

Extension du principe Self-Corrected de l'information extrinsèque au décodage itératif de Turbo Codes

Thibaud TONNELIER¹, Camille LEROUX¹, Bertrand LE GAL¹, Christophe JÉGO¹, Benjamin GADAT², Charly POUILLIAT³

¹IMS Lab, Université de Bordeaux, Bordeaux INP, CNRS UMR 5218, Talence, France

²Thales Alenia Space, Business Segment Navigation and Integrated Communications, Toulouse, France

³Université de Toulouse, IRIT/INP-ENSEEIH/CNRS, Toulouse, France

thibaud.tonnellier@ims-bordeaux.fr, camille.leroux@ims-bordeaux.fr

bertrand.legal@ims-bordeaux.fr, christophe.jego@ims-bordeaux.fr

benjamin.gadat@thalesaleniaspace.com, charly.poulliat@enseeiht.fr

Résumé – Dans cet article, nous étudions une méthode de correction de l'information extrinsèque permettant d'améliorer les performances de l'algorithme Max-Log-MAP pour le décodage de Turbo Codes. Cette méthode intitulée Self-Corrected Max-Log-MAP se base sur une approche originellement proposée pour le décodage des codes LDPC. Son principe consiste en la remise à zéro de l'information extrinsèque de manière dynamique quand certaines conditions sont rencontrées lors du décodage. Grâce à cette méthode, nous obtenons des performances de décodage similaires à celles de l'algorithme optimal MAP mais avec une complexité calculatoire moindre. Cette méthode a été validée par des simulations Monte-Carlo menées dans le cadre du standard de transmission satellitaire CCSDS.

Abstract – In this article, a new method of extrinsic informations manipulation is presented. It enables to improve the Max-Log-MAP algorithm for the iterative decoding of Turbocodes. This method – called Self-Corrected Max-Log-MAP – is based on a approach originally developed for the LDPC codes. Its principle consists in resetting dynamically the extrinsic information during the decoding process each time a given criterion is met. Thanks to the use of this new method, we can achieve the decoding performance of the optimal algorithm MAP with a reduced computational complexity. Successful application to the CCSDS standard is shown by means of Monte-Carlo simulations.

1 Introduction

Les codes correcteurs d'erreurs sont une des solutions permettant d'améliorer la fiabilité d'une communication numérique. Le principe de ces codes est d'introduire de la redondance dans la séquence d'information binaire afin de corriger les erreurs de transmission lors de la réception de l'information. La découverte au début des années 90 des Turbo Codes [1], et plus généralement du principe itératif appliqué aux communications numériques, a révolutionné la manière d'appréhender le traitement numérique de l'information.

Les Turbo Codes, grâce à leur fort pouvoir de correction d'erreurs, sont désormais utilisés dans de nombreux standards de communications sans fil, tels que le standard LTE-Advanced [2] ou le standard CCSDS [3]. Le décodage d'un tel code est réalisé par un processus itératif construit autour de deux décodeurs à Entrées Pondérées et Sorties Pondérées (EPSP). Ces décodeurs s'échangent de l'information, appelée information extrinsèque.

L'algorithme de référence utilisé par les décodeurs EPSP est l'algorithme BCJR [4] (appelé aussi MAP, pour Maximum A Posteriori). Cet algorithme requiert l'utilisation de fonctions transcendentes, de multiplications et de divisions. En conséquence, il ne peut être implanté en l'état dans des systèmes

contraints en surface ou en consommation. Pour résoudre ce problème, des simplifications de l'algorithme BCJR ont été proposées : le Log-MAP et le Max-Log-MAP [5]. Ces algorithmes sous-optimaux ont une complexité calculatoire moindre mais ils souffrent d'une dégradation au niveau des performances de décodage. Cette dégradation est quantifiée par Valenti et Sun dans le cadre du standard UMTS [6]. Ainsi l'algorithme Max-Log-MAP a des performances dégradées de 0.5 dB, en comparaison avec l'algorithme Log-MAP. Néanmoins, en utilisant une remise à l'échelle adéquate de l'information extrinsèque, il est possible de réduire les dégradations à 0.1 dB des performances de l'algorithme MAP [7]. Ainsi, Claussen *et al.* proposent de calculer dans [8] les meilleurs facteurs d'échelles à utiliser. L'amélioration des performances obtenue par cette technique est alors expliquée comme la correction du biais accumulé par l'opération maximum utilisé dans l'algorithme Max-Log-MAP. Il est aussi à noter qu'à la différence du MAP, l'algorithme Max-Log-MAP n'a pas besoin d'estimateur de canal puisqu'il n'est pas dépendant de la variance du bruit du canal.

Tout comme le décodage itératif de Turbo Codes, le décodage des codes LDPC (Low-Density Parity-Check) [9] consiste en un échange d'informations. Savin propose dans [10] une méthode de correction de l'information extrinsèque pour le décodage des codes LDPC. Une implémentation de celle-ci est

présentée dans [11] dans le cadre du standard WiMAX. L'architecture proposée a un coût d'implémentation équivalent à celui de l'algorithme conventionnel (Min-Sum) tout en amenant un gain de performance de correction de 0.5 dB. La principale contribution de notre travail est la transposition de cette méthode au décodage itératif de Turbo Codes. Grâce à celle-ci, nous obtenons les performances de l'algorithme MAP en utilisant une version modifiée de l'algorithme Max-Log-MAP pour les décodeurs EPSP.

Le cadre applicatif sélectionné pour la suite de l'article est le domaine satellitaire. Le Turbo Code retenu est celui du standard CCSDS. Ce standard est l'un des rares utilisant l'algorithme optimal. En effet, les calculs s'effectuant en station de base, les contraintes en terme de consommation et d'intégration sont faibles. C'est pourquoi la norme présente les performances de décodage pour un décodeur exécutant l'algorithme MAP durant 10 itérations [3].

Cet article est organisé comme suit. Dans la section 2 nous fixons le cadre de l'étude. L'algorithme de décodage Self-Corrected Max-Log-MAP est présenté en section 3. Dans la section 4, les résultats de simulation sont discutés. Enfin nous concluons cet article.

2 Le turbo décodeur dédié au standard CCSDS

Le codeur récursif convolutif de ce standard est présenté en figure 1. Comme il s'agit d'un codeur à 16 états, chaque dimension du codeur possède 4 éléments de mémorisation. La fermeture du treillis se faisant via un retour à l'état 0, il faut donc ajouter 4 bits aux informations systématiques et redondantes. Le standard CCSDS intègre un CRC (Cyclic Redundancy Check ou contrôle de redondance cyclique) codé sur 16 bits – CRC-CCITT-16 – validant l'intégrité de chaque trame. Le CRC est un code détecteur d'erreurs. Son principe est d'ajouter à une séquence une valeur de contrôle, basée sur le reste d'une division polynomiale. En effet, les turbo-décodeurs sont des décodeurs complets. Cela signifie qu'ils produisent toujours une trame décodée. Cependant ils ne sont pas capable de détecter une séquence non décodée. Ce CRC est donc utilisé pour identifier les trames erronées en sortie du turbo-décodeur. Nous nous servons alors de ce CRC comme critère d'arrêt durant le processus itératif. Il est à noter que la probabilité de non détection d'erreurs après le turbo décodeur par ce CRC est d'environ 1.5×10^{-5} comme expliqué dans [3].

Parmi les différentes tailles de trame proposées dans la norme, nous nous intéressons uniquement à une longueur de 1784 bits pour l'information systématique. Ainsi le rendement exact final vaut $R = \frac{1784-16}{1784 \times 3 + 4 \times 4} \simeq 0,3294$.

L'architecture générique d'un turbo décodeur pour un code de rendement 1/3 est présentée en figure 2. Chaque décodeur EPSP exploite l'information du canal et l'information *a priori* sous forme de LRV (Log Rapport de Vraisemblance). La première présente deux composantes : le LRV systématique, noté

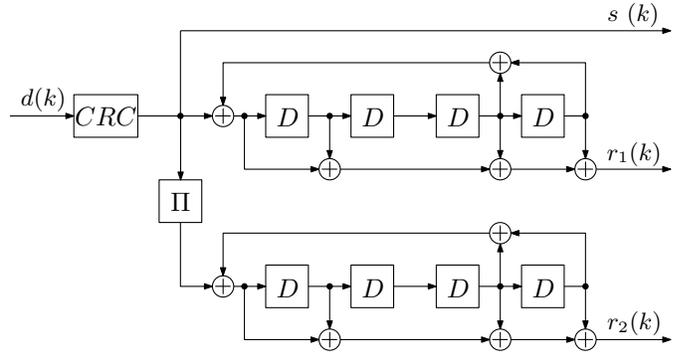


FIGURE 1 – Encodeur de rendement 1/3 de la norme CCSDS

$L_{c,s}(k)$ et le LRV redondant, noté $L_{c,r}(k)$. Le LRV de l'information *a priori* est noté $L_a(k)$. En sortie, il produit l'information extrinsèque $-L_e(k)$ – qui est consommée par l'autre décodeur en tant qu'information *a priori*. Le processus itératif est répété, soit jusqu'à un nombre d'itérations fixé, soit jusqu'à ce qu'un critère d'arrêt soit validé.

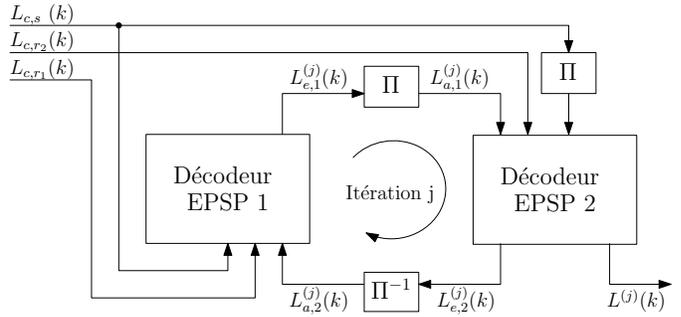


FIGURE 2 – Architecture générique d'un turbo décodeur d'un code de rendement 1/3

3 Le décodage de type Self-Corrected Max-Log-MAP

Le principe de l'approche Self-Corrected (SC) est le suivant : si l'information extrinsèque calculée par un décodeur EPSP pour un certain bit de la trame change de signe d'une itération à la suivante, alors, celle-ci est remise à zéro avant qu'elle ne soit consommée par l'autre décodeur EPSP. L'algorithme 1 représente ce principe en utilisant les notations de la figure 2.

Algorithme 1 : Self-Corrected sur l'une des dimensions du turbo décodeur

Variables : j représente l'itération courante,
 k est l'indice du bit courant dans la trame.
si $sgn(L_{e,1}^{(j)}(k)) \neq sgn(L_{e,1}^{(j-1)}(k))$ **alors**
 $L_{e,1}^{(j)}(k) \leftarrow 0$
fin si

Ainsi, comme le principe de l'approche SC ne concerne que la manipulation de l'information extrinsèque, aucune modification n'est nécessaire du côté du codeur. Les modifications

résultantes de cette approche se situent donc uniquement au niveau de l'interconnexion entre les décodeurs EPSP.

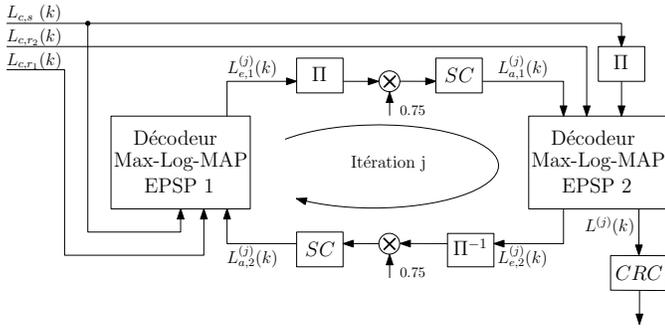


FIGURE 3 – Architecture d'un turbo décodeur Self-Corrected d'un code avec CRC de rendement 1/3

Il est à noter que cette manipulation de l'information extrinsèque ne doit pas être appliquée dès la première itération de décodage. En effet, le principe itératif du turbo décodage se verrait anihilé par le SC. Nous ferions alors face à une dégradation des performances vis-à-vis d'une simple remise à l'échelle. Il est donc préférable de laisser l'algorithme Max-Log-MAP – avec remise à l'échelle – converger durant quelques itérations et ensuite appliquer le principe du SC. Le choix du facteur de remise à l'échelle s'est porté sur 0.75. En effet, cette valeur est communément utilisée car son coût d'implémentation est quasiment nul. Le choix du nombre d'itération sans SC a quant à lui peu d'importance tant qu'il est supérieur à 3. L'algorithme 2 présente toutes les opérations à effectuer par l'approche SC pour décoder une trame, ce en utilisant le CRC-CCITT-16 comme critère d'arrêt. Il est aussi à noter que la vérification du CRC ne doit pas être effectuée dès les premières itérations. En effet, lors de celles-ci, la trame possède encore trop de bits erronés. Le CRC a donc de grandes chances d'être vérifié alors que la trame est erronée. C'est pourquoi nous effectuons la vérification du CRC uniquement à partir de la quatrième itération.

La figure 3 présente l'architecture du turbo décodeur implémentant cette approche.

4 Expérimentations

Dans cette section, nous présentons les résultats de simulations Monte-Carlo. Celles-ci ont été obtenues en utilisant un canal à bruit additif blanc gaussien (BABG) et une modulation de type BPSK. Les simulations ont été effectuées en utilisant une représentation flottante de l'information. Chaque point a été obtenu en simulant jusqu'à 100 trames erronées.

La figure 4 présente une comparaison en terme de Taux d'Erreur Trame (TET) entre trois algorithmes de décodage :

- l'algorithme MAP en appliquant systématiquement 10 itérations comme présenté dans la norme,
- le Scale Factor Max-Log-MAP, dont le facteur de remise à l'échelle vaut 0.75 et en effectuant jusqu'à 32 itérations,

Algorithme 2 : Self-Corrected Max-Log-MAP

```

pour j de 0 à itération maximum faire
    MAX-LOG-MAP_EPSP1( $L_c(s), L_c(r_1), L_{a,2}^{(j-1)}$ )
     $L_{e,1}^{(j)} \leftarrow L_{e,1}^{(j)} \times 0.75$ 
    si j > 4 alors pour tout k
        si  $\text{sgn}(L_{e,1}^{(j)}(k)) \neq \text{sgn}(L_{e,1}^{(j-1)}(k))$  alors
             $L_{e,1}^{(j)}(k) \leftarrow 0$ 
        fin si
    fin si
    MAX-LOG-MAP_EPSP2( $L_c(s), L_c(r_2), L_{a,1}^{(j)}$ )
     $L_{e,2}^{(j)} \leftarrow L_{e,2}^{(j)} \times 0.75$ 
    si j > 4 alors pour tout k
        si  $\text{sgn}(L_{e,2}^{(j)}(k)) \neq \text{sgn}(L_{e,2}^{(j-1)}(k))$  alors
             $L_{e,2}^{(j)}(k) \leftarrow 0$ 
        fin si
    fin si
    si j > 4 et CRC vérifié alors
        SORTIE BOUCLE
    fin si
fin pour

```

- le SC Max-Log-MAP, dont le principe a été détaillé dans la section précédente et en effectuant jusqu'à 32 ou 64 itérations.

Itération	4	5	6	7	8	9	10
RSB=0.6dB	66%	10%	7,5%	5%	3%	2,5%	1,7%
RSB=0.8dB	93%	3%	1,5%	0,8%	0,4%	0,2%	0,1%
RSB=1.0dB	99,5%	0,2%	< 0,1%	< 0,1%	< 0,1%	< 0,1%	< 0,1%

TABLE 1 – Pourcentage des trames corrigées par le Self-Corrected Max-Log-MAP selon l'itération pour 3 RSB différents.

Nous pouvons observer un gain de performance de décodage pour le SC par rapport au Scale Factor. Sur la plage allant d'un rapport signal à bruit (RSB) de 0.6 à 0.9 dB, nous obtenons les mêmes performances que celles recommandées par la norme. Vis-à-vis du Scale Factor Max-Log-MAP, le gain est d'environ une décade, ce qui correspond à un gain de 0.1 dB.

Il est à noter que pour les trois courbes basées sur le Max-Log-MAP, le CRC est utilisé comme critère d'arrêt. Il est alors à prendre en compte dans le calcul du rendement du code. Ainsi le rendement est légèrement plus faible pour ces trois dernières. La variance du bruit du canal étant inversement proportionnelle au rendement du code, à RSB constant, la puissance du bruit est plus importante pour les trois courbes utilisant l'algorithme sous-optimal Max-Log-MAP.

Comme nous employons le critère d'arrêt sur CRC, nous pouvons augmenter le nombre d'itérations maximum. En effet, même si les trames qui nécessitent un grand nombre d'itérations pour être corrigées sont rares, elles permettent d'améliorer le TET. Il y a une limite à ceci. En effet, passer le nombre d'itérations maximum à 64 vis-à-vis de 32 ne provoque qu'un gain de 0.02 dB dans la zone de convergence et n'a pas d'im-

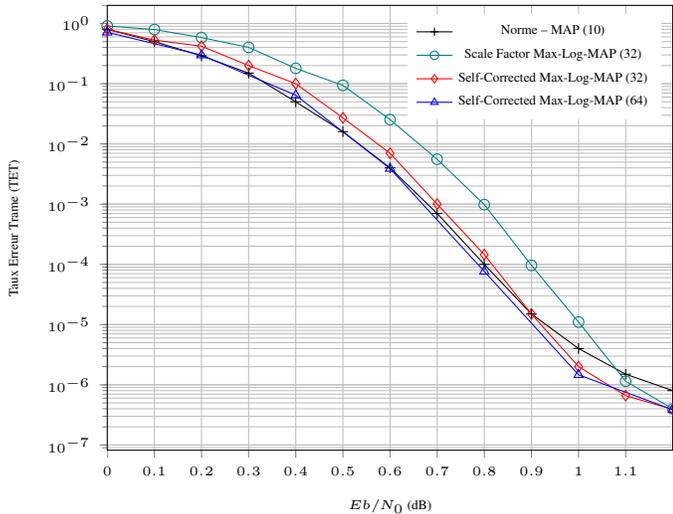


FIGURE 4 – Comparaison du taux d’erreur trame en fonction du RSB entre le MAP, le Scale Factor Max-Log-MAP et le Self-Corrected Max-Log-MAP.

pact sur la zone du plancher d’erreurs. C’est la raison pour laquelle nous avons choisi 32 comme nombre d’itérations maximum. De la sorte, nous augmentons théoriquement la latence maximale du décodeur par 3, en le comparant avec celle d’un décodeur effectuant 10 itérations, comme celui présenté par la norme. Cependant, grâce au critère d’arrêt, le nombre moyen d’itérations est largement inférieur à 10. Par exemple, pour un RSB de 1.0 dB, plus de 99% des trames sont corrigées à la quatrième itération, qui est la première itération durant laquelle le CRC est vérifié. Une statistique du nombre de trames décodées en fonction de l’itération et du RSB est présenté dans le tableau 1 pour l’algorithme SC Max-Log-MAP. Le nombre moyen d’itérations effectuées est quant à lui présenté dans le tableau 2. Nous pouvons constater qu’ils sont faibles. En effet selon [12] le nombre moyen d’itérations utilisées pour décoder des trames CCSDS de rendement 1/3 via un génie – c’est à dire en connaissant la trame transmise – est de 3.3 pour un RSB de 0.8 dB.

RSB	Nb. moy. d’itérations
0.6 dB	5.42
0.8 dB	4.16
1.0 dB	4.01

TABLE 2 – Nombre moyen d’itérations effectuées avec l’algorithme Self-Corrected Max-Log-MAP en fonction du RSB.

Au niveau du surcoût matériel d’implémentation, nous pouvons dire qu’il est faible au regard du gain des performances de décodage. En effet, il est juste nécessaire de stocker le signe de l’extrinsèque correspondant à chaque bit de la trame à l’itération précédente pour chaque domaine. Dans notre cas, cela correspond donc à 1784×2 éléments de mémorisation d’une taille d’un bit. À cela il faut ajouter autant de portes ou-exclusif pour effectuer la comparaison de signe qui contrôlera un multiplexeur sélectionnant la sortie du décodeur ou 0.

5 Conclusion

Dans cet article, nous avons présenté une technique simple de manipulation des informations extrinsèques permettant d’obtenir les performances de l’algorithme MAP. Par l’usage du CRC intégré par le standard en tant que critère d’arrêt, nous réduisons fortement le nombre moyen d’itérations à effectuer pour décoder une trame dans le contexte d’une transmission satellitaire selon le standard CCSDS. Ainsi, nous avons proposé une approche de décodage permettant d’obtenir les performances d’un turbo décodeur optimal tout en conservant une complexité proche de celle d’un algorithme sous-optimal.

Références

- [1] C. Berrou, A. Glavieux and P. Thitimajshima. "Near shannon limit error-correcting coding and decoding : Turbo-codes," *Proc. ICC*, pp. 1064–1070, 1993.
- [2] ETSI 3GPP. "LTE Multiplexing and Channel Coding," *TS 136 212*, V11.1.0, 2013.
- [3] CCSDS. "TM Synchronisation and Channel Coding," *Green Book*, (130x1g2), sec. 7.1–7.14, 2012.
- [4] L. Bahl, J. Cocke, F. Jelinek and J. Raviv. "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, 1974.
- [5] P. Robertson, E. Villebrun and P. Hoeher. "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain", *Proc. ICC*, vol 2, pp 1009–1013, 1995.
- [6] M.C. Valenti and J. Sun "The UMTS Turbo Code and an Efficient Decoder Implementation Suitable for Software-Defined Radios", *International Journal of Wireless Information Networks*, Vol. 8, No. 4, 2000.
- [7] J. Vogt and A. Finger. "Improving the max-log-map turbo decoder", *Electronics Letters*, vol. 36, pp. 1937–1939, 2000.
- [8] H. Claussen, H.R. Karimi and B. Mulgrew "Improved max-log map turbo decoding using maximum mutual information combining", *Proc. PIMRC*, pp. 424–428, 2003.
- [9] R.G. Gallager. "Low-Density Parity-Check Codes", *MA : MIT Press*, 1963.
- [10] V. Savin. Self-corrected Min-Sum decoding of LDPC codes. *Proc. ISIT*, pp. 146–150, 2008. IEEE, 2008.
- [11] O. Boncalo, A. Amaricai and V. Savin. "Memory efficient implementation of self-corrected min-sum LDPC decoder", *Proc. ICECS*, pp 295–298, 2014.
- [12] A. Matache, S. Dolinar and F. Pollara. "Stopping Rules for Turbo Decoders", *TMO Progress Report 42-142*, 2000.