

# Comparaison de deux approches du décodage conjoint source-canal \*

Marion JEANNE<sup>1</sup>, Pierre SIOHAN<sup>2</sup>, Jean-Claude CARLACH<sup>1</sup>

<sup>1</sup>France Télécom R&D, DMR/DDH  
4 rue du Clos Courtel, B.P. 59, 35512 Cesson-Sévigné Cedex, France

<sup>2</sup>INRIA Rennes  
Campus universitaire de Beaulieu, 35042 Rennes Cedex, France  
marion.jeanne@rd.francetelecom.com, pierre.siohan@irisa.fr,  
jeanclaude.carlach@francetelecom.com

**Résumé** – Deux approches du Décodage Conjoint Source-Canal (DCSC) pour des Codes à Longueurs Variables (CLV) sont décrites. L'une utilise un *a priori* sur la source dans le décodage canal, l'autre itère les opérations de décodage de canal et de source. Les techniques sont comparées dans des conditions de simulations identiques et tiennent compte du critère de complexité. A fort Taux d'Erreur Symbole (*TES*), les performances de la seconde méthode, itérative, sont supérieures à celles de la première. A faible *TES* les résultats sont inversés.

**Abstract** – Two approaches of joint source channel decoding for variable length codes are described. The first method uses a source *a priori* to improve the channel decoding. The second method makes some iterations between the channel and the source decoders in order to improve the global decoder. Both techniques are compared under the same conditions of simulation and take into account the complexity criterion. The second method is more suitable for high symbol error rate than the first one, however this result is inverted at low symbol error rate.

## 1 Introduction

Codage de source et codage de canal sont deux fonctions essentielles dans tous les systèmes de communication. Le codage de source élimine au maximum la redondance de la source tandis que le codage de canal ajoute, lui, de la redondance mais de façon contrôlée. La plupart des systèmes existants optimisent ces deux blocs indépendamment, cela est justifié par le théorème de Shannon. Cependant, à cause de limitations pratiques telles que la complexité du codeur/décodeur ou du délai, nous sommes bien loin du schéma théorique de Shannon. Ainsi, il peut être avantageux d'optimiser conjointement le codage/décodage de source et le codage/décodage de canal. En raison de leur grande capacité de compression, les Codes à Longueurs Variable (CLV) sont au cœur des techniques de Décodage Conjoint Source-Canal (DCSC) présentées ici.

Notre but est de comparer deux approches de DCSC afin d'en évaluer leurs avantages et inconvénients. La première approche, cf. par exemple [1], consiste à insérer, au niveau du décodage des turbo-codes, un *a priori* sur la source, supposée sans mémoire puis markovienne d'ordre un. La deuxième méthode améliore le décodage par des itérations successives entre décodage souple du canal puis de la source [2, 3, 4]. Concernant cette seconde approche l'hypothèse de sources sans mémoire est celle retenue par Bauer et Hagenauer [2, 3], tandis que Guyader et al. [4] supposent la source markovienne d'ordre un. De plus le treillis de représentation du CLV diffère. Dans [3] il correspond à une horloge bit tandis que dans [2] et [4], le

treillis est basé sur une horloge symbole.

Dans une première partie nous détaillons ces techniques de DCSC. Puis, dans une deuxième partie nous présentons la comparaison des résultats de simulation de chaque méthode pour une source sans mémoire, ainsi que pour une source markovienne d'ordre un. Enfin nous concluons.

## 2 Les deux méthodes de DCSC

L'intérêt du codage/décodage conjoint source-canal apparaît naturellement du fait des limitations de complexité inhérentes à la réalisation des systèmes de transmission. En conséquence nous allons comparer les performances de deux approches du DCSC dont les complexités sont de même ordre. C'est pourquoi, le schéma « Turbo-décodage avec information *a priori* sur la source » utilise un turbo-code comme code correcteur d'erreur alors que le schéma « Décodage itératif entre source et canal » utilise le principe « turbo » mais avec un simple code convolutif. En effet, cette deuxième méthode utilise des itérations entre deux treillis ce qui est équivalent à un turbo-code au niveau de la complexité et du délai, même si les treillis sont différents.

Dans chaque cas on suppose que l'information *a priori* de la source est connue du décodeur. En pratique elle devra être estimée comme c'est le cas par exemple dans la référence [8]. Nous nous plaçons aussi naturellement dans le cadre d'une hypothèse usuelle en DCSC où l'on considère que le décodeur canal dispose en permanence d'une information fournie par la source. Ce scénario peut se produire par exemple en télécommunications dans le cas de la réception par un mobile d'un signal émis

\* Cette étude a bénéficié d'un financement du ministère français de la recherche dans le cadre du projet RNRT intitulé COSOCATI (<http://www.telecom.gouv.fr/rnrt/index-net.htm>).

par une station de base. Pour s'inscrire dans un contexte plus général des protocoles sont à mettre en place permettant des échanges d'information entre la couche physique et la couche application [7]. On notera que dans ce cas la solution utilisant un turbo code, qui évite des itérations systématiques entre canal et source, présente un avantage en pratique.

## 2.1 Turbo-décodage avec information *a priori* sur la source

Cette technique de DCSC a déjà fait l'objet de quelques articles, par exemple [1], ici nous rappelons brièvement son mécanisme. L'idée est d'améliorer le décodage du code canal par l'utilisation au niveau bit d'une information source *a priori* calculée à partir des probabilités symboles du CLV. Cela est possible en établissant une bijection entre l'arbre du CLV et le treillis du code correcteur d'erreur. Dans [1], le décodage du CLV qui suit le décodage avec *a priori* source du turbo-code est un décodage dur, on intitule « Turbo-Code Décodage Dur » (TC-DD) ce schéma de transmission. Mais ce décodage du CLV peut aussi être souple. En effet, on peut soit utiliser le treillis bit de Balakirsky [5], on appellera « Turbo-Code Décodage Souple Treillis Binaire » (TC-DSTB) cette solution ; soit utiliser le treillis symbole présenté par Bauer et Hagenauer dans [2], cette dernière approche est nommée « Turbo-Code Décodage Souple Treillis Symbole » (TC-DSTS). Ces trois techniques de DCSC sont évaluées ici. Leur principe est rappelé à la figure 1.

On peut remarquer que dans les deux derniers schémas de décodage l'*a priori* sur la source est utilisé une première fois au niveau bit dans le décodage du turbo-code, et une seconde fois sous forme binaire ou symbole, selon le treillis, lors du décodage souple de source.

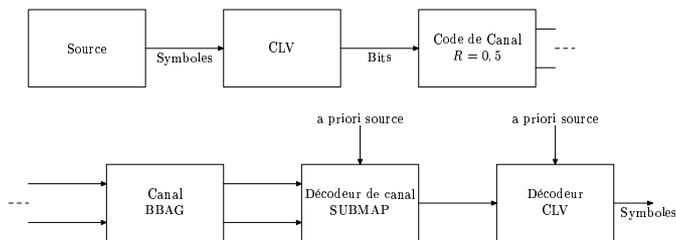


FIG. 1 – Turbo-décodage avec information *a priori* sur la source.

## 2.2 Décodage itératif entre le code correcteur d'erreurs et le CLV

Cette méthode de décodage conjoint est inspirée des turbo-codes séries. Elle consiste à passer, grâce à des itérations, de l'information entre le décodeur du code correcteur d'erreur et le décodeur du CLV. Par similitude au turbo-code série un entrelaceur, E, est inséré à l'encodeur entre le codage de source et celui de canal. Ce schéma de transmission est donné figure 2. Comme nous l'avons vu précédemment le CLV peut être décrit avec deux treillis différents, l'un dit treillis binaire, l'autre dit treillis symbole. Cela conduit à deux algorithmes de décodage : « Code Convolutif Décodage Souple Treillis Bit » (CC-DSTB)

et « Code Convolutif Décodage Souple Treillis Symbole » (CC-DSTS).

Dans ces algorithmes les probabilités *a priori* sur la source sont seulement utilisées pour le décodage souple source car l'entrelaceur casse la bijection entre l'arbre du CLV et le treillis du code convolutif.

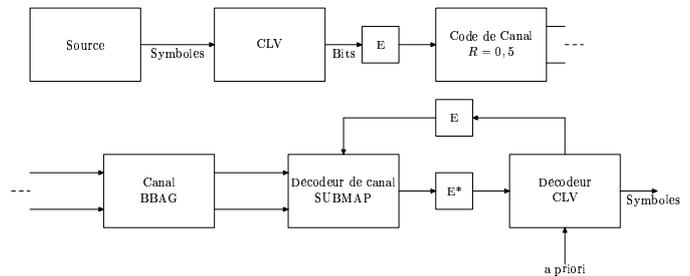


FIG. 2 – Décodage itératif entre le code correcteur d'erreurs et le CLV.

On remarque que pour les méthodes utilisant un décodage dur du CLV ou un décodage souple avec un treillis bit, seul le nombre de bits par paquet est utilisé. Par contre les méthodes où le CLV est représenté par un treillis symbole nécessitent le nombre de bits et le nombre de symboles par paquet.

## 3 Résultats

Pour limiter la propagation des erreurs nous avons fait des paquets de  $M \times M$  bits.  $M \times M$  est aussi la taille de la matrice d'entrelacement. L'entrelaceur suit une loi uniforme, les données sont écrites ligne par ligne dans la matrice carrée et sont lues colonne par colonne. La transmission se termine quand on a transmis  $10^7$  bits ou quand on a 1000 erreurs. Le turbo-code utilisé est une concaténation parallèle de deux codeurs systématiques et récursifs de même polynôme générateur  $(1 + D + D^2 + D^4)/(1 + D^3 + D^4)$ . Ce même polynôme est utilisé pour le code convolutif. Un algorithme SUBMAP est réalisé à chaque décodeur de canal et sur le treillis bit du CLV. Sur le treillis symbole, un BCJR est utilisé. Dans nos simulations nous avons comparé les cinq schémas de décodage décrits précédemment plus le schéma tandem de complexité comparable, *i.e.* un CLV protégé par un turbo-code sans utiliser l'*a priori* sur la source.

### 3.1 Source sans mémoire

La source sans mémoire utilisée est celle à trois symboles {a, b, c} de Murad et Fuja [6], elle est rappelée dans le tableau 1. Si  $R_r$  est la redondance résiduelle relative de la source, *i.e.* la différence entre le taux de codage du code de Huffman et l'entropie de la source le tout divisé par le taux de codage du code de Huffman, alors  $R_r = 0,15$ . La figure 3 montre les performances des différentes méthodes avec une matrice d'entrelacement de  $16 \times 16 = 256$  bits par paquet, *i.e.* 205 symboles en moyenne. La figure 4 donne les performances pour des paquets de taille  $64 \times 64 = 4096$  bits, *i.e.* 3277 symboles en moyenne, ou de taille  $32 \times 32$ . Cette différence de taille d'entrelaceur entre les méthodes utilisant un treillis bit et celle utilisant un treillis symbole pour représenter la source s'explique par une

trop grande complexité du treillis symbole pour des blocs de 4096 bits.

TAB. 1 – Source à 3 symboles.

CLV	Symbole	Probabilité
0	a	0,75
10	b	0,125
11	c	0,125

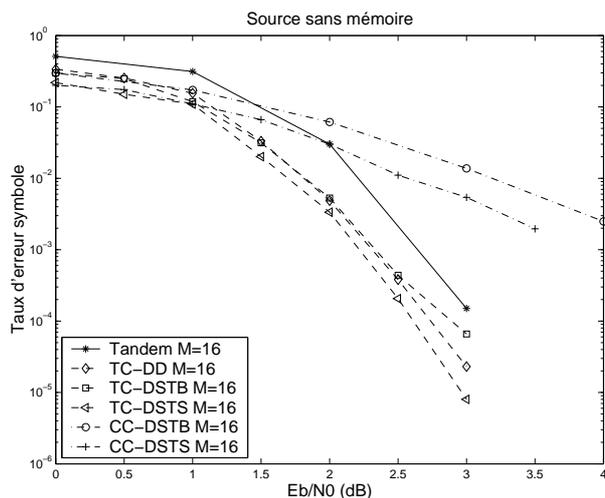


FIG. 3 – Performance des différentes méthodes pour la source sans mémoire de Murad et Fuja.

Pour les techniques utilisant un turbo-code, un décodage souple du CLV apporte peu de gain par rapport à un décodage dur quelle que soit la taille de l'entrelaceur utilisé, dans le meilleur des cas on a 0, 1 dB de gain. Par contre, pour ces méthodes, augmenter la taille de l'entrelaceur améliore les performances, ceci est dû à l'utilisation d'un turbo-code. A un  $TES = 10^{-3}$  on gagne 0, 5 dB en augmentant de  $16 \times 16$  bits à  $64 \times 64$  bits la matrice d'entrelacement pour la méthode TC-DD. Pour les méthodes utilisant le principe « turbo » entre source et canal, un treillis symbole pour décoder le CLV permet de gagner 0, 6 dB à un  $TES = 10^{-2}$  par rapport au treillis bit pour une taille d'entrelaceur de  $16 \times 16$  bits. On remarque aussi qu'une augmentation de la taille de l'entrelaceur détériore les performances. En effet, à  $TES = 10^{-2}$  on perd 0, 7 dB quand on passe de  $M = 16$  à  $M = 32$  pour la technique CC-DSTS et on perd 1 dB quand on augmente  $M$  de 16 à 64 pour CC-DSTB. Cependant, il faut bien noter que cet entrelaceur est tout de même indispensable pour que les méthodes itératives fonctionnent. Pour une telle redondance résiduelle relative, le meilleur schéma itératif entre source et canal, *i.e.* avec une petite taille d'entrelaceur et un treillis symbole n'est jamais meilleur que le meilleur schéma tandem, *i.e.* avec une grande taille d'entrelaceur. Par contre les schémas avec turbo-codes améliorent de 0, 4 dB environ les performances du schéma tandem. Ainsi, pour cette source de faible redondance résiduelle relative, les méthodes utilisant les turbo-codes donnent de meilleurs résultats que les méthodes utilisant les codes convolutifs.

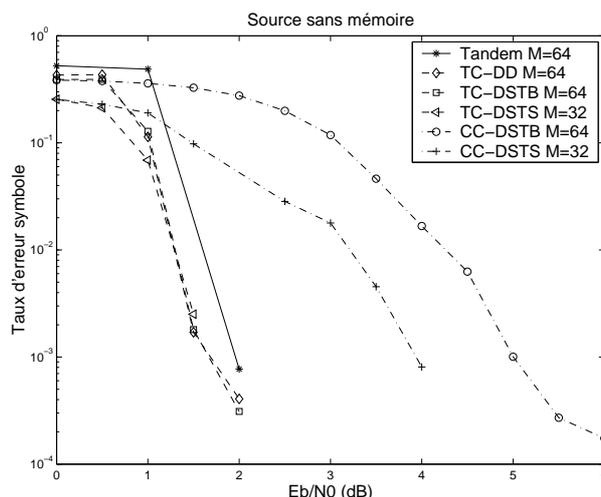


FIG. 4 – Performance des différentes méthodes pour la source sans mémoire de Murad et Fuja.

TAB. 2 – Probabilités de transition de la source de Markov.

Y ↓ X →	a	b	c
a	0,94	0,03	0,03
b	0,18	0,712	0,108
c	0,18	0,108	0,712

### 3.2 Source markovienne d'ordre un

De la même façon qu'au paragraphe 3.1, la source markovienne utilisée est celle à trois symboles de Murad et Fuja [6] dont les probabilités de transition sont rappelées dans le tableau 2. Ici  $R_r = 0, 54$ . Les figures 5 et 6 montrent les performances des différentes méthodes. A la figure 5 la taille de l'entrelaceur est 256 bits. Tandis qu'à la figure 6, et toujours pour des raisons de complexité, la taille de l'entrelaceur est choisie en fonction du type de treillis de décodage : 4096 bits pour le treillis bit, 1024 bits pour le treillis symbole.

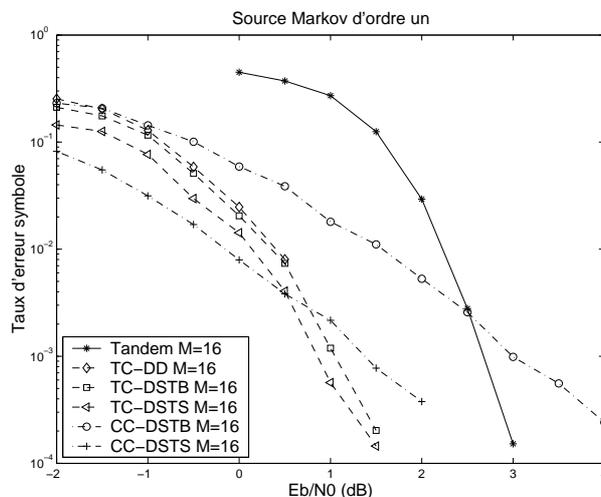


FIG. 5 – Performance des différentes méthodes pour la source markovienne d'ordre un de Murad et Fuja [6].

D'après la figure 5, on gagne jusqu'à 2, 5 dB pour un  $TES =$

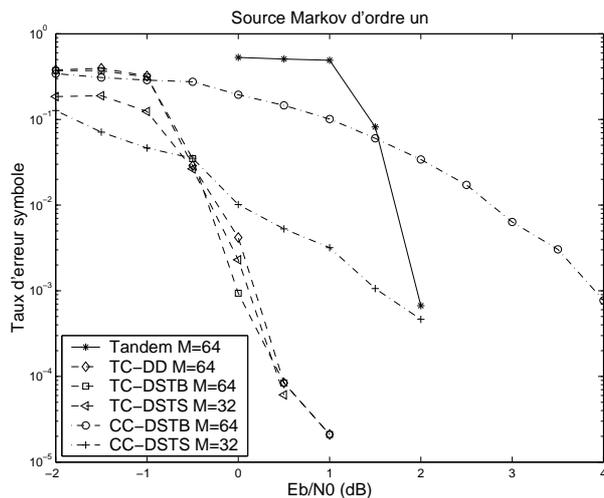


FIG. 6 – Performance des différentes méthodes pour la source markovienne d'ordre un de Murad et Fuja [6].

$10^{-2}$  et  $M \times M = 16 \times 16$  bits en utilisant la meilleure méthode de décodage conjoint source canal. Ce gain s'explique par une large augmentation de la redondance résiduelle relative par rapport au cas sans mémoire. Comme pour une source sans mémoire, pour les méthodes avec un turbo-code, un décodage souple de CLV apporte peu de gain par rapport à un décodage dur. Ici encore, augmenter la taille de l'entrelaceur améliore les performances. Pour les méthodes utilisant le principe « turbo » entre source et canal, un treillis symbole pour décoder le CLV permet de gagner 2 dB à  $TES = 10^{-2}$  par rapport au treillis bit pour une taille d'entrelaceur de  $16 \times 16$  bits. De plus on remarque, qu'une augmentation de la taille de l'entrelaceur détériore les performances. Par rapport à une source sans mémoire, la redondance résiduelle relative augmente quand on considère la source avec mémoire d'ordre un. Dans ce cas, le meilleur schéma itératif entre source et canal, *i.e.* avec une petite taille d'entrelaceur et un treillis symbole est meilleur que le meilleur schéma tandem, *i.e.* avec une grande taille d'entrelaceur, pour des  $TES$  forts. De même, pour des  $TES$  forts, la méthode CC-DSTS donne de meilleures performances que les méthodes de DCSC utilisant les turbo-codes. Cet effet s'inverse pour des  $TES$  plus faibles. Ainsi pour des forts  $TES$ , si la redondance résiduelle en sortie du CLV est suffisante, les techniques utilisant un code correcteur moins performant couplé avec un décodage souple source séparé par un entrelaceur, peuvent donner des résultats plus satisfaisants que les méthodes basées sur des codes plus puissants qui utilisent l'*a priori* de la source au décodage de canal. Cet effet n'est plus vrai pour des faibles  $TES$  où le code correcteur devient l'élément le plus important.

En pratique les codes source utilisés ont généralement plus de 3 éléments. Nous avons donc vérifié, par des simulations non reportées ici, qu'avec des alphabets plus grands les conclusions de la comparaison restaient pratiquement inchangées en ce qui concerne l'évaluation des performances. Par contre du point de vue de la complexité une augmentation de la taille de l'alphabet source pénalise plus fortement l'ensemble des techniques utilisant une technique de décodage souple du treillis source, en particulier celles utilisant le treillis symbole.

## 4 Conclusion

Dans cette étude nous avons présenté une comparaison entre deux approches du DCSC. L'une utilise les probabilités d'apparition des symboles de source au niveau du décodage canal, l'autre applique le principe turbo entre les décodeurs de source et de canal. Par rapport à un décodage tandem classique ces méthodes permettent des gains en performance d'autant plus importants que la redondance résiduelle de la source est élevée. Chaque méthode peut utiliser deux supports de décodage souple pour décoder le CLV, le premier est un treillis binaire, le second est un treillis symbole. Ce deuxième support demande beaucoup plus de complexité. Dans le cas du DCSC utilisant des itérations entre source et canal ce deuxième treillis apporte une forte amélioration par rapport au premier support. Par contre pour les techniques utilisant les turbo-codes avec un *a priori* sur la source le gain est beaucoup plus faible. De plus on remarque que la technique introduite dans [2], fonctionne mieux avec un petit, mais indispensable, entrelaceur, tandis que la technique développée dans [1] préfère un entrelaceur long. Enfin, nous noterons qu'à fort  $TES$  la méthode CC-DSTS donne de meilleures performances que celle de TC-DSTS. A fort  $TES$  cet effet s'inverse.

## Références

- [1] M. Jeanne, J. C. Carlach, P. Siohan et L. Guivarch. Source and joint source-channel decoding of variable length codes. *ICC*, vol. 2, New York, USA, avril 2002.
- [2] R. Bauer et J. Hagenauer. Iterative source/channel-decoding using reversible variable length codes. *DCC*, pp. 93–102, Snowbird, UT, mars 2000.
- [3] R. Bauer et J. Hagenauer. On variable length codes for iterative source/channel decoding. *DCC*, pp. 273–282, Snowbird, UT, mars 2001.
- [4] A. Guyader, E. Fabre, C. Guillemot et M. Robert. Joint source-channel turbo decoding of entropy-coded sources. *IEEE Journal on Selected Areas in Communications*, vol. 19, n° 9, pp. 1680–1696, septembre 2001.
- [5] V. B. Balakirsky. Joint source-channel coding for variable length codes. *ISIT*, p. 419, Ulm, Germany, juin/juillet 1997.
- [6] A. H. Murad et T. E. Fuja. Joint source-channel decoding of variable length encoded sources. *Information Theory Workshop (ITW)*, pp. 94–95, Killarney, Ireland, juin 1998.
- [7] S. Merigeault et C. Lamy. Concepts for exchanging extra information between protocol layers transparently for the standard protocol stack. *ICT'2003*, Tahiti, février 2003.
- [8] C. Weidmann et P. Siohan. Décodage conjoint source-canal avec estimation en ligne de la source. *Coresa'03*, Lyon, France, janvier 2003.