HAL
open science

Contribution à l'amélioration des performances de systèmes de transmissions numériques : turbo-communications et circuits

Catherine Douillard

▶ **To cite this version:**

Catherine Douillard. Contribution à l'amélioration des performances de systèmes de transmissions numériques : turbo-communications et circuits. Théorie de l'information [cs.IT]. Université Bretagne Sud, 2004. tel-01782252

HAL Id: tel-01782252

https://hal.science/tel-01782252v1

Submitted on 1 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Bretagne Sud

ENST Bretagne

Rapport de synthèse

présenté par

Catherine Douillard

pour l'obtention de

l'Habilitation à Diriger des Recherches

Contribution à l'amélioration des performances de systèmes de transmissions numériques : turbo-communications et circuits

Soutenue le 25 mai 2004 devant le jury composé de :

Eric MARTIN	Président
Emmanuel BOUTILLON	Rapporteur
Jean-François HÉLARD	Rapporteur
R. Michael TANNER	Rapporteur
Claude BERROU	
Hikmet SARI	

Remerciements

Je tiens tout d'abord à remercier l'ensemble des évaluateurs de ce travail : Emmanuel Boutillon, Professeur à l'Université de Bretagne Sud, Jean-François Hélard, Professeur à l'Institut National des Sciences Appliquées, pour avoir pris le temps de rapporter sur les résultats présentés dans ce manuscrit et R. Michael Tanner, Doyen de l'Université d'Illinois à Chicago, pour avoir accepté d'apporter un avis extérieur à la communauté scientifique française. Je remercie également Éric Martin, Professeur à l'Université de Bretagne Sud, pour avoir présidé le jury de ma soutenance, ainsi qu'Hikmet Sari, Professeur à l'École Supérieure d'Électricité, pour avoir accepté de faire partie du jury malgré la proximité d'ICC 2004. Et bien sûr, un grand merci à Claude Berrou, Directeur d'Études à l'ENST Bretagne, qui m'a formée à l'électronique il y a presque vingt ans déjà et m'a par la suite entraînée dans la passionnante aventure des turbocodes.

J'adresse également mes sincères remerciements à l'ensemble des membres de la Direction de l'ENST Bretagne pour m'avoir encouragée à me lancer dans la rédaction de ce manuscrit, et tout particulièrement à Alain Glavieux pour son travail minutieux de relecture.

Plus que tout autre, un travail d'habilitation résulte de tâches collectives, menées en partie par des doctorants et des stagiaires : j'adresse mes remerciements à l'ensemble des étudiants que j'ai accueilli en stage et en thèse. Ils ont joué un rôle important dans l'évolution de mes compétences scientifiques.

Un grand merci également à Michel Jézéquel, responsable du département Électronique de l'ENST Bretagne, pour m'avoir permis d'effectuer mes travaux de recherche dans les meilleures conditions. La bonne ambiance qui règne au département Électronique de l'ENST Bretagne contribue grandement à ces bonnes conditions. Je remercie donc tous mes collègues, anciens et nouveaux, avec qui c'est quotidiennement un plaisir de travailler aussi bien en recherche qu'en enseignement.

Introduction 3

1 *Curriculum vitæ* 5
1.1 Informations personnelles 5
1.2 Formation et diplômes 5
1.3 Parcours professionnel 5
1.4 Activités d'enseignement 5
1.5 Activités de recherche 7
1.6 Publications 10

2 Travaux de thèse (1988 – 1992) 15
2.1 Contexte de l'étude 15
2.2 Conception d'un compilateur de machines séquentielles autotestables pour circuits intégrés spécifiques 15

3 Premiers travaux sur les circuits de turbocodage et décodage (1993 – 1994) 17
3.1 Introduction au codage concaténé et aux turbocodes 17
3.2 Synchronisation et supervision du circuit TURBO3 20
3.3 Modélisation du circuit TURBO4 22
3.4 Evolution de mes travaux sur les turbocodes à partir de 1995 22
3.5 Références 23

4 Participation à l'axe de recherche "architecture de processeurs" du département Electronique (1995 – 2000) 25
4.1 Contexte 25
4.2 Thèse de Frédéric Le Roy 25
4.3 Références 26

5 Les turbocodes et les techniques de décodage itératif, suite (1995 – …) 27
5.1 Turbo-détection (1995) 27
5.2 Turbocodes convolutifs pour blocs courts (depuis 1996) 29
5.3 Estimation des performances des turbocodes (2001) 44
5.4 Turbocodes et accès multiple à répartition par entrelacement (depuis 2003) 47
5.5 Conclusion 48
5.6 Références 49

6 Les modulations turbocodées (depuis 1999) 53
6.1 Introduction 53
6.2 Schéma de principe de l'approche pragmatique des modulations turbocodées 54
6.3 Étude du *mapping* optimal des turbocodes sur des constellations à grand nombre d'états : contrat de recherche externe France Telecom [JEZ] (2001-2002) 57
6.4 Projet MHOMS : MOdems for High-Order Modulation Schemes [BER2-3] (avril 2003 à janvier 2004) 62
6.5 Estimation des performances des modulations turbocodées pragmatiques (2002 - …) 65
6.6 Conclusions, travaux en cours et perspectives 69
6.7 Références 69

7 Conclusion et perspectives 73

Annexe 1 75
Annexe 2 85
Annexe 3 89
Annexe 4 95
Annexe 5 103
Annexe 6 113
Annexe 7 135
Annexe 8 141
Annexe 9 159

Introduction

Le premier chapitre de ce manuscrit est constitué d'un *curriculum vitæ*, destiné à synthétiser en quelques pages l'ensemble de mes activités, aussi bien en enseignement qu'en recherche.

Ce document vise ensuite à décrire l'ensemble de mes travaux de recherche et développement, depuis ma thèse de Doctorat, démarrée en 1988, jusqu'à mes travaux en cours. La structuration en chapitres a pour but de faire ressortir à la fois les différentes thématiques abordées et leur ordre chronologique de déroulement.

Mes travaux de thèse sont décrits dans le chapitre 2. Mon laboratoire d'accueil à l'ENST Bretagne était alors spécialisé dans la conception de circuits intégrés spécifiques dédiés aux transmissions numériques. Les circuits étant testés fonctionnellement par nos soins à leur retour de fonderie, la nécessité de l'insertion dans les circuits d'un mécanisme de test automatisé ou semi-automatisé a motivé l'orientation de mon travail de thèse vers la mise au point d'une méthode de génération de machines séquentielles autotestables.

En 1990, l'invention des turbocodes par deux enseignants-chercheurs de l'ENST Bretagne, Claude Berrou et Alain Glavieux, a quelque peu bouleversé les activités de l'équipe. Dans un premier temps, de 1990 à 1994, elle a essentiellement porté ses efforts sur le problème de la réalisation des circuits de turbo-décodage, pour prouver à la communauté scientifique la validité de la technique "turbo" ainsi que sa faisabilité. A partir de 1992, après mon recrutement au département Electronique de l'ENST Bretagne, j'ai donc essentiellement participé à la conception des tous premiers circuits de turbo-décodage, ainsi que je le décris dans le chapitre 3.

En 1995, le département Electronique a entamé une collaboration avec le Centre Norbert Segard (CNS) du CNET[1] à Grenoble. Celle-ci avait pour but la définition d'une architecture de microcontrôleur adaptée aux besoins futurs en télécommunications, pour des applications liées à l'interfaçage du réseau de transport à haut débit ATM (*Asynchronous Transfer Mode*). Cette étude reposait sur les compétences du CNS dans le domaine des méthodes et des outils de développement de processeurs de traitement de signal dédiés aux applications de télécommunications. J'ai alors pris en charge cette activité, qui n'a, hélas, vécu que très peu de temps en raison de la complète restructuration des activités du CNS, un peu plus d'un an plus tard. Le bilan de ces travaux se résume à une thèse, dont les résultats sont succinctement décrits dans le chapitre 4.

J'ai ensuite recentré mes activités sur l'axe de recherche principal du département Electronique, traitant des turbocodes et du décodage itératif. Je me suis toutefois assez rapidement éloignée des problématiques exclusivement matérielles pour évoluer vers le codage de canal et d'autres fonctions des communications numériques : égalisation, modulation, détection multi-utilisateur ("les turbo-communications"). Néanmoins, l'objectif final, lors de la mise au point de fonctions innovantes dans les systèmes de transmission, reste leur mise en œuvre matérielle. Je décris dans le chapitre 5 mes travaux sur l'application du traitement itératif au problème de la suppression de l'interférence entre symboles sur les canaux de transmission (la "turbo-égalisation"), puis sur le turbocodage et décodage appliqué aux transmissions par blocs. En effet, la plupart des applications de télécommunications (communications radio-mobiles, sans fil, …) reposent sur la transmission de blocs de données, de taille parfois réduite (jusqu'à quelques dizaines de bits). Les travaux en cours sur les systèmes de détection multi-utilisateurs sont finalement abordés.

---

[1] Actuellement France Telecom R&D

Depuis 1999, une part importante de mes travaux porte sur l'étude de l'association efficace des turbocodes avec des modulations numériques d'ordre élevé, dans le but de concevoir des systèmes de transmission à forte efficacité spectrale de très haute fiabilité. J'y ai consacré le chapitre 6, avant de conclure sur les perspectives que j'entrevois actuellement pour mes travaux de recherche au chapitre 7.

J'attire l'attention du lecteur sur le fait que les aspects techniques ne sont la plupart du temps pas abordés dans le détail. Seuls le contexte et les résultats obtenus sont décrits. Les détails techniques pourront être consultés dans les articles référencés, dont les principaux sont reproduits en annexe.

1 *Curriculum vitæ*

1.1 Informations personnelles

Catherine DOUILLARD
Née le 13 juillet 1965 à Fontenay le Comte, Vendée.
Nationalité française.

1.2 Formation et diplômes

1992 Obtention du **doctorat de l'Université de Bretagne Occidentale** (Brest), spécialité Electronique

"Conception d'un compilateur de machines séquentielles autotestables pour circuits intégrés spécifiques"

1988 Obtention du **diplôme d'Ingénieur de l'Ecole Nationale Supérieure des Télécommunications de Bretagne** (ENST Bretagne)

1982 Baccalauréat série C
Lycée Georges Clemenceau, Chantonnay (Vendée)

1.3 Parcours professionnel

Depuis 1994 **Maître de Conférences** à l'Ecole Nationale Supérieure des Télécommunications de Bretagne (ENST Bretagne), département Electronique.

1991-1994 **Chargée d'enseignement-recherche** à l'ENST Bretagne, département Electronique et Physique.

1.4 Activités d'enseignement

Charge annuelle moyenne d'enseignement direct : 150 heures (cours : 80 heures, travaux dirigés : 50 heures, travaux pratiques : 20 heures)

1.4.1 Matières enseignées

- Electronique numérique
- Conception et test des circuits intégrés numériques, modélisation VHDL
- Codage correcteur d'erreurs (turbocodes)

1.4.2 Cursus concernés

- 1ère, 2ème et 3ème année de la formation d'ingénieur de l'ENST Bretagne. Certains enseignements de l'option de 3ème année "Circuits Intégrés et Systèmes de Télécommunications" comptent également pour le DEA d'Electronique commun à l'Université de Bretagne Sud et l'ENST Bretagne.

- *Master of Science* “Digital Integrated Circuit Design for Telecommunications” de l’ENST Bretagne.

- DEA francophone “Traitement du signal” de l’Université Polytechnique de Timisoara, Roumanie.

- Formation continue de l’ENST Bretagne

1.4.3 Responsabilités d’enseignement

- Depuis 1993, responsable du module “*Simulation et synthèse logique des systèmes numériques*” (30 heures) dans l’option de 3ème année CIST (Circuits Intégrés et Systèmes de Télécoms) de l’ENST Bretagne.

- De 1994 à 1999, responsable du module optionnel de 2ème année de l’ENST Bretagne “*VHDL : le langage des concepteurs de circuits intégrés*” (30 heures).

- Depuis 1995, responsable du module “*Electronique numérique*” en 1ère année de l’ENST Bretagne (50 heures).

- De 1997 à 2001, responsable du module optionnel de 3ème année de l’ENST Bretagne “*Conception de circuits numériques : de la modélisation comportementale à la synthèse logique*” (30 heures).

- Depuis 1997, responsable du stage de formation continue de l’ENST Bretagne “*VHDL : langage, modélisation et synthèse logique*” (2×5 jours).

- Depuis 2003, responsable du stage de formation continue de l’ENST Bretagne “*VHDL-AMS : langage et modélisation des circuits mixtes* ” (5 jours).

- Depuis 2002, responsable du module optionnel de 3ème année de l’ENST Bretagne “*Initiation à la conception des circuits numériques*” (60 heures).

- Depuis 2003, responsable du module “*Digital Electronics*” du Master of Science de l’ENST Bretagne “Digital Integrated Circuit Design for Telecommunications” (50 heures).

1.4.4 Supports de cours

1. C. Douillard, “Le test des circuits intégrés logiques”, support de cours de l’ENST Bretagne, 47 pages, 1990.

2. C. Douillard, “Introduction à la modélisation des systèmes numériques à l’aide du langage VHDL”, support de cours de l’ENST Bretagne, 58 pages, 1992.

3. C. Douillard et A. Thépaut, “Electronique numérique (1) : logique combinatoire – circuits MOS”, support de cours de l’ENST Bretagne, 129 pages, 1995.

4. C. Douillard, G. Ouvradou et M. Jézéquel, “Electronique numérique (2) : logique séquentielle – techniques d’intégration”, support de cours de l’ENST Bretagne, 122 pages, 1996.

1.5 Activités de recherche

1.5.1 Principaux thèmes étudiés

- Le test intégré des circuits intégrés numériques et son automatisation (1988 – 1993).
- Processeurs de contrôle pour applications de télécommunications à haut débit (1995 – 2000).
- **Turbocodes et techniques itératives** (1993 – 2003) : circuits de turbo-codage/décodage, adaptation du principe turbo à la correction itérative de l'interférence entre symboles, définition de turbocodes performants pour le codage de paquets de données, estimation des performances asymptotiques des turbocodes, association des turbocodes à des modulations à grand nombre d'états, turbocodes et accès multiple.

1.5.2 Encadrement de jeunes chercheurs

1. O. Raoul, "Mise au point d'un compilateur de machines séquentielles auto-testables pour circuits intégrés spécifiques", stage de DEA, Université de Bretagne Occidentale, juillet 1993.
2. R. Schmitt, "Egalisation à sortie pondérée", stage de fin d'études, Université de Kaiserslautern, Allemagne, avril 1994.
3. F. Le Roy, "Microcontrôleur spécifique à haut débit", stage de DEA, Université de Bretagne Occidentale, juillet 1995.
4. E. Maury, "Evaluation des performances de l'association des turbocodes et des modulations à grand nombre d'états", stage de DEA, Université de Bretagne Occidentale, juillet 2002.
5. C. Abdel Nour, "Turbocodes convolutifs et transmissions à hautes efficacités spectrales : étude de l'apport d'un dispositif de turbo démodulation", stage de DEA, Université de Valenciennes, juillet 2003.

1.5.3 Encadrement de thèses

1. Thèse de Frédéric LE ROY,
   - titre : *Proposition d'une méthode de synthèse architecturale de circuits intégrés asynchrones adaptés aux applications à haut débit,*
   - université : Université de Bretagne Occidentale,
   - soutenue le 14 juin 2000,
   - directeur de thèse : Eric MARTIN, Université de Bretagne Sud,
   - taux d'encadrement : 70 %.

Depuis octobre 2000, Frédéric Le Roy est enseignant-chercheur à l'Institut Supérieur de l'Electronique et du Numérique de Brest.

2. Thèse de Fathi RAOUAFI,
   - titre : *Adéquation turbocodes/processeurs de signal*,
   - université : Université de Bretagne Occidentale,
   - soutenue le 13 juillet 2001,
   - directeur de thèse : Léon-Claude CALVEZ, Université de Bretagne Occidentale,
   - taux d'encadrement : 50 %.

En septembre 2001, Fathi Raouafi a été recruté en tant qu'ingénieur de recherche et développement chez IPSIS, une société de la région rennaise, prestataire en études, expertises et prototypage dans des secteurs d'activité très divers tels que les radiocommunications, les radars, l'optique, le spatial, l'énergie, la mécanique, le transport. Fathi a effectué sa première mission chez Philips Semiconductors à Caen.

3. Thèse de Laura CONDE CANENCIA,
   - titre : *Turbocodes et modulations à grande efficacité spectrale*,
   - université : Université de Bretagne Occidentale,
   - soutenance prévue en juin 2004.
   - directeur de thèse : Alain GLAVIEUX, ENST Bretagne,
   - taux d'encadrement : 80 %.

4. Thèse d'Emeric MAURY,
   - titre : *Codage correcteur d'erreurs : idéalité et réalité*,
   - université : Université de Bretagne Occidentale,
   - soutenance prévue en septembre 2005.
   - directeur de thèse : Claude BERROU, ENST Bretagne,
   - taux d'encadrement : 70 %.

5. Thèse de Charbel ABDEL NOUR
   - titre : *Turbocodes et canaux non gaussiens,*
   - université : Université de Bretagne Sud (co-délivrance ENST Bretagne),
   - thèse démarrée en octobre 2003.
   - co-directeurs de thèse : Claude BERROU, ENST Bretagne et Emmanuel BOUTILLON, Université de Bretagne Sud ,
   - taux d'encadrement : 50 %.

6. Thèse de Nadia Ben Lakhal (à venir)
   - titre : *Turbocodes et systèmes Ultra Wide-Band,*
   - université : Université de Bretagne Sud (co-délivrance ENST Bretagne),
   - démarrage de la thèse prévu début juin 2004.

1.5.4 Participation à des contrats de recherches

Depuis 1994, participation à 12 contrats de recherche dont :

Turbo-égalisation : principes et performances. Contrat de recherche CCETT[1] (12 mois), 1994-95. **Réalisation et rédaction de la moitié du contrat et rédaction du brevet d'invention associé**.

Définition d'un microcontrôleur rapide pour applications télécom, Contrat de recherche CNET (36 mois), 1995-98. **Responsable du contrat**.

Turbocodes pour données transmises par salves (turbocodes convolutifs pour blocs courts). Contrat de recherche CCETT (12 mois), 1996-97. **Mise en œuvre de l'ensemble des algorithmes de décodage proposés dans le cadre de ce contrat.**

Etude de complexité des turbocodes pour distribution intérieure radio – lot 1: étude des performances – lot2: étude de complexité. Contrat de recherche CNET (12 mois), 1998-1999. **Responsable de la moitié du contrat (50%) traitant des turbocodes convolutifs**.

Turbo coding for SkyPlexNet. Contrat d'étude avec Alenia Spazio (6 mois), 1999-2000. **Mise en œuvre de l'ensemble des algorithmes de décodage proposés dans le cadre de ce contrat.**

Etude du mapping optimal des turbocodes sur des constellations à grand nombre d'états. Contrat de recherche externe France Telecom R&D (24 mois), 2001-03. **Responsable de la part du contrat (75%) concernant les turbocodes convolutifs.**

Study of coding schemes based on parallel concatenation of convolutional codes (PCCC), in association with 4-ary, 8-ary, 16-ary, and 64-ary modulations for the MHOMS (Modems for high-order modulation schemes) project. Contrat d'étude ESA/Alenia Spazio (10 mois), 2003-2004. **Co-responsable du contrat**.

Validation des technologies UWB (Ultra WideBand) dans les applications de télécommunications domestiques à haut débit : . Projet de recherche du Groupe des Écoles des Télécommunications (12 mois), 2004. **Responsable de la contribution** "*Turbocodes et systèmes UWB à modulation d'impulsion*".

1.5.5 Participation à des comités de lecture sur le thème des turbocodes et du décodage itératif

1 Conférences internationales (en moyenne 5 à 6 par an)

- *IEEE International Conference on Communications (ICC)*
- *IEEE Global Communications Conference (Globecom)*
- *International Symposium on Turbocodes & Related Topics*

2 Revues internationales (depuis 1998)

- *IEEE Transactions on Communications* (3 articles)
- *IEEE Communication Letters* (7 articles)

---

[1] Actuellement France Telecom R&D, Rennes.

- *IEE Proceedings on Communications* (5 articles)
- *European Transactions on Telecommunications* (2 articles)
- *Annales des Télécommunications* (4 articles)
- *Electronics Letters* (*2 articles*)
- *EURASIP Journal on Applied Signal Processing (1 article)*

1.5.6 Responsabilités, rayonnement

- Responsable du projet de recherche "Turbo-Algorithmes et Circuits" du Groupe des Ecoles des Télécommunications : ce projet regroupe 6 enseignants-chercheurs, 1 chargé de recherche CNRS et 7 doctorants.
- Membre du réseau d'excellence européen NEWCOM "Network of Excellence in Wireless COMmunications", participant aux groupes de travail "Coding and Decoding for Short Block Lengths", "Adaptive coding and modulation" et "Bandwith-efficient coding".
- Membre du comité d'organisation de l'"*International Symposium on Turbocodes & Related Topics*" en 2000 et 2003.
- Membre du comité de programme de l'"*International Symposium on Turbocodes & Related Topics*" en 2000 et 2003 et 2006.
- Chairman de deux sessions de conférences du "*3rd International Symposium on Turbocodes & Related Topics*" en septembre 2003.
- Membre de l'*IEEE Communications Society* et de l'*IEEE Information Theory Society.*
- Membre du GDR ISIS (Information, Signal, Images et viSion).

1.6 Publications

1.6.1 Thèse

[1] **C. Douillard**, *Conception d'un compilateur de machines auto-testables pour circuits intégrés spécifiques*, thèse de l'Université de Bretagne Occidentale, soutenue le 8 septembre 1992. Jury : L.-C. Calvez, L. Devos, B. Courtois, C. Berrou, S. Toutain, P. Frison.

1.6.2 Participation à des ouvrages

[2] C. Berrou, **C. Douillard**, M. Jézéquel et A. Picart, "Les turbocodes," chapitre de l'ouvrage *Codage multimédia* du traité scientifique *IC2 : Information – Commande – Communication*, à paraître aux éditions Hermès Science en 2004.

[3] A. Picart, C. Laot et **C. Douillard**, "Turbo-égalisation et turbo-détection," chapitre de l'ouvrage *Signal et Télécommunications*, à paraître aux éditions Hermès en 2004.

En prévision :

[4] Co-auteur de l'ouvrage collectif *Codes et turbocodes*, à paraître aux éditions Springer en 2005.

1.6.3 Revues internationales avec comité de lecture

[5] C. Berrou et **C. Douillard**, "Générateur de machines séquentielles autotestables pour circuits intégrés spécifiques,"*Annales des Télécommunications*, Vol. 45, n° 11-12, Nov-Déc. 1990, pp. 635-641.

[6] C. Berrou and **C. Douillard**, "Pseudo-syndrome method for supervising Viterbi decoders at any coding rate," *Electronic Letters,* 1994, Vol. 30, n° 13, pp. 1036-1037.

[7] **C. Douillard**, A. Picart, P. Didier, M. Jézéquel, C. Berrou and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Transactions on Telecommunications*, Vol. 6, n° 5, Sept.- Oct. 1995, pp. 507-512.

[8] C. Berrou, **C. Douillard** and M. Jézéquel, "Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes," *Annales des Télécomunications*, Vol. 54, n° 3 – 4, Mars-Avril 1999, pp. 166 – 172.

[9] L. Conde Canencia, **C. Douillard**, M. Jézéquel and C. Berrou, "Application of the error impulse method to 16-QAM bit-interleaved turbo coded modulations," *Electronics Letters*, Vol. 39, March 2003, pp. 538 – 539.

Papiers soumis :

[10] E. Maury and **C. Douillard**, "Fast numerical evaluation of the theoretical limits for coding on the AWGN channel," soumis à Electronics Letters.

[11] **C. Douillard** and C. Berrou, "Turbo codes with rate-$m/(m+1)$ constituent convolutional codes," à paraître dans *IEEE Transactions on Communications.*

1.6.4 Congrès internationaux avec comité de lecture et actes

[12] **C. Douillard**, P. Ferry, P. Adde and M. Jézéquel, "The introduction of VHDL in digital design practical courses," *1st European Workshop on Microelectronics Education*, Villard de Lans, France, Feb. 1996. pp. 189 – 192.

[13] M. Jézéquel, C. Berrou, **C. Douillard** and P. Pénard, "Characteristics of a sixteen-state turbo-encoder/decoder (turbo4)," *International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 1997, pp. 280 – 283.

[14] C. Berrou, M. Jézéquel and **C. Douillard**, "Multidimensional turbo codes," *Information Theory Workshop ITW'1999*, Metsovo, Greece, June 1999, p. 27.

[15] C. Berrou, **C. Douillard** and M. Jézéquel, "Designing turbo codes for low error rates," *IEE Colloquium on Turbo codes in Digital Broadcasting – Could It Double Capacity?*, London, United Kingdom, Nov. 1999, pp. 1/1 – 1/7.

[16] F. Raouafi, **C. Douillard** and C. Berrou, "Efficient turbo decoder design and its implementation on a low-Cost, 16-bit fixed-point DSP," *2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2000, pp. 339 – 342.

[17] **C. Douillard**, M. Jézéquel, C. Berrou, N. Brengarth, J. Tousch and N. Pham, "The Turbo code Standard for DVB-RCS," *2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2000, pp. 535 – 538.

[18] C. Berrou, M. Jézéquel, **C. Douillard** and S. Kerouédan, "The advantages of non-binary turbo codes," *Information Theory Workshop ITW'2001*, Cairns, Australia, Sept. 2001, pp. 61 – 63.

[19] C. Berrou, M. Jézéquel, **C. Douillard**, S. Kerouédan and L. Conde Canencia, "Duo-binary turbo codes associated with high-order modulations," *2nd ESA Workshop on Tracking Telemetry and Command Systems for Space Applications, TTC'2001*, Noordwijk, the Netherlands, Oct. 2001.

[20] C. Berrou, **C. Douillard** and M. Jézéquel, "Application of the error impulse method in the design of high-order turbo coded modulations," *Information Theory Workshop ITW'2002*, Bangalore, India, Oct. 2002, pp.41 – 44.

[21] C. Berrou, S. Vaton, M. Jézéquel and **C. Douillard**, "Computing the minimum distance of linear codes by the error impulse method," *IEEE Global Communications Conference, Globecom'2002*, Taipei, Taiwan, Nov. 2002.

[22] L. Conde Canencia and **C. Douillard**, "Performance estimation of 8-PSK turbo coded modulation over Rayleigh fading channels," *3rd International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2003.

[23] C. Berrou, R. De Gaudenzi, **C. Douillard**, G. Gallinaro, R. Garello, D. Giancristofaro, A. Ginesi, M. Luise, G. Montorsi, R. Novello and A. Vernucci, " High speed modem concepts and demonstrator for adaptive coding and modulation with high order in satellite applications," *8th International Workshop on Signal Processing for Space Communications*, *SPSC 2003*, Catania, Italy, Sept. 2003.

[24] C. Berrou, Y. Saouter, **C. Douillard**, S. Kerouédan and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model," *to appear in proceedings of International Conference on Communications, ICC'2004*, Paris, France, June 2004.

[25] L. Conde Canencia and **C. Douillard**, "A new methodology to estimate asymptotic performance of turbocoded modulation over fading channels," *2nd International Symposium on Image/Video Communications over fixed and mobile networks*, Brest, France, July 2004.

Conférence invitée :

[26] **C. Douillard**, "A Pragmatic approach of turbo-coded modulations for transmissions with high spectral efficiency," *International Symposium on Image/Video Communications over fixed and mobile networks*, Brest, France, July 2004.

1.6.5 Congrès francophones avec comité de lecture

[27] C. Berrou et **C. Douillard**, "Générateur de machines séquentielles autotestables pour circuits intégrés spécifiques," *7ème Colloque International de Fiabilité et de Maintenabilité, λμ7*, Brest, Juin 1990, pp. 483-488.

[28] P. Didier, A. Picart, **C. Douillard** et M. Jézéquel, "Application des techniques de décodage itératif à la correction de l'interférence entre symboles," *15ème colloque du GRETSI,* Juan Les Pins, 1995, pp. 549-552.

[29] F. Le Roy, **C. Douillard**, B. Prou, "Une méthode de partitionnement de programmes CHP pour la synthèse architecturale," *Journées thématiques Universités/Industries sur l'adéquation Algorithme-Architecture pour les Applications Temps Réel Industrielles Complexes*, Lille, 1999.

1.6.6 Brevets d'invention

[30] **C. Douillard**, A. Glavieux, M. Jézéquel et C. Berrou, "Dispositif de réception de signaux numériques à structure itérative, module et procédé correspondants", Brevet n° 95 01603 du 07/02/1995, France. Etendu en Europe et aux USA en 1996.

[31] C. Berrou, M. Jézéquel et **C. Douillard**, "Procédé de qualification de codes correcteurs d'erreurs, procédé d'optimisation, codeur, décodeur et application correspondants", Brevet n° 01 11764 du 11/09/2001, France.

1.6.7 Rapports d'études (sélection)

[32] **C. Douillard**, A. Picart, A. Glavieux et P. Didier, *Turbo-égalisation : principes et performances*, Rapport de contrat de recherche CCETT, octobre 1995 (66 pages).

[33] C. Berrou, **C. Douillard**, M. Jézéquel, *Turbocodes pour données transmises par salves (turbocodes convolutif pour blocs courts)*, Rapport de contrat de recherche CCETT, novembre 1997 (60 pages).

[34] A. Picart, **C. Douillard**, P. Adde et M. Jézéquel, *Etude de complexité des turbocodes pour distribution intérieure radio – lot 1: étude des performances – lot2: étude de complexité*, Rapport de contrat de recherche CNET, août 1999 (121 pages).

[35] C. Berrou, **C. Douillard**, M. Jézéquel et S. Vaton, *Turbo coding for SkyPlexNet*, Rapport de contrat d'étude, décembre 2000 (37 pages).

[36] M. Jézéquel, **C. Douillard**, P. Adde, C. Berrou et L. Conde, *Etude du mapping optimal des turbocodes sur des constellations à grand nombre d'états*, Rapport de contrat de recherche externe, France Telecom R&D, avril 2003 (126 pages).

[37] C. Berrou and **C. Douillard**, *Flexible Turbo code for Low Error Rates,* Rapport de contrat ESA/Alenia Spazio dans le cadre du projet MHOMS: "Modem for High-Order Modulation Schemes", novembre 2003 (80 pages).

2 Travaux de thèse (1988 – 1992)

2.1 Contexte de l'étude

Depuis 1984, la principale activité du Laboratoire Circuits Intégrés Télécom de l'ENST Bretagne, devenu en 1994 le département Électronique, consistait à concevoir des circuits intégrés prédiffusés dédiés aux transmissions numériques : décodeur de Viterbi, contrôleur de liaison numérique par alternat, modulateur numérique universel, … La prévision du test fonctionnel du circuit à son retour de fonderie était une contrainte cruciale à prendre en compte au cours de la conception de ces circuits. A la suite d'un stage pendant lequel j'ai été amenée à développer une fonction permettant l'automatisation d'une partie du test fonctionnel d'un circuit contrôleur de liaison par alternat, j'ai démarré, en octobre 1988, une thèse sur le thème de l'automatisation du test des circuits intégrés. Plus précisément, le travail réalisé a eu pour objet la génération automatique d'opérateurs de contrôle autotestables.

2.2 Conception d'un compilateur de machines séquentielles autotestables pour circuits intégrés spécifiques

La première partie de mon manuscrit de thèse présente un panorama des différentes méthodes de test des circuits intégrés recensées dans la littérature en s'attachant plus particulièrement aux techniques adaptées aux circuits à haut degré de séquentialité.

Le test dit *hors ligne* est appliqué en dehors du mode opératoire normal du circuit. Il est principalement destiné à éliminer les composants défectueux en fin de fabrication. La détection de pannes met alors en œuvre l'application de séquences logiques appropriées, les *vecteurs de test*, sur l'entrée du circuit sous test et vérifie la conformité des sorties correspondantes.

Le test dit *en ligne* vise une détection permanente et immédiate des pannes pendant le fonctionnement normal du circuit, notamment pour des applications à haute sûreté de fonctionnement. Parmi les méthodes existantes de conception orientées vers le test, celles qui dotent le circuit de propriétés d'autotestabilité sont particulièrement attractives : grâce à l'intégration d'une partie ou de la totalité du mécanisme de test dans le circuit considéré, la détection des erreurs peut être très rapide, voire immédiate. Une méthode d'autotest judicieusement élaborée peut ainsi permettre de couvrir à la fois les tests hors ligne et en ligne d'un même circuit.

La deuxième partie du document présente une méthode innovante de synthèse de machines séquentielles autotestables en ligne et hors ligne. Cette méthode est également décrite dans un article publié dans les Annales des Télécommunications en novembre 1990 et reproduit en Annexe 1 de ce document.

Des considérations de complexité matérielle et de limitation de la durée du test m'ont conduit à l'adoption d'un double test intégré originellement conçu pour le test en ligne.

Parmi les multiples façons d'aborder le problème de la synthèse des machines séquentielles, la microprogrammation en mémoire morte est une solution attractive lorsqu'est recherchée la simplicité de la synthèse systématique et du placement-routage. Certaines restrictions comportementales (limitation du nombre de transitions issues de chaque état) ont été adoptées afin de résoudre les problèmes de complexité matérielle inhérents à ce choix de réalisation.

La technique de test proposée fait appel à un codage en bloc détecteur, voire correcteur, d'erreurs de nature récurrente qui permet, à tout instant, de vérifier à la fois la cohérence des

données relatives à l'état présent de la machine et le séquencement des états. Le circuit obtenu résulte d'un compromis entre la fiabilité du mécanisme de détection d'erreurs, exprimée sous la forme d'une probabilité de non-détection d'erreur, et la taille de la mémoire nécessaire à sa réalisation. D'autre part, la logique ajoutée pour mettre en œuvre le mécanisme de test peut être elle-même conçue pour une détection automatique de ses propres pannes (structures dites *self-checking*), au prix bien sûr d'une complexité matérielle accrue. La méthode décrite permet ainsi de choisir le degré de testabilité en fonction de l'application et de la complexité maximale autorisée. Le taux de confiance du test est évalué pour différentes classes de défaillances matérielles réalistes.

La troisième et dernière partie de ma thèse traite de la mise en œuvre pratique de la méthode décrite précédemment, sous la forme d'un logiciel de génération automatique de machines séquentielles autotestables pouvant être intégré à une chaîne de compilation de silicium. L'exécution du programme de génération part d'une description de la machine séquentielle à synthétiser sous la forme d'un graphe d'état et délivre une description structurelle du circuit autotestable correspondant. Il effectue successivement les tâches suivantes :

- la vérification de la validité du graphe fourni, *G*, en tant que diagramme d'états d'une machine de Mealy ;
- la transformation du graphe *G*, si nécessaire, en un graphe *G'* prenant en compte les restrictions comportementales liées à l'application de la méthode de synthèse ;
- l'élaboration du contenu de la mémoire en fonction de la probabilité maximale de non détection d'erreurs visée ;
- la détermination de la taille des vecteurs d'entrée de chaque bloc fonctionnel de la machine séquentielle et la génération d'une description structurelle à partir d'une bibliothèque de composants génériques.

Cet outil a été testé sur quelques cas concrets de contrôleurs afin d'en valider le bon fonctionnement et de mesurer le niveau de testabilité obtenu à l'aide d'un simulateur de fautes.

La thèse a été soutenue en septembre 1992. L'étude a été ensuite poursuivie par Olivier Raoul dont j'ai supervisé le stage DEA en 1993. Il a étendu la méthode décrite dans ma thèse à une classe plus large de machines séquentielles, avec des restrictions comportementales moins strictes que dans l'étude initiale. Finalement, un paramètre d'entrée supplémentaire, le nombre maximal de transitions sortantes par état dans le graphe, a ainsi pu être ajouté à l'outil de génération automatique, afin de pouvoir prendre en compte un paramètre vitesse dans les performances du circuit résultant.

3 Premiers travaux sur les circuits de turbocodage et décodage (1993 – 1994)

3.1 Introduction au codage concaténé et aux turbocodes

Les turbocodes constituent une famille de codes correcteurs d'erreurs qui permettent d'avoisiner la limite théorique de correction prédite par Shannon il y a plus de 50 ans [SHA]. Ces codes, inventés à l'ENST Bretagne [BER1] au début des années 90, sont obtenus par la concaténation de deux ou plusieurs codes de faible complexité, séparés par une fonction d'entrelacement introduisant de la diversité. Leur décodage fait appel à un processus itératif (ou *turbo*) utilisant deux ou plusieurs décodeurs élémentaires qui s'échangent des informations de fiabilité, appelées *informations extrinsèques*, afin d'améliorer la correction au fil des itérations.

L'idée de concaténer plusieurs codes simples pour obtenir un code composite puissant, tout en conservant une complexité de décodage raisonnable, remonte aux années 60 avec le travail de thèse de Forney [FOR]. Cette approche de codage *a priori* très prometteuse est illustrée par le schéma de la Figure 3.1. La présence du désentrelaceur entre les décodeurs intérieur et extérieur, et par conséquent de l'entrelaceur entre les codeurs extérieur et intérieur, a pour but de casser les paquets d'erreurs issus du décodage intérieur afin de faciliter le travail du décodeur extérieur.

Les performances du codage concaténé restèrent toutefois médiocres jusqu'à ce que Battail [BAT], Hagenauer et Hoeher [HAG] proposent vers la fin des années 80 un algorithme de décodage des codes convolutifs à sorties pondérées appelé SOVA (*Soft-Output Viterbi Algorithm*). Ainsi, dans le cas où le code intérieur est un code convolutif, les décisions fournies par le décodeur 2 sont assorties d'une indication de fiabilité qui permettent au décodeur 1 d'augmenter le gain de codage global d'environ 1,5 dB.

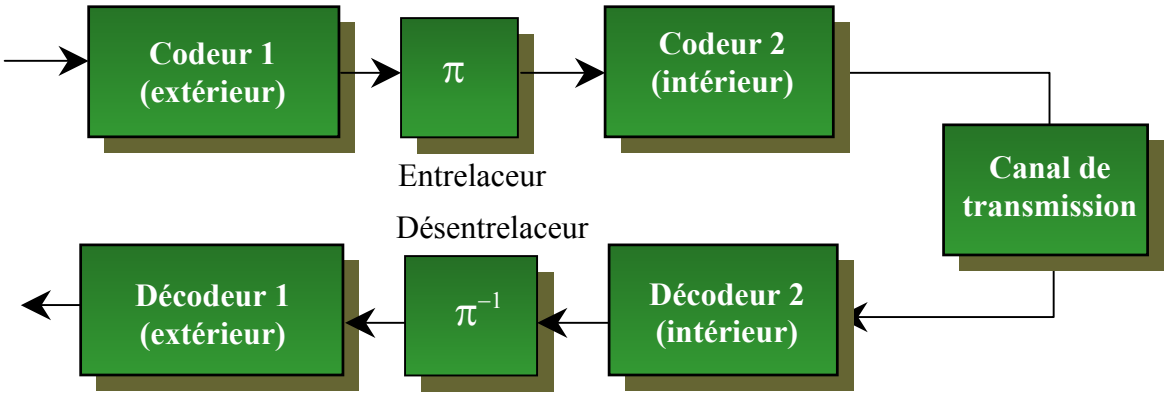


Figure 3.1 : Principe de codage et de décodage d'un code concaténé. La fonction de désentrelacement est destinée à casser les paquets d'erreurs en entrée du décodeur 2.

Mais l'avancée décisive qui a permis de tirer le meilleur parti de la puissance des codes concaténés est le décodage itératif ou décodage *turbo* : il est basé sur l'utilisation de décodeurs extérieur et intérieur à entrées et sorties pondérées ou SISO (*Soft-Input Soft-Output*) qui s'échangent des informations de fiabilité, appelées *informations extrinsèques*, par le biais d'une contre-réaction (Figure 3.2). Le traitement numérique de l'information ne permettant pas la mise en œuvre de la technique de contre-réaction, une procédure de décodage itérative doit, en pratique, être appliquée.

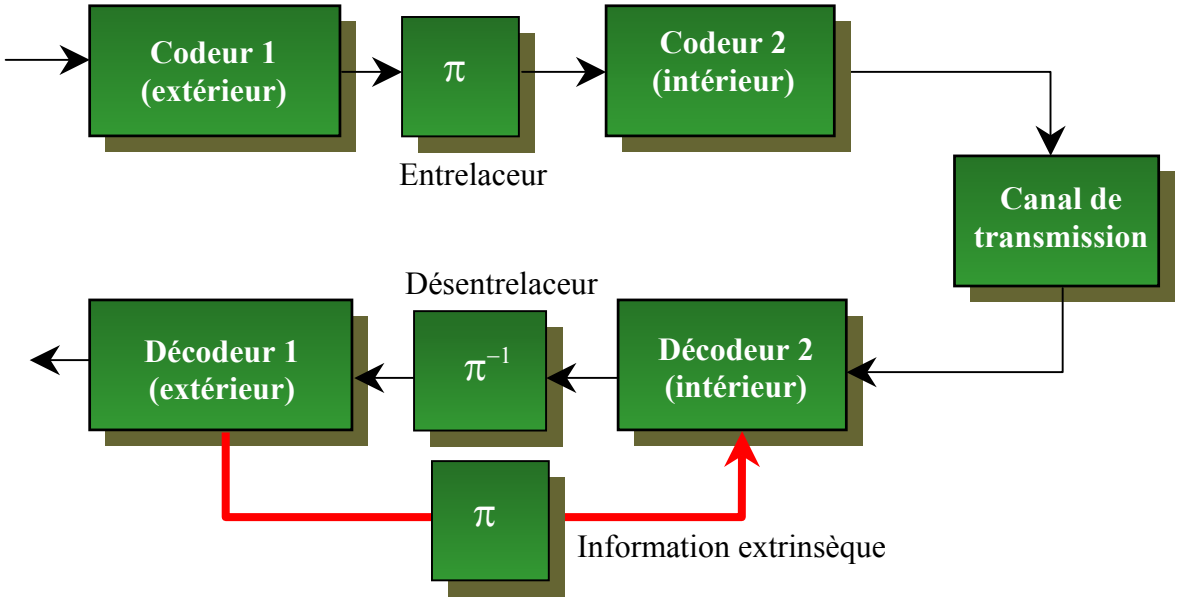


Figure 3.2 : Principe de codage et de décodage itératif d'un code concaténé.

Le premier turbocode proposé par Claude Berrou et Alain Glavieux en 1993 [BER1] fait appel à la concaténation parallèle de deux codes convolutifs systématiques récursifs de mémoire $\nu = 4$, séparés par un entrelaceur non-uniforme de taille 256×256 (cf. Figure 3.3).

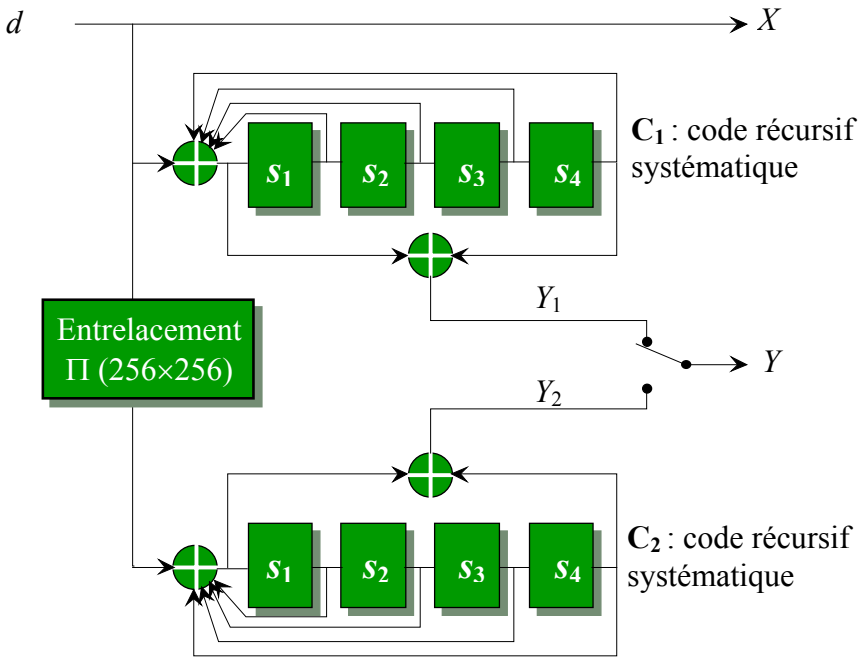


Figure 3.3 : Turbocode proposé dans [BER1]
Polynômes générateurs : 37, 21 (notation octale)

Le choix de codes élémentaires récursifs systématiques, basés sur des structures de registres à décalages rebouclés constituant des générateurs de séquences pseudo-aléatoires, au lieu de

codes non systématiques non récursifs (cf. Figure 3.4) classiquement utilisés jusqu'alors, se justifie par leur meilleur comportement à fort niveau de bruit [THI].

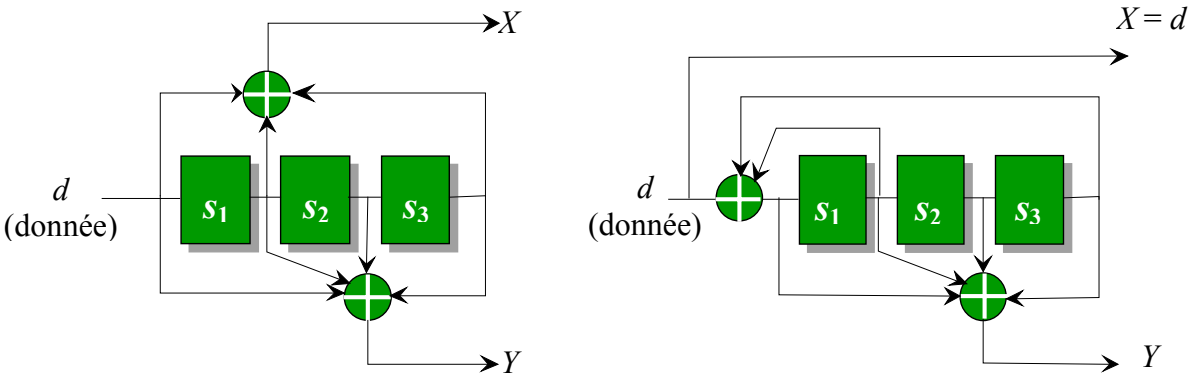


Figure 3.4 : (a) Code classique convolutif non récursif; (b) Code convolutif systématique récursif. Polynômes générateurs : 15, 17 (notation octale)

Le schéma de principe du turbo-décodage tel qu'il est décrit dans [BER1] est présenté en Figure 3.5. Le décodage SISO fait appel à l'algorithme MAP, encore appelé algorithme APP, *backward-forward* ou bien BCJR [BAH]. Les principales étapes du déroulement de cet algorithme sont décrites en Annexe 2. Dans le domaine logarithmique, l'information extrinsèque z produite par chaque décodeur élémentaire est obtenue par une simple soustraction entre la sortie du décodeur et son entrée.

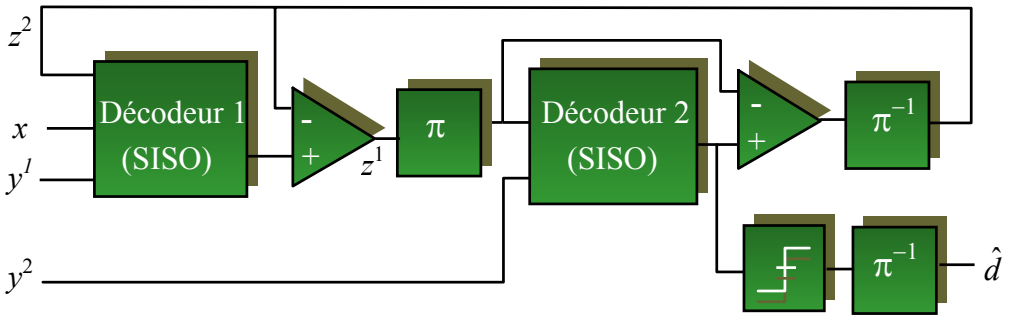


Figure 3.5 : Principe du turbo-décodage décrit dans [BER1]

Le principe du turbo-décodage, associé à la concaténation parallèle de la Figure 3.3 a ainsi permis d'atteindre des performances de correction jusqu'alors inégalées : à un taux d'erreurs binaires (TEB ou BER pour *Bit Error Rate*) de 10^{-5}, le processus de décodage itératif permet d'approcher la limite théorique de Shannon d'un canal à entrées binaires à 0,5 dB, après 18 itérations (Figure 3.6). Les progrès que nous avons faits depuis lors dans la sélection des codes élémentaires nous ont permis de gagner encore 0,15 dB (voir section 5.2.6).

En fait, par le passé d'autres chercheurs, notamment Tanner [TAN] et Gallager [GAL] aux Etats-Unis, avaient déjà imaginé des procédés de codage et de décodage précurseurs des turbocodes, sans toutefois obtenir des performances de correction comparables.

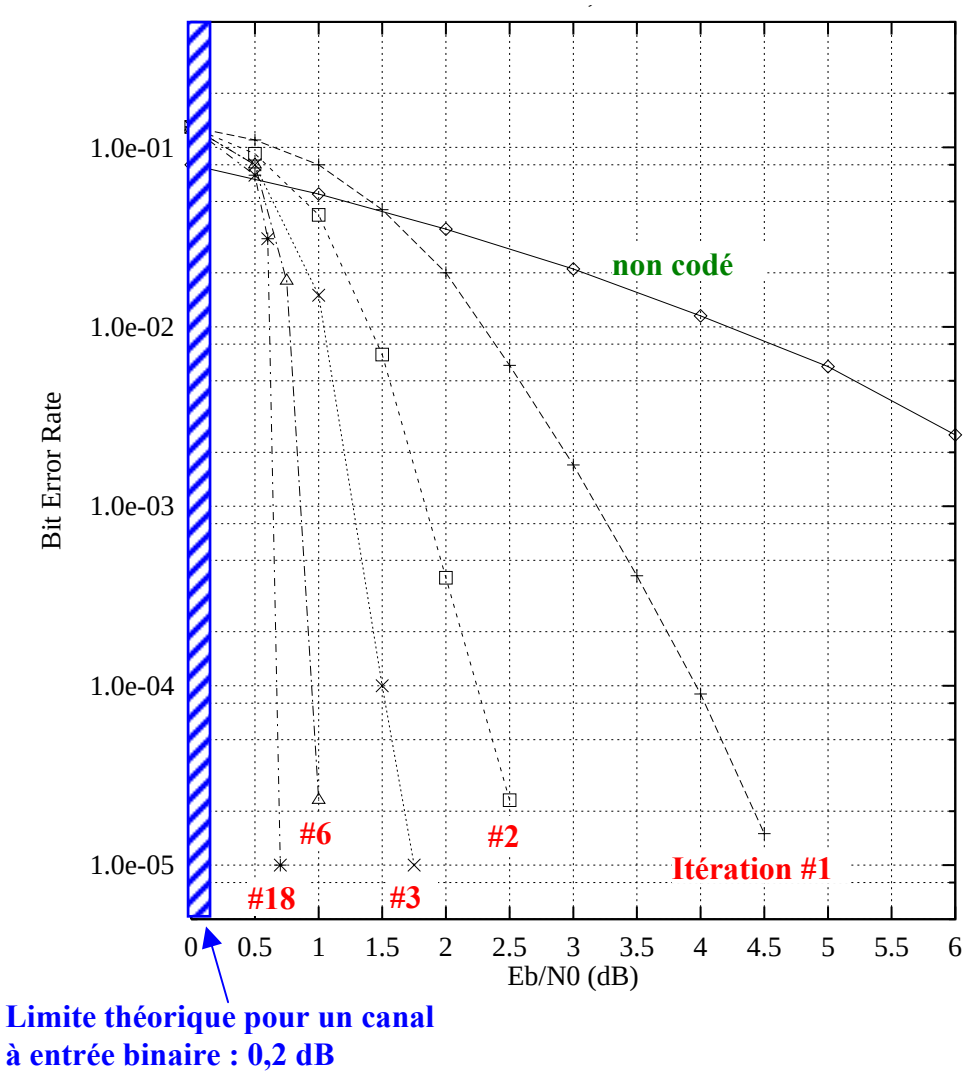


Figure 3.6 : Performance, en taux d'erreurs binaires (TEB), du turbocode présenté dans [BER1]. Canal gaussien, modulation MDP-4. Polynômes générateurs 37, 21 (notation octale). Rendement de codage *R* = 1/2. Entrelacement non uniforme 256×256. Décodage MAP.

Depuis les dépôts de brevets en 1991, puis la première publication en 1993, les turbocodes ont parcouru bien du chemin : ils sont maintenant normalisés dans un certain nombre de standards de télécommunications. Citons à titre d'exemples le standard CCSDS [CCSDS] pour les missions spatiales en espace lointain, le standard M4 d'Inmarsat, la transmission de données dans les réseaux de téléphonie mobile de 3ème génération [3GPP], les standards DVB-RCS [DVB1] et DVB-RCT [DVB2] pour les voies de retour de transmission de la télévision numérique par satellite et par voie terrestre. Les concepts d'"information extrinsèque", de "concaténation parallèle" et de "codes convolutifs systématiques récursifs" sont aujourd'hui intégrés dans le jargon de la théorie de l'information.

3.2 Synchronisation et supervision du circuit TURBO3

En octobre 1991, mon recrutement à l'ENST Bretagne en tant qu'enseignant-chercheur coïncide avec les premières études sur les turbocodes et le décodage itératif. Le premier brevet est alors en cours de dépôt. Dans l'année qui suit, le département Electronique décide de mettre la majeure partie de ses forces sur le problème de la réalisation de circuits de turbo-

codage et décodage, afin de montrer à l'ensemble de la communauté scientifique à la fois leur performance sans précédent et leur faisabilité. Aussi, mes premiers travaux concernant les turbocodes m'en ont d'abord procuré une vision orientée circuit puisque j'ai été amenée à participer au développement des deux premiers ASICs de turbocodage et décodage nommés TURBO3 et TURBO4.

Le premier turbocode à avoir fait l'objet d'une réalisation sous forme d'ASIC fait appel à la concaténation parallèle de deux codes convolutifs systématiques récursifs de mémoire $\nu = 3$, de polynômes générateurs 15 pour le récursivité et 17 pour la redondance (notation octale, voir Figure 3.4) séparés par une matrice d'entrelacement organisée en 32 lignes et 32 colonnes. Le rendement global du code vaut 1/2, les codes élémentaires ayant des rendements égaux à 4/7 pour le code 1 et 4/5 pour le code 2. Ces rendements sont obtenus par poinçonnage, c'est-à-dire par la non transmission, de certains bits de redondance. Le circuit fabriqué et commercialisé par la société Comatlas[1] en 1993 (circuit CAS5093) contient le codeur et 2,5 itérations de décodage[2].

Le principe de synchronisation mis en œuvre dans ce circuit utilise la redondance issue du premier codeur [FER]. Les bits de redondance issus de ce codeur sont périodiquement inversés en fonction d'un masque fourni par un fanion F. Si le $i^{\text{ème}}$ bit de F est égal à 1, le bit de redondance correspondant est inversé, sinon il est inchangé. Pendant la phase de synchronisation, les symboles servant à la synchronisation, ou supposés tels, sont effacés à l'entrée du décodeur 1, ceci entraînant une augmentation temporaire du rendement du code correspondant. L'estimation fournie par le décodeur 1 à la première itération est comparée avec les valeurs correspondantes en sortie du canal par une sommation modulo 2 bit à bit. Le résultat de la comparaison alimente un registre à décalage, dont le contenu est comparé au fanion F. Si le résultat de la comparaison permet la reconnaissance du fanion F, le décodeur est déclaré synchronisé, sinon la phase de recherche de synchronisation est poursuivie. Un certain nombre d'erreurs p est autorisé lors de la reconnaissance du fanion. La valeur de p retenue ($p = 14$ dans le cas présent) résulte d'un compromis consistant à minimiser le temps moyen de synchronisation, ainsi que probabilités de fausse synchronisation et de fausse alarme.

Le décodeur étant synchronisé, il passe alors en phase de supervision, dont le rôle est de repérer les pertes de synchronisation. Pour assurer cette fonctionnalité, nous avons proposé une méthode, dite du *pseudo-syndrome* [BER2]. Elle est basée sur la vérification d'une relation de parité, dépendant de la structure du treillis, qui relie l'état courant du codeur à l'instant i, les polynômes générateurs du code, le bit systématique X_i et le bit de redondance Y_i. Cette relation de parité est calculée à l'entrée du décodeur 2 et à la $1^{\text{ère}}$ itération, lorsque le bit de redondance correspondant est disponible. Pour une transmission sans erreur, tous les pseudo-syndromes PS_i sont nuls. En pratique, on calcule à l'aide d'un compteur, sur une fenêtre de travail de longueur L, le taux de pseudo-syndromes non nuls ($PS_i = 1$). Le décodeur est déclaré désynchronisé lorsque ce taux dépasse un certain seuil S. Le choix des valeurs de L et de S sont déterminées par simulation, de façon à assurer une détection fiable des pertes de synchronisation du décodeur.

Dans le cadre d'un contrat financé par la région Bretagne, j'ai développé un modèle détaillé, en langage VHDL, du dispositif complet de synchronisation et de supervision, qui a servi de référence à la société Comatlas pour la réalisation du circuit TURBO3 en technologie CMOS 0,8µm.

---

[1] Actuellement Philips Semiconductors

[2] Une itération de décodage correspond au traitement successif des données par les deux décodeurs élémentaires.

3.3 Modélisation du circuit TURBO4

Le second circuit de turbocodage et décodage réalisé, TURBO4 [JEZ], fait appel à la concaténation parallèle de deux codes systématiques récursifs de mémoire $\nu = 4$, de polynômes générateurs 23 (récursivité) et 35 (redondance), séparés par un entrelacement non uniforme de taille 64×32. Tous les bits de redondance sont fournis en sortie du codeur, autorisant ainsi des rendements de codage variables à partir de 1/3. Le principe de synchronisation et supervision, disponible pour le rendement 1/2 uniquement, est similaire à celui de TURBO3. La partie décodage du circuit est constituée de deux décodeurs élémentaires, mettant en oeuvre une version simplifiée de l'algorithme SOVA [BER3]. Chaque circuit permet de réaliser une itération de décodage, mais les circuits peuvent être cascadés, permettant ainsi à l'utilisateur de choisir le nombre d'itérations de décodage. La conception du circuit a été assurée par une équipe du CCETT[1] à Cesson-Sévigné, et sa fabrication en technologie CMOS 0,8µm puis 0,6µm et 0,35µm par ST Microelectronics. J'ai, en 1994, écrit un modèle simulable complet du circuit TURBO4 en langage VHDL, qui a servi de modèle de référence pour la conception de ce circuit. Le descriptif technique de ce circuit est détaillé dans l'article [JEZ], reproduit en Annexe 3 de ce document.

3.4 Evolution de mes travaux sur les turbocodes à partir de 1995

A partir de 1995, les travaux de recherche du département Électronique dans le domaine des turbocodes ont pris les orientations suivantes :

- la généralisation du principe de traitement itératif à des domaines connexes au codage de canal (les *turbo-communications*) ;
- l'étude de turbocodes adaptés au codage de données par blocs ;
- l'amélioration du pouvoir de correction des turbocodes par la recherche de schémas de codage plus performants, à faible taux d'erreurs notamment ;
- la conception de turbo-décodeur performants (simplification des algorithmes, turbo-décodeurs analogiques, …) ;
- l'association des turbocodes avec des modulations à grand nombre de points, pour la définition de systèmes de transmissions à grandes efficacités spectrales (modulations turbocodées) ;
- l'association des turbocodes avec des techniques de transmission à accès multiple (turbo-CDMA, turbo-IDMA).

Je me suis investie dans ces différentes thématiques à diverses périodes et à des degrés d'implication variables, tout aussi bien par le biais de participations à des contrats de recherches que de l'encadrement de thèses. Ces différents travaux sont décrits en détail dans les chapitres 6 et 7.

Toutefois, entre 1995 et 2000, j'ai également pris en charge une activité du département Electronique sur un thème fort éloigné des communications numériques, à savoir l'étude d'architectures de processeurs adaptées aux problèmes de contrôle dans les applications de télécommunications à haut débit. Les travaux réalisés sur ce thème font l'objet du chapitre suivant.

---

[1] Actuellement France Telecom R&D Rennes

3.5 Références

[3GPP] 3GPP Technical Specification Group, *Multiplexing and Channel Coding (FDD)*, TS 25.212 v.5.0.0, 2002-2003.

[BAT] G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi," *Annales des Télécommunications*, vol. 42, N°1-2, pp. 31-38, Jan. 1987.

[BER1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-corrrecting coding and decoding: turbo-codes," *International Conference on Communications*, *ICC'93*, Geneva, Switzerland, May 1993, pp. 1064-70.

[BER2] C. Berrou and C. Douillard, "Pseudo-syndrome method for supervising Viterbi decoders at any coding rate," *Electronics Letters*, June 1994, vol. 30, n° 13, pp. 1036-37.

[BER3] C. Berrou, P. Adde, E. Angui and S. Faudeil, "A low-complexity soft-output Viterbi decoder architecture," *International Conference on Communications*, *ICC'93*, Geneva, Switzerland, May 1993, pp. 737-740.

[CCSDS]Consultative Committee for Space Data Systems, *Recommandations for Space Data Systems. Telemetry Channel Coding*, Blue Book, May 1988.

[DVB1] DVB, "Interaction channel for satellite distribution systems," *ETSI EN 301 790, V1.2.2*, Dec. 2000, pp.21-24.

[DVB2] DVB, "Interaction channel for digital terrestrial television," *ETSI EN 301 958, V1.1.1*, Feb. 2002, pp.28-30.

[FER] P. Ferry, P. Adde and G. Graton, "Turbo-decoder synchronisation procedure. Application to the CAS5093 integrated circuit," *3rd International Conference on Electronics, Circuits, and Systems, ICECS'96*, Rhodos, Greece, Oct. 1996, pp. 168-171.

[FOR] G. D. Forney, Jr., *Concatenated codes*, MIT Press, 1966.

[GAL] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. IT-8, Jan. 1962, pp.21-28.

[HAG] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *IEEE Global Communications Conference, Globecom'89*, Dallas, Texas, Nov. 1989, pp. 4711-17

[JEZ] M. Jézéquel, C. Berrou, C. Douillard and P. Pénard, "Characteristics of a sixteen-state turbo-encoder/decoder (turbo4) ," *International Symposium on Turbo codes & Related Topics*, Brest, France, Sept. 1997, pp. 280-283.

[SHA] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, July and Oct. 1948.

[TAN] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. IT-27, Sept. 1981, pp. 533-547.

[THI] P. Thitimajshima, *Les codes convolutifs récursifs systématiques et leur application à la concaténation parallèle*, Thèse de Doctorat de l'Université de Bretagne Occidentale, Brest, France, Déc.1993.

4 Participation à l'axe de recherche "architecture de processeurs" du département Electronique (1995 – 2000)

4.1 Contexte

L'activité "processeurs de contrôle à haut débit" a démarré en mars 1995, dans le cadre d'une coopération entre le département Électronique et l'équipe du CNET de Grenoble[1] chargée du développement de méthodes et d'outils d'aide à la conception de circuits intégrés pour les télécommunications. Cette activité était destinée à assurer la continuité de l'axe de recherche du département Électronique précédemment intitulé "architectures de processeurs spécialisés" en le réorientant vers la définition d'architectures de processeurs paramétrables rapides adaptés aux problèmes de contrôle dans les applications de télécommunications à haut débit, dans les réseaux ATM (*Asynchronous Transfer Mode*) en particulier. Cette étude s'est, en pratique, concrétisée par la supervision du stage DEA de Frédéric Le Roy, puis du co-encadrement de sa thèse avec le département Informatique de l'ENST Bretagne. Les applications pratiques étudiées se rapportent au contrôle de congestion et la réservation de débit dans les commutateurs ATM [ITU].

4.2 Thèse de Frédéric Le Roy

Parmi les différentes cibles architecturales envisageables pour la réalisation de ces opérateurs de contrôle rapides, nous nous sommes orientés vers une solution de type asynchrone. Dans un circuit synchrone classique, un signal d'horloge cadence la mise à jour de l'état interne du circuit et de ses sorties. La présence d'une horloge commune dans un circuit complexe permet d'assurer la fiabilité de son fonctionnement ainsi que des communications entre ses composants. Supprimer cette horloge peut permettre néanmoins :

- d'éviter les problèmes de retard d'horloge dans les systèmes complexes,
- de supprimer la consommation en limitant la dissipation d'énergie liée à son activité continue,
- de limiter la pollution électromagnétique liée aux pointes de courant induites par les commutations simultanées des capacités,
- **mais aussi envisager d'augmenter la fréquence de travail si chaque bloc travaille à sa fréquence limite** et non à la fréquence limite globale de tout le système.

Aussi, le travail de thèse s'est-t-il finalement orienté vers l'élaboration d'une méthode de synthèse architecturale de contrôleurs asynchrones optimisés en vitesse.

Le cœur de cette étude a consisté à proposer une méthode permettant de transformer une description comportementale d'une application écrite en langage CHP (*Communicating Hardware Processes*) [MAR] et à en exploiter le parallélisme pour arriver à une description synthétisable sous la forme d'un circuit asynchrone quasi-insensible aux retards ou QDI (*Quasi Delay Insensitive*) et optimisé en vitesse. CHP est un langage adapté à la description des circuits asynchrones, qui fait appel aux notions de processus et de canaux de communications et permet de mettre en évidence le parallélisme de l'application traitée. Les transformations proposées font appel à des étapes successives de mise en parallèle, s'appuyant sur l'analyse des dépendances entre les variables de l'algorithme, puis de mise à

[1] Actuellement France Telecom R&D Grenoble

plat de l'algorithme décrivant l'application. Deux techniques originales de mise à plat ont été proposées dans la thèse. La première, dite "par extraction", est basée sur l'extraction formelle des structures de contrôle du programme CHP : elle met en œuvre des actions de développement, factorisation, réduction des structures de contrôle. A chaque transformation, la conformité de la dérivation par rapport à la spécification précédente est vérifiée formellement. Malheureusement, le choix et l'ordre des transformations appliquées influent fortement sur le résultat, et l'obtention d'un circuit performant repose essentiellement sur l'expertise du concepteur. La seconde technique de mise à plat proposée peut aisément, au contraire de la précédente, être automatisée et intégrée à un outil de synthèse. Elle fait appel à la construction puis à l'analyse du graphe des synchronisations et des dépendances de l'application, qui sert ensuite de support à l'élimination des éventuels inter-blocages entre les processus. Les méthodes exposées ont été illustrées et comparées tout d'abord en considérant le cas élémentaire du multiplieur-accumulateur du processeur RISC asynchrone ASPRO [REN] puis de l'algorithme de contrôle et d'espacement dans l'ATM. Pour résoudre les problèmes liés à la synthèse des circuits asynchrones, nous nous sommes appuyés sur les compétences de Marc Renaudin, alors responsable de l'antenne de Grenoble de l'ENST Bretagne. La thèse intitulée "Proposition de synthèse architecturale de circuits intégrés asynchrones adaptés aux applications à haut débit" a été soutenue en juin 2000. Un peu plus d'un an après le début du stage DEA de Frédéric Le Roy, France Telecom a cessé toute activité liée au développement de nouvelles méthodes et d'outils pour la conception de circuits intégrés. Le travail réalisé dans le cadre de cette thèse n'a pas eu de suite.

4.3 Références

[ITU] ITU-T Recommendation I371 "Traffic control and congestion in B-ISDN", 1996.

[MAR] A.J. Martin, "Programming in VLSI: from communicating processes to delay insensitive circuits" In C. A. R Hoare, editor, *Developments in concurrency and Communication*, UT Year of Programming Series, Addison-Wesley, 1990.

[REN] M. Renaudin, P. Vivet and F. Robin, "ASPRO-216: a standard cell QDI 16-bit RISC asynchronous microprocessor," *4th International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYN'98)*, San Diego, 1998.

5 Les turbocodes et les techniques de décodage itératif, suite (1995 – …)

Après avoir abordé les turbocodes sous l'angle de la réalisation matérielle des décodeurs, nous avons élargi notre champ d'investigation.

En 1995, nous avons étendu le principe du décodage itératif au problème de la correction de l'interférence entre symboles pour les transmissions sur canaux à trajets multiples. Le principe en est décrit dans le sous-chapitre 5.1.

A partir de 1996, nous avons été sollicités par plusieurs partenaires industriels pour définir des schémas de codage et des algorithmes de décodage adaptés à la transmission de données par paquets. Les travaux menés sur ce thème sont décrits dans le sous-chapitre 5.2. Parmi les résultats issus de ces travaux, citons le code défini pour les standards DVB-RCS et RCT, qui est maintenant considéré comme un code de référence dans de nombreuses publications.

D'autre part, lorsque les applications traitées visent des taux d'erreurs faibles ou très faibles, typiquement inférieurs à 10^{-7} en terme de taux d'erreurs de paquets ou de trames (TEP ou FER pour *Frame Error Rate*), il devient très difficile de qualifier les performances d'un code par simulation. C'est pourquoi nous avons mis au point en 2001 une méthode d'estimation rapide des performances asymptotiques des codes, appelée "méthode de l'impulsion d'erreur", présentée dans le sous-chapitre 5.3.

Le sous-chapitre 5.4 décrit succinctement une étude commencée très récemment sur le thème de l'association des turbocodes avec une nouvelle technique d'accès multiple par entrelacement.

5.1 Turbo-détection (1995)

5.1.1 Contexte

La publication des premiers résultats sur les turbocodes en 1993 a rapidement suscité le démarrage de nombreux travaux sur l'application du procédé de décodage itératif à d'autres types de concaténation, dans le domaine du codage de canal mais aussi dans des domaines connexes. On notera notamment l'application du décodage itératif à la concaténation série de codes algébriques [PYN] et convolutifs [BEN]. Nous nous sommes, pour notre part, penchés sur le problème de l'extension de ce principe à la correction de l'interférence entre symboles pour des transmissions sur canaux à trajets multiples en présence de codage correcteur d'erreurs : cette technique est maintenant connue sous l'appellation *turbo-égalisation* ou *turbo-détection*.

5.1.2 Principe et performances de la turbo-détection

Le principe de la turbo-détection est basé sur la représentation du modulateur et du canal de transmission à trajets multiples par un modèle discret dont le comportement est proche de celui d'un codeur convolutif (cf. Figure 5.1).

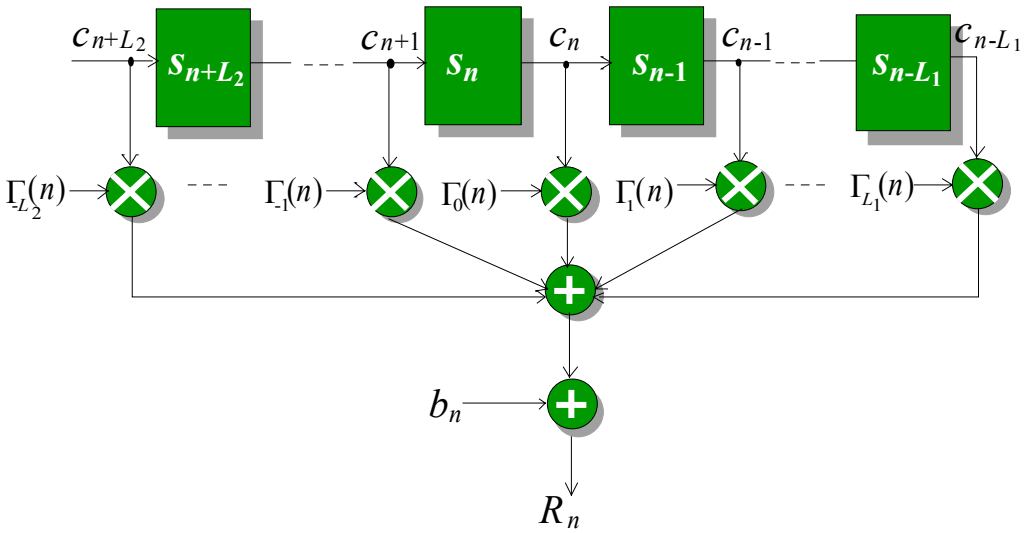


Figure 5.1 : Modèle discret équivalent d'un canal de transmission en présence d'interférence entre symboles

L'insertion d'une fonction d'entrelacement en sortie du codeur conduit à une concaténation similaire à la concaténation série de deux codes convolutifs, qui peut être décodée selon un processus itératif. La mise au point du turbo-détecteur a tout d'abord consisté à développer une fonction d'égalisation suivant le maximum de vraisemblance, fournissant une estimation pondérée des symboles en sortie du canal. Cet égaliseur a ensuite été concaténé en série avec un décodeur SISO. Puis, à l'issue du décodage, une fonction de calcul de l'information extrinsèque a été mise en œuvre sur l'ensemble des bits codés (dans les turbocodes classiques, on n'extrait des informations extrinsèques que sur les bits d'information). Cette information extrinsèque est réinjectée dans le calcul des métriques du processus d'égalisation pour effectuer les itérations successives. Le schéma de principe de ce récepteur est donné en Figure 5.2.

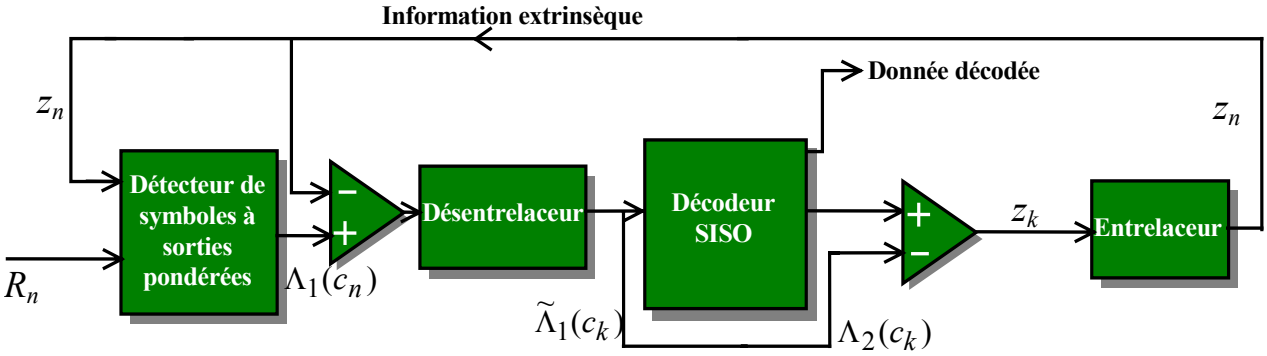


Figure 5.2 : Schéma de principe du turbo-détecteur

Les performances obtenues mettent en évidence que cette technique permet de lutter efficacement contre les effets nuisibles des trajets multiples. Les résultats de simulations reportés en Figure 5.3 montrent qu'au bout de six itérations, le processus de turbo-détection permet d'annuler l'interférence due aux trajets multiples. Des compléments de résultats sont présentés dans [DOUI1].

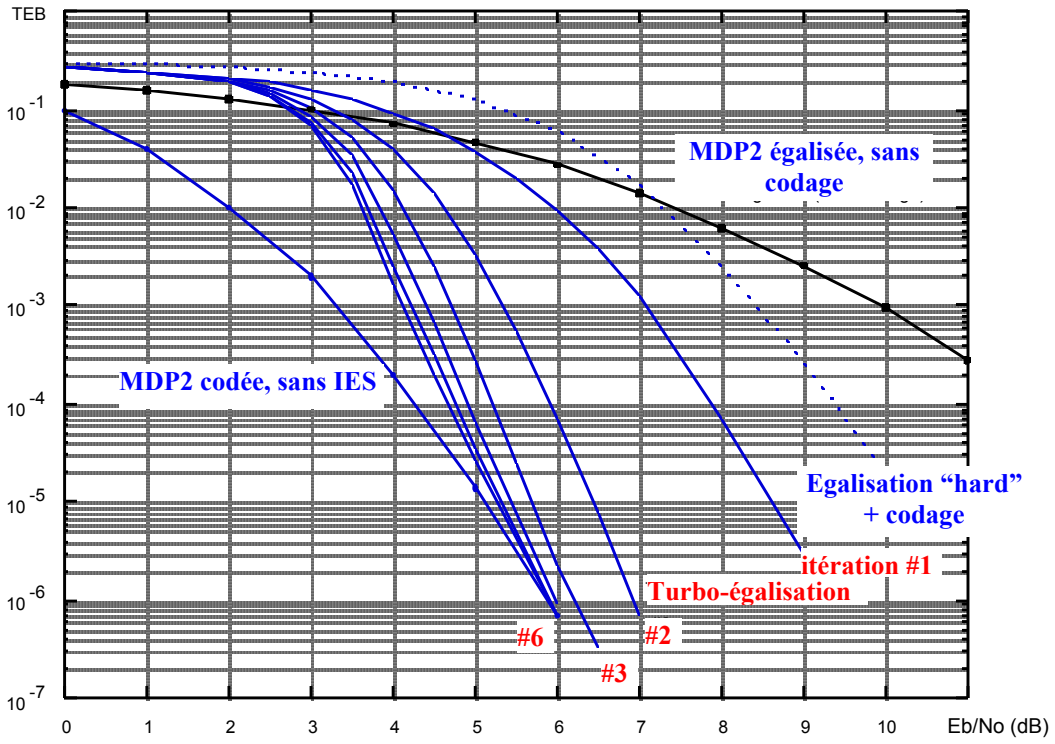


Figure 5.3 : Performances de la turbo-détection sur canal gaussien.
Code convolutif systématique récursif à 16 états , polynôme 23,35.
Entrelacement non uniforme de taille 64×64.
Coefficients du canal : $\Gamma_0(n) = \sqrt{0{,}45}$, $\Gamma_1(n) = \sqrt{0{,}25}$, $\Gamma_2(n) = \sqrt{0{,}15}$, $\Gamma_3(n) = \sqrt{0{,}10}$, $\Gamma_4(n) = \sqrt{0{,}05}$.

Ces travaux ont tout d'abord fait l'objet d'un contrat d'étude avec le CCETT [DOU2]. Ils ont également conduit au dépôt d'un brevet français [DOU3], par la suite étendu en Europe et aux États Unis. L'article [DOUI1], publié en 1995 dans la revue *European Transactions on Communications* et reproduit en Annexe 4, constitue la toute première publication sur le thème de la correction itérative de l'interférence entre symboles pour des transmissions codées. J'ai aussi contribué, au début de l'année 2003, à la rédaction d'un chapitre consacré à la turbo-détection et la turbo-égalisation [PIC] dans l'ouvrage intitulé *Signal et Télécommunications*, à paraître aux éditions Hermès Science.

Le principe de la turbo-détection a par la suite été étendu, par l'équipe du département Signal et Communications de l'ENST Bretagne, à un schéma de récepteur utilisant un égaliseur transverse à la place du détecteur de symboles [GLA][LAO].

5.2 Turbocodes convolutifs pour blocs courts (depuis 1996)

5.2.1 Contexte

Dans leur version initiale, les turbocodes sont construits à partir de deux codes convolutifs concaténés en parallèle (cf. Figure 3.3). Ce schéma de codage est *a priori* particulièrement bien adapté à la transmission de messages très longs, en théorie de longueur infinie. Aussi, les deux premiers circuits de turbocodage/décodage, TURBO3 et TURBO4, ont-ils été développés pour des applications de type diffusion. Cependant, dans la plupart des applications (ATM, Internet, communications radio-mobiles…), les données sont transmises par blocs, de taille parfois réduite (quelques centaines à quelques milliers de symboles).

Nos premiers travaux sur le problème du codage par blocs datent de 1996, dans le cadre d'un contrat d'étude avec le CCETT portant sur les turbocodes pour données transmises par salves (section 5.2.4). Depuis lors, nous n'étudions plus que ce type d'applications.

Le codage par blocs nous a, dans un premier temps, contraint à la résolution des problèmes suivants :

- l'adaptation des codes convolutifs et des turbocodes au codage de blocs courts. Nous avons dû résoudre le problème de la fermeture de treillis des turbocodes, exposé en section 5.2.3. Nous avons expérimenté deux techniques différentes de fermeture, décrites dans les sections 5.2.4 et 5.2.5 ;
- la mise en œuvre d'algorithmes performants pour la réalisation matérielle (FPGA, ASIC) et/ou logicielle (DSP) de décodeurs pour les blocs courts (section 5.2.4). Nous avons dans un premier temps adapté l'algorithme SOVA, jusqu'alors utilisé dans toutes nos circuits, puis nous avons mis en œuvre une version simplifiée de l'algorithme MAP, dite Max-Log-MAP [ROB] (voir Annexe 2), mieux adaptée au décodage des blocs courts.

Nous avons également par la suite porté notre effort sur la recherche de codes élémentaires permettant l'amélioration des performances de correction des turbocodes, aussi bien à faible qu'à fort rapport signal à bruit. Les codes *m*-binaires, présentés en section 5.2.6, nous ont permis de relever ce double défi. Les sections 5.2.7 et 5.2.8 décrivent les caractéristiques de deux turbocodes duo-binaires ($m = 2$), respectivement à huit et seize états, définissant des schémas de codage très flexibles car permettant de traiter une large gamme de taille de blocs et de rendements de codage, pour des complexités de décodage tout à fait compatibles avec une réalisation matérielle.

La section suivante décrit les principaux facteurs entrant en compte dans l'appréciation des performances de correction d'un système turbocodé. Elle constitue un pré-requis à la compréhension de notre approche pour définir des turbocodes toujours plus performants.

5.2.2 Pouvoir de correction des turbocodes

Les performances, en termes de correction, d'un système de codage dépendent du canal de transmission (Gauss, Rayleigh, …), du rendement de codage et de la taille des blocs de données transmises. La Figure 5.4 illustre deux comportements possibles typiques de systèmes de transmission turbocodés.

Les courbes 1 et 4 constituent deux courbes de référence. La courbe 1 affiche les performances limites d'un système de transmission idéal de blocs de 188 octets de données avec un rendement de codage égal à 1/2. Pour des données transmises sous la forme de séquences très longues, la limite théorique est donnée par la capacité du canal. Lorsque des blocs de données de longueur finie sont considérés, une borne inférieure classique sur la probabilité d'erreur pour les codes de taille finie est donnée par la borne dite de "l'empilement de sphères", plus couramment appelée *sphere-packing*, introduite par Shannon en 1959 [SHA]. En pratique, cette borne permet de calculer un facteur correctif à appliquer à la capacité du canal pour prendre en compte une taille de bloc finie donnée et un taux d'erreurs cible [DOL]. La courbe 4 affiche la probabilité d'erreur binaire du système de transmission sans codage pour la modulation considérée (ici MDP-2, modulation à déplacement de phase à deux états).

Les deux paramètres à prendre en compte lors de l'estimation des performances d'une transmission turbocodée sont les suivants :

– le **seuil de convergence** du processus itératif : il s'agit du rapport signal à bruit à partir duquel le système codé devient plus performant que le système de transmission non codé. Lorsque ce seuil est faible, les performances du système à des niveaux de bruit fort et moyen sont proches de la limite théorique. C'est typiquement le cas des systèmes faisant appel à un turbocode ;

– le **comportement asymptotique** du système codé. Le comportement à très faible niveau de bruit du système de transmission codé est essentiellement dicté par la distance minimale $d_{\min}$ du code, à savoir la plus petite distance de Hamming entre deux mots de codes. En effet, en première approximation, l'écart maximal entre la courbe de taux d'erreurs sans codage et la courbe avec codage, appelé gain asymptotique, est donnée par l'expression suivante :

$$G_a \approx 10 \log(R\, d_{\min})$$

où R est le rendement du code. Une faible valeur de $d_{\min}$ entraîne un fort changement de pente (on parle de *flattening*) dans la courbe de taux d'erreurs (cf. courbe 2) dû à un faible gain asymptotique. Lorsque le gain asymptotique est atteint, la courbe avec codage devient parallèle à la courbe sans codage.

La courbe 2 montre un exemple de performance correspondant à un faible seuil de convergence : les performances à faibles et moyens rapports signal à bruit sont très proches des performances théoriques prédites par Shannon. En contrepartie, ce système codé affiche un gain asymptotique limité dû à une distance minimale peu élevée. Ce comportement est typique de celui d'un turbocode lorsque l'entrelacement n'est pas défini correctement.

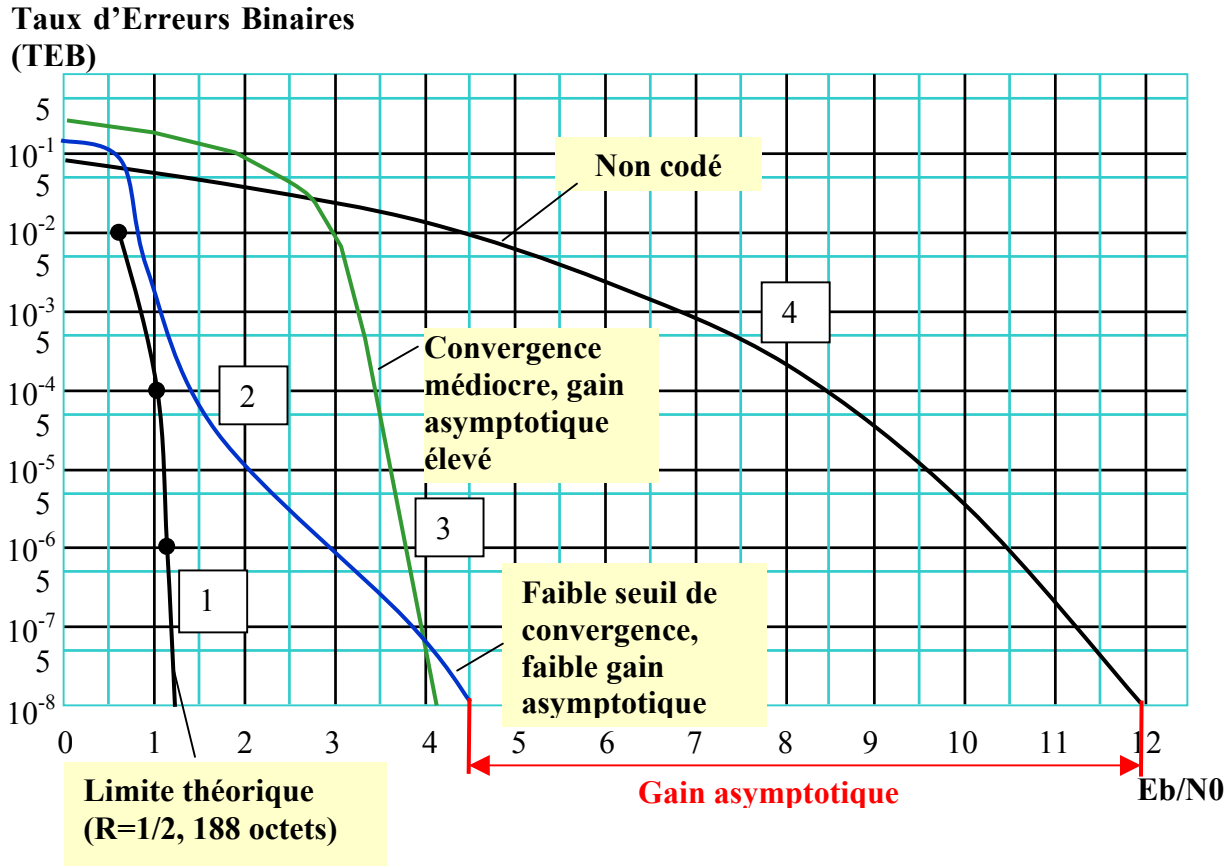


Figure 5.4 : Comportements typiques d'un système de transmission turbocodé sur canal gaussien utilisant une modulation à déplacement de phase à 2 états (MDP-2).

La courbe 3 montre les performances typiques d'un code avec une grande distance minimale, mais une convergence médiocre, liée à un processus de décodage largement sous-optimal.

La recherche d'un schéma de codage et décodage performant doit toujours faire face au dilemme de favoriser la convergence par rapport au gain asymptotique ou *vice versa*, l'amélioration d'un des deux critères conduisant le plus souvent à la dégradation du second.

L'élaboration d'un turbocode présentant un seuil de convergence proche de la limite théorique passe par le choix de codes élémentaires présentant un bon comportement à faible rapport signal à bruit, ainsi qu'à une fonction d'entrelacement limitant le plus possible la corrélation entre les séquences de données dans l'ordre naturel, c'est-à-dire dans l'ordre où elles sont émises par la source d'information, et dans l'ordre entrelacé.

La fonction d'entrelacement détermine également les performances asymptotiques du code composite, en relation étroite avec les propriétés de correction des codes élémentaires.

5.2.3 Le problème de la fermeture de treillis

Le codage d'un bloc d'information à l'aide d'un code convolutif fait apparaître, lors du décodage, des discontinuités aux extrémités de ce bloc. En effet, les algorithmes de décodage des codes convolutifs, algorithme de Viterbi ou algorithme MAP, font appel, pour le décodage du symbole à l'instant i, à l'ensemble des informations antérieures et postérieures à i. Dans le cas d'une transmission par blocs, si l'état initial du codeur est inconnu, les informations en début de bloc ne peuvent pas bénéficier du "passé". De même, si l'on ne dispose d'aucune information sur l'état du codeur en fin de codage du bloc, les dernières données ne peuvent pas bénéficier du "futur". En conséquence, les performances de correction sont dégradées par une moindre protection des symboles situés aux extrémités du bloc. Ce phénomène est d'autant plus marqué que le bloc est de petite taille.

Ce problème peut être résolu si l'on sait "fermer le treillis", c'est-à-dire si le décodeur connaît l'état initial et l'état final du codeur. En pratique la connaissance de l'état initial du codeur ne pose pas de problème particulier, car il suffit de convenir à l'avance de sa valeur : le plus souvent, tous les éléments mémoire des registres de codage sont forcés à la valeur 0 (état "tout zéro", noté **0**) au début du codage de chaque bloc. La connaissance de l'état final pose un problème beaucoup plus délicat, celui-ci dépendant du contenu du bloc à coder.

Parmi les différentes techniques pouvant être mises en œuvre, la plus simple d'entre elles consiste à fermer le treillis d'un ou des deux codes élémentaires à l'aide de bits de terminaison ou *tail bits*. La Figure 5.5 illustre ce principe dans le cas du code élémentaire introduit en Figure 3.4(b). L'application à l'entrée du codeur de $\nu = 3$ données de terminaison d_t suffit à ramener le codeur dans l'état **0**.

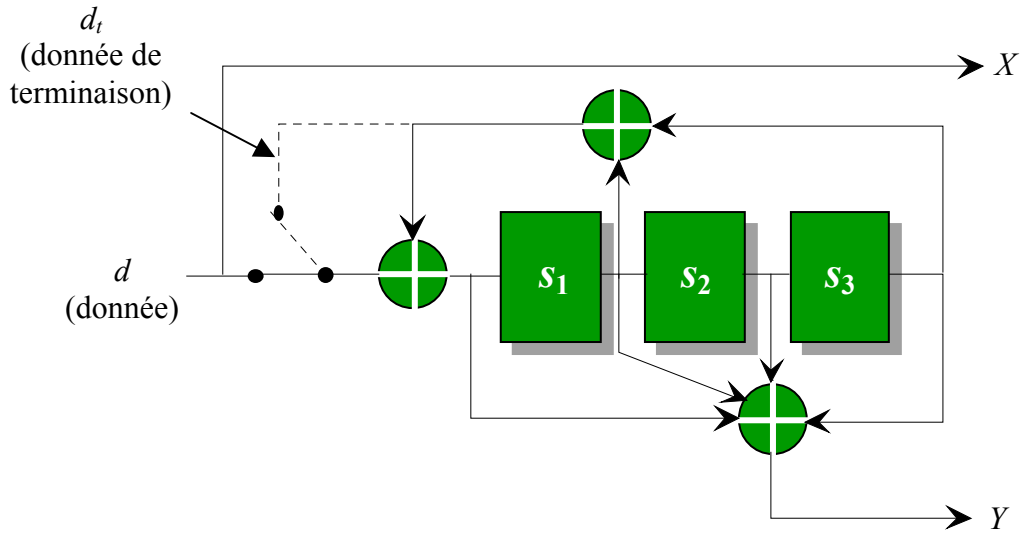


Figure 5.5 : Principe de fermeture du treillis par le biais de bits de terminaison. Les lignes pointillées représentent la configuration du codeur lors de la terminaison

Dans le cas des turbocodes des standards CCSDS [CCSDS] et UMTS [3GPP], les bits assurant la terminaison de l'un des deux treillis ne sont pas utilisés dans l'autre codeur. Ces bits ne sont donc pas *turbocodés* ce qui conduit, mais dans une moindre mesure, aux mêmes inconvénients que ceux présentés par une absence de fermeture. De plus, la transmission des bits de fermeture entraîne une diminution du rendement de codage et donc de l'efficacité spectrale. Par conséquent, cette méthode de terminaison, bien que couramment utilisée pour les codes convolutifs, n'est pas satisfaisante dans le cas des turbocodes.

Pour pallier ces différents problèmes, nous avons proposé deux solutions permettant une fermeture automatique des treillis sans ajout de bit de terminaison. La première d'entre elles est basée sur l'association de turbocodes dits *auto-concaténés* avec une structure d'entrelacement particulière, décrite en section 5.2.4. Elle a été validée dans le cadre d'un contrat d'étude avec le CCETT et a également été mise en œuvre dans un turbo-décodeur réalisé sur DSP. La seconde fait appel à la notion de *code convolutif circulaire*, décrite en section 5.2.5. Cette dernière technique est désormais celle que nous adoptons systématiquement depuis 1999 pour le codage en blocs.

5.2.4 Les turbocodes auto-concaténés

Les deux projets présentés ci-après mettent en œuvre la notion de turbocodes *auto-concaténés* : un seul codeur élémentaire est en fait utilisé pour coder consécutivement le bloc de données dans l'ordre naturel, puis dans l'ordre entrelacé, **sans réinitialisation du codeur entre les deux phases de codage**.

La fonction d'entrelacement est construite de telle sorte que chaque donnée est appliquée en entrée du codeur à deux instants séparés par un nombre d'instants multiple de la période L du du générateur pseudo-aléatoire constitué par le codeur élémentaire. Il est montré dans [BER1] que, dans ces conditions, l'état final après le codage de la séquence par le registre initialisé à **0** est aussi **0**. Ainsi l'état initial et l'état final du codeur sont connus par le décodeur et une séquence quelconque de k données peut être décodée par ce dernier, indépendamment de toute information antérieure ou ultérieure. Cette technique de codage impose à k d'être un multiple de L.

Contrat d'études CCETT, 1996-1997, "Turbocodes pour données transmises par salves (turbocodes convolutifs pour blocs courts)" [BER2]

Nous avons, dans le cadre de cette étude, validé le principe de fermeture décrit précédemment. Nous avons modélisé, sous forme logicielle, le codage des blocs de données, leur transmission sur canal gaussien et canal de Rayleigh avec une modulation à déplacement de phase à deux états (MDP-2), puis le décodage itératif. Les tailles de blocs considérées sont comprises entre 200 et 1000 bits.

Le modèle du turbo-décodeur a été décrit en adéquation avec sa réalisation sous forme de circuit (entrées, sorties et données internes quantifiées), afin de garantir un comportement identique au bit près à celui du circuit utilisant la même représentation des nombres et afin de permettre le dimensionnement et le dénombrement des opérations nécessaires au déroulement de l'algorithme de décodage.

Nous avons évalué deux algorithmes de décodage élémentaire : l'algorithme SOVA (*Soft-Output Viterbi Algorithm*), jusqu'à présent utilisé dans les différents circuits étudiés, et une version simplifiée de l'algorithme MAP (*Maximum A Posteriori*), dite Max-Log-MAP [ROB], que nous mettions en œuvre pour la première fois. L'algorithme Max-Log-MAP est plus naturel à implémenter pour le décodage de blocs que l'algorithme SOVA et ses performances, en terme de correction, sont supérieures : une amélioration de 0,1 à 0,2 dB est observée pour des blocs de plusieurs milliers de bits de données, qui peut aller jusqu'à 0,4 ou 0,5 dB pour des blocs de quelques centaines de bits seulement.

Nous avons également étudié l'évolution du taux de correction des erreurs en fonction des itérations et nous avons observé que pour des valeurs de rapport signal à bruit situées au delà du seuil de convergence du turbo code, dans plus de 90 % des cas, quatre et souvent moins de quatre itérations de décodage suffisent pour corriger les erreurs. Le système étudié étant supposé fonctionner par salves asynchrones, nous avons mis en œuvre un mécanisme de supervision du processus itératif qui décide, à chaque itération, de poursuivre ou non le processus de décodage du bloc concerné. Le **critère d'arrêt** appliqué consiste à comparer, après chaque itération de décodage, les fiabilités de chaque symbole décodé à un seuil préfixé *S*. Si toutes les fiabilités sont supérieures à *S*, le processus itératif est arrêté, sinon il se poursuit. Ce dispositif, très simple à mettre en œuvre, ne dégrade pas de façon significative les performances du décodeur si la valeur du seuil est fixée à une valeur égale à 3 ou 4 fois la valeur moyenne du poids des échantillons en entrée du décodeur.

Les différents points abordés dans cette étude ont par la suite été également étudiés lors de l'implémentation du décodeur sur processeur de traitement de signal, dans le cadre de la thèse de Fathi Raouafi, que j'ai co-encadrée avec Claude Berrou.

Thèse de Fathi Raouafi, 1998-2001: "Adéquation turbocodes/processeurs de signal" [RAO1]

Le travail réalisé dans cette thèse consistait à étudier en détail l'implantation efficiente de turbocodes pour bloc courts, selon les principes précédemment décrits, sur des DSP à faible coût et faible consommation.

Nous nous sommes, en pratique, intéressés à la famille TMS320C54x de Texas Instruments. Ces architectures comportent une unité matérielle, dite *accélérateur de Viterbi*, permettant d'optimiser le calcul et la gestion des métriques de nœuds dans l'algorithme de Viterbi.

L'implémentation des algorithmes de décodage SOVA et Max-Log-MAP a été étudiée, la plupart des opérations à effectuer étant identiques à celles requises par l'algorithme de

Viterbi : calcul de métriques cumulées, comparaison des métriques, sélection et stockage de la métrique à vraisemblance maximale. Les performances médiocres, en terme de vitesse, des codes exécutables obtenus par les compilateurs et optimiseurs de langage de haut niveau nous ont contraints à une optimisation manuelle du code assembleur, afin notamment de tirer le meilleur parti de l'accélérateur de Viterbi.

Après une comparaison approfondie des deux algorithmes en termes de complexité matérielle, de débit de décodage et de taux d'erreurs, l'algorithme Max-Log-MAP a été retenu pour l'implémentation du turbo-décodage sur DSP [RAO2]. Dans cette partie du travail, le principal effort a porté sur le problème de la réduction de la taille mémoire nécessaire à la mémorisation des métriques lors des décodages élémentaires. Quatre méthodes différentes ont été proposées, toutes basées sur l'utilisation d'une fenêtre glissante : le traitement de chaque itération est décomposé en sous-traitements ne portant chacun que sur une partie (fenêtre) du treillis. Les méthodes diffèrent par la technique d'initialisation des métriques aux extrémités de la fenêtre ; elles conduisent à des solutions correspondant à différents compromis de performances débit de décodage/pouvoir de correction. Un mécanisme de critère d'arrêt des itérations similaire à celui cité dans l'étude précédente a également été combiné au traitement par fenêtre afin d'en accélérer le décodage pour des valeurs de rapport signal à bruit suffisamment élevées.

5.2.5 Introduction du codage circulaire (1999)

Le gain de codage asymptotique des turbocodes obtenus par le principe de fermeture de treillis à base de codes auto-concaténés s'avère difficile à maîtriser. En effet, les conditions imposées sur la fonction d'entrelacement sont, dans de nombreux cas, trop contraignantes pour être compatibles avec l'obtention de grandes distances minimales. C'est pourquoi nous avons par la suite élaboré une nouvelle méthode de fermeture de treillis ne faisant plus intervenir la fonction d'entrelacement du code.

Avec la technique des *codes convolutifs circulaires* (ou *tail-biting codes*) le codeur retrouve en fin de codage de chaque bloc son état initial, cet état dépendant du contenu du bloc. Le principe, initialement proposé dans [WEI], a été adapté dans [BER3] au cas des codes récursifs systématiques (cet article est reproduit en Annexe 5).

En résumé, l'état final du codeur après codage d'un bloc de données peut être calculé à partir de la matrice génératrice **G** du code, de l'état initial $\mathbf{S}_0$ et de la séquence de données à coder $\mathbf{d}_1^k$, k représentant la longueur de cette séquence. On montre alors que, si la matrice $\left\langle \mathbf{I} + \mathbf{G}^k \right\rangle$ est inversible (**I** représente la matrice identité), il existe pour chaque séquence de données un état du treillis $\mathbf{S}_c$, appelé état de circulation, qui garantit que, si le codage du bloc démarre à partir de l'état $\mathbf{S}_c$, il se termine également dans l'état $\mathbf{S}_c$.

Pour un code donné, la valeur de $\mathbf{S}_c$ dépend du contenu du bloc à coder et de sa longueur k. En pratique, elle peut facilement être obtenue en trois étapes.

1. le codeur est tout d'abord initialisé à l'état $\mathbf{S}_0 = \mathbf{0}$, puis le message à coder lui est appliqué une première fois, en ignorant la redondance produite pendant cette étape. Notons $\mathbf{S}_0^k$ l'état final correspondant ;

2. l'état de circulation $\mathbf{S}_c$ peut ensuite être calculé à partir de l'expression suivante :

$$\mathbf{S}_c = \left\langle \mathbf{I} + \mathbf{G}^k \right\rangle^{-1} \mathbf{S}_0^k$$

En pratique, l'utilisation d'une table à ν entrées et ν sorties, ν étant la mémoire du code, permet de déterminer $\mathbf{S}_c$ à partir de $\mathbf{S}_0^k$;

3. enfin, le codeur ayant été initialisé à l'état de circulation, le message à coder lui est de nouveau appliqué pour produire la redondance.

Cette méthode, élégante et efficace pour transformer un code convolutif en code en bloc, présente donc l'inconvénient de nécessiter une étape de pré-codage de chaque message pour connaître la valeur $\mathbf{S}_0^k$. Ceci ne représente pas un handicap majeur car la structure du codeur est très simple et celui-ci peut fonctionner à une fréquence d'horloge beaucoup plus élevée que le décodeur (sans problème à une fréquence double).

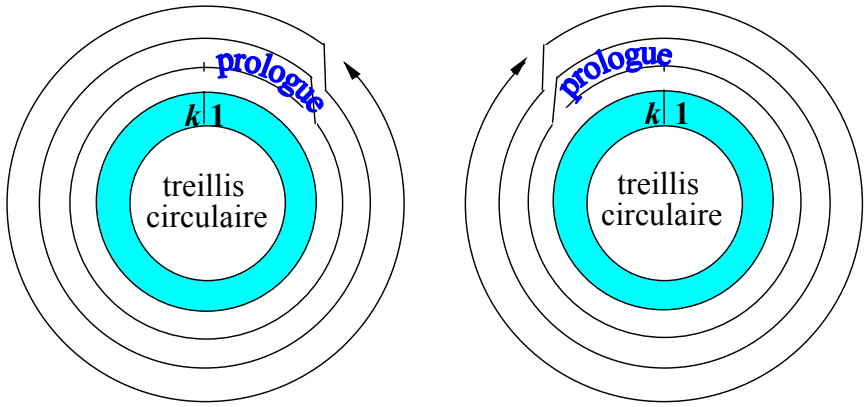
Le treillis du code prend alors la forme d'un cercle et peut être considéré comme un treillis de longueur infinie du point de vue du décodeur.

Le turbocodage de la séquence de données peut donc être représenté par deux treillis circulaires. Le décodage itératif requiert alors, quel que soit l'algorithme élémentaire de décodage utilisé, un parcours répété des treillis circulaires, avec une mise à jour du tableau des informations extrinsèques au fur et à mesure du traitement des données. Grâce à la propriété de circularité, les différentes itérations s'enchaînent naturellement dans la continuité des transitions d'état à état.

Si l'algorithme MAP ou Max-Log-MAP est utilisé, le décodage consiste à parcourir le treillis circulaire dans le sens trigonométrique direct pour le traitement aller, et dans le sens inverse pour le traitement retour (cf. Figure 5.6). Pour chacun des deux traitements, les probabilités calculées à la fin d'un tour de treillis sont réutilisées comme valeurs initiales pour le tour suivant.

Les différentes itérations de décodage s'enchaînent naturellement en tournant de façon continue autour du treillis. Les états initial et final de codage ne sont plus des états singuliers et jouent le même rôle que l'état du codeur à n'importe quel autre instant. Par exemple, avec un codage circulaire, le processus de décodage peut démarrer n'importe où dans le treillis.

La valeur de l'état de départ du décodage n'étant pas transmise au décodeur, celui-ci doit l'estimer par une étape préalable de traitement des données précédant cet état, appelée *prologue*. Le prologue est similaire à une étape de décodage démarrant avec des états initiaux équiprobables, mais ne fournissant ni décision ni information extrinsèque sur les données traitées. En pratique, l'exploitation d'une dizaine de symboles de redondance environ suffit à une estimation fiable de l'état initial ou final.



traitement retour **traitement aller**

Figure 5.6 : Le traitement d'un code circulaire par l'algorithme MAP.

Contrairement au principe des codes auto-concaténés, cette méthode-ci n'impose aucune contrainte forte sur la fonction d'entrelacement du code, la seule contrainte pratique étant que la longueur du bloc de données à coder, k, ne doit pas être multiple de la période du codeur L, ceci afin de garantir l'inversibilité de la matrice $\mathbf{I}+\mathbf{G}^k$. En effet $\mathbf{I}+\mathbf{G}^L=\mathbf{I}+\mathbf{I}=\mathbf{0}$.

5.2.6 Introduction des codes m*-binaires (1999)*

Parallèlement à la résolution du problème de fermeture des treillis par les codes circulaires, nous avons porté un gros effort sur la recherche de codes élémentaires permettant d'améliorer les performances des turbocodes, aussi bien à faible qu'à fort rapport signal à bruit, c'est-à-dire aussi bien du point de vue de la convergence que du comportement asymptotique.

C'est ainsi que nous avons exploré la voie des *codes m-binaires*. Les turbocodes *m*-binaires [BER4][BER5] sont construits à partir de codes convolutifs systématiques récursifs à *m* entrées binaires (Figure 5.7).

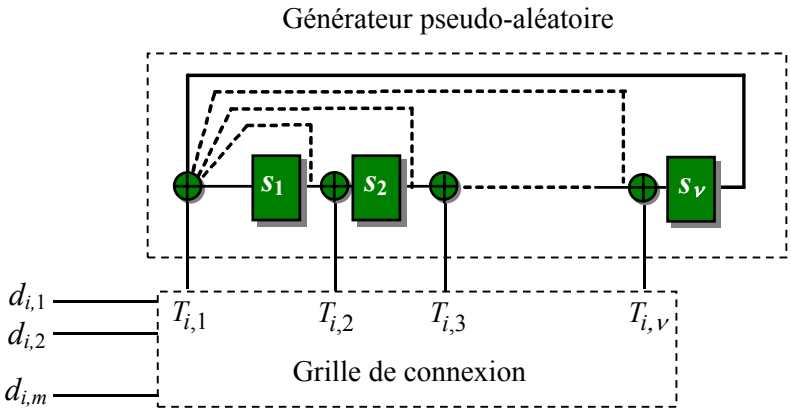


Figure 5.7 : Structure générale d'un codeur convolutif systématique récursif *m*-binaire de mémoire ν

Les avantages principaux de cette construction par rapport au schéma classique des turbocodes binaires (m = 1) sont nombreux :

– **Meilleure convergence du processus itératif**

Les effets de corrélation entre les décodeurs élémentaires jouent un rôle important dans la convergence du processus de décodage itératif. Par exemple, l'augmentation de la longueur de mémoire ν d'un code convolutif permet en général d'augmenter sa distance libre, c'est-à-dire la distance de Hamming minimale entre deux chemins distincts du treillis du code. Dans le contexte du turbocodage, la longueur moyenne des chemins erronés dans le treillis augmente et l'on observe plus fréquemment des motifs provoquant des inter-blocages entre les deux dimensions du code. Ce phénomène est illustré sur la Figure 5.7. En conséquence, le seuil de convergence du turbocode est détérioré.

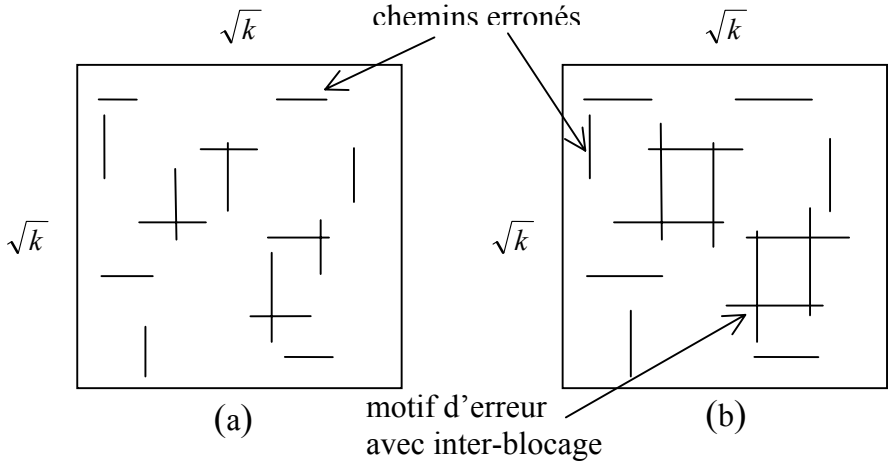


Figure 5.8 : Exemples de chemins erronés lors du décodage itératif d'un turbocode. Le cas (a) correspond à des codes élémentaires de longueur de mémoire ν faible. Dans le cas (b), une valeur de ν plus élevée produit des chemins erronés plus longs et des motifs d'erreurs composites inter-bloqués. Dans cet exemple, on considère un entrelacement régulier de type ligne-colonne.

Lorsque les codes élémentaires sont des codes *m*-binaires, la longueur moyenne des chemins erronés dans le treillis est divisée par 2^{m-1} par rapport à un code classique binaire, car 2^m transitions arrivent et partent de chaque nœud du treillis au lieu de 2. La longueur de la séquence à coder est divisée par *m*, *m* bits étant codés simultanément. Aussi, dans une représentation similaire à celle de la Figure 5.7, chaque dimension du carré est divisée par $\sqrt{m}$ et la densité d'erreur par dimension est divisée par $2^{m-1}/\sqrt{m}$. En conséquence, pour une longueur de mémoire donnée, un turbocode *m*-binaire présente une meilleure convergence qu'un turbocode binaire.

– **Sensibilité réduite vis à vis du poinçonnage pour l'obtention de rendements élevés**

Le rendement de codage naturel, c'est-à-dire sans poinçonnage des données codées, d'un codeur convolutif *m*-binaire est égal à $m/(m+1)$ si une redondance est produite à chaque instant de codage, contre 1/2 pour un code binaire. Ainsi, pour obtenir des rendements de codage élevés, moins de symboles de redondance doivent être poinçonnés que dans le cas binaire. Les turbocodes *m*-binaires sont, par conséquent d'excellents candidats pour les transmissions à grandes efficacités spectrales, associés à des modulations à grand nombre de points, où sont plutôt recherchés des rendements moyens ou élevés.

– **Latence plus faible**

Les données étant traitées par groupes de *m* bits, la latence de codage et de décodage est divisée par *m* par rapport à un code binaire.

- **Meilleure robustesse du décodeur envers la sous-optimalité de l'algorithme de décodage**

  Plus le treillis est compact, c'est-à-dire plus le nombre de transitions entre états est élevé, et plus les algorithmes de décodage (SOVA, Max-Log-MAP, …) sont proches du décodage à maximum de vraisemblance. En conséquence, la différence de performance entre ces algorithmes tend à diminuer lorsque la valeur de *m* augmente.

- **Plus grandes distances minimales**

  Le codage *m*-binaire introduit un degré de liberté supplémentaire dans la construction de la fonction d'entrelacement : la *permutation intra-symbole* permet d'étendre l'espace de recherche par rapport à un code binaire et d'obtenir des gains asymptotiques plus élevés. Pour ce faire, on opère une modification de l'ordre des *m* entrées binaires avant encodage par le second code. L'utilisation de code *m*-binaire permet ainsi de gagner simultanément sur la convergence et le gain asymptotique.

Du point de vue du décodage, chaque décodeur élémentaire travaille en interne sur des symboles *M*-aires, où $M = 2^m$. Dans les turbo-décodeurs que nous avons mis en oeuvre, chaque décodeur élémentaire calcule $M = 2^m$ Log-APPs $L_i(j)$ définies par :

$$L_i(j) = \ln \Pr(\mathbf{d}_i \equiv j \,\big|\, \mathbf{R}_1^N), j = 0 \cdots M$$

$\mathbf{d}_i$ désigne la donnée *m*-binaire codée à l'instant i, j est le nombre issu de la conversion de $\mathbf{d}_i$ en une valeur entière et $\mathbf{R}_1^N$ représente la séquence bruitée reçue en entrée du décodeur. De même, les informations extrinsèques sont des grandeurs *M*-aires. Leur extraction en sortie de chaque décodeur et leur insertion en entrée de l'autre décodeur ne peut plus être directement réalisée suivant le schéma présenté sur la Figure 3.5 du sous-chapitre 3.1. L'ensemble des opérations mises en œuvre dans un décodeur *m*-binaires sont détaillées dans l'article [DOU5], également reporté en Annexe 6.

En pratique, nous n'avons mis en œuvre que le cas $m = 2$ (codes duo-binaires), pour lequel la complexité matérielle reste raisonnable : la complexité du décodeur est double de celle d'un décodeur binaire mais deux bits sont décodés en même temps au lieu d'un, en conséquence la complexité de décodage par bit décodé est à peu près équivalente à celle d'un décodeur binaire.

Par comparaison avec les premiers résultats publiés en 1993, le passage de codes élémentaires binaires à des codes élémentaires duo-binaires a permis de se rapprocher de 0,15 dB de la limite théorique de Shannon, comme le montre la Figure 5.9.

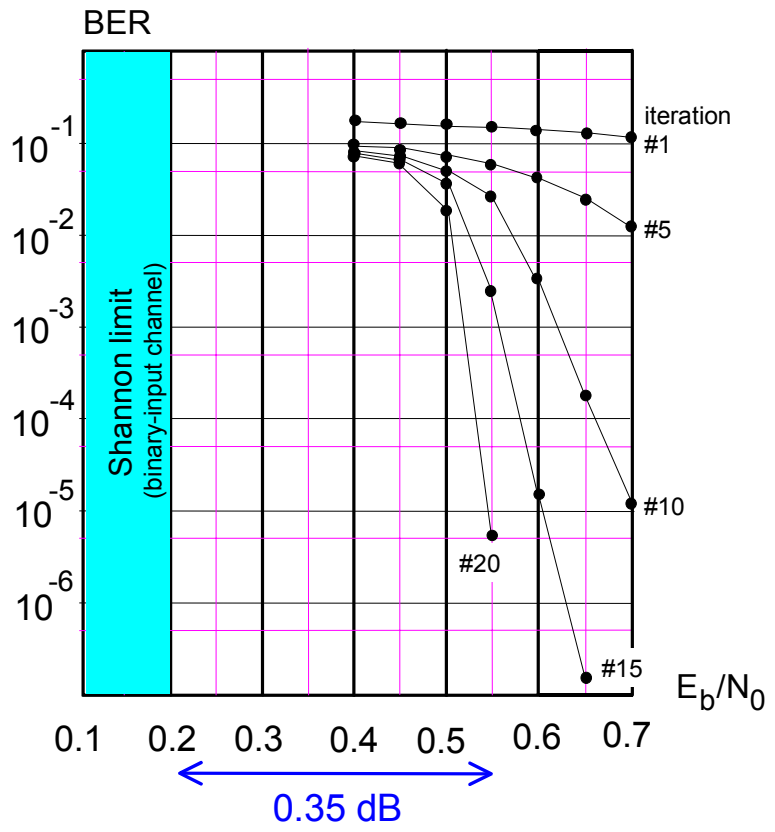


Figure 5.9 : Performance, en taux d'erreurs binaires (TEB), d'un turbocode duo-binaire à 8 états (code élémentaire de la Figure 5.10). Canal gaussien, modulation MDP-4. Rendement de codage R = 1/2. Entrelacement sur 65536 bits. Décodage MAP.

5.2.7 Définition du turbocode DVB-RCS (1999)

En 1999, la société Eutelsat a fait appel à l'ENST Bretagne pour proposer une solution de codage dans le cadre du processus de normalisation de la voie de retour du réseau satellitaire de télévision numérique DVB-RCS [DVB1]. Nous avons proposé une solution faisant appel aux concepts de codes duo-binaires et circulaires présentés précédemment.

Les applications DVB-RCS visent des transmissions de données utilisant sept tailles de blocs possibles (de 12 à 216 octets) et cinq rendements de codage différents (1/2, 2/3, 3/4, 4/5 et 6/7). Le code proposé est un turbocode duo-binaire utilisant deux codes élémentaires identiques à huit états ($\nu = 3$), décrits en Figure 5.10.

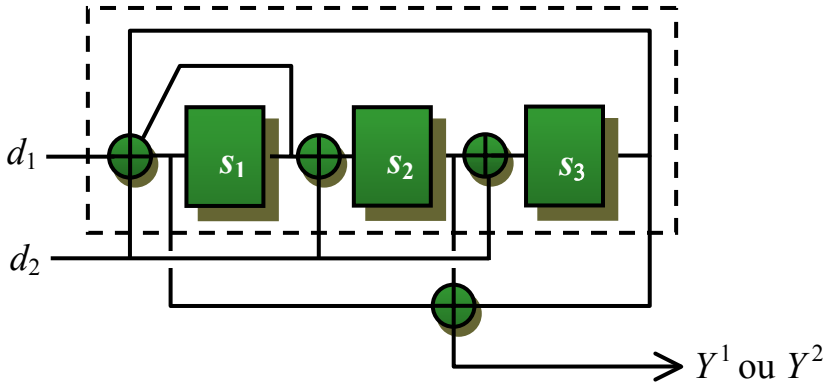


Figure 5.10 : Codeur élémentaire du turbocode DVB-RCS

Les principaux atouts de ce turbocode, dont les caractéristiques sont détaillées dans l'article [DOU4], reproduit en Annexe 7, sont les suivants :

- un simple poinçonnage régulier de la redondance permet d'obtenir les rendements de codage supérieurs à 1/2 ;
- la fonction de permutation fait appel à des équations génériques avec un faible nombre de paramètres (quatre), dont la valeur ne dépend que de la taille du bloc de données ;
- le même décodeur peut être utilisé pour tous les cas, pourvu qu'il soit dimensionné pour la plus grande taille de bloc à traiter.

Il s'agit là d'un code simple, offrant une grande souplesse d'adaptation à différentes tailles de blocs et divers rendements de codage, dont la complexité de décodage est tout à fait raisonnable : sa réalisation matérielle par la société TurboConcept a conduit à une complexité de moins de 20 000 portes logiques élémentaires par itération de décodage, lorsque les données sont décodés au rythme de l'horloge du circuit (mémoire non comprise).

Un exemple de performance de correction du code DVB-RCS est présenté en Figure 5.11. On y observe de bonnes performances en moyenne, qui restent cohérentes vis à vis de la variation avec le rendement de codage : l'écart des courbes simulées par rapport aux limites théoriques affichées varie peu. Cet écart est compris entre 0,6 et 0,7 dB pour un taux d'erreurs de trames de 10^{-4} et entre 0,8 et 1,0 dB pour un taux d'erreurs de trames de 10^{-5}. Une cohérence similaire peut également être observée vis à vis de la variation de performance avec la taille des blocs. La réalisation d'un démonstrateur logiciel[1], fonctionnellement compatible avec une implémentation sur circuit, a permis de défendre avec succès cette solution technique.

Le même schéma de codage a ensuite été adopté dans la norme pour la voie de retour du réseau terrestre de télévision numérique DVB-RCT [DVB2]. Il a également été retenu pour le codage de la voie montante dans le système SkyPlexNet [BER6], développé par la société italienne Alenia Spazio.

[1] L'exécutable est en accès public sur simple inscription à l'URL http://www-turbo.enst-bretagne.fr/download/

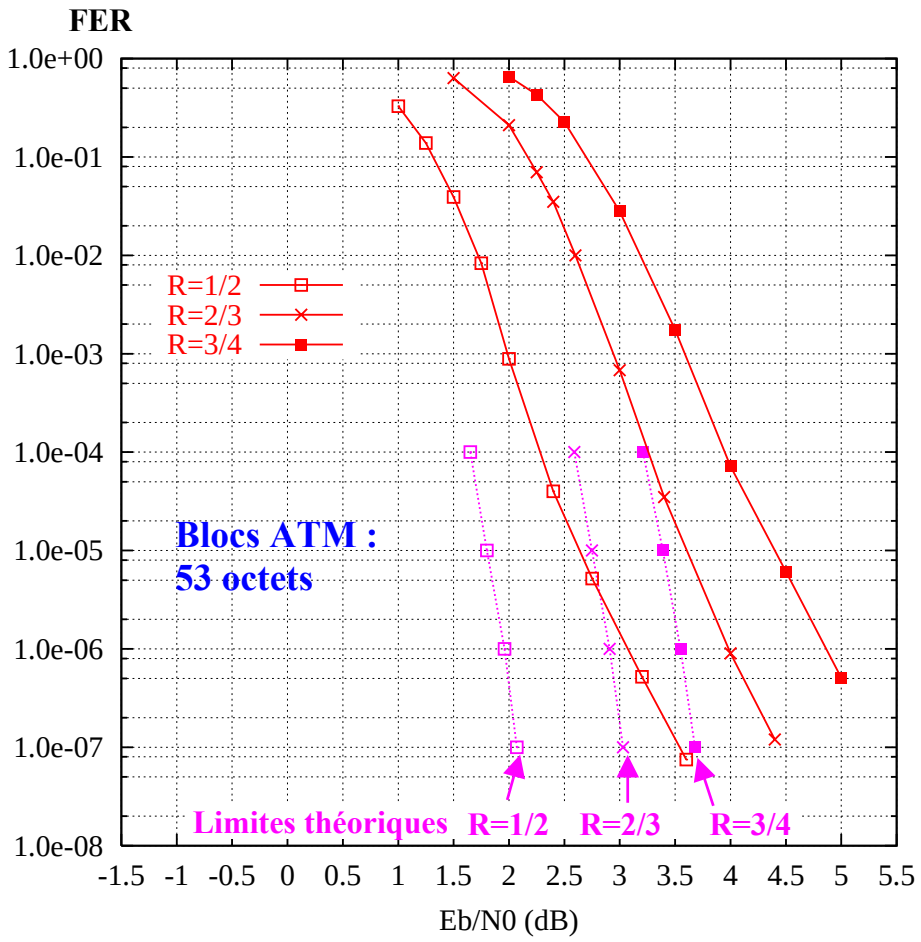


Figure 5.11 : Performance en taux d'erreurs de trames ou *Frame Error Rate* (FER) du code DVB-RCS pour la transmission de blocs ATM (53 octets) avec des rendements de codage 1/2, 2/3 et 3/4. Modulation QPSK et canal gaussien. Huit itérations de décodage Max-Log-MAP avec des échantillons codés sur 4 bits en entrée du décodeur. Les lignes en pointillés représentent la limite théorique prenant en compte le taux d'erreurs ciblé et la taille des blocs.

5.2.8 Extension du turbocode DVB-RCS à un turbocode à seize états (2000)

L'extension du schéma présenté précédemment vers des codeurs élémentaires à seize états permet d'en augmenter significativement les distances minimales. Le meilleur code élémentaire que nous ayons trouvé est décrit par le schéma de la Figure 5.12. Les performances comparées de ce turbocode avec le code DVB-RCS (cf. Figure 5.13) montrent une nette amélioration du comportement du système codé aux faibles taux d'erreurs tout en n'affichant pas de dégradation notable de la convergence aux faibles rapports signal sur bruit. Pour la transmission de blocs de 188 octets, à un taux d'erreurs de trames de 10^{-6}, le gain de codage par rapport au code DVB-RCS varie de 0,6 à 0,9 dB suivant le rendement de codage considéré.

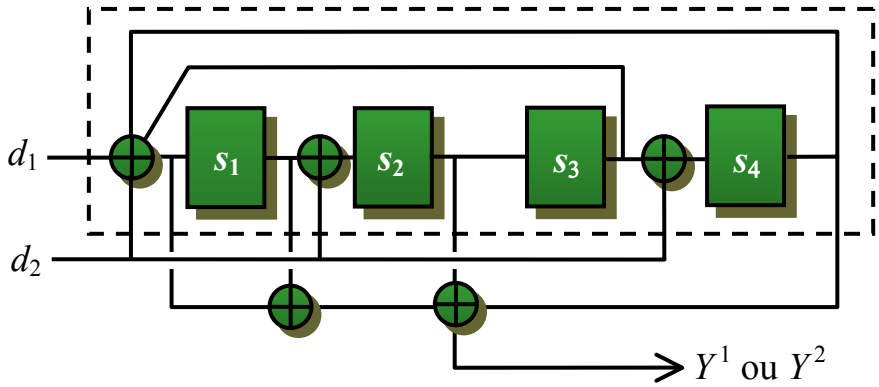


Figure 5.12 : Codeur élémentaire du turbocode à seize états

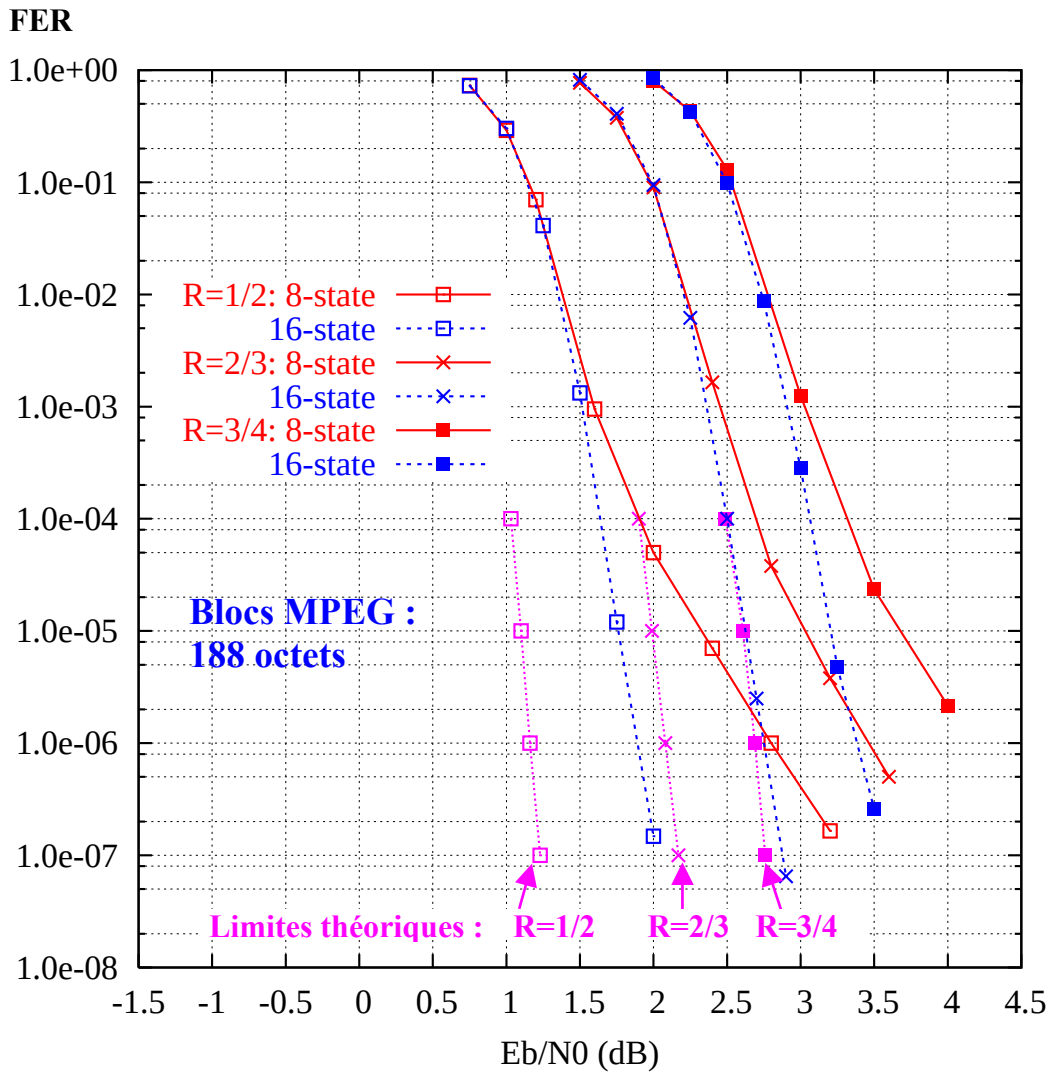


Figure 5.13 : Performance en taux d'erreurs de trames ou *Frame Error Rate* (FER) des turbocodes 8 et 16 états pour la transmission de blocs MPEG (188 octets) avec des rendements de codage 1/2, 2/3 et 3/4. Modulation QPSK et canal gaussien. Huit itérations de décodage Max-Log-MAP avec des échantillons codés sur 4 bits en entrée du décodeur. Les lignes en pointillés représentent la limite théorique prenant en compte le taux d'erreurs cible et la taille des blocs.

Le Tableau 5.1 compare les distances minimales des codes à huit et seize états pour des blocs de données de 188 octets : le passage à seize états permet d'augmenter les distances minimales de 30 à plus de 50% suivant les rendements considérés.

Rendement de codage	1/2	2/3	3/4	5/6
Turbocode duo-binaire à 8 états (DVB-RCS)	19	13	10	7
Turbocode duo-binaire à 16 états	29	19	13	9

Tableau 5.1 : Distances minimales des turbocodes duo-binaires à 8 états et 16 états pour des blocs de données de 188 octets. Evaluation selon la méthode de l'impulsion d'erreur.

Ainsi, un tel circuit de turbo-décodage, traitant des données quantifiées sur quatre bits en entrées avec l'algorithme simplifié Max-Log-MAP, est capable de décoder à un taux d'erreurs de trames de 10^{-6} à moins de 0,7 dB de la limite théorique.

En contre-partie, doubler le nombre d'états des codes élémentaires se traduit également par une multiplication par deux de la complexité de décodage, le nombre d'opérations a effectuer à chaque étape du treillis étant doublé. Le passage à seize états ne présente donc un intérêt que si les taux d'erreurs visés sont suffisamment faibles.

5.2.9 Conclusion

Le sous-chapitre 5.2 décrit l'évolution depuis 1996 de nos travaux sur la recherche de turbocodes adaptés à la transmission de données par blocs. Nous disposons actuellement, avec les turbocodes duo-binaires circulaires, de schémas de codage très flexibles, pouvant s'adapter facilement à de nombreuses tailles de blocs et à une large gamme de rendements de codage et dont les performances sont tout à fait remarquables. Un effort reste cependant encore à faire concernant la réduction de la complexité de décodage, notamment pour les codes à 16 états, soit en apportant des simplifications à l'algorithme de décodage, soit en réduisant le nombre d'itérations requises par le processus turbo.

Les turbocodes duo-binaires étant particulièrement appropriés aux rendements de codage moyens ou élevés, ils s'avèrent également de bons candidats pour les transmissions à grande efficacité spectrale, où ils sont associés à des modulations à grand nombre de points. Ces associations ont notamment fait l'objet de plusieurs contrats d'étude depuis 1999. Mes travaux sur ce thème sont développés au chapitre 6.

5.3 Estimation des performances des turbocodes (2001)

5.3.1 Performances asymptotiques et distance minimale

La qualification par simulation des performances à faibles taux d'erreurs des codes correcteurs d'erreurs demande une puissance de calcul très importante. Il est possible d'estimer ces performances lorsque la *distance minimale* du code est connue. On appelle distance minimale d'un code, $d_{\min}$, la distance de Hamming minimale entre deux mots de code. Elle coïncide, pour un code linéaire, avec le poids de Hamming minimal d'un mot de code non nul. Pour un code linéaire de distance minimale $d_{\min}$, le taux d'erreurs de trames à très fort rapport signal à bruit E_b/N_0 est donné par la relation suivante :

$$FER \approx \frac{1}{2} N_{\min} erfc\left(\sqrt{R d_{\min} \frac{E_b}{N_0}}\right)$$

où $N_{\min}$ représente le nombre de mots de code de poids $d_{\min}$.

La distance minimale d'un code n'est, dans le cas général, pas simple à déterminer sauf si le nombre de mots du code est suffisamment petit pour pouvoir en dresser la liste exhaustive ou bien si des propriétés particulières du code permettent d'établir une expression analytique de cette grandeur (par exemple, la distance minimale d'un code produit est égale au produit des distances minimales des codes constituants). Malheureusement, dans le cas des turbocodes, la distance minimale ne s'obtient pas de manière analytique, les seules méthodes proposées étant basées sur l'énumération, totale ou partielle [GAR], des mots de code dont le poids d'entrée est inférieur ou égal à la distance minimale. Ces méthodes ne sont applicables en pratique que pour des tailles de blocs et des distances minimales faibles.

5.3.2 Méthode dite de l'impulsion d'erreur

En 2001, nous avons mis au point et breveté une méthode de calcul rapide de la distance minimale des turbocodes [BER7-8].

Cette méthode n'est pas basée sur l'analyse des propriétés du code, mais sur la capacité de correction du décodeur. Son principe est illustré par la Figure 5.14. Elle consiste à insérer successivement sur chaque bit i d'entrée d'une séquence de données une impulsion d'erreur dont on fait croître l'amplitude A_i jusqu'à ce le décodeur ne sache plus la corriger.

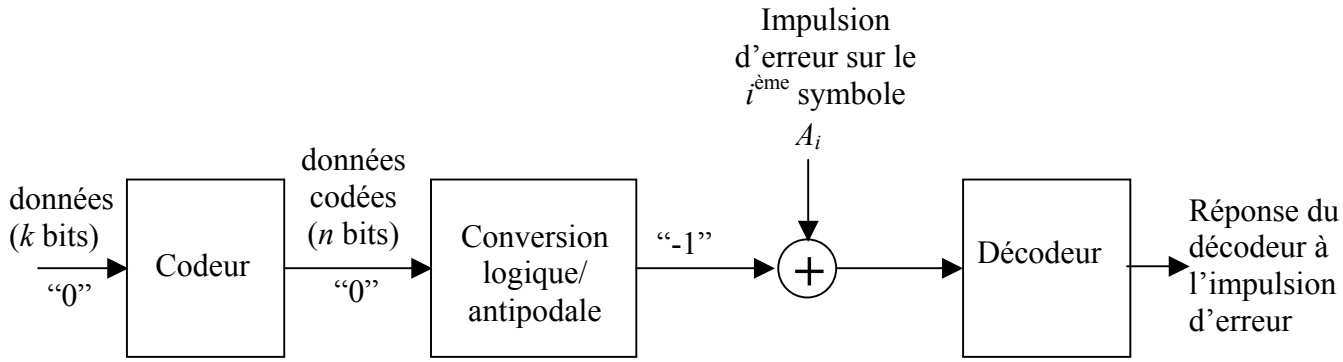


Figure 5.14 : Schéma de principe de la méthode de l'impulsion d'erreur

Le code considéré étant linéaire, nous supposons que la séquence transmise est la séquence "*tout 0*" (ne contenant que des "0" binaires). L'opération de codage produit alors des mots qui ne contiennent, eux aussi, que des "0". L'opération de modulation binaire associe ensuite un symbole "–1" à chaque "0" émis par le codeur (un symbole "+1" serait associé aux "1" émis par le codeur). Si cette succession de symboles était directement transmise au décodeur celui-ci convergerait vers le mot de code ne contenant que des "0".

La méthode proposée consiste à injecter une impulsion d'erreur sur le $i^{\text{ème}}$ symbole de la séquence d'information, c'est-à-dire à transformer un symbole "-1" en un symbole ayant une valeur positive. Si l'amplitude A_i de cette *erreur* est suffisamment importante, le décodeur ne converge pas vers le mot tout "0". L'association codeur/décodeur est alors mise en défaut. Notons A_i^* l'amplitude maximale de l'impulsion ne mettant pas en défaut le système de correction.

On peut montrer [BER8] que si le décodeur effectue un décodage à vraisemblance maximale, la *distance impulsionnelle* du code, d_{imp}, reliée aux amplitudes A_i^* par la relation :

$$d_{\text{imp}} = \min_{i=1\cdots k} (A_i^*)$$

est aussi la distance minimale, $d_{\min}$, de ce code.

En pratique, il n'est en général pas nécessaire de tester toutes les positions de la séquence de données : pour un code invariant par décalage (c'est le cas des codes convolutifs), il suffit d'appliquer l'impulsion d'erreur sur une seule position du bloc de données ; pour un code présentant une périodicité de période *P*, il est nécessaire de tester *P* positions.

Cette méthode est applicable à n'importe quel code linéaire, pour toute taille de bloc et tout rendement de codage et ne requiert que quelques secondes à quelques minutes de calcul sur un ordinateur courant, le temps de calcul étant une fonction linéaire de la taille du bloc.

En pratique, lorsque cette méthode est appliquée à un turbocode, le récepteur fait appel à un décodage itératif, qui est une version sous-optimale du décodeur à vraisemblance maximale. Aussi la distance impulsionnelle obtenue par la méthode décrite ci-dessus prend en compte les limitations du décodeur et représente la *distance minimale effective décodable* du code. Si le décodage est fortement sous-optimal, l'estimation de la distance minimale fournie par la distance impulsionnelle peut être légèrement sous-estimée : un écart allant jusqu'à 5% a pu être observé.

La méthode de l'impulsion d'erreur permet d'estimer la distance minimale du code mais les mots de code à distance minimale restent inconnus.

Il est possible d'en déduire la performance asymptotique en terme de taux d'erreurs de trames en prenant deux hypothèses [BER8]:

1. Un seul mot de code à la distance A_i^* a son $i^{\text{ème}}$ bit systématique positionné à "1",
2. Toutes les valeurs des distances A_i^* sont obtenues à partir de mots de codes distincts.

Compte-tenu de ces deux hypothèses, une estimation du taux d'erreurs de trames est donnée par :

$$FER \approx \frac{1}{2}\sum_{i=1}^{k} \text{erfc}\left(\sqrt{RA_i^* \frac{E_b}{N_0}}\right)$$

La première hypothèse sous-évalue la valeur du taux d'erreurs alors que la seconde la surévalue. Les expérimentations faites montrent que ces deux phénomènes se compensent. La Figure 5.15 compare les performances mesurées du turbocode DVB-RCS avec leur estimation par la méthode de l'impulsion d'erreur. A un taux d'erreurs de trames de 10^{-7}, moins de 0,2 dB séparent les courbes estimées et mesurées.

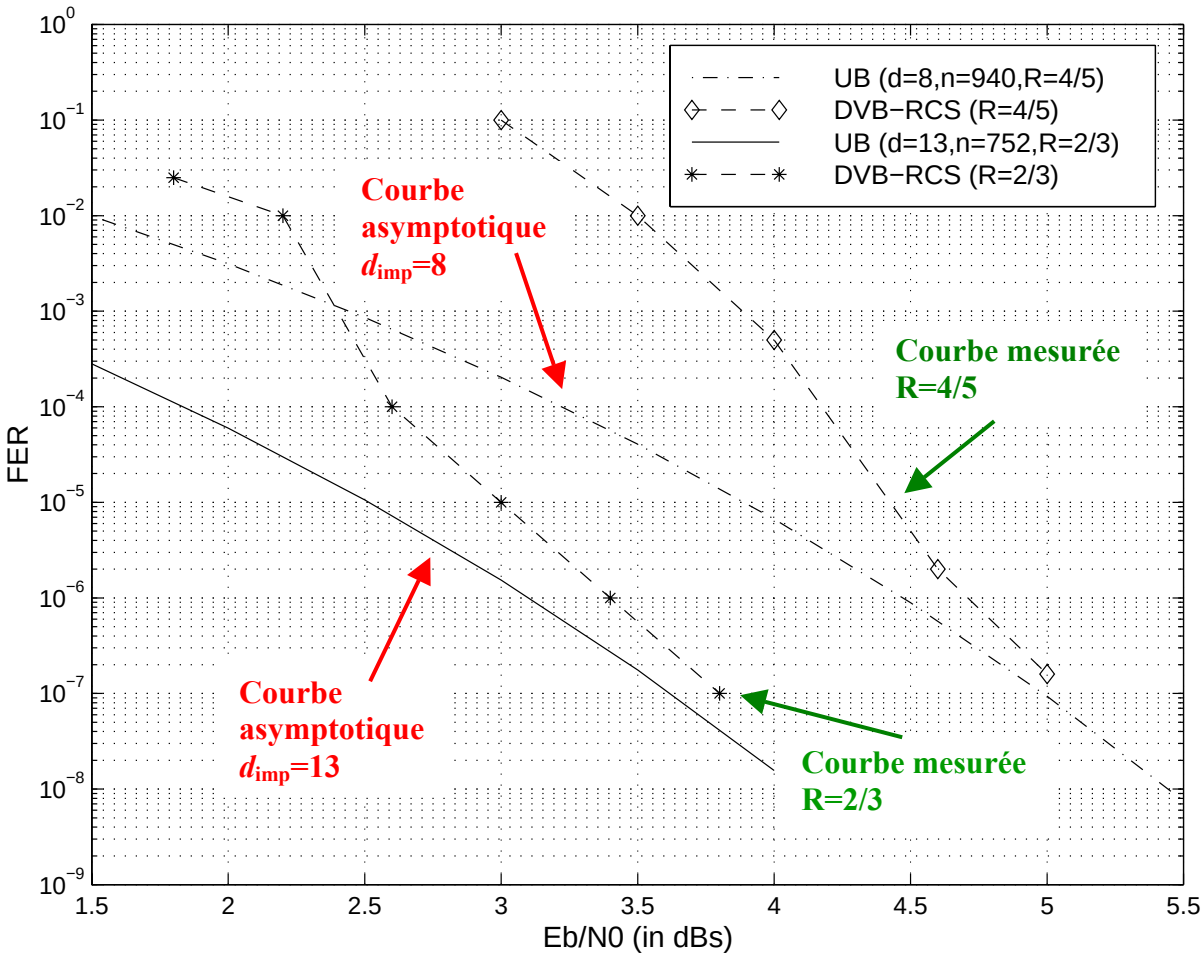


Figure 5.15 : Taux d'erreurs de trames (FER) mesuré et estimé (UB) du turbocode DVB-RCS pour la transmission de blocs MPEG (188 octets) avec des rendements de codage 2/3 et 4/5. Modulation MDP-4 et canal gaussien.

5.3.3 Conclusion

La méthode de l'impulsion d'erreur permet une détermination rapide des performances asymptotiques des turbocodes. Elle constitue ainsi un outil précieux de validation des paramètres d'entrelacement du code permettant d'obtenir de grandes distances minimales. Nous l'avons également par la suite adaptée à l'estimation des performances de modulations turbocodées à entrelacement par bits (section 6.5.2).

Le comportement du turbo-décodeur face à une séquence comportant une impulsion d'erreur n'a cependant pas encore été analysé de manière fine. Nous avons, par exemple, observé que la qualité de l'estimation de la distance minimale dépendait du nombre d'états du turbocode, ce que nous ne savons actuellement pas interpréter. La méthode mise en œuvre actuellement semble perfectible et mérite une étude plus approfondie.

5.4 Turbocodes et accès multiple à répartition par entrelacement (depuis 2003)

En 2002, Li Ping a proposé une technique innovante d'accès multiple appelée *accès multiple à répartition par entrelacement* ou IDMA (*Interleaved-Division Multiple Access*) [LIP1]. La séparation des utilisateurs y est mise en œuvre par le biais d'entrelaceurs. Cette technique, concurrente de l'accès multiple à répartition de code ou CDMA (*Code-Division Multiple*

Access), constitue une réponse au problème d'interférence entre les utilisateurs car elle n'utilise pas de séquences d'étalement : chaque séquence de données est codée puis entrelacée au niveau *chip* (un *chip* est le résultat du codage d'un bit). La structure de l'émetteur dans un système IDMA est décrit en Figure 5.16. Les entrelaceurs doivent être différents pour chaque utilisateur. Ils sont idéalement construits de manière aléatoire et indépendants les uns des autres. Ils dispersent les séquences codées de telle sorte que deux *chips* adjacents ne sont pas, ou du moins très peu, corrélés.

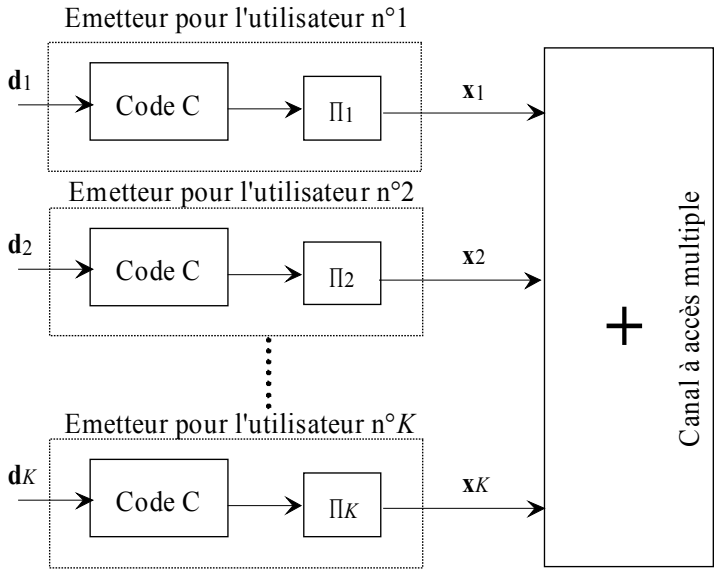


Figure 5.16 : Schéma de principe de l'émetteur dans un système IDMA à *K* utilisateurs.

En réception, le récepteur associé à l'utilisateur n° *i* effectue une détection *chip* par *chip* : les *chips* sont désentrelacés à l'aide de la permutation Π_i^{-1}, puis décodés. Un processus itératif peut avantageusement être mis en œuvre entre le décodeur et le dispositif de détection des *chips*.

La technique IDMA est une technique simple qui présente un bon comportement lorsque le nombre d'utilisateurs est important, que le canal de transmission soit à trajets multiples ou non. De fortes efficacités spectrales peuvent alors être atteintes tout en affichant des performances très proches des valeurs limites théoriques.

Pour que cette technique offre des gains de codage conséquents, il est important de choisir des codes correcteurs d'erreurs C de faible rendement. Ainsi, Li Ping a simulé des transmissions utilisant des codes convolutifs super-orthogonaux [VIT] ou des codes turbo-Hadamard [LIP2]. Dans le cadre de sa thèse, Emeric Maury étudie la pertinence de l'utilisation dans un dispositif IDMA de turbocodes à faibles rendements obtenus par une concaténation parallèle de plus de deux codes convolutifs systématiques (turbocodes multidimensionnels [BER3]) de faible complexité (typiquement $\nu = 2$).

5.5 Conclusion

Après mes premiers travaux, en 1993-1994, liés à la conception des tous premiers turbo-décodeurs, j'ai étendu en 1995 le principe du décodage itératif au problème de la correction de l'interférence entre symboles sur les canaux de transmission à trajets multiples. Par la suite, une part importante de mon activité a porté sur la recherche de turbocodes présentant

des performances de correction améliorées par rapport aux codes historiques de [BER1], tout en ayant comme contrainte leur intégration dans des systèmes de transmission réels. C'est ainsi que j'ai participé à la mise au point des codes duo-binaires, et des codes circulaires pour le codage par blocs. Ces avancées, combinées avec le développement d'outils tels que la méthode de l'impulsion d'erreur, nous ont permis de proposer des turbocodes au comportement tout à fait remarquable, aussi bien du point de la convergence que des performances asymptotiques.

Toutefois, le thème qui a occupé la plus grande place dans mes travaux depuis quatre à cinq ans concerne le problème de l'association des turbocodes avec les modulations d'ordre élevé, pour les systèmes de transmissions à forte efficacité spectrale. C'est pourquoi cette problématique n'a pas été abordée dans ce chapitre et fait l'objet d'un chapitre à part entière de ce document.

5.6 Références

[3GPP] 3GPP Technical Specification Group, *Multiplexing and Channel Coding (FDD)*, TS 25.212 v.5.0.0, 2002-2003.

[BEN] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding," *IEEE Transactions on Information Theory*, vol.44, n°3, May 1998, pp. 909-926.

[BER1] C. Berrou and M. Jézéquel, "Frame-oriented convolutional turbo-codes," *Electronics Letters*, 1996, Vol. 32, N° 15, pp. 1362-1364.

[BER2] C. Berrou, C. Douillard, M. Jézéquel, *Turbo-codes pour données transmises par salves (turbo-codes convolutif pour blocs courts)*, Rapport de contrat de recherche CCETT n° 96 ME 08, novembre 97

[BER3] C. Berrou, C. Douillard and M. Jézéquel, "Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes," *Annales des Télécomunications*, vol. 54, n°3-4, Mars-Avril 1999, pp. 166-172.

[BER4] C. Berrou, C. Douillard and M. Jézéquel, "Designing turbo codes for low error rates," *IEE Colloquium on Turbocodes in Digital Broadcasting – Could It Double Capacity?*, London, United Kingdom, Nov. 1999, pp. 1/1-1/7.

[BER5] C. Berrou, M. Jézéquel, C. Douillard and S. Kerouédan, "The advantages of non-binary turbocodes," *Information Theory Workshop, ITW'2001*, Cairns, Australia, Sept. 2001, pp. 61 - 63.

[BER6] C. Berrou, C. Douillard, M. Jézéquel et S. Vaton, *Turbo coding for SkyPlexNet*, Rapport de contrat d'étude, décembre 2000.

[BER7] C. Berrou, M. Jézéquel et C. Douillard, "Procédé de qualification de codes correcteurs d'erreurs, procédé d'optimisation, codeur, décodeur et application correspondants," Brevet n° 01 11764 du 11/09/2001, France.

[BER8] C. Berrou, S. Vaton, M. Jézéquel and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method," *IEEE Global Communication Conference Globecom'2002*, Taipei, Taiwan, Nov. 2002, pp.1017-1020.

[CCSDS]Consultative Committee for Space Data Systems, *Recommandations for Space Data Systems. Telemetry Channel Coding*, Blue Book, May 1988.

[DOL] S. Dolinar, D. Divsalar and F. Pollara, "Code performance as a function of block size," *TMO progress report 42-133, JPL, NASA*, May 1998.

[DOU1] C. Douillard, A. Picart, P. Didier, M. Jézéquel, C. Berrou and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Transactions on Telecommununications*, vol. 6, n° 5, Sept.- Oct. 1995, pp.507-512.

[DOU2] C. Douillard, A. Picart, A. Glavieux et P. Didier, *Turbo-égalisation : principes et performances*, Rapport de contrat de recherche CCETT n° 94 ME 19, octobre 1995.

[DOU3] C. Douillard, A. Glavieux, M. Jézéquel et C. Berrou, "Dispositif de réception de signaux numériques à structure itérative, module et procédé correspondants", Brevet n°95 01603 du 07/02/1995, France.

[DOU4] C. Douillard, M. Jézéquel, C. Berrou, N. Brengarth, J. Tousch and N. Pham, "The Turbocode Standard for DVB-RCS," *2nd International Symposium on Turbocodes & Related Topics*, Brest, France, Sept. 2000. pp. 535-538.

[DOU5] C. Douillard and C. Berrou, "Turbo codes with rate-$m/(m+1)$ constituent convolutional codes," à paraître dans *IEEE Transactions on Communications.*

[DVB1] DVB, "Interaction channel for satellite distribution systems," *ETSI EN 301 790, V1.2.2*, Dec. 2000, pp.21-24.

[DVB2] DVB, "Interaction channel for digital terrestrial television," *ETSI EN 301 958, V1.1.1*, Feb. 2002, pp.28-30.

[GAR] R. Garello, P. Pierleoni and S. Benedetto, "Computing the free distance of turbocodes and serially concatenated codes with interleavers: algorithms and applications," *IEEE Journal on Selected Areas in Communications*, vol. 19, n° 5, May 2001, pp. 800-812.

[GLA] A. Glavieux, C. Laot and J. Labat, "Turbo equalization over a frequency selective channel," *International Symposium on Turbocodes, Brest*, France, Sept. 1997, pp. 96-102.

[LAO] C. Laot, A. Glavieux and J. Labat, "Turbo equalization: Adaptive equalization and channel decoding jointly optimized," *IEEE Journal on Selected Areas in Communications*, vol. 19, n°9, Sept. 2001, pp. 1744-1752.

[LIP1] Li Ping, W. K. Wu, Lihai Liu and W. K. Leung, "A simple, unified approach to nearly optimal multiuser detection and space-time coding," *Information Theory Workshop ITW'2002*, Bengalore, India, Oct. 2002, pp. 53-56.

[LIP2] Li Ping, W. K. Leung and W. K. Wu, "Low-rate turbo-Hadamard codes," *IEEE Transactions on Information Theory*, vol. 49 ,n° 12, Dec. 2003, pp. 3213-24.

[PIC] A. Picart, C. Laot et C. Douillard, "Turbo-égalisation et turbo-détection", chapitre de l'ouvrage *Signal et Télécommunications*, à paraître aux éditions Hermès Science.

[PYN] R. Pyndiah, "Near optimum decoding of product codes: Block turbocodes," *IEEE Trans. Commun.*, vol. 46, n° 8, pp. 1003-1010, Aug. 1998.

[RAO1] F. Raouafi, *Adéquation turbocodes/processeurs de signal*, Thèse de Doctorat de l'Université de Bretagne Occidentale, Brest, France, Juil. 2001.

[RAO2] F. Raouafi, C. Douillard and C. Berrou, "Efficient turbo decoder design and its implementation on a low-Cost, 16-bit fixed-point DSP," *2nd International Symposium on Turbocodes & Related Topics*, Brest, France, Sept. 2000, pp. 339-342.

[ROB] P. Robertson, P. Hoeher and E. Villebrun, "Optimal and sub-optimal maximum a posteriori decoding algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol. 8, n° 2, pp. 119-125, March-April 1997.

[SHA] C.E. Shannon, "Probability of error for optimal codes in Gaussian channel," *Bell Syst. Tech. Journal*, vol. 38, pp. 611-656, 1959.

[VIT] A. J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread spectrum multiple-access channels," *IEEE Journal on Selected Areas in Communications*, vol. 8, Aug. 1990, pp. 641-649.

[WEI] C. Weiss, C. Bettstetter, S. Riedel and D. J. Costello, "Turbo decoding with tail-biting trellises," *URSI International Symposium on Signals, Systems, and Electronics, ISSSE 98*, Pisa, Italy, Oct. 1998, pp. 343-348.

6 Les modulations turbocodées (depuis 1999)

6.1 Introduction

Les échanges d'information dans les systèmes de télécommunication s'effectuent à des débits toujours plus élevés et dans des bandes de fréquences de plus en plus étroites. On cherche par conséquent à maximiser le rapport débit utile sur bande, c'est-à-dire l'efficacité spectrale des transmissions. Pour ce faire, il apparaît naturel de coupler des modulations numériques à grand nombre de points avec des codes correcteurs d'erreurs à haut rendement.

Les études menées dans ce domaine font essentiellement appel à deux approches : les *turbo-modulations codées en treillis* et les *modulations turbocodées pragmatiques*.

Les turbo-modulations codées en treillis ou TTCM (*Turbo Trellis-Coded Modulations*) ont été introduites par Robertson et Wörz en 1995 [ROB1-2]. Elles font appel à la notion de concaténation parallèle utilisée dans les turbocodes, mais appliquée à deux modulations codées en treillis ou TCM (*Trellis-Coded Modulation*) de Ungerboeck [UNG]. Dans l'approche TTCM, le code correcteur, un code convolutif systématique récursif, et le codage binaire à signal de la modulation sont représentés conjointement à l'aide d'un treillis unique. Le critère d'optimisation de la TCM consiste à maximiser la distance euclidienne minimale entre deux séquences codées. Le décodage de la TTCM résultante est similaire à celui d'un turbocode, à ceci près que, pour éviter toute perte de performance dans le décodage, le décodeur traite directement les symboles issus du démodulateur. Le passage par le calcul d'une estimation des bits de chaque symbole avant décodage constituerait, en effet, une mise en œuvre sous-optimale du récepteur. Dans la technique présentée par Robertson et Wörz, le rendement de codage visé est obtenu par poinçonnage des bits de redondance. La fonction de l'entrelacement de la TTCM ne faisant l'objet d'aucune optimisation particulière, les courbes de taux d'erreurs présentées dans [ROB1-2] font apparaître des changements de pente précoces (BER~10^{-5}) et prononcés. Une variante de cette technique proposée par Benedetto, Divsalar, Montorsi et Pollara [BEN1] a permis d'en améliorer les performances asymptotiques en répartissant judicieusement le poinçonnage entre les bits systématiques et les bits de parité. La méthode a ensuite été étendue par cette même équipe à la concaténation série de TCM [BEN2].

Les TTCM conduisent à d'excellentes performances de correction (~0,3 dB de la limite théorique à un taux d'erreurs binaires de 10^{-5}), car il s'agit là d'une approche *ad hoc*, mais qui présente une flexibilité très limitée : un nouveau code doit être défini pour chaque rendement de codage et chaque modulation considérés. Cet inconvénient est rédhibitoire dans tout système pratique requerrant une certaine flexibilité.

L'approche pragmatique est chronologiquement la première mise en œuvre. Elle a été introduite par Stéphane Le Goff [LEG1-2] dans son travail de thèse à l'ENST Bretagne de 1993 à 1995. Cette technique tient son nom de ses similarités avec la technique d'association code convolutif/modulation adoptée par A. Viterbi [VIT] comme solution alternative aux TCM d'Ungerboeck. Elle ne nécessite pas d'optimisation conjointe du code et de la modulation. Elle utilise un "bon" turbocode, un codage binaire à signal de la modulation considérée qui minimise la probabilité d'erreur binaire en sortie du canal (codage de Gray) et associe les deux par le biais d'une opération de poinçonnage et de multiplexage pour adapter l'ensemble à l'efficacité spectrale visée. En réception, un turbo-décodeur standard est utilisé, ce qui nécessite une estimation de chaque bit contenu dans les symboles démodulés. Une extension de cette méthode à la concaténation série a été proposée par Benedetto et Montorsi [BEN3]. L'approche pragmatique conduit à des performances faiblement dégradées, soit 0,3 à

0,5 dB de perte, par rapport à l'approche *ad hoc* TTCM, qui ne vaut de toute façon que pour des rendements de codage particuliers.

Cette seconde approche a été retenue dans l'ensemble de nos études, car elle présente des performances tout à fait satisfaisantes pour une très grande flexibilité, un facteur primordial dans nombre d'applications pratiques.

J'ai abordé le domaine des modulations turbocodées en 1999, à l'occasion d'un contrat d'études avec le CNET de Grenoble [PIC]. Il s'agissait d'effectuer une étude comparée de deux familles de turbocodes, la première à base de codes convolutifs et la seconde à base de produit de codes BCH, pour des applications de communications radio-mobiles à haut débit (25 Mbit/s de débit utile) en environnement intérieur. La comparaison des deux schémas de codage portait à la fois sur les performances de correction et sur la complexité de réalisation matérielle du décodeur. J'avais en charge la partie de l'étude portant sur la proposition à base de codes convolutifs. Les performances des schémas proposés ont été évaluées pour des transmissions sur canal de Rayleigh, avec une modulation d'amplitude en quadrature à 16 états, ou MAQ-16. En résumé, le turbocode convolutif duo-binaire à 8 états proposé présentait les meilleures performances de correction, avec un écart de 1 dB environ à nombre d'itérations identique, pour une complexité en contre-partie double de celle du code produit proposé.

Cette première étude a permis de mettre en évidence un certain nombre de thèmes à approfondir par la suite, notamment :

- l'étude de l'influence sur les performances de correction de la stratégie de construction des symboles à partir des bits codés, en fonction des canaux de transmissions et des modulations considérées ;
- l'évaluation du degré de sous-optimalité vis-à-vis des limites théoriques des schémas de modulations turbocodées considérés ;
- l'estimation de leurs performances asymptotiques.

Le sous-chapitre qui suit présente dans le détail le principe des modulations turbocodées pragmatiques.

Les sous-chapitres 6.3 et 6.4 présentent les résultats obtenus dans le cadre de deux contrats de recherche récents : les performances de modulations turbocodées sont comparées en fonction de la stratégie de construction des symboles à partir des bits codés, des canaux de transmission, de la modulation, du rendement de codage et du taux d'erreurs visés.

Le sous-chapitre 6.5 présente notre apport au problème de l'estimation des performances asymptotiques des modulations turbocodées, basé sur la méthode de l'impulsion d'erreur présentée au chapitre précédent, ainsi qu'à l'évaluation des limites théoriques correspondantes.

6.2 Schéma de principe de l'approche pragmatique des modulations turbocodées

La Figure 6.1 présente le schéma général de principe de l'émetteur et du récepteur dans le cadre de l'association pragmatique d'un turbocode et d'une modulation à $M = 2^m$ états.

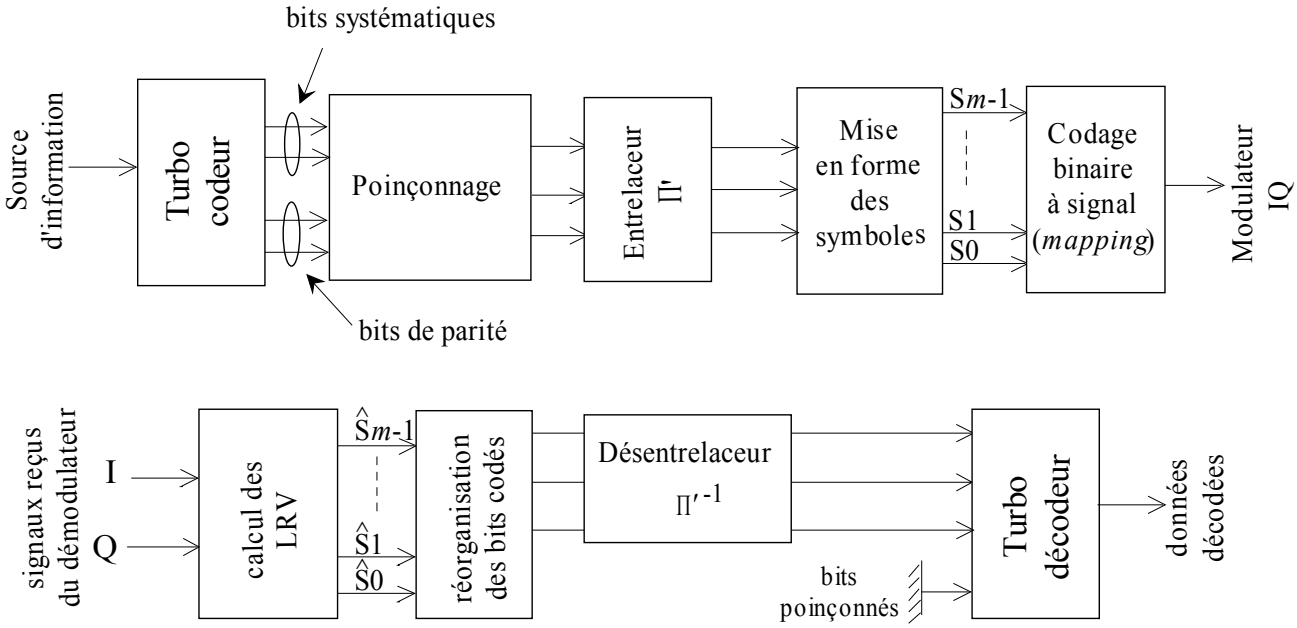


Figure 6.1 : Schéma de principe de l'émetteur et du récepteur dans le cas de l'association pragmatique d'un turbocode et d'une modulation à $M = 2^m$ états.

Le codeur et le décodeur sont des turbocodeur et décodeur standard, identiques pour l'ensemble des tailles de blocs transmis, des rendements de codage et des modulations considérées. Seule la fonction de permutation du code est paramétrable en fonction de la taille de bloc.

Dans la thèse de Le Goff, l'approche pragmatique avait été mise en œuvre sur des blocs de grande taille avec un turbocode binaire et l'algorithme de décodage SOVA. Nous l'avons transposé dans nos études au cas des turbocodes circulaires duo-binaires avec l'algorithme de décodage Max-Log-MAP.

Un turbo-décodeur standard étant utilisé, une estimation des bits portés par chaque symbole modulé doit être calculée en sortie du démodulateur, pour pouvoir assurer le décodage. L'estimation pondérée de chaque bit $\hat{S}_l$ est obtenue par le calcul du Logarithme du Rapport de Vraisemblance (LRV ou LLR pour *Logarithm of Likelihood Ratio*) défini par :

$$\hat{S}_l = \Lambda(S_l) = \frac{\sigma^2}{2} \ln \frac{\Pr(S_l = 1 \mid I, Q)}{\Pr(S_l = 0 \mid I, Q)}$$

Dans le cas d'une transmission sur canal gaussien, $\hat{S}_l$ peut s'écrire :

$$\hat{S}_l = \frac{\sigma^2}{2} \ln \frac{\sum_{m_1} \exp\left(-\frac{d_{m_1}^2}{2\sigma^2}\right)}{\sum_{m_0} \exp\left(-\frac{d_{m_0}^2}{2\sigma^2}\right)}$$

où m_1 (m_0) représente l'ensemble des points de la modulation tels $S_l = 1$ ($S_l = 0$), et d_{m_1} (d_{m_0}) est la distance euclidienne entre le symbole reçu et le point de la constellation considéré.

En pratique, nous appliquons l'approximation Max-Log pour le calcul des $\hat{S}_l$:

$$\ln(\exp(a)+\exp(b)) \approx \max(a,b)$$

et chaque LLR est calculé comme :

$$\hat{S}_l = \frac{1}{4}\left(\min_{m_0}(d_{m_0}^2) - \min_{m_1}(d_{m_1}^2)\right)$$

Lorsque le rendement de codage visé est supérieur au rendement naturel du turbocode, l'opérateur de poinçonnage permet d'effacer, c'est-à-dire de ne pas transmettre, certains bits codés. En réception, un dispositif vient insérer un zéro analogique en entrée du décodeur aux places correspondantes. En pratique, pour des raisons de commodité de réalisation matérielle, le motif de poinçonnage est périodique ou quasi-périodique.

Si possible, seuls les bits de parité sont poinçonnés. En effet, un poinçonnage des bits systématiques entraîne une dégradation rapide du seuil de convergence de décodage, car ces bits participent au processus de décodage des deux codes, contrairement aux bits de redondance. Lorsque le rendement de codage est élevé, un léger poinçonnage des données peut néanmoins améliorer le comportement asymptotique du système codé.

La présence des fonctions de permutation temporelle Π' et Π'^{-1} se justifie par le besoin de décorréler les LLRs calculés en réception pour que le turbo-décodage soit le plus efficace possible. En fait, Le Goff a montré dans sa thèse que l'insertion de cet entrelacement n'a pas d'effet significatif sur le taux d'erreurs en sortie du décodeur, dans le cas d'un transmission sur canal gaussien. En revanche, dans le cas de canaux à évanouissements, l'entrelacement est nécessaire car il s'agit d'éviter que les bits issus d'un même instant de codage ne soient dans le même symbole émis sur le canal, afin de ne pas être affectés simultanément par un évanouissement.

Le code et la modulation n'étant pas conjointement optimisés, à la différence d'un schéma TTCM, le meilleur codage binaire à signal des points de la constellation est celui qui minimise le taux d'erreurs binaires moyen à l'entrée du décodeur. Ceci est dû au fait que le seuil de convergence du processus itératif est directement lié à la valeur du taux d'erreurs binaires à l'entrée du décodeur. Un codage de Gray, lorsqu'il est envisageable, satisfait cette condition. Par souci de simplicité de mise en œuvre du modulateur et du démodulateur, dans le cas des modulations d'amplitude en quadrature ou MAQ carrées (m pair), les voies en phase et en quadrature, I et Q, sont codées indépendamment.

Sur la Figure 6.1, le bloc "mise en forme des symboles" décrit l'organisation des bits codés, systématiques et redondants, dans les symboles de modulation. En effet, parmi l'ensemble des modulations étudiées, seules les modulations à deux ou quatre points offrent le même niveau de protection à tous les bits d'un même symbole. Pour des modulations à plus grand nombre de points, certains bits sont mieux protégés que d'autres. La fonction de mise en forme des symboles est ainsi directement liée à la stratégie adoptée pour répartir les bits systématiques et les bits de redondance suivant le niveau de protection lié à la modulation. Dans son étude, Le Goff était arrivé à la conclusion suivante :

- à faibles rapports signal à bruit, les performances sont d'autant meilleures que les bits systématiques sont bien protégés par rapport aux bits de redondance ;

– à forts rapports signal à bruit, le pouvoir de correction de l'association turbocode/modulation devient indépendant de la répartition des bits systématiques et de redondance dans les symboles.

Les études que nous avons menées sur ce même problème nous ont conduit à des conclusions quelque peu différentes. En effet, en 1995, la puissance de calcul du parc informatique du département Electronique était très inférieure à la puissance actuelle et était alors insuffisante pour la vérification du comportement à faible bruit de l'association d'un turbocode et d'une modulation à huit points ou plus. Avec les moyens de simulation actuels, nous avons montré que la répartition des bits systématiques et de redondance dans les symboles influe également sur les performances asymptotiques du système. Ce problème a été étudié dans le cadre d'un contrat de recherche avec France Telecom R&D et constitue une partie des résultats de la thèse de Laura Conde Canencia.

6.3 Étude du *mapping* optimal des turbocodes sur des constellations à grand nombre d'états : contrat de recherche externe France Telecom [JEZ] (2001-2002)

L'objectif de cette étude consistait à optimiser les performances des modulations turbocodées pragmatiques, dans le contexte des nouveaux systèmes de diffusion et de radio-communications. Ceux-ci sont caractérisés par une taille de bloc relativement courte (quelques centaines à quelques milliers de bits), un rendement de codage variable et l'emploi de modulations à grand nombre d'états (typiquement MAQ-16 et MAQ-64).

Cette étude a été menée en parallèle sur deux familles de turbocodes : des turbocodes en blocs, en l'occurrence des codes produits de codes BCH [PYN], et les turbocodes convolutifs duo-binaires et circulaires présentés dans le sous-chapitre 5.2. J'avais la charge de la partie traitant des turbocodes convolutifs. Le travail effectué a essentiellement porté sur l'étude détaillée de la construction des symboles à partir des bits codés. Deux zones de taux d'erreurs étaient à considérer : taux d'erreurs moyens (*Frame Error Rate*, FER~10^{-2}) et très faibles (*Quasi Error Free*, QEF, c'est-à-dire FER < 10^{-8}). En pratique, nos moyens de calcul ne nous ont permis de valider les solutions retenues pour le cas QEF que jusqu'à des taux d'erreurs de trames de 10^{-6}, voire 10^{-7}. D'autre part, les schémas retenus devaient être évalués sur deux canaux de transmission de référence : le canal à bruit additif blanc gaussien et le canal de Rayleigh sans mémoire.

Nous avons défini deux stratégies de base pour la construction des symboles :

– **schéma A** : les bits les mieux protégés par la modulation sont associés en priorité aux bits systématiques ;

– **schéma Y** : les bits les mieux protégés par la modulation sont associés en priorité aux bits de redondance.

D'autre part, indépendamment de ces schémas, une méthode de construction particulière peut être adoptée dans certains cas particuliers de rendements de codage et de modulations. Cette méthode est à rapprocher du principe des TCM [UNG] : l'ensemble des bits issus du codeur à un instant donné sont transmis dans un même symbole de modulation. Ainsi, l'information portée par chaque branche des treillis de codage est entièrement contenue dans un seul symbole modulé transmis.

Ceci n'est possible que pour des cas particuliers de rendements et de modulations : en effet, le nombre de bits issus du codeur à chaque instant doit être égal à $m = \log_2 M$, si M est le nombre de points de la modulation. Nous avons appelé **TTCM pragmatique** ce cas

particulier de modulation turbocodée. A titre d'exemple, le principe de la TTCM pragmatique peut être appliqué dans le cas de l'association d'un turbocode duo-binaire de rendement 2/3 avec une modulation à 8 états (voir Figure 6.2), ou de rendement 1/2 avec une modulation à 16 états.

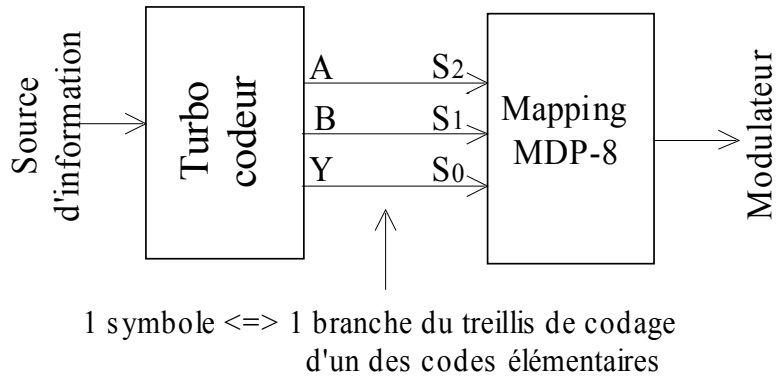


Figure 6.2 : Exemple de TTCM pragmatique associant un turbocode duo-binaire de rendement 2/3 et une modulation MDP-8.

Nous avons étudié l'influence sur le taux d'erreurs de la répartition des bits codés dans les symboles pour le codage de blocs ATM (53 octets), avec les rendements 1/2, 9/16 et 3/4. La transmission utilise une modulation MAQ-16, avec codage de Gray sur les voies en phase et en quadrature. Le codage binaire à signal retenu conduit à deux niveaux de protection des bits dans les symboles de modulation, associés chacun à la moitié des bits transmis.

6.3.1 Cas des taux d'erreurs moyens

Pour les taux d'erreurs moyens et élevés, un turbocode à base de codeurs à huit états, dérivé du code DVB-RCS a été proposé. Nous avons en effet observé qu'avec un choix étudié des codes élémentaires, la différence de performance entre un code à huit ou seize états n'est pas significative à ces taux d'erreurs (voir Figure 5.13). Concernant la répartition des bits codés dans les symboles, nous sommes arrivés à la même conclusion que Le Goff : aux taux d'erreurs forts et moyens, le pouvoir de correction du turbocode est lié à la valeur du seuil de convergence du processus itératif, qui est plus petit pour un schéma A que pour un schéma Y. En effet, dans le processus de décodage, chaque donnée systématique est utilisée en entrée des deux décodeurs. Par conséquent, une erreur sur un bit systématique en sortie du canal provoque une erreur sur l'entrée des deux décodeurs élémentaires, alors qu'une redondance erronée n'affecte l'entrée que d'un des deux décodeurs élémentaires. Aussi, un renforcement de la protection des bits systématiques entraîne une meilleure convergence du processus itératif de décodage. Néanmoins, l'écart de performance entre les deux schémas proposés est fortement dépendant du rendement de codage : à un taux d'erreurs de trames de 10^{-4}, il varie de 0,6 dB pour $R = 1/2$ à 0,1 dB pour $R = 3/4$. Il s'agit là d'un résultat prévisible puisque, à fort rendement du code, le nombre de bits de redondance transmis est faible et les deux schémas A et Y conduisent à des répartitions des bits codés proches.

6.3.2 Cas des taux d'erreurs faibles

Pour les taux d'erreurs faibles et très faibles, un turbocode à base de codeurs à seize états tel que celui décrit en section 5.2.8 a été proposé, afin d'augmenter les distances minimales et le gain asymptotique.

6.3.3 Performances sur canal gaussien

Les résultats obtenus pour une transmission sur canal gaussien sont présentés sur la Figure 6.3 et 7.4. Nous pouvons observer un comportement différent de celui décrit par Le Goff : à faible bruit, le comportement de la modulation turbocodée pragmatique est également dépendant de la stratégie de répartition des bits codés dans les symboles et le schéma Y procure les meilleures performances. L'interprétation de ce résultat fait appel à l'analyse des chemins erronés dans les treillis à fort rapport signal à bruit. Nous avons observé que, dans la majorité des cas, les séquences erronées contiennent un nombre plutôt faible de bits systématiques erronés et un nombre plutôt élevé de bits de redondance erronés, autrement dit les séquences erronées présentent en général un poids d'entrée faible. En particulier, les chemins erronés en question correspondent pour la plupart à des motifs d'erreurs rectangulaires [BER1]. Il en résulte que, du point de vue du comportement asymptotique de la modulation turbocodée, le schéma Y procure les meilleures performances, car il assure une meilleure protection des bits de parité.

Ce comportement est difficile à mettre en évidence par simulation pour les rendements les plus faibles, car le point alors supposé de croisement des courbes est situé à un taux d'erreurs difficile à mesurer (FER ≈ 10^{-8} pour $R = 1/2$).

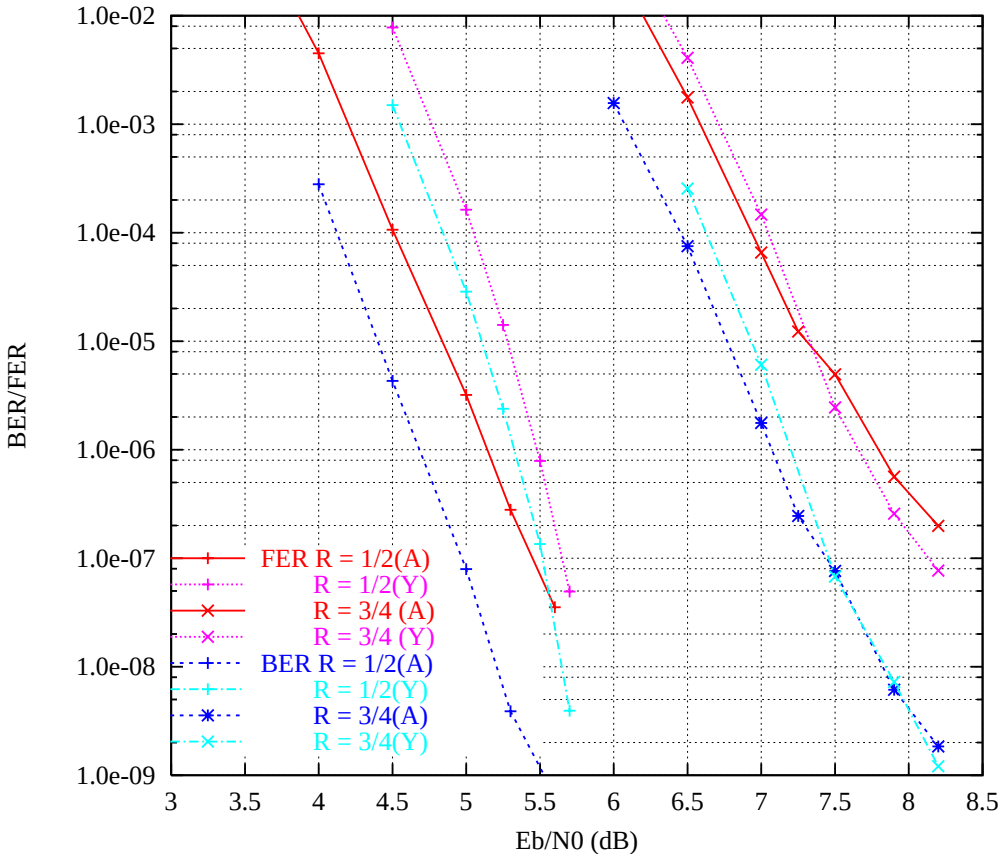


Figure 6.3 : Performance sur canal gaussien de l'association pragmatique d'une MAQ-16 et d'un turbocode duo-binaire 16 états, pour la transmission de blocs de 53 octets. Rendements de codage 1/2 et 3/4. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage.

Les courbes correspondant à un rendement de codage $R = 1/2$ ont été obtenues à partir de schémas de TTCM pragmatiques : chaque symbole transmis est constitué de deux bits systématiques A et B, et des deux bits de redondance Y_1 et Y_2 issus du codage de A et B par les deux codeurs élémentaires. Ainsi, chaque symbole modulé contient les informations binaires portées par une branche de chaque treillis de codage. Nous avons comparé les

performances de ce schéma avec celles d'un schéma pour lequel les bits contenus dans chaque symbole n'ont pas de lien direct de codage entre eux, mais en conservant la même stratégie de répartition des bits systématiques et redondants. La TTCM pragmatique s'avère plus performante : un gain de codage de près de 0,2 dB peut être observé à un taux d'erreurs de trames de 10^{-5}. Ce gain peut s'expliquer par le fait que chaque erreur de transmission n'affecte directement qu'un instant de codage, alors que si les bits codés sont répartis sur plusieurs symboles, un symbole erroné entraîne des erreurs dans le treillis sur plusieurs instants de codage.

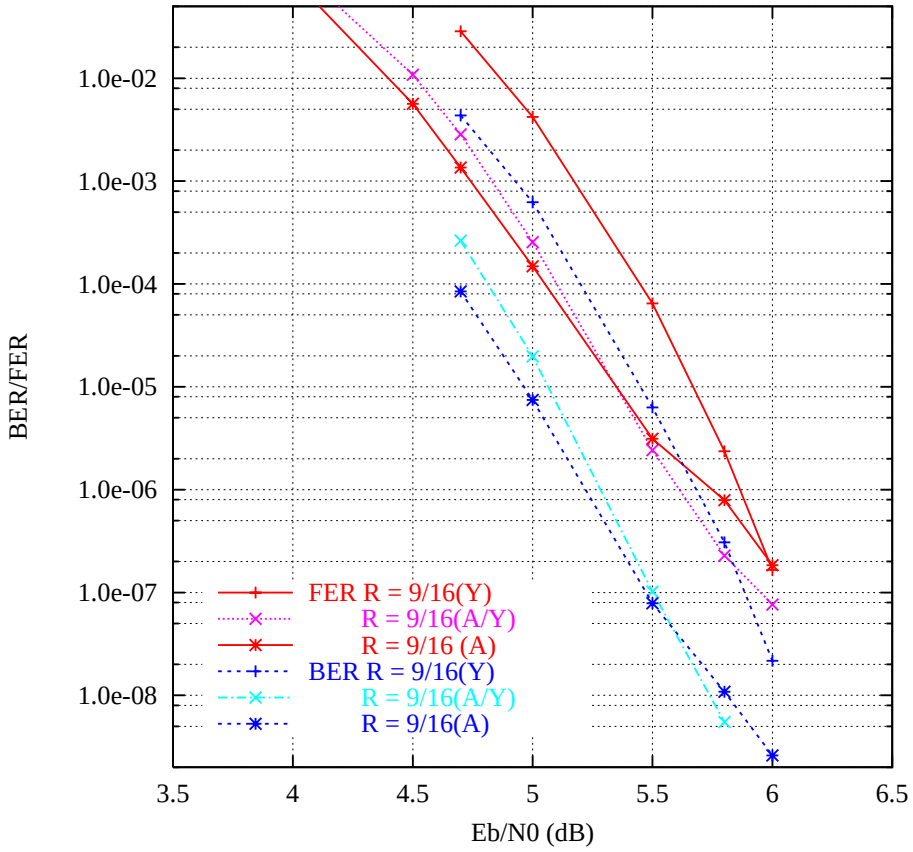


Figure 6.4 : Performance sur canal gaussien de l'association pragmatique d'une MAQ-16 et du code duo-binaire 16 états, pour la transmission de blocs de 53 octets. Rendement de codage 9/16. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage.

Dans le cas $R = 9/16$, nous avons simulé un troisième schéma intermédiaire entre les schémas A et Y , dit schéma "mixte A/Y". Dans ce cas, un quart des bits de redondance ont été associés aux places les mieux protégées. Nous avons ainsi obtenu une modulation turbocodée présentant des performances intermédiaires de celles des schémas A et Y. La courbe de taux d'erreurs de trames correspondante affiche une dégradation du seuil de convergence par rapport au schéma A, qui reste toutefois faible (environ 0,1dB), tandis que le changement de pente lié à l'atteinte du gain asymptotique est retardé d'une décade.

Pour les trois rendements de codage étudiés, le schéma de modulation turbocodée proposé permet de transmettre les données à un taux d'erreurs de trames de 10^{-6}, à moins de 0,9 dB de la limite théorique contrainte par la taille [DOL], sachant que nous avons mis en œuvre l'algorithme de décodage simplifié Max-Log-MAP sur des données quantifiées sur six bits en entrée.

6.3.4 Performances sur canal de Rayleigh

Cette modulation turbocodée a également été simulée pour des transmissions sur un canal de Rayleigh sans mémoire. Pour ce type de canal, un entrelacement au niveau bit Π' (cf. Figure 6.1) est inséré dans la chaîne de transmission, afin d'éviter qu'un évanouissement ne puisse pas affecter l'ensemble des bits issus du codeur au même instant. Cette technique est connue sous la dénomination *Bit-Interleaved Coded Modulation*, ou BICM [CAI]. Elle présente, pour les canaux à évanouissements, des performances supérieures à celles d'une TTCM associée à un entrelacement au niveau symbole [ZEH]. Nous avons donc mis en œuvre cette technique combinée aux deux stratégies de construction des symboles présentées précédemment.

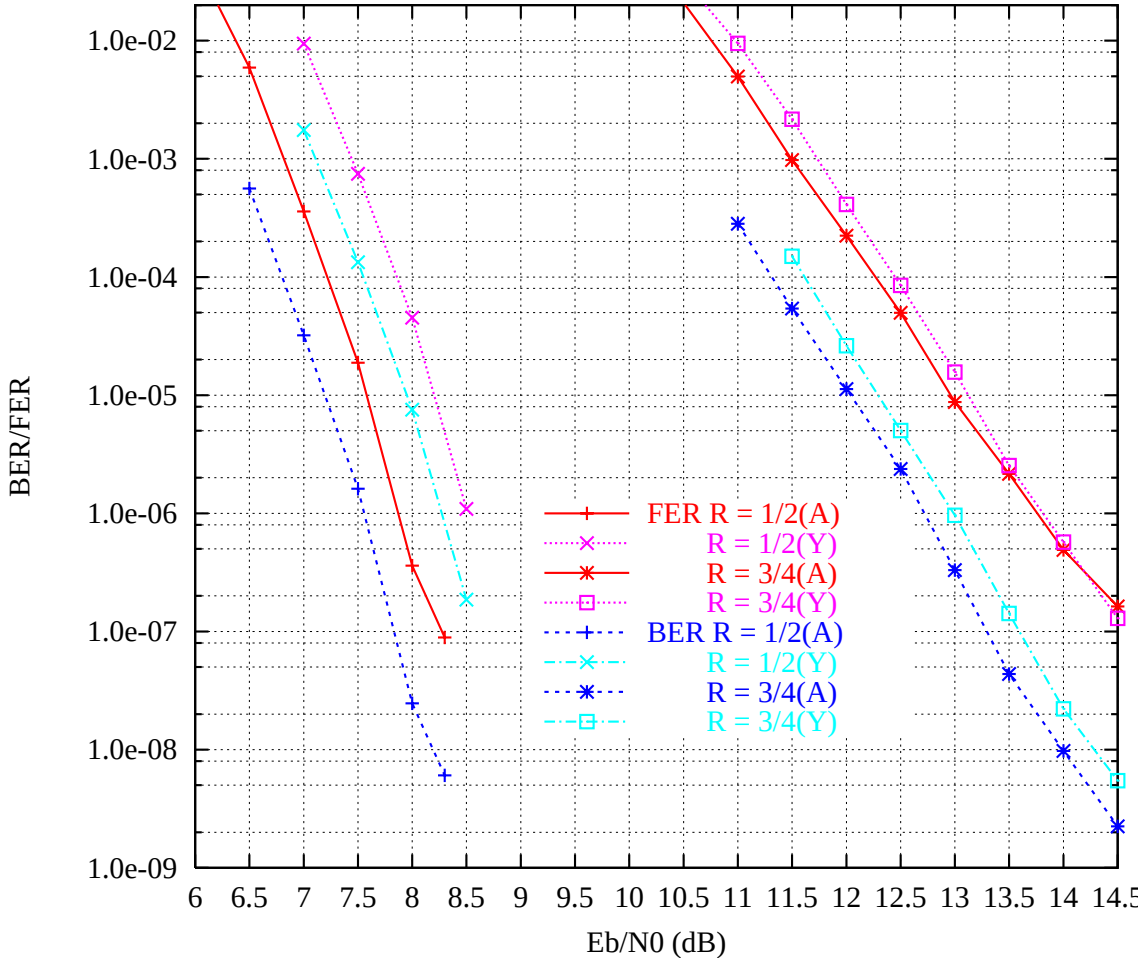


Figure 6.5 : Performance sur canal de Rayleigh de l'association pragmatique d'une MAQ-16 et du code duo-binaire à 16 états, pour la transmission de blocs de 53 octets. Rendements de codage 1/2 et 3/4. Décodage Max-Log-MAP, entrées du décodeur codées sur 6 bits, 8 itérations de décodage.

Les résultats obtenus ont montré un comportement similaire au cas gaussien vis à vis du seuil de convergence du décodage itératif : à un taux d'erreurs de trames de 10^{-4}, on observe des écarts de performance entre les schémas A et Y du même ordre de grandeur que sur canal gaussien pour les rendements simulés (voir Figure 6.5).

En revanche, les comportements asymptotiques diffèrent : pour le cas $R = 1/2$, les deux courbes simulées restent approximativement parallèles pour la gamme de rapports signal à bruit simulés. Pour $R = 3/4$, les deux courbes se rejoignent et semblent même se croiser à un

taux d'erreurs de trames de 10^{-7} environ, mais ce croisement n'est pas visible sur les courbes de taux d'erreurs binaires. L'intérêt d'une meilleure protection des bits de redondance n'apparaît donc pas ici de manière aussi claire que dans le cas gaussien.

6.3.5 Conclusion

Une étude similaire a été menée dans le cadre de ce même contrat par Laura Conde Canencia pour un schéma de transmission sur canaux gaussiens et de Rayleigh utilisant la modulation MAQ-64, pour les rendements de codage 3/5 et 2/3. Dans le cas de la MAQ-64, les niveaux de protection des bits sont au nombre de trois, ce qui permet d'enrichir l'ensemble des stratégies basiques de construction des symboles à partir des bits codés. Toutefois, les conclusions générales de cette partie de l'étude sont similaires à celles obtenues pour la modulation MAQ-16.

Les difficultés rencontrées pour l'obtention par simulation de Monte Carlo des performances de correction des schémas étudiés à des taux d'erreurs de trames inférieurs à 10^{-7} nous ont amenés à étudier le problème de l'estimation des performances asymptotiques des modulations turbocodées. Ces travaux, qui font appel à la méthode de l'impulsion d'erreur décrite dans la section 5.3.2, sont présentés en section 6.5.2.

Les résultats, légèrement surprenants, obtenus sur canal de Rayleigh n'ayant pas fait l'objet d'une analyse poussée dans le cadre de ce contrat, nous avons démarré en octobre 2003 une étude sur le comportement des turbocodes et des modulations turbocodées sur les canaux à évanouissement. Il s'agit du sujet de thèse de Charbel Abdel Nour.

D'autre part, en 2003, un contrat avec l'Agence Spatiale Européenne nous a permis d'étendre cette étude à d'autres modulations. Les travaux réalisés au cours de ce projet sont succinctement décrits dans le sous-chapitre suivant.

6.4 Projet MHOMS : MOdems for High-Order Modulation Schemes [BER2-3] (avril 2003 à janvier 2004)

Ce projet, financé par l'Agence Spatiale Européenne (ESA), a pour objectif la définition et la réalisation d'un démonstrateur de modem numérique reconfigurable au vol, pour liaisons satellitaires à haut débit (jusqu'à 1 Gbit/s pour certaines configurations, 8 Mbit/s pour la voie retour) couvrant une large gamme d'efficacités spectrales (de 0,5 à 5,4 bit/Hz/s). Les contraintes de performances requises pour l'ensemble des modes opérationnels sont très sévères : performances de correction à 1 dB des limites théoriques, sans changement de pente significatif jusqu'à 10^{-6} de taux d'erreurs de trames. D'autre part, les contraintes de flexibilité au niveau de la taille des blocs et des rendements de codage sont ici beaucoup plus fortes que dans le projet décrit précédemment.

Nous avons été chargés de proposer un schéma de modulation turbocodée pour les trois modulations suivantes proposées par l'ESA : MDP-4, MDP-8 et MDAP-16 (Modulation à Déplacement d'Amplitude et de Phase à 16 états), permettant de satisfaire les contraintes de performance citées ci-dessus, tout en offrant des solutions architecturales de décodage permettant de viser les débits spécifiés.

Nous avons proposé une association pragmatique utilisant notre code duo-binaire circulaire à seize états. Nous avons eu à valider pour ce projet les vingt configurations de transmission de la voie retour du système. Sept tailles de blocs d'information ont été considérées, comprises entre 128 à 3008 bits. Le rendement de codage variant entre 1/3 et 0,65278, chaque codeur élémentaire a du être pourvu d'un second bit de redondance pour les rendements inférieurs à

1/2. En raison des valeurs quelque peu “exotiques” des rendements de codage spécifiés, nous avons du modifier notre technique de poinçonnage périodique pour la rendre adaptable à toute valeur de rendement.

La réponse aux contraintes de débit imposées a été apportée par les propriétés de construction de la permutation, ou entrelacement, du turbocode. Les équations de permutation inter-symbole définies pour le code fournissent, en effet, un moyen naturel de mettre en œuvre un haut niveau de parallélisme dans le processus de décodage. La permutation proposée est de la forme :

$$i = \Pi(j) = (Pj + Q + 13) \bmod N$$

où N représente le nombre de couples de bits à coder, i (i = 0, …, N-1) est l’adresse du couple dans l’ordre naturel (non entrelacé) et j (i= 0, …, N-1) l’adresse du couple dans l’ordre entrelacé. Q est un paramètre dont la valeur est donnée par :

- si j mod 4 = 0, $Q = 0$;
- si j mod 4 = 1, $Q = 4Q_1$;
- si j mod 4 = 2, $Q = 4Q_0P+4Q_2$;
- si j mod 4 = 3, $Q = 4Q_0P+4Q_3$.

Le paramètre P est un entier premier avec N. Q_0, Q_1, Q_2, et Q_3 sont des petits entiers (typiquement compris entre 0 et 16). Un jeu de paramètres est défini pour chaque valeur de N.

Les équations précédentes font apparaître un comportement cyclique de cet entrelacement, de cycle quatre : deux positions j et $j + 4$ dans l’ordre naturel présentent un écart constant après entrelacement (ici égal à $4P$), quelle que soit la valeur de j. Grâce à cette propriété, il est possible d’affecter au décodage de chaque bloc de données quatre décodeurs élémentaires travaillant en parallèle, aussi bien pour la séquence dans l’ordre naturel que pour la séquence dans l’ordre entrelacé, sans augmenter la taille de la mémoire nécessaire à la mémorisation des données et des informations extrinsèques. Des degrés de parallélisme plus importants de valeur $4p$ sont envisageables, dès lors que N est un multiple de p.

Le détail des caractéristiques techniques du schéma proposé est décrit dans la *datasheet* [BER3], reproduite en Annexe 8 de ce document.

Les modulations MDP-8 et MDAP-16 offrent deux niveaux de protection des bits dans chaque symbole. Pour satisfaire les contraintes de performance de correction imposées dans les spécifications, nous avons recherché, pour chacun des cas traités, le meilleur compromis entre convergence et comportement asymptotique : nous avons déterminé par simulation le taux minimal de bits redondance à attribuer aux places les mieux protégées par la modulation, pour éviter tout changement de pente prononcé dans les courbes de taux d’erreurs en deçà de $10^{-6}/10^{-7}$ de FER.

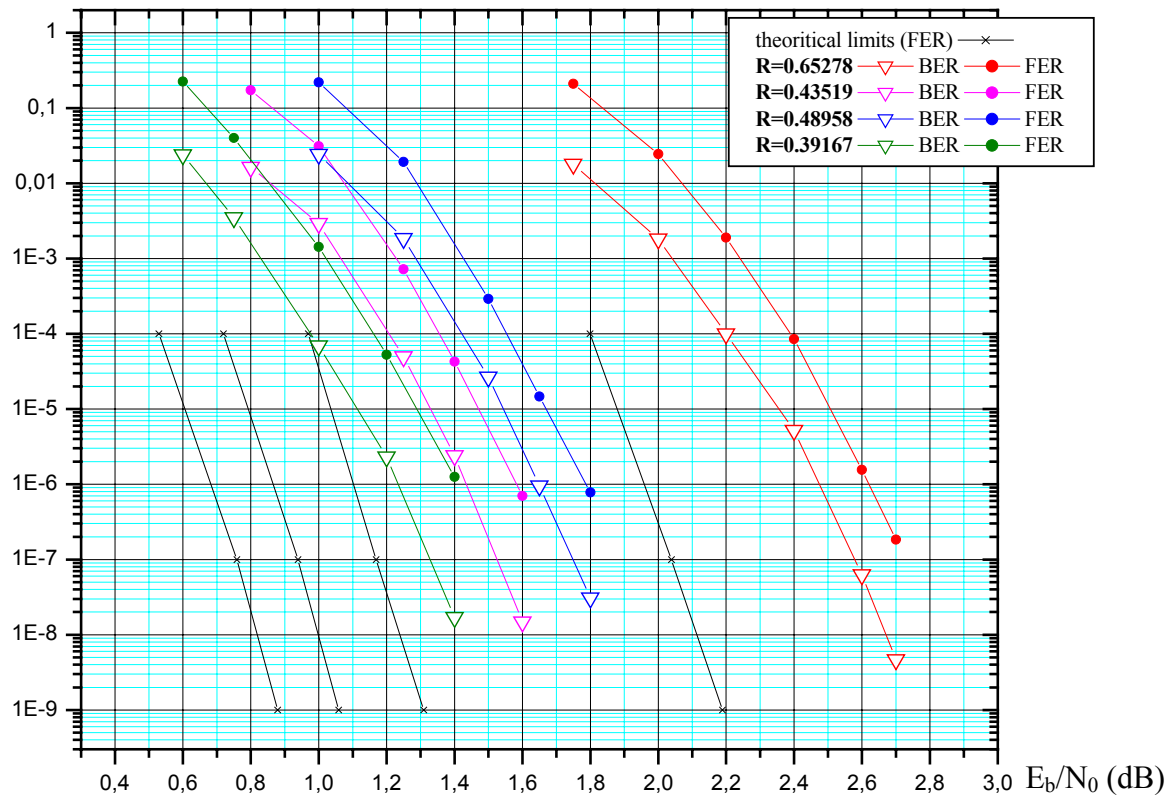


Figure 6.6 : Performance de l'association d'un turbocode duo-binaire à 16 états avec une modulation MDP-4 sur canal gaussien. Décodage Max-Log-MAP, 6 bits de quantification en entrée du décodeur, 8 itérations. Taille des blocs : 188 octets, rendements *R*=0,39167, 0,43519, 0,48958 et 0,65278.

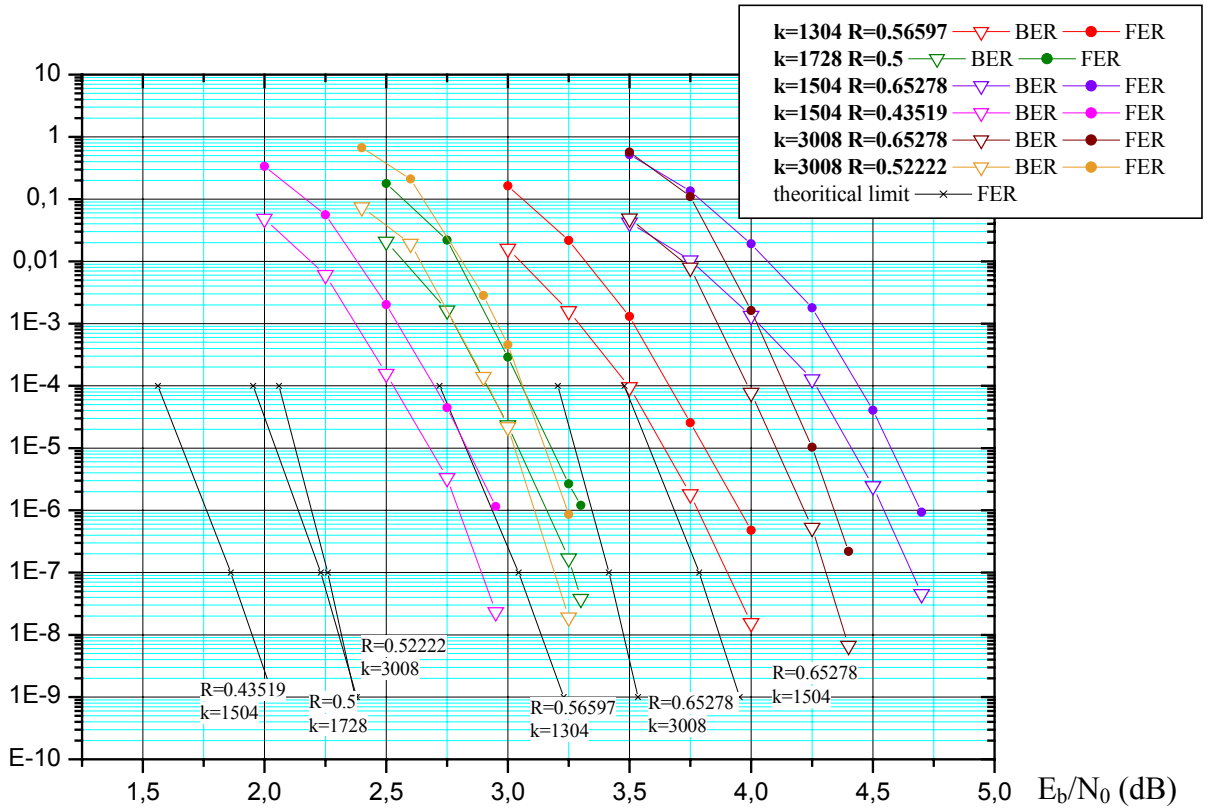


Figure 6.7 : Performance de l'association d'un turbocode duo-binaire à 16 états avec une modulation MDP-8 sur canal gaussien. Décodage Max-Log-MAP, 6 bits de quantification en entrée du décodeur, 8 itérations. Taille des blocs : 163, 188 et 376 octets, rendements *R*=0,56597, 0,5, 0,65278 et 0,52222.

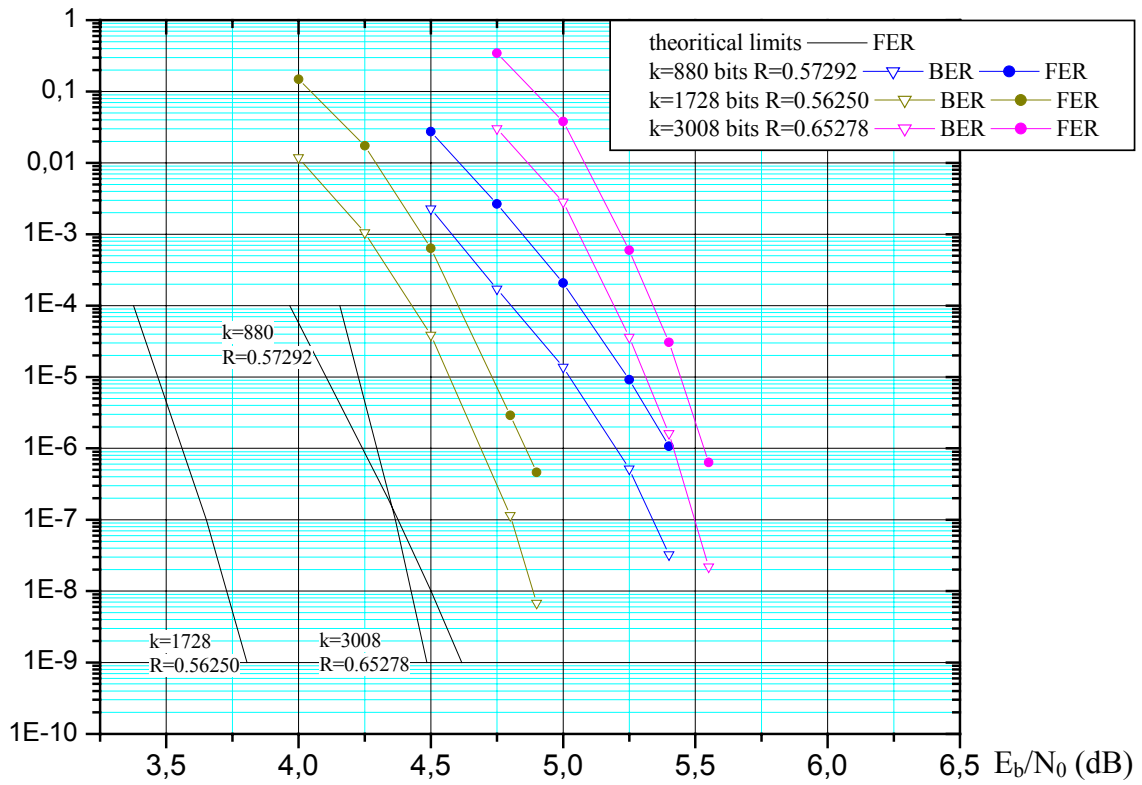


Figure 6.8 : Performance de l'association d'un turbocode duo-binaire à 16 états avec une modulation MDAP-12+4 sur canal gaussien. Décodage Max-Log-MAP, 6 bits de quantification en entrée du décodeur, 8 itérations. Taille des blocs : 110, 216 et 376 octets, rendements R=0,57292, 0,56250 et 0,65278.

Les Figures 6.6, 6.7 et 6.8 montrent les courbes de taux d'erreurs des schémas proposés, ainsi que les limites théoriques correspondantes. A un taux d'erreurs de trames de 10^{-6}, les modulations turbocodées proposées permettent de décoder à 0,7-0,8 dB de la limite théorique pour une modulation MDP-4, à 1,0-1,1 dB pour une modulation MDP-8 et à1,2-1,3 dB pour une modulation MDAP-12+4.

6.5 Estimation des performances des modulations turbocodées pragmatiques (2002 - …)

Un premier critère d'évaluation de l'efficacité de correction d'un schéma de modulation turbocodée est l'importance de l'écart entre la courbe de taux d'erreurs mesurée ou simulée et la limite donnée par la théorie de Shannon. Il apparaît donc important de connaître de manière précise les limites théoriques pour l'ensemble des conditions de transmissions étudiées : canal, modulation, et dans le cas d'une transmission par paquets, taille des paquets transmis, taux d'erreurs visé. Les travaux menés sur ce thème sont décris en section 6.5.1.

D'autre part, l'analyse des résultats de simulation à faibles taux d'erreurs dans les projets décrits précédemment nous a amenés à nous poser le problème de l'estimation des performances asymptotiques des schémas de modulations turbocodées par des moyens autres que la simulation, qui, aux taux d'erreurs visés, requiert des puissances de calcul considérables : à titre d'exemple, pour tracer un point à un taux d'erreurs de trames de 10^{-7} à partir de 30 blocs erronés, il est nécessaire de simuler la transmission et le turbo-décodage de 450 milliard de bits pour des blocs MPEG de 188 octets. Par conséquent, nous nous sommes intéressés à l'application de la méthode de l'impulsion d'erreur pour l'estimation des performances asymptotiques des modulations turbocodées pragmatiques. Les résultats de ces travaux sont décrits en section 6.5.2.

6.5.1 Evaluation des limites théoriques des modulations codées

Un critère important pour juger des performances d'un schéma de codage est d'évaluer l'écart de rapport signal à bruit entre la courbe de taux d'erreurs mesurée (ou simulée) et la limite théorique de Shannon donnée par la capacité du canal. Dans le cas où les séquences transmises ne sont pas de longueur infinie, une borne inférieure sur la probabilité d'erreur pour le codage de blocs de taille donnée est la borne dite de l'empilement de sphères ou *sphere-packing* formulée par Shannon [SHA] en 1959. Cette borne s'éloigne d'autant plus de la capacité que le bloc est court. Ainsi, en pratique, la limite théorique effective à prendre en compte pour la transmission par blocs est la borne *sphere-packing* et non la capacité du canal. Dans [DOL], cette borne a été numériquement évaluée en fonction de la taille des blocs, du taux d'erreurs visé et du rendement de codage, dans le cas d'un canal gaussien à entrée continue. En s'appuyant sur cette contribution, Emeric Maury a développé, sans le cadre de son stage DEA en 2002, un outil permettant de calculer automatiquement la borne *sphere-packing*, pour une transmission sur un canal gaussien en fonction des paramètres suivants : la taille des blocs, le taux d'erreurs de trames visé, le rendement de codage et la modulation considérée (MDP, MAQ, ou MDAP)[1].

Pour une transmission sur canal à évanouissement (Rayleigh, Rice, …), le calcul précis de ces limites s'avère beaucoup plus délicat que dans le cas gaussien, une dimension d'intégration supplémentaire étant nécessaire au calcul de la capacité. Charbel Abdel Nour étudie actuellement les méthodes de calcul numériques les mieux adaptées pour ce type de canaux. Nous disposerons ainsi bientôt d'un outil précieux qui permettra de quantifier avec précision la sous-optimalité des schémas de modulations turbocodées que nous proposons, pour une vaste gamme de canaux de référence.

A la méthode basée sur la théorie de Shannon, nous pouvons ajouter l'expérimentation de méthodes empiriques de détermination des limites théoriques. Emeric Maury en évalue actuellement trois.

La première est basée sur une observation de Claude Berrou dans [BER4], concernant les performances de correction de codes convolutifs systématiques récursifs pour différentes mémoires de code ν et différents rendements de codage R. Il avait remarqué que, à rendement de codage fixé, les différentes courbes de taux d'erreurs se croisent en un point, dont l'abscisse correspond à la limite de Shannon pour le canal considéré. Une interprétation possible de cette observation peut être la suivante : les codes convolutifs systématiques récursifs sont "potentiellement" optimaux, c'est-à-dire qu'ils permettraient d'atteindre la limite de Shannon, à la condition que la valeur de ν tende vers l'infini et que le décodage soit optimal. Nous étudions actuellement le comportement de ces codes sur d'autres canaux, dont nous connaissons la capacité, pour vérifier expérimentalement la véracité de cette hypothèse. Nous disposerions dans ce cas d'un outil expérimental simple pour estimer la limite théorique de transmission sur des canaux dont on ne sait pas calculer la capacité de manière analytique.

Les deuxième et troisième méthodes étudiées utilisent la simulation du turbocodage et décodage d'une séquence de données. Lorsque l'algorithme de décodage élémentaire est optimal (MAP) et que le nombre d'itérations est suffisamment élevé, les performances du turbo-décodage permettent d'approcher la limite de Shannon à quelques dixièmes de dB (0,35 dB avec un turbocode duo-binaire). La perte résiduelle est, au moins partiellement, due

[1] Cet outil est disponible sous la forme d'une *applet* Java à l'adresse suivante :
http://www-elec.enst-bretagne.fr/turbo/

à la sous-optimalité du processus de décodage itératif : une partie de l'information nécessaire au décodage est perdue lors de l'échange des informations extrinsèques.

Une explication possible est que, dans l'algorithme MAP, l'information extrinsèque à l'instant *i* est calculée à partir de données relatives à l'ensemble des nœuds du treillis à cet instant, l'état du codeur à cet instant n'étant pas connu du décodeur. Une idée pour décoder plus près de la limite de Shannon consiste à fournir au décodeur l'état du codeur à chaque instant[1] (il s'agit simplement de l'état **0** si la séquence "tout zéro" est émise). L'expérimentation de cette technique sur un canal gaussien à entrée binaire a conduit à l'observation d'une limite de rapport signal à bruit en deçà de laquelle le décodeur n'arrive plus à corriger les erreurs et qui correspond numériquement à la limite de Shannon de ce canal. Nous appliquons actuellement cette technique à différentes modulations pour vérifier si les résultats obtenus sont similaires. Si cette technique s'avère valide, nous disposerons alors d'un moyen très simple d'estimer expérimentalement les limites théoriques pour des canaux de transmission et des modulations variés. Il est à noter que cette méthode est *a priori* applicable quelle que soit la taille des blocs transmis.

Un second moyen de contrer la sous-optimalité du décodage itératif pour estimer les performances limites consiste à appliquer en entrée du turbo-décodeur des données non bruitées, donc non erronées, pendant les premières itérations de décodage, puis de réintroduire des erreurs pour les itérations suivantes. Là encore, il semble que la valeur du rapport signal à bruit en deçà duquel le décodeur n'arrive plus à corriger les erreurs soit très proche de la valeur limite théorique.

6.5.2 Estimation des performances asymptotiques des modulations turbocodées

Nous avons adapté la méthode de l'impulsion d'erreur, décrite en section 5.3.2, à l'estimation du gain asymptotique d'une association pragmatique d'un turbocode avec une modulation d'ordre supérieur à 2.

La méthode d'impulsion d'erreur ne peut pas être directement transposée aux modulations classiques à plus de quatre états car celles-ci ne sont pas linéaires quand on raisonne au niveau bit, le niveau de protection de chaque bit d'un symbole lié à la modulation dépendant du symbole considéré. Aussi, les bornes qui peuvent être calculées lorsque tous les bits sont protégés de manière identique, comme dans le cas des modulations MDP-2 et MDP-4, ne sont plus valides, car la séquence "tout zéro" ne peut plus servir de séquence de référence. Néanmoins dans le cas particulier d'une association code/modulation de type BICM, où les bits codés, systématiques ou redondants, sont uniformément répartis dans les symboles de modulation, il est possible d'établir une estimation du taux d'erreurs de trames asymptotique.

Nous avons dans un premier temps considéré l'association d'un turbocode duo-binaire avec une modulation MDP-8 sur canal gaussien [BER5]. Le cas des modulations MAQ-M a ensuite été traité [CON1], puis Laura Conde Canencia a étendu la méthode aux canaux à évanouissements de type Rayleigh sans mémoire [CON2-3] (l'article [CON2] est reproduit en Annexe 9). Les résultats obtenus montrent que nous sommes capables d'estimer le comportement asymptotique de BICM turbocodées avec une précision inférieure à 0,5 dB dans tous les des cas de modulation, de rendement de codage et de tailles de blocs considérés, aussi bien sur canal de Gauss que de Rayleigh. Sur les Figures 6.9 et 6.10, la courbe de performance asymptotique simulée n'est distante que de 0,2 dB de la courbe estimée.

[1] Ceci n'est, bien sûr, pas réaliste dans le cas d'une transmission classique.

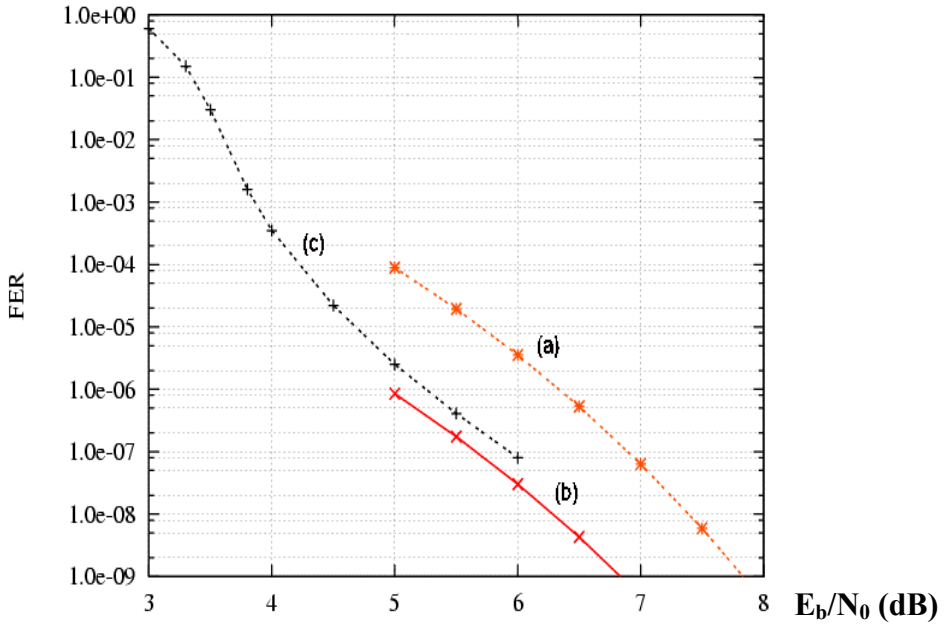


Figure 6.9 : Résultats de simulation et estimations analytiques du taux d'erreurs de trames (FER) asymptotique de l'association pragmatique d' une modulation MAQ-16 et du turbocode DVB-RCS (8 états) pour la transmission de blocs MPEG (188 octets d'information) sur canal gaussien. Rendement de codage : 1/2, efficacité spectrale : 2 bit/s/Hz. Décodage : algorithme Max-Log-MAP, échantillons quantifiés sur 6 bits à l'entrée du décodeur, 8 itérations.
(a) estimation basée sur la distance euclidienne minimale de la constellation
(b) estimation basée sur la méthode proposée
(c) simulation BICM + turbocode

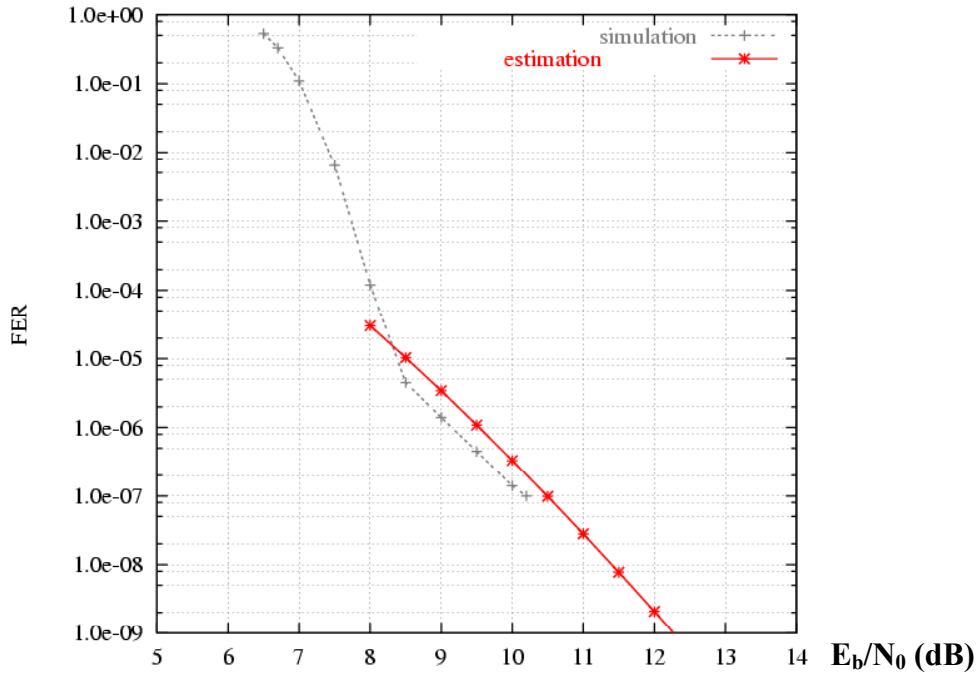


Figure 6.10 : Résultats de simulation et estimation analytique du taux d'erreurs de trames (FER) asymptotique de l'association pragmatique d' une modulation MDP-8 et du turbocode DVB-RCS (8 états) pour la transmission de blocs MPEG (188 octets d'information) sur canal de Rayleigh. Rendement de codage : 2/3, efficacité spectrale : 2 bit/s/Hz. Décodage : algorithme Max-Log-MAP, échantillons quantifiés sur 6 bits à l'entrée du décodeur, 8 itérations.

Malheureusement, les schémas de type BICM ne permettent pas de prendre en compte les stratégies particulières de répartition des bits codés dans les symboles de modulation telles qu'elles sont décrites dans les sous-chapitres précédents. Par conséquent, cette méthode conduit à des performances asymptotiques moyennes, qui peuvent, en pratique, être améliorées par une stratégie étudiée de répartition des bits codés.

6.6 Conclusions, travaux en cours et perspectives

Les différentes études menées sur l'association pragmatique des turbocodes avec des modulations d'ordre élevé, nous ont conduit à la définition de schémas simples et implémentables, présentant des performances situées à environ 1 dB des limites théoriques, sur canal gaussien. Toutefois, l'analyse et l'optimisation des performances de ces schémas pour des transmissions sur des canaux à évanouissement méritent une étude plus poussée. Il s'agit là d'un des objectifs de la thèse de Charbel Abdel Nour, débutée en octobre 2003.

Nous avons également récemment commencé à considérer l'association des turbocodes avec des modulations non conventionnelles. Nous allons étudier l'apport des turbocodes dans un système UWB (Ultra Wide-band), pour des applications de télécommunications domestiques. Cette technique n'utilise pas à proprement parler une modulation avec une porteuse sinusoïdale, mais une transmission par modulation d'impulsions en position. Les impulsions transmises sont de durée très courte, inférieure à la nanoseconde, si bien que le spectre est étalé sur une bande très large, jusqu'à 10 GHz, avec une densité spectrale très faible, située dans le niveau de bruit. Cette technique cause peu d'interférences et n'est pas brouillée par les systèmes de communications en bande étroite. Elle permet, de plus, de mieux traverser les obstacles physiques tels que les murs que les techniques à fréquence porteuse, tout en fournissant des débits importants, au-delà de 400 Mbit/s sur de courtes portées, en deçà d'une dizaine de mètres. Cette étude fait l'objet dans un premier temps d'une contribution à un projet de recherche du Groupe des Ecoles des Télécommunications, démarré en janvier 2004. Elle servira également de point de départ à une nouvelle thèse en mai 2004.

6.7 Références

[BEN1] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Parallel concatenated trellis codes modulation," *International Conference on Communications, ICC'96*, Dallas, Texas, June 1996, pp. 974-978.

[BEN2] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Serial concatenated trellis coded modulation with iterative decoding," *International Symposium on Information Theory*, *ISIT'97*, Ulm, Germany, June. 1997, p. 8.

[BEN3] S. Benedetto and G. Montorsi, "Versatile bandwidth-efficient parallel and serial turbo-trellis-coded modulation," *2nd International Symposium on Turbo codes & Related Topics*, Brest, France, Sept. 2000, pp. 201-208.

[BER1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, n° 10, Oct. 1996, pp.1261-71.

[BER2] C. Berrou and C. Douillard, *Flexible turbo code for low error rates*, Final Report, Contrat ESA-Alenia Spazio, Nov. 2003.

[BER3] C. Berrou, C. Douillard and S. Kerouédan, *TurboΦ for the reverse link*, Datasheet, Contrat ESA-Alenia Spazio, Feb. 2004.

[BER4] C. Berrou, “Some clinical aspects of turbo codes,” *International Symposium on Turbo codes & Related Topics*, Brest, France, Sept. 1997, pp. 26-31.

[BER5] C. Berrou, C. Douillard and M. Jézéquel, “Application of the error impulse method in the design of high-order turbo coded modulations,” *Information Theory Workshop, ITW’2002*, Bangalore, India, Oct. 2002, pp. 41-44.

[CAI] G. Caire, G. Taricco and E. Biglieri, “Bit-Interleaved Coded Modulation,” *IEEE Transactions on Information Theory*, vol. 44, n° 3, May 1998, pp. 927-946.

[CON1] L. Conde Canencia, C. Douillard, M. Jézéquel and C. Berrou, “Application of the error impulse method to 16-QAM bit-interleaved turbocoded modulations,” *Electronic Letters*, vol. 39, n° 6, March 2003, pp. 538-539.

[CON2] L. Conde Canencia and C. Douillard, “Performance estimation of 8-PSK turbocoded modulation over Rayleigh fading channels,” *3rd International Symposium on Turbocodes & Related Topics*, Brest, France, Sept. 2003, pp. 567-570.

[CON3] L. Conde Canencia and C. Douillard, “A new methodology to estimate asymptotic performance of turbocoded modulation over fading channels,” *soumis à 2nd International Symposium on Image/Video Communications over fixed and mobile networks en février 2004.*

[DOL] S. Dolinar, D. Divsalar and F. Pollara, “Code performance as a function of block size,” *TMO progress report 42-133, JPL, NASA*, May 1998.

[JEZ] M. Jézéquel, C. Douillard, P. Adde, C. Berrou et L. Conde, *Etude du mapping optimal des turbocodes sur des constellations à grand nombre d’états*, Rapport final de contrat de recherche externe, France Telecom R&D, mars 2003.

[LEG1] S. Le Goff, *Les turbo-codes et leur application aux transmissions à forte efficacité spectrale*, thèse de doctorat de l’Université de Bretagne Occidentale, Brest, Nov. 1995.

[LEG2] S. Le Goff, A. Glavieux, and C. Berrou, “Turbo-codes and high spectral efficiency modulation,” *International Conference on Communications, ICC'94*, May 1994, pp. 645-649.

[PIC] A. Picart, C. Douillard, P. Adde et M. Jézéquel, *Etude de complexité des turbocodes pour distribution intérieur radio – lot 1: étude des performances – lot2: étude de complexité*, Rapport de contrat de recherche CNET n° 98 1B, août 1999.

[PYN] R. Pyndiah, “Near optimum decoding of product codes: Block turbo codes,” *IEEE Transactions on Communications*, vol. 46, n° 8, Aug. 1998, pp. 1003-1010.

[ROB1] P. Robertson and T. Wörz, “Coded modulation scheme employing turbo codes,” *Electronics Letters*, vol. 31, n° 2, Aug. 1995, pp. 1546-47.

[ROB2] P. Robertson and T. Wörz, “Bandwidth-efficient turbo trellis-coded modulation using punctured component codes,” *IEEE Journal on Selected Areas in Communications*, vol. 16, n° 2, Feb. 1998, pp. 206-218.

[SHA] C. E. Shannon, “Probability of error for optimal codes in Gaussian channel,” *Bell Syst. Tech. Journal*, vol. 38, 1959, pp. 611-656.

[UNG] G. Ungerboeck, “Channel coding with mutilevel/phase signals,” *IEEE Transactions on Information Theory*, vol. IT-28, n°1, Jan. 1982, pp. 55-67.

[VIT] A. J. Viterbi, J. K. Wolf, E. Zehavi and R. Padovani, “A pragmatic approach to trellis-coded modulation,” *IEEE Communications Magazine*, vol. 27, n° 7, July 1989, pp. 11-19.

[ZEH] E. Zehavi, “8-PSK trellis codes for a Rayleigh channel,” *IEEE Transactions on Communications*, vol. 40, n°5, May 1992, pp. 873-884.

7 Conclusion et perspectives

Ma formation puis mon parcours professionnel, décrit dans ce document, m'ont permis d'acquérir un large spectre de compétences, allant de l'électronique numérique et de la conception de circuits intégrés aux communications numériques, en passant par le domaine des architectures de systèmes numériques et la théorie de l'information.

Ce parcours multidisciplinaire m'a ainsi permis de prendre en charge, en octobre 2003, l'animation du nouveau projet de recherche disciplinaire du Groupe des Ecoles des Télécommunications intitulé "Turbo-algorithmes et circuits" (TAC). Ce projet, dont l'objectif principal est de contribuer au maintien de l'ENST Bretagne parmi les leaders mondiaux dans les domaines du codage de canal et des turbo-communications, regroupe six enseignants-chercheurs, un chargé de recherche CNRS et sept doctorants. Dans le domaine du codage de canal, les thèmes abordés sont l'étude des performances comparées des différents types de schémas concaténés, le choix des codes élémentaires, les propriétés de convergence et l'accélération des traitements itératifs, la recherche de permutations, l'association des turbocodes aux modulations à grand nombre d'états, l'étude des performances théoriques des turbocodes et les architectures des circuits de décodage. Dans le domaine des turbo-communications, le projet TAC vise à mettre en œuvre des techniques de traitement itératifs dans les systèmes de transmissions multi-porteuses (OFDM), multi-capteurs (MIMO) et multi-utilisateurs (CDMA, IDMA).

Les thèmes d'actualité dans lesquels je vais personnellement m'impliquer, par le biais de suivi de thèses notamment, sont les suivants :

- l'étude et l'optimisation des systèmes turbocodés pour des transmissions sur canaux non gaussiens ;
- la définition de turbocodes pour des systèmes de transmissions faisant appel à des techniques d'accès multiple ou de modulations innovantes (technique d'accès multiple à répartition par entrelacement, transmission UWB à modulation par impulsion, …).

Le maintien d'un haut niveau de recherche passe aussi par une veille continue sur les thèmes en émergence dans le domaine des transmissions numériques, notamment à travers les principaux colloques organisés par l'*IEEE Communication Society* (*Globecom*, *ICC*), l'*IEEE Information Theory Society* (*ISIT*, *ITW*), ainsi bien sûr que l'*International Symposium on Turbo Codes & Related Topics*. Mon activité régulière de relecture d'articles y contribue également.

De plus, notre démarche ayant toujours intégré une forte interaction entre les principes théoriques, les algorithmes associés et leur réalisation matérielle, nous sommes en mesure d'élaborer, en coopération avec des partenaires industriels, des solutions concrètes pour les applications de télécommunications futures. A titre d'exemple, pour ne citer que les collaborations les plus récentes, nous avons proposé l'an dernier un schéma adaptatif de codage et de modulation pour la voie retour des futurs systèmes de communication par satellite dans le cadre du projet MHOMS financé par l'ESA (partenaires industriels Space Engineering et Alenia Spazio). Nous avons également démarré en janvier une étude sur l'apport des turbocodes dans les systèmes UWB (partenaires industriels France Telecom, TurboConcept).

Dans la même optique, je participe aussi à l'action commune entre l'ENST Bretagne et France Telecom R&D visant à défendre les turbocodes dans différents contextes de

normalisation. Les groupes de normalisation concernés à court et moyen termes sont DVB-RCS2, IEEE 802.11n (*Wireless Local Area Network* à haut débit), IEEE 802.20 (*Mobile Broadband Wireless Access*), IEEE 802.15 (*Wireless Personal Area Network*) et VDSL2.

Un point important restant encore à développer au département Electronique et plus particulièrement au niveau du groupe TAC est le renforcement de nos interactions avec les laboratoires étrangers travaillant dans le même domaine : mon implication avec neuf autres enseignants-chercheurs de l'ENST Bretagne dans le réseau d'excellence Newcom devrait permettre rapidement de concrétiser une telle initiative. Déjà, le prochain *International Symposium on Turbo Codes & Related Topics*, qui aura lieu en Allemagne en avril 2006, fait l'objet d'une organisation commune entre l'Université Technique de Munich et l'ENST Bretagne. Des encadrements conjoints de thèse constitueront l'étape suivante.

Enfin, un de mes objectifs à court terme est la valorisation de l'expertise que j'ai acquise dans le domaine des turbo-communications depuis presque dix ans par la rédaction, pour septembre 2004, d'un ouvrage collectif intitulé "Codes et turbocodes", à paraître aux éditions Springer France en 2005 et dont Claude Berrou est l'éditeur. J'ai la charge des chapitres traitant des codes *m*-binaires et des modulations turbocodées.

Annexe 1

Article “Générateur de machines séquentielles autotestables pour circuits intégrés spécifiques”, *Annales des Télécommunications*, 1990.

Cet article décrit la méthode mise au point pendant ma thèse pour le test automatique en ligne et hors ligne de machines séquentielles.

Générateur de machines séquentielles autotestables pour circuits intégrés spécifiques

Claude BERROU *
Catherine DOUILLARD *

Résumé

Cet article présente une méthode originale de synthèse systématique, à base de mémoire morte, de machines séquentielles synchrones (automate, séquenceur,...) autotestables en ligne et hors ligne, pour circuits intégrés spécifiques compilés. Les choix en matière de codage de graphe et d'architecture de circuit y sont justifiés relativement aux objectifs de simplicité, de faible encombrement, de rapidité de fonctionnement et plus particulièrement de testabilité. Le test, en ligne et hors ligne, fait appel à un codage détecteur d'erreurs de nature récurrente; outre les données utiles, *propres au codage de la machine, la mémoire morte contient des informations redondantes distribuées entre deux états consécutifs, ce qui garantit, non seulement la cohérence de l'état présent, mais aussi la conformité de la transition passée. Ce type de codage, dit à* redondance distribuée, *est à la base du test en ligne; son taux de confiance est donné pour différentes classes de défaillances matérielles. Le test hors ligne, exhaustif, consiste simplement, en exploitant le test à redondance distribuée à parcourir dans l'ordre des adresses de la mémoire toutes les transitions possibles du graphe.*

Mots clés : Machine séquentielle, Testabilité, Circuit ASIC, Code détecteur erreur, Mémoire morte.

A SELF-TESTING SEQUENTIAL MACHINES GENERATOR FOR ASIC's

Abstract

This paper presents an original method using a ROM *memory, of systematically synthesizing on-line and off-line self-testing synchronous sequential machines (automata, sequencers,...), for the design of compiled* ASIC's. *Choices about state transition graph coding and circuit architecture are related to simplicity, compactness, operating rate and especially to testability. The on-line and off-line testing is based on an error detecting code of a recurrent type : in addition to* useful *data, the* ROM *memory contains redundant information which is distributed between two consecutive states. This method guarantees both present state coherence and past transition conformity. On-line testing is based in this type of coding, called* distributed redundancy coding. *Its fault coverage is given for different classes of hardware failures. Off-line testing, which is exhaustive, uses the distributed redundancy coding technique and consists in scanning all the possible graph transitions in memory address order.*

Key words : Sequential machine, Testability ASIC, Error detecting code, Read only storage.

Sommaire

I. *Introduction.*
II. *Le mécanisme de test.*
III. *Performances.*
IV. *Conclusion.*
Bibliographie (5 réf.).

I. INTRODUCTION

I.1. Le test des circuits intégrés.

Le *test hors ligne*, principalement destiné à éliminer les composants défectueux en fin de fabrication, permet

* Ecole nationale supérieure des télécommunications de Bretagne. Laboratoire circuits intégrés, BP 832, 29285 Brest Cedex.

la détection de pannes dans un circuit en dehors de son mode opératoire normal. Il consiste à appliquer sur ses entrées des séquences logiques (vecteurs de test) appropriées et à vérifier la conformité de ses sorties. Tant que la complexité du composant reste faible, il est aisé et rapide de réaliser un test complet capable de valider tous les fonctionnements possibles du circuit. Pour des circuits plus complexes et plus particulièrement pour des circuits à haut degré de séquentialité, un test exhaustif, par de seuls moyens extérieurs, est bien souvent irréalisable. Il est alors nécessaire d'intégrer dans le circuit partie ou totalité des mécanismes utiles à son propre test.

Le *test en ligne* est mis en œuvre lorsqu'une détection permanente et immédiate des pannes est requise pendant le fonctionnement normal du circuit. Le test en ligne fait généralement appel à un codage des sorties et éventuellement des entrées de chaque bloc fonctionnel du circuit et à un contrôle permanent de la validité de chaque mot de code en sortie de ces blocs, et fournit un signal d'indication d'erreur lorsqu'une incohérence est détectée.

D'une manière générale, le test en ligne de fonctions complexes, notamment séquentielles, qui n'ont pas été construites pour en aménager *a priori* la testabilité, nécessite des techniques complexes et coûteuses en circuiterie. La méthode exposée dans cet article est simple à mettre en œuvre car elle s'appuie sur une architecture de machine spécialement adoptée pour y intégrer la testabilité.

I.2. Les machines séquentielles à base de mémoire morte.

Parmi les multiples façons d'aborder le problème de la synthèse de circuits séquentiels, la microprogrammation en mémoire morte est une solution particulièrement attractive lorsque sont recherchées :

— la simplicité de la synthèse systématique,

— la simplicité de la construction et du placement-routage,

— une testabilité aisée, par la possibilité d'adjoindre aux données *utiles* de programmation une information redondante pour la mise en œuvre de codages détecteurs (éventuellement correcteurs) d'erreurs.

Cette solution peut être en contre-partie pénalisante d'un point de vue encombrement et rapidité de fonctionnement. Pour éliminer ces inconvénients, nous adoptons un codage de graphe à réceptivité restreinte et à qualifieur de transition unique, c'est-à-dire que chaque état possède au plus deux arcs sortants différenciés par une condition (*vrai* ou *faux*) sur les valeurs de certaines entrées [1]. L'adressage de la mémoire morte n'est alors construit qu'à partir des seules variables internes de la machine séquentielle. La méthode décrite en [1] est ici étendue à des qualifieurs de transition de type *produit (i.e. et logique)*. Chaque mot de la mémoire correspond donc à un état et contient les six champs suivants :

— le *masque des entrées* concernées par la transition future,

— la *valeur de ces entrées* pour la condition *vrai*,

— l'*état futur* et la *valeur des sorties* pour la condition *vrai*,

— l'*état futur* et la *valeur des sorties* pour la condition *faux*.

Le formalisme utilisé, proche de l'organigramme, s'adapte à tous les cas de machines séquentielles synchrones (automate, séquenceur, contrôleur,...), éventuellement au prix de l'introduction d'états intermédiaires. A titre d'exemple, considérons le graphe de la figure 1a qui décrit le fonctionnement d'un automate à 6 états et 3 entrées (les sorties y ont été omises dans un souci de clarté). Chaque transition y est caractérisée par une condition sur les entrées I_1, I_2 et I_3 ou partie d'entre elles. Ce graphe, bien qu'à réceptivité restreinte, n'est pas à qualifieur de transition unique, de type *produit*, car les états S_1 et S_5 possèdent chacun 3 arcs sortants et, par ailleurs, la condition $I_1 \oplus I_2$ n'est pas de type *produit*.

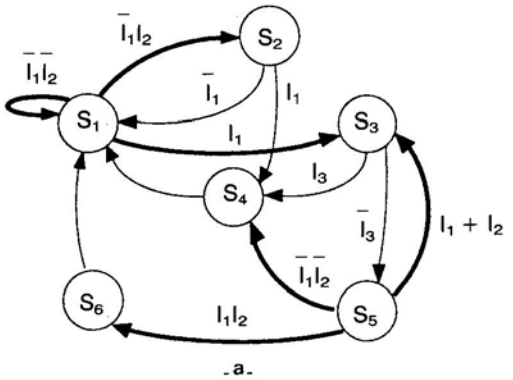


FIG. 1a. — Graphe initial.
Initial state transition graph.

Ce problème est aisément résolu, comme le montre la figure 1b, par l'introduction de deux états intermédiaires S_7 et S_8. Il est, dans tous les cas, possible de rendre ces états intermédiaires *transparents* par l'utilisation d'une horloge de fréquence multiple (le plus souvent double, en pratique) de la fréquence de séquencement de la machine, et d'un système rigoureux de commutation d'horloges. Cela nécessite toutefois d'ajouter dans chaque mot de mémoire deux bits correspondant aux

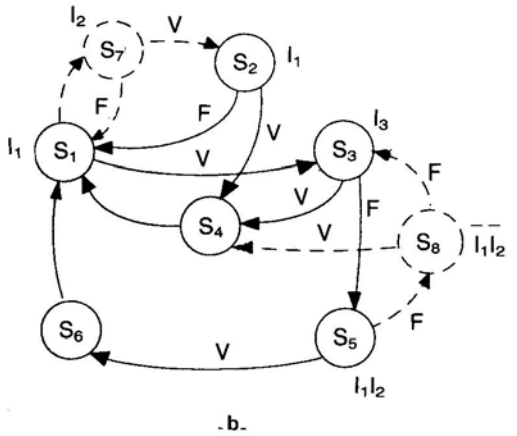


FIG. 1b. — Graphe modifié.
Modified state transition graph.

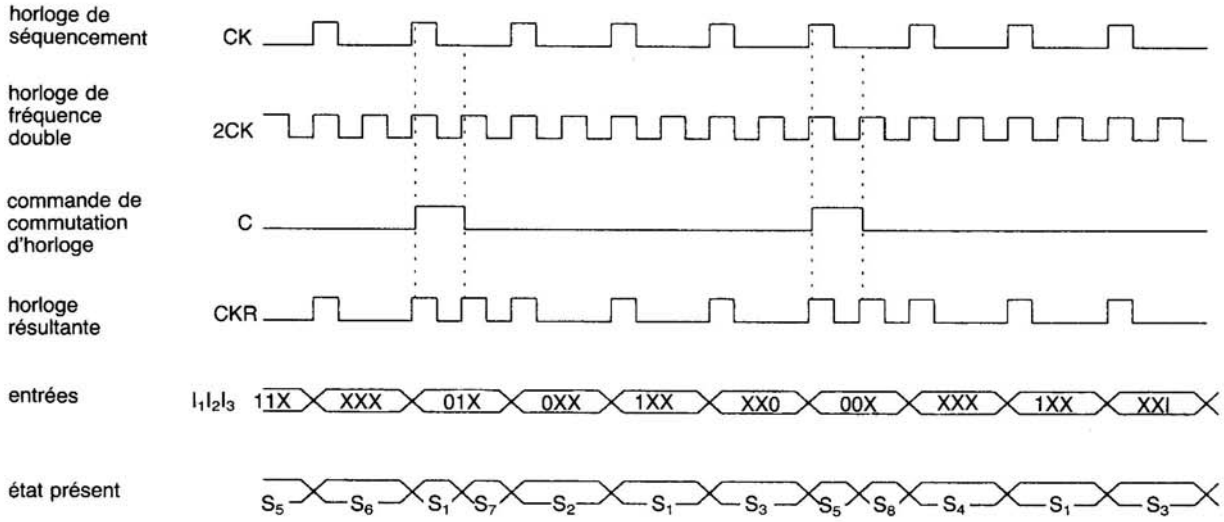


FIG. 2. — Une séquence possible pour la machine de la figure 1b

A possible sequence for the state machine in Figure 1b.

deux transitions possibles et déclenchant la commutation d'horloges si la transition à franchir conduit à un des états intermédiaires. Le chronogramme de la figure 2 décrit une séquence de la machine de la figure 1b pour laquelle un mécanisme de commutation entre l'horloge de séquencement et l'horloge de fréquence double est mis en œuvre. On remarquera l'allure particulière, dite intermittente, de l'horloge résultante CKR effectivement appliquée à la machine.

II. LE MÉCANISME DE TEST

II.1. Principe du codage.

La technique de test en ligne consiste à vérifier, à chaque étape de séquencement de la machine, la cohérence des informations relatives à l'état courant et la conformité de la transition franchie. Sa mise en œuvre fait appel à un codage de nature récurrente : les informations relatives à chaque état, exceptés les champs contenant les deux états futurs possibles, sont codées à l'aide d'un code en bloc séparable. Les bits de redondance que l'on obtient sont séparés en deux champs et distribués de telle sorte que chaque mot de mémoire contienne un champ dit *de redondance locale* concernant l'état présent et deux champs dits de *redondance récurrente* concernant les deux états futurs possibles. Ce codage est dit à *redondance distribuée*.

Le choix du code initial, un code produit de parités, s'est fait sur les avantages de la simplicité de son circuit de décodage et de sa large capacité de détection [2]. Le codage du mot contenant les quatre champs à protéger et les deux bits de commutation d'horloge consiste à en disposer les bits sous la forme d'une matrice à l lignes et c colonnes (après éventuel bourrage) et à calculer les parités de chaque ligne et de chaque colonne, ainsi que la parité globale du mot. Les mots de code ainsi obtenus comportent donc $L = (l+1)(c+1)$ bits, soit lc bits d'information *utiles* et $r = l + c + 1$ bits de redondance.

Un encombrement minimal de la redondance est obtenu pour des valeurs de l et c les plus voisines possibles.

Une autre méthode plus directe, de vérification de la conformité de la transition passée consisterait à inclure dans chaque mot de mémoire, des informations redondantes sur le codage des états précédents. Cependant, le nombre maximal d'antécédents d'un état quelconque est généralement important et cette solution est matériellement coûteuse. La technique de codage à redondance distribuée n'a pas à prendre ce paramètre en compte et garantit simultanément la cohérence de l'état présent et la conformité de la transition passée sans codage explicite des états. Ainsi, la part de la mémoire occupée par la redondance reste modérée : de 10 à 35% suivant le type de graphe et la couverture de fautes recherchée.

II.2. Test en ligne.

L'architecture de la machine, incluant le test en ligne, est présentée en figure 3. Outre la mémoire et le registre d'état, cette machine à m entrées, N états ($N \leq 2^n$) et p sorties, contient :

— le bloc de décision sur la transition, dont la structure duale, donnée en figure 4, fournit les deux valeurs complémentaires *vrai* et *faux* qui commandent la sélection des informations en sortie de la mémoire. Cette structure duale prémunit au moins contre les collages simples et si l'une des deux sorties est erronée, les sorties des multiplexeurs le sont aussi ;

— le registre de contrôle qui mémorise d'un coup d'horloge sur l'autre le champ de redondance récurrente, de longueur k, sélectionné de manière à ce qu'à chaque étape, les mots de codes initiaux soient reconstitués ;

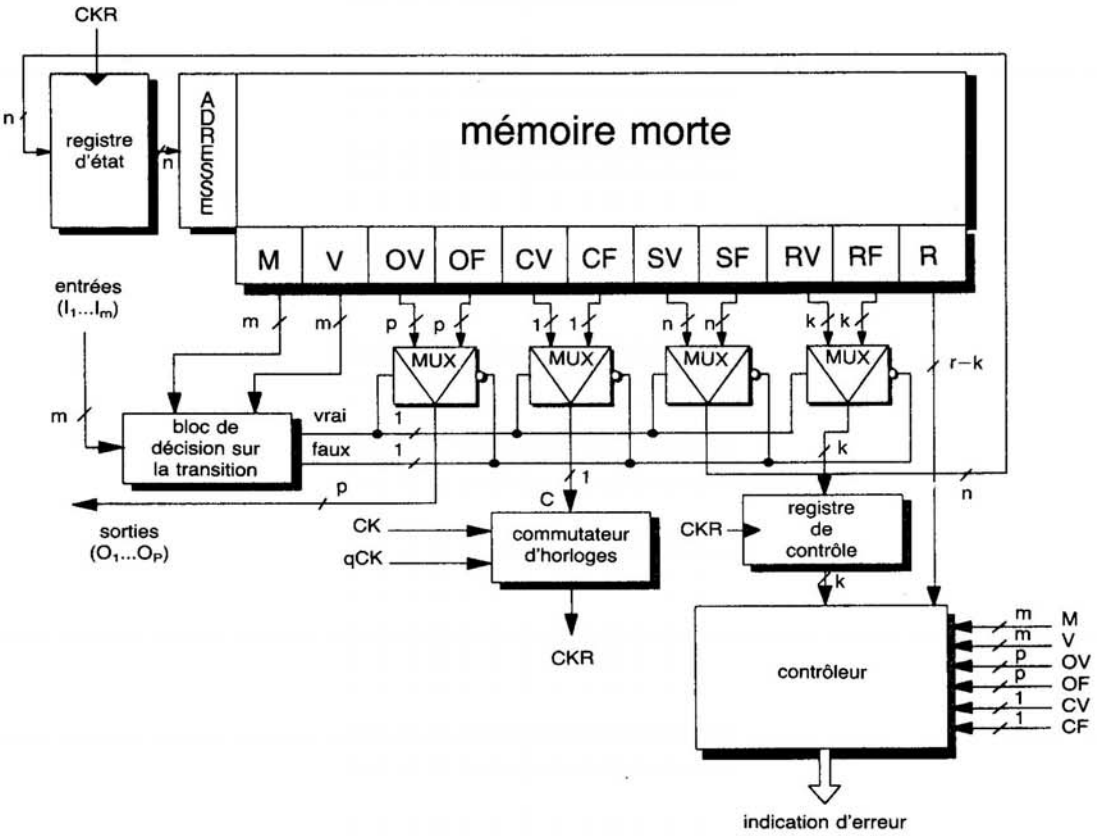


FIG. 3. — Architecture de la machine.

State machine architecture.

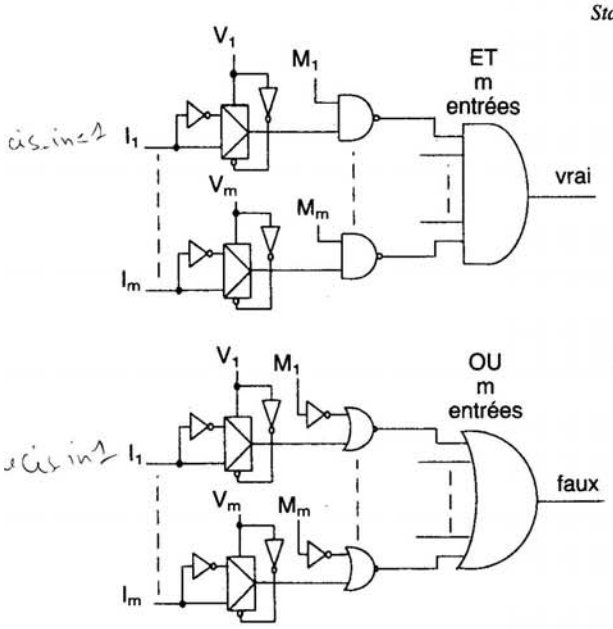


FIG. 4. — Bloc de décision sur la transition.

Transition decision block.

— le contrôleur dont la structure, en *double rail*, prémunit au moins contre les collages simples et dont le taux de détection des collages doubles ou triples est supérieur à 90% [3, 4]. Ce contrôleur autotestable fournit, en cas d'erreurs, deux sorties non complémentaires. Le schéma d'un tel contrôleur pour le décodage de mots de code composés de 8 bits d'information et 7 bits de contrôle est donné en figure 5;

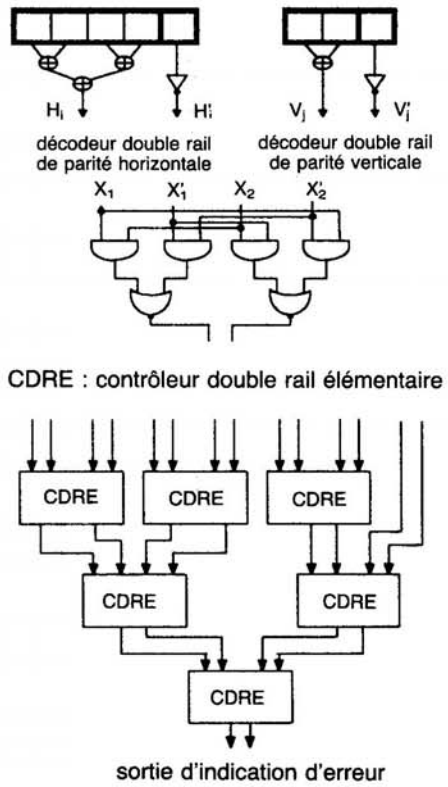


FIG. 5. — Contrôleur *double rail* pour le décodage de mots de code composés de 8 bits d'information et 7 bits de contrôle.

Decoding codewords composed of 8 data bits and 7 check bits with a two-rail code checker.

— le commutateur des horloges CK et qCK ($q \geq 2$), commandé par C (CV ou CF), pour le traitement des états intermédiaires.

Cette première version du circuit ne garantit pas l'intégrité des sorties ni celle du bit C de commande de commutation d'horloges, puisque le test s'effectue sur le contenu de mémoire, avant le multiplexage. Le remplacement d'un multiplexeur par un quadruple multiplexeur, comme le montre la figure 6 pour le bit C, règle ce problème car une faute sur les sorties ou sur C est maintenant répercutée en entrée du contrôleur.

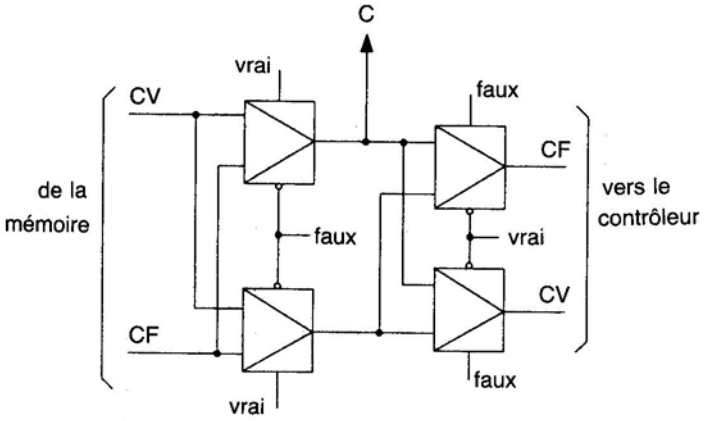


FIG. 6. — Quadruple multiplexeur.

Quadruple multiplexer.

II.3. Test hors ligne.

Le test hors ligne exhaustif utilise le codage à redondance distribuée et consiste à parcourir, dans l'ordre des adresses de la mémoire, l'ensemble des états et à appliquer, pour chacun d'eux, des entrées satisfaisant alternativement aux conditions *vrai* et *faux*. Ces entrées proviennent directement des champs M et V. La vérification de chaque transition nécessite deux périodes de l'horloge de séquencement CK. Le test exhaustif est donc réalisé en un temps 4 N/CK.

III. PERFORMANCES

III.1. Classes de défaillance et testabilité.

Le taux de confiance du codage à produit de parités distribué est évalué par l'intermédiaire d'une probabilité de non-détection d'erreurs (et non d'une probabilité d'avoir des erreurs non détectées, dont la valeur, plus faible [5], n'est pas directement liée au taux de confiance) pour un ensemble de défaillances dont les effets conduisent, soit à des erreurs d'adressage, soit à des informations erronées sur l'état présent.

Erreur d'adressage. Celle-ci peut être provoquée par une erreur de programmation de la mémoire ou un défaut physique affectant les champs *états suivants* d'un mot de mémoire ou bien encore par un défaut situé sur le chemin d'adressage (registre d'état, décodeur d'adresse...). Une telle erreur est détectée si le champ de redondance récurrente contenu dans le registre de contrôle et supposé correct, n'est pas compatible du reste du mot de code. Si l'on admet que toutes les valeurs possibles pour les champs de récurrence sont équiprobables et que ces champs ne contiennent pas de bits corrélés, la probabilité de non-détection d'une telle défaillance est :

$$P_{\mathrm{ndA}} = 1/2^k. \tag{1}$$

Si N_r est le nombre maximal d'états, parmi N, possédant un champ de récurrence identique, la probabilité de non-détection, dans le pire cas, est :

$$P_{\mathrm{ndA,max}} = \frac{N_r}{N}. \tag{2}$$

Pour minimiser $P_{\mathrm{ndA,max}}$, le codage doit être réalisé de telle façon que les champs de redondance récurrente prennent le plus grand nombre possible de valeurs différentes, cela étant bien sûr d'autant plus facile à obtenir que k est grand. Dans le cas où N_r n'est pas égal à un, les risques de non-détection peuvent être diminués par une loi appropriée de codage des états. On pourra par exemple, si les erreurs d'adressage sont de nature *a priori* quelconque, adopter un codage qui maximise la distance de Hamming entre tous les états possédant la même redondance récurrente.

Informations erronées sur l'état présent. Ce type d'erreur peut être dû à une erreur de programmation ou à un défaut physique de la mémoire ou bien encore à un défaut situé entre la sortie de la mémoire et l'entrée du contrôleur. Les cas d'une erreur de programmation et d'une défaillance matérielle (soit dans la mémoire, soit dans le circuit extérieur) sont à traiter séparément. En effet, une erreur de programmation modifie le contenu d'un mot de mémoire d'une manière quelconque, alors qu'une panne au niveau d'une porte logique ou d'une connexion n'altère généralement qu'un nombre restreint de bits.

Cas 1 : erreur de programmation. En supposant que tous les cas d'erreurs sont équiprobables, la probabilité de non-détection d'une telle défaillance est :

$$P_{\mathrm{ndB}} = 1/2^r. \tag{3}$$

Cas 2 : erreur due à une défaillance matérielle. L'étude de ce cas nécessite quelques rappels sur les propriétés de détection du code produit de parités :

a) Le code assure la détection de toute erreur due à un nombre de bits erronés non multiple de 4.

b) Dans le cas où le nombre de bits erronés est égal à 4 l'erreur n'est pas détectée si les bits erronés forment un rectangle dans la disposition matricielle utilisée pour le codage.

c) Dans le cas où le nombre de bits erronés est multiple de 4, l'erreur n'est pas détectée si les bits erronés forment, 4 à 4, des rectangles dans la disposition matricielle.

Si l'on note $P_{\mathrm{nd}}(\varepsilon = i)$, la probabilité de non-détection de i bits erronés, on a donc :

$$P_{\mathrm{nd}}(\varepsilon = i) = 0 \quad \text{pour } i \neq 4j$$

$$P_{\mathrm{nd}}(\varepsilon = 4) = \frac{C_{l+1}^2 C_{c+1}^2}{C_L^4}$$

$$P_{\mathrm{nd}}(\varepsilon = 4j) < (P_{\mathrm{nd}}(\varepsilon = 4))^j \quad \text{pour } j > 1.$$

Cette dernière inégalité vient de ce que le nombre de dispositions rectangulaires possibles pour 4 bits dans une matrice, diminue à chaque placement supplémentaire. Pour $l + c \geq 4$, $P_{\mathrm{nd}}(\varepsilon = 4)$ est majorée par $6/L^2$, ce qui permet d'écrire :

$$P_{\mathrm{nd}}(\varepsilon = 4j) < (6/L^2)^j \quad \text{pour} j \geq 1.$$

Si on note également $P_{\mathrm{c}}(\varepsilon = i)$ la probabilité d'avoir i erreurs, sachant qu'il y en a au moins une (probabilité conditionnelle), la probabilité globale de non-détection d'erreur est donnée par :

$$(4) \qquad P_{\mathrm{ndC}} = \sum_{j=1}^{E(L/4)} P_{\mathrm{nd}}(\varepsilon = 4j) P_{\mathrm{C}}(\varepsilon = 4j),$$

où $E(L/4)$ représente la partie entière de $L/4$.

Puisque $P_{\mathrm{nd}}(\varepsilon = 4j)$ est une fonction décroissante de j, P_{ndC} est majorée par :

$$P_{\mathrm{nd}}(\varepsilon = 4) \sum_{j=1}^{E(L/4)} P_{\mathrm{C}}(\varepsilon = 4j),$$

soit encore par $P_{\mathrm{nd}}(\varepsilon = 4)$ car la somme des probabilités conditionnelles est inférieure ou égale à 1. On a donc :

$$(5) \qquad P_{\mathrm{ndC,max}} = 6/L^2.$$

Cette borne supérieure, quoique numériquement satisfaisante, est, en pratique, une large surestimation de la probabilité de non-détection. Si l'on considère, par exemple, une distribution uniforme des probabilités conditionnelles, c'est-à-dire si :

$$P_{\mathrm{C}}(\varepsilon = i) = 1/L,$$

et en ne retenant que le terme prépondérant dans (4), la probabilité P_{ndC} est réduite à la valeur typique :

$$(6) \qquad P_{\mathrm{ndC,typ}} \approx 6/L^3.$$

III.2. Encombrement et rapidité de fonctionnement.

Le tableau I détaille l'encombrement de chaque bloc de la machine, en fonction de :

— m, $N(N \leq 2^n)$ et p, nombre d'entrées, d'états et de sorties,

— r, nombre total de bits de contrôle de chaque mot de code, et k, longueur du champ de redondance récurrente,

— q, rapport des fréquences d'horloges pour le traitement des états intermédiaires (éventuellement).

La taille de la mémoire morte est :

$$T = N[2(m + n + p + 1) + r + k] \text{ (bits)},$$

TABL. I. — Encombrement du circuit, hors mémoire.

	Encombrement approximatif (portes)
Bloc de décision sur la transition	$7,5\ m$
Registre d'état	$6\ n$
Registre de contrôle	$6\ k$
Contrôleur *double rail*	$8(m + p) + 3,5r + 3$
Commutation d'horloges	$9q + 4$
Multiplexeurs en sortie de ROM (quadruples en partie)	$4(p + 1) + n + k$
Circuiterie spécifique au test hors ligne	$3,5m + 10n + 20$

la redondance occupant :

$$T_r = N(r + k) \text{ (bits)}.$$

On observera que T_r/T est une fonction décroissante de la complexité de la machine puisque T_r ne dépend (linéairement) que des dimensions de la matrice de codage.

A titre d'exemple, une machine à 6 entrées, 50 états et 6 sorties, avec $r = 12$, $k = 6$ et $q = 2$, utilise une mémoire de 2,8 kbits, dont 0,9 kbit de redondance, et 420 portes de circuiterie complémentaire.

Les probabilités de non-détection d'erreurs sont $P_{\mathrm{ndA}} = 1/64$, $P_{\mathrm{ndB}} = 1/4096$, et avec $1 = 7$ et $c = 4$: $P_{\mathrm{ndC,max}} = 3,8\ 10^{-3}$.

Le chemin critique du circuit contient le registre d'état, la mémoire, le bloc de décision sur la transition, un multiplexeur (quadruple) et le contrôleur. Pour le même exemple et sur la base d'une technologie CMOS 1μ, le temps de propagation dans le chemin critique peut être évalué à 25 ns (dont 10 ns pour le temps d'accès de la mémoire, qui ne contient pas de décodeur *colonnes*, et 6 ns pour le contrôleur). La fréquence maximale de séquencement est 20 MHz (car $q = 2$).

IV. CONCLUSION

La méthode de synthèse de machines séquentielles hautement testables que nous avons présentée dans cet article, conduit à des circuits peu encombrants et rapides et est particulièrement adaptée à la génération automatique de fonctions séquentielles complexes pour circuits intégrés spécifiques.

Une deuxième étape de travail porte actuellement sur les aspects informatiques de cette génération automatique qui doit, successivement :

— vérifier la validité du graphe et le transformer, si nécessaire, en un graphe à qualifieurs de transition uniques de type *produit*,

— générer, à partir du graphe et du degré de testabilité souhaité, le contenu de la mémoire morte et déterminer les paramètres de chacun des blocs du circuit,

— établir le fichier de description logique (liste des composants ou *net-list*) du circuit, le graphe et la *net-list* étant décrits à l'aide d'un langage unique, VHDL (VHSIC hardware description language).

Manuscrit reçu le 4 juillet 1990, accepté le 10 octobre 1990.

BIBLIOGRAPHIE

[1] GREEN (D.). Modern logic design. Electronic systems engineering series. *Addison-Wesley* (1986), chap. 4, pp. 92-130.

[2] MICHELSON (A. M.), LEVESQUE (A. H.). Error-control techniques for digital communications. *Wiley-Interscience Publ.* (1985), pp. 55-58.

[3] WAKERLY (J. F.). Error detecting codes, self-checking circuits and applications. *Elsevier North-Holland, Inc.*, New York (1978).

[4] NANYA (T.), MOURAD (S.), MC CLUSKEY (E. J.). Multiple stuck-at fault testability of self-testing checkers. *Dig. 18th Ann. int. Symp. fault-tolerant comput.* (FTCS18), Tokyo, Japan (June 1988), pp. 381-386.

[5] LEUNG (C.). Evaluation of the undetected error probability of single parity-check product codes. *IEEE Trans. Computer* (Feb. 1983), **31,** n° 2, pp. 250-253.

BIOGRAPHIE

Claude BERROU, né à Penmarc'h, France en 1951. Ingénieur diplômé de l'INPG (1975). Actuellement Maître de Conférence et responsable du groupe de recherche en circuits intégrés pour télécommunications à l'ENST de Bretagne.

Catherine DOUILLARD, née à Fontenay-le-Comte, France en 1965. Ingénieur diplômé de l'ENST de Bretagne (1988). Elle y prépare actuellement une thèse de Doctorat sur la synthèse de circuits autotestables.

Annexe 2

Algorithmes de décodage MAP et Max-Log-MAP

Les algorithmes de décodage SISO les plus couramment mis en œuvre pour le décodage itératif de codes convolutifs concaténés sont dérivés de l'algorithme MAP (*Maximum A Posteriori*), encore appelé APP (*A Posteriori Probability*), *backward-forward*, ou bien BCJR [BAH].

L'algorithme MAP permet de calculer le logarithme du rapport de vraisemblance ou LLR (*Log-Likelihood Ratio*) associé à la donnée émise par la source à l'instant i, d_i :

$$\Lambda(d_i) = \ln \frac{\Pr\{d_i = 1 \,|\, \mathbf{R}_1^k\}}{\Pr\{d_i = 0 \,|\, \mathbf{R}_1^k\}}$$

$\mathbf{R}_1^k$ désigne la séquence bruitée reçue par le récepteur : $\mathbf{R}_1^k = \{R_1, \cdots, R_k\}$ où R_i est constitué du symbole bruité systématique et du symbole de redondance reçus par le décodeur à l'instant i : $R_i = (x_i, y_i)$ [1].

Le signe du LLR fournit la valeur binaire de la décision sur d_i ($\hat{d}_i = 0$ si $\Lambda(d_i) \leq 0$ et $\hat{d}_i = 1$ sinon), tandis que la valeur absolue de $\Lambda(d_i)$ représente une mesure de la fiabilité de cette décision.

Le calcul de chaque probabilité *a posteriori* ou APP (*A Posteriori Probability*) $\Pr\{d_i = j \,|\, \mathbf{R}_1^k\}$, j = 0,1, fait appel à la structure du treillis du code et passe par la séparation des informations antérieures et postérieures à l'instant i. On peut montrer que :

$$\Lambda(d_i) = \ln \frac{\sum_{\mathbf{s}'} \sum_{\mathbf{s}} \gamma_1(R_i, s', s)\alpha_{i-1}(s')\beta_i(s)}{\sum_{\mathbf{s}'} \sum_{\mathbf{s}} \gamma_0(R_i, s', s)\alpha_{i-1}(s')\beta_i(s)}$$

- $\gamma_j(R_i, s', s)$, $j = 0,1$, représente la probabilité de transition relative à la transition en états s' $\rightarrow$ s du treillis pour une donnée transmise égale à j et sachant que l'échantillon reçu est R_i. Si aucune branche du treillis ne relie l'état s' à l'état s ou bien si la donnée portée par cette transition ne vaut pas j, $\gamma_j(R_i, s', s) = 0$. Sinon, cette grandeur se calcule comme le produit de la probabilité *a priori* que la source émette la valeur j à l'instant i et de la probabilité de transition sur le canal de transmission. Pour une transmission sur canal gaussien, $\gamma_j(R_i, s', s)$ s'exprime comme :

  $$\gamma_j(R_i, s', s) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - X_i)^2}{2\sigma^2}\right) \exp\left(-\frac{(y_i - Y_i)^2}{2\sigma^2}\right) \Pr(d_i = j)$$

  σ^2 est la variance du bruit et X_i et Y_i représentent ici les symboles binaires modulés (± 1) émis correspondant à la branche s' $\rightarrow$ s du treillis.

[1] Pour simplifier, on considère un code convolutif de rendement 1/2.

– α et β sont appelées probabilités aller (ou *forward*) et retour (ou *backward*) et sont calculées récursivement en parcourant le treillis :

$$\alpha_i(s) = \sum_{\mathbf{s}'} \left(\gamma_0(R_i, s', s)\alpha_i(s') + \gamma_1(R_i, s', s)\alpha_i(s')\right)$$

$$\beta_i(s') = \sum_{\mathbf{s}} \left(\gamma_0(R_{i+1}, s', s)\beta_{i+1}(s) + \gamma_1(R_{i+1}, s', s)\beta_{i+1}(s)\right)$$

Les probabilités α sont calculées récursivement en parcourant le treillis dans le sens direct ($i = 1 \ldots k$) et les probabilités β sont calculées récursivement en parcourant le treillis dans le sens inverse ($i = k \ldots 1$). Leurs valeurs initiales dépendent de la connaissance ou non des états initial et final du codeur.

Les expressions précédentes montrent que le décodage MAP fait appel à un nombre important d'opérations : additions, multiplications et calculs d'exponentielles. La réécriture de ces expressions dans le domaine logarithmique permet des simplifications : les multiplications deviennent des additions et les exponentielles disparaissent.

Les additions peuvent être transformées en appliquant la réécriture suivante :

$$\ln(\exp(x) + \exp(y)) = \max(x, y) + \ln(1 + \exp(-|y - x|))$$

En pratique, notamment pour une mise en œuvre matérielle de cet algorithme, on adopte souvent la simplification suivante, dite Max-Log :

$$\ln(\exp(x) + \exp(y)) \approx \max(x, y)$$

L'algorithme de décodage simplifié utilisant cette simplification est appelé algorithme Max-Log-MAP.

Dans le domaine logarithmique, les probabilités p deviennent des métriques M, par le biais de la transformation suivante :

$$M = -\sigma^2 \ln p$$

Le calcul du LLR associé à la donnée d_i par l'algorithme Max-Log-MAP conduit alors à :

$$\Lambda(d_i) = \frac{1}{2}\left[\min_{s,s'}\{x_i - Y_i y_i + M_{i-1}^F(s') + M_i^B(s)\}_{j=0} - \min_{s,s'}\{-x_i - Y_i y_i + M_{i-1}^F(s') + M_i^B(s)\}_{j=1}\right]$$

où M^F et M^B sont les métriques aller et retour associées aux nœuds du treillis, calculées récursivement par :

$$M_i^F(s) = \min_{s', j=0,1}\left\{M_{i-1}^F(s') - X_i x_i - Y_i y_i\right\}$$

$$M_i^B(s') = \min_{s, j=0,1}\left\{M_{i+1}^F(s) - X_{i+1} x_{i+1} - Y_{i+1} y_{i+1}\right\}$$

L'information extrinsèque z_i relative à la donnée d_i est alors simplement obtenue en effectuant la différence entre la sortie et l'entrée de ce décodeur. :

$$z_i = \Lambda(d_i) - x_i$$

qui peut encore s'écrire, en prenant en compte l'expression de $\Lambda(d_i)$:

$$z_i = \frac{1}{2}\left[\min_{s,s'}\left\{ -Y_i y_i + M_{i-1}^F(s) + M_i^B(s) \right\}_{j=0} - \min_{s,s'}\left\{ -Y_i y_i + M_{i-1}^F(s) + M_i^B(s) \right\}_{j=1} \right]$$

L'algorithme Max-Log-MAP étant une version sous-optimale de l'algorithme MAP, son utilisation dans un processus itératif entraîne une légère dégradation des performances de correction par rapport à l'algorithme original. Un écart de performance de 0,3 à 0,4 dB peut être observé en sortie du turbo-décodeur [ROB]. Cet écart peut souvent être réduit, notamment pour des blocs de données courts (jusqu'à quelques milliers de bits) en procédant à un ajustement du gain de boucle (multiplication de l'information extrinsèque par un coefficient inférieur ou égal à 1 dépendant de l'itération) et en écrêtant l'information extrinsèque.

En contre-partie, l'application de l'algorithme Max-Log-MAP ne nécessite pas la connaissance de la variance de bruit sur le canal, contrairement à l'algorithme MAP.

Références

[BAH] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 284-287, March 1974.

[ROB] P. Robertson, P. Hoeher and E. Villebrun, "A comparison of optimal and sub-optimal decoding algorithms operating in the log domain," *International Conference on Communications, ICC'95*, Seattle, USA, June 1995, pp.1009-13.

Annexe 3

Article “Characteristics of a sixteen-state turbo-encoder/decoder (turbo4)”, *International Symposium on Turbo Codes*, 1997.

Cet article décrit les caractéristiques techniques du circuit de turbo-codage et décodage modulaire conçu conjointement par l’ENST Bretagne et le CCETT en 1994 et fabriqué par ST Microelectronics.

CHARACTERISTICS OF A SIXTEEN-STATE TURBO-ENCODER/DECODER *(TURBO4)*

Michel Jézéquel*, Claude Berrou*, Catherine Douillard* and Pierre Pénard**.

*ENST de Bretagne, Technopôle Brest Iroise, BP 832, 29285 BREST Cedex FRANCE.
Phone: +33 (0)2 98 00 13 06, Fax : 33 (0)2 98 00 13 43

**CCETT, 4 rue du Clos Courtel, BP 59, 35512 CESSON SEVIGNE Cedex FRANCE.
Phone: +33 (0)2 99 12 49 05, Fax: +33 (0)2 99 12 40 98

Email: Michel.Jezequel@enst-bretagne.fr/ Claude.Berrou@enst-bretagne.fr/
Catherine.Douillard@enst-bretagne.fr/ pierre.penard@cnet.francetelecom.fr

ABSTRACT

This paper presents the characteristics of an integrated circuit called "turbo4" which can be used as a turbo-encoder or as a turbo-decoder. The turbo-encoder is built using a parallel concatenation of two recursive systematic convolutional codes with constraint length K=5. The turbo-decoder is cascadable, each circuit processing one iteration of the turbo-decoding algorithm. It is designed around 2 sixteen-modified Viterbi decoders and 2 matrices of 64 x 32 bits for interleaving and deinterleaving. Some measures for Gaussian and Rayleigh channels and for different coding rates are presented.

1 INTRODUCTION

Turbo codes are a new family of error correcting codes introduced by C. Berrou and *al.* [1,2]. They implement a parallel concatenation of recursive and systematic convolutional codes, possibly punctured. The decoding process is iterative. Therefore, the turbo-decoder can be implemented in a modular pipelined structure, in which each module is associated with one iteration. Then, performance in Bit Error Rate (BER) terms is a function of the number of chained modules. Turbo codes show results which are very close to the theoretical channel limit.

Turbo codes have been implemented in two different integrated circuits. The first one called "CAS5093" and distributed by COMATLAS is built around 5 eight-state modified Viterbi decoders [3] and 4 matrices of 32 x 32 bits for interleaving and deinterleaving. This circuit contains 2.5 modules. The second one called *turbo4* includes one module and is cascadable, so the user can create a decoder consisting of several modules.

This paper presents the characteristics of *turbo4*. It is organised as follows : the next section gives the main technical characteristics of *turbo4*. Section 3 is dedicated to the architecture of the circuit. Finally, we conclude by presenting some results of simulations and tests.

2 MAIN CHARACTERISTICS OF *TURBO4*

Turbo4 can be used as an encoder or as a decoder. It is designed with a 2-metal, 0.8µm CMOS technology. The chip with a size of 78 mm^2 contains 0.6 M transistors. Features of the circuit are shown below:

- Turbo code with constraint length K=5 (polynomials 23,35)
- Non uniform interleaving (size 64 x 32 bits)
- Cascadable decoder
- Soft output
- Coding gain for a Gaussian channel with 4 iterations for the decoding process (4 circuits for the decoder):
    - 9 dB @ BER 10^{-7}, R=1/2
    - 8 dB @ BER 10^{-7}, R=2/3
- Encoding latency: 4
- Decoding latency: 2178 per iteration
- Intrinsic self-synchronisation (R=1/2), help for synchronisation (R≠1/2) or external synchronisation
- Output giving an estimation of the channel quality

3 ARCHITECTURE

Turbo4 is built around 4 blocks: the encoder, the decoder, the interleaver/deinteleaver and the synchronisation/supervision block.

3.1 TURBO-ENCODER

The turbo-encoder is built using a parallel concatenation of two recursive systematic convolutional codes (figure 1). The incoming data (XI) is fed into a first encoder that produces redundancy Y1 while the second encoder receives interleaved data and produces redundancy Y2.

The required coding rate is obtained by puncturing Y1, Y2 and, possibly XI. For a 1/2 coding rate, the puncturing function is included in the circuit. In this case, composite redundancy is output following the sequence: Y2 Y1 Y1 Y1. For other coding rates, the puncturing function has to be designed on the board.

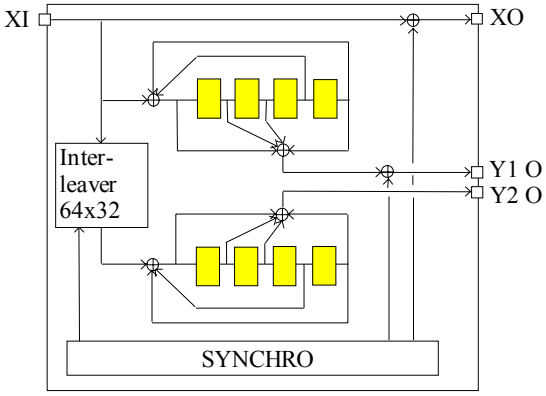


figure 1 : Turbo-encoder

A synchronisation block is added in order to make synchronisation between the encoder and the decoder possible.

3.2 TURBO-DECODER

The decoder processes one turbo-decoding iteration. It is designed (figure 2) around 2 sixteen-state SOVAs (Soft Output Viterbi Algorithm, an acronym proposed by J. Hagenauer [4]) and 2 matrices of 64x32 bits for interleaving and deinterleaving. Some delay lines are added in order to compensate for latency of other blocks like SOVAs.

The decoder is cascadable, so the user can choose the number of iterations, each circuit corresponding to one iteration. Its programmability makes it possible to adapt some coefficients to the number of the iteration. Programmability is ensured by some control inputs like: MY2[1:0], E7, MXZ, ...

The decoder receives noisy symbols X (data), Y1 (redundancy produced by the first encoder) and Y2 (redundancy produced by the second encoder) from the channel. It receives Z (extrinsic information) computed by the previous circuit. These incoming data are coded on 4 bits in 2's complement.

SOVA1 works on redundancy Y1 with noisy data X + Z. SOVA2 processes Y2 and the interleaved output of SOVA1 from which incoming data Z has been subtracted.

The X input of SOVA2 is subtracted from its output and after deinterleaving, ZO may be used by the subsequent module as input Z.

3.3 INTERLEAVING / DEINTERLEAVING

Interleaving and deinterleaving are convolutional, each function using one memory (a matrix of 64 rows by 32 columns). Data are written row by row, then read following specific rules which ensure non-uniformity of the interleaving process. Global latency due to one interleaver and deinterleaver is 2048.

3.3 SYNCHRONISATION / SUPERVISION

Synchronisation of the circuit can be divided into two phases: the synchronisation search phase and the tracking phase, also called supervising phase.

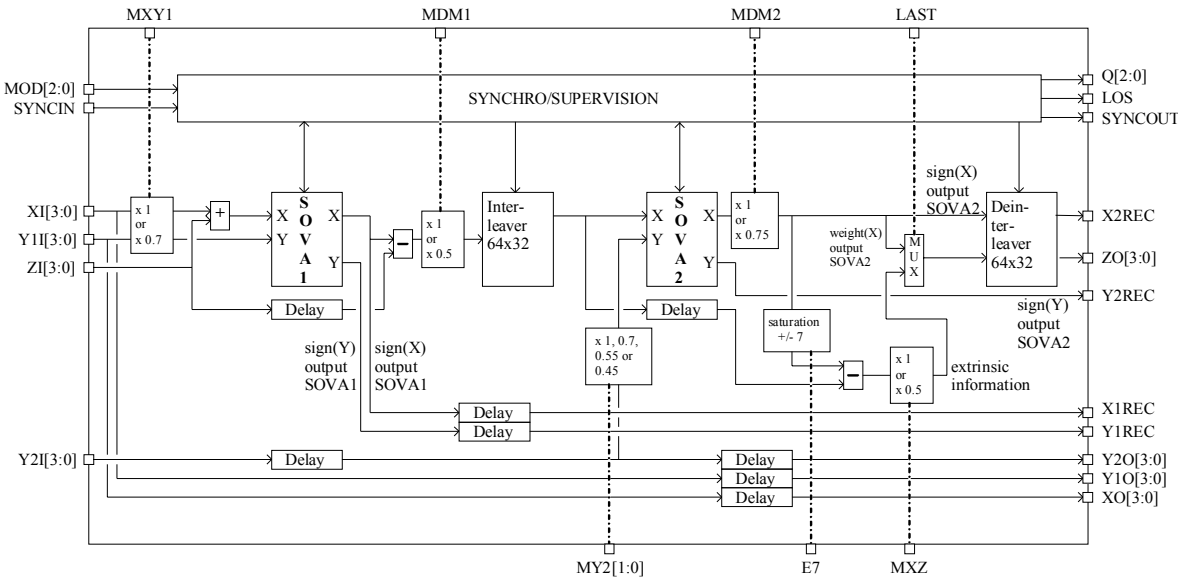


figure 2 : Turbo-decoder

No synchronisation word is required for the synchronisation search phase, thus there is no loss in the coding rate. This phase uses a controlled inversion of symbols coming from the turbo-encoder.

The supervising phase uses the pseudo-syndrome procedure [5]. This method is decorrelated from the one implemented for the synchronisation search phase. It deals with out-of-synchronisation detection and forces the circuit to return automatically to the synchronisation search phase. It also gives an estimation of the channel quality.

Tests on *turbo4* showed an error in the design of the synchronisation block. A new version of the circuit is in process in order to overcome this problem.

4 PERFORMANCE

This section presents results of simulations and tests [6].

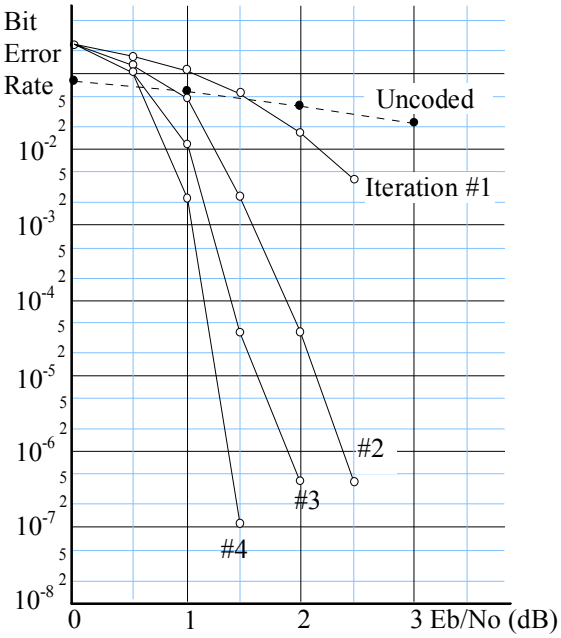


Figure 3 : Gaussian channel, R=1/3

Figure 3 and 4 present simulation results for a Gaussian channel. Figure 3 shows results for 1 to 4 modules with a global coding rate R=1/3, each code working with a coding rate equal to a half. In figure 4 simulations are made with 3 modules (3 iterations) for BPSK or QPSK modulations. These curves are given for different coding rates. If R is the global coding rate, R_1 the coding rate associated with the first code and R_2 the coding rate associated with the second code we have :

- R= 1/2 : R_1 = 4/7, R_2 = 4/5;
- R= 2/3 : R_1 = 4/5, R_2 = 4/5;
- R= 3/4 : R_1 = 6/7, R_2 = 6/7;
- R= 4/5 : R_1 = 8/9, R_2 = 8/9;

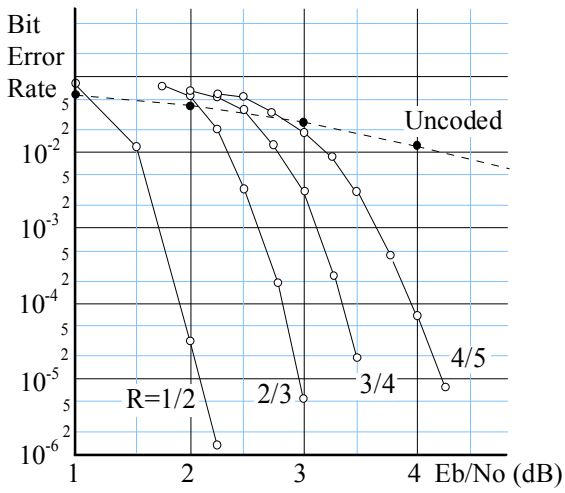


Figure 4 : Gaussian channel (3 iterations)

Figure 5 shows measured results for a Gaussian channel. Measures are made with 1 to 5 modules for a global coding rate R= 1/2 (R_1= 4/7, R_2= 4/5)

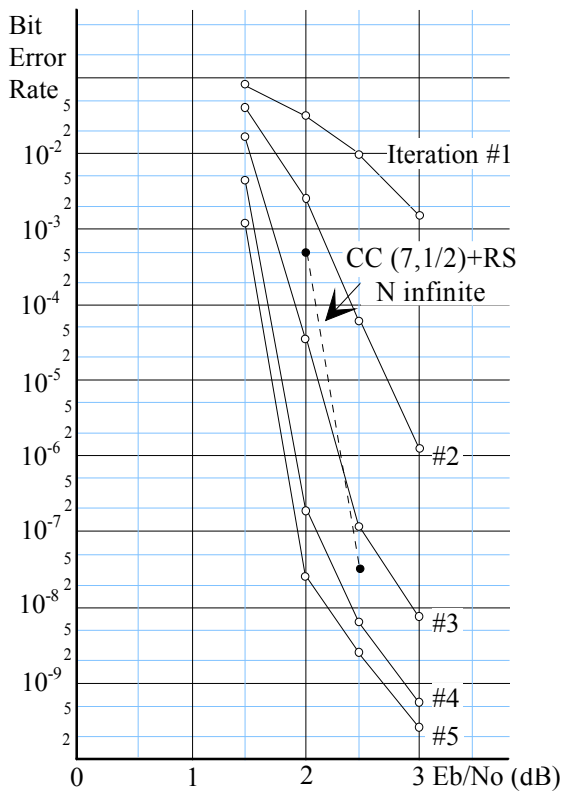


Figure 5 : Gaussian channel, R=1/2

In addition, figure 5 presents a comparison between *turbo4* performance and a classical concatenation of a convolutional code K=7 and a 8-bit (255,223) Reed-Solomon code with an infinite interleaver. This comparison shows that the coding gain given by 4 or 5 modules is better than that given by the classical concatenation for a bit error rate higher than 10^{-8}. This comparison is

made without taking into account the interleaver size which would decrease classical concatenation performance. Moreover the coding rate is more efficient for *turbo4* (R=1/2) than for the classical concatenation (R=0.437).

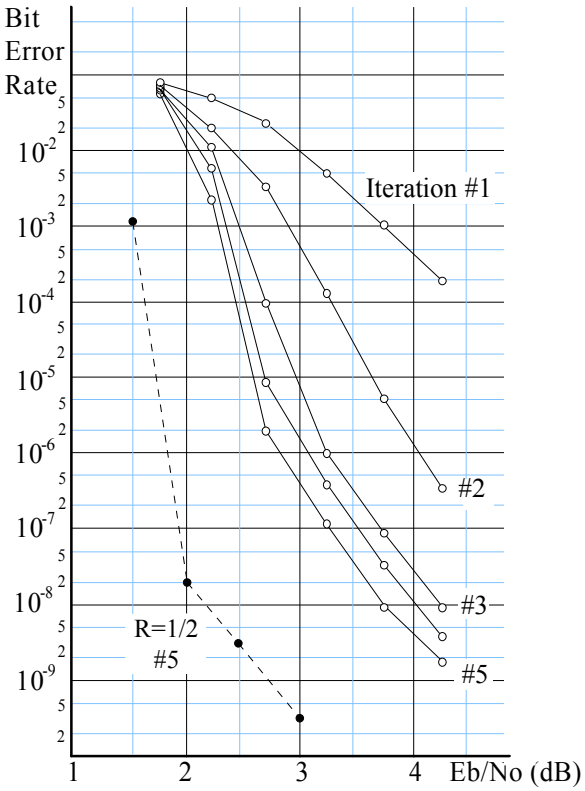


Figure 6 : Gaussian channel, R=2/3

Figure 6 presents measured results for a Gaussian channel. Measures are made with 1 to 5 modules for a global coding rate R=2/3 (R_1= 4/5, R_2= 4/5).

Figure 7 presents measured results for a Rayleigh channel with optimum channel interleaving and weighting. Measures are made with 1 to 5 modules for a global coding rate R = 1/2 (R_1 = 4/7, R_2 = 4/5). Results are most satisfactory, as the slope is the same as with the Gaussian channel with a gap of 2.5 dB.

Figure 5, 6 and 7 show a flattening degradation for low bit error rates. This degradation is not due to turbo codes but to the internal accuracy of *turbo4*, which works with 4 bits [6].

5 Conclusion

The coding gain of turbo codes was verified by tests on *turbo4*. Except for the Big Viterbi Decoder (using the constraint length 15), *turbo4* is at the moment the circuit with the best coding gain, on both Gaussian and Rayleigh channels.

The error detected in the design of the synchronisation block will be overcome in a new version of the circuit.

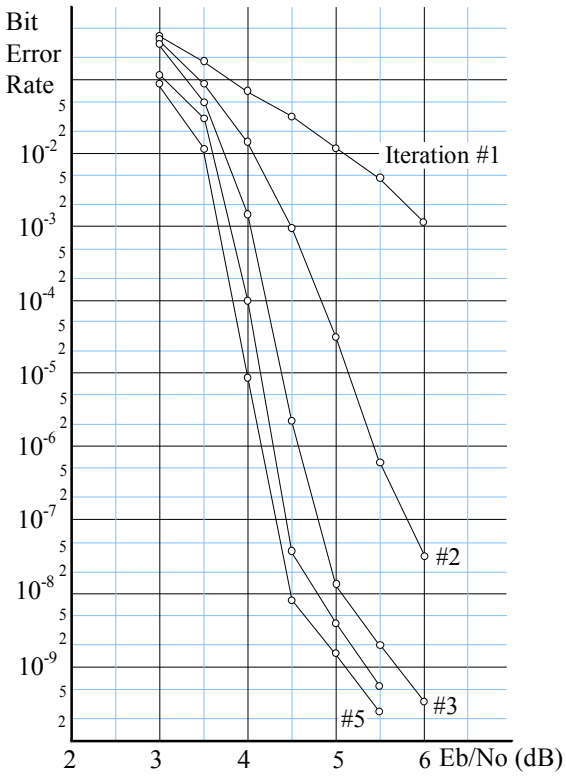


Figure 7 : Rayleigh channel, R=1/2

Acknowledgements.
The authors would like to thank P. Ferry and J.R. Inisan for their help.

References

[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, "Near optimum error correcting coding and decoding : turbo-codes", IEEE Transactions on communications, Vol. 44, N°10, pp. 1261-1271.

[3] 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.

[4] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", Proc. IEEE Globecom'89,Dallas, Texas, Nov. 1989, p.47.1.1-47.1.7.

[5] C. Berrou and C. Douillard, "Pseudo-syndrome method for supervising Viterbi decoders at any coding rate", Electron. Lett., Vol 30, n°13, pp 1036-1037, June 1994.

[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.

[7] Onyszchuk, I. M., "Coding Gains and Error Rates From the Big Viterbi Decoder," TDA PR 42-106: April-June 1991, pp.170-174, August 15, 1991.

Annexe 4

Article “Iterative correction of intersymbol interference : turbo-equalization”, *European Transactions on Telecomunications*, 1995.

Cet article décrit l’application du principe du décodage itératif au problème de la correction de l’interférence entre symboles pour des transmissions sur canaux à trajets multiples, en présence d’un code convolutif.

Iterative Correction of Intersymbol Interference: Turbo-Equalization

Catherine Douillard, Michel Jézéquel, Claude Berrou
Département Electronique
Annie Picart, Pierre Didier, Alain Glavieux
Département Signal et Communications
ENST de Bretagne
BP 832, 29285 Brest Cedex - France

Abstract. This paper presents a receiving scheme intended to combat the detrimental effects of intersymbol interference for digital transmissions protected by convolutional codes. The receiver performs two successive soft-output decisions, achieved by a symbol detector and a channel decoder, through an iterative process. At each iteration, extrinsic information is extracted from the detection and decoding steps and is then used at the next iteration as in turbo-decoding. From the implementation point of view, the receiver can be structured in a modular way and its performance, in bit error rate terms, is directly related to the number of modules used. Simulation results are presented for transmissions on Gauss and Rayleigh channels. The results obtained show that turbo-equalization manages to overcome multipath effects, totally on Gauss channels, and partially but still satisfactorily on Rayleigh channels.

1. INTRODUCTION

With the growth of mobile radio systems, digital communications have to deal with the problem of transmitting messages over multipath channels. In such cases, channels appear to be frequency-selective and give rise to Doppler effects. Several approaches may be employed in order to overcome channel selectivity. A first possibility consists in equalizing the channel so as to minimize InterSymbol Interference (ISI) at the receiving filter output. Another solution, which we have chosen, takes the channel memory effect into account. In the latter approach, the modulator, the transmission channel and the demodulator are represented by an equivalent discrete-time model that behaves similarly to a convolutional encoder. Symbol detection is based on a Maximum-Likelihood Sequence Estimation (MLSE) and is achieved through the application of the Viterbi algorithm [1]. In the case where the transmission channel is protected by a convolutional encoder and a decoder using the Viterbi algorithm, the detection and decoding modules may be associated in the same way as in turbo-decoding. In order to take the best advantage of the encoding function, the symbol detection has to provide soft outputs and the samples at the output of the equivalent discrete-time channel are processed in an iterative way. This approach is referred to as turbo-equalization in what follows, by analogy with turbo-codes.

2. PRESENTATION OF THE TRANSMISSION CHANNEL

Let us suppose that the binary digits d_k delivered by the source are encoded by a convolutional encoder. The encoded data c_k are reordered by an interleaver, and applied at the input of a Binary Phase Shift Keying (BPSK) modulator. The transmitted signal $e(t)$ is provided by the output of a filter whose impulse response is $h_e(t)$. The signal emitted can be expressed in the form:

$$e(t) = A \sum_k c_k \, h_e(t - kT) \exp j(2\pi f_0 t + \varphi_0) \qquad (1)$$

where A denotes an amplitude, f_0 is the carrier frequency, φ_0 is a constant phase whose value is in $[0, 2\pi]$, and c_k are binary symbols (± 1) transmitted at the rate of one symbol every T seconds.

In the case of a multipath channel, the received signal $Y(t)$, can be written as follows:

$$Y(t) = \sum_{m=0}^{M-1} A_m(t) \sum_k c_k \, h_e(t - \tau_m - kT) \cdot \exp j(2\pi f_0 t + \varphi_0) + B(t) \qquad (2)$$

where $A_m(t)$ are complex-valued independent multiplicative noise processes, gaussian in the case of a Rayleigh-type channel and constant in the case of a Gauss-

type channel. The delays τ_m take into account the different propagation delays on each of the M paths. $B(t)$ is a complex-valued zero mean white gaussian noise with a power bilateral spectral density equal to $2\,N_0$. After demodulation, the samples taken from the output of the receiving matched filter are expressed as:

$$R_n \triangleq R(nT) = \sum_{m=0}^{M-1} A_m(n) \sum_k c_{n-k}\, h_s(kT - \tau_m) + b_n \quad (3)$$

where $A_m(n)$ equals $A_m(nT)$ by definition, b_n denotes the response of the receiving matched filter to the noise $B(t)$, sampled at time nT. $h_s(t)$ is defined by $h_s(t) = h_e(t) \times h_e^*(-t)$ and satisfies the Nyquist criterion. Let:

$$\Gamma_k(n) = \sum_{m=0}^{M-1} A_m(n)\, h_s(kT - \tau_m) \quad (4)$$

and let us suppose that the ISI is limited to $(L_1 + L_2)$ symbols. Eq. (3) may be written in the form:

$$R_n = \sum_{k=-L_2}^{L_1} \Gamma_k(n)\, c_{n-k} + b_n \quad (5)$$

Quantities $\Gamma_k(n)$ are expressed as a linear combination of the multiplicative noises $A_m(n)$, therefore, they are gaussian in the case of a Rayleigh-type channel and constant in the case of a Gauss-type channel. Consequently, the set of modules made up of the modulator, the transmission channel and the demodulator can be represented by an equivalent discrete-time channel (Fig. 1).

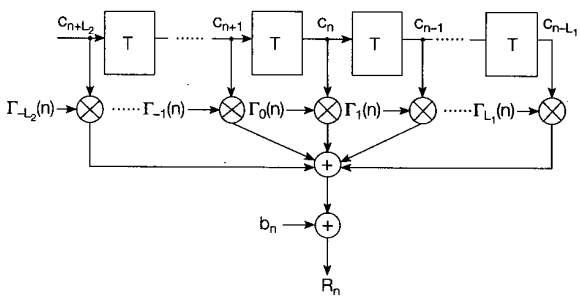


Fig. 1 - Equivalent discrete-time model of a channel with intersymbol interference.

After a change of variable k, eq. (5) may also be expressed in the form:

$$R_n = \sum_{k=0}^{L_1+L_2} \Gamma_{k-L_2}(n)\, c_{n+L_2-k} + b_n \quad (6)$$

If we denote $S_n = (c_{n+L_2}, \ldots, c_{n-L_1+1})$ the state of the equivalent discrete-time channel at time nT, sample R_n depends on the channel state S_{n-1} and on the symbol c_{n+L_2}. Therefore, the equivalent discrete-time channel

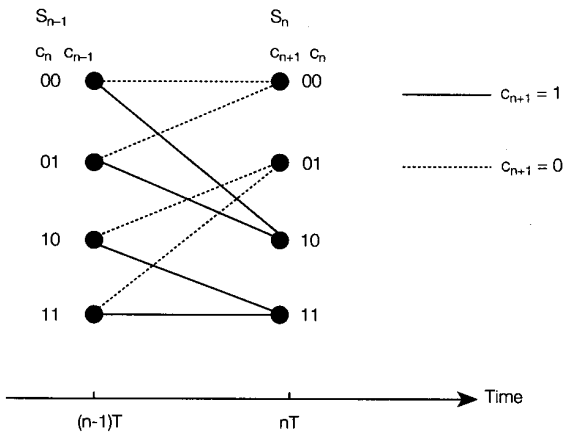


Fig. 2 - Trellis diagram for $L_1 = L_2 = 1$.

can be modeled as a Markov chain and its behavior can be represented by a trellis diagram (Fig. 2).

3. PRINCIPLE OF TURBO-EQUALIZATION

In order to use a soft-input channel decoder, the symbol detector has to provide information about the reliability of the symbols estimated. This information may be obtained by using a Soft-Output Viterbi Algorithm (SOVA) [2 - 4], that associates an estimation of the Logarithm of its Likelihood Ratio (LLR), $\Lambda_1(c_n)$, to each symbol c_n detected:

$$\Lambda_1(c_n) = \log \frac{\Pr\{c_n = +1 | \boldsymbol{R}\}}{\Pr\{c_n = -1 | \boldsymbol{R}\}} \quad (7)$$

where $\boldsymbol{R}$ denotes the vector of samples that constitutes the observation. After deinterleaving, the SOVA decoder provides a new LLR value of c_k, $\Lambda_2(c_k)$, that may be derived by analogy with the calculations made in [5] and expressed in the form:

$$\Lambda_2(c_k) = \Lambda_1(c_k) + z_k \quad (8)$$

where z_k is the extrinsic information associated with symbol c_k and provided by the channel decoder (Fig. 3). In fact, the extrinsic information z_k is another estimation of the LLR of symbol c_k conditioned on the decoding step:

$$z_k = \log \frac{\Pr\{c_k = +1 | \mathit{decoding}\}}{\Pr\{c_k = -1 | \mathit{decoding}\}} \quad (9)$$

Hence, z_k may be used through a feedback loop by the symbol detector, after interleaving. This is the basis of turbo-equalization principle.

To evaluate the LLR of symbol c_{n-L_1}, the Viterbi algorithm used in the detector has to calculate a metric at

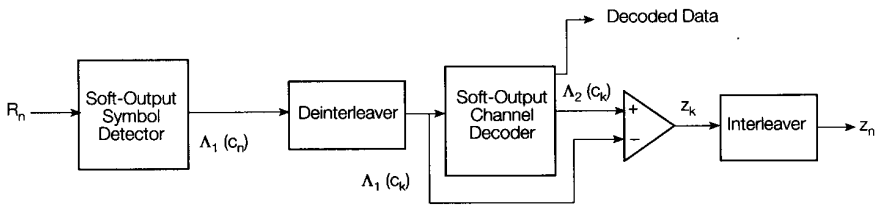


Fig. 3 - Extrinsic information deriving scheme.

time nT for every branch in the trellis. This metric may be written in the form [1]:

$$\lambda_n^i = \left| R_n - r_n^i \right|^2 - 2\sigma_b^2 \log \Pr\left\{c_{n-L_1} = i\right\} \qquad i = \pm 1 \quad (10)$$

where:

$$r_n^i = \sum_{k=0}^{L_1+L_2-1} \hat{\Gamma}_{k-L_2}(n) \cdot c_{n+L_2-k} + \hat{\Gamma}_{L_1}(n) \cdot i \qquad i = \pm 1 \quad (11)$$

and $\hat{\Gamma}_{k-L_2}(n)$, $0 \le k \le L_1 + L_2$, represents an estimation of quantity $\Gamma_{k-L_2}(n)$ and σ_b^2 denotes the variance of noise b_n, that is $\sigma_b^2 = E[|b_n|^2]$.

The *a priori* probabilities $\Pr\{c_{n-L_1} = i\}$ used in relation (10) may be estimated from the extrinsic information z_{n-L_1}, if we assume that

$$z_{n-L_1} \approx \log \frac{\Pr\left\{c_{n-L_1} = +1\right\}}{\Pr\left\{c_{n-L_1} = -1\right\}} \qquad (12)$$

Then, the following expressions can be derived from (12):

$$\Pr\left\{c_{n-L_1} = +1\right\} \approx \frac{\exp z_{n-L_1}}{1 + \exp z_{n-L_1}} \qquad (13a)$$

$$\Pr\left\{c_{n-L_1} = -1\right\} \approx \frac{1}{1 + \exp z_{n-L_1}} \qquad (13b)$$

Using (13 a), (13 b) and (10), metrics λ_n^i are equal to:

$$\lambda_n^{+1} = \left| R_n - r_n^{+1} \right|^2 - \gamma z_{n-L_1} \qquad (14a)$$

$$\lambda_n^{-1} = \left| R_n - r_n^{-1} \right|^2 \qquad (14b)$$

Note that the common term $\log(1 + \exp z_{n-L_1})$ has been suppressed in eqs. (14 a) and (14 b). Coefficient γ is a weight introduced to take into account variance σ_b^2 and the fact that the extrinsic information is only an estimation of the *a priori* probability. Its value depends on the signal-to-noise ratio, that is to say the reliability of the extrinsic information.

4. ITERATIVE IMPLEMENTATION OF TURBO-EQUALIZATION

The different processing stages in the turbo-equalizer present a non-zero internal delay, so turbo-equalizing can only be implemented in an iterative way. At each iteration, a new value of extrinsic information is calculated and used by the symbol detector at the next iteration. Therefore, the turbo-equalizer can be implemented in a modular pipelined structure, where each module is associated with one iteration. Then, performance in Bit Error Rate (*BER*) terms is a function of the number of chained modules. An example of turbo-equalization implementation is illustrated in Fig. 4 in the case of a 4-stage process. The rank q module, $1 \le q \le 4$, has two inputs and three outputs. Input R^q receives the samples from the receiving matching filter after a delay equal to the latency of the $(q-1)$ previous modules. Input z^q represents the extrinsic information of the previous iteration provided by the rank $(q-1)$ module. Output R^{q+1} is equal to input R^q delayed of the latency of the module, and z^{q+1} is the extrinsic information provided by the current iteration. These outputs are unused for the last module and do not appear on the figure. Output D^q provides the decoded data and is only used at the last module output.

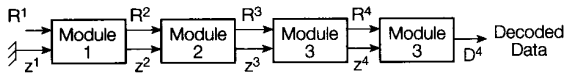


Fig. 4 - Modular pipelined structure of a turbo-equalizer for a 4-iteration process.

When extrinsic information is used by the symbol detector, it can be proved that, at iteration q, the LLR of symbol c_n, $\Lambda_1^q(c_n)$, may be expressed as:

$$\Lambda_1^q(c_n) = \hat{\Lambda}_1^q(c_n) + \gamma^q z_n^{q-1} \qquad (15)$$

where $\tilde{\Lambda}_1^q$ is a term depending on the samples of observation $\boldsymbol{R}$ and on z_n^{q-1}, $k \ne n$, and z_n^{q-1} denotes the extrinsic information of symbol c_n determined at iteration $q-1$. If we apply the same approach as in turbo-decoding, the quantity $\gamma^q z_n^{q-1}$ provided by the channel decoder at the previous iteration has to be subtracted from $\Lambda_1^q(c_n)$, as illustrated in Fig. 5. Hence, after deinterleaving, the channel decoder input is in fact equal to:

$$\tilde{\Lambda}_1^q(c_k) = \Lambda_1^q(c_k)\Big|_{z_k^{q-1}=0} \qquad (16)$$

At the channel decoder output, the extrinsic information z_k^q may also be written as follows, using (8):

$$z_k^q = \Lambda_2^q(c_k)\Big|_{\tilde{\Lambda}_1^q(c_k)=0} \qquad (17)$$

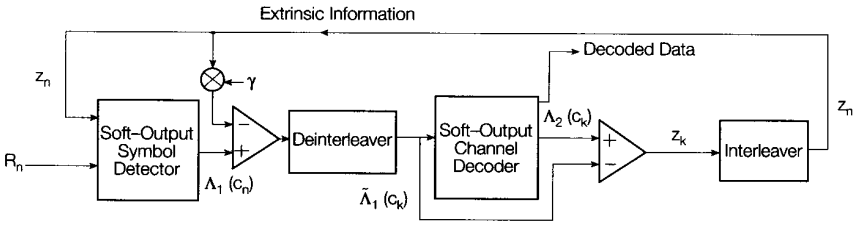


Fig. 5 - Principle of turbo-equalization (under zero internal delay assumption).

5. SIMULATION RESULTS

Performance of this device has been evaluated for a rate $R = 1/2$ recursive systematic encoder with constraint length $K = 5$, and generators $G_1 = 23$, $G_2 = 35$. Bits were interleaved in a non uniform matrix whose dimensions are 64 by 64. The modulation used was a BPSK modulation, with a Nyquist filter whose transfer function $H_s(f)$ was a raised cosine with a rolloff $\alpha = 1$, on both gaussian and Rayleigh channels.

For both channels, $M = 5$ independent paths were considered, each with a mean power $P_m = E[|A_m(n)|^2]$, so that the total mean power was normalized: $\sum_{m=0}^{M-1} P_m = 1$. The delays τ_m were chosen as multiples of T ($\tau_m = mT$, $\Gamma_k(n) = A_k(n)$ since $h_s[(k-m)T] = \delta_{k-m,0}$). The coefficients for the gaussian channel were chosen equal to:

$$\Gamma_0(n) = \sqrt{0.45}, \Gamma_1(n) = \sqrt{0.25}, \Gamma_2(n) = \sqrt{0.15}, \Gamma_3(n) = \sqrt{0.10}, \Gamma_4(n) = \sqrt{0.05}$$

On the Rayleigh channel considered, the five paths had equal mean power ($P_i = 1/M$, $\forall i \in [1, M]$). A parameter BT, which is the product of the Doppler bandwidth and the symbol duration, fixes the variation velocity of the channel: the smaller BT is, the more slowly the channel parameters vary during a time interval symbol.

The discrete-time equivalent channel was modeled by a 16-state trellis, and the symbole detector was working on the SOVA algorithm presented in [4]. The channel coefficients $\Gamma_k(n)$ were supposed perfectly known. After deinterleaving, the soft estimations provided by the SOVA detector were used by the decoder which also worked on a 16-state trellis and the SOVA algorithm. The extrinsic information extracted from the decoder was used by the symbol detector according to the principle depicted in Fig. 5.

The BER was computed as a function of signal to noise ratio E_b/N_0, where E_b is the mean energy received per information bit d_k and N_0 is the noise power bilateral spectral density. The signal to noise ratio E_b/N_0 may be expressed as:

$$\frac{E_b}{N_0} = \frac{\sum_{m=0}^{M-1} P_m}{\sigma_b^2} \tag{18}$$

Results are presented in Figs. 6, 7 and 8. On a gaussian channel (Fig. 6), the gain at the first iteration, compared to the classical Viterbi detector which provides a binary decision, is 1.7 dB for a BER of 10^{-5}. At the second iteration, the gain is 2.2 dB better. After the fifth iteration, the total gain is 5.2 dB. Finally, the turbo-equalizer manages to completely compensate for the degradation due to the interference generated by the multipath effects after six iterations. Then, the BER is the same as on a non selective gaussian channel (without intersymbol interference).

Two types of Rayleigh channels with $BT = 0.1$ (Fig. 7) and $BT = 0.001$ (Fig. 8) were examined. In both of these cases, the gain after the first iteration is about 2.2 dB, that is to say better than that on the gaussian channel. This can be explained by the fact that the system takes advantage of the diversity created by the multiple paths. The global gain after the third iteration remains inferior to that on the gaussian channel. The limit that has been considered is the BPSK modulation with encoding and without interference on a gaussian channel. This limit is not completely achieved: the third iteration is 0.8 dB from this limit for $BT = 0.1$ and 2 dB for $BT = 0.001$. A degradation is noted in the case of $BT = 0.001$ compared to the case of $BT = 0.1$, this is due to the use of the same 64 by 64 interleaving matrix in both cases. Such dimensions were chosen to simulate a realistic receiver from the implementation point of view. When $BT = 0.001$, the channel parameters

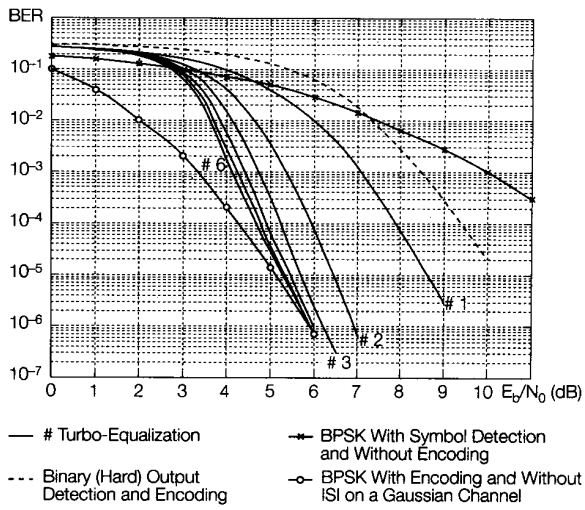


Fig. 6 - Performance of turbo-equalization over a gaussian channel (convolutional encoding with $K = 5$).

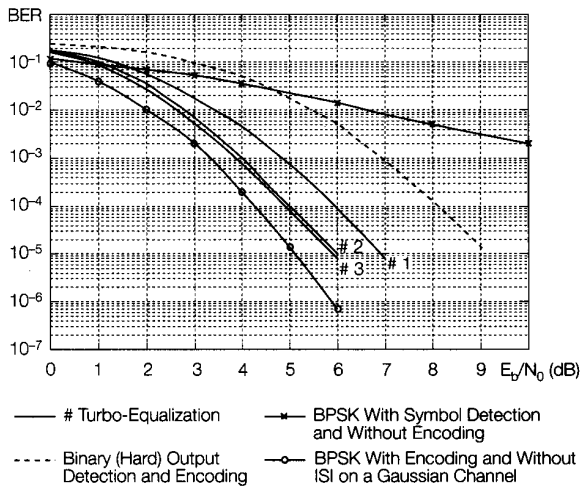


Fig. 7 - Performance of turbo-equalization over a Rayleigh channel with $BT = 0.1$ (convolutional encoding with $K = 5$).

vary so slowly that, when fading occurs, it can last so long that it can affect a number of symbols which is approximately the size of the interleaving matrix. In such a case, a 64 by 64 interleaving is less effective. Therefore, it is necessary to adapt the interleaver dimensions according to the variation speed of the channel.

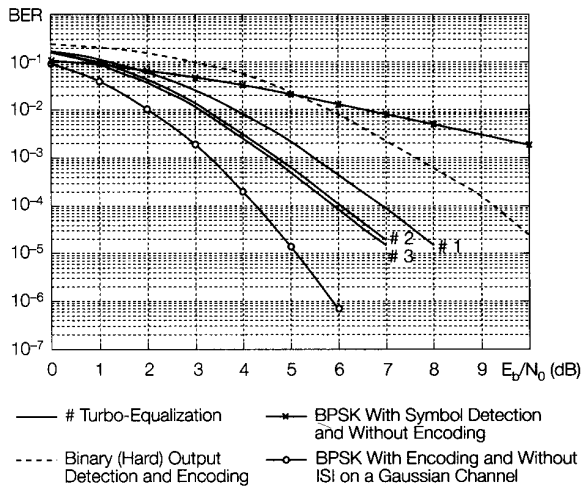


Fig. 8 - Performance of turbo-equalization over a Rayleigh channel with $BT = 0.001$ (convolutional encoding with $K = 5$).

6. CONCLUSION

In this paper a new method of combating the detrimental effects of intersymbol interference on multipath channels has been presented. It is based on the association of two successive soft-output decisions, achieved by a symbol detector and a channel decoder, in an iterative process. At each iteration a new piece of information, called extrinsic information, is calculated and used at the next iteration. By analogy with turbo-codes, this receiving scheme has been called turbo-equalization.

The receiver is made up of identical pipelined elementary modules, and the rank q elementary module uses data information coming from the demodulator and extrinsic information provided by the rank $(q - 1)$ module in order to take decisions. The performance of turbo-equalizing is directly related to the number of iterations.

The results presented in this paper have been obtained from simulations of Gauss and Rayleigh channels. In the case of a gaussian channel, the compensation for interference is complete and the behavior of a Gauss channel without multipath effects is reached after a few iterations. In the case of a Rayleigh channel, compensation is only partially achieved. However, this is still satisfactory because the bit error rate is close to that obtained on a gaussian channel without intersymbol interference. It is to be noted that these results were obtained in the case where the channel coefficients are supposed known. In practice, these coefficients have to be estimated, and the bit error rate is therefore degraded. However, further simulations are currently being processed in these conditions to show that the iterative process is also able to correct the degradation due to estimation, and turbo-equalization seems to provide results of great interest. Finally, the results obtained clearly show that turbo-equalization is an effective method of overcoming multipath effects in digital transmissions.

Acknowledgment

The authors would like to thank P. Combelles and D. Callonnec from the Centre Commun d'Etudes des Télécommunications (CCETT), and R. Schmitt from the University of Kaiserslautern, for their helpful collaboration.

Manuscript received on January 10, 1995.

REFERENCES

[1] J. G. Proakis: *Digital communications*. McGraw-Hill Series in Electrical Engineering, 2nd Edition, 1989.

[2] J. Hagenauer, P. Hoeher: *A Viterbi algorithm with soft-decision outputs and its applications*. In: Proc. IEEE Globecom' 89, Dallas, Texas, Nov 1989, p. 47.1.1-47.1.7.

[3] G. Battail: *Pondération des symboles décodés par l'algorithme de Viterbi*. (in French), "Ann. Télécomm.", No. 1-2, Jan-Fév. 1987, p. 31-38.

[4] C. Berrou, P. Adde, E. Angui, S. Faudeil: *A low complexity soft-output Viterbi decoder architecture*. In: Proc. IEEE Int. Conf. on Comm., ICC' 93, Vol 2/3, May 1993, p. 737-740.

[5] C. Berrou, A. Glavieux, P. Thitimajshima: *Near Shannon limit error-correcting coding and decoding: Turbo-codes*. In: Proc. IEEE Int. Conf. on Comm., ICC' 93, Vol 2/3, May 1993, p. 1064-1071.

Annexe 5

Article “Multiple concatenation of circular recursive systematic convolutional (CRSC) codes”, *Annales des Télécommunications*, 1999.

Cet article décrit le principe du codage circulaire des codes convolutifs récursifs systématiques et montre qu’une concaténation parallèle multiple de tels codes permet d’approcher le comportement d’un code aléatoire, conférant au turbocode ainsi obtenu une grande distance minimale.

pp. 166-172

Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes

Claude BERROU*
Catherine DOUILLARD*
Michel JÉZÉQUEL*

Abstract

This paper deals with a family of error correcting codes based on circular recursive systematic convolutional (CRSC) codes. A multiple or multidimensional parallel concatenation of CRSC codes is proposed in order to reach minimum distances comparable to those of random codes. For information blocks of size k, minimum distances as large as k/4, for a 1/2 coding rate, may be obtained if the code dimension is raised to 4 or 5. Such codes can be decoded using an iterative ("turbo") process relying on the extrinsic information concept.

Key words : Error correcting code, Convolutional code, Concatenation, Random coding, Block transmission, Turbo code.

CONCATÉNATION PARALLÈLE MULTIPLE DE CODES CONVOLUTIFS RÉCURSIFS SYSTÉMATIQUES CIRCULAIRES

Résumé

Cet article présente une famille de codes correcteurs d'erreurs bâtie autour de codes convolutifs récursifs systématiques circulaires (CRSC). Une concaténation parallèle multiple ou multidimensionnelle de codes CRSC est utilisée pour atteindre des distances minimales comparables à celles qui sont obtenues avec les codes aléatoires. Pour une taille k de bloc d'information, des distances minimales proches de k/4, pour un rendement de codage de 1/2, peuvent être ainsi obtenues lorsqu'on porte la dimension du code à une valeur de 4 ou 5. De tels codes peuvent être décodés en utilisant une procédure itérative («turbo») s'appuyant sur le concept d'information extrinsèque.

Mots clés : Code correcteur erreur, Code convolutif, Concaténation, Codage aléatoire, Transmission bloc, Turbo code.

Contents

I. *Introduction*
II. *Circular convolutional codes*
III. *Multidimensional parallel concatenation of CRSC codes*
IV. *Conclusion*
Appendix
References (16 ref.)

I. INTRODUCTION

Since Shannon's pioneering work [1], random coding has always represented a reference for error correction. The systematic random encoding of k information bits, providing n-bit codewords, requires k $(n-k)$-bit "markers" to be drawn at random, once and for all, and to be stored in a memory, where the storage address is i $(1 \leq i \leq k)$. Then, the redundancy of any k-bit information block is calculated by modulo-2 adding all the markers whose addresses i correspond to the places of logical '1's in the information block. The final codeword consists of the concatenation of the k information bits and the $n-k$ redundancy bits. The code rate R is equal to k/n. This very simple way of building up codewords, using the linearity principle, leads to large minimum distances for large enough values of $n-k$. Since two codewords may only differ on one information bit, and since the redundancy is chosen at random, the mean distance is equal to $1 + \frac{n-k}{2}$. However, as the minimum distance of this code is a random variable, its different realizations may be smaller than this mean value. Appendix A provides the expression of the probability that minimum distance $d_{\min}$ is greater than or equal to any given value D.

For usual values of n and k, decoding random codes is not practically feasible. The only way of decoding consists in inspecting the 2^k different codewords, and retaining the one closest to the received word (i.e. the most likely codeword). Therefore, the decoder complexity increases exponentially with the length k of the sequence, and becomes unusable for practical applications.

* ENST Bretagne, BP 832, F 29285 Brest Cedex, France, Claude.Berrou@enst-bretagne.fr

This paper proposes a code family imitating random codes and presenting moderate decoding complexity. We especially focus on minimum distances achieved by these codes based on a multiple parallel concatenation of recursive systematic convolutional codes. In addition, elementary encoding is organized so as to perform block encoding, thanks to the circular coding concept, which is developed in the next chapter.

II. CIRCULAR CONVOLUTIONAL CODES

Convolutional codes are not *a priori* really suited for encoding information transmitted in block form. Knowing the initial state of the trellis is not a problem, as the "all zero" state is, in general, forced by the encoder. However, the decoder has no special information available regarding the final state of the trellis. This problem is even more serious for N-dimensional turbo codes (i.e. having N component encoders), since the decoder does not know the final states of the N trellises after the N encoding processes. Several answers can solve this problem, for example:

– **doing nothing**, that is, no information concerning the final states of the trellises is provided to the decoder. The decoding process is less effective for the last encoded data and the asymptotic gain may be reduced. This degradation is a function of the block length and may be low enough to be accepted for a given application.

– **forcing the encoder state** at the end of the encoding phase for one or all dimensions. This solution has been adopted by the CCSDS standard [2] for instance. Tail bits are used to "close" the trellises and are then sent to the decoder. This method presents two major drawbacks. First, minimum weight w_{min} is no longer equal to 2 for all information data[1], since, at the end of each block, the second '1' bringing the encoder back to the " all zero " state may be a part of the tail bits. In this case, turbo decoding is handicapped if tail bits are not encoded another time. Next, the spectral efficiency of the transmission is degraded and the cost is increased as the blocks are shorter and N higher.

– adopting **circular coding**. With circular convolutional codes, the encoder retrieves the initial state at the end of the encoding operation. The decoding trellis can therefore be seen as a circle and decoding may be initialized everywhere on this circle. This technique, well known for non recursive codes (so-called "tail-biting"), has been adapted to the specificity of recursive codes.

1. The weight w of a binary word is defined as the number of information bits equal to '1', that is the number of information bits differing from the "all zero" word, which is used as a reference for linear codes. For a recursive code, when the final states are fixed by the encoder, the minimum value for w is 2. For more details, see [3-5] for example.

II.1. The principle of circular recursive systematic convolutional (CRSC) codes

Let us consider a recursive convolutional encoder, for instance the encoder depicted in Figure 1 (quaternary or double-binary code with memory v equal to 3). At time i, register state S_i is a function of previous state S_{i-1} and input vector X_i. Let G be the generator matrix of the considered code. States S_i and S_{i-1} are linked by the following recursion relation:

$$S_i = G\,S_{i-1} + X_i. \tag{1}$$

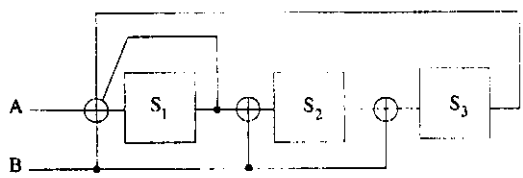


FIG. 1. — Recursive convolutional (double binary) encoder with memory $v = 3$. The output, which is not relevant to the operation of the shift register, has been omitted

Codeur convolutif récursif (double binaire) avec mémoire de code $v = 3$. La sortie, qui n'intervient pas dans le comportement du registre à décalage, n'est pas représentée

For Figure 1 encoder, vectors S_i and X_i, and matrix G are given by:

$$S_i = \begin{bmatrix} s_{1,i} \\ s_{2,i} \\ s_{3,i} \end{bmatrix};\ X_i = \begin{bmatrix} A_i + B_i \\ B_i \\ B_i \end{bmatrix};\ G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

From (1) we can infer:

$$\begin{aligned} S_i &= G\,S_{i-1} + X_i, \\ S_{i-1} &= G\,S_{i-2} + X_{i-1}, \\ &\vdots \\ S_1 &= G\,S_0 + X_1. \end{aligned}$$

Hence, S_i may be expressed as a function of initial state S_0 and of data feeding the encoder between times 1 and i:

$$S_1 = G^i S_0 + \sum_{p=1}^{i} G^{i-p} X_p. \tag{2}$$

If k is the input sequence length, it is possible to find a state S_c such that $S_c = S_k = S_0$. Its value is derived from (2):

$$S_c = \left(I + G^k\right)^{-1} \sum_{p=1}^{k} G^{k-p} X_p \tag{3}$$

where I is the identity matrix.

State S_c depends on the sequence of data and exists only if $I + G^k$ is invertible[2]. In particular, k cannot be a multiple of the period L of the encoding recursive generator, defined as:

$$G^L = I.$$

S_c is the circulation state. That is, if the encoder starts from state S_c, it comes back to the same state when the encoding of the k data (k couples for the Figure 1 encoder) is completed. Such an encoding process is called circular because the associated trellis may be viewed as a circle, without any discontinuity on transitions between states.

Determining S_c requires a pre-encoding operation. First, the encoder is initialized in the "all zero" state. Then, the data sequence of length k is encoded once, leading to final state S_k^0. Thus, from (2):

$$S_k^0 = \sum_{p=1}^{k} G^{k-p} X_p.$$

Combining this result with (3), the value of circulation state S_c can be linked to S_k^0 as follows:

$$S_c = \left(I + G^k\right)^{-1} S_k^0. \tag{4}$$

In a second operation, data are definitely encoded starting from state S_c.

In practice, the relation between S_c and S_k^0 is provided by a small combinational operator with v input bits and v output bits, if v represents the code memory. The disadvantage of this method lies in having to encode the sequence twice: once from the "all zero" state and the second time from state S_c. Nevertheless, in most cases, the double encoding operation can be performed at a frequency much higher than the data rate, so as to reduce the latency effects.

II.2. Circular codes and turbo codes

Circular codes are well suited to turbo decoding concepts. For example, let us consider the binary turbo encoder in Figure 2. The data sequence to be encoded, made up of k information bits, feeds the CRSC encoder input twice: first, in the natural order of the data (switch in position 1), and next in an interleaved order, given by time permutation function Π (switch in position 2). In fact, the circular code principle may be applied in two slightly different ways, according to whether the code is self-concatenated or not.

Case 1: the code is self-concatenated, that is the second encoding step directly follows on from the first step without intermediate reinitializing of the register state. Circulation state S_c is calculated for the whole sequence of length $2k$. At reception, the decoder performs a decoding of the double length sequence.

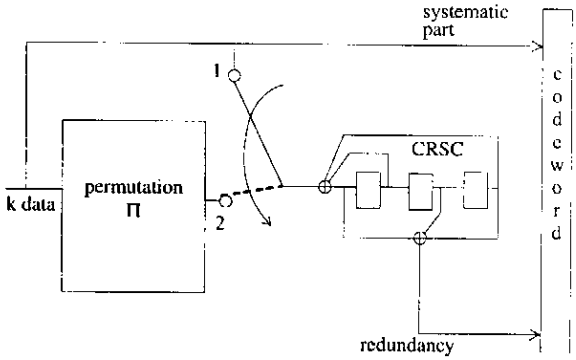


Fig. 2. — Turbo encoder built from a CRSC code

Turbo codeur utilisant un code CRSC

Case 2: the code is not self-concatenated that is the encoder is initialized at the beginning of each encoding stage. Two circulation states S_{c_1} and S_{c_2}, corresponding to both encoded sequences, are calculated. At reception, both sequences of length k are decoded separately.

Depending on the case, data encoding may be represented by one or two circular trellises. Whatever elementary algorithm is used, iterative decoding requires repeated turns around the circular trellis(es), the extrinsic information table being continuously updated during data processing. Iterations naturally follow one after the other without any discontinuity between transitions from state to state.

In the case where the APP algorithm (also called MAP, backward-forward, or BCJR [6] algorithm), or one of its simplified versions [7] is applied, decoding the sequence consists of going round the circular trellis anticlockwise for the backward process, and clockwise for the forward process (Fig. 3), during which data are decoded and extrinsic information is built. For both processes, probabilities computed at the end of a turn are used as initial values for the next turn. The number of turns performed around the circular trellis(es) is equal to the number of iterations required by the iterative process. In practice, the iterative process is preceded by a "prologue" decoding step, performed on a part of the circle for a few v. This is intended to guide the process towards an initial state which is a good estimate of the circulation state.

If the decoding algorithm is a soft-output version of the Viterbi algorithm [8-10], iterative decoding requires the circular trellis to be processed anticlockwise for metrics computation, with partial clockwise returns to calculate decisions and extrinsic information. Like with the APP algorithm, iterations naturally follow one after

2. Note that some matrices G are not suitable.

the other, and the metrics computed at the end of each turn are used as initial metrics for the next iteration. An estimate of the circulation state may also be obtained by a "prologue" step.

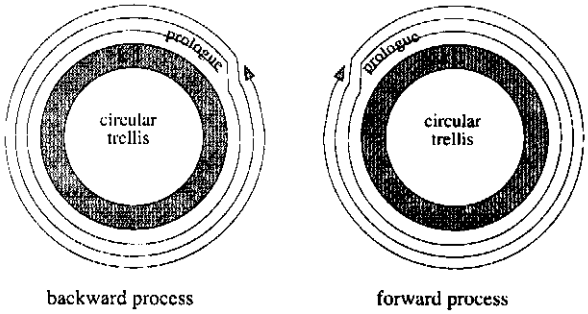


FIG. 3. — Processing a circular code by the backward-forward algorithm

Le traitement d'un code circulaire par l'algorithme aller-retour

III. MULTIDIMENSIONAL CONCATENATION OF CRSC CODES

Let us consider the encoding process of a k-data block, using the parallel concatenation of N CRSC codes in a similar way to a standard turbo code [3]. Figure 4 represents the whole encoder and the N corresponding circular trellises. Permutations $\Pi_j (1 \leq j \leq N)$ are chosen randomly, except the first one which is the identity permutation. In order to have $R = 1/2$ (as a particular example), each elementary encoder provides k/N redundancy bits Y_j^p $(1 \leq j \leq N,\ 1 \leq p \leq k\ /\ N)$, regularly emitted during the sequence encoding process[3]. The study of the distance properties of this multiconcatenated code is not an obvious problem in general and would not give any practical information of interest. However, three particular cases, corresponding to values of N equal to 1, 2, and k, can be considered. For $N = 1$, we have to deal with a simple convolutional code; for $N = 2$, the code is a classical turbo code. These two particular cases have already been widely examined, especially in [11] for a turbo code using a random permutation.

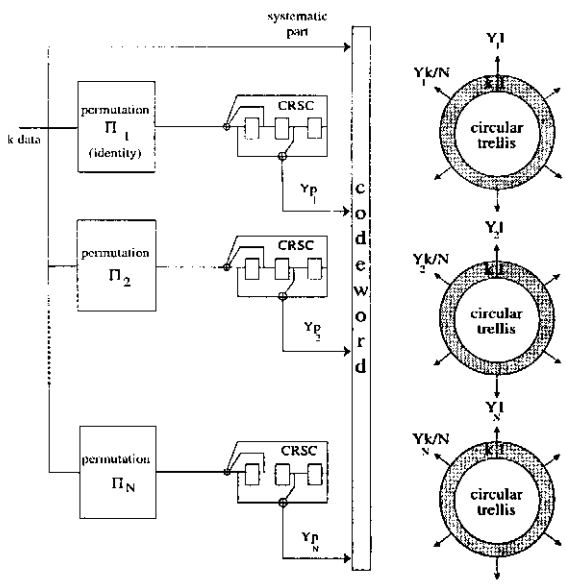


FIG. 4. — Multiple parallel concatenation of N CRSC codes

Concaténation parallèle multiple de N codes CRSC

The extreme case $N = k$, where each elementary encoder provides only one redundancy symbol Y_j^1, behaves in a way very similar to random coding. Let us assume that symbol Y_j^1 is calculated when the sequence encoding starts, that is when the encoder is in the circulation state S_c. Then, two cases have to be examined, according to the value of the weight w of the sequence.

$w = 1$. Relation (3) shows that the circulation state S_c can never be the "all zero" state (besides, the encoder never goes to the "all zero" state during all the encoding process, whatever the permutation). The redundancy symbol may be '0' or '1' with equal probability.

$w \geq 2$. The circulation state may be any state. The probability that S_c is different from the "all zero" state is $\frac{2^v - 1}{2^v}$ and all non-zero states is equiprobable. Consequently, whatever the input bit to be encoded, the probability that the redundancy symbol is equal to '1' is $\frac{1}{2}\ \frac{2^v - 1}{2^v}$. For values of v great enough (say $v \geq 3$), it can be assumed that both logical values for Y_j^1 are equiprobable.

Thus, the code redundancy is made up of a set of values Y_j^1 $(1 \leq j \leq k)$ comparable to a set of binary random values: the value of each redundancy symbol Y_j^1 depends on S_c, which is itself directly linked to the permutation implemented at level j of the concatenated encoder, and this permutation is drawn at random. Hence this code must show minimum distances of the same order as those allowed by probabilities given by relation (A-4). In order to confirm this assertion, we programmed, for different dimensions, three multiconcatenated codes associated with three values of k retained as examples in Figure A-1: 40, 60, and 80. The elementary encoders used the same generator polynomials 15, 13 ($v = 3$). The program tried to obtain minimum distances as great as possible[4] for each case and by trying different permutations. Results are presented in Figure 5. We can notice that minimum distances $d_{\min}$

3. If k is not a multiple of N, some minor adjustments may be done on the distribution of the redundancy bits stemming from the encoders.

4. The search, which can last several days on an Ultra Sparc computer, was performed for weight values up to 8. It was not possible to go further. Nevertheless, it can be shown (numerically) that the result of the multiplication in (A-4) is not much affected in limiting increment w to value 8, for $k \leq 80$. This is most probably the same for $d_{\min}$.

increase with dimension N, but do not exceed the values predicted by the curves in Figure A-1, that is about $k/4$. Even if we performed an exhaustive search with a more powerful computer, the probabilities of obtaining greater values would be very small. Besides, the maximum values were obtained for dimensions much lower than maximum dimension k, i.e. in the order of 4 or 5 for the cases considered. Consequently, it is not necessary to use maximum dimension codes to get large values for d_{min}. Much lower dimensions than k provide features comparable to those of random codes.

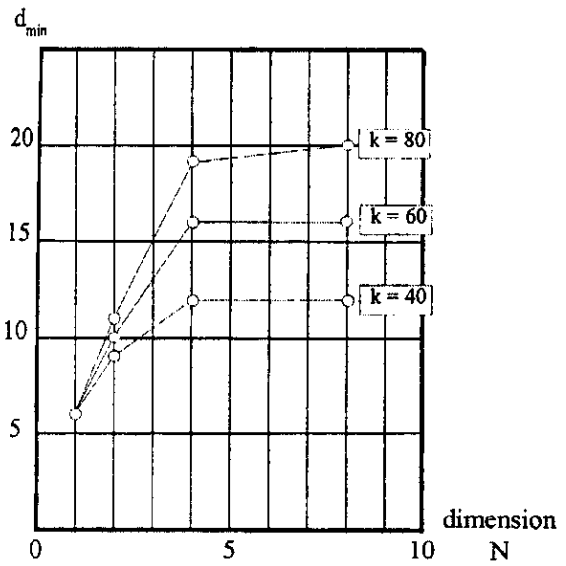


FIG. 5. — Minimum distances as a function of the dimension N of the code, for block sizes $k = 40, 60, 80$ and coding rate 1/2. Results obtained by computer search

Distances minimales en fonction de la dimension N du code, pour des blocs de taille k = 40, 60, 80 et un rendement de codage 1/2. Résultats obtenus par recherche sur ordinateur

Multiconcatenated codes may be decoded using the "turbo" principle. Each elementary decoder in place j ($1 \leq j \leq N$) provides for each binary information data d_i ($1 \leq i \leq k$), an extrinsic piece of information Z_j^i, as a probability or a LLR (logarithm of likelihood ratio). Each decoder in place j' takes advantage of the work performed by the $N - 1$ other decoders which pass on their extrinsic piece of information $Z_{j \neq j'}^i$ to it. At each decoder input, the different available extrinsic values relating to the same datum are simply combined through a product, for probabilities, or a sum, for LLR.

However, turbo decoding is not optimal [12], so it is quite possible that, for great values of N (i.e. for elementary encoders with high redundancy rate) the loss due to iterative decoding is increased. Thus, it would not be possible to take a complete advantage of the error-correcting capability of large dimension codes. Fortunately, we have shown that, with dimensions 3, 4, or 5, distances close to maximum values are obtainable and that turbo decoding suboptimality does not seem to be a real handicap (e.g., see [13] for a three-dimensional turbo code).

IV. CONCLUSION

We have shown that a maximum dimension concatenation of CRSC codes may be compared to random coding. Hence, minimum distances of such codes are large, in the order of a quarter of the block length. In fact, an experiment with several values of k led us to observe that these large values of minimum distance could be obtained for dimensions much lower than k, but greater than 2, the dimension of a standard turbo code.

Two-dimensional turbo codes can be penalized because of insufficient minimum distances for certain applications, especially for short blocks when a very low bit error rate is required. A possible answer to this problem consists in increasing the code dimension, up to 3, 4, or 5 in practice. In return, hardware complexity is raised by 50, 100 or 150 %, for the same decoding latency. For bit error rates greater than 10^{-5} typically, increasing the dimension of the turbo code beyond $N = 2$ does not give any extra gain.

From a more conceptual point of view, a remark can be made on the codes presented in this paper in relation to Shannon's statistical approach. When establishing his famous theorems, he considered a set of random codes and he predicted average statistical performance results, arguing that, amongst all the possible codes, there is at least one which shows performance equal or higher than average. Now, considering a multiple parallel concatenation of CRSC codes, and especially in the extreme case $N = k$, we can take the advantage of the whole error-correcting capability only if **all the codes are actually used, and not only the best one** (that we would be unable to define). Nowadays, the "turbo" technique allows a great number of linked codes to be decoded together. It appears in fact to be an indirect means of taking advantage of a statistically good code, without having to search for the best one amongst all the possibilities.

APPENDIX

The purpose of this appendix is to calculate the probability Pr $\{d_{min} \geq D\}$ that a codeword produced by a random encoding of k-bit data blocks, with code rate 1/2, has a minimum distance d_{min} greater than or equal to D. The principle of random coding is described in section I.

As the code is linear, distances may be determined relatively to the "all zero" word. The distance d of a

codeword may be expressed as the sum of the weights (that is, the number of logical '1's) of the systematic part w and of the redundancy d_r:

$$d = w + d_r \tag{A-1}$$

For a given weight w, $\binom{k}{w}$ redundancy sequences or sequence combinations (obtained by modulo-2 additions) have to be considered. The probability that one of these sequences or sequence combinations contains at least $D - w$ '1' is:

$$\Pr\{d_r \geq D - w \,/\, \text{one combination of weight } w\} = 2^{-k} \sum_{i=D-w}^{k} \binom{k}{i} \tag{A-2}$$

The probability that all the redundancy sequences or combinations contain at least $D - w$ '1' is:

$$\Pr\{d_r \geq D - w \,/\, \text{all combinations of weight } w\} = \left[2^{-k} \sum_{i=D-w}^{k} \binom{k}{i}\right]^{\binom{k}{w}} \tag{A-3}$$

In order for all the codewords to have a distance greater than or equal to D, the above relation has to be multiplied by itself for all the possible values of w, from 1 to $D - 1$. Therefore,

$$\Pr\{d_{\min} \geq D\} = \prod_{w=1}^{D-1} \left[2^{-k} \sum_{i=D-w}^{k} \binom{k}{i}\right]^{\binom{k}{w}}$$

This relation may also be expressed as:

$$\Pr\{d_{\min} \geq D\} = \prod_{w=1}^{D-1} \left[1 - 2^{-k} \sum_{i=0}^{D-w-1} \binom{k}{i}\right]^{\binom{k}{w}} \tag{A-4}$$

This probability is plotted in Figure A-1 as a function of D for six values of k, from 40 to 140. It can be noticed that a minimum distance $d_{\min}$ in the order of $k/4$ can be easily reached, but because of the steep decrease in the probabilities, it seems unrealistic to get distances beyond this typical value, even with very powerful computers which would try to adapt redundancy patterns so as to raise $d_{\min}$. For a more theoretical approach, see for instance [14-16]. In particular, note that the results are in accordance with the Gilbert-Varshamov lower bound on the minimum distance, that is 0.22 k for $R = 1/2$ [15].

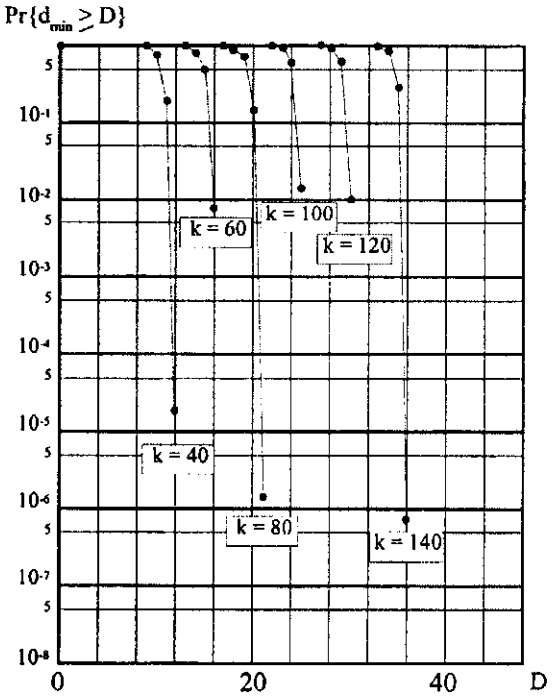


FIG. A-1. — Probability that random coding of a k-bit data word, with coding rate 1/2, leads to minimum distance $d_{\min}$ greater than or equal to D

Probabilité pour que le codage aléatoire de mots de k bits d'information, avec rendement de codage 1/2, conduise à une distance minimale d_{min} au moins égale à D

Acknowledgements

The authors wish to thank Professors Gerard Battail (ENST), Alain Glavieux (ENST Bretagne) and Shlomo Shamai (Shitz) (Technion) for their useful help.

Manuscrit reçu le 27 février 1999

REFERENCES

[1] SHANNON (C. E.). A mathematical theory of communication, *Bell System Technical Journal*, 27, (July and October 1948).

[2] *** Draft CCSDS recommendation for telemetry channel coding (updated to include turbo codes), *Consultative Committee for Space Data Systems*, Rev. 4, (May 1998).

[3] BERROU (C.), GLAVIEUX (A.). Near optimum error correcting coding and decoding : turbo-codes, *IEEE Trans. Com*, **44**, n° 10, pp. 1261-1271, (Oct. 1996).

[4] BERROU (C.), GLAVIEUX (A.). Turbo codes, general principles and applications, *Proc. of the 6th Int. Tirrenia Workshop on Digital Communications*, Pisa, Italy, pp. 215-226, (Sept. 1993).

[5] THITIMAJSHIMA (P.). Les codes convolutifs récursifs systématiques et leur application à la concaténation parallèle, Ph. D. n° 284, *Université de Bretagne Occidentale*, Brest, France, (Dec. 1993).

[6] BAHL (L. R.), COCKE (J.), JELINEK (F.), RAVIV (J.). Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Inform. Theory*, **20**, pp. 248-287, (Mar. 1974).

[7] ROBERTSON (P.), HOEHER (P.), VILLEBRUN (E.). Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding, *European Trans. Telecommun.*, **8**, pp. 119-125, (March-Apr. 1997).

[8] BATTAIL (G.). Pondération des symboles décodés par l'algorithme de Viterbi, *Ann. Télécommun.*, Fr., **42**, n° 1-2, pp. 31-38, (Jan. 1987).

[9] HAGENAUER (J.), HOEHER (P.). A Viterbi algorithm with soft-decision outputs and its applications, *Proc. of Globecom 89*, Dallas, Texas, pp. 47. 11-47-17, (Nov. 1989).

[10] BERROU (C.), ADDE (P.), ANGUI (E.), FAUDEIL (S.). A low complexity soft-output Viterbi decoder architecture, *Proc. of ICC '93*, Geneva, pp. 737-740, (May 1993).

[11] BENEDETTO (S.), MONTORSI (G.). Unveiling turbo-codes : some results on parallel concatenared coding schemes, *IEEE Trans. IT.*, **42**, n° 2, pp. 409-429, (Mar. 1996).

[12] BERROU (C.). Some clinical aspects of turbo codes, *Proc. of Int. Symposium on Turbo Codes*, Brest, France, pp. 26-31, (Sept. 1997).

[13] LI PING. Modified turbo codes with low decoding complexity, *Electronics. Letters*, **34**, n° 23, pp. 2228-2229, (Nov. 1998).

[14] PIERCE (J.N.). Limit distribution of the minimum distance of random linear codes, *IEEE Trans. IT*, **13**, n° 4, (Oct. 1967).

[15] SVIRID (Y. V.), Weight distributions and bounds for turbo codes », *Eur. Trans. Telecomm.*, **6**, n° 5, pp. 543-555, (Sept./Oct. 1995).

[16] BATTAIL (G.). Théorie de l'information, application aux techniques de communication, *Masson*, Paris, (1997).

Annexe 6

Article “Turbo codes with rate-m / m + 1 constituent convolutional codes”, *à paraître dans IEEE Transactions on Communications*.

Cet article présente la structure des codes convolutifs *à m* entrées binaires proposés et les avantages de leur utilisation pour la construction de turbocodes. Des exemples de performance sont donnés pour le cas $m = 2$.

Turbo Codes with Rate-m / (m + 1) Constituent Convolutional Codes

Catherine Douillard and Claude Berrou

PRACOM/ENST Bretagne
Technopole Brest-Iroise - CS 83818

29 238 Brest Cedex

France

Abstract

The original turbo codes, presented in 1993 by Berrou *et al.*, consist of the parallel concatenation of two rate-1/2 binary Recursive Systematic Convolutional (RSC) codes. This paper explains how replacing rate-1/2 binary component codes by rate-m / (m + 1) binary RSC codes can lead to better global performance. The encoding scheme can be designed so that decoding can be achieved closer to the theoretical limit while showing better performance in the region of low error rates. These results are illustrated with some examples based on double-binary (m = 2) 8-state and 16-state turbo codes, easily adaptable to a large range of data block sizes and coding rates. The double-binary 8-state code has already been adopted in several telecommunication standards.

Keywords

turbo code, rate-m / (m + 1) RSC code, iterative decoding, permutation.

I. Introduction

A classical turbo code (TC) [1] is a parallel concatenation of two binary Recursive Systematic Convolutional (RSC) codes based on single-input Linear Feedback Shift Registers (LFSRs).

The use of multiple-input LSFRs, which allows several information bits to be encoded or decoded at the same time, offers several advantages compared to classical TCs. In the past, the parallel concatenation of multiple-input LFSRs had mainly been investigated for the construction of turbo trellis-coded modulation schemes [2][3][4], based on Ungerboeck trellis codes. Actually, the combination of such codes, providing high natural coding rates, with high-order modulations leads to very powerful coded modulation schemes.

In this paper, we propose the construction of a family of TCs calling for RSC constituent codes based on m-input LFSRs, that outperforms classical TCs. We provide two examples of TCs with reasonable decoding complexity, that allow decoding to be achieved very close to the theoretical limit and, at the same time, show good performance in the region of low error rates.

Section II describes the structure adopted for m-input RSC component encoders, along with some conditions that guarantee large free distances regardless of coding rate.

In section III, we describe the turbo encoding scheme, and the advantages of this construction compared with classical TCs.

Section IV presents some practical examples of TCs with $m = 2$ and their simulated performance. The 8-state family has already been adopted in the DVB standards for return channel via satellite (DVB-RCS) [5] and the terrestrial distribution system (DVB-RCT) [6], and also in the 802.16a standard for local and metropolitan area networks [7]. Combined with the powerful technique of circular trellises, this $m = 2$ TC offers good performance and versatility for encoding blocks with various sizes and rates, while keeping reasonable

decoding complexity. Replacing the 8-state component encoder by a 16-state encoder allows better performance at low error rates, at the price of a doubled decoding complexity. Minimum Hamming distances are increased by 30% to more than 50%, with regard to 8-state TCs, and allow Frame Error Rate (FER) curves to decrease below 10^{-7} without any noticeable change in the slope (the so-called *flattening* effect).

Finally, conclusions and perspectives are summarized in Section V.

II. RATE-*m* / (*m* + 1) RSC ENCODERS BASED ON *m*-INPUT LFSRS

In this section, we will define the constituent RSC codes to be used in the design of the proposed TCs. Fig. 1 depicts the general structure of the RSC encoder under study. It involves a single ν-stage LFSR, whose ν-row and ν-column generator matrix is denoted $\mathbf{G}$. At time *i*, the *m*-component input vector $\mathbf{d}_i = \left(d_{i,1} \cdots d_{i,l} \cdots d_{i,m}\right)^T$ is connected to the ν possible taps via a connection grid represented by a ν-row and *m*-column binary matrix denoted $\mathbf{C}$. The tap column vector at time *i*, $\mathbf{T}_i$, is then given by:

$$\mathbf{T}_i = \mathbf{C}\mathbf{d}_i \tag{1}$$

In order to avoid parallel transitions in the corresponding trellis, the condition $m \leq \nu$ has to be satisfied. Except for very particular cases, this encoder is not equivalent to a single input encoder fed successively by $d_{i,1}, d_{i,2}, \cdots d_{i,m}$, that is, the *m*-input encoder is not generally decomposable.

The redundant output of the machine – not represented in Fig. 1 – is calculated, at time *i*, as:

$$y_i = \sum_{j=1\ldots m} d_{i,j} + \mathbf{R}\mathbf{S}_i \tag{2}$$

where $\mathbf{S}_i$ denotes the ν-component column vector describing the encoder state at time *i* and $\mathbf{R}$ is the ν-component row redundancy vector. The pth component of $\mathbf{R}$ is equal to 1 if the pth component of $\mathbf{S}_i$ is present in the calculation of y_i, and 0 otherwise.

The code being linear, we assume that the "all zero" sequence is encoded. Let us define a *Return To Zero* (RTZ) sequence as an input sequence of a recursive encoder, that makes the encoder leave state $\mathbf{0}$ and return to it again. Calculating the minimum free distance of such a code involves finding the RTZ sequence path with minimum output Hamming weight.

Leaving the null path at time *i* implies that $\mathbf{S}_i \equiv \mathbf{0}$ and, at least one component of $\mathbf{d}_i$ is equal to 1. In this case, (2) ensures that the Hamming weight of $(d_{i,1}, d_{i,2}, \cdots d_{i,m}, y_i)$ is at least 2 when leaving the reference path, since the inversion of one component of $\mathbf{d}_i$ implies the inversion of y_i.

Moreover, as

$$\mathbf{S}_{i+1} = \mathbf{G}\mathbf{S}_i + \mathbf{T}_i \tag{3}$$

it can be shown that y_i may also be written as

$$y_i = \sum_{j=1\ldots m} d_{i,j} + \mathbf{R}\mathbf{G}^{-1}\mathbf{S}_{i+1} \tag{4}$$

on the condition that

$$\mathbf{R}\,\mathbf{G}^{-1}\,\mathbf{C} \equiv \mathbf{0} \tag{5}$$

Consequently, if the code is devised to verify condition (5), expression (4) ensures that if the RTZ sequence retrieves the all-zero reference path at time i ($\mathbf{S}_{i+1} \equiv \mathbf{0}$) because one of the information bits $d_{i,j}$ is equal to 1, y_i is also equal to 1.

Hence, relations (2) and (4) together guarantee that the minimum free distance of the unpunctured code, whose rate is $R = m / (m + 1)$, is at least 4, whatever m. In practice, defining codes satisfying condition (5) does not pose a real problem. The two codes proposed in this paper for $m = 2$ meet this requirement.

The minimum Hamming distance of a concatenated code being larger than that of its constituent codes, provided that the permutation function is carefully devised, we can imagine that large minimum distances may be obtained for TCs, for low as well as for high coding rates.

Choosing large values of m implies high decoding complexity because ν also has to be large and 2^m paths have to be processed for each trellis node. For this reason, only low values of m can be contemplated for practical applications for the time being (typically $m = 2$, possibly 3 or 4).

Up to now, we have only investigated the case $m = 2$ in order to construct practical coding and decoding schemes in this new family of TCs. In the sequel, we call such $m = 2$ RSC or turbo encoders *double-binary*. Double-binary RSC codes have a natural rate of 2/3. When higher coding rates are required, a simple regular or quasi-regular puncturing pattern is applied.

The decoding solution calling for the application of the *Maximum A Posteriori* (MAP) algorithm based on the dual code [8] can also be considered for large values of m, since it requires fewer edge computations than the classical MAP algorithm for high coding rates. However, for practical implementations, its application in the log-domain means the computation of transition metrics with a far greater number of terms and the computation of the max* (Jacobian) function requires great precision. Consequently, the relevance of this method for practical use is not so obvious when the rate of the mother code is not close to 1.0.

III. Block Turbo Coding with Rate-m / (m + 1) Constituent RSC Codes

The TC family proposed in this paper calls for the parallel concatenation of two identical rate-m / (m + 1) RSC encoders with m-bit word interleaving. Blocks of k bits, $k = mN$, are encoded twice by this bi-dimensional code, whose rate is m / (m + 2).

A. Circular RSC codes

Among the different techniques aiming at transforming a convolutional code into a block code, the best way is to allow any state of the encoder as the initial state and to encode the sequence so that the final state of the encoder is equal to the initial state. The code trellis can then be viewed as a circle, without any state discontinuity. This termination technique, called *tail-biting* [9][10] or circular, presents three advantages in comparison with the classical trellis termination technique using tail bits to drive the encoder to the all-zero state. Firstly, no extra bits have to be added and transmitted; thus there is no rate loss and the spectral efficiency of the transmission is not reduced. Next, when classical trellis termination is applied for TCs, a few codewords with input Hamming weight 1 may appear at the end of the

block (in both coding dimensions) and can be the cause of a marked decrease in the minimum Hamming distance of the composite code. With tail-biting RSC codes, only codewords with minimum input Hamming weight 2 are transmitted. In other words, tail-biting encoding avoids any side effects. Moreover, in a tail-biting or circular trellis, the past is also the future and *vice versa*. This means that a non-RTZ sequence produces effects on the whole set of redundant symbols stemming from the encoder, around the whole circle. Thanks to this very informative redundancy, there is very little probability that the decoder will fail to recover the correct sequence.

In practice, the circular encoding of a data block consists of a two-step process [9]: at the first step, the information sequence is encoded from state **0** and the final state is memorized. During this first step, the outputs bits are ignored. The second step is the actual encoding, whose initial state is a function of the final state previously memorized. The double encoding operation represents the main drawback of this method, but in most cases it can be performed at a frequency much higher than the data rate.

The iterative decoding of such codes involves repeated and continuous loops around the circular trellis. The number of loops performed is equal to the required number of iterations. The state probabilities or metrics, according to the chosen decoding algorithm, computed at the end of each turn are used as initial values for the next turn. With this method, the initial – or final – state depends on the encoded information block and is *a priori* unknown to the decoder at the beginning of the first iteration. If all the states are assumed to be equiprobable at the beginning of the decoding process, some side errors may be produced by the decoders at the beginning of the first iteration. These errors are removed at the subsequent iterations since final state probabilities or metrics computed at the end of the previous iteration are used as initial values.

B. Permutation

Among the numerous permutation models that have been suggested up to now, the apparently most promising ones in terms of minimum Hamming distances are based on regular permutation calling for circular shifting [11] or the co-prime [12] principle. After writing the data in a linear memory, with address i ($0 \le i \le N-1$), the information block is likened to a circle, both extremities of the block ($i = 0$ and $i = N - 1$) then being contiguous. The data are read out such that the jth datum read was written at the position i given by:

$$i = \Pi(j) = Pj + i_0 \mod N \tag{6}$$

where the skip value P is an integer, relatively prime with N, and i_0 is the starting index. This permutation does not require the block to be seen as rectangular, that is, N may be any integer.

In [13] and [14], two very similar modifications of (6) were proposed, which generalize the permutation principle adopted in the DVB-RCS/RCT or IEEE802.16a TCs. In the sequel, we will consider the *Almost Regular Permutation* (ARP) model detailed in [14], which changes relation (6) into:

$$i = \Pi(j) = Pj + Q(j) + i_0 \mod N \tag{7}$$

where $Q(j)$ is a small integer, whose value is taken in a limited set $\{0, Q_1, Q_2, ..., Q_{C-1}\}$, in a cyclic way. C, called the *cycle* of the permutation, must be a divider of N and has a typical value 4 or 8. For instance, if $C = 4$, the permutation law is defined by:

$$\text{if } j = 0 \mod 4,\ i = \Pi(j) = Pj + 0 + i_0 \mod N$$

$$\text{if } j = 1 \mod 4,\ i = \Pi(j) = Pj + Q_1 + i_0 \mod N$$

$$\text{if } j = 2 \mod 4,\ i = \Pi(j) = Pj + Q_2 + i_0 \mod N \tag{8}$$

$$\text{if } j = 3 \mod 4,\ i = \Pi(j) = Pj + Q_3 + i_0 \mod N$$

and N must be a multiple of 4, which is not a very restricting condition, with respect to flexibility.

In order to ensure the bijection property of Π, the Q values are not just any values. A straightforward way to satisfy the bijection condition is to choose all Qs as multiples of C.

The regular permutation law expressed by (6) is appropriate for error patterns which are simple RTZ sequences for both encoders, that is, RTZ sequences which are not decomposable as a sum of shorter RTZ sequences. A particular and important case of a simple RTZ sequence is the 2-symbol RTZ sequence, which may dominate in the asymptotic characteristics of a TC (see [15] for TCs with $m = 1$). A 2-symbol sequence is a sequence with two nonzero m-bit input symbols, which may contain more than one nonzero bit. Let us define the *total spatial distance* (or *total span*) $S(j_1,j_2)$ as the sum of the two spatial distances, before and after permutation according to (6), for a given pair of positions j_1 and j_2:

$$S(j_1, j_2) = \mathrm{f}(j_1, j_2) + \mathrm{f}(\Pi(j_1), \Pi(j_2)) \tag{9}$$

where

$$\mathrm{f}(u, v) = \min\{|u - v|, N - |u - v|\} \tag{10}$$

Finally, we denote $S_{\min}$ the minimum value of $S(j_1,j_2)$, for all possible pairs j_1 and j_2:

$$S_{\min} = \min_{j_1, j_2}\{S(j_1, j_2)\} \tag{11}$$

It was demonstrated in [16] that the maximum possible value for Smin, when using regular interleaving, is:

$$S = (S_{\min})_{\max} = \sqrt{2N} = \sqrt{\frac{2k}{m}} \tag{12}$$

If any 2-symbol RTZ sequence for one component encoder is transformed by Π or Π^{-1} into another 2-symbol RTZ sequence for the other encoder, the upper bound given by (12) is amply sufficient to guarantee a large weight for parity bits, and thus a large minimum binary Hamming distance. This is the same for any number of symbols, on the condition that both RTZ sequences before and after permutation, are simple RTZ sequences.

On the other hand, ARP aims at combating error patterns which are not simple RTZ sequences but are combinations of simple RTZ sequences for both encoders. Instilling some controlled disorder, through the *Q(j)* values in (7), tends to break most of the composite RTZ sequences. Meanwhile, because the *Q(j)* values are small integers, the good property of regular permutation for simple RTZ sequences is not lost, and a total span close to $\sqrt{2N}$ can be achieved. Reference [14] describes a procedure to obtain appropriate values for P and for the set of Q parameters.

The algorithmic permutation model described by (7) is simple to implement, does not require any ROM and the parameters can be changed on-the-fly for adaptive encoding and decoding. Moreover, as explained in [14], massive parallelism, allowing several processors to run at the same time without increasing the memory size, can be exploited.

In addition to the ARP principle and the advantages developed above, the rate-m / (m + 1) component code adds one more degree of freedom in the design of permutations: intra-symbol permutation, which enables some controlled disorder still to be added into the permutation without altering its global quasi-regularity. Intra-symbol permutation means modifying the contents of the m-bit symbols periodically, before the second encoding, in such a way that a large proportion of composite RTZ sequences for both codes can no longer subsist. Let us develop this idea in the simplest case of $m = 2$.

Fig. 2(a) depicts the minimal rectangular error pattern (weight $w = 4$), for a parallel concatenation of two identical binary RSC encoders, involving a regular permutation (linewise writing, columnwise reading). This error pattern is a combination of two weight-2 RTZ sequences in each dimension, leading to a composite RTZ pattern with distance 16, for coding rate 1/2. If the component encoder is replaced by a double-binary encoder, as illustrated in Fig. 2(b), RTZ sequences and error patterns involve couples of bits, instead of binary values. Fig. 2(b) gives two examples of rectangular error patterns, corresponding to the minimum distance of 18, still for coding rate 1/2 (*i.e.* no puncturing). Data couples are numbered from **0** to **3**, with the following notation: (0,0):**0**; (0,1):**1**; (1,0):**2**; (1,1):**3**. The periodicities of the double-binary RSC encoder, depicting all the combinations of pairs of input couples, different from **0**, that are RTZ sequences, are summarized in the diagram of Fig. 3. For instance, if the encoder, starting from state **0**, is fed up with successive couples **1** and **3**, it retrieves state **0**. The same behavior can be observed with sequences **201**, **2003, 30002**, **3000001** or **30000003** for example.

The change from binary to double-binary code, though leading to a slight improvement in the minimum distance (18 instead of 16), is not sufficient to ensure very good performance at low error rates. Let us suppose now that couples are inverted (**1** becomes **2** and *vice versa*) once every other time before second (vertical) encoding, as depicted in Fig 4. In this way, the error patterns displayed in Fig. 2(b) no longer remain error patterns. For instance, **20000002** is still an RTZ sequence for the second (vertical) encoder but **10000002** is no longer RTZ. Thus, many error patterns, especially short patterns, are eliminated thanks to the disorder introduced inside input symbols. The right-hand side of Fig 4 shows two examples of rectangle error patterns, that remain possible error patterns after the periodic inversion. The resulting minimal distances, 24 and 26, are large enough for the transmission of short data blocks [17]. For longer data blocks (a few thousand bits), combining this intra-symbol permutation with inter-symbol ARP, as described above, can lead to even larger minimum distances, at least with respect to the rectangular error patterns with low input weights we gave as examples.

C. Advantages of TCs with Rate-m / (m + 1) RSC Constituent Codes

Parallel concatenation of m-input binary RSC codes offers many advantages in comparison with classical (1-input) binary TCs, which have already been partly commented on in [18].

1) Better convergence of the iterative process

This point was first observed in [19] and commented on in [20]. The better convergence of the bi-dimensional iterative process is explained by a lower error density in each code dimension, which leads to a decrease in the correlation effect between the component decoders.

Let us consider again relation (12) which gives the maximum total span achievable when using regular or quasi-regular permutation. For a given coding rate R, the number of parity bits involved all along the total span, and used by either one decoder or the other, is:

$$n_{\text{parity}}(S) = \left(\frac{1-R}{R}\right)\frac{m}{2}S = \left(\frac{1-R}{R}\right)\sqrt{\frac{mk}{2}} \tag{13}$$

Thus, replacing a classical binary (m = 1) with a double-binary (m = 2) TC multiplies this number of parity bits by $\sqrt{2}$, though dividing the total span by the same value. Because the parity bits are not a matter of information exchange between the two decoders (they are just used locally), the more numerous they are, with respect to a given possible error pattern (here the weight-2 patterns), the less correlation effects between the component decoders.

Raising m beyond 2 still improves the turbo algorithm, regarding correlation, but the gains get smaller and smaller as m increases.

2) Larger minimum distances

As explained above, the number of parity bits involved in simple 2-symbol RTZ sequences for both encoders is increased when using rate-$m/(m + 1)$ component codes. The number of parity bits involved in any simple RTZ sequence, before and after permutation, is at least equal to $n_{\text{parity}}(S)$, regardless of the number of nonzero symbols in the sequence. The binary Hamming distances corresponding to all simple RTZ sequences are then high, and do not pose any problem with respect to the minimum Hamming distance of the composite code. This comes from error patterns made up of several (typically 2 or 3) short simple RTZ sequences on both dimensions of the TC. Different techniques can be used to break most of these patterns, one of them (ARP) having been presented in section *III.B*.

3) Less sensitivity to puncturing patterns

In order to obtain coding rates higher than $m/(m + 1)$, from the RSC encoder of Fig. 1, fewer redundant symbols have to be discarded, compared with an $m = 1$ binary encoder. Consequently, the correcting ability of the constituent code is less degraded. In order to illustrate this assertion, Fig. 5 compares the performance, in terms of Bit Error Rate (BER), of two 8-state RSC codes for m = 1 and m = 2, with the same generator polynomials (15, 13), in octal notation. The 2-input RSC code displays better performance than the 1-input code, for both simulated coding rates.

4) Higher throughput and reduced latency

The decoder of an m / (m + 1) convolutional code provides m bits at each decoding step. Thus, once the data block is received, and for a given processing clock, the decoding throughput is multiplied by m and the latency is divided by m, compared to the classical case (m = 1).

However, the critical path of the decoder being the ACS (Add-Compare-Select) unit, the decoder with $m > 1$ has a lower maximum clock frequency than with m = 1. For instance, for m = 2, the Compare-Select operation has to be done on 4 metrics instead of 2, thus with an increased propagation delay. The use of specialized look-ahead operators and/or the introduction of parallelism, in particular the multi-streaming method [21] makes it possible to increase significantly the maximum frequency of the decoder, and even to reach that of the decoder with m = 1.

5) Robustness of the decoder

Fig. 6 represents the simulated performance, in Frame Error Rate (FER) as a function of E_b/N_0, of four coding/decoding schemes dealing with blocks of 1504 information bits and

coding rate 4/5: binary and double-binary 8-state TCs mentioned in subsection III.C.2, both with the full MAP decoding algorithm [22][23] and with the simplified Max-Log-MAP version [24]. In the latter case, the extrinsic information is less reliable, especially at the beginning of the iterative process. To compensate for this, a scaling factor, lower than 1.0, is applied to extrinsic information [25]. The best observed performance was obtained when a scaling coefficient of 0.7 for all the iterations, except for the last one, was applied.

In Fig. 6, both codes have ARP internal permutation with optimized span. We can observe that the double-binary TC performs better, at both low and high E_b/N_0, and the steeper slope for the double-binary TC indicates a larger minimum binary Hamming distance. These characteristics were justified by points 1) and 2) of this section.

What is also noteworthy is the very slight difference between the decoding performance of the double-binary TC, when using the MAP or the Max-Log-MAP algorithms. This property of non-binary turbo decoding actually makes unnecessary the full MAP decoder (or the Max-Log-MAP decoder with Jacobian logarithm correction [24]), which requires more operations than the Max-Log-MAP decoder. Also, the latter does not need the knowledge of the noise variance on Gaussian channels, which is a nice advantage. The rigorous explanation for this quasi-equivalence of the MAP and the Max-Log-MAP algorithms, when decoding m-input TCs, has still to be found.

IV. Performance of Double-Binary TCs

This section describes two examples of double-binary TCs, with memory 3 and 4, whose reasonable decoding complexity allows them to be implemented in actual hardware devices for practical applications. Simulation results for transmissions over an AWGN channel with QPSK modulation are provided.

A. Eight-State Double-Binary TC

The parameters of the component codes are:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \mathbf{R}_1 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \quad \mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{14}$$

The diagram of the encoder is described in Fig. 7.Redundancy vector $\mathbf{R}_2$ is only used for coding rates less than 1/2. For coding rates higher than 1/2, puncturing is performed on redundancy bits in a regular periodical way, following patterns that are described in [5]. These patterns are identical for both constituent encoders.

The permutation function $i = \Pi(j)$ is performed on two levels, as explained in section III.B:

For $j = 0, \cdots, N-1$

- Level 1: Inversion of $d_{j,1}$ and $d_{j,2}$ in the data couple, if $j \bmod 2 = 0$.
- Level 2: this permutation level is described by a particular form of (8)

  $i = (P \times j + Q(j) + 1) \bmod N$, with

$$Q(j) = 0 \qquad \text{if } j \bmod 4 = 0$$

$$Q(j) = N/2 + P_1 \quad \text{if } j \bmod 4 = 1 \tag{15}$$

$$Q(j) = P_2 \qquad \text{if } j \bmod 4 = 2$$

$$Q(j) = N/2 + P_3 \quad \text{if } j \bmod 4 = 3$$

Value $i_0 = 1$ is added to the incremental relation in order to comply with the odd-even rule [26]. The disorder is instilled in the permutation function, according to the ARP principle, in two ways:

- A shift by $N/2$ is added for odd values of j. This is done because the lowest sub-period of the code generator is 1 (see Fig. 3). The role of this additional incrementation is thus to spread to the full the possible errors associated with the shortest error patterns.
- P_1, P_2 and P_3 act as local additional pseudo-random fluctuations.

Notice that the permutation equations and parameters do not depend on the coding rate considered. The parameters can be optimized to provide good behavior, on average, at low error rates for all coding rates, but seeking parameters for a particular coding rate could lead to slightly better performance.

B. Sixteen-State Double-Binary TC

The parameters of the best component code we have found are:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} \tag{16}$$

The diagram of the encoder is described in Fig. 8.

Puncturing is performed on redundancy in a periodical way with identical patterns for both constituent encoders. It is usually regular, except when the puncturing period is a divisor of the LFSR period. For example, for coding rate 3/4, the puncturing period is chosen equal to 6, with puncturing pattern [101000].

For this code, the permutation parameters have been carefully chosen following the procedure described in [14] in order to guarantee a large minimum Hamming distance, even for high rates. The level 1 permutation is identical to the intra-permutation of the 8-state code. The level 2 – inter-symbol – permutation is given by:

For $j = 0, \cdots, N-1$

$$i = (P \times j + Q(j) + 3) \bmod N \text{, with}$$

$$Q(j) = 0 \qquad \text{if } j \bmod 4 = 0$$

$$Q(j) = Q_1 \qquad \text{if } j \bmod 4 = 1 \tag{17}$$

$$Q(j) = 4Q_0 + Q_2 \quad \text{if } j \bmod 4 = 2$$

$$Q(j) = 4Q_0 + Q_3 \quad \text{if } j \bmod 4 = 3$$

The spirit in which this permutation was designed is the same as that already explained for the 8-state TC. The only difference is that the lowest sub-period of the 16-state generator is 2, instead of 1. That is why the additional shift (by $4Q_0$) is applied consecutively twice every four values of j.

Table I compares the minimum binary Hamming distances of the proposed 8-state and 16-state TCs, for 188-byte data blocks and 4 different coding rates. The distance values were estimated with the so-called "Error Impulse Method", a fast computational method described in [27], that provides distance values with a precision of approximately 5 %. We can observe a significant increase in the minimum distance when using 16-state component codes: the gain varies from 30% up to more than 50% depending on the case considered. With this code, we were also able to define permutation parameters leading to minimum distances as large as 33 for $R = 1/2$, 22 for $R = 2/3$ and 16 for $R = 3/4$ for (10×188)-byte blocks.

From an implemention point of view, the complexity of the corresponding decoder is about twice the complexity of the 8-state decoder.

C. Simulation results

We have simulated and compared these two codes for two block sizes and three coding rates for transmissions over an AWGN channel with QPSK modulation. The simulation results take actual implementation constraints into account. In particular, the decoder inputs are quantized for hardware complexity considerations. In our experience, performance degradation due to input quantization is not significant beyond 5 bits. The observed loss is less than 0.15 dB for 4-bit quantization and about 0.4 dB for 3-bit quantization. When quantization is applied, clipping extrinsic information at a threshold around twice the maximum range of the input samples does not degrade the performance, while limiting the amount of required memory.

Solid line curves in Figs. 9 and 10 show the Frame Error Rate (FER) as a function of E_b / N_0 for the transmission of ATM (53-byte) and MPEG (188-byte) packets for 3 different values of coding rate of the 8-state double-binary TC. The component decoders use the Max-Log-MAP algorithm, with input samples quantized on 4 bits. Eight iterations were simulated and at least 100 erroneous frames were considered for each point indicated, except for the lowest points where approximately 30 erroneous frames were simulated.

We can observe good average performance for this code whose decoding complexity is very reasonable: for a hardware implementation, less than 20,000 logical gates are necessary to implement one iteration of the decoding process when decoding is performed at the system clock frequency, plus the memory required for extrinsic information and input data. Its performance improves predictably with block size and coding rate in relation to the theoretical limit. The reported limit points take the block size and the target FER into account. They are derived from Shannon's sphere packing bound, as described in [28]. At FER = 10^{-4}, the simulated curves lie within 0.6 to 0.8 dB from the limit, regardless of block size and coding rate. To improve the performance of this code family at FER below 10^{-5}, the more powerful 16-state component code has to be selected so as to increase the overall minimum Hamming distance of the composite code.

Dotted line curves in Figs. 9 and 10 show the 16-state TC performance for the same simulation conditions as for the 8-state code. Similarly to this code, the permutation parameters are related to the block size, not to the coding rate.

We can observe that the selected code does not lead to a convergence threshold shift of the iterative decoding process in comparison with the previous 8-state code: for FERs above 10^{-4}, 16-state and 8-state codes behave similarly. For lower error rates, thanks to the increase in

distance, there is no noticeable floor effect for the simulated SNR ranges, that is, down to a FER of 10^{-7}. Again, performance improves predictably with block size and coding rate in relation to the theoretical limit. At FER = 10^{-6}, the simulated curves lie within 0.7 to 1.0 dB from the limits, regardless of block size and coding rate, even with the simplified Max-Log-MAP algorithm.

V. CONCLUSION

Searching for perfect channel coding presents two challenges: encoding in such a way that large minimum distances can be reached and achieving decoding as close to the theoretical limit as possible. In this paper, we have explained why *m*-input binary TCs combined with a two-level permutation represent a better answer to these challenges than classical one-input binary TCs.

In practice, with $m = 2$, we have been able to design coding schemes with moderate decoding complexity and whose performance approaches the theoretical limit by less than 1 dB at FER = 10^{-6}. The 8-state TC with $m = 2$ has already found practical applications through several international standards.

Furthermore, the parallel concatenation of RSC circular codes leads to flexible composite codes, easily adaptable to a large range of data block sizes and coding rates. Consequently, as *m*-input binary codes are well suited for association with high order modulations, in particular *M*-QAM and *M*-PSK modulations, TCs based on these constituent codes appear to be good candidates for most future digital communication systems based on block transmission.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes", *IEEE Trans. Commun.*, vol. 44, n° 10, pp. 1261-1271, Oct.1996.

[2] P. Robertson and T. Wörz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 206-218, Feb. 1998.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis coded modulation," in *Proc. IEEE Int. Conf. Communications.*, vol. 2, Dallas, TX, June 1996, pp. 974-978.

[4] C. Fragouli and R. D. Wesel, "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation," *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 425-435, March. 2001.

[5] DVB, "Interaction channel for satellite distribution systems," *ETSI EN 301 790,* V1.2.2, pp.21-24, Dec. 2000.

[6] DVB, "Interaction channel for digital terrestrial television," *ETSI EN 301 958,* V1.1.1, pp.28-30, Feb. 2002.

[7] IEEE Std 802.16a, "IEEE standard for local and metropolitan area networks," 2003, available at http://standards.ieee.org/getieee802/download/802.16a-2003.pdf.

[8] S. Riedel, "Symbol-by-symbol MAP decoding algorithm for high-rate convolutional codes that use reciprocal dual codes," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 175-185, Feb. 1998.

[9] C. Weiss, C. Bettstetter, S. Riedel and D. J. Costello, "Turbo decoding with tail-biting trellises," *in Proc. IEEE Int'l Symposium on Signals, Systems, and Electronics*, Pisa, Italy, October 1998, pp. 343-348.

[10] R. Johannesson, K. Sh. Zigangirov, *Fundamentals of convolutional coding*, Chap. 4, New York, IEEE Press, Series on digital & mobile communication, 1999.

[11] S. Dolinar and D. Divsalar, "Weight distribution of turbo codes using random and nonrandom permutations," *TDA Progress Report 42-122*, JPL, NASA, Aug. 1995.

[12] C. Heegard and S. B. Wicker, *Turbo Coding*, Chap. 3, Kluwer Academic Publishers, 1999.
[13] S. Crozier, J. Lodge, P. Guinand and A. Hunt, "Performance of turbo codes with relatively prime and golden interleaving strategies," in *Proc. 6th Int'l Mobile Satellite Conf.*, Ottawa, Canada, June 1999, pp. 268-275.
[14] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model," in *Proc. IEEE Int. Conf. Communications*, Paris, France, June 2004.
[15] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591 – 600, May 1996.
[16] E. Boutillon and D. Gnaedig, "Maximum Spread of D-dimensional Multiple Turbo Codes," *submitted to IEEE Trans. Commun.*, temporarily available at http://lester.univ-ubs.fr:8080/~boutillon/articles/IEEE_COM_spread.pdf.
[17] C. Berrou, E. A. Maury and H. Gonzalez, "Which minimum Hamming distance do we really need?," in *Proc. 3rd Symposium on Turbo Codes*, Brest, France, Sept. 2003, pp. 141-148.
[18] C. Berrou, M. Jézéquel, C. Douillard and S. Kerouédan, "The advantages of non-binary turbo codes," *Proc. Information Theory Workshop*, Cairns, Australia, Sept. 2001, pp. 61-63.
[19] C. Berrou, "Some clinical aspects of turbo codes," *Proc. Int. Symp. on Turbo Codes & Related Topics*, pp. 26-31, Brest, France, Sept. 1997.
[20] C. Berrou and M. Jézéquel, "Non binary convolutional codes for turbo coding," *Electronics Letters*, Vol. 35, N° 1, pp. 39-40, Jan. 1999.
[21] H. Lin and D. G. Messerschmitt, "Algorithms and architectures for concurrent Viterbi decoding," in *Proc. IEEE Int. Conf. Communications*, Boston, June 1989, pp. 836-840.
[22] 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, March 1974.
[23] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 22–24, Jan. 1997.
[24] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. Telecommun.*, vol. 8, pp. 119-125, Mar./Apr. 1997.
[25] J. Vogt and A. Finger, "Improving the max-log-MAP turbo decoder," *Electron. Lett.*, vol. 36, no. 23, pp. 1937-1939, Nov. 2000.
[26] A. S. Barbulescu, *Iterative Decoding of Turbo Codes and Other Concatenated Codes*, Ph.D. thesis, University of South Australia, Feb. 1996
[27] C. Berrou, S. Vaton, M. Jézéquel and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method," *Proc. Globecom'02*, Taipei, Taiwan, Nov. 2002.
[28] S. Dolinar, D. Divsalar and F. Pollara, "Code performance as a function of block size," *TMO Progress Report 42-133*, JPL, NASA, May 1998.

FIGURES AND TABLE

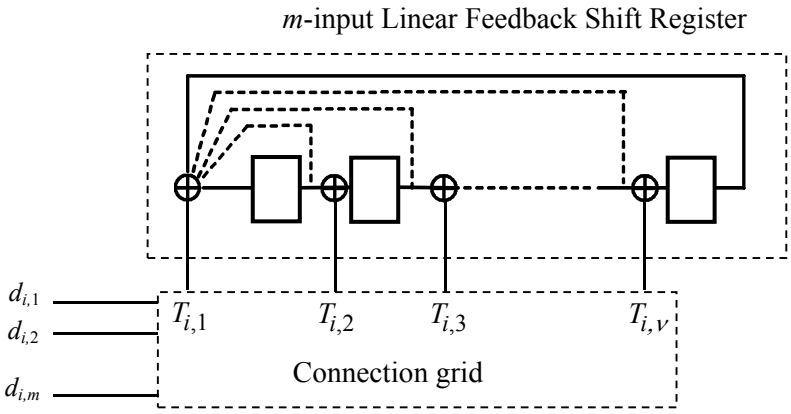


Fig. 1. General structure of a rate-m / (m + 1) RSC encoder with code memory ν.

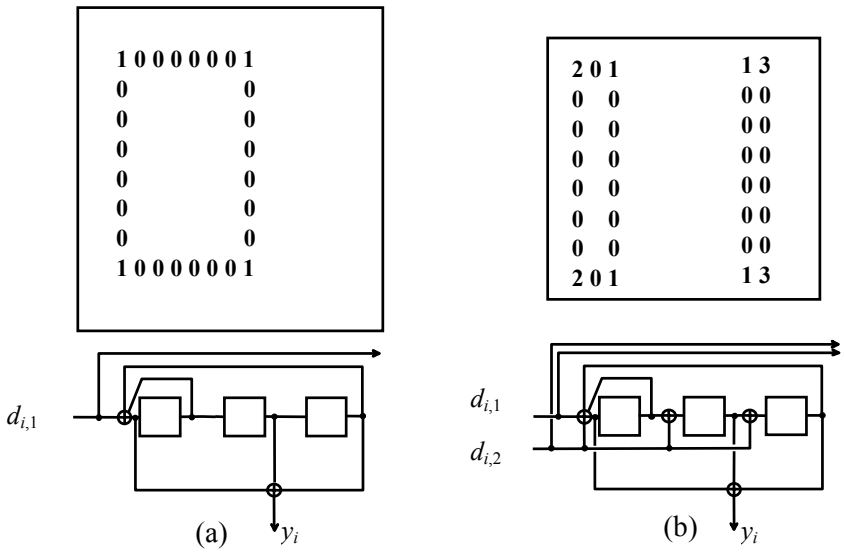


Fig. 2. Possible rectangular error patterns for (a) binary and (b) double-binary TCs with regular permutations.

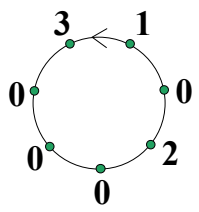


Fig. 3. Periodicities of the double-binary encoder of Fig. 3(b). Input couples (0,0), (0,1), (1,0) and (1,1) are denoted **0**, **1**, **2** and **3**, respectively.

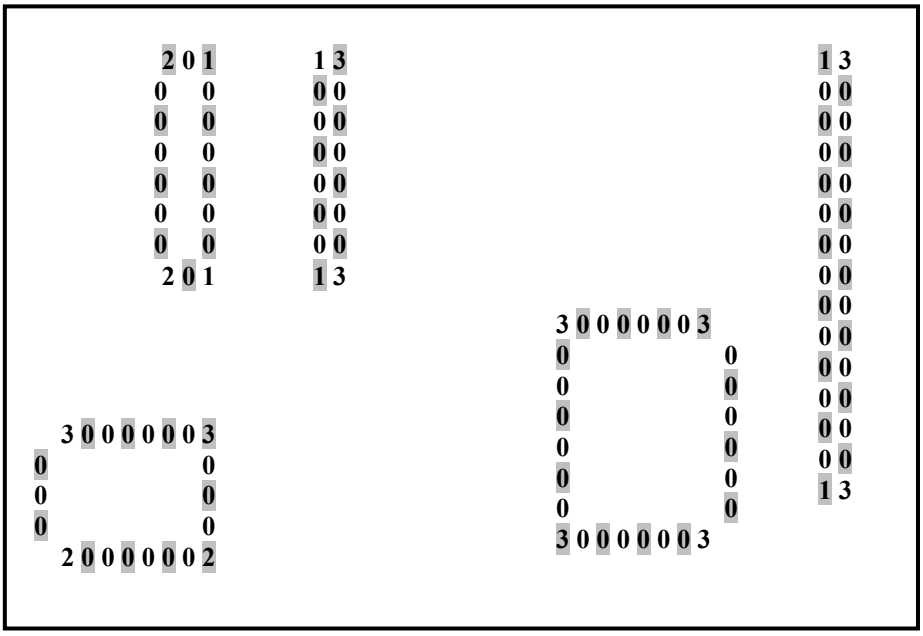


Fig. 4. Couples in gray spaces are inverted before second (vertical) encoding. **1** becomes **2**, **2** becomes **1**; **0** and **3** remain unaltered. The three patterns on the left-hand side are no longer error patterns. Those on the right-hand side remain possible error patterns, with distances 24 and 26 for coding rate 1/2.

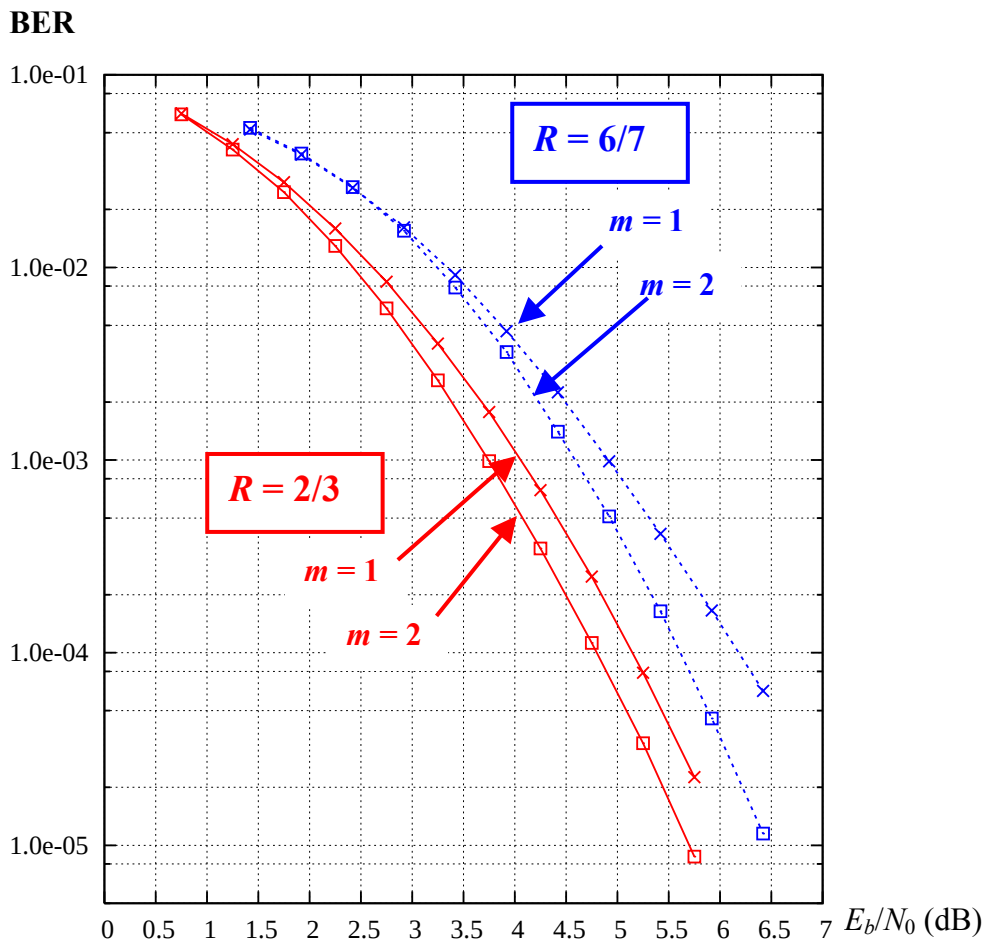


Fig. 5. Performance in Bit Error Rate (BER) of 8-state RSC codes with $m = 1$ and $m = 2$. Encoder polynomials: 15 (feedback) and 13 (redundancy) in octal form (DVB-RCS constituent encoder for $m = 2$). Coding rates are 2/3 and 6/7. BPSK/QPSK modulation, AWGN channel and MAP decoding. No quantization and regular puncturing.

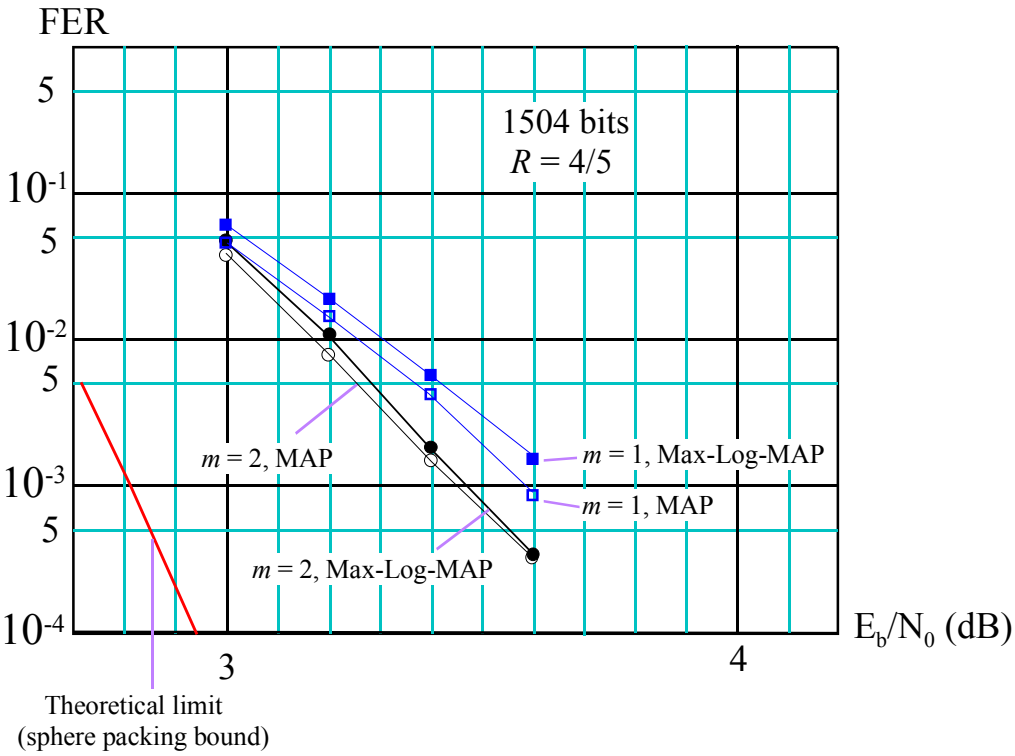


Fig. 6. Comparison of performance in Frame Error Rate (FER) of both 8-state RSC codes of Fig. 5, with $k = 1504$, $R = 4/5$, for MAP and Max-Log-MAP decoding. AWGN channel, QPSK modulation, 8 iterations. Scaling factor for Max-Log-MAP decoding: 0.7 for iterations 1 to 7, 1.0 for iteration 8. No quantization.

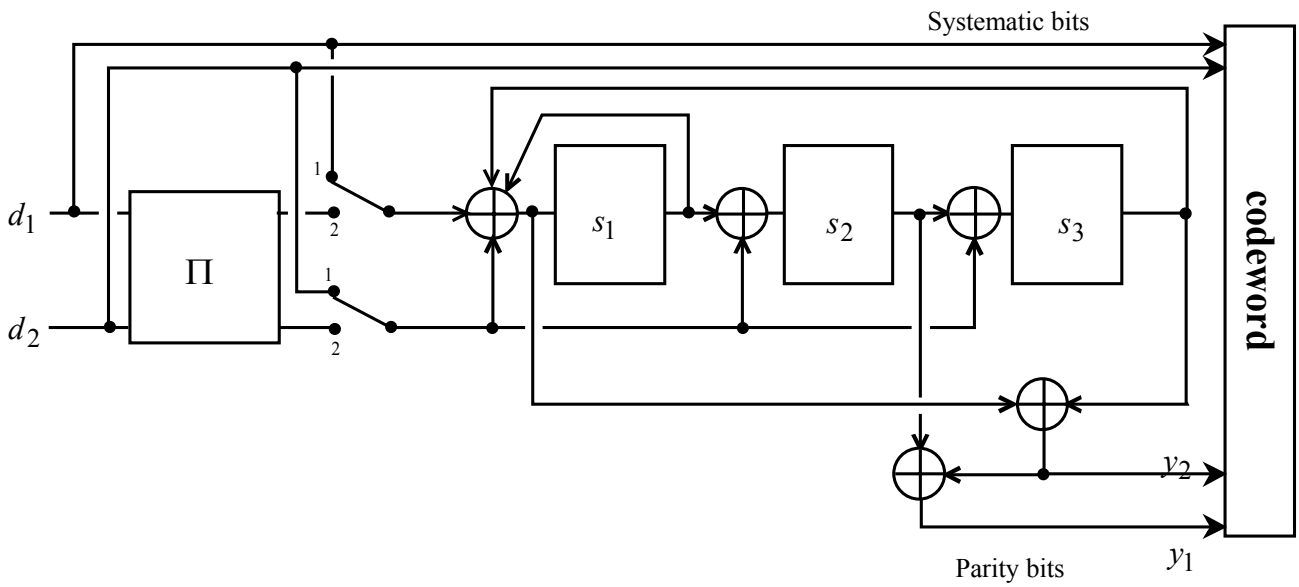


Fig. 7. Structure of the 8-state encoder. Redundancy y_2 is only used for coding rates less than 1/2.

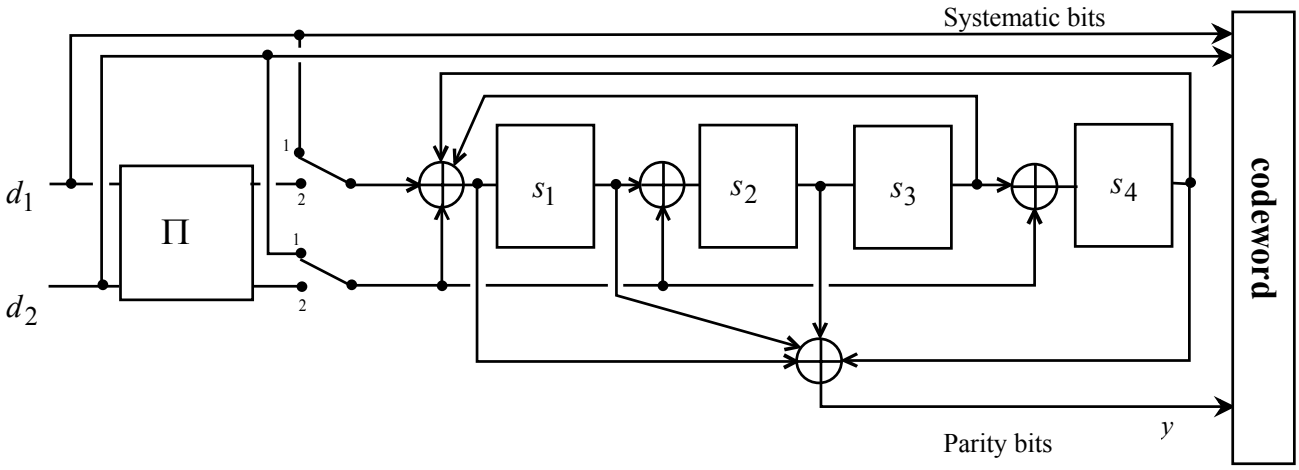


Fig. 8. Structure of the proposed 16-state double-binary turbo encoder.

Coding rate	1/2	2/3	3/4	5/6
8-state double-binary turbo code ($P_0 = 19, P_1 = 376, P_2 = 224, P_3 = 600$)	19	13	10	7
16-state double-binary turbo code ($Q_0 = 35, Q_1 = 6, Q_2 = 4, Q_3 = 10$)	29	19	13	9

Table I. Estimated values of minimum binary Hamming distances $d_{\min}$ of the proposed 8-state and 16-state double-binary TCs for 188-byte data blocks. The distances were estimated with the error impulse method [27].

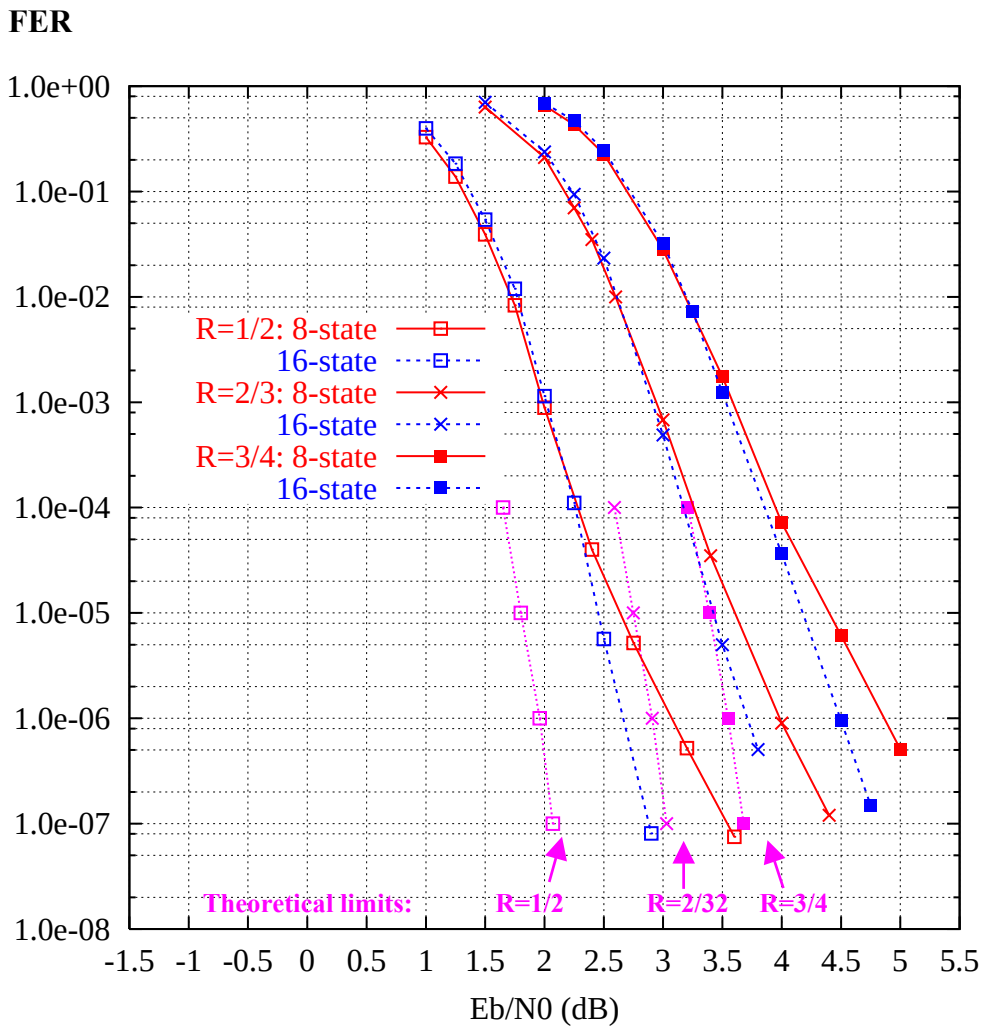


Fig. 9. Performance in Frame Error Rate (FER) of 8-state and 16-state double-binary TCs for ATM (53-byte) blocks and rates 1/2, 2/3 and 3/4. QPSK modulation and AWGN channel. Max-Log-MAP decoding with 4-bit input samples and 8 iterations. The theoretical limits on FER as a function of block size are also provided.

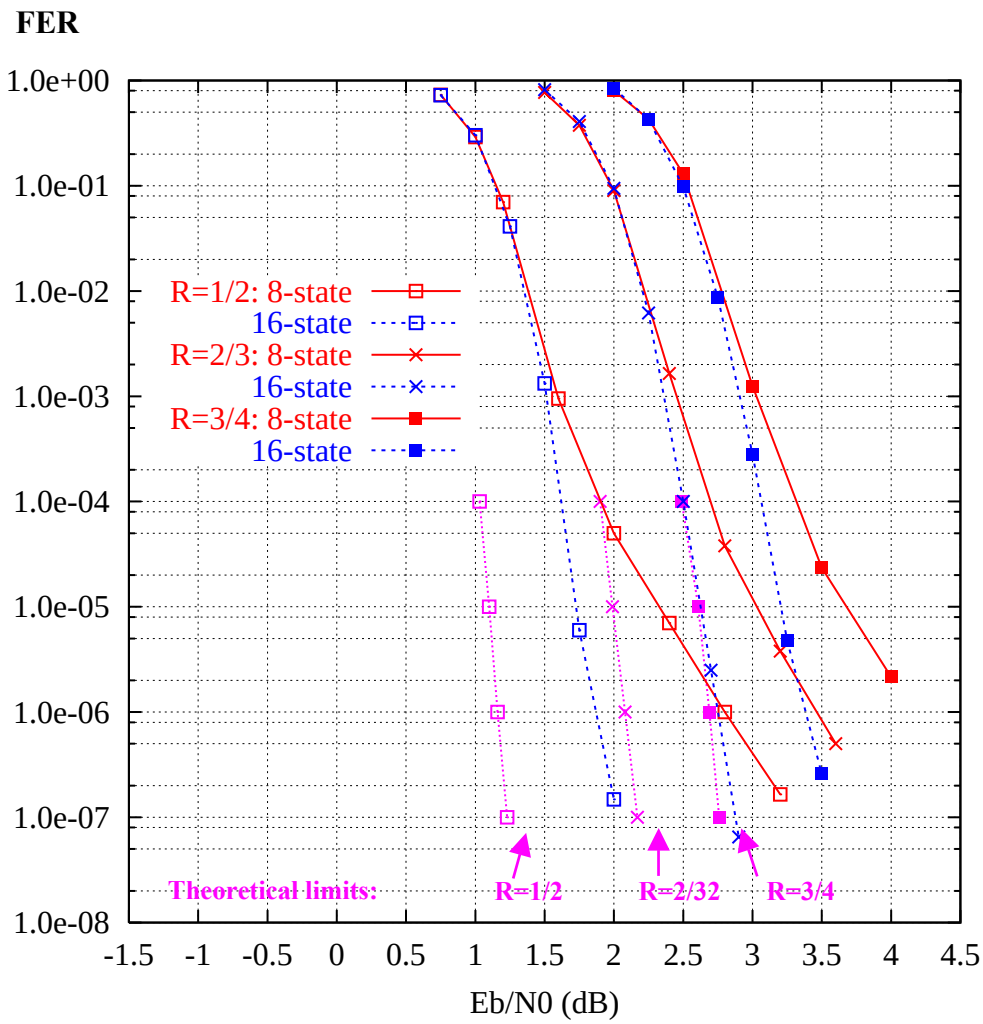


Fig. 10. Performance in Frame Error Rate (FER) of 8-state and 16-state double-binary TCs for MPEG (188-byte) blocks and rates 1/2, 2/3 and 3/4. QPSK modulation and AWGN channel. Max-Log-MAP decoding with 4-bit input samples and 8 iterations. The theoretical limits on FER as a function of block size are also provided.

Annexe 7

Article “The turbo code standard for DVB-RCS”, *2nd International Symposium on Turbo Codes & Related Topics*, 2000.

Cet article décrit les caractéristiques techniques du turbocode adopté par le standard DVB pour la voie de retour par satellite, DVB-RCS. Des résultats de mesures de taux d’erreurs obtenus sur une réalisation matérielle sur FPGA de ce turbo-décodeur sont présentés.

The Turbo Code Standard for DVB-RCS

C. Douillard*, M. Jézéquel*, C. Berrou*,
N. Brengarth**, J. Tousch** and N. Pham***
*ENST Bretagne, BP 832, 29285 Brest Cedex, France
Catherine.Douillard@enst-bretagne.fr
**TurboConcept, 1 av. du technopôle, 29280 Plouzané, France
Nathalie.Brengarth@enst-bretagne.fr
***Eutelsat, 70 rue Balard, 75015 Paris, France
npham@eutelsat.fr

Abstract: *Convolutional turbo codes are very flexible codes, easily adaptable to a large range of data block sizes and coding rates. This is the main reason for their being adopted in the DVB standard for Return Channel via Satellite (DVB-RCS). The paper presents the turbo coding/decoding scheme specified in this standard, for twelve block sizes and seven coding rates. Simulation results show the performance of the coding scheme chosen, in particular for the transmission of ATM cells and MPEG transport stream packets. The company TurboConcept has developed a hardware prototype for the double purpose of very low error rate performance measurement and demonstrates the hardware feasibility of this new turbo code. Some hardware measurement results are also discussed.*
Keywords: turbo codes, circular RSC codes, double-binary codes, two-level interleaver.

1. INTRODUCTION : DVB-RCS

1.1. System considerations

In a race involving ADSL and Cable modem to provide "broadband access", the DVB Committee recently approved a standard - known as DVB-RCS, for Return Channel via Satellite, or EN 301 790 in ETSI - to provide two-way, full-IP, asymmetric communications via satellite. One advantage that satellite offers is that the service can be deployed quickly, all over a large area, once a single "hub" infrastructure is in place. Another advantage is that service quality and the cost per susbscriber is independent of the distance between the terminal and the access point. This places satellite in a favourable position and especially as a necessary complement in countries where ADSL and Cable modem cannot economically cover more than 75% of the population.

This standard specifies an air interface allowing a large number of small terminals to send "return" signals to a central gateway, also called a "hub" and at the same time receive IP data from that hub on the "forward" link in the usual DVB/MPEG2 broadcast format, thus leveraging the ubiquity of both DVB and IP technologies, while avoiding a return connection via terrestrial means (such as a dial-up telephone line).

The return channel speed (terminals-to-hub) can range from 144 Kbps to 2 Mbps and the satellite resource on this return link is shared among the terminals transmitting small packets and using MF-TDMA (Multi-Frequency TDMA)/DAMA (Demand-Assigned Multiple Access) techniques. Since this access is packet-based and on-demand, once the session is established with the hub, the link is permanently open, thus providing an "always-on" IP connection, with efficient use of satellite resources (provided the duty-cycle of the terminals is low on the return link).

The new standard includes such advanced features as a software-radio system (the terminal acquires the characteristics of the transmission parameters it has to use from signals broadcast by the hub), network synchronisation in the digital domain (all terminals are synchronised to the same precise clock transmitted digitally by the hub), sophisticated bandwidth-on-demand protocols which can mix constant-rate, dynamic-rate, volume-based and best-effort bandwidth allocation and advanced forward-error-correction turbo coding (as will be used in the new On-Board satellite processing system, Skyplex [1]). This paper presents the rationale behind the selection of this FEC and the performance achieved on a hardware implementation of the decoder.

1.2. Coding requirements

Since DVB-RCS applications involve the transmission of data using various block sizes and coding rates, the coding scheme has to be very flexible, with better performance than the classical concatenation of a convolutional code and a Reed-Solomon code. Finally, it has to be able to process data so as to allow the transmission of data bit rates up to 2Mbps, as indicated in the introduction.

Convolutional turbo codes seemed to be a good candidate :

- A simple puncturing device is sufficient to adapt coding rate,
- The use of double-binary circular recursive systematic convolutional (CRSC) component codes makes turbo codes very efficient for block coding [2][3],
- The different permutations (interleavers) may be achieved using generic equations with only a restricted number of parameters.

– The same decoding hardware can be used to manage every block size / coding rate combination.

2. DETAILED FEATURES OF THE CODING SCHEME PROPOSED

Small component codes have been chosen for two main reasons. On the one hand, efficient turbo coding requires component codes with small minimum distances (i.e. with small constraint lengths for convolutional codes) in order to ensure convergence at very low signal to noise ratios and to minimize the correlation effects [3]. On the other hand, the material complexity of the decoder grows exponentially with code memory, so a hardware implementation on a single integrated circuit is only conceivable for reasonable constraint lengths. The solution chosen uses memory $\nu = 3$ component codes. This complexity/performance compromise allows the implementation of the decoder on a single FPGA (Field Programmable Gate Array) circuit. The same code size was also chosen for the UMTS standard.

Furthermore, the code proposed calls for two recent techniques in turbo coding:

– Parallel concatenation of circular recursive systematic convolutional (CRSC) codes [4] makes convolutional turbo codes efficient for block coding,
– Double-binary elementary codes provide better error-correcting performance than binary codes for equivalent implementation complexity [2].

2.1. Circular recursive systematic (CRSC) convolutional codes

Adopting circular coding avoids the degradation of the spectral efficiency of the transmission when forcing the value of the encoder state at the end of the encoding stage by the addition of tail bits.

Circular coding is an adaptation of the so – called "tail-biting" technique to recursive convolutional codes. It ensures that, at the end of the encoding operation, the encoder retrieves the initial state, so that data encoding may be represented by a circular trellis. The existence of such a state, called *circulation state* $\mathbf{S}_c$, is ensured when the size of the encoded data block, *N*, is not a multiple of the period of the encoding recursive generator. The value of the circulation state depends on the contents of the sequence to encode and determining $\mathbf{S}_c$ requires a pre-encoding operation: first, the encoder is initialised in the "all zero" state. The data sequence is encoded once, leading to a final state $\mathbf{S}_N^0$. $\mathbf{S}_c$ value is then calculated from expression $\mathbf{S}_c = \left\langle \mathbf{I} + \mathbf{G}^N \right\rangle^{-1} \mathbf{S}_N^0$. In practice, the relation between $\mathbf{S}_c$ and $\mathbf{S}_N^0$ is provided by a small combinational operator with $\nu = 3$ input and output bits. Finally, to perform a complete encoding operation of the data sequence, two circulation states have to be determined, one for each component encoder, and the sequence has to be encoded four times instead of twice. This is not a real problem, as the encoding operation can be performed at a frequency much higher than the data rate.

From the decoder point of view, a simplified version of the APP algorithm [5] was implemented. Decoding a sequence then consists of going around the circular trellis anticlockwise for the backward process and clockwise for the forward process, during which the data are decoded and extrinsic information is generated. For both processes, probabilities computed at the end of a turn are used as initial values for the next turn. The number of turns performed around the circular trellises is equal to the number of iterations required by the iterative process. In practice, the iterative process is preceded by a "prologue" decoding step, performed on a part of the circle for a few ν. This is intended to guide the process towards an initial state which is a good estimation of the circulation state.

2.2. Double-binary codes

Using double-binary codes as component codes represents a simple means to reduce the correlation effects, as explained in [2]. The substitution of binary codes by double-binary codes has a direct incidence on the erroneous paths in the trellises, which leads to a lowered path error density and reduces correlation effects in the decoding process. This leads to better performance, even in comparison with 16-state binary turbo codes.

Furthermore, using double-binary codes leads to extra advantages:

– The influence of puncturing is less significant than with binary codes (the natural rate of the double-binary turbo code being 1/2 instead of 1/3 for a binary turbo code).
– It has also been observed that the performance degradation due to the application of a simplified version of the APP algorithm is less significant in the case of double-binary codes (less than 0.1 dB) than in the case of binary codes (0.3 to 0.4 dB).
– From a material implementation point of view, the bit rate at the decoder output is twice that of a binary decoder processing the same number of iterations, with the same circuit clock frequency and with an equivalent complexity per decoded bit (the material complexity of the decoder, for a given memory length, is about twice the complexity of a binary decoder, but two bits are decoded at the same time). Moreover, given the data block size, the latency of the decoder is divided by 2 compared with the binary case because the size of the permutation matrix is halved.

– The use of double-binary codes also makes the introduction of two-level permutations possible, as explained in section 2.3.

2.3. Interleaving with local disorder

The performance of parallel concatenation of convolutional codes (PCCC) at low error rates, is essentially governed by the permutation that links the component codes. The simplest way to achieve interleaving in a block is to adopt uniform or regular interleaving: data are written linewise and read columnwise in a rectangular matrix. This kind of permutation behaves very well towards error patterns with weight 2 or 3, but is very sensitive to square or rectangular error patterns, as explained in [6]. Classically, in order to increase distances given by rectangular error patterns, non-uniformity is introduced in the permutation relations. Many proposals have been made in this direction, especially for the UMTS application. The CCSDS turbo code standard may also be cited as an example of non-uniform permutation. However the disorder that is introduced with non-uniformity can affect the scattering properties concerning weight 2 or 3 error patterns.

With double-binary codes, non-uniformity can be introduced without any repercussion on the good scattering properties of regular interleaving [3]. The principle involves introducing local disorder into the data couples, for example (A,B) becoming (B,A) – or $(B,A+B)$, or others – periodically before the second encoding. This helps to avoid many error patterns that would appear without applying it. Therefore, this appears to be a significant gain in the search for large minimum distances.

2.4. Summary

This section sums up the technical features of the turbo code adopted for DVB-RCS standard.

Encoder structure: the encoder consists of a parallel concatenation of two identical component codes with generators (in octal notation) 15 (recursion), 13 (redundancy Y_1), 11 (redundancy Y_2) as depicted in figure 1. Redundancy Y_2 is only used for coding rates less than 1/2, i.e. 1/3 and 2/5.

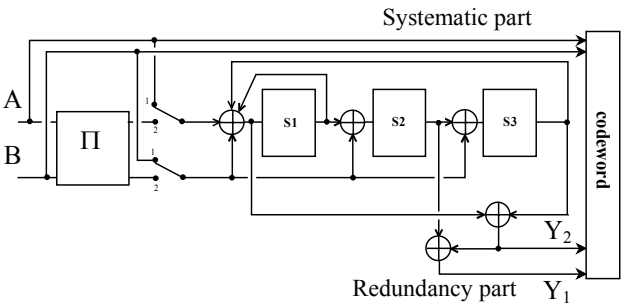


Figure 1: DVB-RCS encoder

Block sizes: 12, 16, 53, 55, 57, 106, 108, 110, 188, 212, 214 and 216 bytes.

Coding rates: 1/3, 2/5, 1/2, 2/3, 3/4, 4/5, and 6/7.

Permutation equations $i = \Pi(j)$:

Let N be the number of data couples in each block at the encoder input (each block contains $2N$ data bits)

For $j = 0, \cdots, N-1$

- 1st level: Inversion of A_j and B_j in the data couple, if $j \bmod 2 = 0$,
- 2nd level: $i = (P_0 \times j + P) \bmod N$, with

$P = 0$ if $j \bmod 4 = 0$

$P = N/2 + P_1$ if $j \bmod 4 = 1$

$P = P_2$ if $j \bmod 4 = 2$

$P = N/2 + P_3$ if $j \bmod 4 = 3$

where the values of parameters P_0, P_1, P_2, and P_3 depend on N (table 1).

Block size (bytes)	P_0	P_1	P_2	P_3
12	11	24	0	24
16	7	34	32	2
53	13	106	108	2
55	23	112	4	116
57	17	116	72	188
106	11	6	8	2
108	13	0	4	8
110	13	10	4	2
188	19	376	224	600
212	19	2	16	6
214	19	428	224	652
216	19	2	16	6

Table 1: Values of parameters P_0, P_1, P_2, and P_3.

2.5. Simulation results

Table 2 gives some examples of the DVB-RCS turbo code performance observed over a Gaussian channel at Frame Error Rate (FER) $=10^{-4}$, compared to the theoretical limits [7]. The component decoding algorithm applied is the Max-Log-APP, with samples quantized on 4 bits. Good convergence, close to the theoretical limits – from 1.0 dB to 1.5 dB, depending on the coding rate – can be observed, thanks to the double-binary component code.

Block size / Coding rate	ATM 53 bytes	MPEG 188 bytes
1/2	2.3 (1.3)	1.8 (0.8)
2/3	3.3 (2.2)	2.6 (1.7)
3/4	3.9 (2.6)	3.2 (2.1)
4/5	4.6 (3.1)	3.8 (2.6)
6/7	5.2 (3.8)	4.4 (3.3)

Table 2: E_b / N_0 (dB) @ FER $=10^{-4}$, 8 iterations, 4-bit input quantization, simulation over Gaussian channel, (theoretical limits in brackets).

3. IMPLEMENTATION

3.1. Description

TurboConcept has developed a turbo decoder that has been used to measure low error rate performances and also to demonstrate to system builders that these efficient turbo codes are affordable with reasonable complexity.

The main features of this implementation (TurboConcept TC1000 product [8]) are :

- Single-chip FPGA solution, e.g. Altera APEX 200 or Xilinx Virtex 400. This leads to fast and easy system integration.
- User bit rate up to 4 Mbit/s with 6 iterations.
- Every block size / coding rate combination is supported by the same hardware.

Additionally, the implementation takes advantage of the high flexibility of this code to allow the user to configure "on the fly" the block size and coding rate, with no guard time needed for interleaver pre-calculation, for example.

A Max-log-APP algorithm is implemented, with input quantization on 4 bits, which is a good trade-off between complexity and performance.

To reduce the decoding latency, two buffers are used at the input enabling bank-swapping. This is possible thanks to the high density of embedded memory blocks in recent FPGA device families.

3.2. Measurement results

The following figures exhibit TurboConcept's hardware turbo decoder FER performances, for block length of 53 and 188 bytes. We can observe that:

- Very close matching between simulation and hardware is obtained.
- FER down to 10^{-8} (equivalent to BER=10^{-10} / 10^{-11}) measurements show the absence of error floor.
- Regular behaviour regarding puncturing patterns (between different coding rates).

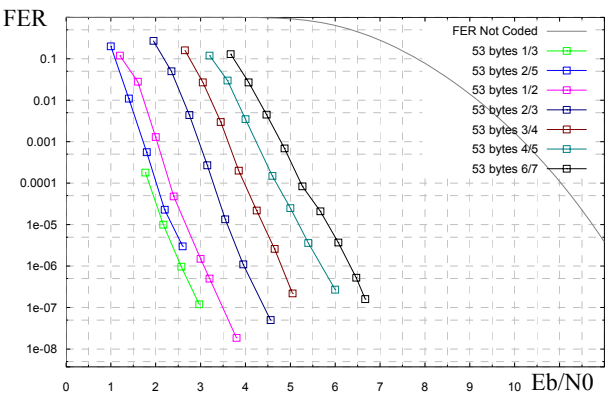


Figure 2: Frame Error Rate for ATM blocks, 8 iterations (TC1000 hardware measurements)

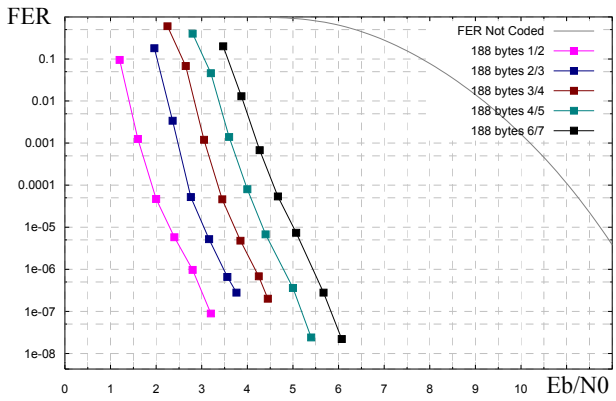


Figure 3: Frame Error Rate for MPEG blocks, 8 iterations (TC1000 hardware measurements)

4. CONCLUSION

The turbo code that was proposed for DVB-RCS applications is powerful, very flexible and can be implemented with reasonable complexity.

This code could also be easily adjusted to many other applications, for various configurations of block sizes and code rates while retaining excellent coding gains.

REFERENCES

[1] N. Brengarth, R. Novello, N. Pham, V. Piloni and J. Tousch, "DVB-RCS turbo code on a commercial OPB statellite payload: Skyplex", *Proc. of the 2nd Int'l Symp. on Turbo Codes",* Brest, France, Sept. 2000.

[2] C. Berrou and M. Jézéquel, "Non binary convolutional codes for turbo coding", *Electronics Letters,* Vol. 35, N°1, pp. 39-40, Jan. 1999.

[3] C. Berrou, C. Douillard, and M. Jézéquel, "Designing turbo codes for low error rates", *Digest of IEE Colloq. on "Turbo codes in digital brodcasting – could it double capacity?",* Vol. 165, Nov. 1999.

[4] C. Berrou, C. Douillard, and M. Jézéquel, "Multiple parallel concatenation of circular recursive convolutional (CRSC) codes", *Annals of Telecommunications,* Vol. 54, N°3-4, pp. 166-172, March-April 1999.

[5] P. Robertson, P. Hoeher and E. Villebrun, "Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding", *ETT,* Vol. 8, pp. 119-125, March-Apr. 1997.

[6] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes", *IEEE Trans. Com.,* Vol. 44, N°10, pp. 1261-1271, Oct. 1996.

[7] S. Dolinar, D. Divsalar and F. Pollara, "Code performance as a function of block size", TMO progress report 42-133, JPL, NASA.

[8] "TC1000 DVB-RCS turbo decoder data sheet", TurboConcept (http://www.turboconcept.com)

Annexe 8

Datasheet “TURBOΦ for the reverse link”, Contract ESA-Alenia Spazio, Modem for High-Order Modulation Schemes (MHOMS), 2003.

Ce document décrit les caractéristiques techniques détaillées du schéma de modulation turbocodée proposé dans le cadre du projet MHOMS.

MODEM for High-Order Modulation Schemes (MHOMS)

Contract ESA-Alenia Spazio 16593/02/NL/EC

Phase 1, WorkPackage 1415

TURBOΦ for the reverse link

Datasheet

Authors:
Claude Berrou, Catherine Douillard and Sylvie Kerouédan
with contributions from
Yannick Saouter, Laura Conde, Michel Jézéquel, Raoul Crespo, Jacky Tousch.

V1.2, Febuary 2004

1. Introduction

A flexible error-correcting turbo code has been devised for adaptive coding and modulation in high throughput satellite applications. This code, nicknamed TurboΦ, is derived from the extension of the DVB-RCS turbo code to 16 states. This code is intended to offer near-Shannon performance on Gaussian channel, in most situations of block size, coding rate up to 8/9 (or slightly more if needed) and associated modulation. It is expected that the FER (frame error rate) versus E_b/N_0 curve does not display any dramatic change in the slope (*flattening*) down to 10^{-7}, in the cases defined for the **reverse link of the MHOMS project**. Besides its high degree of flexibility regarding the system requirements and thanks to the generic model of its internal permutation, the specification of TurboΦ gives the natural possibility of using a high level of parallelism in the associated decoder.

This datasheet provides the full technical characteristics of the proposed scheme, together with examples of performance in various cases. A detailed presentation of the different concepts used in the specification of TurboΦ is available in the companion document "**Flexible Turbo Code for Low Error Rates**", Final report of the contract ESA-Alenia Spazio 16593/02/NL/EC, Phase 1, WorkPackage 1415, November 2003.

2. TURBOΦ code description for $R \geq 1/2$

The encoder is a parallel concatenation of two 16-state duo-binary recursive systematic convolutional (RSC) encoders, fed by blocks of k bits ($N = k/2$ couples). N is not a multiple of 15, and must be a multiple of 4, because of the permutation law.

Both component encoders have identical features (Fig. 2.1):
- 16-state duo-binary convolutional code
- polynomials 23 (recursivity, period 15), 35 (redundancy)
- first bit (A) on tap 1, second bit (B) on taps 1, D and D^3
- circular termination for both component encoders

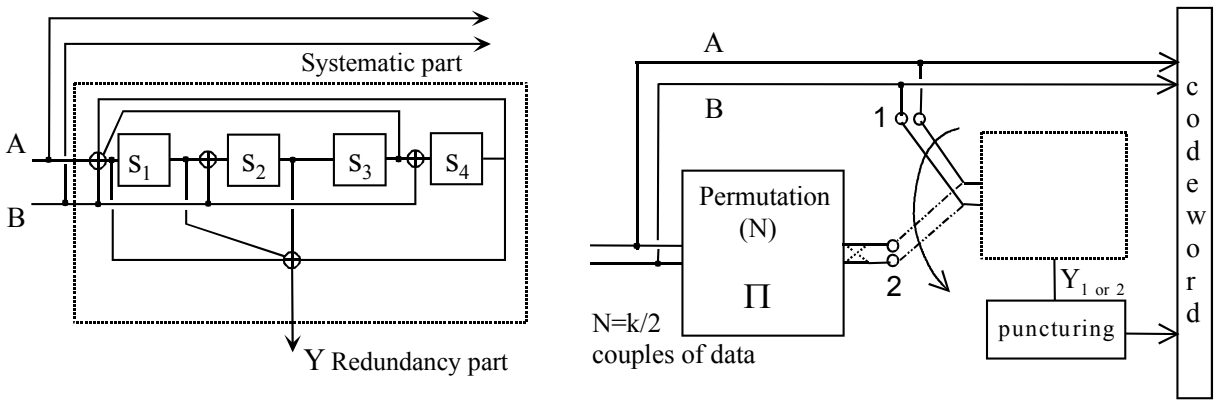


Figure 2.1. The TURBOΦ encoder (R ≥ 1/2).

The encoding of a block is performed according to the following sequence:

- calculating the circulation state in the natural order,
- encoding in the natural order (switch in position "1"), possible puncturing of redundancy Y_1,
- permuting couples ((A,B) becomes (B,A)) once every other time – the *draughtboard principle*, see section 3,

- calculating the circulation state in the permuted order,
- encoding in the permuted order (switch in position "2"), possible puncturing of redundancy Y_2.

Each circulation state is obtained as follows. The block, in its natural or permuted order, is encoded, starting from state **0**. The final state is then $\mathbf{S}_0$ ($\mathbf{S}$ is defined as $8.s_1 + 4.s_2 + 2.s_3 + s_4$, according to Fig. 1). *N* modulo 15 gives the row to consider in Table 2.1, and the intersection of this row with the value of $\mathbf{S}_0$ provides the value of the circulation state.

	$\mathbf{S}_0$															
N mod 15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
⋁																
1	0	14	3	13	7	9	4	10	15	1	12	2	8	6	11	5
2	0	11	13	6	10	1	7	12	5	14	8	3	15	4	2	9
3	0	8	9	1	2	10	11	3	4	12	13	5	6	14	15	7
4	0	3	4	7	8	11	12	15	1	2	5	6	9	10	13	14
5	0	12	5	9	11	7	14	2	6	10	3	15	13	1	8	4
6	0	4	12	8	9	13	5	1	2	6	14	10	11	15	7	3
7	0	6	10	12	5	3	15	9	11	13	1	7	14	8	4	2
8	0	7	8	15	1	6	9	14	3	4	11	12	2	5	10	13
9	0	5	14	11	13	8	3	6	10	15	4	1	7	2	9	12
10	0	13	7	10	15	2	8	5	14	3	9	4	1	12	6	11
11	0	2	6	4	12	14	10	8	9	11	15	13	5	7	3	1
12	0	9	11	2	6	15	13	4	12	5	7	14	10	3	1	8
13	0	10	15	5	14	4	1	11	13	7	2	8	3	9	12	6
14	0	15	1	14	3	12	2	13	7	8	6	9	4	11	5	10

Table 2.1. The table providing the circulation state.

3. The permutation law

i (i = 0, …N-1) is the couple address in the natural order. j (j = 0, …N-1) is the couple address in the permuted order.

if $j = 0$ mod. 4, then $Q = 0$;

if $j = 1$ mod. 4, then $Q = 4Q_1$;

if $j = 2$ mod. 4, then $Q = 4Q_0P + 4Q_2$; (3.1)

if $j = 3$ mod. 4, then $Q = 4Q_0P + 4Q_3$.

Finally, $i = \Pi(j) = Pj + Q + 13 \mod. N$ (3.2)

[Be careful about the way that the permutation is defined: i = Π(j) gives the natural address of the couple, when incrementing the address for second encoding]
P is an integer, relatively prime with N. Q0, Q1, Q2 and Q3 are small integers (typically from 0 up to 16). Equation (3.2) satisfied the "odd-even" rule (i and j have not same parity).
In addition to the inter-symbol permutation defined by (3.1) and (3.2), intra-permutation is performed according to the following rule (the so-called draughtboard principle):

if j = 0 mod. 2, then invert (A,B) (i.e. (A,B) becomes (B,A); (3.3)

That is, the couple is permuted once every other time before second encoding, the process beginning with the permutation of the first couple.

Table 3.1 provides the parameters obtained for the block sizes considered in the reverse link. The minimum Hamming distances (MHDs), estimated by the Error Impulse Method (EIM), are also provided for some typical coding rates. Values in parentheses give the part of bits, A or B, which belong to codewords at the minimum distance (normalized multiplicity in the EIM sense).

k	*N=k/2*	*R*=1/3	*R*=1/2	*R*=2/3	P	Q0	Q1	Q2	Q3
128	64	*d*min=19 (0.25)	*d*min=11 (0.125)	*d*min=8 (0.75)	19	1	3	8	1
456	228	*d*min=34 (0.125)	*d*min=20 (0.25)	*d*min=12 (0.5)	23	1	11	2	3
880	440	*d*min=38 (0.125)	*d*min=21 (0.125)	*d*min=13 (0.125)	49	1	9	2	6
1304	652	*d*min=40 (0.125)	*d*min=26 (0.125)	*d*min=17 (0.375)	107	2	10	15	1
1504	752	*d*min=45 (0.125)	*d*min=27 (0.125)	*d*min=17 0.125	43	1	1	1	2
1728	864	*d*min=46 (0.25)	*d*min=26 (0.125)	*d*min=17 (0.25)	67	2	4	13	5
3008	1504	*d*min=46 (0.125)	*d*min=29 (0.125)	*d*min=20 (0.5)	53	2	1	2	2

Table 3.1. Permutation parameters for the different block sizes. Minimum Hamming distances estimated by the Error Impulse Method, with associated multiplicities, are also given for R = 1/3, 1/2 and 2/3.

4. TURBOΦ code description for *R* < 1/2

In order to proceed with 1/3 ≤ R < 1/2, a second redundancy symbol, denoted W, is generated for both component encoders by the way indicated in Fig. 4.1, and corresponding to generator polynomial 27.

The contribution of W in the transition metrics does not require the metrics format to be extended, because the contributions of A, B and Y already need 4 times the dynamics of the decoder inputs.

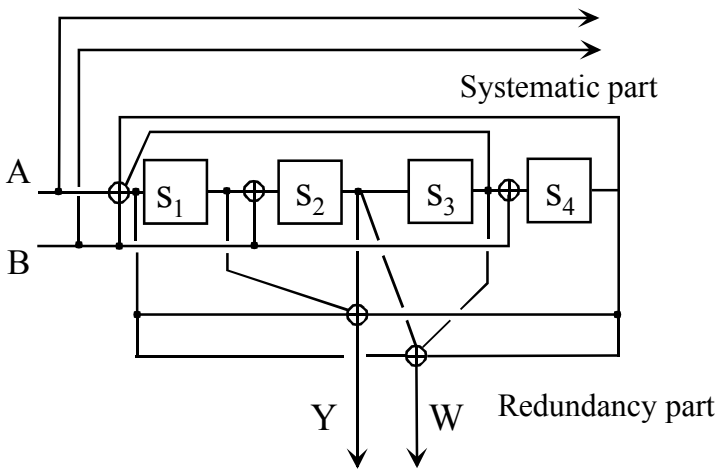


Figure 4.1 The component encoder of TURBOΦ for R ≤ 1/2.

5. Puncturing patterns

The puncturing patterns are identical for both component encoders of TurboΦ. If $R \geq 1/2$, the puncturing operation is performed on redundancy Y in Fig. 2.1. If $R < 1/2$, the puncturing operation is performed on redundancy W in Fig. 4.1.

The puncturing rule is defined in a way that alleviates the detrimental effects of irregular puncturing. Figure 5.1 depicts the patterns that are chosen to fit the desired rate. In each of them, "1" means "puncturing", "0" transmitting, "P" possible puncturing. The upper pattern is for puncturing rates lower than 0.5; the second one is for 0.5 <Puncturing rate ≤ 0.75, and so on. Each L places (L = 2, 4, 8 or 16), the parity bits are systematically transmitted. Each other L places, at equal distance from the systematically transmitted bits, the parity bits can either be transmitted or punctured. The other locations correspond to parity bits that are systematically punctured.

L = 2, 0 < Puncturing rate ≤ 0.5	0P0P0P0P0P0P0P0P0P0P0P0P....
L = 4, 0.5 < Puncturing rate ≤ 0.75	01P101P101P101P101P101P10....
L = 8, 0.75 < Puncturing rate ≤ 0.875	0111P1110111P1110111P1110....
L = 16, 0.875 < Puncturing rate ≤ 0.9375	01111111P111111101111111P....

Figure 5.1. Puncturing patterns for different puncturing rates. "1": parity bit punctured, "0": parity bit transmitted, "P": possible puncturing.

The puncturing principle, at places denoted "P", is depicted in Fig. 5.2, where an adder/accumulator adds a number, denoted I_p (*puncturing increment*), to the previous content of the accumulator, for each new information symbol (couple). The accumulator stores the fractional part of the summation result, so that the integer part of the summation result is either 0 or 1. Each time that the output of the puncturing machine is 1, the parity bit stemming from the encoder considered is punctured (not transmitted).

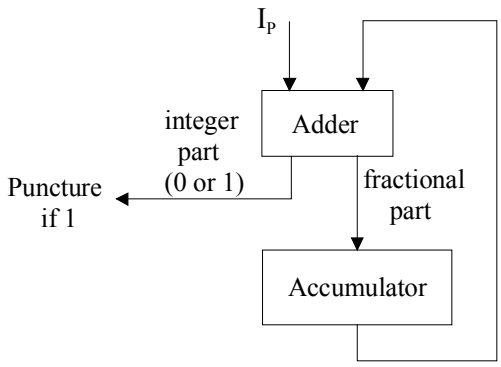


Figure 5.2. The transmitting operator. I_p is a truncated positive real number.

Calling R_p the global puncturing rate for the parity bit in question, and introducing the integer n_R which takes value 1 for $R \geq 1/2$ and 2 for $R < 1/2$, we have the relations:

$$R_p = \frac{R(n_R + 1) - 1}{R}; \quad I_p = L(R_p - 1 + \frac{2}{L}); \tag{5.1}$$

The accuracy needed for the representation of I_p is given by the worst case, that is, the maximum value for the number of information symbols (couples): N_{max} = 1504. If we denote by I_p^* the rounded value of I_p, with $I_p^* > I_p$, the condition for correct functioning is:

$$(I_p^* - I_p)N_{max} < 1$$

that is, with the actual value of N_{max}: $I_p^* - I_p < 7.10^{-4}$. I_p^* can then be represented with 4 digital numbers, the last digit being that of I_p plus 1. Table 5.1 provides the parameters for each size/rate of the reverse link. Because the fractional part of I_p^* is less than 9999, the adder/accumulator can be implemented with 14-bit format.

information bits	nR one parity bit (1) two parity bits (2)	information symbols	parity bits (total)	parity bits (component)	codeword length	rate			
k		N=k/2	P	P/2	n	R	RP	L	IP *
128	2	64	256	128	384	0,33333	0,00000	2	0,0001
456	1	228	312	156	768	0,59375	0,31579	2	0,6316
880	1	440	656	328	1536	0,57292	0,25455	2	0,5091
880	1	440	656	328	1536	0,57292	0,25455	2	0,5091
880	2	440	1424	712	2304	0,38194	0,38182	2	0,7637
1304	1	652	1000	500	2304	0,56597	0,23313	2	0,4663
1304	1	652	1000	500	2304	0,56597	0,23313	2	0,4663
1304	2	652	1768	884	3072	0,42448	0,64417	4	0,5767
1304	2	652	2536	1268	3840	0,33958	0,05521	2	0,1105
1504	1	752	800	400	2304	0,65278	0,46809	2	0,9362
1504	2	752	1952	976	3456	0,43519	0,70213	4	0,8086
1504	2	752	1568	784	3072	0,48958	0,95745	32	0,6383
1504	2	752	2336	1168	3840	0,39167	0,44681	2	0,8937
1728	1	864	1344	672	3072	0,56250	0,22222	2	0,4445
1728	1	864	1728	864	3456	0,50000	0,00000	2	0,0001
1728	1	864	1344	672	3072	0,56250	0,22222	2	0,4445
1728	2	864	2112	1056	3840	0,45000	0,77778	8	0,2223
3008	1	1504	1600	800	4608	0,65278	0,46809	2	0,9362
3008	1	1504	1600	800	4608	0,65278	0,46809	2	0,9362
3008	1	1504	2752	1376	5760	0,52222	0,08511	2	0,1703

Table 5.1. Puncturing increments I_p^* for all the cases of the return link (some cases have same Ip^*). The actual value of I_p^* for k = 1504, R = 0.48958 is not 0.6383 as indicated by the table, because N is not a multiple of L = 32. 0.6383 has been replaced by 0.6700

For the block lengths which are not multiple of L, the value of I_p^* indicated in table 5.1 may be not the proper one. There is one case: k = 1504, R = 0.48958, which needs to correct the value provided by formula (5.1). In this particular case, 0.6383 has been replaced by 0.6700.

6. Decoding

The component decoding algorithm is either the full Log-MAP algorithm, or its simplified version Max-Log-MAP. For all simulations, we used the latter, not only for speed convenience, but also because the difference of performance between the two algorithms is not a determining factor (less than 0.1 dB) when working at low error rates.

Among several, one important parameter in the turbo decoder is the peak value of extrinsic pieces of information, denoted Z_{max}. Because MHDs can be rather large for low coding rates, we recommend to raise this peak value to $8.V_{max}$, where V_{max} is the peak absolute value of the decoder input symbols. Lower values could have impairing effects at low error rates, especially for long blocks and low coding rates. In the practical implementation of the turbo decoder, the $8.V_{max}$ limitation allows quaternary extrinsic information to be quantized with 8 bits, if the channel samples have 6-bit quantization. Nevertheless, for high coding rates, the peak value of extrinsic information can advantageously be limited to $2.V_{max}$ or $4.V_{max}$.

As for the feedback gain (the multiplying factor for extrinsic pieces of information, denoted γ), which is an essential parameter in Max-Log-MAP turbo decoding, we took it equal to 0.5 for the first two iterations, 0.75 or 0.85 for the following ones, except the last one which has a feedback gain of 1. The latter value enables the decoder to profit from the MHD to the full. For the intermediate iterations, choosing $\gamma = 0.75$ will favour the Bit Error Rate, while choosing 0.85 will favour the Frame Error Rate. Nevertheless, for short blocks (i.e. 57 bytes), a feedback gain of 0.75 is preferable for intermediate iterations.

7. Decoding parallelism

The kind of parallelism proposed, that is, the possibility to use several processors without increasing the memory requirements, is inherent to the permutation law[1]

The permutation, defined by (3.1) and (3.2), has a cycle of 4, which is a divider of all possible sizes N considered. The cycle of a permutation Π is the lowest integer T that satisfies:

$$\Pi(j+T) - \Pi(j) = \text{constant} \ \ \text{mod.}\, N, \text{whatever } 0 \le j \le N-1 \qquad (7.1)$$

In our case, with cycle 4, the constant is precisely $4.P$. Relation (7.1) is valid for all values of j only if T is a divider of N.

Because the permutation cycle is 4, the congruences of j and $\Pi(j)$, modulo 4, are periodic. A parallelism with degree $T = 4$ is then possible, for both decoding in natural and permuted order, according to the technique depicted in Fig. 7.1. Four SISO (forward or backward) processors are assigned to the four quadrants of the circle under process. At the same time, the four units deal with data that are located at places corresponding to the four possible congruences.

[1] The trivial parallelism related to the natural decomposition of turbo processing in backward/forward and first/second encoding is not addressed here

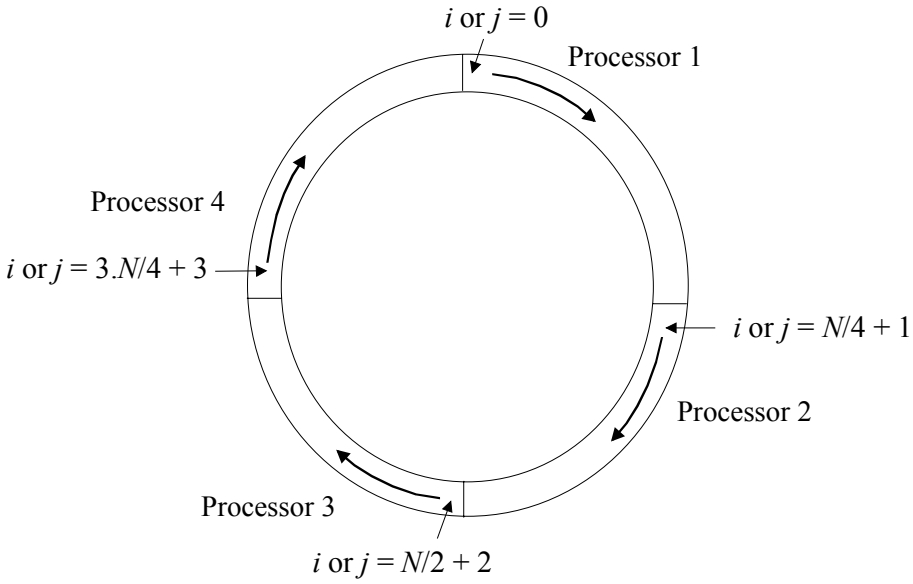


Figure 7.1. The circle under process is divided into four quadrants, with four associated SISO (forward or backward) processors, allowing a parallelism degree of 4.

For instance, at the beginning of the process (not taking into account a possible prologue), the first processor deals with data located at an address with congruence 0, the second with congruence 1, and so on. The clock period just after, the first processor will handle data with congruence 1 address, the second one with congruence 2 address, etc. So, through a shift barrel which directs the four processors towards four distinct memory pages corresponding to the four congruences, a parallelism with degree 4 is feasible.

Larger parallelism with degree $4p$ is possible, where p is any integer, provided that N is also a multiple of p. From (7.1), T being the cycle of Π, any multiple pT of T is also a cycle of the permutation, provided that $4p$ is a divider of N. Under this condition, j mod. $4p$ and $\Pi(j)$ mod. $4p$ are periodic over the circles of length N, and a parallelism of degree $4p$ is feasible. For instance, parallelism with degree 128 is workable for $N = 4096$ ($k = 8192$ bits).

8. Association of TurboΦ with high-order modulations

Since high flexibility is required for MHOMS, the so-called "pragmatic" turbo coded modulation approach is adopted:

- Same turbo encoder and decoder are used for all the possible associations with the different modulations. The required coding rate is achieved by puncturing some parity check bits at the encoder output. At the demodulator output, an m-ary to binary conversion is performed (an LLR is computed for each bit in the received symbol).
- The mapping of modulation constellations calls for classical Gray or quasi-Gray encoding.

For 8PSK and 4+12APSK modulations, the bits defining a symbol are not equally protected by the modulation. In practice, two different average protection levels have to be considered. The way systematic and redundant bits are protected has an influence on both the convergence and the asymptotic behaviour of the turbo coded modulation scheme. For each MHOMS case, we have searched for the minimum part θ of redundant bits to be associated with potentially better protected places, allowing a target frame error rate of 10-6-10-7 to be achieved without any significant flattening.

The set of values for θ are provided in Tables 8.1 and 8.2.

Block size k (bits)	1304	1728	1504	1504	3008	3008
Coding rate R	0.56597	0.5	0.43519	0.65278	0.52222	0.65278
Percentage of redundancy potentially better protected θ	61%	69%	74%	67%	63%	67%

Table 8.1. Percentage θ of redundancy bits potentially better protected, for the cases of the return link using 8PSK modulation.

Block size k (bits)	880	1728	3008
Coding rate R	0.57292	0.56250	0.65278
Percentage of redundancy potentially better protected θ	39%	38%	86%

Table 8.2. Percentage θ of redundancy bits potentially better protected, for the cases of the return link using 4+12APSK modulation.

In the 4+12APSK case, an additional interleaving to break data couples turned out to be necessary to avoid any change in the slope. The three 4+12APSK cases were simulated with the following regular permutation Π' performed on bits systematic bits B before symbol mapping:

$$i' = \Pi'(j) = P'j + 13 \quad \text{mod. } N, \qquad j = 0, \ldots N\text{-}1 \tag{8.1}$$

The following values of P' were used for the simulations:

$$\begin{aligned} k = 880 &\Rightarrow P' = 29 \\ k = 1728 &\Rightarrow P' = 41 \\ k = 3008 &\Rightarrow P' = 55 \end{aligned} \tag{8.2}$$

The construction of the modulation symbols calls on two adder/accumulator structures similar to the one used for the puncturing patterns, in order to ensure that the systematic and parity bits with a higher average protection level are regularly spread along the code trellis.

At each time i, $0 \le i \le N-1$, the accumulator contents are incremented with two numbers R_s (*systematic rate*) and R_r (*redundancy rate*), such as

$$\begin{aligned} R_s &= k_1 / k \\ R_r &= P_1 /(n-k) = (n - kR_s)/(n-k) \end{aligned} \tag{8.3}$$

where we have denoted by k_1 and P_1 the number of systematic and redundant bits that are placed in the better protected bits of the modulation symbols.

Table 8.3 provides the values of R_s and R_r for all the 8PSK and 4+12APSK cases of the return link.

Information bits	Modulation	Number of bits potentially better protected by the modulation	Code rate	Number of potentially better protected systematic bits		Number of potentially better protected parity bits	
k		*n*1	*R*	*k*1	*Rs*	*P*1	*Rr*
1304	8PSK	1536	0,56597	922	0,7072	614	0,6141
1504	8PSK	2304	0,43519	852	0,5666	1452	0,7440
1504	8PSK	1536	0,65278	844	0,5613	692	0,8651
1728	8PSK	2304	0,50000	1106	0,6401	1198	0,6934
3008	8PSK	3840	0,52222	2112	0,7022	1728	0,6280
3008	8PSK	3072	0,65278	2006	0,6670	1066	0,6664
880	16APSK	768	0,57292	512	0,5819	256	0,3903
1728	16APSK	1536	0,56250	1024	0,5927	512	0,3811
3008	16APSK	2304	0,65278	922	0,3066	1382	0,8639

Table 8.3. Number of systematic and parity bits associated with the better protected places of the modulation symbols and values of Rs and Rr for all the 8PSK and 4+12APSK cases of the return link, according to Tables 8.1 and 8.2.

When filling up the modulation symbols, at each time i, if the output of the "systematic" adder/accumulator is 1, the corresponding data couple (A_i, B_i) is put in the two better protected places in the modulation symbol. Otherwise it is put in two less protected places in the symbol sequence. This adder/accumulator is incremented once every time i.

Regarding the parity bits, the associated adder/accumulator is incremented once every time i if $R \geq 0.5$ (for parity couple (Y1i, Y2i)) and twice if $R < 0.5$ (once for parity couple (Y1i, Y2i) and once for parity couple (W1i, W2i)). If its output is 1, the corresponding parity couple, if not punctured, is put in the two better protected places in the modulation symbol. If the adder/accumulator output is 0, the parity couple, if not punctured is put in two less protected places in the symbol sequence.

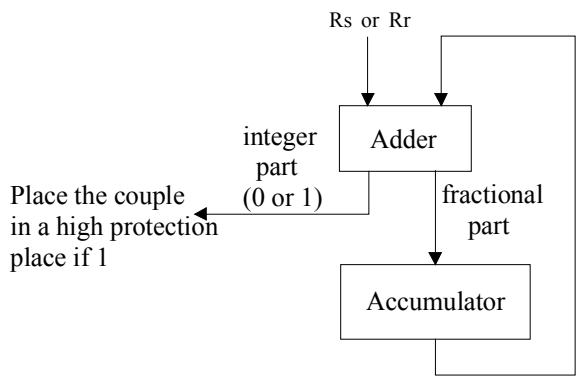


Figure 8.1. The basic operator used for symbol construction. Values of R_s and R_r are given in Table 8.3.

9. Performance results

This section provides performance, in terms of BER and FER, of the TurboΦ code associated with QPSK, 8PSK and 4+12APSK modulations. In each case, the FER performance is compared with the theoretical limit calculated from the java tool available at www-elec.enst-bretagne.fr/turbo/LIMIT/.

Figures 9.1 to 9.5 show TurboΦ simulation results with QPSK modulation on AWGN channel for five block sizes: $k = 456$ bits, $k = 880$ bits, $k = 1304$ bits, $k = 1504$ bits and $k = 3008$ bits. The FER performance compared with the theoretical limit shows a difference varying from 0.6 to 0.75 dB depending on the block size and on the code rate, for FER = 10^{-6}.

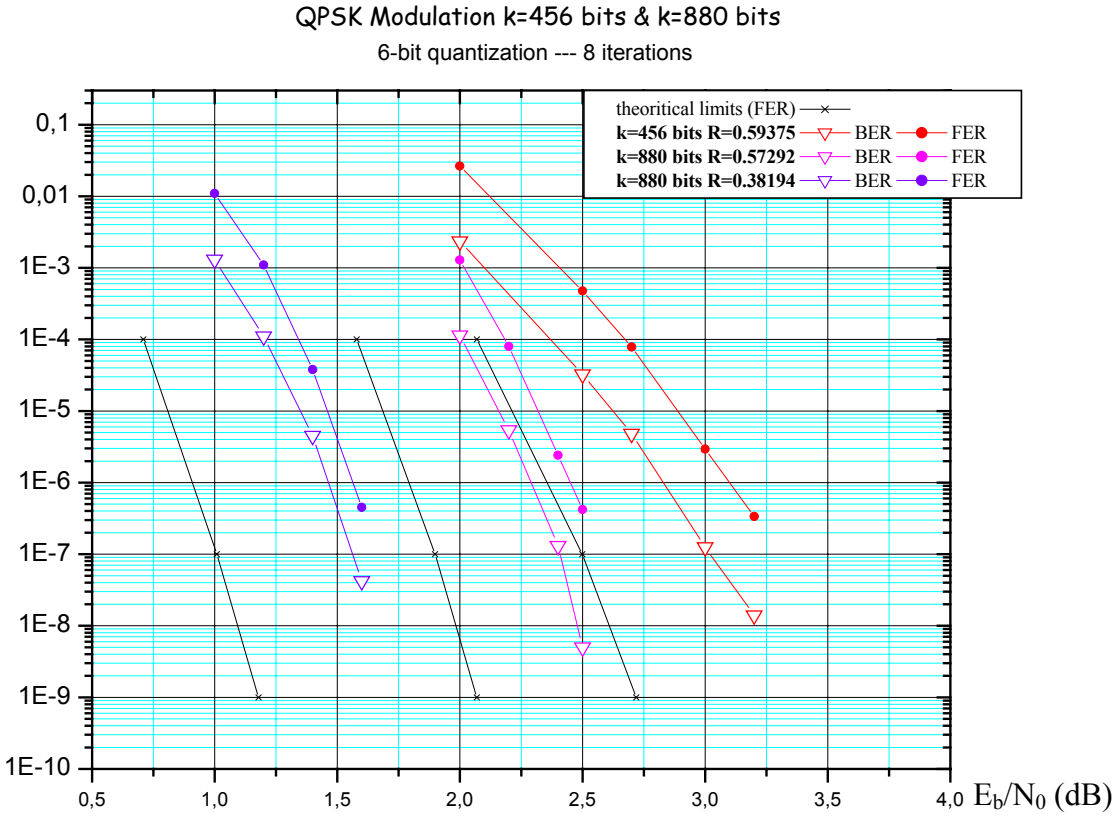


Figure 9.1. Performance of TurboΦ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN. ***57 bytes****: R = 0.59375, $Z_{max} = 2.V_{max}$, $\gamma = 0.5, 0.75, 1$.* ***110 bytes****: R = 0.38194 and 0.57292, $Z_{max} = 8.V_{max}$, $\gamma = 0.5, 0.85, 1$.*

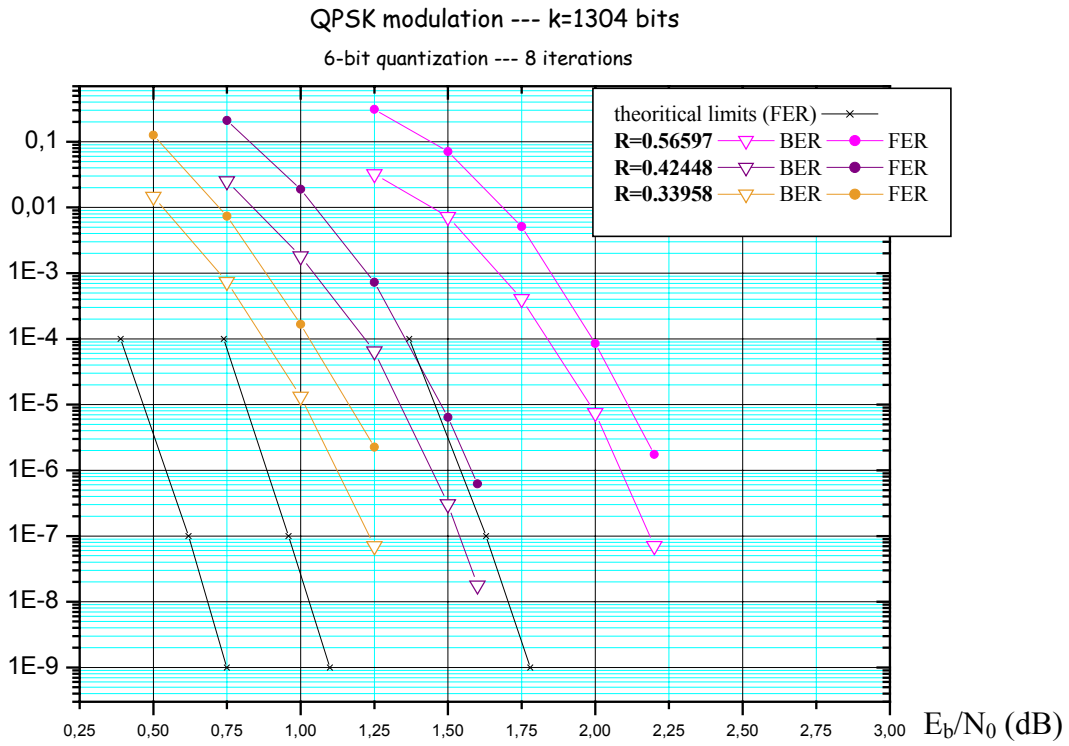


Figure 9.2. Performance of TurboΦ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN. ***163 bytes****: R=0.33958, 0.42448 and 0.56597, Z_{max} = 8.V_{max}, γ=0.5, 0.85, 1.*

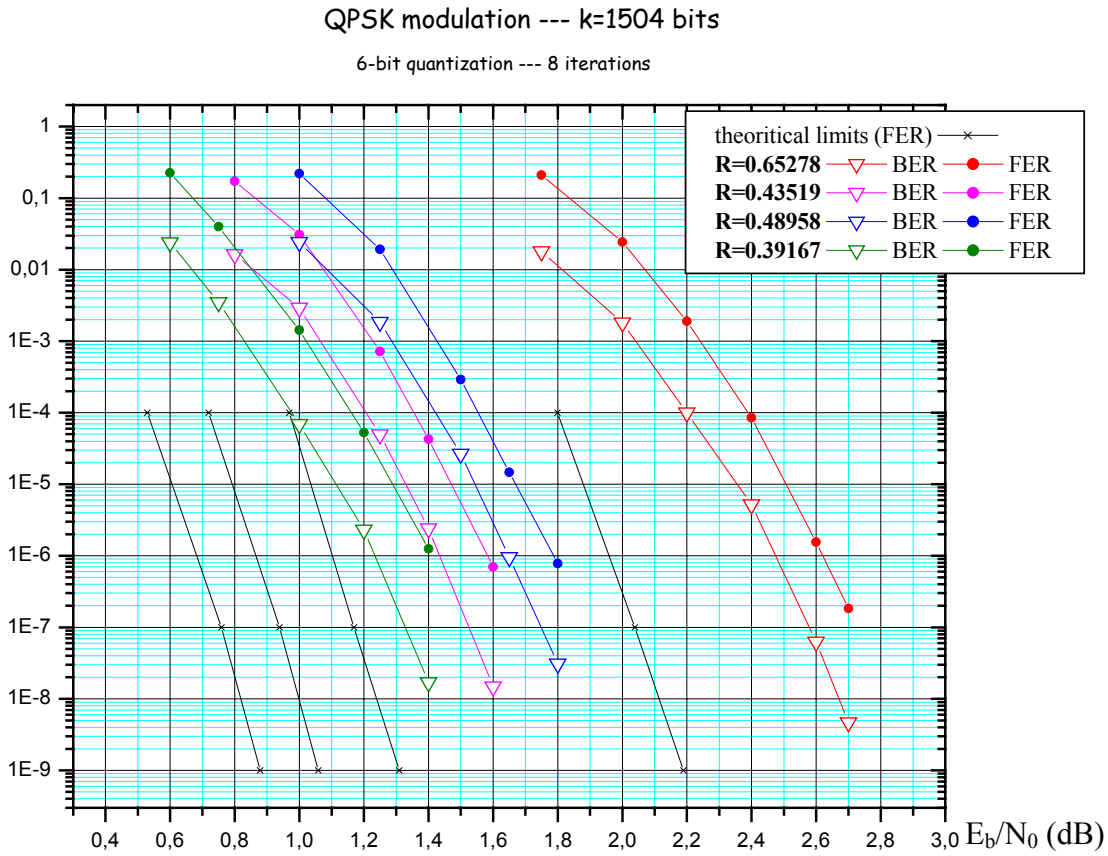


Figure 9.3. Performance of TurboΦ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN. ***188 bytes****: R=0.39167, 0.43519, 0.48958 and 0.65278, Z_{max} = 8.V_{max}, γ=0.5, 0.85, 1.*

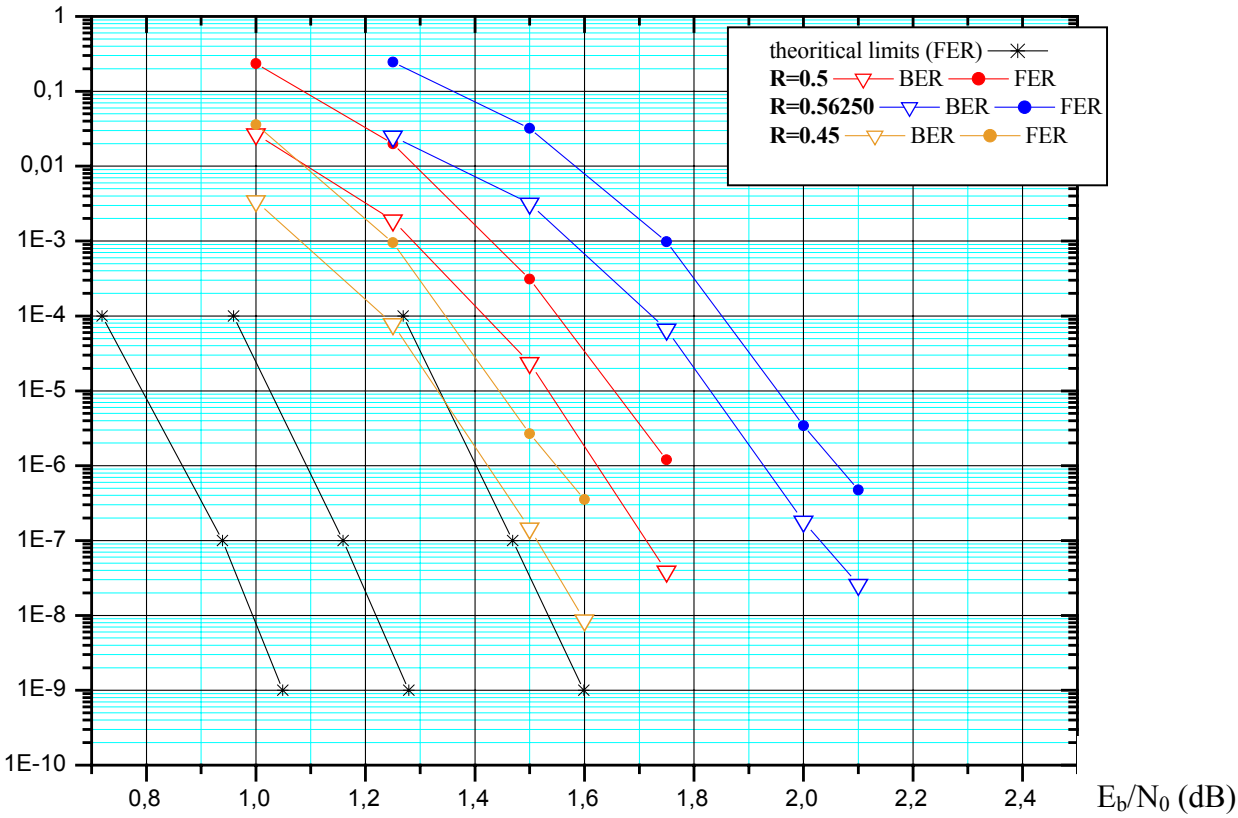


Figure 9.4. Performance of Turbo Φ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN. ***216 bytes****: R=0.45, 0.5, and 0.56250,* $Z_{max}=8.V_{max}$, $\gamma=0.5, 0.85, 1$.

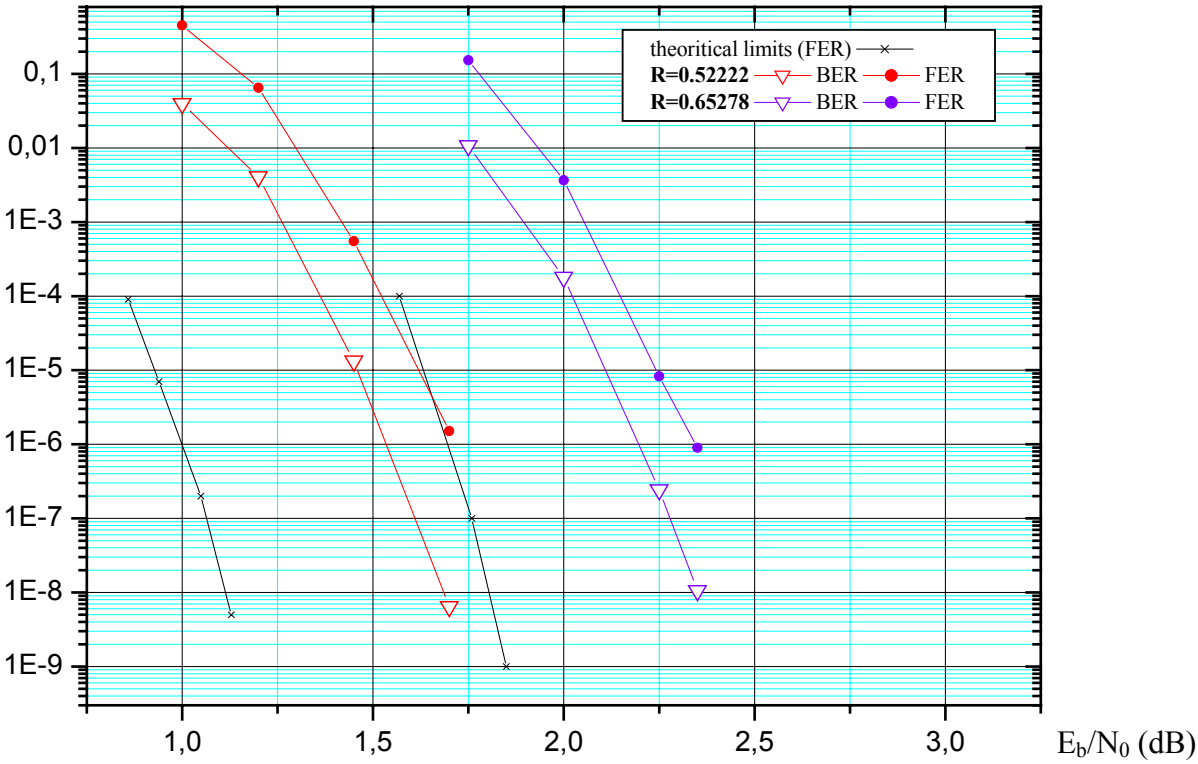


Figure 9.5. Performance of Turbo Φ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN. ***376 bytes****: R=0.52222 and 0.65278,* $Z_{max}=8.V_{max}$, $\gamma=0.5, 0.85, 1$.

Figure 9.6 shows simulation results for a transmission over an AWGN channel using an 8PSK modulation for the different proposed block sizes and coding rates. The difference between the simulated performance and the theoretical limits at FER = 10^{-6} varies from 1.0 to 1.15 dB.

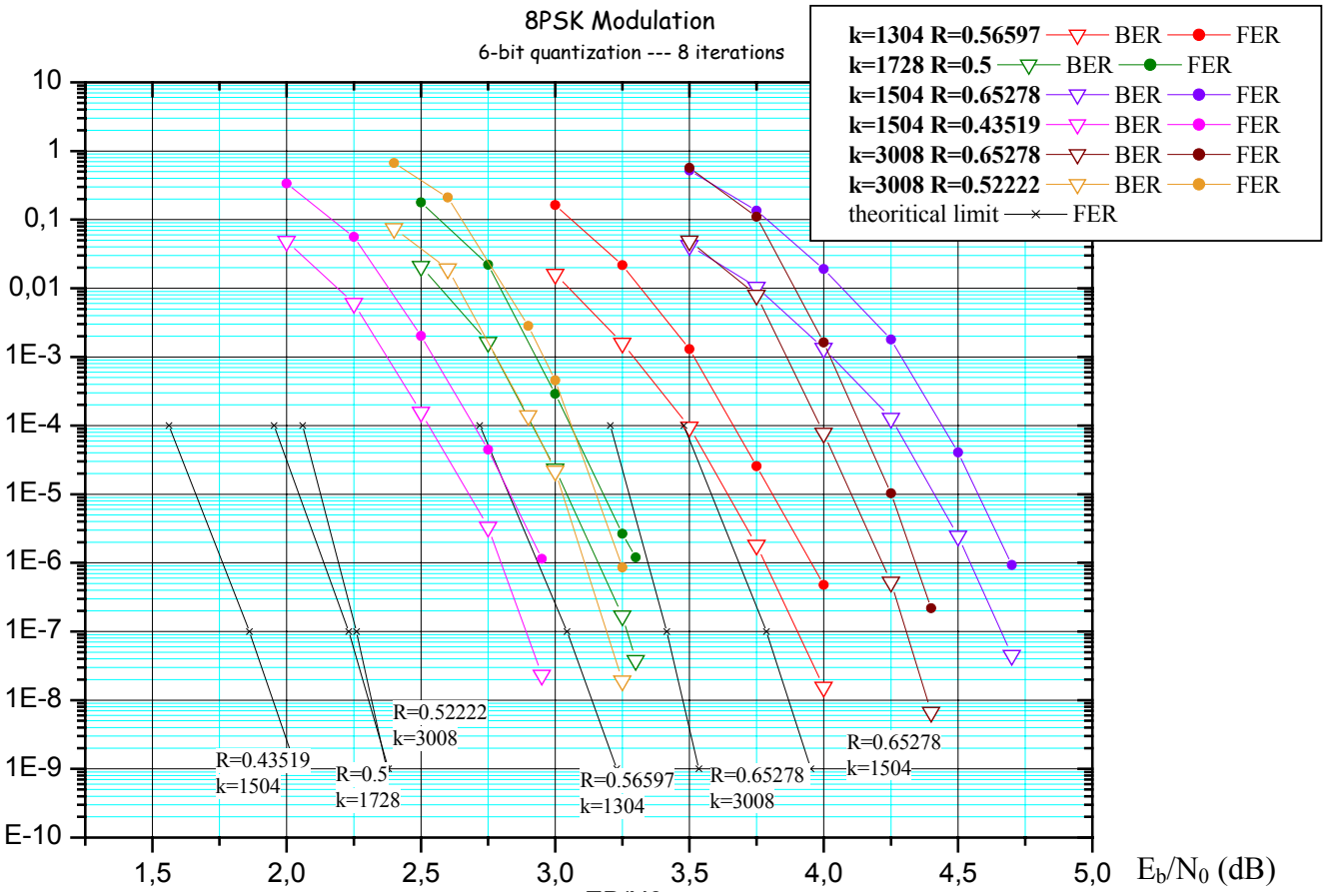


Figure 9.6. Performance of TurboΦ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN with 8PSK modulation. ***163 bytes****: R=0.56597, 188 bytes: R=0.43519 and 0.65278,* ***216 bytes****: R=0.5 and* ***376 bytes****: R=0.52222 and R=0.65278, Z_{max}=8.V_{max}, γ=0.5, 0.85, 1.*

Figure 9.7 shows simulation results for a transmission over an AWGN channel using a 4+12PSK modulation. A second interleaver was added to prevent from an error floor around FER=10^{-5}, at the expense of a slight loss of convergence (less than 0.1dB). Thus, at FER=10^{-6}, the difference between the simulated performance and the theoretical limits is between 1.1 and 1.3 dB, without any change of the curve slope.

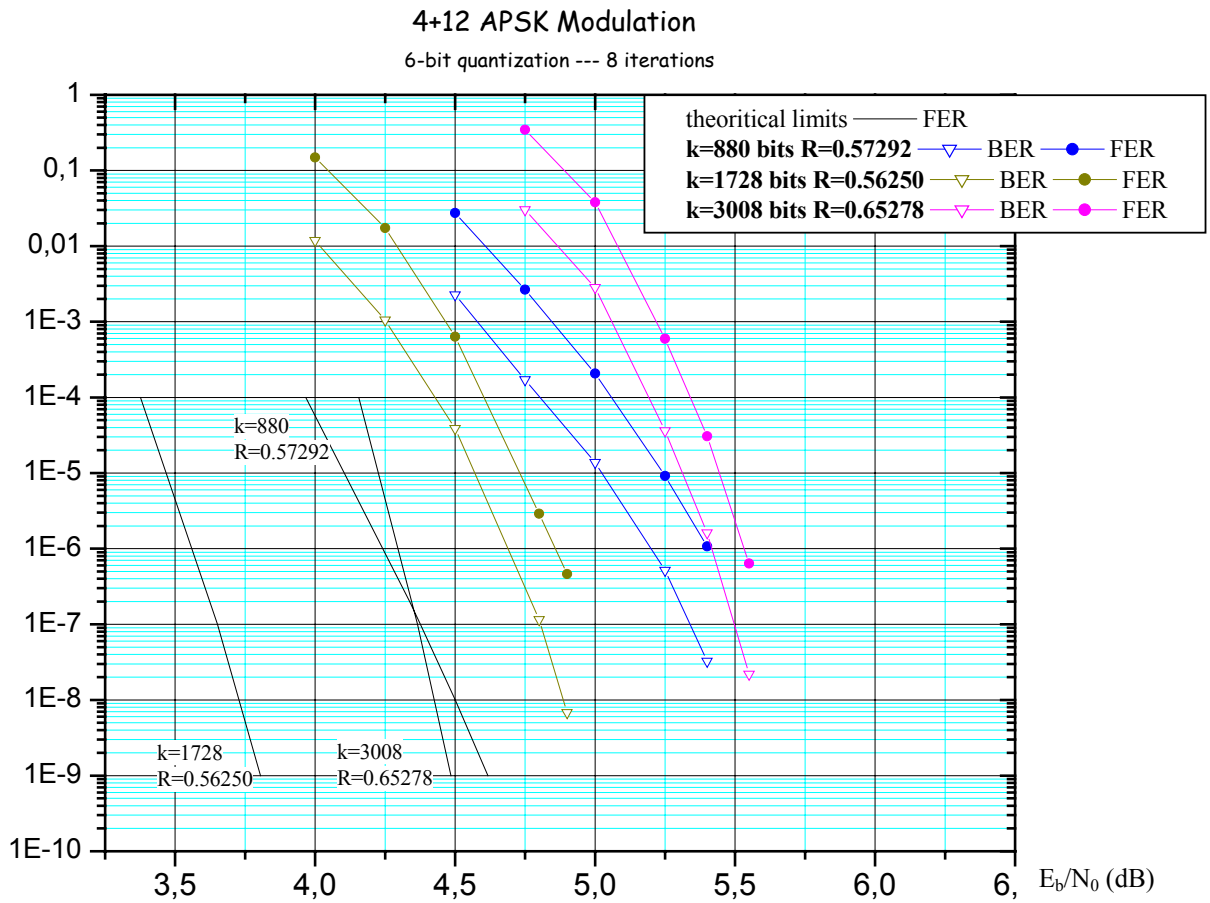


Figure 9.7. Performance of TurboΦ with Max-Log-MAP decoding algorithm, 6-bit quantization, 8 iterations, AWGN with 4+12-APSK modulation. ***110 bytes****: R=0.57292,* ***216 bytes****: R=0.56250 and* ***376 bytes****: R=0.65278, $Z_{max}=8.V_{max}$, $\gamma=0.5, 0.85, 1$.*

Annexe 9

Article "Performance estimation of 8-PSK turbocoded modulation over Rayleigh fading channels", *3rd International Symposium on Turbo Codes & Related Topics*, 2003.

Cet article décrit l'utilisation de la méthode de l'impulsion d'erreurs pour l'estimation des performances asymptotiques des modulations turbocodées sur canal de Rayleigh.

Performance estimation of 8-PSK turbo-coded modulation over Rayleigh fading channels

Laura Conde Canencia and Catherine Douillard

PRACOM/ENST Bretagne, Technopôle Brest Iroise, BP 832, 29285 BREST Cedex FRANCE
Phone: (+33) 2 29 00 13 52, Fax: (+33) 2 29 00 11 84
E-mail: conde-canencia@ieee.org

Abstract: *We present new estimation techniques that make it possible to predict the asymptotic performance of turbo-coded modulation over Rayleigh fading channels. These techniques require the knowledge of the minimum Hamming distance of the turbo code. The Error Impulse Method provides this information. The derived expressions present two main advantages over previous work: they are valid for any kind of code interleaver and do not require any information about the component codes.*

Keywords: coded modulation, union bound, fading channel, bit-interleaving, turbo code

1. INTRODUCTION

Transmission in bandwidth-constrained channels can be error-protected if a coded-modulation technique is used. These techniques make channel coding possible without expanding the signal bandwidth. The most popular of these techniques is Trellis-Coded Modulation (TCM), introduced by Ungerboeck [1]. As TCMs attracted the interest of many researchers, they led to considerable research [2]: theory formalization, performance bounds and the search for new TCM schemes.

The evaluation of performance bounds for TCMs in fading channels was considered in [3], where a Chernoff bound was applied to upper-bound the error probability. However, this bound (commonly used because of its simplicity) is too loose for most of the signal-to-noise ratio values, especially for the Rayleigh fading channel. In [4], Slimane *et al* presented new tight bounds for Rayleigh fading channels. Finally, the exact union bound for TCMs over fading channels was evaluated in [5].

Turbo-coded modulation is an alternative to TCM. Different approaches to turbo (trellis)-coded modulation were presented in [6], [7], [8]. Before [9], the only method used to study the performance of turbo-coded modulations over fading channels was computer Monte-Carlo simulations. In [9], the authors presented a methodology to derive bounds for specific turbo-coded modulation schemes. This method requires the knowledge of the weight enumerating function of the component codes. Unfortunately, it is only valid when the interleaver in the turbo code is uniform (i.e. statistically uniform).

An original new method for computing minimum distances of linear codes, in particular turbo codes, was presented in [10]. This method, called the Error Impulse Method (EIM), provides an estimation of the distance spectrum of the turbo code. The prediction of performance at low error rates of turbo code-BPSK/QPSK associations is then straightforward for Gaussian channels. In [11] and [12], the EIM was applied to estimate the performance of bit-interleaved turbo codes associated with high-order modulations over the Gaussian channel.

In this paper, we present the derivation of three expressions to estimate the performance of bit-interleaved turbo-coded modulation at low error rates over the Rayleigh fading channel. Our method is valid for any kind of code interleaver, as it is based on the application of the EIM (in fact, the EIM is a powerful tool when designing good interleavers for turbo codes). Moreover, no information about the component codes is required.

Throughout the paper, we adopt a slow-fading memoryless non-selective channel model. The derived expressions will be particularized for the Rayleigh channel. Coherent detection, maximum-likelihood (ML) decoding and ideal Channel State Information (CSI) are assumed in our study.

The paper is organized as follows. Section 2 presents the principle of the transmission scheme. In Section 3, we present the estimations on the error probability, that are a function of the effective distance of the code and the product distance of the coded modulation. In Section 4, we state the hypotheses that will enable us to apply the EIM to our scheme. The final computable expressions are also given. Finally, Section 5 gives some examples of comparisons between the simulation curves and the estimations.

2. TRANSMISSION SCHEME

We consider the transmission scheme depicted in Fig. 1. This scheme follows the principle of the pragmatic approach [6], together with the principle of Bit-Interleaved Coded Modulation (BICM) [13]. Our scheme contains a turbo code, a random bitwise in-

terleaver and a memoryless 8-PSK modulator. In the emitter, the information sequence is turbo encoded before being bitwise interleaved. The bit-interleaver (π) is assumed to be ideal (i.e. infinite depth and completely random). Its purpose is to break the sequential fading correlations and increase the diversity order to the minimum Hamming distance of the turbo code. We consider an 8-PSK modulation that follows a classical Gray mapping. The constellation and mapping are presented in Fig. 2.

The theoretical study assumes ML decoding. However, in our simulations the decoding performs a symbol-to-bit LLR calculation, followed by bit deinterleaving $(\pi)^{-1}$ and a turbo decoder that uses the Max-Log-MAP algorithm.

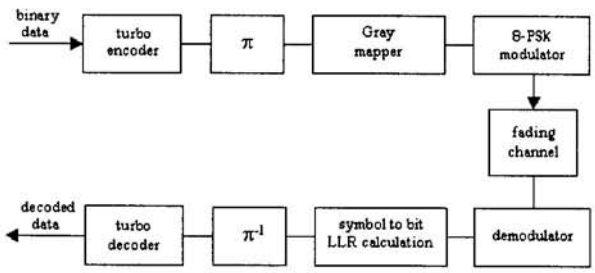


Figure 1: Transmission scheme

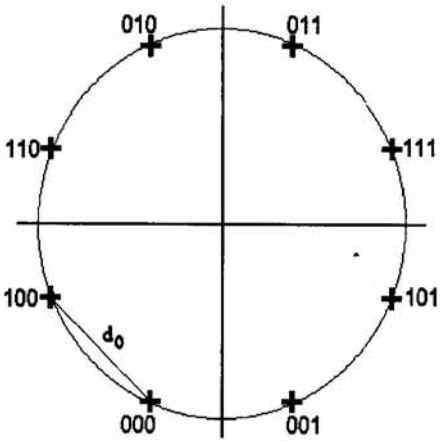


Figure 2: 8-PSK constellation with Gray mapping

3. UNION BOUND FOR TURBO-CODED MODULATIONS

3.1. Notation

In our transmission scheme, the input bits are turbo encoded and then mapped to produce a sequence of signals $\mathbf{s}_l = (s_1, s_2, ..., s_l)$, where l is the number of symbols in the sequence and each signal s_i is a two-dimensional vector chosen from the 8-PSK signal set (see Fig. 2).

Let $\mathbf{s}_l = (s_1, s_2, \cdots, s_l)$ be the transmitted sequence and $\mathbf{r}_l = (r_1, r_2, \cdots, r_l)$ the received sequence. The decoder makes an error if it decodes $\hat{\mathbf{s}}_l = (\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_l)$ instead of $\mathbf{s}_l$ (if $\hat{\mathbf{s}}_l \neq \mathbf{s}_l$). The received signal at time i can be written as:

$$r_i = a_i \cdot s_i + n_i \tag{1}$$

where a_i is the amplitude of the fading process and n_i is a sample of a zero-mean complex Gaussian noise process with variance $N_0/2$. The *pairwise error probability*, denoted by $P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l)$, is the probability that the decoder chooses $\hat{\mathbf{s}}_l$ instead of $\mathbf{s}_l$.

3.2. Bounds on the pairwise error probability

We denote by Q the set of all i for which $s_i \neq \hat{s}_i$ and by l_Q the cardinal of Q. If we use a Chernoff bound, $P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l)$ is upper-bounded as follows [9]

$$P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \leq \prod_{i=1}^{l_Q} \frac{1}{1 + \frac{1}{4N_0}|s_i - \hat{s}_i|^2} \tag{2}$$

For high signal-to-noise ratios, the upper-bound can be expressed as

$$P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \leq \frac{1}{(\frac{1}{4N_0})^{l_Q} d_p^2(l_Q)} \tag{3}$$

where $d_p^2(l_Q)$ is the squared product distance, given by

$$d_p^2(l_Q) = \prod_{i \in Q} |s_i - \hat{s}_i|^2 \tag{4}$$

In [4], the authors show that, for the case of Rayleigh fading channels with CSI, a tighter bound can be derived. Considering these results, a tight approximation to the *pairwise error probability* for high signal-to-noise ratios is given by

$$P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \approx \frac{(2L-1)!!}{2^{L+1}L!} \prod_{i=1}^{L} \frac{1}{1 + \frac{1}{4N_0}|s_i - \hat{s}_i|^2} \tag{5}$$

where $L = min(l_Q)$ is the effective distance of the code and

$$(2L-1)!! = (2L-1) \cdot (2L-3) \cdot ... \cdot 3 \cdot 1 \tag{6}$$

3.3. Estimations on the error event probability

An upper-bound on the error event probability, P_e, is obtained from the union bound [14]. The term with the smallest l_Q and $d_p^2(l_Q)$ dominates P_e

for high signal-to-noise ratios. Let $\gamma(l_Q, d_p^2(l_Q))$ be the average number of sequences having the effective length l_Q and the squared product distance $d_p^2(l_Q)$. By substituting $P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l)$ from equations (2), (3) and (5) in the following expression:

$$P_e \approx \gamma(L, d_p^2(L)) P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l) \tag{7}$$

three approximations to P_e can be obtained. Note that only the case $l_Q = L$ must be considered when using equations (2) and (3).

4. APPLICATION OF THE EIM

4.1. Hypotheses to apply the EIM

The EIM [10] provides an estimation of the Hamming minimum distance, d_{Hmin}, of a turbo code. It also provides an estimation of the average number of coded sequences at distance d_{Hmin}.

Our goal is to compute the approximations in equation (6) by using the information provided by the EIM. Because of bit-interleaving (π in Fig. 1), we can assume the following hypothesis:

Hyp. 1: $\forall i$, if $s_i \neq \hat{s}_i$ there is only one bit that changes between them.

As we consider high signal-to-noise ratios, the following hypothesis can also be assumed:

Hyp. 2: $\forall i$, if $s_i \neq \hat{s}_i$, s_i and $\hat{s}_i$ are adjacent signals in the constellation.

From **Hyp. 1**, it follows that $L = d_{Hmin}$. Denoting by d_0 the minimum Euclidean distance of the constellation, **Hyp. 2** can also be expressed as

$$|s_i - \hat{s}_i| = \begin{cases} 0 & \text{if } s_i = \hat{s}_i \\ d_0 & \text{if } s_i \neq \hat{s}_i \end{cases} \tag{8}$$

Then, the minimum squared product distance of the turbo-coded modulation can be calculated as follows

$$d_p^2(L) = (d_0^2)^{d_{Hmin}} \tag{9}$$

4.2. 8-PSK turbo-coded modulation

For 8-PSK modulation, d_0 and the average signal energy E_s are related by

$$d_0 = 2\sqrt{E_s} \sin \frac{\pi}{8} \tag{10}$$

For simplicity, we denote $\gamma(d_{Hmin}, d_p^2(d_{Hmin}))$ by γ. Taking equations (9) and (10) into account, we can now substitute $P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l)$ from equations (2), (3) and (5) in equation (7) to obtain three final expressions that estimate P_e for 8-PSK turbo-coded modulations. The first computable expression comes from (3):

$$P_e \approx \frac{\gamma}{(\frac{E_s}{N_0} sin^2 \frac{\pi}{8})^{d_{Hmin}}} \tag{11}$$

And by substituting $P_2(\mathbf{s}_l, \hat{\mathbf{s}}_l)$ from (2) and (5), we obtain

$$P_e \approx \frac{\gamma}{(1 + \frac{E_s}{N_0} sin^2 \frac{\pi}{8})^{d_{Hmin}}} \tag{12}$$

$$P_e \approx \frac{(2d_{Hmin} - 1)!!}{2^{d_{Hmin}+1} d_{Hmin}!} \frac{\gamma}{(1 + \frac{E_s}{N_0} sin^2 \frac{\pi}{8})^{d_{Hmin}}} \tag{13}$$

respectively.

5. EXAMPLES

Simulation results and estimations from equations (11)-(13) were obtained for different block sizes. The 8-state duo-binary turbo code adopted for the DVB-RCS/RCT standard [15] [16] was considered. We present in Figures 3 and 4 simulated vs. estimated performance over a Rayleigh slow-fading channel with perfect CSI. Estimations consider a memoryless channel model and assume coherent detection and ML decoding. Simulations use the Max-Log-MAP algorithm (with 8 iterations and 6-bit quantized samples). Figure 3 considers the transmission of 188-byte data blocks, and Figure 4 the transmission of 54-byte data blocks.

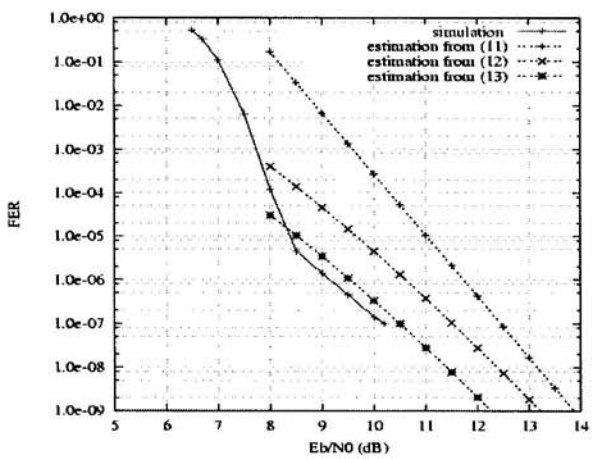


Figure 3: DVB-RCS turbo code. Rate=2/3. 8-PSK turbo-coded modulation. Transmission of 188-byte data blocks. Max-Log-MAP algorithm (with 8 iterations and 6-bit quantized samples).

As expected, the estimation from equation (13) is the closest of the three. For 188-byte data blocks, this estimation is only 0.3 dB from the simulation results, for FER values smaller than 10^{-6}. For 54-byte data blocks, the estimation is about 1 dB from the simulation results, for the same FER values. This

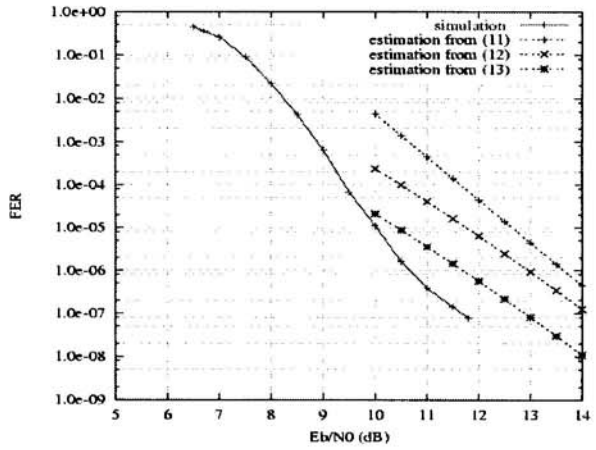


Figure 4: DVB-RCT turbo code. Rate=2/3. 8-PSK turbo-coded modulation. Transmission of 54-byte data blocks. Max-Log-MAP algorithm (with 8 iterations and 6-bit quantized samples).

may be because **Hypothesis 1** no longer holds when considering short data blocks. We also observe that the estimation from equation (11) is too far from the simulation results. This is probably due to the fact that this expression contains two high signal-to-noise ratio approximations.

6. CONCLUSIONS

A new solution to estimate asymptotic performance of turbo-coded modulation has been presented and justified. This solution relies on the application of the Error Impulse Method. The examples show good agreement between estimation and simulation curves. In addition, the estimation expressions enable us to better understand the parameters that dominate turbo-coded modulation performance in fading channels.

REFERENCES

[1] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 56-67, Jan. 1982.

[2] E. Biglieri, D. Divsalar, P.J. McLane, and M. K. Simon, "Introduction to Trellis-Coded Modulation with Applications", New York: Macmillan, 1991.

[3] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channels", *IEEE Trans. Commun.*, vol. COM-36, pp. 1004-1012, Sept. 1988.

[4] S. B. Slimane and T. Le-Ngoc, "Tight bounds on the error probability of coded modulation schemes in Rayleigh fading channels, *IEEE Trans. on Vehicular Technology*, Vol.44, No. 1, pp. 121-129, Feb. 1995.

[5] C. Tellambura, "Evaluation of the exact union bound for Trellis-Coded Modulations over fading channels", *IEEE Trans. Comm.*, vol. 47, No. 4, April 1999.

[6] S. Le Goff, A. Glavieux and C. Berrou, "Turbo-codes and high spectral efficiency modulation", *in Proc. IEEE Int. Conf. Commun.*, pp. 645-649, 1994.

[7] P. Robertson and T. Wörz, "Novel bandwidth efficient coding scheme employing turbo codes", *in Proc. IEEE Int. Conf. Commun.*, pp. 962–7, 1996.

[8] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Parallel concatenated trellis-coded modulation", *in Proc. IEEE Int. Conf. Commun.*, pp. 974–8, 1996.

[9] T. M. Duman and M. Salehi, "The union bound for turbo-coded modulation systems over fading channels", *IEEE Trans. Comm.*, vol. 47, No. 10, Oct. 1999.

[10] C. Berrou, S. Vaton, M. Jézéquel and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method", *Proc. GLOBECOM'02*, Taipei, Taiwan, Nov. 2002.

[11] C. Berrou, M. Jézéquel, C. Douillard and L. Conde, "Application of the error impulse method in the design of high-order turbo coded modulation", *Proc. Information Theory Workshop*, pp. 41-44, Bangalore, India, Oct. 2002.

[12] L. Conde Canencia, C. Douillard, M. Jézéquel, and C. Berrou, "Application of the error impulse method to 16-QAM bit-interleaved turbo coded modulations", *IEE Electronics Letters*, vol. 39, No. 6, pp. 538-9, 20th March 2003.

[13] G. Caire, G. Taricco and E. Biglieri, "Bit-interleaved coded modulation", *IEEE Trans. Inform. Theory*, vol. 44, pp. 927-946, May 1998.

[14] S. H. Jamali, T. Le-Ngoc, "Coded-modulation techniques for fading channels", Section 5.1.2, Boston, MA: Kluwer, 1994.

[15] ETSI EN 301 790, "Interaction channel for satellite distribution system", Dec. 2000, V1.2.2, pp. 21-24

[16] ETSI EN 301 958, "Interaction channel for digital terrestrial television", Aug. 2001, V1.1.1, pp. 28-30