

Le principe du calcul stochastique appliqué au décodage des turbocodes

Quang Trung DONG¹, Matthieu ARZEL¹ et Christophe JÉGO²

1 : Institut Télécom; Télécom Bretagne,
CNRS Lab-STICC UMR 3192
Technopôle Brest Iroise, CS 83818 29238 Brest
Université Européenne de Bretagne, France
prenom.nom@telecom-bretagne.eu

2 : IPB, ENSEIRB-MATMECA,
CNRS IMS, UMR 5218
351 Cours de la Libération, 33405 Talence
Université de Bordeaux, France
christophe.jego@ims-bordeaux.fr

Résumé – L'article proposé traite de la montée en débit dans les architectures de décodage de codes correcteurs d'erreurs dans le domaine des communications numériques. Dans ce papier, nous présentons l'extension du principe des calculs stochastiques pour le décodage de l'une des deux grandes classes de codes correcteur d'erreur à savoir les codes de type convolutif comme les turbocodes. Le principal défi fut de réduire le nombre de cycle de décodage. Pour ce faire, nous avons considéré deux techniques complémentaires : le passage dans le domaine exponentiel pour les additions stochastiques et l'exploitation du parallélisme au sein du décodeur à l'aide d'une approche multi-flux. L'exploitation de ces deux techniques a permis de proposer un décodage stochastique pour les codes convolutifs et les turbocodes ayant des performances similaires à celles de décodages classiques. Dans un second temps, une implémentation – la première à notre connaissance – d'un décodeur stochastique pour un turbocode a été effectuée sur un circuit FPGA. Ce prototype matériel démontre les capacités en termes de performance et de complexité matérielle d'un tel décodeur stochastique.

Abstract – Stochastic decoding that is inspired by stochastic computation is an alternative technique for decoding of error-correcting codes. The extension of this approach to decode convolutional codes and turbo codes is discussed in this article. The switching activity sensitivity is circumvented and the latching problem is reduced by transforming the stochastic additions into stochastic multiplications in the exponential domain and using multiple streams with deterministic shufflers. The number of decoding cycles is thus considerably reduced with no performance degradation. Stochastic decoding, previously applied to the decoding of LDPC codes, can now be applied to decoding of turbo codes. In addition, the first hardware architecture for stochastic decoding of turbo codes is presented. The proposed architecture makes fully-parallel turbo decoding viable on FPGA devices. Results demonstrate the potential of stochastic decoding to implement fully-parallel turbo decoders.

1. Introduction

Les codes correcteurs d'erreurs sont une des solutions permettant d'améliorer la qualité des communications numériques. Leur principe est d'introduire de la redondance dans la séquence d'information binaire afin de corriger les erreurs de transmission durant la réception de l'information. Deux grandes classes de codes correcteurs d'erreurs existent : les codes convolutifs et les codes en blocs linéaires. Au début des années 90, une nouvelle famille de codes correcteurs d'erreurs a été découverte par C. Berrou [1] : les turbocodes. Cette famille de codes correcteurs d'erreurs est construite par concaténation parallèle de codes convolutifs. Les turbocodes convolutifs sont le résultat de deux innovations majeures: la concaténation parallèle de deux codes convolutifs pour le codage et le décodage itératif. Le décodage itératif est appliqué à des décodeurs élémentaires à entrées et à sorties pondérées (EPSP). Le turbo décodage a de très bonnes performances et une complexité calculatoire raisonnable. Le concept général de décodage itératif appliqué à des décodeurs élémentaires EPSP a ensuite été étendu aux codes produits.

Les principes du calcul stochastique ont été posés dans les années 1960 [2] pour permettre d'effectuer des opérations complexes à faible coût. Par exemple, si des probabilités doivent être traitées, chaque probabilité est préalablement transformée en une séquence de Bernoulli

dont la probabilité d'occurrence de '1' est égale à la probabilité à manipuler. Ces dernières années, des études ont été menées pour appliquer les principes du calcul stochastique au domaine des communications numériques et en particulier à la fonction de décodage des codes correcteurs d'erreurs. Dans ce contexte, le principal intérêt de l'approche stochastique est de tenter de maîtriser la complexité matérielle induite par le développement d'architectures massivement parallèles. Le besoin de ce type d'architecture est quand à lui motivé par des demandes de débit toujours plus élevés à savoir au delà du Gbit/s.

Jusqu'à présent, le principe du décodage stochastique a été appliqué aux codes en blocs linéaires. Dans un premier temps, il a été démontré que ce principe était directement utilisable pour des codes en blocs linéaires de tailles réduites. Ainsi, des décodeurs stochastiques ont été respectivement proposés pour le code de Hamming (7,4) et le code LDPC (16,8) dans [3] et [4]. Les principes du calcul stochastique ont ensuite été étendus dans [5] à un algorithme de décodage utilisant une représentation sous forme de treillis pour le code de Hamming (16,11). Enfin, une première implémentation proprement dite d'un décodeur stochastique a été détaillée dans [6] pour un code LDPC (16,8). Cependant, tous ces travaux ne concernaient que des codes correcteurs d'erreurs de petites tailles, peu utilisés dans les applications de communications numériques. A partir de 2005, des travaux complémentaires

ont permis d'étendre le principe du décodage stochastique à des codes en blocs linéaires de type LDPC ayant des longueurs adéquates avec les standards existants. Pour ce faire, la corrélation entre les différents flux aléatoires au sein du graphe de Tanner sur lequel est appliqué le décodage stochastique est prise en considération. En effet, cette contrainte est forte car le décodage est par définition itératif, ce qui introduit de la corrélation entre les messages et par conséquent, altère les performances du décodage stochastique. Deux techniques ont été proposées afin de réduire ce problème de corrélation. La première reposant sur la structure de « Supernode » [7], favorise la régénération des messages corrélés à partir des probabilités de l'ensemble des messages stochastiques reçus pour le nœud considéré. La seconde technique [8], utilise quant à elle, un effet mémoire sur les arcs du graphe de Tanner pour éliminer la corrélation au cours du processus itératif du décodage stochastique. Les deux approches rendent ainsi possible l'extension du principe du décodage stochastique à des codes LDPC conformes à ceux des standards. Cependant, elles ont pour inconvénient d'augmenter la complexité matérielle des architectures associées. Malgré cela, les résultats d'implémentation d'une première architecture entièrement parallèle de décodage stochastique pour le code LDPC(1024,512) sur un circuit FPGA [9], tente à prouver la pertinence de l'application des principes du calcul stochastique au domaine du décodage des codes correcteurs d'erreurs.

Les principes du calcul stochastique ont également été démontrés pour un algorithme de décodage utilisant une représentation sous forme de treillis [5]. Il est donc possible de l'étendre aux codes dits convolutifs et en particulier à la famille des turbocodes convolutifs. De plus, il est à noter qu'il existe des similitudes entre le principe du décodage stochastique et celui du décodage analogique. Or, des études menées à Télécom Bretagne ont démontré les atouts de l'approche analogique pour le décodage des codes correcteur d'erreurs [10]. L'extension du principe stochastique au décodage de codes convolutifs et de turbocodes est donc l'objet de l'étude détaillée dans cet article.

2. Décodage stochastique à entrées et sorties pondérées

Pour les turbocodes, un processus de traitement itératif est appliqué à des décodeurs élémentaires à entrées et sorties pondérées. Ce type de décodage est une solution qui fournit de très bonnes performances tout en nécessitant un niveau de complexité raisonnable. Parmi les algorithmes de décodage de type EPSP, l'algorithme BCJR [11] s'est finalement imposé. Il consiste à calculer la probabilité de chaque symbole d'une représentation en treillis à partir des probabilités de tous les chemins possibles au sein du treillis entre les états initial et final. Le décodage stochastique de turbocode implique l'application de calcul stochastique à un algorithme basé sur les probabilités a posteriori (APP). La Figure 1 détaille l'échange d'information stochastique entre les différentes sections composant le décodeur reposant sur l'algorithme APP. Il est à noter que dans notre approche, il y a autant de sections que de symboles à

décoder. Chaque section se compose de 5 modules. Le module Γ reçoit les symboles u^i and v^i qui correspondent au $i^{ème}$ symbole transmis et au bit de redondance correspondant. Ce module assure la conversion de u^i et de v^i en des probabilités a priori représentées par des flux stochastiques. Puis les métriques de branche sont calculées. Les métriques de nœud aller et retour sont quant à elles calculées respectivement dans les modules A et B . A l'aide d'un processus récursif, les nouvelles métriques de nœuds α^{i+1} et β^i sont respectivement produites à partir des métriques α^i et β^{i+1} . Puis le module DEC détermine la valeur finale pour chaque symbole. Un module supplémentaire $Extr$ est indispensable si le décodeur EPSP est l'une des composantes d'un turbo-décodeur. Ce module calcule la probabilité sortante de l'information extrinsèque Pr^{ex}_{out} qui est transmise au second décodeur en tant que probabilité entrante de l'information extrinsèque Pr^{ex}_{in} . Tous ces modules échangent des flux stochastiques d'information à travers un réseau de portes logiques correspondant à la représentation sous forme de treillis du code. Chaque étape de décodage traitant un des bits du flux stochastique est considérée comme un cycle de décodage. Le processus de décodage s'achève lorsqu'un nombre fixé de cycle de décodage est atteint.

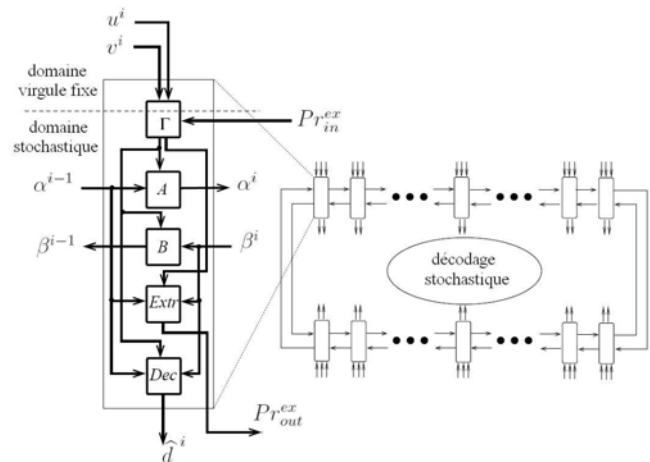


FIG. 1 : décodeur stochastique traitant des probabilités a posteriori

Une comparaison des performances simulées a été effectuée pour différentes versions du décodeur stochastique pour des codes convolutifs et des turbocodes. La Figure 2 détaille les performances en termes de taux d'erreur binaire (TEB) du décodage stochastique pour un code convolutif de longueur 400 bits et de rendement 0.5. Le nombre de cycles de décodage est fixé à 30 000 dans la version de base et différentes optimisations ont été expérimentées. Tout d'abord, nous pouvons constater qu'un décodage combinant l'ajout de mémoire (*Edge Memories, ED*) et un coefficient de pondération (*Noise Dependent Factor, NDS*), comme proposé dans [8] pour traiter le problème de la corrélation, a des performances similaires à celles d'un décodage conventionnel en virgule flottante. Par ailleurs, une étude spécifique s'est focalisée sur le problème de l'addition dans le domaine stochastique. En effet, autant la multiplication de deux flux stochastiques s'effectue très facilement à l'aide d'une simple porte ET,

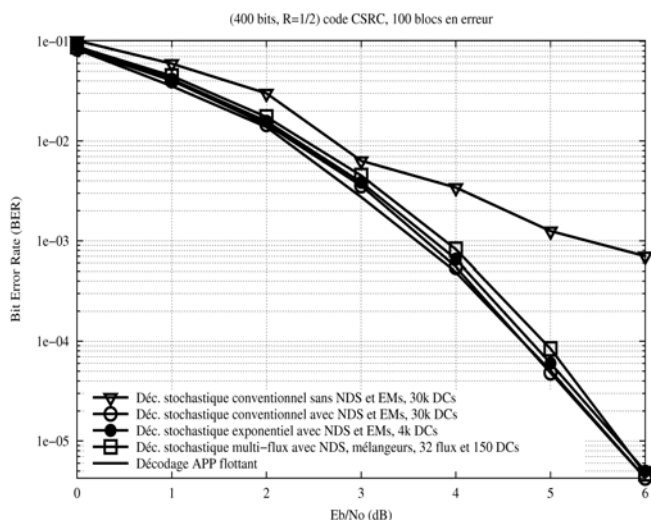


FIG. 2 : performances du décodage stochastique pour un code convolutif de rendement 0,5 et un mot de code de 400 bits

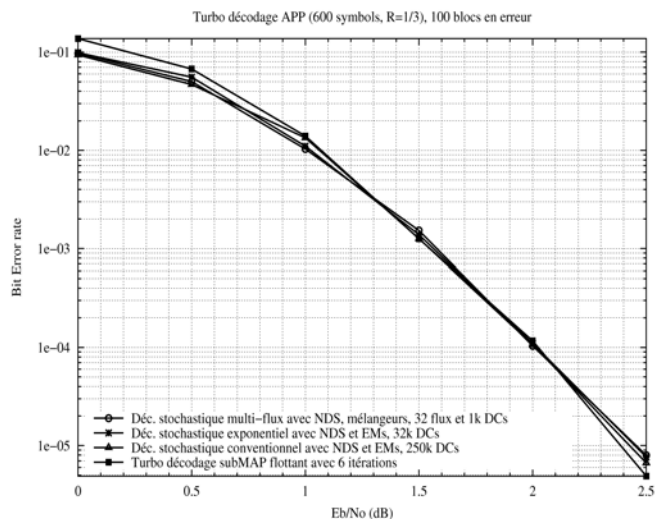


FIG. 3 : performances du décodage stochastique pour un turbocode de rendement 0.33 et un mot de code de 600 bits

autant l'addition de deux flux stochastiques s'effectue par approximation à l'aide par exemple d'un multiplexeur. C'est pourquoi une solution alternative reposant sur le passage dans le domaine exponentiel pour l'opération addition a été expérimentée. Les détails de cette investigation est donné dans [12]. L'intérêt principal est alors de favoriser une diminution du nombre de cycles de décodage. Dans le cas du code convolutif étudié, un gain de 7,5 (30 000 à 4 000) est obtenu pour une légère dégradation des performances d'environ 0,1dB. L'extension du calcul stochastique au décodage des codes convolutifs est donc tout à fait possible. Les performances du décodage stochastique pour un turbocode de longueur 600 bits et de rendement 0.33 sont quant à elles présentées dans la Figure 3. Là encore, un décodage combinant l'ajout de mémoire et un coefficient de pondération permet d'obtenir des performances similaires à décodage itératif classique en virgule flottante. En revanche, 250 000 cycles de décodage sont nécessaires pour obtenir des performances équivalentes à un processus de six itérations. Fort heureusement, il est possible de diminuer cette valeur jusque 32 000 si les additions sont effectuées dans le domaine exponentiel.

Une diminution supplémentaire du nombre de cycle est obtenue par l'introduction de parallélisme au sein du décodeur de base. Nous avons proposé dans [13], une approche multi-flux pour le décodage stochastique qui permet l'introduction de ce type de parallélisme. L'idée est de concevoir une architecture pouvant traiter plusieurs flux en parallèle représentant une même probabilité. Ainsi si chaque probabilité est représentée par p flux stochastiques distincts, alors le nombre de cycles nécessaires au décodage d'un mot de code peut être divisé par p . En revanche une réorganisation de l'architecture est indispensable pour éviter que la complexité matérielle de l'architecture résultante ne soit pas p fois supérieure à celle de l'architecture initiale. Pour ce faire, nous proposons de dupliquer uniquement la logique combinatoire associée aux flux et d'utiliser de simples mélangeurs pour la génération des bits aléatoires et la régénération de bits lors de

problème de corrélation. Dès lors, les EMs ne sont plus nécessaires car la régénération d'un bit est obtenue à partir des autres flux représentant la même probabilité. Cette approche a également été simulées pour des codes convolutifs et des turbocodes. Pour le code convolutif de longueur 400 bits et de rendement 0.5, l'utilisation de 32 flux stochastiques permet d'obtenir des performances similaires pour seulement 150 cycles de décodage (cf Figure 2). Dans le cas d'un turbocode de longueur 600 bits et de rendement 0.33, des performances similaires aux précédentes sont obtenues pour 1000 cycles de décodage comme le montre la Figure 3. L'ensemble de ces contributions démontrent les atouts du décodage stochastique pour la conception d'architectures très haut débit (à savoir au-delà du Gbits) pour le décodage de turbo-décodeur.

3. Implémentation d'un turbo-décodeur stochastique

La conception puis l'implémentation d'architectures de décodeur stochastique pour des turbocodes est un défi majeur. Dans cette section, l'implémentation d'un turbo-décodeur basé sur une approche multi-flux et la réalisation des additions stochastiques dans le domaine exponentiel est détaillée. Il est à noter que le coût additionnel pour les opérations additions dans le domaine exponentiel est raisonnable comme expliqué dans [12]. En effet, l'utilisation de série de Taylor du second ordre est suffisante dans ce contexte pour les transformations exponentielles et logarithmiques. En fait, une architecture stochastique où toutes les probabilités sont représentées par huit flux indépendants a finalement été conçue. Cette architecture offre un bon compromis entre d'une part, le nombre de cycle de décodage et d'autre part, la complexité matérielle pour une première implémentation. Les résultats de synthèse pour un turbo-décodeur adressant un code de longueur 40 symboles et de rendement 1/3 sont récapitulés dans le Tableau 1.

TABLEAU. 1 : Résultats de synthèse logique du décodeur stochastique 8 flux pour un turbocode ($n=40$ bits, $R=1/3$)

Virtex5 LX330		calculs stochastiques		génération de bit aléatoire	
		LUT	Flip-Flop	LUT	Flip-Flop
une section du décodeur APP	Γ	80	0	112	269
	A/B	1 360	768		
	Extr	216	168		
	Dec	152	48		
décodeur APP (40 sections)		72 320	39 360	4 480	10 760
turbo-décodeur (avec entrelaceur)		144 640	80 000		

Tout d'abord, les coûts d'implémentation des cinq modules composant une section du décodeur APP pour les calculs stochastiques et la génération des bits aléatoires sont donnés. Rappelons que des générateurs aléatoires sont indispensables aux décodeurs stochastiques. Ces bits aléatoires sont utilisés pour le contrôle des multiplexeurs et l'adressage des générateurs de flux stochastiques. Fort heureusement, l'impact en termes de complexité de la génération de ces bits aléatoires peut être fortement réduit par leur réutilisation au sein des modules et entre les deux décodeurs APP constituant le turbo-décodeur. Dans notre architecture, 1 920 LUTs et 1 253 bascules Flip-Flops sont nécessaires pour une section du décodeur APP. Par ailleurs, la complexité pour un décodeur APP comprenant 40 sections est donnée dans le Tableau 1. Au final, le coût d'implémentation de l'ensemble du turbo-décodeur stochastique sur un circuit FPGA Virtex5 LX330 est de 90 760 bascules Flip-Flops et de 149 640 LUTs.

Afin de valider la fonctionnalité du turbo-décodeur stochastique, des mesures de TEB ont été réalisées sur une carte de prototypage contenant un circuit FPGA Virtex5 LX330. Pour ce faire, une chaîne de communications numériques comprenant un générateur pseudo-aléatoire, un turbo codeur de longueur 40 symbols et de rendement 1/3, un émulateur de canal Gaussien et le turbo-décodeur stochastique préalablement décrit, a été implémentée sur le circuit FPGA. L'émulateur de canal est conçu à partir d'un générateur d'une variable aléatoire gaussienne suivant la méthode de Wallace comme expliqué dans [14]. Les performances mesurées à partir de la chaîne de communications numériques sont finalement similaires aux performances théoriques d'un décodage conventionnel en virgule flottante. Ce premier prototypage démontre donc qu'il est possible de mettre en œuvre un turbo-décodeur stochastique ayant des performances conformes aux études algorithmiques présentées dans la section précédente.

Conclusion

Dans ce papier, nous avons présenté l'extension du principe des calculs stochastiques pour le décodage de l'une des deux familles de code correcteur d'erreurs s'approchant de la limite de Shannon à savoir les turbocodes. Nous avons en particulier exploré deux

techniques – le passage dans le domaine exponentiel pour les additions stochastiques et l'exploitation du parallélisme au sein du décodeur à l'aide d'une approche multi-flux – afin de réduire le nombre de cycle de décodage. Puis, la conception l'implémentation et le prototypage sur circuit FPGA d'un turbo-décodeur stochastique ont permis de démontrer la faisabilité d'architectures reposant sur des calculs stochastiques. Ces résultats nous encouragent à poursuivre nos investigations autour du décodage stochastique comme solution alternative pour favoriser la montée en débit des architectures de décodage de code correcteur d'erreurs.

Références

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima. *Near shannon limit error-correcting coding: Turbo codes*. In Proc. IEEE Int. Conf. on Communications, pages 1064–1070, May 1993.
- [2] B. Gaines. *Advances in Information Systems Science*, chapter 2, pp 37–172. Plenum, New York, 1969.
- [3] A. Rapley, C. Winstead, V. Gaudet, and C. Schlegel. *Stochastic iterative decoding on factor graphs*, in Proc. 3rd Int. Symp. on Turbo Codes and Related Topics 2003, pp. 507–510.
- [4] V. Gaudet and A. Rapley. *Iterative decoding using stochastic computation*. Electron. Lett., 39(3):299–301, Feb. 2003.
- [5] C. Winstead, V. Gaudet, A. Rapley, and C. Schlegel. *Stochastic iterative decoders*, in Proc. IEEE Int. Symp. on Information Theory 2005, pp.1116–1120.
- [6] W. J. Gross, V. C. Gaudet and A. Milner. *Stochastic Implementation of LDPC Decoders*, Proceedings of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, pp. 713-717, October 30 - November 2, 2005.
- [7] C. Winstead. *Error-control decoders and probabilistic computation*. In Tohoku Univ. 3rd SOIM-COE Conf., Sendai, Japan, Oct. 2005.
- [8] S. Sharifi Tehrani, Shie Mannor, and W. J. Gross. *Survey of Stochastic Computation on Factor Graphs*, Proceedings of the 37th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2007), May 14-16, 2007, Oslo, Norway.
- [9] S. Sharifi Tehrani, Shie Mannor, and W. J. Gross. *An Area-Efficient FPGA-based Architecture for Fully-Parallel Stochastic LDPC Decoding*, Proceedings of the IEEE 2007 Workshop on Signal Processing Systems (SIPS'07), Shanghai, China, Oct. 17-19, 2007.
- [10] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedig, M. Jezequel. *Semi-Iterative Analog Turbo Decoding*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, Volume 54, Issue 6, June 2007.
- [11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. on Inf. Theory, vol. 20, no. 2, Mar 1974.
- [12] Q. T. Dong, M. Arzel, C. Jégo, and W. J. Gross. *Stochastic Decoding of Turbo Codes*, Signal Processing, IEEE Transactions on, Volume 58, Issue 12, Dec. 2010.
- [13] M. Arzel, C. Lahuec, C. Jégo, W. J. Gross and Y. Bruned. *Stochastic multiple-stream decoding of Cortex codes*, Signal Processing, IEEE Transactions on, vol. PP, n°99, pp.1, 0 doi: 10.1109/TSP.2011.2138699.
- [14] O. Sanchez Gonzalez, M. Arzel, C. Jégo, A. Garcia, M. Guerrero, "Design and implementation of a MIMO channel emulator onto FPGA device", IWS'09: XV proyecto Iberchip, 25-27 March, 2009.