lagrange. Nous voulons tout d'abord évaluer le gradient de ce Lagrangien par rapport à $l(d_i)$. C'est la raison pour laquelle nous considérons cette expression équivalente du Lagrangien :

$$\mathcal{L} = A_i + \sum_{k \neq i} A_k - B_i - \sum_{k \neq i} B_k - C - D \tag{5.115}$$

où

$$A_k = \sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log(\alpha_k l(d_k) f_{d_k}(l)) \tag{5.116}$$

$$B_k = \sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log(\beta_k q(d_k) g_{d_k}(q)) \tag{5.117}$$

$$C = \sum_k \lambda_k (\sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) - 1) \tag{5.118}$$

$$D = \sum_k \mu_k (\sum_{d_k} \beta_k q(d_k) g_{d_k}(q) - 1) \tag{5.119}$$

Le gradient de chaque terme par rapport à $l(d_i)$ est donné par :

$$\frac{\partial A_i}{\partial l(d_i)} = \alpha_i f_{d_i}(l) \log(\alpha_i l(d_i) f_{d_i}(l)) + \alpha_i f_{d_i}(l) \tag{5.120}$$

$$\frac{\partial A_k}{\partial l(d_i)} = \\ \sum_{d_k} (\alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)) \log \alpha_k l(d_k) f_{d_k}(l) + \sum_{d_k} \alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j) \tag{5.121}$$

$$\frac{\partial B_i}{\partial l(d_i)} = \alpha_i f_{d_i}(l) \log(\beta_i q(d_i) g_{d_i}(q)) \tag{5.122}$$

$$\frac{\partial B_k}{\partial l(d_i)} = \sum_{d_k} (\alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)) \log(\beta_k q(d_k) g_{d_k}(q)) \tag{5.123}$$

$$\frac{\partial C}{\partial l(d_i)} = \lambda_i \alpha_i f_{d_i}(l) + \sum_{k \neq i} \lambda_k \alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j) = f_{d_i}(l) \sum_k \alpha_k \lambda_k \tag{5.124}$$

$$\frac{\partial D}{\partial l(d_i)} = 0 \tag{5.125}$$

La constante de normalisation α_k a pour expression :

$$\alpha_k = \frac{1}{\sum_{d_k} l(d_k) f_{d_k}(l)} \tag{5.126}$$

et dépend donc de $l(d_k)$. Cependant la dérivée de α_k par rapport à $l(d_k)$ est :

$$\frac{\partial \alpha_k}{\partial l(d_k)} = -f_{d_k}(l) \alpha_k^2 \tag{5.127}$$

qui, après simplifications de l'expression du gradient, rentre dans la constante de normalisation finale. C'est pour cela que, dans nos développements, nous ne considérons pas la dérivée de α_k par rapport à $l(d_k)$.

Nous en déduisons le gradient de Lagrangien par rapport à $l(d_i)$:

$$\begin{aligned} &\frac{\partial \mathcal{L}}{\partial l(d_i)} = \alpha_i f_{d_i}(l) \log \frac{\alpha_i l(d_i) f_{d_i}(l)}{\beta_i q(d_i) g_{d_i}(q)} + \alpha_i f_{d_i}(l) + f_{d_i}(l) \textstyle\sum_{k \neq i} \alpha_k - f_{d_i}(l) \sum_k \alpha_h \lambda_k \\ &+ \textstyle\sum_{k \neq i} \sum_{d_k} (\alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)) \log \frac{\alpha_k l(d_k) f_{d_k}(l)}{\beta_k q(d_k) g_{d_k}(q)} \end{aligned}$$

Nous voulons trouver la valeur de $l(d_i)$ qui annule ce gradient. Nous pouvons donc vérifier que $l(d_i)$ vérifie l'équation suivante :

$$\begin{aligned} &\alpha_i l(d_i) f_{d_i}(l) = \\ &K_i q(d_i) g_{d_i}(q) \exp\left(\tfrac{-1}{\alpha_i f_{d_i}(l)} \textstyle\sum_{k \neq i} \sum_{d_k} (\alpha_k l(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j \neq k,i} l(d_j)) \log \frac{\alpha_k l(d_k) f_{d_k}(l)}{\beta_k q(d_k) g_{d_k}(q)}\right) \end{aligned} \tag{5.128}$$

Une expression analytique de $l(d_i)$ n'est pas facile à trouver puisque $f_{d_k}(l)$ dépend aussi

de $l(d_i)$. De plus, résoudre ce problème d'une manière itérative nécessite l'évaluation de $n-1$ expressions du type $\sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j\neq k,i} l(d_j)$. Cette évaluation nécessiterait une modification de l'algorithme de BCJR et se traduirait par une augmentation notable de la complexité : le BCJR devant fournir non plus n sorties (les marginales sur chacun des bits) mais $\frac{n(n-1)}{2}$ sorties (les marginales sur tous les couples possibles de 2 bits).

L'algorithme de One Step Late proposé par Green [Gre90] est une technique où l'on remplace la variable par sa valeur calculée à l'itération précédente. Cette substitution sera effectuée dans l'expression du gradient, où on remplace uniquement les termes qui nous empêchent de trouver une expression analytique de notre variable optimale.

Dans ce qui suit, nous supposons que nous sommes à l'itération (n+1) et que nous voulons calculer la variable $l(d_i) = l^{(n+1)}(d_i)$

Essayons maintenant d'utiliser cette technique de One Step late de telle manière à ce que la solution classique du décodage itératif qu'on connait :

$$q^{(n)}(d_i) = f_{d_i}(l^{(n)}) \tag{5.129}$$

$$l^{(n+1)}(d_i) = g_{d_i}(q^{(n)}) \tag{5.130}$$

soit aussi une solution de notre problème d'optimisation.

Pour cela nous introduisons le One Step Late dans l'équation (5.128) de la manière suivante :

$$\alpha_i l^{(n+1)}(d_i) f_{d_i}(l^{(n)}) = K_i q^{(n)}(d_i) g_{d_i}(q^{(n)})$$
$$\exp\left(\frac{-1}{\alpha_i f_{d_i}(l^{(n)})} \sum_{k\neq i}\sum_{d_k} (\alpha_k l^{(n+1)}(d_k) \sum_{\underline{d}:d_k,d_i} I_{\mathcal{C}}(\underline{d}) \prod_{j\neq k,i} l^{(n)}(d_j)) \log \frac{\alpha_k l^{(n+1)}(d_k) f_{d_k}(l^{(n)})}{\beta_k q^{(n)}(d_k) g_{d_k}(q^{(n)})}\right) \tag{5.131}$$

Et comme ça nous aurons bien que $l^{(n+1)}(d_i) = g_{d_i}(q^{(n)})$ et $q^{(n)}(d_i) = f_{d_i}(l^{(n)})$ sont une solution de notre problème.

Maintenant l'idée étant de remonter, une fois nous avons ajouté les itérations dans le gradient, afin de définir le critère qui va correspondre à ce gradient modifié.

Cela va nous permettre de définir le critère que nous voulons minimiser de la manière suivante :

$$\begin{cases} \min\limits_{l(d_k)_{1\leq k\leq n}\in\mathbb{R}^+, q(d_k)_{1\leq k\leq n}\in\mathbb{R}^+} \sum\limits_{k=1}^{n}\sum\limits_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log \dfrac{\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)})}{\beta_k q(d_k) g_{d_k}(q)} \\ \sum\limits_{d_k} \alpha_k l(d_k) f_{d_k}(l) = 1 \quad 1 \leq k \leq n \\ \sum\limits_{d_k} \beta_k q(d_k) g_{d_k}(q) = 1 \quad 1 \leq k \leq n \end{cases} \tag{5.132}$$

Vu que notre critère n'est plus symétrique par rapport à l et q, il faut vérifier, qu'en partant du même critère que celui pour le calcul de l, la solution classique vérifira toujours les équations sur le bloc de décodeur (calcul de q).
En effet, le critère qu'on cherche à minimiser, pour les deux blocs est le suivant :

$$\sum_{k=1}^{n}\sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log \frac{\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)})}{\beta_k q(d_k) g_{d_k}(q)} \tag{5.133}$$

En faisant le même genre de calcul que nous avons détaillé pour le calcul de $l(d_i)$, nous obtenons ainsi :

$$\frac{\alpha_i l(d_i) f_{d_i}(l)}{q(d_i)} = -g_{d_i}(q)\beta_i\mu_i - g_{d_i}(q)\sum_{k\neq i}\beta_k\mu_k - \sum_{k\neq i}\sum_{d_k}\alpha_k l(d_k) f_{d_k}(l) \frac{\sum\limits_{\underline{d}:d_k,d_i} p(y|d)\prod\limits_{j\neq k,i} q(d_j)}{g_{d_k}(q)} \tag{5.134}$$

Introduisons maintenant la technique de One Step late, comme dans le cas de calcul de l, pour vérifier si la solution classique $l^{(n+1)}(d_i) = g_{d_i}(q^{(n)})$ et $q^{(n+1)}(d_i) = f_{d_i}(l^{(n+1)})$ est une solution ou pas.

En introduisant les itérations d'une manière convenable, nous retrouvons :

$$\alpha_i l^{(n+1)}(d_i) f_{d_i}(l^{(n+1)}) = q^{(n+1)}(d_i) g_{d_i}(q^{(n)})(-\beta_i\mu_i - \sum_{k\neq i}\beta_k\mu_k - \sum_{k\neq i}\alpha_k) \tag{5.135}$$

Or μ_k étant le multiplieur de Lagrange calculé d'une manière à ce que $\sum_{d_k}\beta_k q^{(n+1)}(d_k) g_{d_k}(q^{(n)}) = 1$, nous pouvons écrire alors $\beta_i = \alpha_i - \sum_k \alpha_k - \sum_k \beta_k\mu_k$
Or dans le cas classique, nous avons bien que $\alpha_k = \beta_k$, d'où $\mu_k = -1$

on voit donc bien, avec $\mu_k = -1$, que la solution classique du décodage itératif des BICM est aussi une solution de notre nouveau critère.

Maintenant, une fois qu'on a vérifié que la solution classique est une solution de notre critère, le but étant de proposer une interprétation de ce critère basé sur l'algorithme du point proximal afin d'améliorer ce processus de décodage itératif et de pouvoir éventuellement prouver la convergence.

Nous avons déjà vu que le critère qu'on veut minimiser est le suivant :

$$\sum_{k=1}^{n}\sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log \frac{\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)})}{\beta_k q(d_k) g_{d_k}(q)} \tag{5.136}$$

Ce critère pourra être écrit de la manière suivante :

$$\sum_k D(\alpha_k l(d_k) f_{d_k}(l) || \beta_k q(d_k) g_{d_k}(q)) - \sum_k D(\alpha_k l(d_k) f_{d_k}(l) || \alpha_k^{'} l(d_k) f_{d_k}(l^{(n)})) \tag{5.137}$$

Le premier terme de cette somme n'est autre que le critère de (5.24), alors que le deuxième terme est un terme de compensation pour tomber sur le critère qu'on veut pratiquement minimiser.

Essayons maintenant de voir quel terme il faut ajouter de telle manière à ce que le terme proximal soit non négatif et qui tend vers 0 à la convergence.

Notons que le but du terme proximal étant d'assurer que la nouvelle solution reste toujours au voisinage de la solution de l'itération précédente afin de garantir la convergence de l'algorithme.

Une première option consiste à rajouter un terme de type :

$$\mu \sum_k D(\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)}) || \alpha_k^{(n)} l^{(n)}(d_k) f_{d_k}(l^{(n)})) \tag{5.138}$$

où μ est le pas que nous choisissons de manière à avoir une bonne approche proximale.

Le terme proximal correspondant sera :

$$\begin{array}{c}\mu \sum_k D(\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)}) || \alpha_k^{(n)} l^{(n)}(d_k) f_{d_k}(l^{(n)})) - \\ \sum_k D(\alpha_k l(d_k) f_{d_k}(l) || \alpha_k^{'} l(d_k) f_{d_k}(l^{(n)}))\end{array} \tag{5.139}$$

Et le critère à minimiser est :

$$\begin{array}{c}\sum_{k=1}^{n} \sum_{d_k} \alpha_k l(d_k) f_{d_k}(l) \log \frac{\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)})}{\beta_k q(d_k) g_{d_k}(q)} + \\ \mu \sum_k D(\alpha_k^{'} l(d_k) f_{d_k}(l^{(n)}) || \alpha_k^{(n)} l^{n)}(d_k) f_{d_k}(l^{(n)}))\end{array} \tag{5.140}$$

La difficulté ici réside dans la recherche des équations de mise à jour des variables $l(d_i)$ et $q(d_i)$: en effet, une expression analytique est très difficile à trouver (il en est de même pour le choix de μ) vu que dans la même équation on a $l(d_i)$ et $f_{(d_k)}(l)$ qui dépend de $l(d_i)$. Donc cela augmente la complexité de l'algorithme de BCJR comme on l'a expliqué avant.

6

Lien entre Maximum de Vraisemblance et décodage itératif

6.1 Introduction

A l'origine, le décodage itératif des BICM n'a pas été introduit comme solution d'un problème d'optimisation. Ceci rend sa structure précise adhoc et l'analyse de sa convergence et stabilité très difficile. Une approche géométrique a été considérée, elle fournit une interprétation intéressante en termes de projections pouvant ainsi servir à analyser ces algorithmes itératifs. Le cas particulier des BICM-ID a été étudié dans [MDdC02] menant à une bonne caractérisation du démodulateur et du décodeur. Cependant, une caractérisation du processus entier reste à exploiter. D'autres travaux [Ric00] interprètent, du point de vue géométrie de l'information, le turbo décodage itératif comme

étant un système dynamique. Cela mène à de nouveaux résultats quant au lien existant entre ces algorithmes du décodage itératif et le critère du maximum de vraisemblance mais qui restent néanmoins toujours incomplets. Ces résultats incomplets sont essentiellement dus à la manque d'une description efficace d'échange d'informations extrinsèques. La relation entre le décodage optimal par maximum de vraisemblance et le décodage itératif n'est pas encore complètement illustrée bien qu'elle soit introduite dans les travaux de Richardson [Ric00] : l'existence de deux codes dans le schéma de turbo décodage rend la marginalisation de la fonction de vraisemblance assez difficile voire non faisable en pratique. Ceci amène à considérer séparément chacun de ces deux codes et à réaliser le décodage d'une manière séparée via l'algorithme de BCJR [BCJR74]. Les deux blocs échangent ensuite des informations entre eux pour permettre à chacun des blocs de prendre en compte l'information provenant de l'autre, et cela d'une manière itérative, jusqu'à la convergence du processus. Cet échange d'extrinsèque restait toujours intuitif, ce qui a poussé Richardson à considérer une approche géométrique afin de trouver un lien intuitif pouvant exister entre le turbo décodage itératif et le maximum de vraisemblance. Il présente ainsi une interprétation géométrique de l'algorithme de turbo décodage en termes de projections sur des espaces de probabilité. Une perspective géométrique indique clairement la relation entre cet algorithme et le décodage par maximum de vraisemblance. Cependant, ses principaux résultats concernent essentiellement la recherche des conditions d'existence et d'unicité des points fixes ainsi que les conditions de stabilité de l'algorithme de turbo décodage itératif après une analyse profonde de la géométrie correspondante. Plusieurs travaux viennent dans la suite modéliser analytiquement les principaux résultats géométriques obtenus par Richardson notamment le lien existant entre le turbo décodage itératif et le décodage par maximum de vraisemblance. Parmi ces travaux, nous pouvons citer ceux de Walsh [WRJ05, WRJ06, RW07] qui reformulent le processus de décodage itératif comme étant la solution d'un problème d'optimisation avec contraintes cherchant à maximiser la vraisemblance et étudient la convergence de l'algorithme de turbo décodage itératif. Cependant, les contraintes figurant dans ce problème d'optimisation varient avec les itérations, raison pour laquelle les outils classiques ne sont plus efficaces pour analyser le comportement d'une telle procédure itérative à la convergence. Ce qui nous a conduit à illustrer analytiquement la relation existant entre le turbo décodage itératif et le décodage par maximum de vraisemblance afin d'éliminer

certaines contradictions dans les résultats de Walsh et compléter les résultats de Richardson en donnant une description efficace d'échange d'informations extrinsèques qui manquait dans les travaux précédents.

Dans ce chapitre, nous allons d'abord présenter le principe du décodage par maximum de vraisemblance. Un critère approché (mais calculable) est déduit à partir d'une formulation équivalente et convenable du critère optimal. Nous prouvons que, dans certains cas spécifiques, le maximum global du critère approché atteint celui du critère optimal de maximum de vraisemblance. Nous considérons ensuite la maximisation itérative et nous prouvons qu'un choix particulier du processus de mise à jour mène aux équations classiques utilisées dans le décodage (turbo) itératif. Enfin, dans la partie simulation, ces résultats sont appliqués à la détection des solutions suspectes i.e., solutions avec un nombre important d'erreurs.

Ces résultats ont été présentés à la conférence ICASSP 2011. L'article correspondant est mis en annexe du manuscrit (Annexe B).

6.2 Critère maximum de vraisemblance et critère approché

Soit $\mathbf{p}(\mathbf{y} \mid \mathbf{d})$ la densité de probabilité correspondant à l'effet du canal de transmission. Soit $\mathbf{I}_c(\mathbf{d})$ la fonction indicatrice du code. Le maximum de vraisemblance est donné par :

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}} \mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{I}_c(\mathbf{d}) \tag{6.1}$$

Nous considérons ici la fonction de vraisemblance évaluée sur la séquence des bits codés et entrelacés. Une correspondance une à une existe cependant entre la séquence entrelacée d et le message binaire b permettant le passage facile entre ces quantités.

Ce problème peut être reformulé selon :

$$\hat{\mathbf{p}}(\mathbf{d}) = \arg\max_{\mathbf{p}\in\mathcal{E}} \sum_{\mathbf{d}} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{p}(\mathbf{d}) \tag{6.2}$$

où $\mathcal{E}$ est l'ensemble des lois de probabilité sur $\mathbf{d}$ et $\mathbf{d} \in \{0, 1, ..., 2^n - 1\}$. La densité $\hat{\mathbf{p}}(\mathbf{d})$ est donnée par $\hat{\mathbf{p}}(\mathbf{d}) = 1$ si $\mathbf{d} = \hat{\mathbf{d}}$ et 0 sinon. Elle joue le rôle de pointeur sur la séquence $\hat{\mathbf{d}}$. En introduisant les marginales sur le bit k, on peut réécrire (6.2) sous la forme équivalente :

$$\hat{\mathbf{p}}(\mathbf{d}) = \arg\max_{\mathbf{p}\in\mathcal{E}} \sum_{\mathbf{d_k}} \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{p}(\mathbf{d}) \tag{6.3}$$

En pratique n est grand et l'ensemble des 2^n valeurs de $\mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{I}_c(\mathbf{d})$ n'est pas calculable. Un critère approché s'obtient en remplaçant les marginales du produit par le produit des marginales. Si les deux lois considérées sont séparables alors il ne s'agit plus d'une approximation puisque le produit des marginales est bien égal aux marginales du produit des deux lois. Cela est vrai en particulier pour les lois de probabilité de type "Dirac" qui sont un sous-ensemble des lois séparables. La solution $\hat{\mathbf{p}}(\mathbf{d})$ de (6.2) est une loi de probabilité de type "Dirac". Considérons maintenant le pointeur $\mathbf{p}(\mathbf{d})$ comme le produit de deux lois de probabilité séparables : $\mathbf{p}(\mathbf{d}) = \mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$ où $\mathbf{l}(\mathbf{d})$ et $\mathbf{q}(\mathbf{d})$ appartiennent à $\mathcal{E}_S$ l'ensemble des densités de probabilité séparables. On peut remarquer que le critère $\sum_{\mathbf{d}} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{p}(\mathbf{d})$ s'écrit de manière équivalente selon $\sum_{\mathbf{d}} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\alpha\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$ où α est la constante de normalisation du produit $\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$ par conséquent $\alpha\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$ est une loi de probabilité. On peut alors chercher à maximiser par rapport à $\mathbf{l}$ et $\mathbf{q}$: $\frac{1}{\alpha}\sum_{\mathbf{d}} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\alpha\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$. Le terme $\sum_{\mathbf{d}} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\alpha\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$ est maximisé par $\hat{\mathbf{l}}(\mathbf{d}) = \hat{\mathbf{q}}(\mathbf{d}) = 1$ si $\mathbf{d} = \hat{\mathbf{d}}$ et 0 sinon. La constante de normalisation α est toujours supérieure ou égale à 1, le terme $\frac{1}{\alpha}$ est donc également maximisé par ce choix de $\hat{\mathbf{l}}$ et de $\hat{\mathbf{q}}$. On obtient ainsi une nouvelle écriture du problème d'optimisation :

$$\left(\hat{\mathbf{l}}(\mathbf{d}), \hat{\mathbf{q}}(\mathbf{d})\right) = \arg\max_{\mathbf{l},\mathbf{q}\in\mathcal{E}_s} \left(\sum_{\mathbf{d_k}} \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d})\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d}) \right) \tag{6.4}$$

On note $\mathcal{C}$ le critère défini ci-dessus. On peut maintenant définir le critère approché. On le note $\tilde{\mathcal{C}}_k$. On a

$$\tilde{\mathcal{C}}_k = \left(\sum_{\mathbf{d_k}} \left(\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_j q_j(d_j) \right) \left(\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_j l_j(d_j) \right) \right) \tag{6.5}$$

soit encore

$$\tilde{\mathcal{C}}_k = \left(\sum_{\mathbf{d_k}} q_k(d_k) l_k(d_k) g_{d_k}(\mathbf{q}) f_{d_k}(\mathbf{l}) \right) \tag{6.6}$$

avec $g_{d_k}(\mathbf{q}) = \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j)$ et $f_{d_k}(\mathbf{l}) = \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j)$ et $l_j(d_j)$ et $q_j(d_j)$ les marginales sur le bit j.

6.3 Maximisation du critère approché

6.3.1 Maximum global

Proposition 1 *Le maximum du critère $\tilde{\mathcal{C}}_k$ est obtenu pour $\mathbf{q} = \hat{\mathbf{q}}$ et $\mathbf{l} = \hat{\mathbf{l}}$ définis par $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 1$ pour $(\mathbf{d}, \mathbf{d}') = (\hat{\mathbf{d}}, \hat{\mathbf{d}}')$ et $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 0$ pour tous les autres couples $(\mathbf{d}, \mathbf{d}')$ avec $(\hat{\mathbf{d}}, \hat{\mathbf{d}}') = \arg\max_{\mathbf{d},\mathbf{d}'} \mathbf{p}(\mathbf{y} \mid \mathbf{d}')\mathbf{I}_c(\mathbf{d})$.*

Démonstration :

$$\tilde{\mathcal{C}}_k = \left(\sum_{\mathbf{d_k}} \left(\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_j q_j(d_j) \right) \left(\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_j l_j(d_j) \right) \right) \tag{6.7}$$

soit encore

$$\tilde{\mathcal{C}}_k = \sum_{\mathbf{d_k}} \left(\sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \mathbf{I}_c(\mathbf{d}) \mathbf{p}(\mathbf{y} \mid \mathbf{d}') \prod_j q_j(d_j) l_j(d'_j) \right) \tag{6.8}$$

On peut montrer que $\sum_{d_k} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \prod_j q_j(d_j) l_j(d'_j) = \sum_{d_k} l_k(d'_k) q_k(d_k)$. En effet

$$\sum_{d_k} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \prod_j q_j(d_j) l_j(d'_j) =$$
$$\sum_{d_k} q_k(d_k) l_k(d'_k) \sum_{d_1,d_2,\ldots,d_{k-1},d_{k+1},\ldots,d_n} \sum_{d'_1,d'_2,\ldots,d'_{k-1},d'_{k+1},\ldots,d'_n} \prod_{j \neq k} q_j(d_j) l_j(d'_j) =$$
$$\sum_{d_k} q_k(d_k) l_k(d'_k) \sum_{d_1,d_2,\ldots,d_{k-1},d_{k+1},\ldots,d_n} \prod_{j \neq k} q_j(d_j) \left(\sum_{d'_1,d'_2,\ldots,d'_{k-1},d'_{k+1},\ldots,d'_n} \prod_{j \neq k} l_j(d'_j) \right) = 1$$

En d'autres termes

$$\frac{1}{\sum_{d_k} l_k(d'_k) q_k(d_k)} \sum_{d_k} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \prod_j q_j(d_j) l_j(d'_j) = 1$$

Donc

$$\left(\frac{1}{\sum_{d_k} l_k(d'_k)q_k(d_k)} \sum_{\mathbf{d_k}} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \mathbf{I}_c(\mathbf{d})\mathbf{p}(\mathbf{y} \mid \mathbf{d}') \prod_j q_j(d_j)l_j(d'_j)\right)$$

est maximisé par $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 1$ pour $(\mathbf{d}, \mathbf{d}') = (\hat{\mathbf{d}}, \hat{\mathbf{d}}')$ et $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 0$ pour tout les autres couples $(\mathbf{d}, \mathbf{d}')$ intervenant dans (6.8). On peut remarquer que les couples $(\mathbf{d}, \mathbf{d}')$ concernés sont tels que $\mathbf{d}$ et $\mathbf{d}'$ ont la même valeur pour le $k^{ième}$ bit donc $\hat{\mathbf{d}}$ et $\hat{\mathbf{d}}'$ ont la même valeur pour le $k^{ième}$ bit. Par conséquent, le coefficient de normalisation $\sum_{d_k} l_k(d'_k)q_k(d_k)$ est également maximisé par ce choix de $\mathbf{l}$ et de $\mathbf{q}$. Ce qui implique que $\tilde{\mathcal{C}}_k$ est maximisé par $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 1$ pour $(\mathbf{d}, \mathbf{d}') = (\hat{\mathbf{d}}, \hat{\mathbf{d}}')$ et $\mathbf{q}(\mathbf{d})\mathbf{l}(\mathbf{d}') = 0$ pour tout les autres couples $(\mathbf{d}, \mathbf{d}')$. De cette proposition on peut conclure que si $\hat{\mathbf{d}}'$ est un mot de code (c'est à dire si la probabilité canal est maximisée pour un mot qui correspond à un mot de code) alors $(\hat{\mathbf{d}}', \hat{\mathbf{d}}') = \arg\max_{\mathbf{d},\mathbf{d}'} \mathbf{p}(\mathbf{y} \mid \mathbf{d}')\mathbf{I}_c(\mathbf{d})$ et $\tilde{\mathcal{C}}_k$ est maximisé par la même loi de probabilité que $\mathcal{C}_1$. Dans cette situation, le maximum de $\tilde{\mathcal{C}}_k$ fournit le maximum de vraisemblance. Le critère $\tilde{\mathcal{C}}_k$ est donc un critère approché pertinent.

En revanche, si la probabilité canal est maximisée par un mot qui n'est pas un mot de code et si on laisse une indépendance relative à $\mathbf{l}$ et $\mathbf{q}$ alors le maximum de $\tilde{\mathcal{C}}_k$ ne pointera pas nécessairement sur un mot de code.

Il faut en tenir compte pour mettre au point une stratégie pour maximiser $\tilde{\mathcal{C}}_k$.

Le critère $\tilde{\mathcal{C}}_k$ est le critére approché correspondant à la marginale k. Il y a donc n critères $\tilde{\mathcal{C}}_k$ pour $1 \leq k \leq n$. Chaque critère constitue une approximation du maximum de vraisemblance, en effet :

$$\tilde{\mathcal{C}}_k = \sum_{d_k} \sum_{\mathbf{d}:d_k} I_c(\mathbf{d})p(\mathbf{y} \mid \mathbf{d})\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d}) + \sum_{d_k} \sum_{\mathbf{d}:d_k, \mathbf{d}':d'_k, \mathbf{d}\neq\mathbf{d}'} I_c(\mathbf{d})p(\mathbf{y} \mid \mathbf{d}')\mathbf{l}(\mathbf{d}')\mathbf{q}(\mathbf{d}) \qquad (6.9)$$

On constate que le premier terme est exactement le critère $\mathcal{C}$ alors que le deuxième terme est lié à l'approximation et rend une forme différente selon la valeur de k.

Proposition 2 *Si le maximum global de* $\tilde{\mathcal{C}}$ *est obtenu pour* $\mathbf{lq}(\mathbf{d}) = 1$ *pour* $\mathbf{d} = \mathbf{d}_0$ *et* 0 *ailleurs alors* $\mathbf{d}_0 = \hat{\mathbf{d}}$ *avec* $\hat{\mathbf{d}} = \arg\max_{\mathbf{d}} I_c(\mathbf{d})p(\mathbf{y} \mid \mathbf{d})$.

Démonstration :

Notons $\hat{\mathbf{l}}\hat{\mathbf{q}}$ le maximum global de $\tilde{\mathcal{C}}$, on a $\tilde{\mathcal{C}}(\hat{\mathbf{l}}\hat{\mathbf{q}}) \geq \tilde{\mathcal{C}}(\mathbf{lq})$. On a aussi $\tilde{\mathcal{C}}(\mathbf{lq}) \geq n\mathcal{C}(\mathbf{lq})$. De plus $\tilde{\mathcal{C}}(\hat{\mathbf{l}}\hat{\mathbf{q}}) = n\mathcal{C}(\hat{\mathbf{l}}\hat{\mathbf{q}})$. Donc $\mathcal{C}(\hat{\mathbf{l}}\hat{\mathbf{q}}) \geq \mathcal{C}(\mathbf{lq})$ pour tout couple $\mathbf{l}$, $\mathbf{q}$ et $\hat{\mathbf{l}}\hat{\mathbf{q}}$ est aussi le

maximum global de $\mathcal{C}$.

6.3.2 Maximisation itérative

Stratégie de maximisation itérative

La maximisation directe de $\tilde{\mathcal{C}}_k$ n'est pas évidente ce qui conduit à considérer un processus itératif. Essayons de remettre à jour chaque marginale tour à tour. A partir du critère $\tilde{\mathcal{C}}_k$ on cherche donc :

$$\left(\hat{l}_k, \hat{q}_k\right) = \arg\max_{l_k, q_k \in \mathcal{E}} \left(\sum_{\mathbf{d_k}} q_k(d_k) l_k(d_k) \left(\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j) \right) \left(\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j) \right) \right) \tag{6.10}$$

La solution est donnée par :

$$\begin{aligned} \hat{l}_k(d_k)\hat{q}_k(d_k) \quad &= 1 \;\; si \quad \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j) > \\ &\qquad \sum_{\mathbf{d}:\overline{d_k}} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j) \sum_{\mathbf{d}:\overline{d_k}} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j) \\ &= 0 \;\; sinon \end{aligned}$$

Il s'agit donc de décisions dures qui peuvent conduire assez fréquemment à des maxima locaux. La décision souple correspondant à ce processus itératif est donnée par :

$$\begin{aligned} l_k(d_k)\hat{q}_k(d_k) \quad &\propto \quad \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j)) \\ l_k(\overline{d_k})\hat{q}_k(\overline{d_k}) \quad &\propto \quad \sum_{\mathbf{d}:\overline{d_k}} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j(d_j) \sum_{\mathbf{d}:\overline{d_k}} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j(d_j)) \end{aligned}$$

Le produit $\mathbf{lq}$ est clairement défini. Cependant pour pouvoir itérer le processus nous devons faire un choix pour $\mathbf{l}$ et pour $\mathbf{q}$ compatible avec la valeur du produit. On peut regarder ce que donnent ces équation si l'on initialise $\mathbf{l}$ et $\mathbf{q}$ avec $\mathbf{l}^{(0)} = \mathbf{q}^{(0)} = (1/2)^n$. Commençons par estimer $\mathbf{l}^{(1)}$ on a

$$l_k^{(1)}(d_k)q_k^{(0)}(d_k) \quad \propto \quad \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(0)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(0)}(d_j))$$

$$\propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \tag{6.11}$$

On en déduit

$$l_k^{(1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \tag{6.12}$$

On peut maintenant calculer $\mathbf{q}^{(1)}$:

$$l_k^{(1)}(d_k) q_k^{(1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(0)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(1)}(d_j)) \tag{6.13}$$

soit encore

$$q_k^{(1)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(1)}(d_j))}{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d})} \tag{6.14}$$

On passe ensuite au calcul de $\mathbf{l}^{(2)}$

$$l_k^{(2)}(d_k) q_k^{(1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(1)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(1)}(d_j)) \tag{6.15}$$

ce qui conduit à

$$l_k^{(2)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(1)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \tag{6.16}$$

On a ensuite

$$q_k^{(2)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(2)}(d_j)}{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d})} \tag{6.17}$$

De manière générale cela conduit à

$$l_k^{(n+1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(n)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \tag{6.18}$$

$$q_k^{(n+1)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(n+1)}(d_j)}{\sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d})} \tag{6.19}$$

Dans (6.11), si nous avions cherché à estimer en premier $\mathbf{q}^{(1)}$ au lieu de $\mathbf{l}^{(1)}$, nous aurions obtenu comme équations de remise à jour :

$$q_k^{(n+1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(n)}(d_j) \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \tag{6.20}$$

$$l_k^{(n+1)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(n+1)}(d_j)}{\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d})} \tag{6.21}$$

(6.22)

Sachant que la plupart des codeurs convolutifs vérifient $\sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) = \sum_{\mathbf{d}:\overline{d_k}} \mathbf{I}_c(\mathbf{d})$, on obtient

$$\hat{q}_k^{(n+1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{p}(\mathbf{y} \mid \mathbf{d}) \prod_{j \neq k} l_j^{(n)}(d_j) \tag{6.23}$$

$$\hat{l}_k^{(n+1)}(d_k) \propto \sum_{\mathbf{d}:d_k} \mathbf{I}_c(\mathbf{d}) \prod_{j \neq k} q_j^{(n+1)}(d_j) \tag{6.24}$$

On retrouve ainsi les équations du décodage itératif classique.
En général, l'algorithme s'arrête quand un compromis est atteint entre les APP évaluées par le démodulateur et les APP calculées par le décodeur. Cela correspond avec nos notations à :

$$q_k^{(it)}(d_k) l_k^{(it-1)}(d_k) \propto q_k^{(it)}(d_k) l_k^{(it)}(d_k) \tag{6.25}$$

avec $d_k \in \{0; 1\}$ et pour tout $k \in \{1, ..., n\}$.

6.4 Simulation

Les simulations sont faites avec les paramètres suivants :

- Nombre de bits (avant codage) : $n = 400$
- Modulation : 16QAM
- Code convolutif : [5 7]

L'objectif est de voir si le critère $\tilde{\mathcal{C}}_k$ permet de distinguer les séquences pour lesquelles la solution obtenue est proche du maximum de vraisemblance (dans ce cas le nombre d'erreurs devrait être faible) des séquences pour lesquelles l'optimisation n'a pas conduit au maximum de vraisemblance. Dans les sections précédentes le critère $\tilde{\mathcal{C}}_k$ donnait un rôle à part au bit k ($1 \leq k \leq n$). Nous définissons ici le critère $\tilde{\mathcal{C}}$ par $\tilde{\mathcal{C}} = \sum_{k=1}^{n} \log(\tilde{\mathcal{C}}_k)$. Nous allons devoir fixer un seuil tel que : si $\tilde{\mathcal{C}}$ est au dessus du seuil alors la solution obtenue est considérée comme acceptable sinon elle est considérée comme suspecte. Ces simulations sont faites pour des rapports signaux à bruit faisant apparaître des solutions avec peu d'erreurs mais également pour certaines séquences un nombre important d'erreurs. Nous avons fait des simulations pour des EbN0 égaux à 4dB, 5dB et 6dB. Les

EbN0	4dB	5dB	6dB
Seuil	-50	-30	-10
BER_a	$7,9.10^{-3}$	$2,95.10^{-3}$	$1,02.10^{-3}$
BER_s	$0,23$	$0,18$	$1,9.10^{-3}$
% rejet	69%	17,7%	0,8%
% rejet à tort	2,7%	10%	50%

FIG. 6.1 – Effet de EbN0 sur l'évaluation des solutions acceptables et suspectes

Seuil	-20	-10	-5
BER_a	$8,78.10^{-4}$	$4,68.10^{-4}$	$2,08.10^{-4}$
BER_s	$0,205$	$0,13$	$9,28.10^{-2}$
% rejet	6,4%	10,8%	14,8%
% rejet à tort	2,5%	36,4%	53,83%

FIG. 6.2 – Effet du seuil sur l'évaluation des solutions acceptables et suspectes

résultats sont rassemblés dans le tableau (6.1) pour lequel BER_a correspond au taux d'erreur binaire pour les séquences jugées acceptables alors que BER_s correspond au taux d'erreur binaire pour les séquences jugées suspectes. Le pourcentage de rejet indique le pourcentage de séquences pour lesquelles la solution obtenue est jugée suspecte. Le pourcentage de rejet à tort correspond au nombre de séquences pour lesquelles le nombre d'erreurs est inférieur à 20 et qui ont été jugées suspectes (à tort).

On peut constater pour EbN0=6dB qu'il y a peu de séquences pathologiques puisque le taux de rejet est faible (0.8%) et le taux d'erreur binaire est semblable pour les séquences acceptées et suspectes. En revanche pour 4dB et 5dB, les taux d'erreurs binaires BER_a et BER_s sont très différents. Le critère permet de sélectionner correctement les séquences. Le pourcentage de rejet à tort reste faible.

Considérons maintenant un environnement avec un EbN0 variant dans l'intervalle [4dB - 12dB]. Plusieurs seuils sont testés. Le pourcentage de rejet à tort correspond à des solutions présentant moins de 6 erreurs. La simulation est arrêtée lorsque l'on atteint 200 erreurs sur les solutions acceptées (tableau 6.2).

On constate que l'augmentation du seuil diminue (modestement) le taux d'erreur binaire BER_a cela se fait au détriment du pourcentage de rejet qui augmente. Les histogrammes ci-après montrent la répartition du nombre d'erreurs/séquence pour les solutions acceptées et pour les solutions suspectes. On constate sur les histogrammes correspondant au

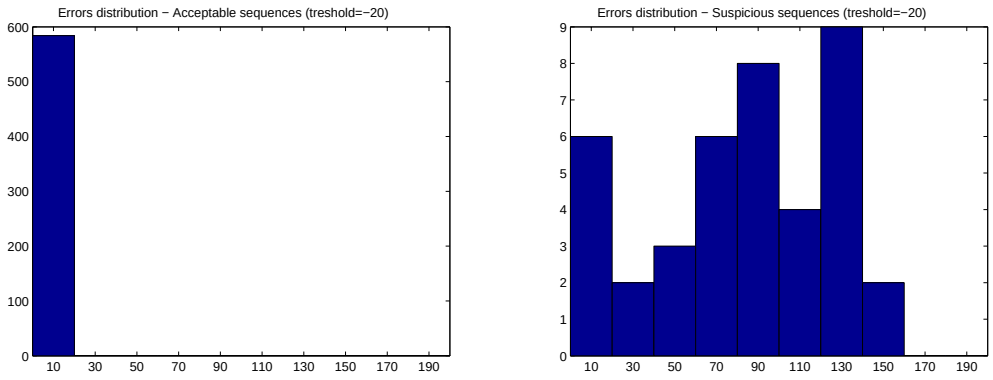

FIG. 6.3 – Histogramme des erreurs (seuil = -20) - Gauche : solutions acceptées - Droite : solutions suspectes

Seuil	-20
BER_a	$9,44.10^{-4}$
BER_s	$0,294$
% rejet	$35,34\%$
% rejet à tort	$0,65\%$

FIG. 6.4 – Seuil fixé à -20

seuil de -20 que les solutions acceptées présentent toutes peu d'erreurs (< 20) alors que les solutions rejetées peuvent présenter jusqu'à 153 erreurs (sur 400) avec une forte majorité de solutions présentant plus de 20 erreurs (seulement 6 ont moins de 20 erreurs). Ces résultats sont représentés sur la figure (6.3) et dans le tableau (6.4) correspondant. Le critère permet donc une séparation correcte des solution acceptables et pathologiques. Lorsque l'on augmente le seuil à -10, on s'aperçoit que le nombre de solutions rejetées pour lesquelles le nombre d'erreurs est inférieur à 20 a largement augmenté (60 contre 6 pour un seuil de -20). Cela a toutefois permis de diviser par 2 le BER_a.
Toutes ces simulations ont été réalisées en utilisant les équations classiques du décodage itératif (6.23-6.24). La simulation suivante a été faite en utilisant les équations alternatives (6.18-6.19). On reprend la situation d'un EbN0 variant entre 4dB et 12dB. Le seuil est égal à -20. Le pourcentage de rejet à tort correspond à des solutions présentant moins de 6 erreurs. La simulation est arrêtée lorsque l'on atteint 200 erreurs sur les solutions acceptées. Les résultats sont rassemblés dans le tableau ci-dessous.

Ces équations donnent de moins bons résultats que les équations classiques. Les erreurs sont plus nombreuses (BER_a et BER_r en augmentation ainsi que pourcentage de rejet

(peut-être dû à la nature séquentielle de l'émetteur : codage puis modulation qui impose au récepteur de commencer par le démodulateur puis le décodeur).

6.5 conclusion

Dans ce chapitre, nous avons dérivé le turbo décodage itératif à partir du décodage par maximum de vraisemblance. Les approximations nécessaires pour obtenir une solution calculable sont bien montrées et quelques preuves concernant le maximum global du critère approché sont établies. Nous avons aussi montré que le turbo décodage pourra être obtenu à partir d'une implémentation hybride de type Jacobian/Gauss-Seidel du processus de maximisation. La propagation des extrinsèques est naturellement introduite, c'est une conséquence directe des équations de mise à jour. Dans la partie simulation, nous avons présenté une éventuelle application de ces résultats.

7

Conclusion générale et perspectives

Dans ce mémoire, nous avons présenté les notions de base de la géométrie de l'information, une des techniques les plus efficaces pour l'analyse et l'illustration des algorithmes itératifs tout en montrant les outils mathématiques utilisés. Nous avons aussi étudié un algorithme itératif d'optimisation bien connu : l'algorithme itératif du point proximal et ses éventuels applications aux algorithmes de décodage itératif.

Nous avons aussi traité l'algorithme classique de Blahut-Arimoto, un algorithme itératif pour le calcul de la capacité d'un canal arbitraire discret sans mémoire comme exemple d'un algorithme itératif opérant avec des estimations de densités de probabilité. Nous avons proposé une interprétation géométrique de cet algorithme basée sur des projections de lois de probabilité sur des familles linéaires et exponentielles de probabilité bien précises dans le but d'introduire l'approche proximale. Nous avons ainsi proposé deux classes d'algorithmes itératifs pour le calcul de la capacité tout en montrant que

l'algorithme classique de Blahut-Arimoto (BA) est un cas particulier de notre étude. La formulation de ces algorithmes est basée sur une approche gradient naturel et la méthode du point proximal. Une analyse théorique de la convergence ainsi que des résultats de simulation ont montré que nos deux nouveaux algorithmes apportent un bonus par rapport au cas classique notamment en ce qui concerne l'accélération de la vitesse de convergence (convergence super linéaire).

Ensuite, nous sommes passés à d'autres applications de la géométrie de l'information et de l'algorithme du point proximal toujours pour des algorithmes itératifs : le décodage itératif des BICM. Plusieurs reformulations et interprétations (basées sur la géométrie de l'information) du problème de décodage itératif sont proposées dans le but de trouver une interprétation de type point proximal et profiter de quelques propriétés importantes de cette méthode quant à la nature de convergence (convergence super linéaire). Nous avons proposé une interprétation point proximal par blocs séparés (interprétation point proximal de chacun des blocs de démodulateur et de décodeur pris séparément) assurant la diminution d'un critère bien précis au fil des itérations mais ne garantissant toujours pas la convergence. Des développements analytiques restent à effectuer quant à la convergence de processus global de cette approche proximale du décodage itératif afin de mettre en évidence le bonus qu'elle apporte par rapport au cas classique où la convergence n'est pas assurée.

Nous avons aussi présenté le lien entre le décodage itératif des BICM et le décodage par maximum de vraisemblance. Nous avons montré qu'en partant du critère de maximum de vraisemblance nous pouvons retrouver les équations classiques de décodage itératif. Une approximation est cependant nécessaire pour établir le lien et obtenir une solution qui pourra être calculée analytiquement. Nous avons aussi montré que le décodage turbo peut être obtenu à partir d'une implémentation hybride, de type Jacobian/Gauss-Seidel, du processus de maximisation. La propagation des extrinsèques est naturellement introduite, c'est une conséquence directe de la mise à jour. Enfin, une partie de simulation montre une application possible des résultats obtenus.

Notons que les résultats obtenus pendant cette thèse ont ouvert beaucoup des pistes devant nous pour une éventuelle suite de ces travaux et une éventuelle généralisation de

ces résultats. Nous citons ainsi le lien qui pourra exister entre la géométrie de l'information d'une part et les Exit charts. En effet, l'interprétation des algorithmes itératifs utilisant la géométrie de l'information est basée sur la convergence du processus turbo itératif pour une seule réalisation du canal. Cependant, les Exit charts sont basés sur un modèle statistique approché des LLRs, i.e., les Exit charts fournissent des distributions approchées des quantités (LLRs) sur lesquelles la géométrie de l'information est basée. Le but de ces travaux sera donc d'obtenir une interprétation des Exit charts basée sur la géométrie de l'information, ainsi qu'une analyse statistique (respectant les coefficients du canal) des caractéristiques de convergence obtenue par la géométrie de l'information.

En plus, et en ce qui concerne la partie approche point proximal du décodage itératif des BICMs, une étude analytique de convergence est à considérer mettant en évidence la nature de la convergence de ce processus itératif (convergence linéaire, convergence super linéaire, quadratique, ...) ainsi que le choix des pas associés afin de proposer certaines améliorations vis à vis du cas classique.

Ajoutons à cela, l'extension des résultats obtenus quant à l'interprétation géométrique des algorithmes de décodage itératifs sur une classe large d'algorithmes itératifs, citons comme exemple les turbo codes, les LDPC codes et autres en se servant des divers travaux effectués dans ce sens.

Bibliographie

[AD98] S. Amari et S. C. Douglas : Why natural gradient ? *In Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 1213–116, May 1998.

[AN00] S. Amari et H. Nagaoka : Methods of information geometry. *American Mathematical Society and Oxford University Press*, 2000.

[AND11] F. Alberge, Z. Naja et P. Duhamel : From maximum likelihood to iterative decoding. *In Proc. International Conference on Acoustics, Speech and Signal Processing*, 2011.

[Ari72] S. Arimoto : An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Trans. Inf. Theory*, 18:14–20, 1972.

[Aus87] A. Auslender : Numerical methods for nondifferentiable convex optimization. *Mathematical Programming Study*, 30:102–126, 1987.

[BBC04] H. Bauschke, M. Borwein et L. Combettes : Bregman monotone optimization algorithms. *SIAM Journal on control and optimization*, 42(2):596–636, 2004.

[BBL00] J.J. Boutros, F. Boixadera et C. Lamy : Bit-interleaved coded modulations for multiple-input multiple-output channels. *In Proc. IEEE 6th Int. Symp. on Spread-Spectrum Tech. & Appli.*, pages 123–126, New Jersy, USA, Sept. 2000.

[BCJR74] L. Bahl, J. Cocke, F. Jelinek et J. Raviv : Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Inform. Theory*, 20(2):284–287, Mar. 1974.

[BDMP97] S. Benedetto, D. Divsalar, G. Montorsi et F. Pollara : A soft-input soft-output APP module for iterative decoding of concatenated codes. *IEEE Commun. Letters*, 1:22–24, Jan 1997.

[BGT93] C. Berrou, A. Glavieux et P. Thitimajshima : Near Shannon limit error-correcting coding and decoding : Turbo codes. *In Proc. IEEE Int. Conf. Commun*, pages 1064–1070, 1993.

[Bla72] R. E. Blahut : Computation of channel capacity and rate-distortion functions. *IEEE Trans. Inf. Theory*, 18:460–473, 1972.

[CGB98] G. Caire, G.Taricco et E. Biglieri : Bit-interleaved coded modulation. *IEEE Trans. Inf. Theory*, 4:927–946, May 1998.

[CH98] S. Chretien et A. Hero : Acceleration of the em algorithm via proximal point iterations. *In IEEE international symposium on information theory*, 1998.

[CH99] S. Chrétien et Alfred O. Hero : Kullback Proximal Algorithms for Maximum Likelihood Estimation. Rapport technique, INRIA,RR-3756, Aug 1999.

[CH08] S. Chretien et A. Hero : On em algorithms and their proximal generalizations. *ESAIM*, 12:308–326, May 2008.

[CL93] R. Correa et C. Lemaréchal : Convergence of some algorithms for convex minimization. *Mathematical Programming Study*, 62:261–275, 1993.

[Csi75] I. Csiszár : I-divergence geometry of probability distributions and minimization problems. *Ann. Prob.*, 3:146–158, 1975.

[CT84] I. Csiszár et G. Tusnády : Information geometry and alternating minimization procedure. *Statistics and Decisions*, supplement issue 1:205–237, 1984.

[CT91] T. M. Cover et J. A. Thomas : *Elements of Information Theory.* Wiley, New York, 1991.

[Dau06] J. Dauwels : *On Graphical Models for Communications and Machine Learning : Algorithms, Bounds, and Analog Implementation.* Thèse de doctorat, May 2006.

[DYW04] F. Dupuis, W. Yu et F. Willems : Arimoto-Blahut algorithms for computing channel capacity and rate-distortion with side-information. *In ISIT*, 2004.

[FG98] G.J. Foshini et M.J. Grans : On limits of wireless communication in a fading environment when using multiple antennas. *Wireless Personnal Communications*, 6(3):311–335, Mar. 1998.

[Gal68] R. G. Gallager : *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[GH01] H. El Gamal et A.R. Hammons : Analysing the turbo decoder using the Gaussian approximation. *IEEE Trans. on Inform. Theory*, 47:671–686, Feb. 2001.

[Gre90] P. J. Green : On use of the EM algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society*, 1990.

[ITA04] S. Ikeda, T. Tanaka et S. Amari : Information geometry of turbo and low-density parity check codes. *IEEE Trans. Inf. Theory*, 50(6):1097–1114, June 2004.

[Kav01] A. Kavcic : On the capacity of markov sources over noisy channels. *In IEEE GLOBECOM*, pages 2997–3001, Nov. 2001.

[KFL01] F.R. Kschischang, B.J. Frey et H.A. Loeliger : Factor graphs and the sum-product algorithm. *IEEE Trans. on Inform. Theory*, 47:498–519, Feb. 2001.

[KLM+06] L. Kocarev, F. Lehmann, G.M. Maggio, B. Scanavino, Z. Tasev et A. Vardy : Nonlinear dynamics of iterative decoding systems : analysis and applications. *IEEE Trans. Inf. Theory*, 52(4):1366–1384, 2006.

[LCR02] X. Li, A. Chindapol et J.A. Ritcey : Bit interleaved coded modulation with iterative decoding and 8-PSK signaling. *IEEE trans Commun.*, 50:1250–1257, Aug 2002.

[LM08] A. Lewis et J. Malicke : Alternating projections on manifolds. *Mathematics of Operations Research*, 33:216–234, Feb. 2008.

[Mar70] B. Martinet : Régularisation d'inéquations variationnelles par approximations successives. *Rev. Francaise Inf. Rech. Oper.*, pages 154–159, 1970.

[MD04] G. Matz et P. Duhamel : Information geometric formulation and interpretation of accelerated Blahut-Arimoto-Type algorithms. *In Proc. Information Theory Workshop*, 2004.

[MDdC02] B. Muquet, P. Duhamel et M. de Courville : A geometrical interpretation of iterative turbo decoding. *In Proc. Int. Symposium on Inform. Theory*, Lausanne, Switzerland, May 2002.

[MG98] M. Moher et T.A. Gulliver : Cross-entropy and iterative decoding. *IEEE Trans. on Inform. Theory*, 44(7):3097–3104, Nov. 1998.

[Muq01] B. Muquet : *Novel receiver and decoding schemes for wireless OFDM systems with cyclic prefix or zero padding.* PhD Thesis, 2001.

[NAD09] Z. Naja, F. Alberge et P. Duhamel : Geometrical interpretation and improvements of the blahut-arimoto's algorithm. *In Proc. International Conference on Acoustics, Speech and Signal Processing*, 2009.

[Nag98] H. Nagaoka : Algorithm of arimoto-blahut type for computing quantum channel capacity. *In IEEE International Symposium on Information Theory*, page 354, Aug. 1998.

[Pea88] J. Pearl : *Probabilistic Reasoning in Intelligent Systems : Network of Plausible Inference.* San Francisco, CA : Morgan Kaufmann, 1988.

[RG04] M. Rezaeian et A. Grant : A generalization of the arimoto-blahut algorithm. *In IEEE International Symposium on Information Theory*, June/July 2004.

[Ric00] T. Richardson : The geometry of turbo-decoding dynamics. *IEEE Trans. on Inform. Theory*, 46(1):9–23, 2000.

[Roc76] R. T. Rockafellar : Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.

[RW07] P. Regalia et J. Walsh : Optimality and duality of the turbo decoder. *In Proceedings of the IEEE*, volume 95 Issue : 6, pages 1362–1377, June 2007.

[tB01] S. ten Brink : Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE trans Commun.*, 49:1727–1737, Oct 2001.

[Tse04] P. Tseng : An analysis of the em algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29:27–44, Feb. 2004.

[Ung82] G. UNGERBOECK : Channel coding with multilevel/phase signals. *IEEE Trans. on Inform. Theory*, 28(1):55–67, Jan. 1982.

[Ung87] G. UNGERBOECK : Trellis-coded modulation with redundant signal sets part 1 and part 2. *IEEE Communications Magazine*, 25(2):5–21, Feb. 1987.

[Von01] P. O. VONTOBEL : A generalized Blahut-Arimoto algorithm. *In In Proc. IEEE GLOBECOM-2001*, pages 2997–3001. Wiley, New York, 2001.

[VWZP89] A. VITERBI, J. WOLF, E. ZEHAVI et R. PADOVANI : A pragmatic approach to trellis-coded modulation. *IEEE Communications Magazine*, 27(7):11–19, July 1989.

[WJR05] J. WALSH, C.R. JOHSON et P. REGALIA : A refined information geometric interpretation of turbo decoding. *In Proc. International Conference on Acoustics, Speech and Signal Processing*, 2005.

[WRJ05] J. WALSH, P. REGALIA et C.R. JOHSON : A convergence proof for the turbo decoder as an instance of the gauss-seidel iteration. *In IEEE International Symposium on Information Theory*, pages 734–738, Sept. 2005.

[WRJ06] J. M. WALSH, P.A. REGALIA et C. R. JOHNSON : Turbo decoding as Iterative Constrained Maximum-Likelihood Sequence Detection. *IEEE Trans. Inf. Theory*, 52:5426–5437, Dec. 2006.

[Zeh92] E. ZEHAVI : 8-PSK trellis codes for a Rayleigh fading channel. *IEEE Trans. Commun.*, 40:873–883, May 1992.

A

Accelerating the Blahut-Arimoto-Algorithm via Information Geometry

Accelerating the Blahut-Arimoto-Algorithm via Information Geometry

Gerald Matz†, Ziad Naja‡, Florence Alberge‡, and Pierre Duhamel‡

† Institute of Telecommunications Vienna University of Technology Gusshausstrasse 25/389 A-1040 Wien, AUSTRIA

‡ L2S, UMR 8506 CNRS - SUPELEC - Univ Paris-Sud, 3 Rue Joliot-Curie, F-91190 Gif-sur-Yvette, FRANCE

phone: +33 1 69 85 17 57, fax: +33 1 69 85 17 65

email: gerald.matz@tuwien.ac.at, {ziad.naja,florence.alberge,pierre.duhamel}@lss.supelec.fr

Abstract. We propose two related classes of iterative algorithms for computing the capacity of discrete memoryless channels. The celebrated Blahut-Arimoto algorithm is a special case of our framework. The formulation of these algorithms is based on the natural gradient and proximal point methods. We also provide interpretations in terms of notions from information geometry. A theoretical convergence analysis and simulation results demonstrate that our new algorithms have the potential to significantly outperform Blahut-Arimoto in terms of convergence speed.

I. Introduction

More than 30 years ago, R. Blahut and S. Arimoto simultaneously proposed an algorithm for computing channel capacity and rate-distortion functions [1, 2]. Both received the Information Theory Paper Award since these papers were a major breakthrough providing practical means to determine the capacity of channels and rate-distortion pairs of sources for which analytical solutions are impossible to obtain. Originally devised for discrete memoryless channels (DMCs), the Blahut-Arimoto (BA) algorithm nowadays has become the standard tool to numerically evaluate capacities of more difficult

Parts of this paper were previously presented at ITW 2004, San Antonio (TX).
Funding by FWF Grant N10606

channels. For example, there exist extensions to ISI channels [3, 4], multi-access channels [5], channels with side information [6], and quantum channels [7].

In [8], an information geometric interpretation of the Blahut-Arimoto (BA) algorithm in terms of alternating information projections was provided.

In this paper, we reconsider the problem of computing the capacity of DMCs from an information geometric point of view. This approach has been pioneered in [8], where the BA algorithm was interpreted in terms of alternating information projections. The main contributions of the paper are:

- It is shown that capacity computation is equivalent to an information geometric "equidivergence" problem.
- We propose a natural gradient algorithm [9] and an accelerated BA algorithm for capacity computation. We demonstrate that close to the optimum solution the recursions of the accelerated BA and NG algorithms are approximately equivalent.
- The accelerated BA and NG algorithms are rephrased in a unifying proximal point framework. The penalty terms between successive iterates is, respectively, their Kullback-Leibler divergence and chi-square divergence.
- We provide a convergence analysis of the accelerated BA algorithm which roughly also characterizes the convergence of the NG algorithm.
- Numerical experiments confirm our theoretical results and verify that our algorithms converge significantly faster than BA.

The rest of the paper is organized as follows. Section II provides some necessary background. The information geometric "equidivergence" game is described in Section III. In Section IV, we introduce the natural gradient algorithm and the accelerated BA algorithm. The unifying framework in terms of proximal point methods is discussed in Section VI. In Section VII we provide some convergence results. Simulation examples are illustrated in Section VIII. A brief summary of the relevant information geometric facts and notions is provided in Appendix.

II. Background

We consider a DMC with input symbol X taken from the size $M+1$ input alphabet $\{x_0, \ldots, x_M\}$, output symbol Y in the size $N+1$ alphabet $\{y_0, \ldots, y_N\}$, and transition probabilities $Q_{i|j} = \Pr(Y = y_i|X = x_j)$. We define the $(N+1) \times (M+1)$ channel matrix $\mathbf{Q}$ as $[\mathbf{Q}]_{ij} = Q_{i|j}$. The distributions of the input and output symbol are characterized by the probability vectors $\mathbf{p} = [p_0 \ldots p_M]^T$ and $\mathbf{q} = [q_0 \ldots q_N]^T = \mathbf{Q}\mathbf{p}$, respectively, with $p_j = \Pr(X = x_j)$ and $q_i = \Pr(Y = y_i) = \sum_{j=0}^{M} Q_{i|j}\, p_j$.

The mutual information $H(Y) - H(Y|X)$ of X and Y equals [10, 11]

$$I(\mathbf{Q}, \mathbf{p}) = \sum_{j=0}^{M} \sum_{i=0}^{N} p_j\, Q_{i|j} \log \frac{Q_{i|j}}{q_i}\,.$$

Here, $\mathbf{Q}_j = [Q_{0|j} \ldots Q_{N|j}]^T$ denotes the jth column of $\mathbf{Q}$. The capacity of the channel equals

$$C(\mathbf{Q}) = \max_{\mathbf{p}} I(\mathbf{Q}, \mathbf{p}).$$

A useful reformulation of mutual information is given by

$$I(\mathbf{Q}, \mathbf{p}) = \sum_{j=0}^{M} p_j D(\mathbf{Q}_j \| \mathbf{q})$$

where we used the Kullback-Leibler divergence (KLD) [10], defined as

$$D(\mathbf{p}\|\mathbf{p}') = \sum_j p_j \log \frac{p_j}{p'_j}\,.$$

While being asymmetric and not satisfying the triangle inequality, the KLD can be interpreted as a (squared) distance measure for probability distributions.

The Kuhn-Tucker conditions [11]

$$D(\mathbf{Q}_j \| \mathbf{Q}\mathbf{p}^*) = C, \quad p_j^* > 0, \tag{1a}$$

$$D(\mathbf{Q}_j \| \mathbf{Q}\mathbf{p}^*) < C, \quad p_j^* = 0, \tag{1b}$$

are necessary and sufficient for an input distribution $\mathbf{p}^*$ to be capacity-achieving.

We note that for any input distribution $\mathbf{p}$ we have the inequalities [1, 11]

$$\sum_{j=0}^{M} p_j D(\mathbf{Q}_j \| \mathbf{Q}\mathbf{p}) \;\le\; C \;\le\; \max_j D(\mathbf{Q}_j \| \mathbf{Q}\mathbf{p})\,. \tag{2}$$

These inequalities become equalities in the case of a capacity-achieving input distribution and can be used as a termination criterion for all the iterative algorithms below.

The BA algorithm [1, 2] builds on the key observation that

$$I(\mathbf{Q}, \mathbf{p}) = \max_{\mathbf{P}} J(\mathbf{Q}, \mathbf{p}, \mathbf{P})$$

where

$$J(\mathbf{Q}, \mathbf{p}, \mathbf{P}) = \sum_{j=0}^{M} \sum_{i=0}^{N} p_j Q_{i|j} \log \frac{P_{j|i}}{p_i}$$

and $\mathbf{P}$ is an arbitrary $(M+1) \times (N+1)$ transition probability matrix with entries $[\mathbf{P}]_{ji} = P_{j|i}$. Hence, it follows that

$$C(\mathbf{Q}) = \max_{\mathbf{p}} \max_{\mathbf{P}} J(\mathbf{Q}, \mathbf{p}, \mathbf{P}) \,.$$

The BA algorithm alternatively performs these two maximizations in an iterative fashion starting from an initial guess $\mathbf{p}^0$, i.e.,

$$\mathbf{P}^{k+1} = \arg\max_{\mathbf{P}} J(\mathbf{Q}, \mathbf{p}^k, \mathbf{P}), \tag{3a}$$

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}} J(\mathbf{Q}, \mathbf{p}, \mathbf{P}^{k+1}). \tag{3b}$$

In [8], these maximizations were re-interpreted as alternating projections based on KLD minimizations. The maximizations in (3) can be solved explicitly. Combining the solutions into one step yields the BA recursion [1, 2]

$$p_j^{k+1} \;=\; p_j^k \, \frac{\exp(D_j^k)}{\sum_{j=0}^{M} p_j^k \exp(D_j^k)}. \tag{4}$$

Here, $D_j^k \triangleq D(\mathbf{Q}_j \| \mathbf{q}^k)$ with $\mathbf{q}^k = \mathbf{Q}\mathbf{p}^k$ is the KLD of the current output distribution $\mathbf{q}^k$ and the jth column of $\mathbf{Q}$.

III. An Equidivergence Game

Based on the arguments below, capacity computation is equivalent in information geometric terms to the following "equidivergence" game (for simplicity of exposition, we restrict to the case $p_j > 0$, $j = 0, \ldots, M$). Consider the set of length-$N+1$ probability vectors $\mathbf{q}$ that have the same KLD to all columns of $\mathbf{Q}$:

$$\mathcal{Q} = \big\{ \mathbf{q} : \; D(\mathbf{Q}_0 \| \mathbf{q}) = D(\mathbf{Q}_1 \| \mathbf{q}) = \cdots = D(\mathbf{Q}_N \| \mathbf{q}) \big\}. \tag{5}$$

It can be shown that this is an log-linear (exponential) [12, 13] family of probability vectors. Hence, the reverse I-projection [12, 13] of the jth column of $\mathbf{Q}$ onto $\mathcal{Q}$, defined as

$$\mathbf{q}^* = \arg\min_{\mathbf{q}\in\mathcal{Q}} D(\mathbf{Q}_j\|\mathbf{q})\,, \tag{6}$$

belongs to the linear (mixture) family

$$\mathcal{L} = \left\{\mathbf{q}:\ \mathbf{q} = \mathbf{Q}\mathbf{p} = \sum_{j=0}^{N} \mathbf{Q}_j\, p_j, \quad \sum_{j=0}^{N} p_j = 1\right\},$$

which is dual to $\mathcal{Q}$. Using the compensation identity

$$\sum_{j=0}^{N} p_j D(\mathbf{Q}_j\|\mathbf{Q}\mathbf{p}) = \sum_{j=0}^{N} p_j D(\mathbf{Q}_j\|\mathbf{q}) - D(\mathbf{Q}\mathbf{p}\|\mathbf{q}),$$

with $\mathbf{q} = \mathbf{q}^*$, it follows that $\sum_{j=0}^{N} p_j D(\mathbf{Q}_j\|\mathbf{Q}\mathbf{p}) \le D(\mathbf{Q}_j\|\mathbf{q}^*)$, with equality iff $\mathbf{Q}\mathbf{p} = \mathbf{q}^*$, and thus

$$\mathbf{q}^* = \mathbf{Q}\mathbf{p}^* \quad \text{with} \quad \mathbf{p}^* = \arg\max_{\mathbf{p}} \sum_{j=0}^{N} p_j D(\mathbf{Q}_j\|\mathbf{Q}\mathbf{p})\,.$$

We conclude that the equi-divergence game (6) is equivalent to capacity computation.

From an algorithmic point of view, the equidivergence game means that given a current guess $\mathbf{p}^k$, we should check the KLDs $D_j^k = D(\mathbf{Q}_j\|\mathbf{Q}\mathbf{p}^k)$ and move the output distribution closer to those $\mathbf{Q}_j$ for which D_j^k is large. This can be achieved by increasing the respective weights p_j^k, consistent with the BA recursion (4) that increases (decreases) those input probabilities for which $\exp(D_j^k)$ is above (below) the average $\sum_{j=0}^{M} p_j \exp(D_j^k)$.

IV. Natural Gradient Algorithm

In this section, we propose a novel algorithm for capacity computation that is based on the *natural gradient* (NG) [9]. Ordinary gradient ascent methods for maximizing $I(\mathbf{p})$ in general fail since they usually are incapable to preserve the property that the iterate is a probability vector. We are going to exploit the fact that the input probability vectors $\mathbf{p}$ constitute an M-dimensional Riemannian manifold with the Fisher information matrix representing the associated Riemannian metric. The advantage of the NG approach is that the curvature of this manifold is inherently taken into account [9].

To formulate our NG algorithm, we describe the input probability vectors in terms of their M dual (or expectation) parameters $\eta_j = p_j$, $j = 1, \ldots, M$ (note that $p_0 = 1 - \sum_{j=1}^{M} \eta_j$). In terms of these

parameters, the score function $\bar{I}(\boldsymbol{\eta}) = I(\mathbf{p})$ we want to maximize reads

$$\bar{I}(\boldsymbol{\eta}) = \sum_{j=1}^{M}\sum_{i=0}^{N} \eta_j Q_{i|j} \log \frac{Q_{i|j}}{q_i} + \Big[1 - \sum_{j=1}^{M} \eta_j\Big] \sum_{i=0}^{N} Q_{i|0} \log \frac{Q_{i|0}}{q_i}$$

with $q_i = \sum_{j=1}^{M} Q_{i|j}\eta_j + Q_{i|0}\big[1 - \sum_{j=1}^{M} \eta_j\big]$. To climb the peak of this score function, we propose a NG ascent algorithm with the parameter updates

$$\boldsymbol{\eta}^{k+1} = \boldsymbol{\eta}^k + \mu_k \widetilde{\nabla} \bar{I}(\boldsymbol{\eta}^k)\,, \tag{7}$$

where μ_k is a step-size parameter and $\widetilde{\nabla}\bar{I}(\boldsymbol{\eta})$ is the NG of $\bar{I}(\boldsymbol{\eta})$ obtained by pre-multiplying the ordinary gradient with the inverse of the Riemannian metric $\mathbf{G}(\boldsymbol{\eta})$:

$$\widetilde{\nabla}\bar{I}(\boldsymbol{\eta}) = \mathbf{G}^{-1}(\boldsymbol{\eta})\, \nabla \bar{I}(\boldsymbol{\eta}), \qquad \nabla \bar{I}(\boldsymbol{\eta}) = \Big[\frac{\partial \bar{I}(\boldsymbol{\eta})}{\partial \eta_1} \dots \frac{\partial \bar{I}(\boldsymbol{\eta})}{\partial \eta_M}\Big]^T.$$

Using the Riemannian metric $\mathbf{G}(\boldsymbol{\eta})$ derived in Appendix and the fact that the jth component of $\nabla \bar{I}(\boldsymbol{\eta})$ equals $D(\mathbf{Q}_j\|\mathbf{q}) - D(\mathbf{Q}_0\|\mathbf{q})$, the jth component of the NG is obtained as

$$\big[\widetilde{\nabla}\bar{I}(\boldsymbol{\eta})\big]_j = \big[D(\mathbf{Q}_j\|\mathbf{q}) - \bar{I}(\boldsymbol{\eta})\big]\, \eta_j\,. \tag{8}$$

In accordance with our previous observations, the NG has the property of pointing towards the directions for which the KLD $D(\mathbf{Q}_j\|\mathbf{q})$ is (too) large. Note also that for a stationary point $\boldsymbol{\eta}^*$, $\big[\widetilde{\nabla}\bar{I}(\boldsymbol{\eta}^*)\big]_j = 0$ implies that either $D(\mathbf{Q}_j\|\mathbf{q}^*) = \bar{I}(\boldsymbol{\eta}^*)$ or $\eta_j^* = p_j^* = 0$, consistent with the Kuhn-Tucker conditions (1).

Plugging (8) into (7) and using the definitions $p_j^k = \eta_j^k$, $j = 1, \dots, M$, $p_0^k = 1 - \sum_{j=1}^{M} \eta_j^k$, the NG recursions for the input probabilities can be shown to be (recall that $D_j^k = D(\mathbf{Q}_j\|\mathbf{q}^k)$ with $\mathbf{q}^k = \mathbf{Q}\mathbf{p}^k$)

$$p_j^{k+1} \;=\; p_j^k \big[1 + \mu_k (D_j^k - I^k)\big]. \tag{9}$$

Here, we used the short-hand notation $I^k \triangleq I(\mathbf{p}^k)$ for the current estimate (actually, lower bound) of capacity. Note that like the BA recursion, (9) amounts to a multiplicative update. However, the computational complexity of the NG update is slightly less than that of the BA update since the exponentiation is avoided. Furthermore, the NG update (9) is consistent with information geometric equidivergence interpretation of capacity computation.

Due to $\sum_{j=0}^{M} p_j\, D_j^k = I^k$, (9) guarantees $\sum_{j=0}^{M} p_j^{k+1} = 1$. However, we have to take care to chose the step-size μ_k small enough such that $p_j^{k+1} \geq 0$. This constraint is in conflict with our desire of choosing

a large step-size for faster convergence. In fact, non-negativity of p_j^{k+1} is ensured for $\mu_k \leq -\frac{1}{\min_j D_j^k - I^k}$. Numerical simulations indicated that choosing μ_k close to $-\frac{1}{\min_j D_j^k - I^k}$ leads to unstable behavior of the NG recursions. Stable convergence was observed e.g. with $\mu_k = \frac{1}{I^k} \leq -\frac{1}{\min_j D_j^k - I^k}$. With this step-size (and also for various fixed step-sizes $\mu_k = \mu > 1$), the NG algorithm can outperform BA significantly in terms of convergence speed. This can be explained as follows.

In contrast to the iterated BA maximizations (3), the NG algorithm can be recast as

$$\mathbf{P}^{k+1} = \arg\max_{\mathbf{P}} J(\mathbf{p}^k, \mathbf{P}),$$
$$\mathbf{p}^{k+1} = \mathbf{p}^k + \mu_k \widetilde{\nabla} J(\mathbf{p}^k, \mathbf{P}^{k+1}),$$

i.e., the second BA maximization is replaced with a NG ascent step. While this step misses the local maximum along the $\mathbf{p}$-axis, it allows to hop much closer to the global maximum. In essence, the NG algorithm has the potential to avoid traversing back and forth a ridge of $J(\mathbf{p}, \mathbf{P})$.

V. Accelerated Blahut-Arimoto Algorithm

A. Ad-Hoc Formulation

When comparing the BA and NG algorithms numerically, it turned out that for fixed $\mu_k = \mu$, the NG algorithm converges μ times faster. Put another way, convergence properties are very much the same for $\mu = 1$.

To see why this is the case, we divide numerator and denominator of (4) by $\exp(I^k)$ and then use a first-order Taylor series approximation of the exponentials:

$$p_j^{k+1} = p_j^k \frac{\exp(D_j^k - I^k)}{\sum_{j=0}^M p_j^k \exp(D_j^k - I^k)} \approx p_j^k \left[1 + (D_j^k - I^k)\right].$$

The right-hand side is the NG recursion (9) for $\mu_k = 1$. The Taylor series approximation is accurate for $D_j^k - I^k \approx 0$, i.e., when equi-divergence is almost achieved which will be true in the vicinity of the optimum solution. We conclude that BA and NG with $\mu_k = 1$ are asymptotically equivalent. In fact, this relation between the two algorithms motivates the *ad hoc* formulation of a generalized BA algorithm:

$$p_j^{k+1} = p_j^k \frac{\exp(\mu_k D_j^k)}{\sum_{j=0}^M p_j^k \exp(\mu_k D_j^k)}. \tag{10}$$

Using the same arguments as before, this algorithm can be shown to be asymptotically equivalent to the NG algorithm. In fact, using the same step-size $\mu_k = \mu$, both algorithms feature the same convergence speed which is μ times faster than that of the ordinary BA algorithm. For that reason, we refer to (10) as *accelerated BA* algorithm. More insights with regard to the convergence behavior of the accelerated BA and NG algorithms and the choice of μ_k are discussed in the context of a proximal point reformulation of these algorithms in Section VI.

B. Interpretation via e- and m-Geodesics

Next, we briefly discuss the information geometric significance of the accelerated BA and NG algorithms. Assume that the current guess for the optimum input and output probabilities are $\mathbf{p}$ and $\mathbf{q} = \mathbf{Q}\mathbf{p}$. Let $\mathbf{p}_{\mathrm{BA}}$ and $\mathbf{p}_{\mathrm{NG}}$ be the probabilities obtained by applying to $\mathbf{p}$ the BA and NG updates (10) and (9) with $\mu_k = 1$ and $\mu_k = 1/I^k$, respectively. It is then easily verified that the accelerated BA and NG updates for general μ_k can be written as

$$\begin{aligned}\mathbf{p}_{\mathrm{BA}}(\mu_k) &= c(\mu_k)\,\mathbf{p}^{1-\mu_k}\,\mathbf{p}_{\mathrm{BA}}^{\mu_k}\,,\\ \mathbf{p}_{\mathrm{NG}}(\mu_k) &= (1-\mu_k\,I_k)\,\mathbf{p} + \mu_k\,I_k\,\mathbf{p}_{\mathrm{NG}}\,,\end{aligned}$$

where $c(\mu_k)$ is a normalization constant. (Note that $\mathbf{p}_{\mathrm{BA}}(0) = \mathbf{p}_{\mathrm{NG}}(0) = \mathbf{p}$.) Hence, the probability vectors $\mathbf{p}_{\mathrm{BA}}(\mu_k)$ constitute an exponential (log-linear) family, parametrized by μ_k, that corresponds to the e-geodesic [12] connecting $\mathbf{p}$ and $\mathbf{p}_{\mathrm{BA}}$. In contrast, the $\mathbf{p}_{\mathrm{NG}}(\mu_k)$ constitute the mixture (linear) family, again parametrized by μ_k, that corresponds to the m-geodesic [12] connecting $\mathbf{p}$ and $\mathbf{p}_{\mathrm{NG}}$. It can be verified that the "extremal" points of $\mathbf{p}_{\mathrm{BA}}(\mu_k)$ are $p_{j,\mathrm{BA}}(-\infty) = \delta_{j-\underline{j}}$ and $p_{j,\mathrm{BA}}(\infty) = \delta_{j-\overline{j}}$, with $\underline{j} = \arg\min_j D(\mathbf{Q}_j\|\mathbf{q})$ and $\overline{j} = \arg\max_j D(\mathbf{Q}_j\|\mathbf{q})$ the indices of the columns of $\mathbf{Q}$ that are closest/farthest to $\mathbf{q}$. On the other hand, the "extremal" points of $\mathbf{p}_{\mathrm{NG}}(\mu_k)$ are the intersection of the straight line connecting $\mathbf{p}_{\mathrm{NG}}(0)$ and $\mathbf{p}_{\mathrm{NG}}(1/I^k)$ with the $\underline{j}$th and $\overline{j}$ face of the probability simplex. Thus, we conclude that with increasing μ_k, $\mathbf{q}_{\mathrm{BA}}(\mu_k)$ moves along an e-geodesic starting at $\mathbf{Q}_{\underline{j}}$ and ending in $\mathbf{Q}_{\overline{j}}$ while $\mathbf{q}_{\mathrm{NG}}(\mu_k)$ moves along an m-geodesic starting in the vicinity of $\mathbf{Q}_{\underline{j}}$ and ending close to $\mathbf{Q}_{\overline{j}}$. For $|\mu_k(D_j^k - I^k)| \ll 1$, these two geodesics virtually coincide in the vicinity of $\mathbf{p}$.

VI. Proximal Point Interpretations

We previously provided some information geometric insights regarding the accelerated BA and NG algorithms and investigated their asymptotic equivalence. In this section, we demonstrate that in fact both algorithms can be derived within a common framework that also provides an *a posteriori* justification for the *ad hoc* definition of the accelerated BA recursions (10).

A. Proximal point interpretation of B.A. and amelioration in terms of convergence speed

The Arimoto-Blahut Algorithm has been interpreted in [14] as an alternate projection algorithm based on the Kullback-Leibler divergence (KLD). This interpretation leads to another proof of the monotonic convergence but does not give insights for accelerating the convergence of the method. The KLD-based projections over linear families of probabilities are related with the proximal point methods. This is a classical optimization technique with possible superlinear convergence. In this section, we re-formulate the classical Blahut-Arimoto algorithm and prove that it can be interpreted as a proximal point.

The proximal point in its basic version has been well studied in [15]. This is an iterative solution for finding the minimum (or maximum) of a convex (concave) function. Let f denote a closed concave function, the proximal point algorithm generates the sequence z^{k+1} such that

$$z^{k+1} \in \arg\max_{z} \left\{ f(z) - \gamma_k \parallel z - z^k \parallel^2 \right\} \tag{11}$$

where γ_k is a positive scalar parameter. The quadratic term $\parallel z - z^k \parallel^2$ can be understood as a regularization term that renders the function to be maximized strictly concave and coercive. As a consequence, the maximum in (11) is attained at a unique point [16]. The degree of regularization is controlled by γ_k. The rate of convergence is shown to be linear if γ_k stays small enough and superlinear if $\gamma_k \to 0$. From the original version, alternatives schemes have been studied that consider the KLD divergence [17] or more generally a Bregman divergence instead of the euclidian distance [18]. Generally speaking a proximal point algorithm generates the sequence z^{k+1} using the following update rule:

$$z^{k+1} \in \arg\max_{z} f(z) - \gamma_k d(z, z^k) \tag{12}$$

where $d(z, z^k)$ is always non-negative and is equal to zero if and only if $z = z^k$. These two conditions on the regularization term are useful to guarantee the monotonic convergence of the sequence z^k. From the alternating minimization (3), we build a new formulation for the Blahut-Arimoto algorithm. The solution of the first maximization is given by

$$P_{j|i}^{k+1} = \frac{Q_{i|j} p_j^k}{\sum_{j'=0}^{M} Q_{i|j'} p_{j'}^k} \tag{13}$$

By plugging this solution into the second maximization problem we obtain

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \sum_{j=0}^{M}\sum_{i=0}^{N} p_j Q_{i|j} log \frac{P_{j|i}^{k+1}}{p_j} \tag{14}$$

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \left\{ \sum_{j=0}^{M} p_j D_j^k - D(\mathbf{p} \parallel \mathbf{p}^k) \right\} \tag{15}$$

where $D_j^k = D(\mathbf{Q}_j \parallel \mathbf{Q}\mathbf{p}^k)$ and where $\tilde{I}^k(\mathbf{p}, \mathbf{Q}) = \sum_{j=0}^{M} p_j D_j^k$ can be viewed as an approximation of the true score function $I(\mathbf{p}, \mathbf{Q})$ obtained by replacing the Kullback-Leibler divergences $D(\mathbf{Q}_j \parallel \mathbf{Q}\mathbf{p})$ by $D(\mathbf{Q}_j \parallel \mathbf{Q}\mathbf{p}^k)$. Written as such, the algorithm in (15) is not a proximal point. In the definition, the score function $f(z)$ is independent of z^k whereas in (15) the score function is a function of $\mathbf{p}^k$. We can give an equivalent expression of the score function in (15) that involves the desirable score function $I(\mathbf{p}, \mathbf{Q})$ as

$$\begin{aligned} \sum_{j=0}^{M} p_j D_j^k &= I(\mathbf{p}, \mathbf{Q}) + \sum_{j=0}^{M}\sum_{i=0}^{N} Q_{i|j} \log \frac{q_i}{q_i^k} && (16) \\ &= I(\mathbf{p}, \mathbf{Q}) + D(\mathbf{q} \parallel \mathbf{q}^k) && (17) \end{aligned}$$

As a consequence, the BA algorithm reads

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \left\{ I(\mathbf{p}, \mathbf{Q}) - \left(D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k) \right) \right\} \tag{18}$$

From this equation, we can conclude that the BA algorithm is a proximal point provided that the penalty term is a distance. The proof is based on the observation that the penalty term reads

$$\begin{aligned} D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k) &= \sum_{i,j} Q_{i|j} p_j \log\left(\frac{p_j}{p_j^k}\right) - \sum_{i,j} Q_{i|j} p_j \log\left(\frac{q_i}{q_i^k}\right) && (19) \\ &= -\sum_{i,j} Q_{i|j} p_j \log\left(\frac{p_j^k q_i}{p_j q_i^k}\right) && (20) \end{aligned}$$

According to Jensen's inequality, we obtain

$$D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k)) \geq -\log\left(\sum_{i,j} \frac{Q_{i|j} p_j p_j^k \sum_{j'} Q_{i|j'} p_{j'}}{p_j \sum_{j'} Q_{i|j'} p_{j'}^k}\right) = 0 \tag{21}$$

As a consequence, the BA algorithm is a proximal point with constant stepsize $\gamma_k = 1$. Since the convergence rate is related with the value of γ_k, we introduce a modified version of the BA algorithm with update rule

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \left\{ I(\mathbf{p}, \mathbf{Q}) - \gamma_k \left(D(\mathbf{p} \parallel \mathbf{p}^k) - D(\mathbf{q} \parallel \mathbf{q}^k) \right) \right\} \tag{22}$$

When γ_k is less than 1 the whole function to be maximized is closer to the mutual information than in the classical BA algorithm and the convergence rate is likely to increase. The solution to this optimization problem (after normalization) is given by :

$$p_j^{k+1} = C^{k+1} p_j^k exp\left(\sum_i Q_{i|j} \log \frac{q_i^{k+1}}{q_i^k} + \frac{1}{\gamma_k} \sum_i Q_{i|j} \log \frac{Q_{i|j}}{q_i^{k+1}} \right) \tag{23}$$

where C^{k+1} is a normalization constant. We can observe that eq. (23) is not a closed-form expression. A second iterative procedure is necessary for the computation of $\mathbf{p}^{k+1}$. In other words, the gain with respect to the convergence rate in the main loop is counterbalanced by a more complicated update rule. In order to have a closed form expression, the right term in (23) should be independent of $\mathbf{p}^{k+1}$ and $\mathbf{q}^{k+1}$. This can be obtained in two ways: *(i)* setting $\gamma_k = 1$, *(ii)* replacing the new value q_i^{k+1} by the value q_i^k of the previous iteration. The first option is exactly the classical BA algorithm. The second option is known as the One Step Late algorithm and was originally proposed in [19]. An intuitive justification for this strategy is that if the algorithm converges slowly, the derivatives computed at q_i^k and q_i^{k+1} will not be much different. Moreover, this method leads to the same fixed points than the original algorithm. In the general case, the OSL algorithms are not guaranteed to converge neither to increase the target score function. Applied to (23), the OSL technique leads to the following update strategy

$$p_j^{k+1} = C^{k+1} p_j^k exp\left(\frac{1}{\gamma_k} \sum_i Q_{i|j} \log \frac{Q_{i|j}}{q_i^k} \right) \tag{24}$$

$$= p_j^k \frac{\exp(\frac{1}{\gamma_k} D_j^k)}{\sum_{j=0}^M p_j^k \exp(\frac{1}{\gamma_k} D_j^k)}. \tag{25}$$

This is exactly the update rule in eq (10) that has been obtained through an ad hoc formulation in section V-A. Eq (25) is the solution of the following optimization problem

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}}\left\{\tilde{I}^k(\mathbf{p},\mathbf{Q}) - \gamma_k D(\mathbf{p}\parallel\mathbf{p}^k)\right\} \tag{26}$$

$$= \arg\max_{\mathbf{p}\in\mathcal{P}}\left\{I(\mathbf{p},\mathbf{Q}) - \left(\gamma_k D(\mathbf{p}\parallel\mathbf{p}^k) - D(\mathbf{q}\parallel\mathbf{q}^k)\right)\right\} \tag{27}$$

The interpretation as a proximal point is relevant as long as the penalty term is comparable to a distance. If the stepsize γ_k is chosen such that $\gamma_k D(\mathbf{p}\parallel\mathbf{p}^k) \geq D(\mathbf{q}\parallel\mathbf{q}^k)$, the convergence of the method is guaranteed and the mutual information will increase with the iterations.

B. NG Update

We next demonstrate that like the accelerated BA algorithm our NG algorithm can be viewed as proximal point method. The modification that is required pertains to the penalty term which in the accelerated BA algorithm is formulated in terms of the KLD. Obviously, there exist countless other distance functions that can be used to force the update to the vicinity of the current guess. Since we are iterating on the manifold of probability vectors, Euclidean distance is not a reasonable choice. In contrast, the general class of f-divergences [20] of probability distributions appears well-suited. For reasons that will become clear presently, we choose the so-called χ^2-divergence defined as

$$\chi^2(\mathbf{p},\mathbf{p}') = \frac{1}{2}\sum_j \frac{(p_j - p'_j)^2}{p'_j}\,.$$

Like for the KLD, $\chi^2(\mathbf{p},\mathbf{p}') \geq 0$ with equality iff $\mathbf{p} = \mathbf{p}'$. It can then easily be shown that the NG update (9) is the solution of the problem

$$\mathbf{p}^{k+1} = \arg\max_{\mathbf{p}}\left\{\tilde{I}^k(\mathbf{p},\mathbf{Q}) - \gamma_k\,\chi^2(\mathbf{p}\|\mathbf{p}^k)\right\},$$

obtained with $\mu_k = 1/\gamma_k$. Thus, accelerated BA and NG can both be viewed as proximal point methods using the same cost function $\tilde{I}^k(\mathbf{p},\mathbf{Q})$ but different distance measure for the proximity penalty. Their asymptotic equivalence follows from the well-known fact that $\chi^2(\mathbf{p},\mathbf{p}') \approx D(\mathbf{p}\|\mathbf{p}')$ for $\mathbf{p}$ close to $\mathbf{p}'$.

C. Choice of Step-Size

A fundamental property of the BA algorithm is that the mutual information $I(\mathbf{p}^k,\mathbf{Q}) = \sum_{j=1}^M p_j^k\, D_j^k$ which represents the current capacity estimate is non-decreasing. For the accelerated BA algorithm,

the proximal point formulation (27) immediately implies

$$\sum_{j=1}^{M} p_j^{k+1}\, D_j^k \;\geq\; I(\mathbf{p}^k) + \gamma_k\, D(\mathbf{p}^{k+1}\|\mathbf{p}^k)\,.$$

Furthermore, it can be shown that $I(\mathbf{p}^{k+1}) = \sum_{j=1}^{M} p_j^{k+1}\, D_j^{k+1} = \sum_{j=1}^{M} p_j^{k+1}\, D_j^k - D(\mathbf{q}^{k+1}\|\mathbf{q}^k)$. Hence,

$$I(\mathbf{p}^{k+1}, \mathbf{Q}) \;\geq\; I(\mathbf{p}^k) + \gamma_k\, D(\mathbf{p}^{k+1}\|\mathbf{p}^k) - D(\mathbf{q}^{k+1}\|\mathbf{q}^k)\,.$$

A sufficient condition for $I(\mathbf{p}^k)$ to be non-decreasing thus is $\gamma_k\, D(\mathbf{p}^{k+1}\|\mathbf{p}^k) - D(\mathbf{q}^{k+1}\|\mathbf{q}^k) \geq 0$. Note that for the ordinary BA algorithm with $\gamma_k = 1/\mu_k = 1$, this condition holds always true as can be seen by applying the log-sum inequality. In general, however, we will have to ensure that

$$\mu_k \leq \frac{D(\mathbf{p}^{k+1}\|\mathbf{p}^k)}{D(\mathbf{q}^{k+1}\|\mathbf{q}^k)} = \frac{D(\mathbf{p}^{k+1}\|\mathbf{p}^k)}{D(\mathbf{Q}\mathbf{p}^{k+1}\|\mathbf{Q}\mathbf{p}^k)}\,. \tag{28}$$

Motivated by the similarity to the squared maximum matrix eigenvalue $\sup_{\mathbf{x}\neq\mathbf{y}} \frac{d_E^2(\mathbf{A}\mathbf{x},\mathbf{A}\mathbf{y})}{d_E^2(\mathbf{x},\mathbf{y})}$ with $d_E^2(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$, we define the *maximum KLD-induced "eigenvalue"* of $\mathbf{Q}$ as

$$\lambda_{\mathrm{KL}}^2(\mathbf{Q}) \triangleq \sup_{\mathbf{p}\neq\mathbf{p}'} \frac{D(\mathbf{Q}\mathbf{p}\|\mathbf{Q}\mathbf{p}')}{D(\mathbf{p}\|\mathbf{p}')}\,.$$

Note that $0 \leq \lambda_{\mathrm{KL}}^2(\mathbf{Q}) \leq 1$ and further $\lambda_{\mathrm{KL}}^2\Big(\frac{1}{N+1}\mathbf{1}_{N+1}\mathbf{1}_{M+1}^T\Big) = 0$, $\lambda_{\mathrm{KL}}^2(\mathbf{I}) = 1$. Thus, small $\lambda_{\mathrm{KL}}^2(\mathbf{Q})$ means that the channel is noisy. Using this definition, a sufficient condition for $I(\mathbf{p}^k, \mathbf{Q})$ to be non-decreasing is given by

$$\mu_k \leq \frac{1}{\lambda_{\mathrm{KL}}^2(\mathbf{Q})}\,. \tag{29}$$

In fact, we observed in our simulations that when using a fixed step-size $\mu_k = \mu$, maximum convergence speed was achieved with $\mu = 1/\lambda_{\mathrm{KL}}^2(\mathbf{Q})$. The problem of course is to obtain a reasonable estimate of $\lambda_{\mathrm{KL}}^2(\mathbf{Q})$. For practical implementations, we thus rather advocate the adaptive step-size $\mu_k = \frac{D(\mathbf{Q}\mathbf{p}^k\|\mathbf{Q}\mathbf{p}^{k-1})}{D(\mathbf{p}^k\|\mathbf{p}^{k-1})}$. While this choice might violate (29) and we were not able to obtain theoretical results regarding convergence speed, we observed excellent performance (in fact, superlinear convergence) in our numerical experiments.

Since in the vicinity of the optimum solution accelerated BA and NG behave identical, the above arguments can also be used to choose the step-size of the NG algorithm.

VII. Convergence Analysis

In the foregoing discussion we saw that larger step sizes in the accelerated BA and NG algorithms allow for "larger hops" and thus have the potential for increased convergence speed. In this section, we provide more explicit results regarding the convergence of the accelerated BA algorithm. While explicit results are difficult to obtain for the NG algorithm, the approximate equivalence to the accelerated BA algorithm suggests that it also inherits the convergence properties of the latter.

A. General Convergence

Let us next turn our attention to convergence, generalizing the arguments of [2]. We assume $\mu_{\inf} = \inf_k \mu_k > 0$. Define $I^k = \sum_j p_j^k D_j^k$ and $L^k = \frac{1}{\mu_k} \log\big(\sum_j r_j^k\big)$ with $r_j^k = p_j^k \exp(\mu_k D_j^k)$. If (29) is satisfied for all k, it can be shown that $C \geq \cdots \geq I^k \geq L^k \geq I^{k-1} \geq \ldots$. Furthermore, with $\mathbf{p}^*$ denoting a capacity-achieving input distribution and $\mathbf{q}^* = \mathbf{Q}\mathbf{p}^*$ the corresponding output distribution,

$$\begin{aligned}
D(\mathbf{p}^*\|\mathbf{p}^k) - D(\mathbf{p}^*\|\mathbf{p}^{k+1}) &= \sum_j p_j^* \log \frac{p_j^{k+1}}{p_j^k} \\
&= \sum_j p_j^* \log \frac{r_j^k}{p_j^k \sum_{j'} r_{j'}^k} \\
&= -\mu_k L^k + \sum_j p_j^* \log \frac{r_j^k}{p_j^k} = -\mu_k L^k + \mu_k \sum_j p_j^* D_j^k \\
&= -\mu_k L^k + \mu_k \sum_j \sum_i Q_{i|j} p_j^* \log \frac{Q_{i|j}}{\sum_{j'} Q_{i|j'} p_{j'}^k} \\
&= -\mu_k L^k + \mu_k \sum_j \sum_i Q_{i|j} p_j^* \log \frac{Q_{i|j}}{q_j^k} \\
&= -\mu_k L^k + \mu_k C + \mu_k D(\mathbf{q}^*\|\mathbf{q}^k) \geq \mu_k\big(C - L^k\big).
\end{aligned}$$

Summing the first an the last expression from $k = 0$ to $k = K-1$, and using $\mu_{\inf} \leq \mu_k$, we obtain

$$\begin{aligned}
\sum_{k=0}^{K-1} \big(C - L^k\big) &\leq \frac{1}{\mu_{\inf}} \big[D(\mathbf{p}^*\|\mathbf{p}^0) - D(\mathbf{p}^*\|\mathbf{p}^K)\big] \\
&\leq \frac{1}{\mu_{\inf}} D(\mathbf{p}^*\|\mathbf{p}^0)\,.
\end{aligned}$$

For $p_j^0 > 0$, the right-hand side is finite and independent of K. Since the sequence $C - L^k$ is non-negative and non-increasing, it follows that

$$\lim_{k\to\infty} L^k = \lim_{k\to\infty} I^k = C\,,$$

and furthermore that the convergence rate is at least proportional to $1/k$,

$$C - L^k < \frac{D(\mathbf{p}^*\|\mathbf{p}^0)}{\mu_{\inf}\,(k+1)}\,.$$

This clearly reflects that the accelerated BA algorithm ($\mu_k > 1$) converges faster than ordinary BA ($\mu_k = 1$).

B. Fixed Points

We next demonstrate that for fixed $\gamma_k = \gamma = 1/\mu$, the fixed points $\mathbf{p}^*$ of the proximal point formulation of accelerated BA and NG achieve capacity. These fixed points satisfy

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} \left\{ \sum_j p_j D(\mathbf{Q}_j, \mathbf{Q}\mathbf{p}^*) - \gamma\, d(\mathbf{p}\|\mathbf{p}^*) \right\},$$

where $d(\mathbf{p}\|\mathbf{p}^k) = D(\mathbf{p}\|\mathbf{p}^k)$ for accelerated BA and $d(\mathbf{p}\|\mathbf{p}^k) = \chi^2(\mathbf{p}, \mathbf{p}^k)$ for NG. Unless $p_j^* = 0$ for some j, this implies that the gradient of the score function on the right-hand side w.r.t. the dual coordinates $\boldsymbol{\eta}$ of $\mathbf{p}$ vanishes at $\mathbf{p}^*$

$$\nabla \left[\sum_j p_j D(\mathbf{Q}_j, \mathbf{Q}\mathbf{p}^*) - \gamma\, d(\mathbf{p}\|\mathbf{p}^*) \right]_{\mathbf{p}^*} = 0.$$

Since it is easily verified that $\nabla d(\mathbf{p}\|\mathbf{p}^*)|_{\mathbf{p}^*} = 0$, we have

$$\nabla \left[\sum_j p_j D(\mathbf{Q}_j, \mathbf{Q}\mathbf{p}^*) \right]_{\mathbf{p}^*} = 0,$$

which implies

$$D(\mathbf{Q}_j, \mathbf{Q}\mathbf{p}^*) = D(\mathbf{Q}_0, \mathbf{Q}\mathbf{p}^*), \qquad j = 1, \ldots, M\,,$$

and thus, by the first part of the Kuhn-Tucker conditions, that $\mathbf{p}^*$ achieves capacity.

VIII. Numerical Example

For purposes of illustration of our results, consider the channel $\mathbf{Q} = \begin{pmatrix} 0.7 & 0.1 \\ 0.2 & 0.2 \\ 0.1 & 0.7 \end{pmatrix}$ from Fig. 4.5.1 in [11]. Let us ignore the fact that this channel can be recognized as being symmetric and that the optimum input distribution is thus uniform. To compute the capacity of this channel, we ran the BA algorithm, the accelerated BA algorithm, and the natural gradient algorithm with (the same)

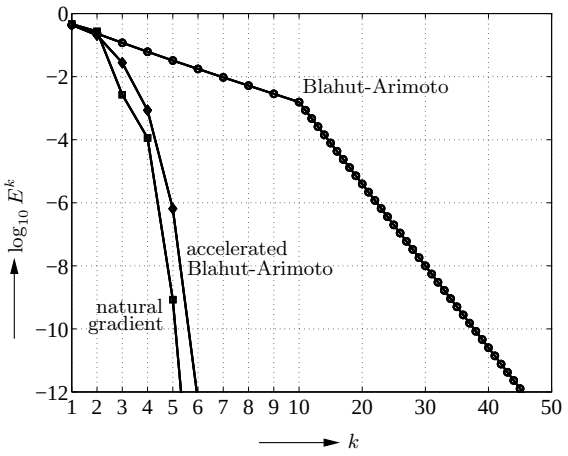

Fig. 1. *Convergence of BA algorithm and accelerated BA and NG algorithms with adaptive step-size (note the nonuniform abscissa scaling).*

randomly picked initial guess $\mathbf{p}^0$. The step-size in our algorithms was chosen in an adaptive fashion as $\mu_k = D(\mathbf{Q}\mathbf{p}^k \| \mathbf{Q}\mathbf{p}^{k-1}) / D(\mathbf{p}^k \| \mathbf{p}^{k-1})$ for $k > 1$ and $\mu_1 = 1$. As performance (and stopping criterion) we used $E^k = \max_j D_j^k - I^k$ since (2) implies $C(\mathbf{Q}) - I^k \leq E^k$. The convergence results for a desired accuracy of 12 decimals are shown in Fig. 1. (all algorithms delivered $C = 0.365148445440$ bit). It is seen that the convergence of the NG and accelerated BA algorithms (6 iterations for the desired accuracy) is significantly faster than that of the BA algorithm (46 iterations). After five iterations, BA yields only one correct decimal while accelerated BA and NG achieve already 6 and 9 correct decimals, respectively. Fig. (1) also clearly verifies that the accelerated BA and NG algorithm with adaptive step-size feature superlinear convergence.

IX. Conclusions

We have proposed improvements on the Blahut-Arimoto (BA) algorithm for computing the capacity of discrete memoryless channels (DMC). An accelerated BA algorithm and a natural gradient (NG) algorithm have been introduced that have the potential for significantly faster (in fact, often superlinear) convergence as compared to the conventional BA algorithm. Recasting the capacity computation problem as an equi-divergence game, intuitive interpretations of all these algorithms have been given via information geometric arguments. We also provided a unifying framework for the (accelerated)

BA and NG algorithm in terms of proximal point methods. This enables the formulation of some statements regarding the convergence of our algorithms.

While our presentation focused on DMCs, our results carry over to the cases of multi-access channels [5], quantum channels [7], ISI channels [3, 4], and channels with side information [6]. In all of these cases, the computational savings achieved using our technique will be even more pronounced.

Furthermore, we conjecture that our approach can be applied to the computation of rate-distortion curves and to portfolio optimization, both of which represent problems closely related to capacity computation [1, 8].

Appendix

Information Geometry Information geometry is concerned with the geometry of the manifold of probability distributions. In this appendix, we briefly summarize the aspects relevant to this paper. Further details can be found in [12, 13, 21].

We focus on the M-dimensional dually flat manifold $\mathcal{P}$ of probability measures $p(x)$ on discrete finite alphabets $x \in \mathcal{X} = \{x_0, \ldots, x_M\}$. This manifold can be viewed as an exponential (log-linear) family,

$$p(x;\boldsymbol{\theta}) = \exp\Big(a(x) + \sum_{i=1}^{M} \theta_i f_i(x) - \psi(\boldsymbol{\theta})\Big),$$

with $a(x) = 0$, $f_i(x) = \delta(x - x_i)$, $\theta_i = \log \frac{p(x_i)}{p(x_0)}$, and the cumulant-generating function $\psi(\boldsymbol{\theta}) = \log\big(1 + \sum_{i=1}^{M} \exp(\theta_i)\big)$. The parameters θ_i are referred to as natural parameters. $\mathcal{P}$ also constitutes a mixture family,

$$p(x;\boldsymbol{\eta}) = b(x) + \sum_{i=1}^{M} \eta_i \, g_i(x),$$

with $b(x) = \delta(x - x_0)$, $g_i(x) = \delta(x - x_i)$, and $\eta_i = p(x_i)$. The parameters η_i are referred to as dual (or, expectation) parameters. Both natural and dual parameters can be used as coordinates for the manifold $\mathcal{P}$.

The KLD between any two distributions $\mathbf{p}, \mathbf{p}' \in \mathcal{P}$ can be shown to equal

$$D(\mathbf{p}\|\mathbf{p}') = \psi(\boldsymbol{\theta}) + \phi(\boldsymbol{\eta}') - \boldsymbol{\theta}^T \boldsymbol{\eta}'. \tag{30}$$

where $\phi(\boldsymbol{\eta})$ is the convex dual of $\psi(\boldsymbol{\theta})$ (in our case $\phi(\boldsymbol{\eta})$ equals the negative entropy of $\mathbf{p}$). Since $D(\mathbf{p}\|\mathbf{p}) = \psi(\boldsymbol{\theta}) + \phi(\boldsymbol{\eta}) - \boldsymbol{\theta}^T \boldsymbol{\eta} = 0$, it follows that the natural and dual parameters are related via the

Legendre transformations

$$\eta_i = \frac{\partial}{\partial \theta_i}\psi(\boldsymbol{\theta}) = \frac{\exp(\theta_i)}{1+\sum_{i'=1}^{M}\exp(\theta_i')}, \tag{31a}$$

$$\theta_i = \frac{\partial}{\partial \eta_i}\phi(\boldsymbol{\eta}) = \log\Big(\frac{\eta_i}{1-\sum_{i'=1}^{M}\eta_i'}\Big). \tag{31b}$$

For infinitesimally close distributions $\mathbf{p}$ and $\mathbf{p}+\mathrm{d}\mathbf{p}$, there is

$$D(\mathbf{p}\|\mathbf{p}+\mathrm{d}\mathbf{p}) = \frac{1}{2}\mathrm{d}\boldsymbol{\eta}^T\mathbf{G}(\boldsymbol{\eta})\mathrm{d}\boldsymbol{\eta} = \frac{1}{2}\mathrm{d}\boldsymbol{\theta}^T\mathbf{G}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta},$$

where $\mathbf{G}(\boldsymbol{\eta})$ and $\mathbf{G}(\boldsymbol{\theta})$ are the coefficient matrices of the Riemannian metric tensor in the $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ coordinates. Here, they equal the (positive definite) Fisher information matrices of the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$, respectively, which are related as $\mathbf{G}(\boldsymbol{\eta})\mathbf{G}(\boldsymbol{\theta}) = \mathbf{I}$ and can be computed as

$$\mathbf{G}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}^T}\frac{\partial}{\partial \boldsymbol{\theta}}\,\psi(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}^T}\boldsymbol{\eta}(\boldsymbol{\theta})\,, \tag{32}$$

$$\mathbf{G}(\boldsymbol{\eta}) = \frac{\partial}{\partial \boldsymbol{\eta}^T}\frac{\partial}{\partial \boldsymbol{\eta}}\,\phi(\boldsymbol{\eta}) = \frac{\partial}{\partial \boldsymbol{\eta}^T}\boldsymbol{\theta}(\boldsymbol{\eta})\,. \tag{33}$$

Using these facts and 31a, the inverse of $\mathbf{G}(\boldsymbol{\eta})$ required for the natural gradient in Section IV is thus straightforwardly obtained as

$$\mathbf{G}^{-1}(\boldsymbol{\eta}) = \mathbf{G}(\boldsymbol{\theta}(\boldsymbol{\eta})) = \frac{\partial}{\partial \boldsymbol{\theta}^T}\boldsymbol{\eta}(\boldsymbol{\theta}) = \mathrm{diag}\{\boldsymbol{\eta}\} - \boldsymbol{\eta}\boldsymbol{\eta}^T\,.$$

References

[1] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, pp. 460–473, 1972.

[2] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, pp. 14–20, 1972.

[3] A. Kavcic, "On the capacity of Markov sources over noisy channels," in *Proc. IEEE GLOBECOM-2001*, San Antonio, TX, Nov. 2001, pp. 2997–3001.

[4] P. O. Vontobel, "A generalized Blahut-Arimoto algorithm," in *Proc. IEEE Int. Symp. Info. Theory*, Yokohama, Japan, june/july 2003, p. 53.

[5] M. Rezaeian and A. Grant, "A generalization of the Arimoto-Blahut algorithm," in *Proc. IEEE ISIT 2004*, Chicago, IL, June/July 2004.

[6] F. Dupuis, W. Yu, and F. M. J. Willems, "Blahut-Arimoto algorithms for computing channel capacity and rate-distortion with side information," in *Proc. IEEE ISIT 2004*, Chicago, IL, June/July 2004.

[7] H. Nagaoka, "Algorithms of Arimoto-Blahut type for computing quantum channel capacity," in *Proc. IEEE ISIT 1998*, Cambridge, MA, Aug. 1998, p. 354.

[8] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplement Issue No. 1*, pp. 205–237, 1984.

[9] S. Amari and S. C. Douglas, "Why natural gradient?" in *Proc. IEEE ICASSP-98*, Seattle, WA, May 1998, pp. 1213–1216.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[11] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[12] S. Amari and H. Nagaoka, *Methods of Information Geometry*. New York: American Mathematical Society and Oxford University Press, 2000.

[13] I. Csiszár and F. Matúš, "Information projections revisited," *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1474–1490, June 2003.

[14] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedure," *Statistics and Decisions*, vol. supplement issue 1, pp. 205–237, 1984.

[15] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, pp. 877–898, 1976.

[16] D. P. Bertsekas, *Convex Optimization Theory*. Athena scientific, 2009.

[17] S. Chrtien and A. O. Hero, "Kullback proximal algorithms for maximum likelihood estimation," *IEEE Transactions on Information Theory*, vol. 46, pp. 1800–1810, 1998.

[18] J. Eckstein, "Nonlinear proximal point algorithms using bregman functions, with applications to convex programming," *MATHEMATICS OF OPERATIONS RESEARCH*, vol. 18, no. 1, pp. 202–226, 1993.

[19] P. J. Green, "On use of the EM algorithm for penelized likelihood estimation," *Journal of the Royal Statistical Society*, 1990.

[20] I. Csiszár, "Information type measures of difference of probability distributions and indirect observations," *Studia Scientiarum Mathematicarum Hungarica*, vol. 2, pp. 299–318, 1967.

[21] S. Amari, "Information geometry of the EM and em algorithms for neural networks," *Neural Networks*, vol. 8, no. 9, pp. 1379–1408, 1996.

B

From Maximum Likelihood to Iterative Decoding

FROM MAXIMUM LIKELIHOOD TO ITERATIVE DECODING

F. Alberge, Z. Naja, P. Duhamel

Laboratoire des Signaux et Systèmes
Univ. Paris-Sud, UMR8506 Orsay, F-91405; CNRS, Gif-sur-Yvette, F-91192;
Supelec, Gif-sur-Yvette, F-91192
e-mail:{alberge, naja, pierre.duhamel}@lss.supelec.fr

ABSTRACT

Iterative decoding is considered in this paper from an optimization point of view. Starting from the optimal maximum likelihood decoding, a (tractable) approximate criterion is derived. The global maximum of the approximate criterion is analyzed: the maximum likelihood solution can be retrieved from the approximate criterion in some particular cases. The classical equations of turbo-decoders can be obtained as an instance of an hybrid Jacobi/Gauss-Seidel implementation of the iterative maximization for the tractable criterion. The extrinsics are a natural consequence of this implementation. In the simulation part, we show a practical application of these results.

Index Terms— Maximum likelihood decoding, iterative turbo-decoding, BICM

1. INTRODUCTION

Bit-Interleaved Coded Modulation (BICM) was first suggested by Zehavi in [1] to improve the Trellis Coded Modulation performance over Rayleigh-fading channels. In BICM, the diversity order is increased by using bit-interleavers instead of symbol interleavers. This improvement is achieved at the expense of a reduced minimum Euclidean distance leading to a degradation over non-fading Gaussian channels [1], [2]. This drawback can be overcome by using iterative decoding (BICM-ID) at the receiver. BICM-ID is known to provide excellent performance for both Gaussian and fading channels.
The iterative decoding scheme used in BICM-ID is very similar to serially concatenated turbo-decoders. Indeed, the serial turbo-decoder makes use of an exchange of information between computationally efficient decoders for each of the component codes. In BICM-ID, the inner decoder is replaced by demapping which is less computationally demanding than a decoding step. Even if this paper focus on iterative decoding for BICM, the results can be applied to the large class of iterative decoders including serial or parallel concatenated turbo-decoders.
The turbo-decoder and more generally iterative decoding was not originally introduced as the solution to an optimization problem rendering the analysis of its convergence and stability very difficult. Among the different attempts to provide an analysis of iterative decoding, the EXIT chart analysis and density evolution have permitted to make significant progress [3] but the results developed within this setting apply only in the case of large block length. Another tool of analysis is the connection of iterative decoding to factor graphs [4] and belief propagation [5]. Convergence results for belief propagation exists but are limited to the case where the corresponding graph is a tree which does not include turbo-code. A link between iterative decoding and classical optimization algorithms has been made also in [6] where the turbo-decoding is interpreted as a nonlinear block Gauss-Seidel iteration for solving a constrained optimization problem. In [7], the turbo-decoding is interpreted in a geometric setting as a dynamical system leading to new but incomplete results. The failure to obtain complete results is mainly due to the inability to efficiently describe extrinsic information passing. The relation between the optimal maximum likelihood decoding and iterative decoding is not yet fully understood.
In this paper, we first review the principle of maximum likelihood decoding. An approximate (and tractable) criterion is derived from an equivalent and convenient formulation of the optimal criterion. We prove that, in specific cases, the global maximum of the approximate criterion yields the maximum likelihood optimum. We then consider the iterative maximization and prove that the choice of a particular scheduling leads to the classical updates used in the iterative (turbo) decoding. In the simulation part, these results are applied to the detection of suspicious solutions *ie* with a possible large number of errors.

2. MAXIMUM-LIKELIHOOD DECODING

A conventional BICM system [2] is built from a serial concatenation of a convolutional encoder, a bit interleaver and an M-ary bits-to-symbol mapping (where $M = 2^m$) as shown in Figure 1. The sequence of information bits $\mathbf{b}$ of length n_b is first encoded by a convolutional encoder to produce the output encoded bit sequence $\mathbf{c}$ of length n which is then scrambled by a bit interleaver (as opposed to the channel symbols in the symbol interleaved coded sequence) operating on bit indexes. Let $\mathbf{d} = \pi(\mathbf{c})$ denote the interleaved sequence. Then, m consecutive bits of $\mathbf{d}$ are grouped as a symbol. The complex transmitted signal s_k, $1 \leq k \leq n/m$, is then chosen from an M-ary constellation ψ where ψ denotes the mapping scheme. For simplicity, we consider transmission over the AWGN channel. The received signals can be written as:

$$y_k = s_k + n_k \quad 1 \leq k \leq \frac{n}{m} \tag{1}$$

where n_k is a complex white Gaussian noise with independent in-phase and quadrature components having two-sided power spectral density σ_c^2. The maximum likelihood sequence detection takes the

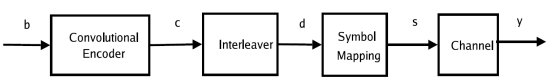

Fig. 1. BICM transmission scheme

form:

$$\hat{\mathbf{b}}_{MLD} = \arg \max_{\mathbf{b}\in\{0,1\}^{n_b}} p(\mathbf{y} \mid \mathbf{b}) \quad (2)$$

where $p(\mathbf{y} \mid \mathbf{b})$ is the likelihood function which results from concatenating the encoder with the channel. Since there is a one-to-one correspondence between the binary message $\mathbf{b}$ and the interleaved sequence $\mathbf{d}$, eq. (2) is equivalent to searching $\hat{\mathbf{d}}_{MLD}$ as:

$$\hat{\mathbf{d}}_{MLD} = \arg \max_{\mathbf{d}\in\{0,1\}^{n}} p_{ch}(\mathbf{y} \mid \mathbf{d}) I_{co}(\mathbf{d}) \quad (3)$$

where $p_{ch}(\mathbf{y} \mid \mathbf{d})$ is the probability of receiving $\mathbf{y}$ when the sequence transmitted through the channel is the mapping of $\mathbf{d}$ and where $I_{co}(\mathbf{d})$ is the indicator function of the code meaning that $I_{co}(\mathbf{d}) = 1$ if $\mathbf{c} = \pi^{-1}(\mathbf{d})$ is a codeword and 0 elsewhere. Another way to tackle this problem consists in finding the prior PMF on $\mathbf{d}$ which maximizes the *a posteriori* probability of having received $\mathbf{y}$

$$\hat{\mathbf{p}}_{MLD}(\mathbf{d}) = \arg \max_{\mathbf{p}\in\mathcal{E}_s} \sum_{\mathbf{d}} \mathbf{I}_{co}(\mathbf{d}) \mathbf{p}_{ch}(\mathbf{y} \mid \mathbf{d}) \mathbf{p}(\mathbf{d}) \quad (4)$$

where $\mathcal{E}_S$ stands for the set of all possible **separable** PMFs on $\mathbf{d}$. A PMF $\mathbf{p}(\mathbf{d})$ is separable if $\mathbf{p}(\mathbf{d}) = \prod_i p_i(d_i)$ with $p_i(d_i)$ the probability for bit i to be equal to d_i. The optimal solution $\hat{\mathbf{p}}_{MLD}(\mathbf{d})$ takes the form

$$\hat{\mathbf{p}}_{MLD}(\mathbf{d}) = \begin{cases} 1, & \mathbf{d} = \hat{\mathbf{d}}_{MLD} \\ 0, & otherwise \end{cases} \quad (5)$$

Any other weighting (with the constraint $\sum_{\mathbf{d}} p(\mathbf{d}) = 1$) produces a lower likelihood. The formulation in (4) is equivalent to the original problem in (3). The practical implementation of this optimal optimization problem is dismissed due to the presence of a random bit interleaver and to the (large) numerical value of n. In the next section, we present a sub-optimal criterion derived from (4) in a constructive way and establish some properties.

3. A SUB-OPTIMAL MAXIMUM LIKELIHOOD DECODING

3.1. Assumptions and approximations

We observed that the optimal maximum likelihood decoding is infeasible due to the interleaver and to the computational complexity involved by the computation and storage of the 2^n taps of the PMF. A solution regarding the interleaver is to consider separately the two-blocks (mapping and coding) in a particular sense to be defined later. The problem of the computational complexity can be handled by working on the bit-marginals rather than on the PMF of the whole sequence. For that purpose we split the variable $\mathbf{p}(\mathbf{d})$ into the product of the two separable PMFs ($\mathbf{l}(\mathbf{d})$ and $\mathbf{q}(\mathbf{d})$) and introduce the computation of the bit-marginals into the optimal criterion as

$$\begin{aligned} &\left(\hat{\mathbf{l}}_{MLD}(\mathbf{d}), \hat{\mathbf{q}}_{MLD}(\mathbf{d})\right) = \\ &\arg \max_{\mathbf{l},\mathbf{q}\in\mathcal{E}_s} \sum_{d_k} \sum_{\mathbf{d}:d_k} \mathbf{I}_{co}(\mathbf{d}) \mathbf{p}_{ch}(\mathbf{y} \mid \mathbf{d}) \mathbf{l}(\mathbf{d}) \mathbf{q}(\mathbf{d}) \end{aligned} \quad (6)$$

The double sum above is exactly the same as the some over all the words $\mathbf{d}$. The global maximum is again obtained for the optimal choice of the weights $\mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d})$. Since $\mathbf{l}(\mathbf{d})$ and $\mathbf{q}(\mathbf{d})$ are PMF, $\sum_{\mathbf{d}} \mathbf{l}(\mathbf{d})\mathbf{q}(\mathbf{d}) \leq 1$, and the optimal weighting strategy is again

$$\hat{\mathbf{l}}_{MLD}(\mathbf{d}) = \hat{\mathbf{q}}_{MLD}(\mathbf{d}) \begin{cases} 1, & \mathbf{d} = \hat{\mathbf{d}}_{MLD} \\ 0, & otherwise \end{cases} \quad (7)$$

The formulation in (6) is then an equivalent form on the original problem since the two solutions $\hat{\mathbf{l}}_{MLD}(\mathbf{d})$ and $\hat{\mathbf{q}}_{MLD}(\mathbf{d})$ both select the optimal sequence $\hat{\mathbf{d}}_{MLD}$ of the maximum likelihood decoding problem. Let $\mathcal{C}_{MLD}$ denote the criterion in (6). The direct maximization of $\mathcal{C}_{MLD}$ is still untractable. We need to separate the coding part from the mapping and channel part, this can done by replacing the bit-marginals of the product of two PMFs by the product of the bit-marginals of the two PMFs taken separately. This is of course an approximation leading to the new criterion $\tilde{\mathcal{C}}_k$ defined as

$$\tilde{\mathcal{C}}_k = \left(\sum_{\mathbf{d_k}} \left(\sum_{\mathbf{d}:d_k} \mathbf{I}_{co}(\mathbf{d}) \mathbf{q}(\mathbf{d})\right) \left(\sum_{\mathbf{d}:\mathbf{d_k}} \mathbf{p_{ch}}(\mathbf{y} \mid \mathbf{d}) \mathbf{l}(\mathbf{d})\right)\right) \quad (8)$$

This approximation deserves some comments. First, the bit-marginals in $\tilde{\mathcal{C}}_k$ are now computable in practice. For example $\sum_{\mathbf{d}:d_k} \mathbf{I}_{co}(\mathbf{d})\mathbf{q}(\mathbf{d})$, $1 \leq k \leq n$, $d_k \in \{0, 1\}$ is exactly the output given by a BCJR [8]. Next, the criterion $\tilde{\mathcal{C}}_k$ is dependant of k: the quantities involved in the criterion are not the same for two different values of k (whereas $\mathcal{C}_{MLD}$ is independent of k). This suggests that criterion $\tilde{\mathcal{C}}_k$ should be used for the **maximization over the k^{th} bit-marginal**. The maximization of the unique criterion $\mathcal{C}_{MLD}$ has been turned into a distributed optimization of the n criteria $\tilde{\mathcal{C}}_k$. Last, the criteria $\mathcal{C}_{MLD}$ and $\tilde{\mathcal{C}}_k$, $1 \leq k \leq n$, are the same (meaning there is no approximation) if the two PMFs involved $\mathbf{I}_{co}(\mathbf{d})\mathbf{q}(\mathbf{d})$ and $\mathbf{p}_{ch}(\mathbf{y} \mid \mathbf{d})\mathbf{l}(\mathbf{d})$) are separable. We can notice that for $\mathbf{l}(\mathbf{d}) = \hat{\mathbf{l}}_{MLD}(\mathbf{d})$ and $\mathbf{q}(\mathbf{d}) = \hat{\mathbf{q}}_{MLD}(\mathbf{d})$ (defined in (7)) the two PMFs are indeed separable. This is also true for all the class of "Kronecker" PMFs in which the global optimum is always lying.
In the next subsection, we focus on the maximization of the sub-optimal criteria $\tilde{\mathcal{C}}_k$ and derive some interesting properties.

3.2. Sub-optimal criterion and global maximum

We prove in the two propositions below that in some special cases, the criteria $\tilde{\mathcal{C}}_k$ yields the same global maxima as the optimal criterion $\mathcal{C}_{MLD}$.

Proposition 1 *The maximum of any criterion $\tilde{\mathcal{C}}_k$, $1 \leq k \leq n$ is obtained for $\mathbf{q} = \hat{\mathbf{q}}$ and $\mathbf{l} = \hat{\mathbf{l}}$ such that*

$$\hat{\mathbf{l}}(\mathbf{d}')\hat{\mathbf{q}}(\mathbf{d}) = \begin{cases} 1, & (\mathbf{d}, \mathbf{d}') = (\hat{\mathbf{d}}, \hat{\mathbf{d}}') \\ 0, & otherwise \end{cases} \quad (9)$$

where $(\hat{\mathbf{d}}, \hat{\mathbf{d}}') = \arg\max_{\mathbf{d},\mathbf{d}'\in\{0,1\}^n} \mathbf{p}_{ch}(\mathbf{y} \mid \mathbf{d}')\mathbf{I}_{co}(\mathbf{d})$.

Proof: Since $\mathbf{q}$ and $\mathbf{l}$ are in $\mathcal{E}_s$, $\tilde{\mathcal{C}}_k$ reads

$$\tilde{\mathcal{C}}_k = \sum_{\mathbf{d_k}} \left(\sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \mathbf{I}_{co}(\mathbf{d}) \mathbf{p}(\mathbf{y} \mid \mathbf{d}') \prod_j q_j(d_j) l_j(d'_j)\right) \quad (10)$$

and $\sum_{d_k} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \prod_j q_j(d_j) l_j(d'_j) = \sum_{d_k} l_k(d'_k) q_k(d_k)$. In other words,

$$\frac{1}{\sum_{d_k} l_k(d'_k) q_k(d_k)} \sum_{d_k} \sum_{\mathbf{d}:d_k} \sum_{\mathbf{d}':d_k} \prod_j q_j(d_j) l_j(d'_j) = 1 \quad (11)$$

Thus $\frac{1}{\sum_{d_k} l_k(d'_k) q_k(d_k)} \mathcal{C}_k$ is maximum for $\hat{\mathbf{q}}(\mathbf{d}')\hat{\mathbf{l}}(\mathbf{d}) = 1$ for $(\mathbf{d}, \mathbf{d}') = (\hat{\mathbf{d}}, \hat{\mathbf{d}}')$ and $\hat{\mathbf{q}}(\mathbf{d})\hat{\mathbf{l}}(\mathbf{d}') = 0$ for the other pairs involved in (10). All the pairs $(\mathbf{d}, \mathbf{d}')$ have the same value for the bit k, this is true in particular for $(\hat{\mathbf{d}}, \hat{\mathbf{d}}')$. Then $\sum_{d_k} l_k(d'_k) q_k(d_k)$ and consequently $\tilde{\mathcal{C}}_k$ are also maximized by $\hat{\mathbf{q}}(\mathbf{d})\hat{\mathbf{l}}(\mathbf{d}')$.

□

Let $\hat{\mathbf{d}}_{ch} = \arg\max_{\mathbf{d}\in\{0,1\}^n} p(\mathbf{y} \mid \mathbf{d})$. Let suppose that $\hat{\mathbf{d}}_{ch}$ is a **codeword**. This is likely to be so at high SNR. Then $(\hat{\mathbf{d}}_{ch}, \hat{\mathbf{d}}_{ch}) = \arg\max_{\mathbf{d},\mathbf{d}'\in\{0,1\}^n} p_{ch}(\mathbf{y} \mid \mathbf{d}')I_{co}(\mathbf{d})$ and also $\hat{\mathbf{d}}_{ch} = \hat{\mathbf{d}}_{MLD} = \arg\max_{\mathbf{d}\in\{0,1\}^n} p_{ch}(\mathbf{y} \mid \mathbf{d})I_{co}(\mathbf{d})$. Then each criterion $\tilde{\mathcal{C}}_k$ has a global maximum at $(\hat{\mathbf{l}}_{MLD}(\mathbf{d}), \hat{\mathbf{q}}_{MLD}(\mathbf{d}))$. We can also remark that the others global maximizers of the individual criterion $\tilde{\mathcal{C}}_k$ are not global maximizers of all the others criteria $\tilde{\mathcal{C}}_i$, with $1 \leq i \leq n$ and $i \neq k$. As a conclusion, if the channel probability $p_{ch}(\mathbf{y} \mid \mathbf{d})$ reaches its maximum for a particular value of $\mathbf{d}$ corresponding to a codeword then the joint global maximization of criteria $\mathcal{C}_k$ for $1 \leq k \leq n$ yields the same solution (given in (7)) than the maximum likelihood decoding. We can now define a new criterion $\tilde{\mathcal{C}} = \sum_{k=1}^{n} \tilde{\mathcal{C}}_k$ which appears to be a relevant approximation of the optimal criterion $\mathcal{C}$.

Proposition 2 *Let suppose that $\tilde{\mathcal{C}}$ has a global maximum at $(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}})$. If $(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}})$ is such that $\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}\hat{\mathbf{q}}_{\tilde{\mathcal{C}}}(\mathbf{d}) = 1$ at $\mathbf{d} = \mathbf{d}_0$ and 0 otherwise then $\mathbf{d}_0 = \hat{\mathbf{d}}_{MLD} = \arg\max_{\mathbf{d}\in\{0,1\}^n} p_{ch}(\mathbf{y} \mid \mathbf{d})I_{co}(\mathbf{d})$.*

Proof: $\tilde{\mathcal{C}}$ has a global maximum at $(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}})$ then $\tilde{\mathcal{C}}(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}}) \geq \tilde{\mathcal{C}}(\mathbf{l}, \mathbf{q})$ for any PMF $\mathbf{l}$, $\mathbf{q}$. From the definition of $\tilde{\mathcal{C}}$ we have $\tilde{\mathcal{C}}(\mathbf{l}, \mathbf{q}) \geq n\mathcal{C}_{MLD}(\mathbf{l}, \mathbf{q})$. Moreover $\tilde{\mathcal{C}}(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}}) = n\mathcal{C}_{MLD}(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}})$. Thus $\mathcal{C}_{MLD}(\hat{\mathbf{l}}_{\tilde{\mathcal{C}}}, \hat{\mathbf{q}}_{\tilde{\mathcal{C}}}) \geq \mathcal{C}_{MLD}(\mathbf{l}, \mathbf{q})$ for any PMF $\mathbf{l}$, $\mathbf{q}$.

□

If we manage to find the global maximum of the sub-optimal criterion $\tilde{\mathcal{C}}$ and if the corresponding argument turns out to be a Kronecker PMF then this is also the argument of a global maximum of the optimal criterion $\mathcal{C}_{MLD}$ associated with the maximum likelihood decoding. The practical usefulness of criterion $\tilde{\mathcal{C}}$ will be emphasized in section 4. In the next subsection, we build an iterative strategy of maximization.

3.3. Iterative maximization

We observed in section 3.2 that the sub-optimal criterion $\tilde{\mathcal{C}}_k$ was derived from $\mathcal{C}_{MLD}$ when dealing with the k^{th} bit-marginal. We propose here to consider a distributed maximization strategy where $l_k(d_k)$ and $q_k(d_k)$ are chosen in order to maximize $\tilde{\mathcal{C}}_k$ as

$$\left(\hat{l}_k, \hat{q}_k\right) = \arg \max_{l_k, q_k \in \mathcal{F}} \tilde{\mathcal{C}}_k \tag{12}$$

where $\mathcal{F}$ is the set of all possible PMFs on d_k. The solution of (12) is given by

$$\begin{array}{c} \hat{l}_k(d_k)\hat{q}_k(d_k) = 1 \quad if \\ f_{d_k}(\mathbf{q}, I_{co}) f_{d_k}(\mathbf{l}, p_{ch}(\mathbf{y} \mid \mathbf{d})) > f_{\overline{d}_k}(\mathbf{q}, I_{co}) f_{\overline{d}_k}(\mathbf{l}, p_{ch}(\mathbf{y} \mid \mathbf{d})) \\ \hat{l}_k(d_k)\hat{q}_k(d_k) = 0 \quad otherwise \end{array} \tag{13}$$

where $f_{d_k}(\mathbf{q}, I_{co}) = \sum_{\mathbf{d}:d_k} \mathbf{I}_{co}(\mathbf{d}) \prod_{j\neq k} q_j(d_j)$, $f_{d_k}(\mathbf{l}, p_{ch}(\mathbf{y} \mid \mathbf{d})) = \sum_{\mathbf{d}:d_k} \mathbf{p}_{ch}(\mathbf{y} \mid \mathbf{d}) \prod_{j\neq k} l_j(d_j)$ and $\overline{d}_k = 1 - d_k$. An iterative process propagating hard estimates (0 or 1) is likely to get stuck in a local minima. A classical solution is to propagate instead soft-estimates (in $[0; 1]$) and take hard decisions at the end of the iterative process. For the maximization problem in (12), possible soft estimates are :

$$\begin{array}{l} \hat{l}_k(d_k)\hat{q}_k(d_k) \propto f_{d_k}(\mathbf{q}, I_{co}) f_{d_k}(\mathbf{l}, p_{ch}(\mathbf{y} \mid \mathbf{d})) \\ \hat{l}_k(\overline{d}_k)\hat{q}_k(\overline{d}_k) \propto f_{\overline{d}_k}(\mathbf{q}, I_{co}) f_{\overline{d}_k}(\mathbf{l}, p_{ch}(\mathbf{y} \mid \mathbf{d})) \end{array} \tag{14}$$

Equation (14) characterizes the product $\hat{l}_k\hat{q}_k$. The individual values of $\hat{l}_k$ and $\hat{q}_k$ depend on the scheduling of the successive updates. Since we have no prior information, a natural choice for the initialization is $l_k^{(0)}(d_k) = q_k^{(0)}(d_k) = \frac{1}{2}$ for $1 \leq k \leq n$ and $d_k \in \{0; 1\}$. Let consider first the update of variables q_k. Following a Jacobi implementation, $q_k^{(it)}$ is obtained from (14) by setting $l_k(d_k) = l_k^{(it-1)}(d_k)$ for $1 \leq k \leq n$ and $q_i(d_i) = q_i^{(it-1)}(d_i)$ for $i \neq k$. In particular, the update at the first iteration is:

$$q_k^{(1)}(d_k) \propto \sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d}) \sum_{\mathbf{d}:d_k} p_{ch}(\mathbf{y} \mid \mathbf{d}) \tag{15}$$

for $d_k \in \{0; 1\}$ and for all $k \in \{1, ..., n\}$. The update of $\mathbf{l}_k$ also comes from (14) by setting $l_i(d_i) = l_i^{(it-1)}(d_i)$ for $i \neq k$ and $q_k(d_k) = q_k^{(it)}(d_k)$ for $1 \leq k \leq n$ since $q_k^{(it)}(d_k)$ has just been computed and is available. This is an hybrid Jacobi/Gauss-Seidel implementation. At the first iteration, the update for l_k is:

$$l_k^{(1)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d}) \prod_{j\neq k} q_j^{(1)}(d_j)}{\sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d})} \tag{16}$$

The generalization to iteration (it) reads:

$$\begin{array}{l} q_k^{(it)}(d_k) \propto \sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d}) \sum_{\mathbf{d}:d_k} p_{ch}(\mathbf{y} \mid \mathbf{d}) \prod_{j\neq k} l_j^{(it-1)}(d_j) \\ l_k^{(it)}(d_k) \propto \frac{\sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d}) \prod_{j\neq k} q_j^{(it)}(d_j)}{\sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d})} \end{array} \tag{17}$$

for $d_k \in \{0; 1\}$ and for all $k \in \{1, ..., n\}$. In general, for convolutive codes, $\sum_{\mathbf{d}:d_k} I_{co} = \sum_{\mathbf{d}:\overline{d}_k} I_{co}$. The iterative updates are then obtained through:

$$\begin{array}{l} q_k^{(it)}(d_k) \propto \sum_{\mathbf{d}:d_k} p_{ch}(\mathbf{y} \mid \mathbf{d}) \prod_{j\neq k} l_j^{(it-1)}(d_j) \\ l_k^{(it)}(d_k) \propto \sum_{\mathbf{d}:d_k} I_{co}(\mathbf{d}) \prod_{j\neq k} q_j^{(it)}(d_j) \end{array} \tag{18}$$

This is exactly the equations that are used in the iterative decoding of BICM where $l_k(d_k)$, $q_k(d_k)$ are usually called *extrinsics* and $l_k(d_k)q_k(d_k)$ (after normalization) is the *APP* (*A Posteriori Probability*) [9]. The extrinsics are a direct consequence of the maximization of a well-defined criterion and of the choice of a particular scheduling in the computation of the successive updates. The algorithm stops in general when an agreement is reached between the APP computed at the demapper and the APP computed at the decoder. This corresponds with our notations to:

$$q_k^{(it)}(d_k) l_k^{(it-1)}(d_k) \propto q_k^{(it)}(d_k) l_k^{(it)}(d_k) \tag{19}$$

for $d_k \in \{0; 1\}$ and for all $k \in \{1, ..., n\}$. This means that the tentative maximization (via soft estimates) of $\tilde{\mathcal{C}}_k$ for $1 \leq k \leq n$ with respect either to $l_k(d_k)$ or to $q_k(d_k)$ produces exactly the same result regarding the product (APP) $l_k(d_k)q_k(d_k)$. The derivation in this paper are mainly based on the block structure (code, mapping) of the BICM and not on the specificity of each block. The line of arguments followed in this paper and the conclusions apply to a wide range of problems including iterative turbo-decoding. For instance, the serial turbo-decoder is obtained from BICM by replacing the PMF $p_{ch}(\mathbf{y} \mid \mathbf{d})$ with $p_{ch}(\mathbf{y} \mid \mathbf{d})I_{co1}(\mathbf{d})$ where $I_{co1}(\mathbf{d})$ is the indicator function of the inner coder involved at the transmitter. Since, no assumptions has been made on $p_{ch}(\mathbf{y} \mid \mathbf{d})$ within this paper, the whole conclusions apply to serial turbo-decoders as well as to any decoder with a structure similar to BICM.

4. SIMULATIONS

The criterion $\tilde{\mathcal{C}} = \sum_{k=1}^{n} \tilde{\mathcal{C}}_k$ is an approximation of the MLD criterion $\mathcal{C}$. The solution for the approximate criterion $\tilde{\mathcal{C}}$ is expected to be close to the MLD solution and hence to achieve low bit error rate (BER). Iterative turbo-decoding is known to provide excellent performance in terms of BER. We can however observe at low SNR a degradation in the performance mainly due to inaccurate solutions of the iterative decoder for some isolated frames. In the following, we use $\tilde{\mathcal{C}}$ has a detector to separate the acceptable solutions (few errors expected) from the suspicious ones (many more errors expected).
We used a classical transmitter BICM scheme with a $(5, 7)$ convolutional code of rate $1/2$. The number of information bits is $n_b = 400$ (a frame). The code bits were passed through a random interleaver and modulated to 16-QAM symbols. The signal to noise ratio is defined as $\frac{E_b}{N_0}$, where E_b denotes the energy per information bit and N_0 is the noise variance. We consider an environment with varying $\frac{E_b}{N_0}$ in the range $\{4dB, 5dB, ..., 11dB, 12dB\}$. The values are chosen randomly from a uniform distribution. The iterative decoding is performed using eq. (18). For each frame, the iterative process is run until an agreement is reached for all the bits (eq. 19) or the maximum number of iterations is reached. The solutions are qualified of acceptable if $\sum_{k=1}^{n} \log(\tilde{\mathcal{C}}_k)$ is greater (at the end of the iterative process) than a certain threshold (to be chosen) and are qualified of suspicious if $\sum_{k=1}^{n} \log(\tilde{\mathcal{C}}_k)$ is under the threshold. We compute BER_a resp. BER_s the bit error rates of the acceptable solutions resp. the BER of suspicious solutions. We expect BER_s to be many more large than BER_a. We also evaluate the proportion of suspicious solutions identified (p_s) and the false alarm proportion ($p_{s,false}$) which counts the solutions wrongly qualified of suspicious. We consider that the solutions with less than 6 errors (among the 400 bits) should not have been qualified of suspicious. The algorithm stops when the total number of errors among the acceptable sequences is greater than 200. The results are reported in table 1.

Threshold	-20	-10	-5
BER_a	$8,78.10^{-4}$	$4,68.10^{-4}$	$2,08.10^{-4}$
BER_s	$0,205$	$0,13$	$9,28.10^{-2}$
p_s %	$6,4\%$	$10,8\%$	$14,8\%$
$p_{false,s}$ %	$2,5\%$	$36,4\%$	$53,83\%$

Table 1. Evaluation of Acceptable/Suspicious solutions

The validity of $\tilde{\mathcal{C}}$ as a relevant approximation of $\mathcal{C}$ is confirmed by this simulation. The BER is strongly correlated with the value of $\tilde{\mathcal{C}}$ reached at the end of the iterative process. If the value is close to the global maximum, the solution is also close (or equal) to the optimal MLD solution. At the opposite, a low value of $\tilde{\mathcal{C}}$ is often associated with a large number of errors. This is also observable in fig. (2) where the distribution of the errors (histograms) are plotted for the acceptable solutions and compared to the histogram obtained for the suspicious solutions in the case where the threshold is -20. We can conclude from this experiment that the suspicious solutions are mainly due to local minima ($\tilde{\mathcal{C}}$ is often very far from the threshold) rather than to an inaccuracy of $\tilde{\mathcal{C}}$ as an approximate criterion for $\mathcal{C}$. For a practical point of view, the simulation above illustrates how we can guarantee a given performance (in terms of BER) independently of the SNR (at the cost of a rejection/re-emission of the suspicious frames).

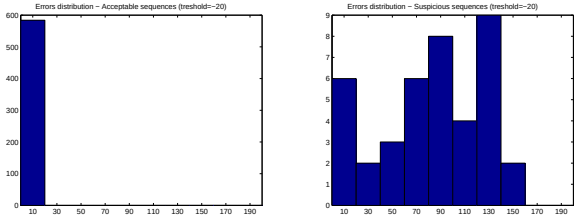

Fig. 2. Histogram representation of the error distribution for : (left) acceptable solutions - (right) suspicious solutions

5. CONCLUSION

In this paper, we derived the (turbo) iterative decoding from maximum likelihood decoding. The approximation needed to obtain a tractable solution are clearly stated and some proofs have been established concerning the global maximum of the approximate criterion. It has also be shown that the turbo-decoding follows from an hybrid Jacobian/Gauss-Seidel implementation of the maximization process. The propagation of extrinsics is naturally introduced and is a direct consequence of the scheduling. In the simulation part, a possible application of these results has been presented. In this paper, we proved that iterative turbo-decoding can be interpreted as a distributed optimization strategy reminiscent of mixed cooperative/non cooperative games. The theoretical tools developed within the game theory framework constitute a new open perspective for completing the analysis of turbo-like decoding.

6. REFERENCES

[1] E. Zehavi, "8-PSK trellis codes for a Rayleigh fading channel," *IEEE Trans. on Commun.*, vol. 40, pp. 873–883, May 1992.

[2] G. Caire, G.Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. on Inform. Theory*, vol. 4, pp. 927–946, May 1998.

[3] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. on Commun.*, vol. 49, pp. 1727–1737, Oct 2001.

[4] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*, San Francisco, CA: Morgan Kaufmann, 1988.

[6] J. M. Walsh, P.A. Regalia, and C. R. Johnson, "Turbo decoding as Iterative Constrained Maximum-Likelihood Sequence Detection," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 5426–5437, Dec. 2006.

[7] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. on Inform. Theory*, vol. 46, no. 1, pp. 9–23, 2000.

[8] L.R. Bahl, J. Cocke, F. Jelinek, and J.Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Inform. Theory*, pp. 284–287, March 1974.

[9] F. Alberge, "Iterative decoding as Dykstra's algorithm with alternate I-projection and reverse I-projection," in *EUSIPCO Proc.*, Lausanne, Switzerland, August 2008.

C

Liste de publications

PRODUCTIONS SCIENTIFIQUES

Articles de Revue:

G. MATZ and **Z. NAJA** and F. ALBERGE and P. DUHAMEL
"Accelerating the Blahut-Arimoto-Algorithm via Information Geometry", submitted to IEEE transaction on information theory, February, 2011

C. DELPHA and A. ZAIDI and **Z. NAJA** and R. BOYER and P. DUHAMEL
"A quantization based Robust Audio Watermarking Scheme using a perceptual model", submitted to Elsevier journal on Digital Signal Processing, September, 2010

Conférences nationales et internationales avec comité de lecture :

F. ALBERGE and **Z. NAJA** and P. DUHAMEL
"From Maximum Likelihood to Iterative Decoding", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, Czech Republic, 22-27 May, 2011

S. SCHWANDTER and **Z. NAJA** and P. DUHAMEL and G. MATZ
"Complexity Reduction in BICM-ID Systems Through Selective Log-Likelihood Ratio Updates", 11th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Marrakech, Morocco, 20-23 June , 2010

Z. NAJA and F. ALBERGE and P. DUHAMEL
"Geometrical interpretation and improvements of the Blahut-Arimoto's algorithm", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 19-24 April, 2009

F. ALBERGE and **Z. NAJA** and P. DUHAMEL
"New Criteria for Iterative Decoding", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 19-24 April, 2009

Z. NAJA and F. ALBERGE and P. DUHAMEL
"Méthode du point proximal: principe et applications aux algorithmes itératifs", 22e colloque GRETSI sur le traitement du signal et des images, Dijon , France, 8-11 Septembre, 2009.

Autres Publications:

Z. NAJA and F. ALBERGE and P.DUHAMEL and L. SZCZECINSKI
"Bit-Interleaved Coded Modulation with Iterative Decoding/Demapping (BICM-ID)", Tutorial in the NEWCOM++ 2010 Winter School on Iterative Techniques in Wireless Communications, Department of Communication Technology, Aalborg University, Aalborg, Denmark, 24-26 February 2010

Chapitre 4, délivrable 1, WPR4, NEWCOM++ : « Theoritical Framework for Iterative Processing» (http://www.newcom-project.eu:8080/Plone/public-deliverables/research/DR4.1-final-1.pdf/view)

Participation aux chapitres 3 et 4 du délivrable 2, WPR4, NEWCOM++ : «Non-binary information combining - Towards a theory of mismached and fixed point decoding - Performance and Robustness of turbo synchronization methods »
(http://www.newcom-project.eu:8080/Plone/public-deliverables/research/DR.4.2_draft.pdf/view)

Contribution aux chapitres 4 et 5 du délivrable final du projet NEWCOM++: « Final Project Report »
(C:\Publications\Newcom++\DR4.3_final.pdf)

Mémoires:

Z. NAJA, "Evaluation d'algorithmes de tatouage robuste", Mémoire de Master 2, Université de Bretagne Occidentale (UBO), Juin 2007

Z. NAJA, "Evaluation d'algorithmes de tatouage robuste", Mémoire de projet de fin d'études, Université Libanaise Faculté de Génie Branche 1, Juillet 2007 (dans le cadre de double diplôme)

Workshops et séminaires:

- Présentation : "Geometrical interpretation of iterative algorithms", First NEWCOM++ Meeting, Munich, Allemagne, Février, 2008
- Présentation: "Bit-Interleaved Coded Modulation with Iterative Decoding: interpretation and comments", Second NEWCOM++ Meeting, Istanbul, Turquie, Janvier, 2009
- Présentation: "Advances in understanding the iterative decoding of Bit-Interleaved Coded Modulation (BICM-ID)", Third NEWCOM++ Meeting, Poznan, Pologne, Décembre, 2009
- Présentation: "Interprétation et Amélioration d'une procédure de démodulation itérative", séminaire Telecom L2S, Novembre, 2007
- Présentation : "Interprétation géométrique des algorithmes itératifs ", séminaire Telecom L2S, Juin, 2008
- Participation au fifth IEEE Workshop on Advanced Information Processing for Wireless Communication Systems, Nokia, Copenhague, Danmark, 22-23 Avril, 2010
- Présentation: "Information geometry and its applications on iterative decoding", mini workshop, Université d'Aalborg, Aalborg, Danmark, 21-23 Juin, 2010