

Un Décodeur à Inversion Dynamique pour les Codes Polaires

Ludovic CHANDESIRIS^{1,2}, Valentin SAVIN¹, David DECLERCQ²

¹Laboratoire CEA-Leti
68B Avenue des Martyrs, 38000 Grenoble

²Laboratoire ETIS /UCP
2 avenue Adolphe Chauvin 95032 Cergy-Pontoise Cedex
Ludovic.Chandesris@cea.fr, Valentin.Savin@cea.fr
declercq@ensea.fr

Résumé – Un nouvel algorithme de décodage à inversion, appelé D-SCFlip, est proposé pour améliorer les performances des codes polaires en longueur finie. Articulé autour d’une sélection dynamique des nouvelles tentatives et d’un ordonnancement basé sur une métrique spécifiquement optimisée pour le décodeur, les simulations montrent que ce décodeur est une alternative sérieuse à l’algorithme de décodage par liste, offrant par ailleurs une complexité très proche de celle du décodage par annulation successive.

Abstract – A new decoding algorithm for polar codes, called dynamic SCFlip (D-SCFlip), is proposed for short and moderate code-lengths. Based on a dynamic updating rule and an accurately designed metric for selecting and ordering new attempts, simulations show that this decoder is an interesting alternative to the list-decoder of polar codes, while keeping a computational complexity close to the one of the successive cancellation (SC) decoder.

1 Introduction

Les codes polaires constituent une toute récente famille de codes correcteurs d’erreurs, dont la découverte en 2009 par E. Arikan [1] a apporté la première solution constructive d’un décodeur atteignant asymptotiquement la capacité de Shannon pour tout canal binaire discret sans mémoire – propriété qu’ils ne partagent à ce jour qu’avec les Spatially-coupled LDPC. Leur construction repose sur une structure récursive permettant de générer N sous-canaux virtuels à partir de N réalisations indépendantes du canal de propagation, et permettant un décodage séquentiel dit par "annulation successive" (SC pour successive cancellation) de complexité loglinéaire. Les performances asymptotiques de ce décodeur résultent du fait qu’une partie des sous-canaux virtuels deviennent parfaits, tandis que les autres deviennent totalement bruités, phénomène connu sous le nom de "phénomène de polarisation". Néanmoins, pour des codes relativement courts, les performances du décodeur SC restent malgré tout nettement insuffisantes pour concurrencer les codes LDPC et Turbo, qui se sont imposés dans la majorité des standards de télécommunication, vidéos (Wimax, DVB) et téléphonie (UMTS, LTE). Développer des algorithmes de décodage plus compétitifs a donc constitué – et reste – un aspect primordial de la recherche sur les codes polaires.

L’algorithme actuel offrant les meilleures performances pour le décodage des codes polaires en longueur finie est connu sous le nom de "décodage par liste" (SCL) [2] et approche au plus près de la borne Maximum Likelihood (ML), toutefois assez faible dans le cas des codes polaires. Mieux encore, le plein

potentiel de ce décodeur peut être obtenu en concaténant avec un code externe détecteur d’erreur (ex. CRC) [2] afin d’identifier plus précisément le bon mot de code dans la liste, ce qui a permis d’égaliser voire surpasser les standards LDPC et consorts, au prix cependant d’une complexité nettement accrue et augmentant proportionnellement avec la taille de la liste.

Le décodeur à inversion SCFlip [3] (pour "Successive Cancellation Flip") est une alternative au SCL s’efforçant de limiter la complexité, en remplaçant l’exploration parallèle des chemins par une exploration en série conditionnée par l’échec des précédentes tentatives, et où le succès ou l’échec d’une tentative est déterminé par un code détecteur (CRC). La première tentative est identique au décodage SC, tandis que les – éventuelles – tentatives suivantes sont obtenues en inversant la décision la moins fiable et non encore explorée du SC. Néanmoins, il s’avère que ce décodeur souffre d’une saturation restreignant ses performances, due à la stratégie d’exploration écartant prématurément des tentatives prometteuses, et qui plus est, que la métrique utilisée est sous-optimale pour ordonner les tentatives par probabilité de succès décroissante. Cet article prolonge nos travaux de [5] en combinant la métrique optimisée avec une nouvelle stratégie d’exploration dite "dynamique".

2 Codes Polaires et Décodage par Annulation Successive

Un code polaire est caractérisé par le triplet (N, K, T) où $N = 2^n$ est la longueur du code, K le nombre de bits d’in-

formation (rendement $R = K/N$), tandis que l'ensemble \mathcal{I} indique la position des bits d'information dans le vecteur de longueur N , les bits restants étant qualifiés de "bits figés" pour lesquels la valeur est fixée au préalable et connue par l'encodeur et le décodeur. En appelant $\mathbf{U} = u_1^N$ le vecteur des données, utiles (index $i \in \mathcal{I}$) ou figées (index $i \notin \mathcal{I}$), le vecteur encodé \mathbf{X} est défini par :

$$\mathbf{X} = \mathbf{U} \cdot \mathbf{G}_N$$

où \mathbf{G}_N est définie dans [1]. On définit également la séquence reçue du canal \mathbf{Y} , qui est aussi l'entrée du décodeur, et le vecteur décodé $\hat{\mathbf{U}} = \hat{u}_1^N$, où \hat{u}_i correspond à l'estimation du bit u_i .

Dans le décodage SC, chaque bit d'information $u_i, i \in \mathcal{I}$ est décodé successivement via le calcul de son logarithme de vraisemblance (LLR pour log-likelihood ratio), dépendant de la séquence reçue du canal \mathbf{Y} ainsi que des estimations dures des bits décodés précédemment \hat{u}_1^{i-1} :

$$L_i = \log \left(\frac{\Pr(u_i = 0 | \mathbf{Y}, \hat{u}_1^{i-1})}{\Pr(u_i = 1 | \mathbf{Y}, \hat{u}_1^{i-1})} \right) \quad (1)$$

l'estimation dure \hat{u}_i du bit u_i est alors donnée par le signe du LLR :

$$\hat{u}_i = h(L_i) \stackrel{\text{def}}{=} \begin{cases} u_i & \text{if } i \notin \mathcal{I} \\ \frac{1 - \text{sign}(L_i)}{2} & \text{if } i \in \mathcal{I} \end{cases} \quad (2)$$

où par convention $\text{sign}(0) = \pm 1$ avec équiprobabilité.

Il peut être utile pour la suite de noter que la réutilisation des décisions précédentes est à l'origine d'un phénomène d'erreur en cascade dans le décodage SC : dès lors qu'une décision prise est erronée, celle-ci se propage et a tendance à rendre la suite du décodage particulièrement inefficace, avec de nombreuses nouvelles erreurs.

3 Décodage à Inversion Dynamique

3.1 Généralités

Fondamentalement, le principe du décodage à inversion est d'utiliser un code détecteur (ex. CRC) externe pour valider le succès ou au contraire consigner l'échec d'un décodage. Dès lors, en cas d'échec avéré, une nouvelle tentative est effectuée et ce jusqu'à ce qu'aucune erreur ne soit détectée, ou qu'un nombre maximal de tentatives autorisées n'ait été atteint.

La subtilité du décodage à inversion repose sur la manière de "choisir" chaque nouvelle tentative de sorte à s'assurer d'explorer systématiquement celle maximisant la probabilité de succès parmi toutes celles non effectuées. Il est clair qu'étudier l'ensemble des tentatives disponibles est impraticable, et il devient donc nécessaire de (a) définir une méthode d'exploration permettant de se concentrer sur un nombre limité de candidats pertinents (b) évaluer précisément pour chacun sa probabilité de succès, afin d'en déterminer l'ordre dans lequel les tentatives doivent être effectuées.

Le décodeur dont il est question dans ce papier s'inspire de [3], mais bénéficie de deux améliorations complémentaires

portant respectivement sur les aspects (a) et (b). La première a pour but d'élargir la liste des candidats pertinents considérés, en tenant compte non seulement des résultats du SC mais aussi de toutes les nouvelles tentatives effectuées, et minimisant ainsi tout risque d'écarter arbitrairement un bon candidat. La seconde vise à mieux évaluer la probabilité de succès de chaque candidat au moyen d'une métrique soigneusement optimisée pour le décodeur utilisé.

3.2 Exploration par bit-flips

Dans un décodeur à inversion, chaque tentative est décrite par rapport au décodage SC standard (section 2) à l'aide d'un "bit-flip", *i.e.* une séquence de positions dont les décisions sont inversées.

Definition 1 *Un bit-flip d'ordre ω ($0 \leq \omega \leq K + r$) est un ensemble de ω index $\mathcal{E} = \{i_1, i_2 \dots i_\omega\} \subset \mathcal{I}$ et tels que $i_1 < i_2 < \dots < i_\omega$. Le décodage associé, noté $SC[\mathcal{E}]$ est similaire à un décodage SC, mais dans lequel la fonction de prise de décision dure h (Eq. 2) est remplacée par $h_{\mathcal{E}}$:*

$$\forall i \in \mathcal{I}, \quad \hat{u}[\mathcal{E}]_i = h_{\mathcal{E}}(L_i) \stackrel{\text{def}}{=} \begin{cases} h(L_i) & \text{if } i \notin \mathcal{E} \\ 1 - h(L_i) & \text{if } i \in \mathcal{E} \end{cases} \quad (3)$$

Naturellement, pour tout message reçu \mathbf{Y} , il existe un unique bit-flip assurant un décodage correct. A noter que $SC[\emptyset]$ est le décodage SC standard. Dans [3], seuls les bit-flips d'ordre $\omega = 1$ (pas plus d'une inversion par rapport au SC) sont considérés. Néanmoins, et les simulations de la partie 4 l'illustreront, cette restriction radicale impose une saturation des performances préjudiciable, du fait de l'écroulement prématuré des bit-flips d'ordres supérieurs ($\omega > 1$).

Dans [5], nous avons envisagés une approche où les bit-flips étaient testés par ordre ($\omega \leq \omega_{max}$) croissants, offrant un gain notable par rapport à [3], mais conservant plusieurs inconvénients parmi lesquels le choix délicat du nombre de tentatives pour chaque valeurs de ω , le fait que les performances restaient saturées à cause de la restriction $\omega \leq \omega_{max}$ (d'où la difficulté à concurrencer le SCL) et finalement, une complexité accrue pour décoder lorsque le bit-flip correct est d'ordre élevé.

La solution proposée ici, appelée D-SCFlip, surpasse ces précédents algorithmes en se basant sur un processus de sélection dynamique exploitant toutes les informations glanées au cours de l'ensemble des tentatives effectuées et où l'ordre des tentatives effectuées n'est dicté que par la métrique. Pour ce faire, le décodeur est muni d'une liste de bit-flips appelée $\mathcal{L}_{\text{flip}}$, parcourue dans l'ordre et caractérisant chaque nouvelle tentative. Initialement, cette liste est vide, mais se remplit au fur et à mesure des nouvelles tentatives effectuées. Après chaque tentative, cette liste est mise à jour dynamiquement en y intercalant les bit-flips relatifs à cette tentative. Par "relatifs à une tentative", on entend que, après tout décodage $SC[\mathcal{E}]$ d'un bit-flip $\mathcal{E} = \{i_1, \dots i_\omega\}$ d'ordre ω , la probabilité de succès de chaque bit-flip $\mathcal{E}' = \{\mathcal{E}, k\}, k \in \mathcal{I}, k > i_\omega$ d'ordre $\omega + 1$ est évaluée et chacun est alors incorporé à la liste $\mathcal{L}_{\text{flip}}$ de l'ensemble

des bit-flips déjà évalués, positionné selon la valeur de sa métrique, et est susceptible d'être testé dans les tentatives à venir. On parlera d'exploration incrémentale en ω .

En pratique, on se fixe un nombre maximal de nouvelles tentatives durant le décodage, noté T , et qui peut donc être utilisé pour définir la taille de la liste $\mathcal{L}_{\text{flip}}$: si un bit-flip a une probabilité de succès inférieure à celles de tous les bit-flips déjà présents dans la liste, celui-ci sera tout bonnement écarté. Reste maintenant à développer une métrique fonctionnant pour tout ordre ω et permettant d'évaluer précisément la probabilité de succès, afin de pouvoir déterminer l'ordre dans lequel les tentatives doivent être réalisées.

3.3 Probabilité de succès d'un bit-flip

Dans un décodeur à inversion, et plus encore dans le processus de sélection dynamique proposé, la métrique utilisée joue un rôle crucial pour garantir de bonnes performances. D'une part parce que toute tentative non fructueuse est synonyme d'accroissement de latence et de complexité. D'autre part, parce que dans le cas où le bit-flip permettant de corriger le message reçu \mathbf{Y} est d'ordre $\omega_{\mathbf{Y}} > 1$, l'exploration incrémentale en ω impose d'identifier et tester toute la série de bit-flips d'ordres $\omega' = 1, \dots, \omega_{\mathbf{Y}}$.

Il est important de comprendre que la probabilité de succès d'un bit-flip est corrélée avec la probabilité que celui-ci corrige la toute première erreur de la trajectoire. En effet, inverser une décision située après la première erreur n'a aucune chance de corriger le message reçu et qui plus est contraint le décodeur à considérer d'autres bit-flips qui n'ont pas plus de chance de réussir. A ce titre, la métrique proposée dans [3], basée directement sur la valeur absolue du LLR courant, est sous-optimale, car elle permet seulement d'identifier une position dont la probabilité d'être en erreur est la plus élevée. Plus généralement, identifier le bon bit-flip d'ordre $\omega > 1$ revient, du fait de l'exploration incrémentale, à trouver le bit-flip $\{i_0\}$ d'ordre $\omega = 1$ correspondant à la première erreur du SC, puis le bit-flip $\{i_0, i_1\}$ d'ordre $\omega = 2$ où i_1 est la première erreur dans la trajectoire du SC $\{\{i_0\}\}$ et ainsi de suite.

Dès lors, pour tout bit-flip $\mathcal{E} = \mathcal{E}_- \cup \{j\}$, avec $\max(\mathcal{E}_-) < j$, la métrique du bit-flip \mathcal{E} se calcule d'après :

$$M_{\alpha}(\mathcal{E}) = M_{\alpha}(\mathcal{E}_-) \cdot M_{\alpha}(\{j\} | \text{SC}[\mathcal{E}_-]), \quad (4)$$

où $M_{\alpha}(\{j\} | \text{SC}[\mathcal{E}_-])$ est la probabilité que la position j soit la première erreur sur la trajectoire SC $[\mathcal{E}_-]$, et donnée par :

$$M_{\alpha}(\{j\} | \text{SC}[\mathcal{E}_-]) = \frac{1}{1 + \exp(\alpha \cdot |\text{L}[\mathcal{E}_-]_j|)} \prod_{\substack{k=1+\max(\mathcal{E}_-) \\ k \in \mathcal{I}}}^{j-1} \frac{1}{1 + \exp(-\alpha \cdot |\text{L}[\mathcal{E}_-]_k|)}, \quad (5)$$

avec α un paramètre à optimiser, et par convention $M_{\alpha}(\emptyset) = 1$.

Pour commencer, il apparait que, quelle que soit la valeur de α , $M_{\alpha}(\mathcal{E}) < M_{\alpha}(\mathcal{E}_-)$. Cette propriété d'*imbrication* (*nesting property* en anglais), associée au processus de mise-à-jour

dynamique, garantit que lorsque le décodeur procède à une tentative donnée dans la liste $\mathcal{L}_{\text{flip}}$, alors toutes les tentatives dont la métrique est plus élevée sont nécessairement positionnées avant, et ont donc *de facto* été déjà testées. Autrement dit, dans le D-SCFlip, les t tentatives effectuées correspondent exactement aux t bit-flips ayant la métrique la plus élevée (parmi les 2^{K+r} bit-flips existants).

L'optimisation du paramètre α est également essentielle pour garantir les meilleures performances de décodage. Il est possible de montrer que la métrique dans [3] revient à choisir $\alpha = +\infty$ pour les bit-flips d'ordre $\omega = 1$, ce qui comme expliqué précédemment, ne permet pas d'identifier la toute première erreur. A l'inverse, choisir $\alpha = 0$ revient à négliger totalement l'information donnée par la valeur absolue du LLR de chaque bit pour ne tenir compte que de la position du dernier bit inversé. Entre ces deux extrêmes, un compromis doit être trouvé par interpolation de l'information donnée par le LLR du/des bits inversés et des valeurs des LLRs des bits précédemment décodés. Ce compromis est obtenu par une optimisation du paramètre α par simulation Monte-Carlo. Il convient de souligner de plus que la valeur optimale dépend du rapport signal à bruit (SNR pour signal-to-noise ratio).

4 Résultats de simulation

Les résultats présentés dans cette partie sont des simulations sur canal gaussien à entrée binaire (BI-AWGN). La position des bits d'information du code polaire (\mathcal{I}) a été obtenue par évolution de densité d'après la méthode proposée dans [4]. Le code détecteur utilisé est systématiquement un CRC de 16 bits dont le polynôme générateur est $g(x) = x^{16} + x^{15} + x^2 + 1$.

La figure 1 présente les performances des décodeurs SC, SCFlip [3] avec $T = 10$ et $T = 50$, D-SCFlip avec $T = \{10, 20, 30, 50, 100\}$ et enfin du SCL avec une taille de liste $L = \{4, 16\}$ pour un code polaire de paramètres ($N = 512, K = 256$). Excepté le décodeur SC, tous les autres utilisent une concaténation avec un CRC de $r = 16$ bits.

Il convient tout d'abord de noter la saturation annoncée du SCFlip [3], pour lequel l'augmentation de T n'apportera qu'un gain insignifiant par rapport aux performances avec $T = 50$. Au contraire, le D-SCFlip offre un gain continu lorsque T augmente. Par ailleurs, les améliorations proposées procurent un gain de 0.35 dB par rapport au SCFlip [3] pour $T = 10$ et de 0.5 dB pour $T = 50$ pour un taux d'erreur bloc (FER) de 10^{-4} . Finalement, les simulations montrent que le D-SCFlip parvient à concurrencer le décodeur SCL jusqu'à une taille de liste $L = 16$ pour le code considéré.

Des résultats similaires apparaissent pour un code de paramètres ($N = 1024, K = 512$), comme montré sur la figure 2. On observe toutefois que le D-SCFlip a plus de difficulté à concurrencer le SCL $L = 16$ et qu'un décodage à inversion est donc une stratégie à privilégier sur des codes relativement courts. A titre de comparaison, on a ajouté les performances du standard Wimax obtenu sur un code de longueur

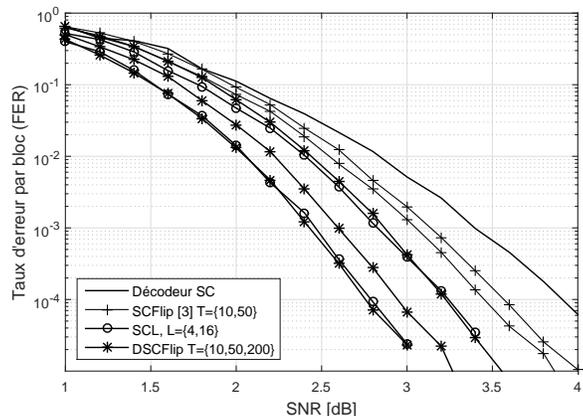


FIGURE 1 – Comparaison des performances des décodeurs SC, SCF [3], SCL et DSCFlip pour un code ($N = 512, K = 256$).

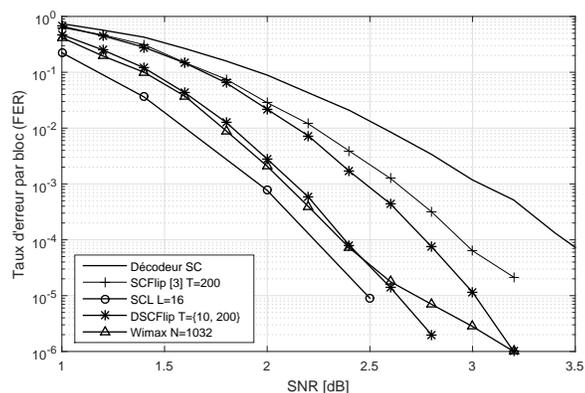


FIGURE 2 – Comparaison des performances des décodeurs SC, SCF [3], SCL et DSCFlip pour un code ($N = 1024, K = 512$).

$N = 1032$ et $K = 516$, utilisant un code LDPC avec décodage par propagation de croyance (BP) et un nombre d'itérations $I_{max} = 100$. On observe que le D-SCFlip est capable de concurrencer également ce standard, et tout particulièrement lorsque l'*error floor* inhérent aux codes LDPC commence à se faire sentir ($FER < 10^{-5}$).

Si le D-SCFlip a besoin d'une valeur de T très élevée pour s'approcher du SCL avec une taille de liste $L = 16$ par exemple, cela ne signifie pas que la complexité du D-SCFlip est supérieure pour autant. En effet, il convient de rappeler que dans le D-SCFlip toute nouvelle tentative n'est réalisée qu'en cas d'erreur avérée des précédentes, tandis que le SCL décode en parallèle les L tentatives. En particulier, le D-SCFlip se contentera d'une seule tentative lorsque le décodage SC réussit. En guise d'illustration, la figure 3 indique le nombre moyen de tentatives effectuées. On observe un net avantage de l'approche dynamique sur le SCFlip [3] et que de plus, ce nombre tend très rapidement vers 1, offrant ainsi une réduction de complexité considérable par rapport au décodeur SCL.

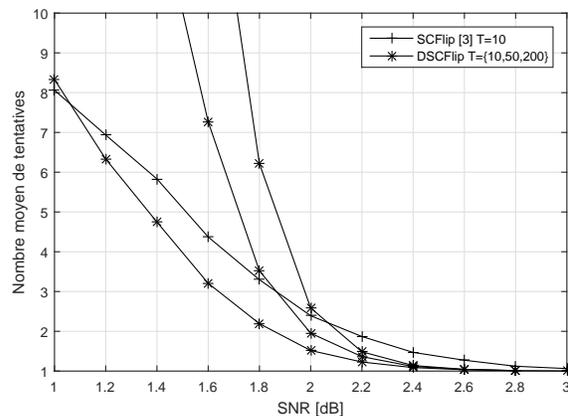


FIGURE 3 – Nombre moyen de tentatives dans le D-SCFlip et SCFlip [3] pour un code ($N = 1024, K = 512$).

5 Conclusion

Le décodeur DSCFlip proposé bénéficie de deux améliorations par rapport à [3]. La première concerne la stratégie dynamique d'exploration des bit-flips, permettant d'éviter d'écartier des candidats pertinents des tentatives considérées, et conséquemment de se soustraire au phénomène de saturation auquel est sujet le décodeur [3]. Mieux encore, la stratégie proposée garantit que les tentatives effectuées sont celles avec la plus forte probabilité de succès (selon la métrique utilisée) parmi tous les tentatives existantes. La deuxième porte sur la métrique optimisée, adaptée à l'exploration des bit-flips d'ordres supérieurs et affinant l'estimation de la probabilité de succès, et offrant ainsi un gain substantiel en terme de performance comme de complexité. Les simulations montrent que le DSCFlip obtenu offre un gain substantiel sur le SCFlip de l'état de l'art et parvient à concurrencer le décodage par liste des codes polaires pour une taille raisonnable de liste tout en conservant une complexité moyenne proche de celle du décodeur SC.

Références

- [1] E. Arıkan. *Channel polarization : A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*. IEEE Transactions on Inf. Theory, vol. 55, no. 7, pp. 3051-3073, 2009.
- [2] I. Tal, et A. Vardy. *List decoding of polar codes*. IEEE Transactions on Inf. Theory, vol. 61, no. 5, pp. 2213-2226, 2015.
- [3] O. Afisiadis, A. Balatsoukas-Stimming, et A. Burg. *A low-complexity improved successive cancellation decoder for polar codes*. IEEE 48th Asilomar Conf. on Signals, Systems and Computers, pp. 2116-2120, 2014.
- [4] R. Mori and T. Tanaka. *Performance and construction of polar codes on symmetric binary-input memoryless channels*. IEEE International Symposium on Information Theory, pp. 1496-1500, 2009.
- [5] L. Chandesris, V. Savin, D. Declercq. *An Improved SCFlip Decoder for Polar Codes*. IEEE, Global Communications Conference, pp. 1-6, 2016.