

Influence de la quantification sur les performances des Turbo décodeurs

Michel Jézéquel, Claude Berrou, Alain Glavieux et Jean-René Inisan

École Nationale Supérieure des Télécommunications de Bretagne
Technopôle de Brest Iroise
B.P. 832
29285 BREST CEDEX
FRANCE

RÉSUMÉ

Les performances des *turbo codes* pour des faibles taux d'erreurs binaires (10^{-7} à 10^{-10}) ne sont pas encore bien connues. Les tests fonctionnels de 2 circuits intégrés assurant le codage et le décodage de *turbo codes* ont permis de mieux appréhender leur comportement pour ces taux d'erreurs. Ce papier présente les résultats de mesures effectuées à l'aide d'un de ces circuits (*turbo4*). Il montre qu'une dégradation du gain de codage peut apparaître, mais qu'il est possible d'éviter cette dégradation à condition de choisir des paramètres de décodage appropriés.

1 Introduction

Les *turbo codes* constituent une nouvelle famille de codes correcteurs d'erreurs binaires [1,2,3]. Il s'agit, en fait, d'une concaténation parallèle de codes convolutifs systématiques et récurrents suivie d'un éventuel poinçonnage. Le processus de décodage est itératif, chaque itération apportant un gain de codage.

Il existe actuellement deux circuits intégrés codeur/décodeur de *turbo codes*. Le premier, appelé "CAS5093" est distribué par COMATLAS [4]. Il est constitué de 5 décodeurs de Viterbi à décisions pondérées [5] et de 4 matrices 32x32 pour l'entrelacement et le désentrelacement des données. Ce circuit réalise 2,5 itérations de décodage. Le second circuit, appelé *turbo4* n'effectue qu'une itération de décodage mais il est cascable. Ainsi l'utilisateur peut concevoir un décodeur ayant un nombre d'itérations adapté à son application.

Ces 2 circuits ont permis de mieux appréhender le comportement des *turbo codes* pour des faibles taux d'erreurs (10^{-7} à 10^{-10}), difficiles à évaluer par simulation. En effet des questions telles que : y a-t-il un taux d'erreurs plancher ? ou plus simplement y a-t-il une dégradation du gain de codage lorsque le rapport signal à bruit augmente ? ne pouvaient pas avoir de réponses argumentées sans passer par une phase expérimentale.

Ce papier présente d'abord le circuit *turbo4* puis le système qui a permis son test fonctionnel. Enfin il montre, grâce aux résultats de ces tests, l'importance du choix des paramètres de décodage en fonction du rang de l'itération.

ABSTRACT

Performances of *turbo codes* are not well known for low binary bit error rates (10^{-7} to 10^{-10}). Functional tests of 2 circuits implementing *turbo coding* and *turbo decoding* have given some indications about the behavior of *turbo codes* for low bit error rates. Results of the tests on one of these circuits (*turbo4*) are presented in this paper. Results show a flattening degradation. However this degradation can be avoided if decoding parameters are well chosen.

2 Turbo codeur/décodeur *Turbo4*

Turbo4 est un circuit codeur/décodeur correcteur d'erreurs binaires utilisant le principe des *turbo codes*. Il a été conçu pour servir de circuit prototype dans de nombreuses applications, pour des rendements de codage allant de $R=1/3$ à $R=5/6$. Il offre des facilités pour la synchronisation automatique du décodeur. Des entrées de programmation sont disponibles afin d'optimiser le processus itératif de décodage en fonction de l'application.

Il ne peut assurer la fonction de codage et celle de décodage simultanément. Le choix du mode de fonctionnement se fait en configurant une entrée du circuit.

2.4 Codage

Le codeur est réalisé par la concaténation parallèle de deux codes convolutifs systématiques récurrents de longueur de contrainte $K=5$, séparés par un entrelaceur convolutif de taille 64x32. Le codeur fournit dans le cas général 3 symboles : X, Y1 et Y2. X représente la donnée d'information, Y1 et Y2 sont les redondances produites par les deux codes systématiques récurrents. Différents rendements de codage peuvent être obtenus en poinçonnant les symboles Y1 et/ou Y2 et éventuellement X pour des rendements élevés. Pour le cas particulier du rendement $R=1/2$, le poinçonnage est effectué à l'intérieur du circuit.

2.5 Décodage

Le circuit utilisé en décodeur réalise une seule itération de *turbo* décodage. Il est cependant cascable ce qui permet d'adapter le nombre d'itérations à l'application visée.

La figure 1 présente un synoptique très simplifié du décodeur utilisé dans le circuit *turbo4*. Il est construit autour de 2 décodeurs de Viterbi à sorties pondérées [5], d'un entrelaceur et d'un désentrelaceur.

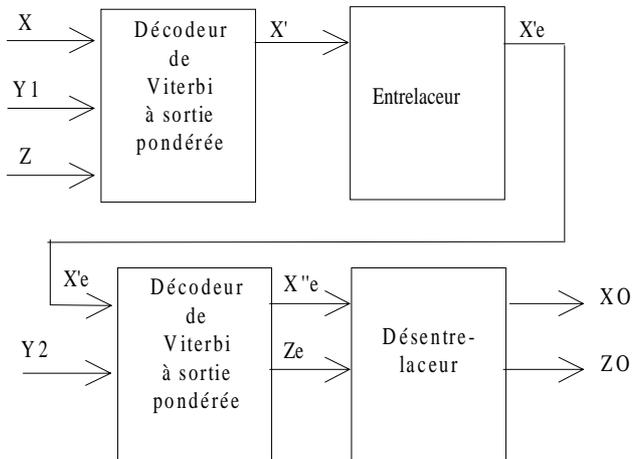


Figure 1 : Schéma de principe du *turbo décodeur*

Le premier décodeur reçoit du canal des informations bruitées X (données) et $Y1$ (redondance du premier code) ainsi que Z (information extrinsèque) provenant de l'itération précédente. Ce premier décodeur fournit une estimation des données (X') qui, après entrelacement ($X'e$), est transmise au second décodeur. Celui-ci calcule une nouvelle estimation ($X''e$) des données et de l'information extrinsèque (Z_e) en utilisant $X'e$ et $Y2$ (redondance du second code). Les données et l'information extrinsèque sont disponibles en sortie du circuit (XO, ZO) après désentrelacement.

$X, Y1, Y2$ et Z sont codés sur 4 bits.

2.6 Entrelacement/désentrelacement

L'entrelacement et le désentrelacement sont du type convolutif et n'utilisent qu'une seule mémoire de taille 64×32 pour chacune des fonctions. La latence totale apportée par un entrelacement et un désentrelacement, exprimé en bits est de 2048. Dans l'entrelaceur, les données sont écrites régulièrement, ligne par ligne, et sont extraites dans un "désordre contrôlé" satisfaisant certaines règles, en particulier un degré suffisant de non-uniformité. Lors du désentrelacement, écriture et lecture s'effectuent de façon duale : écriture "désordonnée", lecture régulière, ligne par ligne.

2.7 Synchronisation/supervision

Comme tout décodeur de code concaténé, le *turbo décodeur* doit être synchronisé. Compte tenu de la diversité des applications envisagées pour le circuit *turbo4*, il n'a pas été possible de proposer une synchronisation automatique pour toutes les configurations de fonctionnement du circuit. Seule une utilisation avec un code de rendement $1/2$ permet de bénéficier d'une telle fonctionnalité. Toutefois le décodeur peut réaliser une grande partie du travail de synchronisation lors d'une utilisation du codeur avec d'autres rendements.

3 Tests du circuit

Le test fonctionnel de circuits de codage et de décodage nécessite l'utilisation d'un simulateur de canal. Celui-ci est généralement de type analogique. Dans ce cas, l'emploi d'un générateur de bruit analogique rend indispensable l'utilisation d'un convertisseur analogique-numérique en aval du circuit à tester. De plus, il est souvent très compliqué de fixer et de reproduire précisément les caractéristiques du bruit du canal considéré. D'autre part la mesure de la puissance de bruit s'effectue à l'aide d'un appareillage (filtres, voltmètres, ...) haut de gamme et demande beaucoup de précautions.

La génération programmée de ces bruits est de nature à simplifier la mise en oeuvre du système de test. La figure 2 présente le synoptique du système de test utilisé.

Le codeur ainsi que les modules 1 à 5 du décodeur sont réalisés à l'aide de circuits *turbo4*. Le décodeur peut être synchronisé sur le codeur soit par sa synchronisation interne soit par le système de test.

L'analyseur de transmission génère des séquences pseudo aléatoires de bits. Ceux-ci sont codés par un circuit *turbo4*. Il fournit 3 informations : $X, Y1$ et $Y2$ correspondant à la donnée initiale et à la redondance des deux codeurs élémentaires. Ces informations sont bruitées par l'émulateur de canal puis poinçonnées avant d'être transmises au décodeur constitué de 5 circuits *turbo4*.

Les données issues des modules 1 à 5 sont comparées à celles émises par l'analyseur de transmission. Ces comparaisons permettent de mesurer le taux d'erreurs après 1 à 5 itérations de décodage en fonction du type de canal et du rapport signal à bruit simulé par l'émulateur de canal.

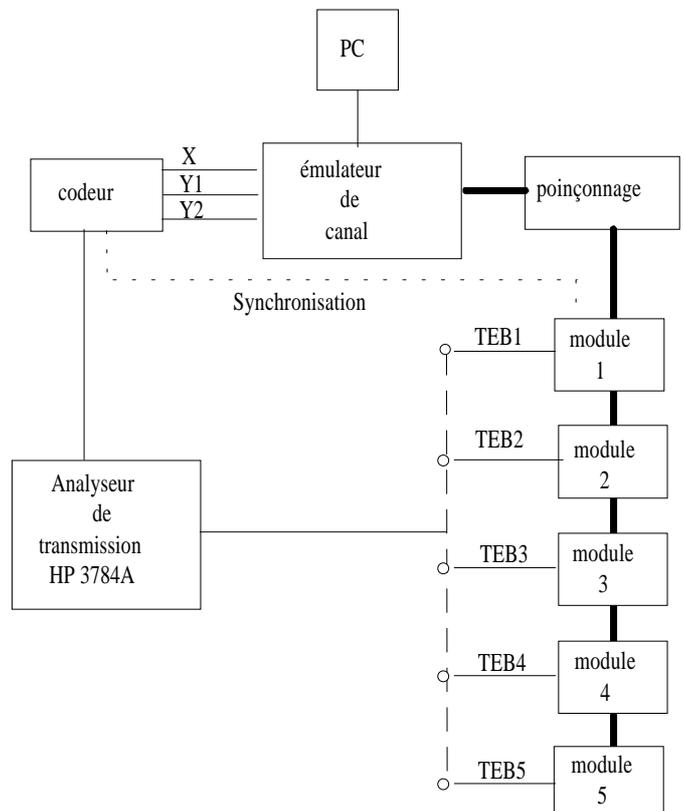


Figure 2 : Schéma de principe du système de test

La simulation du canal est, en fait, réalisée par l'association de l'émulateur de canal et du PC (ordinateur personnel). Le principe consiste à transformer chaque bit reçu par l'émulateur en un symbole bruité codé en complément à 2 sur 4 bits. Ce format de codage correspond à celui attendu par le circuit de décodage sur ses entrées X, Y1 et Y2. Le décodeur reçoit donc des informations équivalentes à celles qu'il recevrait dans le cas d'une transmission réelle où les échantillons reçus seraient analogiques puis quantifiés sur 4 bits, à l'aide d'un convertisseur analogique-numérique.

Cette transformation de bits en échantillons bruités pourrait être directement effectuée par le PC mais, dans ce cas, le débit auquel fonctionnerait le système de test serait relativement faible. Ce qui interdirait les mesures à taux d'erreurs faibles.

La méthode utilisée consiste à faire calculer par le PC un grand nombre d'échantillons d'information bruités représentatifs du canal simulé. Ces échantillons sont transmis à l'émulateur de canal qui les enregistre dans une mémoire à accès rapide (RAM). Toutes ces opérations sont réalisées avant l'utilisation du système en tant que simulateur de canal.

Lors de la mesure l'émulateur de canal associe à chaque bit reçu un échantillon bruité. Cette association se fait de manière aléatoire par un jeu de générateurs pseudo-aléatoires qui balayent l'ensemble des adresses de la mémoire. Cette méthode permet de simuler un canal par une méthode numérique à des débits élevés (10 Mbit/s dans le cas présent). Il faut cependant s'assurer que la taille de la mémoire RAM soit suffisante et que le caractère aléatoire de la lecture soit garanti.

Dans une première étape un nombre important de mesures ont été effectuées afin de valider le système de test et les fonctionnalités du circuit *turbo4* pour différents types de canaux [6]. Dans une seconde étape des mesures mettant en évidence le comportement des *turbo codes* pour des faibles taux d'erreurs binaires (10^{-7} à 10^{-10}) ont été réalisées.

4 Interprétation des résultats

La figure 3 présente le taux d'erreurs binaires obtenu après 1 à 5 itérations de décodage, pour un canal gaussien et pour un rendement de 1/2. Les mesures à faible taux d'erreurs binaires (TEB) font apparaître un changement de pente des courbes du TEB.

Pour comprendre la cause de cette dégradation il est nécessaire de revenir sur la structure interne du décodeur de *turbo4* (cf. § 2.5).

Comme le montre la figure 1 le second décodeur reçoit X'e et Y2 constitués respectivement par une suite d'échantillons x_n et y_{2n} avec :

$$x_n = d_n + \tilde{b}_n$$

où d_n est la donnée émise par le codeur.

$$y_{2n} = c_n + b_n$$

où c_n est la redondance émise par le codeur.

b_n est un bruit gaussien, centré et \tilde{b}_n peut être assimilé à un bruit gaussien, centré. Ces deux bruits sont non corrélés et ont respectivement pour variance :

$$\sigma_b^2 = E\left\{\left(b_n\right)^2\right\} \text{ et } \tilde{\sigma}_b^2 = E\left\{\left(\tilde{b}_n\right)^2\right\}$$

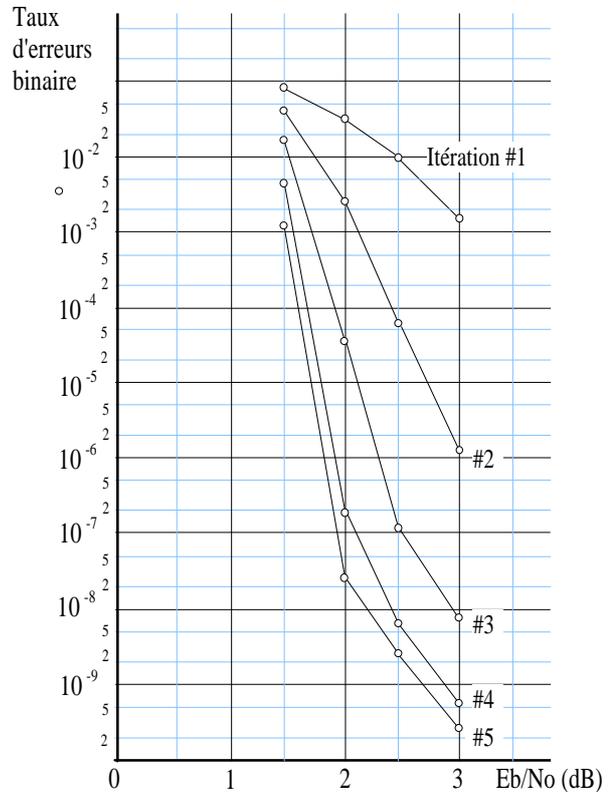


Figure 3 : Test de *turbo4* canal gaussien, rendement du code = 1/2

Le second décodeur prend la décision que la suite i notée, $\{d_n^i, c_n^i\}$ a été émise si :

$$\Pr\left[\{d_n^i, c_n^i\} / \{x_n, y_{2n}\}\right] > \Pr\left[\{d_n^j, c_n^j\} / \{x_n, y_{2n}\}\right] \quad (1)$$

où $\{x_n, y_{2n}\}$ est la séquence d'observation

En utilisant la règle de Bayes et en supposant que toutes les suites i sont équiprobables l'expression (1) devient :

$$p\left[\{x_n, y_{2n}\} / \{d_n^i, c_n^i\}\right] > p\left[\{x_n, y_{2n}\} / \{d_n^j, c_n^j\}\right] \quad (2)$$

Conditionnellement à $\{d_n^i, c_n^i\}$ les variables x_n, y_{2n} sont gaussiennes et mutuellement indépendantes, d'où la relation (2) peut encore s'écrire :

$$\sum_n \log p(x_n, y_{2n} / d_n^i, c_n^i) > \sum_n \log p(x_n, y_{2n} / d_n^j, c_n^j) \quad (3)$$

soit encore :

$$\sum_n \left[(x_n - d_n^i)^2 + \frac{\tilde{\sigma}_b^2}{\sigma_b^2} (y_{2n} - c_n^i)^2 \right] > \sum_n \left[(x_n - d_n^j)^2 + \frac{\tilde{\sigma}_b^2}{\sigma_b^2} (y_{2n} - c_n^j)^2 \right] \quad (4)$$

En développant l'expression (4) et en tenant compte du fait que $(d_n^i)^2 = (c_n^i)^2 = 1$ on prend la décision i si :

$$\sum_n x_n d_n^i + \beta y_{2n} c_n^i > \sum_n x_n d_n^j + \beta y_{2n} c_n^j$$

avec $\beta = (\tilde{\sigma}_b)^2 / (\sigma_b)^2$

Dans un schéma de décodage classique, β est égal à 1 car la variance du bruit affectant les données est la même que celle du bruit affectant la redondance. Par contre ce n'est pas le cas pour les *turbo codes* puisque X'e devient de plus en plus fiable au fil des itérations.

Ainsi, par exemple, dans le cas, présenté figure 3, d'un *turbo code* travaillant sur canal gaussien avec un rendement de code R=1/2, le taux d'erreurs sur X'e pour la 5^{ème} itération est aux environs de 10^{-7} correspondant à un Eb/No supérieur à 10 dB soit à un β de l'ordre de 0,1.

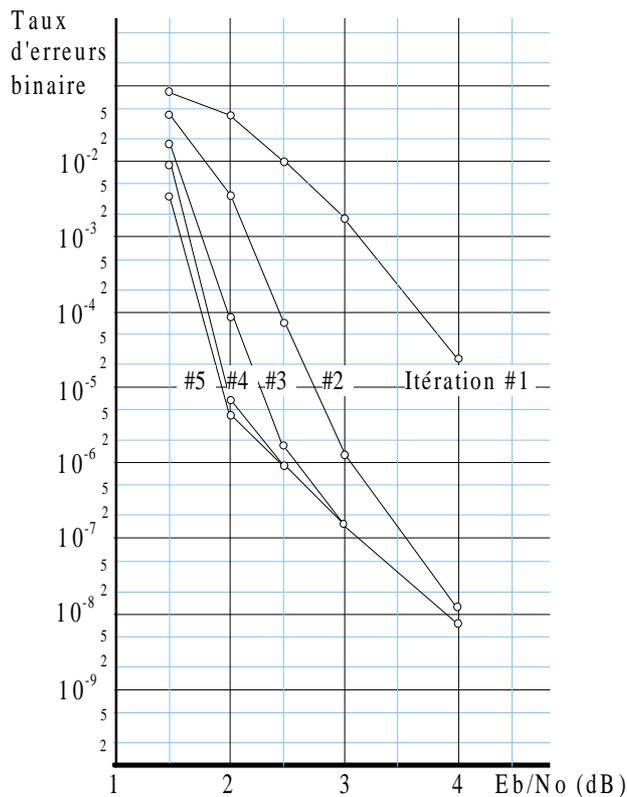


Figure 4 : Test de *turbo4*
canal gaussien, rendement du code = 1/2, $\beta=1$

Une simple multiplication de Y2 par β permet d'utiliser un décodeur de Viterbi à sorties pondérées classique, pour le second décodeur d'un *turbo code*. C'est la solution qui a été retenue pour la conception de *turbo4*. Ainsi le coefficient β du circuit est programmable ce qui permet de l'ajuster en fonction du rang de l'itération. Des tests ont été réalisés en programmant β à 1 pour toutes les itérations (figure 4). Le changement de pente de la courbe de TEB lorsque le rapport signal à bruit augmente montre l'importance de l'adaptation du coefficient au rang de l'itération.

En réalité, et malheureusement, le circuit *turbo4* propose uniquement 4 valeurs pour β (1 ; 0,7 ; 0,55 ; 0,45). Ainsi la valeur théorique de 0,1 (précédemment citée comme adaptée à la cinquième itération avec un rendement 1/2) ne peut être programmée. C'est ce qui explique la dégradation de la pente observée sur la figure 3.

D'autre part la quantification sur 4 bits des échantillons présentés au décodeur *turbo4*, permet de coder une valeur neutre, 7 positives et 7 négatives. Ainsi multiplier Y2 par $\beta=0,1$ conduit à ne pouvoir utiliser que 3 de ces valeurs : la neutre ainsi que la positive et la négative ayant les plus faibles valeurs absolues. Pour éviter ce problème il faut quantifier les données sur 5 bits minimum. Dans ce cas les opérateurs de calcul de métriques et de sélection doivent être dimensionnés, eux aussi, sur 5 bits ce qui augmente la taille du silicium nécessaire à la réalisation de ces opérations. Cependant cette augmentation n'est pas critique puisque qu'elle ne concerne pas tout le circuit (environ 75% de la surface) et que la complexité est proportionnelle au nombre de bits. Par contre cette augmentation du nombre de bits peut devenir un facteur limitant pour la fréquence de fonctionnement du circuit.

7 Conclusions

En conclusion, il est possible d'éviter les changements de pente de la courbe de TEB pour les faibles taux d'erreurs. Cependant, cela se fait au prix d'une augmentation de la complexité du circuit due à l'utilisation d'une quantification plus fine des données.

Ainsi, plutôt que d'essayer de réaliser un décodeur universel, capable de répondre à toutes les applications, il est préférable de concevoir des circuits spécifiques pour chaque application ce qui permet de choisir les paramètres de décodage adaptés à celle-ci.

En effet, à titre d'exemple, une modification de la pente du TEB pour des taux d'erreurs inférieurs à 10^{-7} n'est pas gênante pour une application nécessitant un TEB inférieur à 10^{-6} . Il n'est donc, dans ce cas, pas utile d'augmenter la complexité du circuit.

8 Références

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes", Proc. of IEEE ICC '93, Geneva, pp. 1064-1070, May 1993.
- [2] C. Berrou and A. Glavieux, "Turbo-codes, general principles and applications", Proc. of the 6th Int. Tirrenia Workshop on Digital Communications, Pisa, Italy, pp. 215-226, September 1993.
- [3] C. Berrou, A. Glavieux, "Near optimum error correcting coding and decoding : turbo-codes", IEEE Transactions on communications, Vol. 44, N°10, pp. 1261-1271.
- [4] C. Berrou and G. Lochon, "CAS5093 : Turbo encoder/decoder", Data sheet, COMATLAS, Chateaubourg, France, November 1993.
- [5] C. Berrou, P. Adde, E. Angui and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture", Proc. of IEEE ICC '93, Geneva, pp. 737-740, May 1993.
- [6] M. Jézéquel, C. Berrou, J. R. Inisan and Y. Sichez, "Test of a Turbo-Encoder/Decoder", TURBO CODING Seminar, Lund, Sweden, pp. 35-41, 28-29 August 1996.