



# Traitements temps réel en codage source et canal pour des Communications hertziennes et acoustiques sous-marines

André Goalic

## ► To cite this version:

André Goalic. Traitements temps réel en codage source et canal pour des Communications hertziennes et acoustiques sous-marines. Traitement du signal et de l'image. Université de Bretagne occidentale, 2006. <tel-00693927>

**HAL Id: tel-00693927**

**<https://tel.archives-ouvertes.fr/tel-00693927>**

Submitted on 3 May 2012

**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.

# **Habilitation à Diriger des Recherches**

présentée à

**L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE**

par

**André GOALIC**

**Traitements temps réel en codage  
source et canal pour des  
Communications hertziennes et  
acoustiques sous marines**

---

Travaux réalisés au sein du département  
Signal et Communications  
l'École Nationale Supérieure des Télécommunications de Bretagne



---

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Curriculum Vitae</b>	<b>9</b>
2.1	Expérience Professionnelle Enseignement . . . . .	9
2.1.1	I.U.T de LORIENT puis de QUIMPER . . . . .	9
2.1.2	I.U.T de QUIMPER . . . . .	9
2.1.3	1996 - 2005 : Maître de Conférences (192 TD) . . . . .	10
2.1.4	ENST-BRETAGNE (École Nationale Supérieure des Télécommunications de Bretagne) Département Signal et Communications Technopôle de BREST IROISE BP 832 29285 BREST CEDEX . . . . .	10
2.1.5	Texas A & M University Electrical Engineering Department College Station Tx 77843-3128, USA . . . . .	10
2.2	Expérience Professionnelle Recherche . . . . .	11
2.2.1	ENST-BRETAGNE (École Nationale Supérieure des Télécommunications de Bretagne) Département Signal et Communications (SC) Technopôle de BREST IROISE BP 832 29285 BREST CEDEX . . . . .	11
2.2.2	Texas A & M University Electrical Engineering Department College Station Tx 77843-3128, USA . . . . .	12
2.2.3	Formation . . . . .	12
2.3	Encadrement de Recherche . . . . .	12
2.3.1	Encadrement de Thèse 1999-2005 : . . . . .	12
2.3.2	Encadrement de Jeunes Chercheurs : . . . . .	12
<b>3</b>	<b>Codage de la Parole</b>	<b>15</b>

3.1	Introduction . . . . .	15
3.2	Modélisation du canal vocal . . . . .	17
3.2.1	Algorithme de Levinson éclaté . . . . .	17
3.2.2	Méthode de KABAL utilisant les polynômes de Chebyshev . . . . .	18
3.2.3	Comparaison des performances . . . . .	20
3.2.4	Quantification scalaire des CLSP . . . . .	20
3.3	Analyse long terme . . . . .	24
3.4	Dictionnaire de formes d'ondes d'excitation . . . . .	27
3.4.1	Analyse par synthèse . . . . .	27
3.4.2	Pondération perceptuelle du critère d'erreur . . . . .	29
3.4.3	Structure simplifiée . . . . .	31
3.5	Décodeur CELP . . . . .	32
3.5.1	Introduction . . . . .	32
3.5.2	Dictionnaire d'Excitation . . . . .	33
3.5.3	Reconstruction du Filtre Prédicteur long Terme . . . . .	33
3.5.4	Reconstruction du Filtre Prédicteur Court Terme . . . . .	33
3.6	Conclusions . . . . .	33
3.7	Publications . . . . .	35
<b>4</b>	<b>CODAGE DE CANAL</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Décodage itératif des codes produits . . . . .	38
4.3	Décodage des Turbo Codes en blocs BCH . . . . .	39
4.3.1	Décodage des Turbo Codes en blocs avec l'algorithme de Chase Pyndiah . . . . .	39
4.3.2	Décodage des Turbo Codes en blocs avec l'algorithme de Hartmann-Rudolph-Nazarov HRN . . . . .	45
4.4	Les Turbo Codes en Blocs de type RS Reed-Solomon . . . . .	52
4.4.1	Les Codes de Reed-Solomon . . . . .	52
4.4.2	Décodage à Entrée Souple des Codes RS par l'algorithme de Chase . . . . .	53
4.4.3	Les Turbo-Codes en Blocs Reed-Solomon . . . . .	53
4.5	Comparaison TCB-BCH et TCB-RS . . . . .	56
4.5.1	Conclusions . . . . .	57
<b>5</b>	<b>APPLICATIONS DES TURBO CODES EN BLOCS</b>	<b>59</b>
5.1	Introduction . . . . .	59
5.2	Réseau Indor - Canal 60 GHz . . . . .	60

5.2.1	Introduction . . . . .	60
5.2.2	Conclusions - Perspectives . . . . .	61
5.3	Turbo Codes pour BLUETOOTH . . . . .	61
5.3.1	Introduction . . . . .	61
5.3.2	Turbo Codes Programmables . . . . .	62
5.3.3	Performances obtenues . . . . .	63
5.3.4	Conclusion . . . . .	66
5.4	Étude des Nouvelles Modulations VLF/LF Codage de Canal . . . . .	66
5.4.1	Introduction . . . . .	66
5.4.2	Effet d'un Turbo Code . . . . .	67
5.4.3	Conclusions . . . . .	67
<b>6</b>	<b>Traitement Temps Réel</b>	<b>69</b>
6.1	Introduction . . . . .	69
6.2	Téléphonie acoustique sous-marine . . . . .	69
6.2.1	Introduction . . . . .	69
6.2.2	Codeur/décodeur CELP Ternaire (TCELP codeur/décodeur) . . .	70
6.2.3	Conclusions-Perspectives . . . . .	73
6.3	Banc d'essai de Turbo Codes . . . . .	73
6.3.1	Introduction . . . . .	73
6.3.2	Gestion de l'interface graphique . . . . .	74
6.4	Conclusions-Perspectives . . . . .	75
<b>7</b>	<b>Conclusion</b>	<b>79</b>
7.1	Publications . . . . .	81
7.2	Rapports de Recherche . . . . .	83
<b>A</b>		<b>85</b>
A.1	Real time low bit rate speech transmission through underwater acoustic channel . . . . .	85
A.2	Potential applications of low complexity non-binary high code rate block turbo codes . . . . .	85
A.3	Real-time turbo decoding of block turbo codes using the Hartmann-Nazarov algorithm on the DSP Texas TMS320C6201 . . . . .	85
A.4	Real-time turbo-decoding of product codes on a digital signal processor .	85
A.5	An intrinsically reliable and fast algorithm to compute the line spectrum pairs (LSP) in low bit rate CELP coding . . . . .	86

A.6 Toward a digital acoustic underwater phone . . . . .	86
--	----

# CHAPITRE 1 Introduction

---

les Sciences et technologies de l'Information et de la Communication (STIC) continuent à progresser de façon spectaculaire dans tous les domaines de l'ingénierie ou de la production, des secteurs les plus traditionnels jusqu'aux industries de pointe. Les technologies mettant en œuvre le traitement du signal et l'informatique permettant la mise au point de modèles de plus en plus réalistes, sont et seront à la base d'importantes innovations dans presque tous les secteurs de l'activité humaine : ingénierie, télécommunications, éducation, médecine, environnement, prévisions météorologiques, défense, transports, commerce et distribution, finance, loisirs, etc. Avec les outils de traitement et de modélisation, les STIC permettent également de nombreuses avancées dans les autres sciences : physique, chimie, biologie, sciences de la terre, sciences humaines ou sociales, etc. Dans les télécommunications, de nombreux domaines sont concernés par l'apport de ces nouveaux outils. Dans le domaine du traitement de la parole, les performances des codeurs ont tendance à se dégrader au fur et à mesure que leurs débits diminuent, même si leurs conceptions font appel à des modèles de plus en plus sophistiqués. Le choix d'un codeur de parole est un compromis entre différents facteurs : qualité souhaitée, débit et complexité. En téléphonie acoustique numérique sous-marine, la capacité du canal dépend de la portée envisagée. Une réduction importante du débit est nécessaire pour atteindre des distances acceptables. On fera appel à des codeurs bas, voir très bas débits suivant la qualité de parole souhaitée.

La montée en puissance des processeurs de signaux (Digital Signal Processor DSP), une puissance multipliée, par un facteur de l'ordre de 1000 dans les quinze dernières années, permet d'offrir aujourd'hui une large gamme de choix pouvant répondre à la plupart des situations rencontrées dans le domaine industriel. Après les codeurs temporels à hauts débits type MIC (64kbits/s), MICDA DECT(32kbits/s), MIC bande élargie(64kbits/s), les codeurs mixtes (paramétriques et temporels) à bas débits de type CELP<sup>1</sup> (4.8kbits/s) ou MELP<sup>2</sup> (2.4kbits/s), avec leurs nouveaux algorithmes de compression offrent de très bonne qualité de parole. La première partie de mes travaux de recherche a été consacrée au codage de source avec l'étude et le développement d'un codeur de type CELP (5, 45kbits/s) et à son implémentation sur un DSP<sup>3</sup>. Compte tenu des contraintes temps réel, l'intégration a été entièrement réalisée en langage d'assemblage, elle est présentée au Chapitre 3.

---

<sup>1</sup>Code Excited Linear Prediction

<sup>2</sup>Mixed Excited Linear Prediction

<sup>3</sup>de type Motorola fréquence d'horloge 27 Mhz, 13.5 MIPS



Dans le domaine des communications numériques, après transmission sur un canal physique, le flux de données est entaché d'erreurs provenant de phénomènes d'évanouissements, d'interférences entre symboles, de bruits d'origine diverses (thermique, milieu, etc.). En codage canal, l'arrivée des systèmes itératifs, connus sous le nom de "Turbo Codes" [2], [20], est à l'origine de nombreuses avancées permettant de récupérer l'information émise quasiment sans erreur dans les conditions de fonctionnement. Le codage de canal permet ainsi en réception, après égalisation éventuelle et démodulation, de corriger les erreurs de parcours avec un pourcentage de réussite très élevé. Les chapitre 4 et 5 présentent les activités de recherche que j'ai eu l'occasion de mener depuis l'obtention de ma thèse en 1994. Le chapitre 6 est plus particulièrement consacré à la présentation d'une approche des activités temps réel. Ce chapitre nous permet de présenter la téléphonie acoustique sous-marine et notamment les différentes plateformes mises en œuvre au cours des essais de 1994 en rade de Brest [8] et ceux de 2003 dans la même rade de Brest [11]. Un plate forme temps réel d'essais de Turbo codes en blocs est également présentée, son objectif est plus pédagogique. Une sélection d'articles publiés en congrès est jointe en annexe.

# CHAPITRE **2** Curriculum Vitae

---

## **2.1 Expérience Professionnelle Enseignement**

### **2.1.1 I.U.T de LORIENT puis de QUIMPER**

- 1975 - 1983 : Ingénieur Contractuel (384 heures/année)
- Automatique, Électricité, Électrotechnique et Mesures Physiques : Cours, Travaux dirigés, Travaux pratiques et encadrements de projets Création des laboratoires de Physique Industrielle de 1ère et 2ème année

### **2.1.2 I.U.T de QUIMPER**

1983 - 1996 : Assistant (192 TD) <sup>1</sup>

61ème Section : Génie informatique, Automatique et Traitement du Signal

- Automatique (Cours, TD, TP et encadrements de projets ) :
  - applications Agro-alimentaires sur Automates Programmables Industriels.
  - modélisations de processus.
- Informatique, Informatique industrielle (TD, TP, et encadrements de projets) :
  - création module d'enseignement en langage PASCAL
  - création module d'enseignement d'Informatique Industrielle sur Moniteur Multi-tâches temps réel.
- Thermique Industrielle (Cours, TD, TP, et encadrements de projets) :
  - Module d'enseignement de Froid Industriel.
  - Banc d'essai de machines thermiques : groupe frigorifique, pompe à chaleur.
- Électrotechnique (Cours, TD, TP, et encadrements de projets) :
  - Module d'enseignement : Courants Triphasés, Machines tournantes

1986-1989 en partenariat avec :

- l'ADEPA (Agence pour le Développement de la Production Automatisée) de RENNES
- La C.C.I (Chambre de Commerce et d'Industrie) de QUIMPER.

spécialisation Post D.U.T-B.T.S (30 heures) en :

- Automatique et Informatique Industrielle pour l'Agroalimentaire

---

<sup>1</sup>Pendant la période 1983-1989 ma charge horaire effective a atteint plus du double (un poste en automatique non pourvu et inexistance de poste de technicien dans la matière)

- Systèmes Asservis et applications aux régulations de températures, niveaux, pressions etc.

### 2.1.3 1996 - 2005 : Maître de Conférences (192 TD)

61ème Section : Génie informatique, Automatique et Traitement du Signal

- Automatique (Cours, TD, TP et encadrements de projets ) - Industrie Agroalimentaire 2ème Année
- Informatique Industrielle (TD, TP et encadrements de projets)
- Thermique Industrielle (Cours Thermique Industrielle, Régulation) Diplôme National de Technologie Spécialisé (DNTS) post DUT-BTS

### 2.1.4 ENST-BRETAGNE (École Nationale Supérieure des Télécommunications de Bretagne) Département Signal et Communications Technopôle de BREST IROISE BP 832 29285 BREST CEDEX

1994 - 2005

- Codage de la parole (Cours) :
  - Formation initiale cycle ingénieur ENST : 6 heures
  - Formation initiale cycle ingénieur ENSAI Bruz : 6 heures
  - Formation continue extérieure : Modules de 6 ou 2 heures Motorola Toulouse, Matra Kemper, Alcatel Strasbourg, Lucent Technology Lannion
- Processeurs de signaux en traitement du signal :
  - Formation initiale 3ème Année 12 heures
  - Formation continue DSP : moniteur multi-tâches (BIOS Built In Operating System) 12 heures
- Turbo Codes (Cours, TP) : 4 heures
- Structure d'un système de communications numériques : 12 heures
- Encadrements : élèves ingénieurs, stagiaires D.E.A, doctorant.

### 2.1.5 Texas A & M University Electrical Engineering Department College Station Tx 77843-3128, USA

janvier-juin 2000 (Spring Semester) : Séjour sabbatique Visiting Professor

- Digital Signal Processing (Course, Homeworks<sup>2</sup>) "Undergraduated Course"  
Dernière année du cycle Ingénieur, 44 heures.

---

<sup>2</sup>préparation à la charge du "Professor", 12 homeworks/semester

- Encadrement projet étudiant : Digital Signal Processor, "Graduated Course" : Digital Signal processor

## 2.2 Expérience Professionnelle Recherche

### 2.2.1 ENST-BRETAGNE (École Nationale Supérieure des Télécommunications de Bretagne) Département Signal et Communications (SC) Technopôle de BREST IROISE BP 832 29285 BREST CEDEX

1990-1994 : Doctorant<sup>3</sup>

Domaines d'activité :

- compression de l'information, codage de la parole à bas débit,
- codeur à prédiction linéaire excité par code (Code Excited Linear Prediction CELP),
- modélisation de fonction de transfert, conditionnement temps réel d'algorithme sur processeurs de signaux DSP,
- téléphonie acoustique sous-marine,
- adaptation du codeur aux caractéristiques statistiques des signaux de parole de plongeurs, (enregistrement des signaux plongeurs en baie de Concarneau)

23 juin 1994 : Doctorat d'Électronique

"Étude et Réalisation de Codeur/Décodeur de Parole à bas Débit pour la Téléphonie Acoustique Sous-Marine"

1994 - 2004 : Chercheur Associé<sup>4</sup>, département SC

- Traitement de la Parole :
  - codage moyen débit, bas délai,
  - filtrage adaptatif avec utilisation des paires de raies spectrales (Line Spectrum Pairs),
  - téléphonie acoustique sous-marine,
- Codage Canal :
  - codage correcteur d'erreurs : codes convolutifs, codes Bose-Chaudhuri-Hocquenghem (BCH), codes Reed Solomon (RS),
  - turbo code en bloc : codes élémentaires BCH et RS,
  - turbo décodage de Codes Produits : Algorithme Chase-Pyndiah (BCH), Hartmann-Rudolph-Nazarov (BCH) et Berlekamp-Massey (RS),
- Conditionnements temps réel et implémentations sur Processeurs de signaux DSP :
  - turbo décodage de codes en blocs en temps réel sur processeurs de signaux,
  - banc d'essai de codeur/décodeur de parole sur processeurs de signaux,

---

<sup>3</sup>séjour sabbatique à l'ENST-Bretagne janvier-juin 1991

<sup>4</sup>séjour sabbatique à l'ENST-Bretagne janvier-juin 2004

## **2.2.2 Texas A $\delta$ M University**

### **Electrical Engineering Department**

### **College Station Tx 77843-3128, USA**

janvier-juin 2000 : Séjour sabbatique au TEXAS USA

Turbo-Coding :

- Banc d’essai temps réel de mesure des performances en terme de taux d’erreurs binaires des Turbo-codes en bloc de type BCH,
- mise en œuvre de l’assembleur linéaire.

## **2.2.3 Formation**

- 1973 : Ingénieur Physicien UBO, 1972 : Maîtrise de Physique U.B.O
- Stages :
  - 1972 IFREMER : Traitements de signaux sismiques par transformation de Fourier
  - 1975 THOMSON BREST : Essai de matériels en vibration - Déverminage
- Langues parlées : Breton, Français, Anglais
- Divers :
  - Activités culturelles : langue et culture bretonne : enseignement en cours du soir et stages
  - Développement d’alternative énergétique en Bretagne : filière solaire, éolienne, biogaz
  - Organisation de salons Énergie et Environnement.

## **2.3 Encadrement de Recherche**

### **2.3.1 Encadrement de Thèse 1999-2005 :**

1. Septembre 1997- juin 2002 : N. Chapalain, Application des Turbo Codes en blocs pour les réseaux locaux sans fil à haut débit(50%)\*
2. Novembre 2001-avril 2005 : Rong Zhou, Etude des Turbo Codes en Blocs Reed-Solomon et leurs Applications(50%)\*

### **2.3.2 Encadrement de Jeunes Chercheurs :**

1. O. Grosbras et D. Sion "Real-time Turbo-decoding of Product codes on the Texas DSP TMS320C6201", Projet de recherche ENSTB 1999 (100%)\*
2. Rajesh Gopinath, "Real time BCH Block Turbo Decoding on the TMS320C62xx Digital Signal Processor", Term project ELEN 689 ; C6x Signal Processing Spring 2000 TA  $\delta$  MU, College Station,Tx USA. (100%)\*
3. W. Tapsoba, "Use of fast Hadamard transform for Block Turbo Codes Decoding and Real Time Implementation on a Texas TMS320C6201 Platform", Projet de recherché

- Master International in Télécommunications ENST-Bretagne et Texas Instruments DSP and Analog Challenge 2001. (100%)\*
4. L. Omarjee et P. Fezelot, "Annulation successive d'Interférence et Codage Correcteur d'erreur pour le CDMA", projet de recherche 3ème année ENSTB 2002. (50%)\*
  5. O. Orhouma, "Comparaison des Turbo Codes BCH et des Codes RS sur un canal de Gauss et sur des canaux binaires et ternaires ", stage fin d'étude ENIT TUNIS, Tunisie ENSTB 25/02/02-30/06/02. (100%)\*
  6. L. Akrimi, E. Gabarron, G.. Galataud, Y. Ge, P. Koenig, S. Pasqualini et N. Wimmer, "Démonstrateur temps réel de turbo-décodage de codes en blocs sur plate forme TI TMS320C6201", janvier-juin 2002. (75%)\*
  7. G. Farage et E. Royuela Romero, "Performances Optimization of a block turbo-decoder and real time implementation on DSP TI TMS320C6201", Projet de recherche Mastère ENSTB 2003. (100%)\*
  8. D. Singhal and A. Karym, "Implementation of Berlekamp Algorithm on TMS320C62x DSP for decoding BCH(127, 92, 5) Code", Projet de recherche ENSTB 2004. (50%)\*
  9. Ali Al Ghouwayel, "Codage de canal en Acoustique sous-marine ", DEA UBO, ENSTB 01/03/04-30/06/04. (50%)\*
  10. Samar Changuel, "Codage/décodage des codes de Reed Solomon par l'algorithme de Guruswani-Sudan", DEA UBO, ENSTB 01/03/04-30/06/04 (50%)\*
  11. M. Grima, F. Fieau, B. Jouari, A El Mowahid et J.J. Wang, "Implémentation en temps réel d'un codeur/décodeur de parole sur une plate-forme DSP TMS320C6211", Projet d'ingénieur ENSTB janvier-juin 2004. (50%)\*
  12. Javier Fernandez, "Codage canal en acoustique sous-marine", Projet de Master, Avril-Septembre 2005 (50%)\*
  13. L. Diquelou, "Implémentation en temps réel d'un codeur/décodeur de parole de type Melp sur une plate-forme DSP TMS320C6211", Projet d'ingénieur ENSTB janvier-juin 2005. (50%)\*
  14. Laurent Diquelou, "Implementation temps réel d'un codeur de parole à bas débit de type Melp", stage ingénieur Juillet-Août 2005 (50%)\*.

(\*) taux d'encadrement



# CHAPITRE 3 Codage de la Parole

## 3.1 Introduction

En téléphonie acoustique numérique sous-marine, la capacité du canal dépend de la portée envisagée. Une réduction importante du débit est nécessaire pour atteindre des distances acceptables. Le codeur de type CELP permet de fournir un débit de l'ordre de 6 kbit/s compatible avec les exigences canal. Malgré le surcroît de complexité, les processeurs de signaux permettent la mise au point de systèmes fonctionnant en temps réel et de qualité compatible avec l'application [8], les techniques analogiques n'assurent qu'un faible confort d'écoute.

Le codeur/décodeur de parole utilisé est un système de type CELP (Code Excited Linear Prediction). Il appartient à la famille des codeurs dits mixtes dans la mesure où il associe du codage à la fois temporel et paramétrique, permettant de descendre à de très bas débits de codage. Ce type de codeur a été proposé en 1985 par Atal et Schroeder [2]. Ils comportent généralement trois parties principales (Fig. 3.1) :

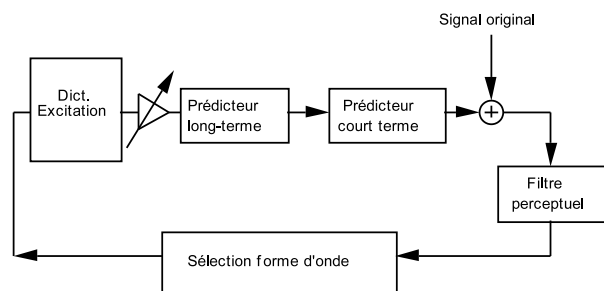


FIG. 3.1 — Codeur CELP

- **Analyse court terme** : elle permet de définir un filtre dont la fonction de transfert modélise le conduit vocal. Cette fonction de transfert est modélisée par un filtre tous pôles (3.1),  $p$  étant l'ordre de prédiction :

$$H(z) = \frac{1}{A_p(z)} \quad (3.1)$$

Le polynôme  $A_p(z)$  peut s'exprimer à partir de deux fonctions de transfert (3.2) correspondant respectivement à une ouverture  $P_{p+1}(z)$  et une fermeture  $P_{p+1}^*(z)$



complète de la glotte.

$$A_p(z) = \frac{P_{p+1}(z) + P_{p+1}^*(z)}{2} \quad (3.2)$$

Les polynômes  $P_{p+1}(z)$  et  $P_{p+1}^*(z)$  sont respectivement symétrique et anti-symétrique et possèdent des propriétés très intéressantes :

1. pour un filtre stable, toutes les racines de  $P_{p+1}(z)$  et de  $P_{p+1}^*(z)$  sont sur le cercle unité.
2. les racines de 2 polynômes, symétriques et anti-symétriques s'alternent 2 à 2 sur le cercle unité.

Le signal de parole, filtré par l'inverse de la fonction précédente, permet d'extraire les redondances court terme du signal vocal. Le signal obtenu correspond au signal glottique. Plusieurs méthodes ont été introduites pour définir la fonction de transfert du canal. La caractérisation du canal vocal par les paires de raies spectrales (Line Spectrum Pairs : LSP) [12], [22] permet d'obtenir une meilleure qualité de parole de synthèse à bas débit [23]. L'extraction des LSP a également fait l'objet de nombreux travaux. La méthode utilisant une seule version de l'algorithme de LEVINSON éclaté conduit à une recherche rapide et intrinsèquement fiable des racines des polynômes prédicteurs singuliers [23]. L'algorithme de LEVINSON éclaté est détaillé avant la présentation d'une autre méthode due à KABAL et al [13], dérivant de l'algorithme de DURBIN et utilisant les polynômes de CHEBYSHEV. Les deux méthodes, conduisant aux mêmes résultats, sont donc équivalentes d'un point de vue mathématique.

- **Analyse long terme** : elle conduit à la définition d'un filtre dont la fonction de transfert modélise les redondances de la source vocale. Comme dans le cas de l'analyse court terme le filtrage inverse extrait les redondances long terme du signal. Après filtrage inverse court et long terme, le signal résiduel est pratiquement un bruit blanc. Un algorithme simplifié d'extraction des paramètres long terme est présenté. Appelé "Correlation Peak-Picking method", il aboutit à une détermination sous-optimale du pitch et du facteur de voisement.
- **Dictionnaire de formes d'ondes d'excitation** : le signal original, débarrassé de ses redondances court et long terme, donne un signal résiduel. Pour prendre en compte les limites des deux filtrages précédents, le codeur CELP modélise ce résidu par une forme d'onde choisie dans un dictionnaire, suivant un critère d'erreur énergétique incluant les caractéristiques physiologiques de l'oreille humaine [3]. De nombreux types de dictionnaire ont été proposés (gaussiens, stochastiques, algébriques, combinaisons linéaires de vecteurs indépendants...) [1], [24], [14]. Les contraintes temps réel et l'application envisagée (téléphonie sous-marine) nous ont conduit à utiliser un dictionnaire algébrique ternaire [14], [4]. L'objectif est d'obtenir une parole synthétisée de qualité meilleure que les systèmes analogiques actuels dont le confort d'écoute n'est pas satisfaisant.

La liste des paramètres, avec leurs débits binaires, est présentée sur le tableau (Tab. 3.1). Le codeur extrait donc, au cours des différentes analyses dont la fréquence dépend du traitement effectué, les paramètres suivants :

- le cosinus des paires de raies spectrales au nombre de 10 (CLSP) : c'est l'analyse court terme destinée à modéliser le conduit vocal. Ce traitement s'effectue une fois

Paramètres	Trame(ms)	Bits/trame	Débit(bits/s)
CLSP(10)	20	30 x 1	1500
Glpc	20	7 x 1	350
b ltp	10	3 x 2	300
Pitch	10	7 x 2	700
Gq	5	3 x 4	600
I index	5	10 x 4	2000
	TOTAL	109	5450

TAB. 3.1 — Paramètres et allocation de bits

- par trame.
- le gain syllabique  $G_{lpc}$ , également renouvelé une fois par trame.
  - le pitch  $P$  et le coefficient de prédiction long terme  $b$  : c'est l'analyse long terme destinée à modéliser les redondances de la source vocale. Ce traitement s'effectue deux fois par trame.
  - l'indice de la forme d'onde et son gain associé : c'est la modélisation du signal d'excitation. Ce traitement s'effectue quatre fois par trame.

## 3.2 Modélisation du canal vocal

Le signal de parole est segmenté en trames de 20 ms. Préalablement au processus d'extraction des paires de raies spectrales, les étapes suivantes sont à réaliser :

- **acquisition** : le signal de parole est échantillonné à la fréquence de 8 KHz. Chaque échantillon est codé sur 16 bits par un Convertisseur Analogique Numérique (CAN). Un buffer circulaire d'une capacité de 320 échantillons permet d'assurer le traitement temps réel. Une option DMA<sup>1</sup> est mise en œuvre ainsi qu'une stratégie de gestion de buffer de type "Ping-Pong".
- **fenêtre de hamming-Recouvrement inter trames** : pour assurer une continuité de l'analyse, on procède à un recouvrement inter-trame. L'analyse LPC s'effectue sur la trame indice  $n$  à laquelle on ajoute les 32 derniers échantillons de la trame  $n - 1$  et les 32 premiers de la trame  $n + 1$ , soit en tout 224 échantillons.

### 3.2.1 Algorithme de Levinson éclaté

C'est la version anti-symétrique de l'algorithme qui est ici mis en œuvre [9]. Il est évidemment possible de traduire directement cet algorithme en assembleur mais le code obtenu ne serait optimisé ni en temps ni en occupation mémoire. Une connaissance approfondie de l'algorithme et notamment de la durée de vie des variables permet de réduire au minimum le besoin en stockage de paramètres internes. En résumé, l'algorithme reçoit les 11 valeurs de la fonction d'autocorrélation  $c_i$  et restitue les 20 valeurs intermédiaires, immédiatement utilisés par la suite pour définir les deux matrices dont les valeurs propres

<sup>1</sup>Direct Memory Access

donnent les Paires de Raies Spectrales. Les calculs sont également réalisés en format mixte.

**extraction des valeurs propres - méthode de la bisection** : les valeurs propres  $x_i$  de ces matrices sont donc des expressions de la forme  $x_i = 2\cos\omega_i$ , les  $\omega_i$  étant les paires de raies spectrales (LSP). Dans la pratique, nous nous arrêtons à la recherche de ces fonctions cosinus (CLSP), et ce pour deux raisons principales. La première : le processeur ne comportant pas de fonction de haut niveau, l'extraction des angles exigerait l'emploi de développements limités incompatible avec un traitement temps réel. La seconde tient au fait que la fonction de transfert du conduit vocal s'exprime en utilisant le cosinus des paires de raies spectrales (CLSP). La méthode de la bisection apporte une solution efficace et d'une fiabilité intrinsèque, pour la recherche des valeurs propres dans un processus temps réel. Elle permet :

- de connaître le nombre de valeurs propres contenues dans un intervalle et, par un processus itératif, d'isoler celui qui contient la  $i^{ieme}$  valeur.
- d'approcher, avec la précision souhaitée sur cet intervalle, la  $i^{ieme}$  valeur propre  $x_i$

L'algorithme de Levinson éclaté, version anti-symétrique, conduit donc à l'utilisation d'une méthode très intéressante par sa fiabilité intrinsèque pour la recherche de paramètres, dans le cadre d'un processus temps réel. Les valeurs propres sont déterminées avec une précision de  $2^{-7} = 0.0078125$ , ce qui donne une résolution en fréquence allant de 5 Hz, pour le milieu du spectre, à 20 Hz pour les plus basses et plus hautes fréquences du signal de parole. Le tableau (Tab. 3.2) présente, pour notre base de données les distances minimales entre deux CLSP consécutifs et entre deux CLSP consécutifs de même parité.

CLSP	Distance minimale	Distance maximale
CLSP paires	0.03125	1.194
CLSP impaires	0.0306	1.1895
CLSP	0.001465	1.069

TAB. 3.2 — Distances entre les CLSP

### 3.2.2 Méthode de KABAL utilisant les polynômes de Chebyshev

La méthode de KABAL [13], également très utilisée pour le calcul des CLSP, peut paraître, dans une première approche, plus simple à mettre en oeuvre. Malheureusement, cette méthode ne présente pas les mêmes avantages que celle qui utilise l'algorithme de LEVINSON éclaté. Elle présente d'autre part une redondance intrinsèque dans la mesure où elle entraîne la détermination des coefficients de prédiction  $\{a_i\}_{i=1..p}$  préalablement à la détermination des paires de raies spectrales. Les opérations préliminaires (acquisition, fenêtrage par Hamming, autocorrélation) étant rigoureusement les mêmes, KABAL et

al passent par l'algorithme de DURBIN puis par les polynômes de CHEBYSHEV pour obtenir les paires de raies spectrales.

**Algorithme de DURBIN** la version de base de cet algorithme calcule les coefficients PARCOR  $\{k_i\}_{i=1,\dots,p}$ , ceux-ci présentant l'avantage d'être bornés à un pour l'implémentation sur un processeur à virgule fixe. L'algorithme utilisé calcule simultanément les coefficients  $\{a_i\}_{i=1,\dots,p}$  qui sont également bornés mais ces bornes dépendent de l'ordre de prédiction et leurs valeurs absolues théoriques dépassent, comme pour les paramètres de l'algorithme de LEVINSON éclaté, la valeur de 128. Pour un ordre de prédiction  $p$ , la valeur absolue du  $i^{\text{ème}}$  coefficient de prédiction  $|a_{i,p}|$  est bornée par la valeur suivante :

$$|a_{i,p}| \leq \frac{p!}{i!(p-i)!} \quad (3.3)$$

Pour permettre le conditionnement de ces paramètres, l'étude statistique de ces coefficients a été réalisée sur notre base de données. Le tableau (Tab. 3.3) donne pour les paramètres LPC les limites théoriques et pratiques.

Paramètres	$a_i \quad i = 1, \dots, p$	
Limites théoriques	Mini	-252
	Maxi	+252
Limites pratiques	Mini	-6.877
	Maxi	7.406

**TAB. 3.3** — limites théoriques et pratiques des coefficients LPC

Après le calcul des coefficients de prédiction, l'algorithme détermine les coefficients  $\{c_k\}_{k=0,\dots,\frac{p}{2}}$  et  $\{c_k^*\}_{k=0,\dots,\frac{p}{2}}$  des polynômes prédicteurs  $R_{p+1}(w)$  et  $R_{p+1}^*(w)$  [9], les racines de ces polynômes étant les CLSP. Ces polynômes s'exprimant en fonction des cosinus des paires de raies spectrales, leurs calculs sur un processeur de signal dans une optique temps réel posent un problème majeur dans la mesure où il n'y a pas de fonctions de haut niveau sur de tels systèmes. KABAL a proposé la solution qui tient compte des propriétés des polynômes de CHEBYSHEV. Compte tenu de la relation de récurrence, le calcul se ramène à un calcul récursif à partir des coefficients  $c_k$  et  $c_k^*$ . Ces derniers dérivant des coefficients LPC sont également bornés. Les valeurs obtenues à partir de notre base de données sont les suivantes :

$$-5.5189 \leq c_k, c_k^* \leq +5.829 \quad (3.4)$$

La suite du calcul consiste simplement à extraire les racines des deux polynômes  $R_{p+1}(w)$  et  $R_{p+1}^*(w)$ . La méthode ici n'est pas aussi simple que dans le cas de la méthode

précédente car on n'a pas accès directement à la  $i^{\text{ème}}$  ( $i = 0, \dots, 9$ ) racine de l'un ou l'autre des polynômes, La recherche ici s'effectue sur le segment  $]-1, \dots, +1[$  par une méthode dichotomique. Pour celle-ci un certain nombre de contraintes sont à prendre en compte :

- les racines des deux polynômes sont entrelacées sur l'intervalle. L'algorithme devra donc, dès qu'une racine de l'un des polynômes est localisée, continuer la recherche sur l'autre et ainsi de suite jusqu'à la localisation des  $p$  racines (10).
- s'agissant d'une méthode dichotomique, le ou les pas d'incrémentations [13] doivent être choisis de telle sorte que toutes les racines puissent être détectées (sous peine de divergence). Il faut donc prendre en compte la distance minimale  $\delta$  entre deux CLSP consécutifs :

$$\delta < \text{Min}(x_i - x_{i-1}) \quad (3.5)$$

- pour respecter les contraintes temps réel, le choix d'un seul pas d'incrémentations (d'une longueur inférieure à l'écart  $\delta$ ) n'est pas envisageable. Un premier pas de longueur inférieure à la distance minimale  $\Delta$  (3.6), entre les CLSP de même parité, est utilisé pour localiser grossièrement les racines, puis le second  $\delta$  sert à affiner la précision de détermination des CLSP.

$$\Delta < \text{Min}(x_i - x_{i-2}) \quad (3.6)$$

### 3.2.3 Comparaison des performances

L'ensemble des tests de performance, concernant le temps réel, ont été réalisés sur le banc d'essai [6] [4] (Voir Chap.VI parag. 6.2). Les deux méthodes ont été testées sur le même fichier de parole, elles conduisent rigoureusement aux mêmes valeurs pour les CLSP. Suivant les valeurs que l'on prend pour  $\Delta$  et  $\delta$ , la divergence a cependant été constatée pour des valeurs trop grandes de  $\Delta$  ou de  $\delta$ . Il est donc primordial pour la méthode de KABAL, utilisant les polynômes de CHEBYSHEV, de prendre de grandes précautions pour les pas d'incrémentations. L'utilisation de faibles pas augmente du même coup le nombre d'évaluation des polynômes. Les résultats présentés (tableau 5) tiennent compte des valeurs du tableau 2 pour le choix de  $\Delta$  et  $\delta$  :

- Méthode de LEVINSON éclaté : précision de détermination 0.0078.
- Méthode KABAL :  $\Delta = 0.015625$   
 $\delta = 0.000975$

On s'aperçoit que la méthode de KABAL paie en temps de traitement le fait que la recherche des CLSP dépend de la statistique du signal, à savoir de l'écart entre les différents CLSP. La surcharge de traitement est d'environ 5%. Cette valeur prend de l'importance si l'on compare les temps (1 ms soit 47% de plus pour la méthode de KABAL pour extraire les CLSP). A noter également que, pour les deux méthodes, le temps de traitement de trame varie très peu.

### 3.2.4 Quantification scalaire des CLSP

La détermination des coefficients CLSP (cosinus des paires de raies spectrales) est immédiatement suivie par leurs quantifications. De nombreuses méthodes ont vu le jour

Algorithme	Analyse LPC (en ms)	Extraction CLSP seule (en ms)
LEVINSON éclaté	2.12 (10.6%)	1.62 (8.1%)
KABAL Chebyshev	3.12 (15.6%)	2.62 (13.1%)

TAB. 3.4 — Extraction CLSP (temps de traitement)

pour ce faire, le but étant de diminuer le bruit de quantification. Celle qui a été retenue est la quantification non uniforme de LLOYD-MAX [19], [20]. Sa mise en oeuvre passe par la connaissance des densités de probabilités des paramètres à quantifier. Ces densités ont donc été évaluées dans une base de données constituée à partir de signaux de parole de plongeurs.

– **Densité de probabilité des Paires de Raies Spectrales**

Les paramètres CLSP permettent d'évaluer la fonction de transfert du conduit vocal, à la fois au codeur et au décodeur. Avant transmission ou stockage, ces paramètres doivent être quantifiés et codés. Pour réduire l'importance de l'erreur de quantification, il convient d'adapter la loi de quantification à la densité de probabilité des coefficients CLSP. Un nouvel estimateur non paramétrique de la densité de probabilité introduit par S. SAOUDI et al [23], est utilisé pour estimer les densités de probabilités  $f_{x_i}(x)$  des  $x_i$ , valeurs propres des matrices tridiagonales [9], valeurs propres représentant, à un facteur constant près, le cosinus des paires de raies spectrales.

– **Quantification des CLSP**

Les CLSP sont codés sur 30 bits. L'ensemble des paramètres relatifs au processus de quantification - régions de décision (au nombre de 70), niveaux de quantification (au nombre de 80) - est préalablement stocké en mémoire. Les niveaux de quantification sont issus d'un quantificateur non uniforme de LLOYD-MAX [19], [20]. L'algorithme de quantification assure par ailleurs le maintien de la relation d'ordre sur les CLSP pour assurer la stabilité du filtre court terme. L'algorithme vérifie et éventuellement modifie le niveau de quantification du  $i^{\text{ème}}$  CLSP en le comparant à celui du  $(i - 1)^{\text{ème}}$ , de manière à ce que la condition de stabilité du filtre soit respectée.

– **Signaux de parole plongeurs**

Notre base de données se compose d'un certain nombre de phrases représentant 32604 trames de 20 ms (10.868 mn). Le microphone est placé dans le masque du plongeur et relié par une liaison filaire à notre système d'acquisition (magnétophone numérique de type DAT). La fréquence d'échantillonnage de 48 kHz correspond à celle du DAT, chaque échantillon est codé sur 16 bits. L'objet de cet enregistrement est d'adapter notre codeur de parole bas débit aux caractéristiques spectrales des signaux. Préablement à tous traitements, les signaux sont filtrés avec un filtre

passé-bande 300..3400 Hz. Les figures (Fig. 3.2) et (Fig. 3.3) présentent les densités de probabilité des CLSP pour les signaux de parole de plongeurs et de plongeuses (notre seconde base de données de signaux féminins ne comporte à ce jour qu'une seule locutrice : 2000 trames). La comparaison avec les signaux téléphoniques

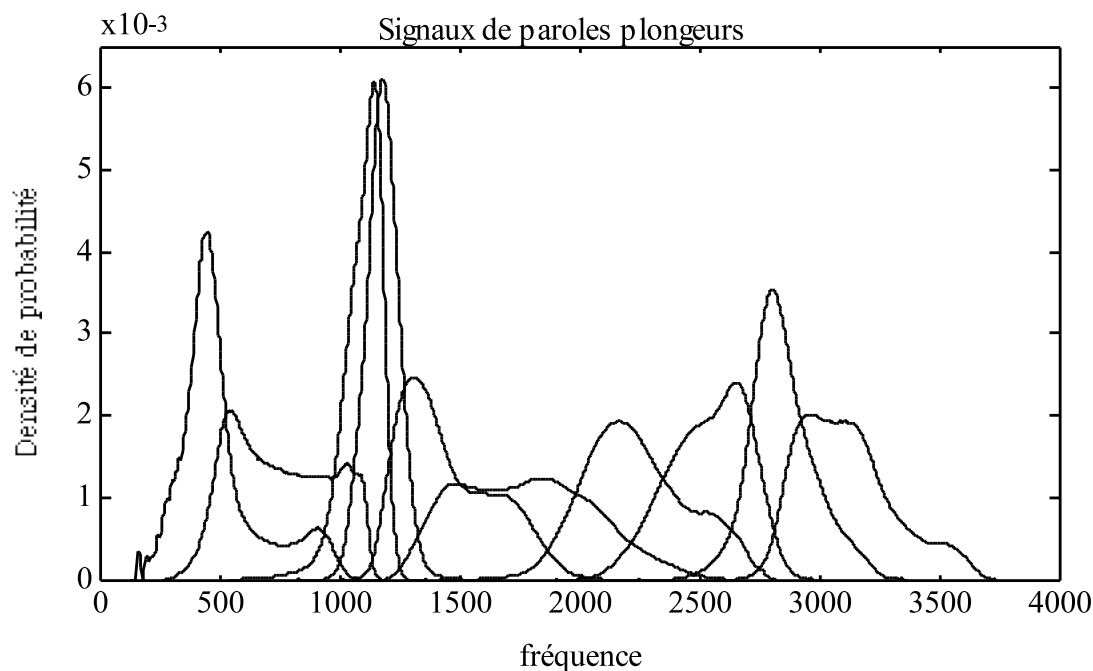


FIG. 3.2 — Densité de probabilité des CLSP signaux plongeurs

classiques (Fig. 3.4) laisse apparaître une différence importante en ce qui concerne les 3<sup>ème</sup> et 4<sup>ème</sup> CLSP, principalement dans la gamme de fréquence 1000..1500 Hz, différence liée à l'effet de bulle lié au sifflement du détendeur. La figure (Fig. 3.5) présente le spectre de puissance d'un enregistrement de bruit de bulles (2.5 sec). L'importance de la raie au delà de 1000 Hz explique le pic pour les 3<sup>ème</sup> et 4<sup>ème</sup> CLSP.

Les coefficients CLSP sont les premiers paramètres que nous transmettons au décodeur. Ils contiennent une information de type spectrale dans la mesure où ils permettent de modéliser la fonction de transfert du conduit vocal. Le formant ou la résonance du conduit vocal sera d'autant plus accentué que deux CLSP seront plus proches l'un de l'autre. Une fois transmis le paramètre suivant, le gain syllabique  $G_{lpc}$  comporte une information sur le niveau énergétique du signal. Sa détermination passe par deux étapes, la première le calcul du filtre prédicteur, permettant l'extraction des redondances court terme du signal, et la seconde la détermination des échantillons du signal glottique, correspondant à l'erreur de prédiction court terme. C'est l'écart type de ce signal qui est quantifié et transmis au décodeur, cela permet de prendre en compte les erreurs de quantification court terme. L'algorithme de LEVINSON permet également de calculer la variance de l'erreur, donc l'écart type, avant quantification.

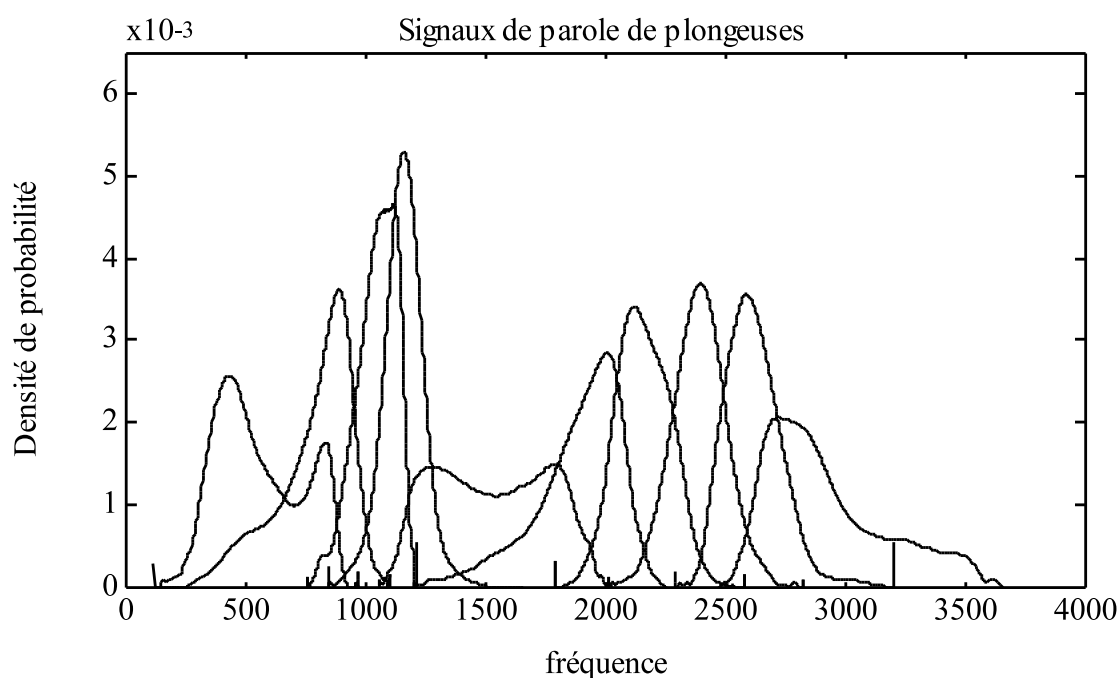


FIG. 3.3 — Densité de probabilité des CLSP signaux plongeurs

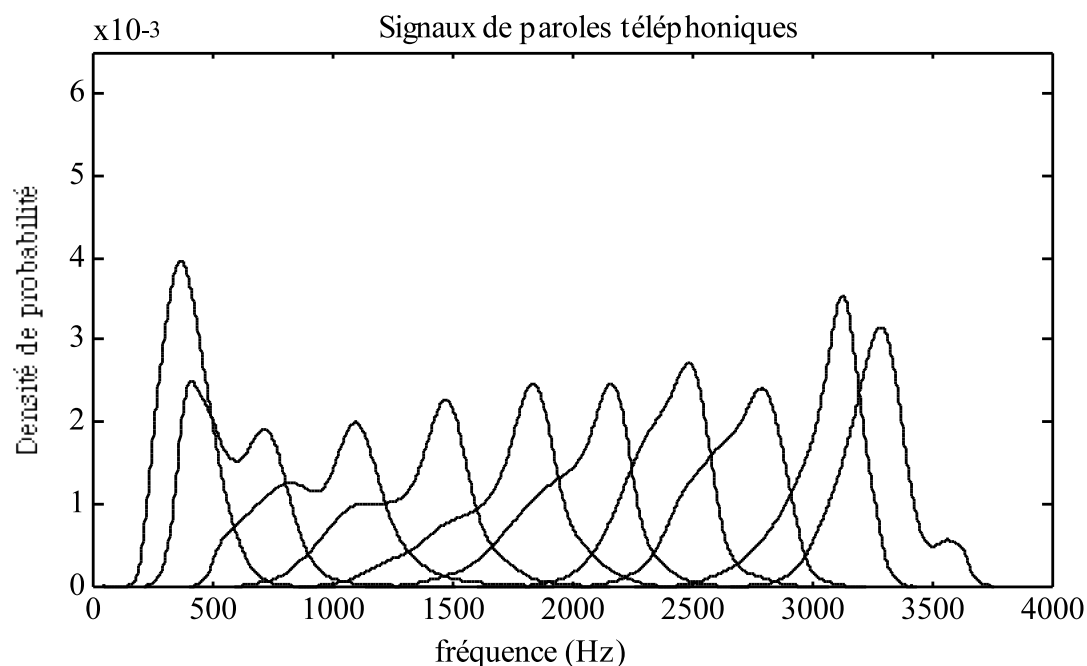


FIG. 3.4 — Densité de probabilité des CLSP signaux téléphoniques

### Détermination du filtre prédicteur court terme :

Avant de déterminer les paramètres long terme, les redondances court terme sont extraites du signal de parole. Il suffit pour cela de filtrer le signal original par la fonction de transfert inverse du conduit vocal. Cela nous amène donc à déterminer au codeur les com-



posantes du polynôme prédicteur court terme  $A_p(z)$ . Afin de prendre en compte les erreurs de quantification, celui-ci doit être calculé à partir des valeurs quantifiées des CLSP. Le filtre peut directement s'exprimer en fonction des CLSP. Différentes possibilités existent pour calculer le polynôme prédicteur. La méthode la plus simple, compte tenu de l'ordre de prédiction, est d'exprimer sous forme polynomiale les coefficients  $\{a_i\}$  à partir des différentes paires de raies spectrales. Les échantillons du signal glottique  $e(n)$  sont par la suite évalués par filtrage inverse du signal de parole suivant l'expression :

$$e(n) = \sum_{i=0}^p a_i S(n-i) \quad (3.7)$$

Le gain syllabique  $G_{lpc}$  est un moyen d'évaluation des performances de l'analyse court terme. Les paires de raies spectrales sont plus particulièrement caractérisées par leur meilleure aptitude à modéliser l'enveloppe spectrale du signal à bas débit, grâce aux propriétés de quantification [23]. Le signal glottique issu du filtrage est caractérisé par un niveau énergétique réduit c'est à dire un écart type plus petit. C'est donc cette grandeur qui, après quantification transmet l'information sur le niveau énergétique de la trame synthétisée.

$$G_{lpc} = \left[ \frac{1}{N} \sum_{n=0}^{N-1} e^2(n) \right]^{1/2} \quad (3.8)$$

Le gain syllabique est quantifié non uniformément sur 7 bits et sera noté  $G_{lpc}$ . Dans la pratique, ce paramètre permet d'adapter le codeur à la dynamique des signaux d'entrée. L'adaptabilité reste cependant limitée, dans la mesure où il ne s'agit pas de changer de table de quantification suivant la dynamique des signaux. C'est à partir des échantillons du signal glottique qu'est évalué le prédicteur long terme.

### 3.3 Analyse long terme

L'analyse long terme a pour but de modéliser les redondances de la source vocale, principalement pour les sons voisés. Au décodeur, lors de la synthèse du signal, le filtre long terme restitue la période fondamentale du signal. Au codeur, il s'agit d'extraire du signal glottique les paramètres susceptibles de modéliser les périodicités de la source vocale.

#### – Schéma général d'un prédicteur long terme

Le prédicteur long terme permet d'extraire les redondances d'un signal de parole périodique en prédisant l'échantillon présent à partir d'une combinaison linéaire des échantillons précédents retardés de  $P$  échantillons. La forme générale d'un prédicteur d'ordre  $l$  impair avec un retard  $P$  et des coefficients de prédiction  $b(k)$  est la suivante :

$$P(z) = 1 - \sum_{k=-(l-1)/2}^{(l-1)/2} b(k) z^{-(P+k)} \quad l=1,3,\dots \quad (3.9)$$

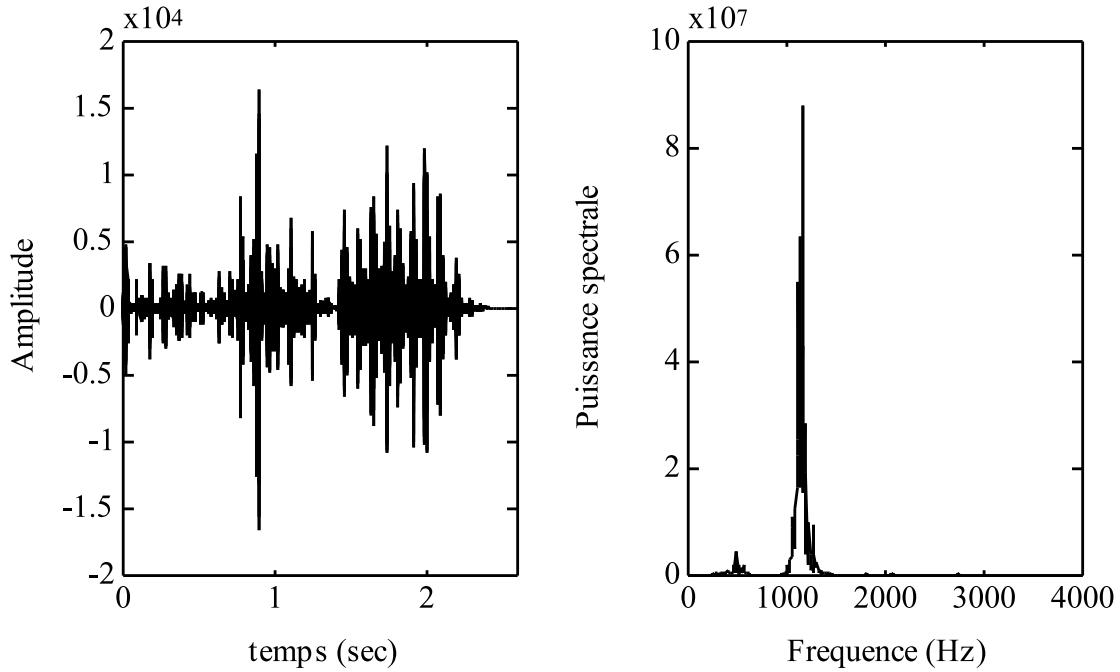


FIG. 3.5 — Influence des bulles sur les CLSP

La valeur de  $P$  est équivalente, en nombre d'échantillons, à un retard pouvant varier de 2 à 20 ms. Pour les signaux périodiques, l'inverse du retard correspond à la fréquence fondamentale  $F_0$ . L'efficacité du prédicteur long terme dépend de plusieurs paramètres comme la fréquence de renouvellement des coefficients du prédicteur, l'ordre de prédiction et le taux de voisement du signal de parole traitée. Pour ne pas augmenter le débit binaire, on se contente la plupart du temps d'un prédicteur à un seul coefficient. La détermination d'un prédicteur du premier ordre consiste à établir une relation entre l'échantillon  $e(n)$  du signal glottique et l'échantillon du même signal décalé de  $P$  et pondéré par le facteur de voisement  $b$ . Pour une trame de  $N$  échantillons, le calcul du prédicteur consiste à minimiser l'expression suivante :

$$E(P, b) = \sum_{n=0}^{N-1} [e(n) - \hat{e}(n)]^2 = \sum_{n=0}^{N-1} [e(n) - be(n - P)]^2 \quad (3.10)$$

Pour un retard  $P$ , la valeur optimale de  $b$  est obtenue en annulant la dérivée de  $E(P, b)$  par rapport à  $b$ , ce qui conduit à :

$$b = \frac{\sum_{n=0}^{N-1} e(n)e(n - P)}{\sum_{n=0}^{N-1} e^2(n - P)} \quad (3.11)$$

On en déduit immédiatement l'expression de l'erreur :

$$E(P) = \sum_{n=0}^{N-1} e^2(n) - \frac{\left[ \sum_{n=0}^{N-1} e(n)e(n - P) \right]^2}{\sum_{n=0}^{N-1} e^2(n - P)} \quad (3.12)$$

La minimisation de l'erreur est obtenue pour la valeur maximale de l'expression :

$$\frac{\left[ \sum_{n=0}^{N-1} e(n)e(n-P) \right]^2}{\sum_{n=0}^{N-1} e^2(n-P)} \quad (3.13)$$

Cette grandeur devrait donc être calculée pour chaque valeur du retard  $P$ , soit donc pour des valeurs variant de 16 à 143

– Correlation Peak-Picking Method

Différentes méthodes sont utilisables pour la détermination du filtre prédicteur long terme mais la contrainte de fonctionnement temps réel nous conduit à utiliser une méthode moins coûteuse en temps de calcul. Elle est connue sous le nom de "Correlation Peak-Picking Method" et consiste à évaluer non pas l'expression précédente, pour chaque valeur de  $P$ , mais la fonction d'autocorrélation du signal de parole après décorrélation court terme, c'est à dire la fonction d'autocorrélation du signal glottique. La valeur non optimale du retard correspond au maximum de cette fonction soit :

$$MaxE(P) = Max \sum_{n=0}^{N-1} e(n)e(n-P) \quad (3.14)$$

$$P = ArgMax(E(P)) = ArgMax\left(\sum_{n=0}^{N-1} e(n)e(n-P)\right) \quad (3.15)$$

Compte tenu de la plage de variation du fondamental, la recherche de  $P$  (aussi appelé pitch) s'effectue pour des valeurs comprise entre 16 et 143. Le calcul du coefficient de prédiction  $b$  ne pose pas de problème particulier et ce d'autant plus que ce paramètre doit être quantifié, la division (3.11) est ici remplacée par l'opération multiplication-quantification.

– Quantification des paramètres long terme

Le signal de parole est échantillonné à la fréquence  $F_e$ ,  $p$  est donc aussi le nombre d'échantillons contenus dans une période fondamentale :

$$p = \frac{F_e}{F_o} \quad (3.16)$$

Pour des fréquences fondamentales s'étalant de 56 à 500 Hz, susceptibles de modéliser les voix enfantines, féminines et masculines le retard  $p$  varie de 16 à 143. Le retard est aléatoire pour les sons non voisés. La plage de variation du pitch est donc de 128, d'où sa quantification sur 7 bits.

– Répartition du pitch

La figure (Fig. 3.6) présente la répartition du pitch pour la base de données constituée à partir de signaux de parole de plongeurs, les fréquences fondamentales se situant aux alentours de 200 Hz (voix masculines). Le coefficient  $b$  représente le degré de voisement du signal. Pour les sons voisés sa valeur approche l'unité. La stabilité du filtre long terme n'est assurée que pour les valeurs de  $b$  inférieures à un. Le coefficient  $b$  est quantifié sur 3 bits et est noté  $b_q$ .

– facteur de voisement du prédicteur long terme

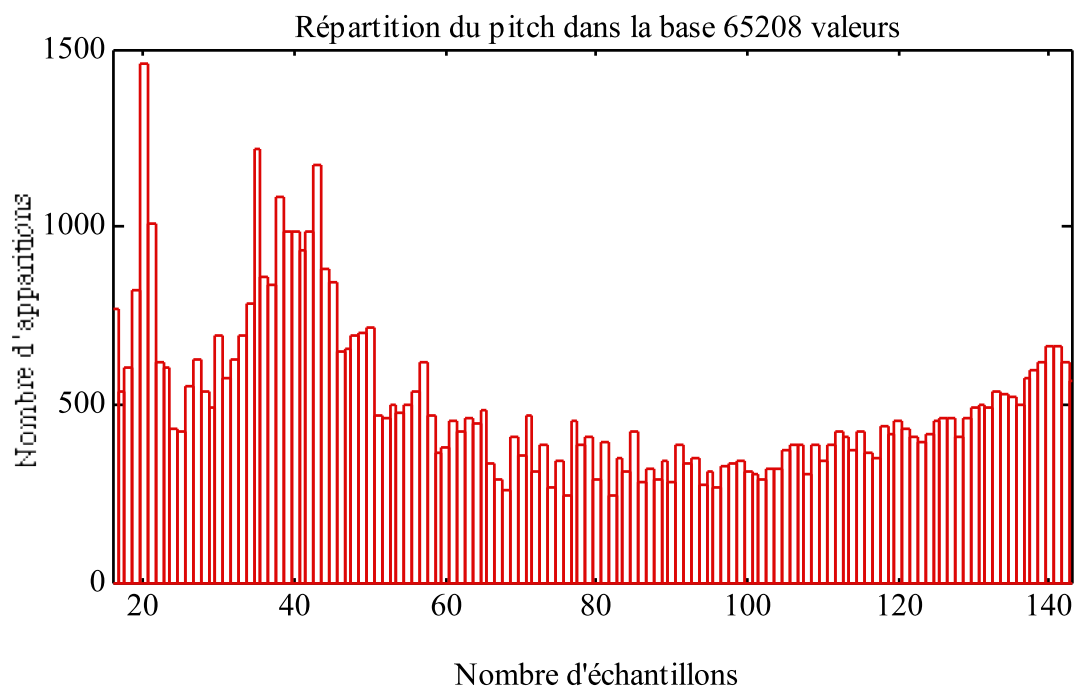


FIG. 3.6 — Répartition du pitch

La figure (Fig. 3.7) présente la densité de probabilité du facteur de voisement pour la base de données. Théoriquement borné à un, il ne l'est pas dans la pratique. La valeur de  $b$  n'est en effet qu'une estimation. Dans la plupart des cas  $b$  est quantifié uniformément sur  $[0, \dots, 1]$ . Une quantification non uniforme a cependant été testée. Après décorrélation court et long terme le signal de parole se présente sous la forme d'un résidu. Les codeurs CELP le modélisent par des formes d'ondes stockées dans un dictionnaire.

## 3.4 Dictionnaire de formes d'ondes d'excitation

Pour rechercher l'excitation optimale, la méthode CELP utilise l'analyse par synthèse. Celle-ci consiste à modéliser le résidu par une forme d'onde choisie parmi un ensemble de  $N$  formes stockées dans un dictionnaire, la forme choisie rendant minimale l'énergie de la différence entre le résidu à coder et la version synthétisée de ce même résidu. L'introduction dans l'expression de l'erreur de codage d'une pondération perceptuelle permet, en tenant compte des caractéristiques de l'oreille, de masquer le bruit de quantification sous les formants du signal de parole [2]. Cela a pour effet d'améliorer considérablement la qualité de la parole synthétisée.

### 3.4.1 Analyse par synthèse

La sélection de la forme d'onde optimale utilise le principe de l'analyse par synthèse. Ce principe conduit à minimiser le critère d'erreur  $E(i)$  entre une séquence  $S(n)$  du si-

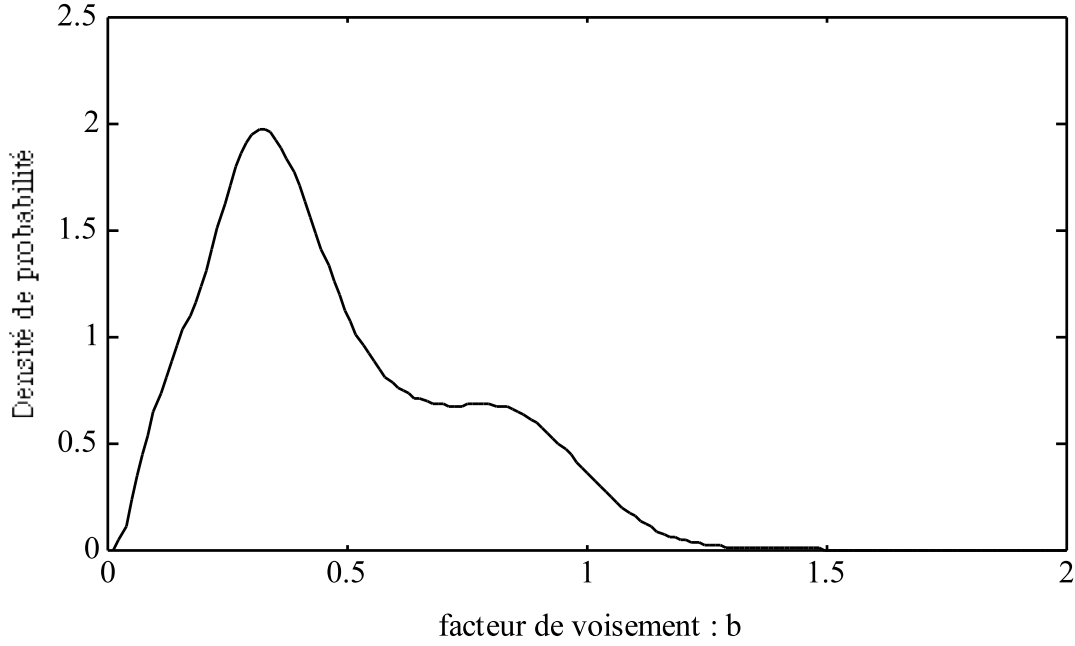


FIG. 3.7 — facteur de voisement du prédicteur long terme

gnal et une estimation paramétrique de ce même signal. Les estimations paramétriques  $\hat{S}_i(n)$  sont obtenues à partir du filtrage, à travers les filtres court et long terme, des formes d'ondes stockées dans le dictionnaire. Pour prendre en compte les erreurs de quantification, les filtres prédicteurs sont reconstruits à partir des valeurs quantifiées des paramètres. Le codeur comporte donc de fait un décodeur local. Si on note  $S(z)$ , (resp.  $\hat{S}_i(z)$ ), la transformée en  $Z$  du signal  $S(n)$ , (resp.  $\hat{S}_i(n)$ ), on obtient (théorème de Parseval) :

$$E(i) = \sum_n e^2(n) = \sum_n \left( S(n) - \hat{S}_i(n) \right)^2 = \frac{1}{2j\pi} \oint_C \left( S(z) - \hat{S}_i(z) \right) \left( S(z^{-1}) - \hat{S}_i(z^{-1}) \right) \frac{dz}{z} \quad (3.17)$$

On en déduit le critère d'erreur permettant de choisir l'excitation optimale :

$$E(i) = \frac{1}{2j\pi} \oint_C \left\| S(z) - \hat{S}_i(z) \right\|^2 \frac{dz}{z} \quad (3.18)$$

( $S^*(z) = S(z^{-1})$  car le signal est réel),  $C$  est un parcours orienté dans le sens positif appartenant au domaine de convergence de  $\left( S(z) - \hat{S}_i(z) \right)$ . Si on effectue le changement de variable  $z = e^{j2\pi f}$ ,  $f \in \left[ -\frac{1}{2}, \frac{1}{2} \right]$ ,  $dz = 2j\pi z df$ . L'expression du critère d'erreur devient :

$$E(i) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left\| S(f) - \hat{S}_i(f) \right\|^2 df \quad (3.19)$$

avec  $S(f) = \sum_n S(n)e^{-j2\pi fn}$ , transformée de Fourier du signal discret  $S(n)$ .

### 3.4.2 Pondération perceptuelle du critère d'erreur

Une propriété intéressante du point de vue du traitement de signal est la particularité pour le système auditif d'enregistrer les sons de façon fréquentielle. Le spectre d'un signal de parole est caractérisé par l'existence de formants (résonances du canal vocal) et de vallées. Il est donc à priori intéressant de répartir différemment la puissance de bruit de quantification, en augmentant son niveau au-dessous des formants et en le diminuant dans les vallées. Le bruit est alors masqué par la puissance du signal. Une nouvelle expression du critère d'erreur introduit cet effet de masquage :

$$E(i) = \oint_C \left\| \left[ S(z) - \hat{S}_i(z) \right] \frac{A(z)}{A_g(z)} \right\|^2 \frac{dz}{z} \quad (3.20)$$

avec

$$\frac{A(z)}{A_g(z)} = \frac{1 + \sum_{k=1}^P a_k z^{-k}}{1 + \sum_{k=1}^P a_k g^k z^{-k}} \quad (3.21)$$

L'effet de ce filtre est d'atténuer faiblement en fréquence le signal d'erreur au niveau des formants et au contraire de façon importante au niveau des vallées. La figure (Fig. 3.9) présente la réponse fréquentielle du filtre modélisant le conduit vocal et du filtre de pondération perceptuelle ( $g=0.8$ ), pour une trame de parole.

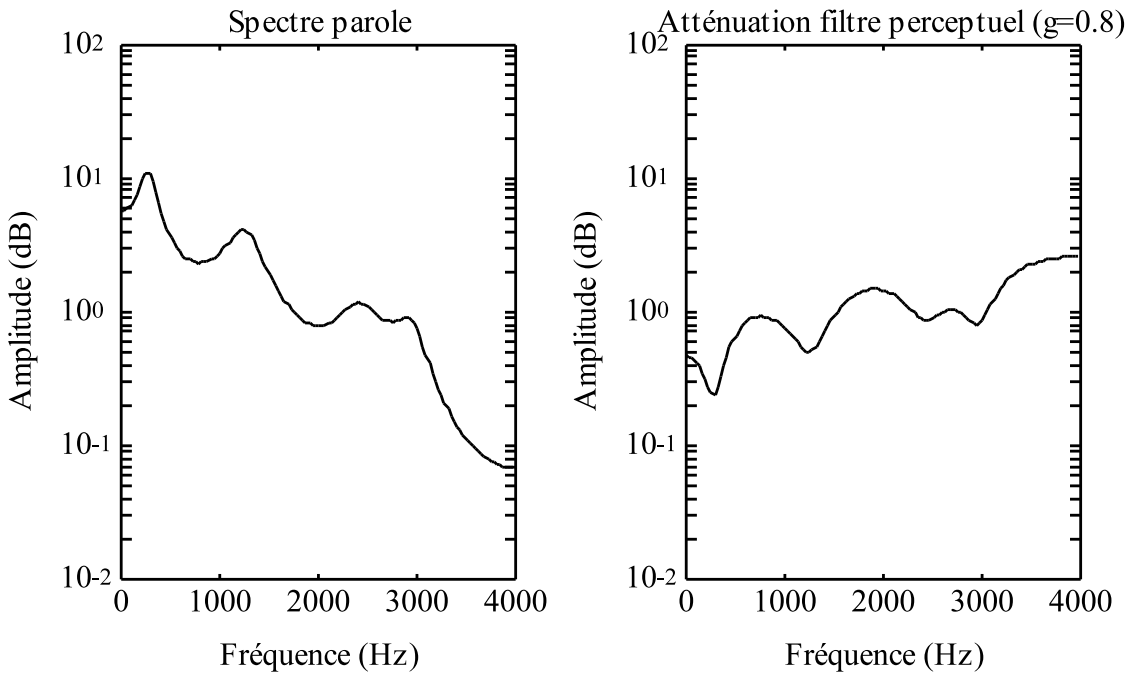
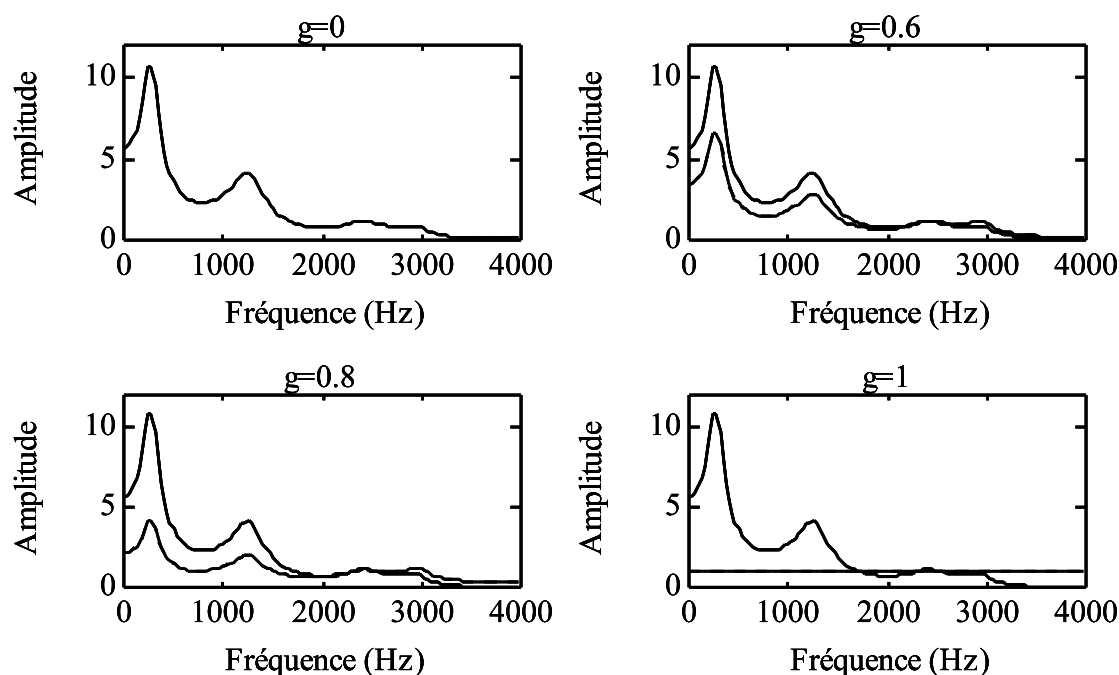


FIG. 3.8 — Filtre de pondération perceptuelle

La figure (Fig. 3.8) quant à elle, présente le spectre du bruit de codage suivant les valeurs du coefficient  $g$ . Pour  $g$  égal zéro, la réponse fréquentielle du filtre perceptuel correspond à celle du conduit vocal alors que pour  $g$  égal un, l'effet du filtre perceptuel disparaît. Une bonne répartition fréquentielle du bruit est obtenue pour des valeurs de  $g$

FIG. 3.9 — Spectre du bruit de codage en fonction de  $g$ 

comprise entre 0.7 et 0.9. Le schéma de principe de recherche de la forme d'onde optimale est le suivant :

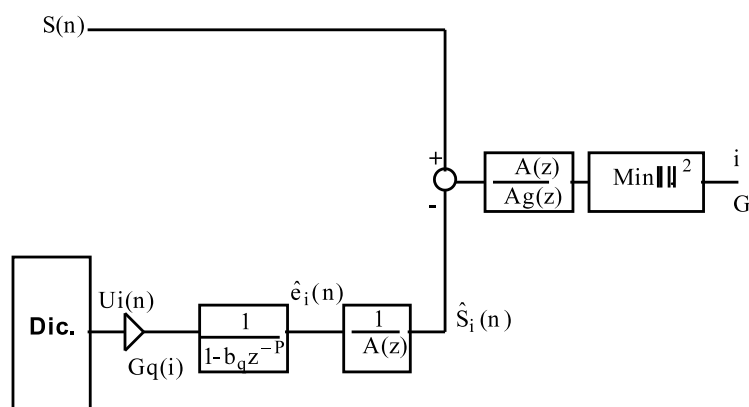


FIG. 3.10 — Structure de base de sélection de la forme d'onde optimale

- $S(n)$  Signal de parole
- $\hat{S}_i(n)$  Signal de parole synthétisée à partir de la forme d'indice  $i$
- $A(z)$  Filtre de prédiction court terme
- $U_i(n)$  Forme d'onde d'indice  $i$
- $G_q(i)$  Gain associé à la forme d'indice  $i$
- $\hat{e}_i(n)$  Signal glottique estimé correspondant à la forme  $i$
- $P, b$  Paramètres LTP

Le calcul de la forme d'onde optimale demande de très lourds calculs de filtrage dans sa version de base (Fig. 3.10). Chaque forme du dictionnaire donne après filtrage court

et long terme un signal reconstitué susceptible de modéliser le signal original. Le critère d'erreur retenu dans les codeurs de type CELP prend en compte les caractéristiques physiologiques de l'oreille, entraînant une autre opération importante de filtrage. Le filtre perceptuel, agissant sur le spectre du signal de parole est en fait dérivé du filtre court terme. Quand on examine de plus près le traitement subi par les différentes formes d'ondes, il apparaît clairement (Fig. 3.10) qu'elles traversent, après filtrage long terme, le filtre prédictif court terme  $H(z) = \frac{1}{A(z)}$  puis, immédiatement après, le filtre perceptuel dont la forme entraîne indirectement un passage à travers l'inverse du filtre prédictif court terme. La suppression de ce filtrage redondant permet d'envisager une structure simplifiée. L'opération est à exécuter sur toutes les formes avant la sélection de la forme optimale.

### 3.4.3 Structure simplifiée

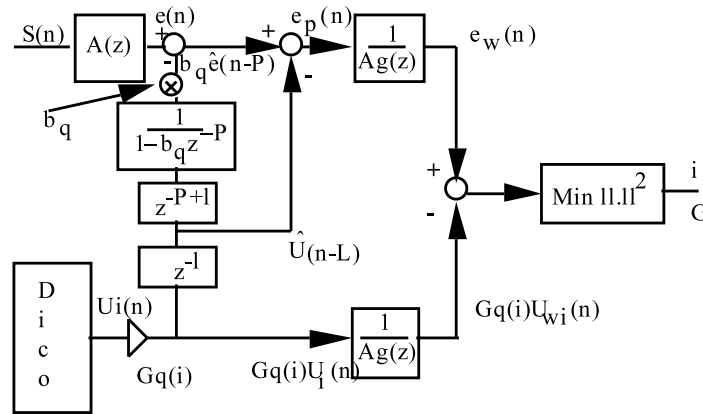


FIG. 3.11 — Structure simplifiée de sélection de la forme d'onde

Le signal d'excitation est remis à jour plus souvent que les filtres prédictifs ; la forme d'onde d'excitation est ainsi réactualisée quatre fois par trame. Un certain nombre de modifications peuvent être apportées à la structure de base pour réduire le nombre d'opérations et optimiser le temps de calcul [15], [16]. Dans la nouvelle structure (Fig. 3.11), les formes d'ondes du dictionnaire ne subissent plus l'opération filtrage-filtrage inverse court terme et, du même coup, le critère d'erreur ne s'applique plus à la différence entre le signal original et sa valeur reconstituée, résultant du passage à travers le filtre perceptuel. Le critère d'erreur s'applique en fait aux résidus de signaux passés à travers le filtre perceptuel réduit  $\frac{1}{A_g(z)}$ . Ce changement de structure fait apparaître la possibilité de réaliser deux traitements indépendants : Le premier, à effectuer une fois par trame, consiste à filtrer l'ensemble des formes d'onde  $U_i(n)$  du dictionnaire par le filtre perceptuel réduit. La cadence est évidemment calquée sur la fréquence de renouvellement du filtre prédictif court terme. Le second, à réaliser au rythme de renouvellement des formes d'ondes, tient compte du passé récent du processus de production du signal vocal. Il induit un certain nombre d'opérations :

- filtrage inverse court terme du signal original, en vue d'obtenir le signal glottique  $e(n)$ .



- obtention de la version reconstruite du signal glottique  $\hat{b}e(n - p)$ , par le passage de la forme d'onde précédemment sélectionnée. Ce passage permet de prédire le signal glottique.
- filtrage du résidu prédit  $e_p(n)$  par le filtre perceptuel réduit : différence entre la prédiction et le signal original.

La compression du signal de parole par le codeur CELP conduit à la détermination de 23 paramètres codés sur 109 bits. L'ensemble de ces paramètres est transmis au décodeur pour synthétiser le signal de parole. Le paragraphe suivant présente les divers traitements liés à l'obtention du signal de parole synthétisée.

## 3.5 Décodeur CELP

### 3.5.1 Introduction

Le décodeur de parole CELP est évidemment beaucoup plus simple. On y retrouve dans l'ordre inverse les éléments du codeur à savoir (Fig. 3.12) :

- Dictionnaire d'excitation
- Filtre prédicteur long terme
- Filtre prédicteur court terme

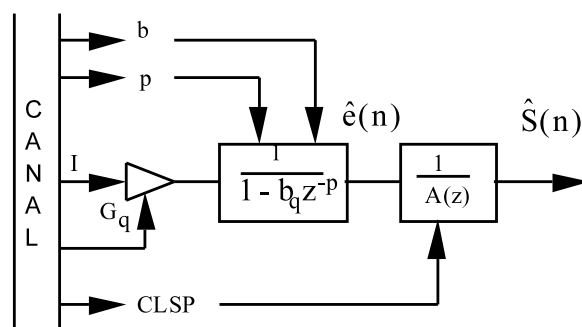


FIG. 3.12 — Décodeur - schéma d'implantation

Le décodeur reçoit du canal de transmission ou d'un fichier (cas du stockage de parole comprimée) les 109 bits qui vont permettre de sélectionner dans les tables de quantification et dans le dictionnaire les différents paramètres pour reconstruire les filtres prédicteurs et l'excitation. La figure (Fig. 3.13) présente le schéma d'implantation du décodeur sur le processeur.

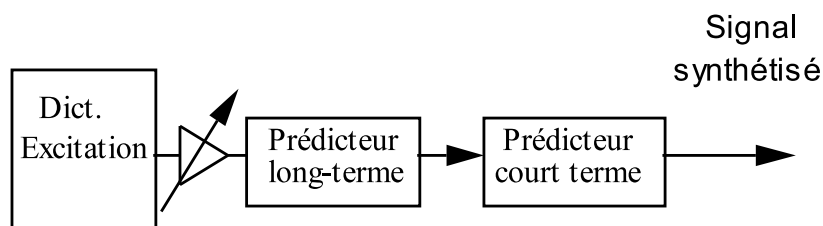


FIG. 3.13 — Décodeur - schéma d'implantation

### 3.5.2 Dictionnaire d'Excitation

Les échantillons de la forme d'onde sélectionnée sont multipliés par la valeur quantifiée du gain associé à la forme d'onde. Rappelons que ce gain est le produit de deux termes : le gain syllabique  $G_{lpc}$  constant sur toute la trame et le gain d'excitation  $g_{exc}(i)$  facteur modulant du gain syllabique. Ce dernier est renouvelé au rythme de l'excitation, c'est à dire quatre fois par trame. L'expression du gain est la suivante :

$$G_q(i) = G_{lpc}g_{exc}(i) \quad (3.22)$$

Le signal d'excitation est le résultat de la multiplication de chaque échantillon par le gain  $G_q(i)$ .

### 3.5.3 Reconstruction du Filtre Prédicteur long Terme

Le signal d'excitation sert à piloter le filtre prédicteur long terme pour générer le signal glottique  $\hat{e}(n)$ .

$$\hat{e}(n) = G_{lpc}g_{exc}(i)U_i(n) + b_q\hat{e}(n - P) \quad (3.23)$$

Le filtre prédicteur long terme restitue au signal d'excitation sa composante pseudo-périodique pour les sons voisés au travers du terme  $b_q\hat{e}(n - P)$ . Ces sons sont caractérisés par des valeurs de  $b$  proche de 1. Pour les sons non voisés, les valeurs sont proches de zéro.

### 3.5.4 Reconstruction du Filtre Prédicteur Court Terme

Après synthèse du signal glottique, le signal de parole est simplement obtenu par filtrage court terme :

$$\hat{S}(n) = \hat{e}(n) - \sum_{i=1}^P a_i \hat{S}(n - i) \quad (3.24)$$

Les composantes du filtre prédicteur court terme sont calculées à partir des valeurs quantifiées des CLSP. Le filtre prédicteur court terme restitue au signal glottique, préalablement prédit, ses composantes spectrales.

## 3.6 Conclusions

Après une mise en sommeil de plusieurs années, l'activité compression de la parole a été relancée en juin 2003. En effet, le département Signal et Communications développe, depuis le mois de novembre 2000 une plate forme de TRansmission d'Images, de Données, de texte EN Temps réel (TRIDENT) par onde acoustique. Le GESMA (Groupe d'Études Sous-Marines de l'Atlantique) maître d'ouvrage du projet a souhaité étendre les possibilités de la plate forme à la parole. Des essais réalisés en juillet 2003 ont été couronnés de succès avec transmission d'un signal de parole, avec un facteur de compression de 23,5 (débit binaire 5.45 kbits/s) sur une distance de 2 km. Le nouveau

projet démarrant en janvier 2006 vise à transmettre un signal de parole avec un facteur de compression de 53.33 soit un débit binaire de 2400 bits/s. Afin d'améliorer la qualité de la communication, le projet prévoit également d'adjoindre au système TRIDENT un codage de canal.

---

# Bibliographie

## 3.7 Publications

- [1] J.P. Adoul et al *Fast CELP coding based on algebraic codes* Proc. ICASSP'87, pp. 1957-1960, 1987.
- [2] B.S. Atal and M.R. Schroeder *Predictive coding of speech signal and subjective error criteria*, IEEE Trans. on Acoustics, Speech and Signal Processing ASSP-27, 1979, pp. 247-254.
- [3] Calliope *La parole et son traitement automatique* CNET-ENST, Collection technique et scientifique des télécommunications, Masson, Paris, 1989
- [4] A. Goalic, C Laot and S. Saoudi, *Real time implementation of a low bit rate coder for an acoustic underwater phone on a fixed point DSP MOTOROLA56001* ICSPAT, Boston, USA, pp. 921-927, November 1992.
- [5] A. Goalic et S. Saoudi, *Implémentation en temps réel de Codeur/Décodeur (CELP) de parole à bas débit sur processeur de signal en virgule fixe MOTOROLA DSP56001* GRETSI 93, Nice, FRANCE, pp. 1359-1362, Septembre 1993.
- [6] A. Goalic and S. Saoudi *Real time processing of a 5.45 kbits/sec TCELP coder/decoder on DSP MOTOROLA56001* ICSPAT'93, Santa clara, USA, pp. 1400-1406, October 1993.
- [7] A. Goalic *Étude et Réalisation de Codeur/Décodeur de Parole à bas débit*, Thèse de doctorat, Université de Bretagne Occidentale, Brest, juin 1994
- [8] A. Goalic, J. Labat, J. Trubuil, S. Saoudi and D. Rioualen *Toward a digital acoustic underwater phone* OCEANS 94, BREST, FRANCE, Vol III, pp 489-494, Septembre 1994
- [9] A. Goalic et S. Saoudi *An intrinsically reliable and fast algorithm to compute the Line Spectrum Pairs (LSP) in low bit rate CELP coding* ICASSP-95, Detroit, Michigan, U.S.A, Vol I, pp 728-731, Mai 1995
- [10] A. Goalic, J. Labat, J. Trubuil *Vers un téléphone acoustique numérique sous-marin* Acoustique Sous-Marine, IFREMER, BREST, FRANCE, pp 97-102, june 1995
- [11] Andre GOALIC, Joel TRUBUIL, Gérard LAPIERRE and Joël LABAT *Real Time low bit rate Speech transmission through underwater Acoustic channel* OCEANS'05 EUROPE, June 2005, Brest, Brittany.

- [12] F. Itakura and N. Sugamura *LSP speech synthesizer, its principle and implementation* Trans. Commit. Speech Res. Acoust. Soc., Japan, 1979.
- [13] P. Kabal and R.P. Ramachandran *The computation of line Spectral Frequencies using Chebyshev polynomials* Vol. ASSP-34, N° 6, pp. 1419-1425, December 1986
- [14] C. Laflamme, J.P. Adoul, H.Y. Su and S. Morissette *On reducing computational complexity of codebook search in CELP coder through the use of algebraic codes* Proc. ICASSP'90, Albuquerque, New Mexico, USA, April 1990, pp. 177-180
- [15] C. Laot *Etude d'un codeur de parole bas débit, simulation en langage C* ENST de Bretagne, 1992.
- [16] A. Le Guyader, D. Massaloux and F. Zurcher *A robust and fast CELP coder at 16 Kbits/s* Speech Communication, Vol. 7, N° 2, pp.217-226,1988.
- [17] A. Le Guyader, D. Massaloux and J.P. Petit *Robust and fast Code Excited Linear Predictive Coding of speech signals* Proc. ICASSP'89, May 1989, pp. 120-123.
- [18] A. Le Guyader, D. Massaloux and J.P. Petit *Robust and fast Code Excited Linear Predictive Coding of speech signals* Proc. ICASSP'89, May 1989, pp. 120-123.
- [19] S.P. Lloyd *Least square quantization in P.C.M.* IEEE Trans.Inform.Theory, Vol. IT-28, N°. 2, pp. 129-137, March 1982.
- [20] J. Max *Quantizing for minimum Distortion* IEEE Trans.Inform.Theory, pp. 7-12, March 1960.
- [21] N.Naja, A. Goalic, A. El Kesri and S. Saoudi *A real time processing TCELP co-der/decoder at 4.8 kbit/sec using LSP split vector quantization* EUSIPCO'96, European Signal Processing Conference, Trieste, September 10-13, 1996
- [22] S. Saito and K. Nakata *Line spectrum pairs* Academic press, Japan, 1985, Chap. 9, pp.127-132
- [23] S. Saoudi *Codage de la parole par les paires de raies spectrales* Thèse de doctorat, Université de Rennes 1, Décembre 1990
- [24] M.R. Schroeder and B.S. Atal *Code-Excited Linear Prediction (CELP) : High quality speech at very low bits rates* Proc. ICASSP'85, Tampa, Florida, USA pp. 937-940, March, 1985

## 4.1 Introduction

L'évolution rapide des communications numériques s'est accompagné d'une forte croissance des besoins des usagers, notamment en terme de débit et de mobilité, ce qui place des contraintes fortes sur l'utilisation des ressources physiques disponibles pour la transmission (puissance, bande passante,...). Le stockage ou la transmission de l'information nécessite la manipulation de signaux physiques naturellement sujets à diverses perturbations (bruit thermique, interférences de différentes natures, ...). L'influence de ces perturbations se traduit généralement par l'introduction d'erreurs sur le message binaire transmis qui affectent la qualité de la transmission. On mesure alors toute l'importance de disposer d'un procédé efficace de correction des erreurs en réception, afin de préserver l'intégrité du message. En 1948, Claude E. Shannon a établi l'existence de Codes Correcteurs d'Erreurs (CCE) capables d'assurer des transmissions numériques avec une probabilité d'erreur arbitrairement petite sous réserve que la quantité moyenne d'information transmise soit inférieure à un paramètre clé, appelé capacité du canal de transmission.

Par la suite, de nombreux travaux de recherche ont visés à définir des CCE performants à la fois simples à encoder et décoder. Ces travaux ont donné lieu à des contributions remarquables, tels que les codes Bose-Chaudhuri-Hocquenghem (BCH), les codes de Reed-Solomon (RS) ou bien encore les codes convolutifs. Au début des années 80, les chercheurs pensaient avoir atteint ce que l'on pouvait faire de mieux en pratique dans le domaine du codage de canal alors qu'il subsistait encore un écart de 3 dB par rapport aux performances optimales promises par Shannon. En 1993, C. Berrou et A. Glavieux ont proposé un nouveau schéma de CCE baptisé Turbo-Code Convolutif (TCC) capable d'opérer à moins de 0.5 dB de la limite de Shannon. Ce dernier reposait sur un décodage à entrée et sortie souple (Soft-Input Soft-Output, ou SISO) de deux codes convolutifs récurrents systématiques concaténés en parallèle par le biais d'un entrelaceur. Ce concept a ensuite été rapidement étendu aux codes en blocs, en 1994, par Ramesh Pyndiah (Turbo-Codes en Blocs, ou TCB) puis généralisé à d'autres fonctions de la chaîne de réception (Turbo-Egalisation, Turbo-Synchronisation, etc.). Les TCC et les TCB commencent à être intégrés dans différentes normes ([40] (satellites, communications mobiles, télévisions numériques etc.) et de nombreux travaux visent à les intégrer dans les secteurs les plus divers. Le département Signal et Communications de l'ENST-Bretagne s'intéresse plus particulièrement au TCB, à la fois à partir des codes BCH binaires, ([20], [8], [3]) et des codes de Reed-Solomon RS ([1], [33]). En 1996 j'ai commencé les premiers travaux

d'intégration des TCB sur processeurs de signaux (Digital Signal Processor DSP). Avec ses très grandes possibilités de parallélisme, l'option circuit permet d'atteindre des débits très élevés. L'option DSP apporte une très grande souplesse malgré son parallélisme limité et permet une évolutivité au niveau de l'intégration et de la mise à jour rapide des systèmes. Pour s'adapter à l'application cible, les codes et les turbo codes en blocs (BCH et RS) offrent un certain nombre de degrés de liberté. Car il est possible de jouer sur :

- taille du mot de code, le nombre de bits d'information, la différence entre ces deux paramètres dépendant du pouvoir de correction du code.
- l'emploi de codes élémentaires différents pour la définition du code produit.
- la mise en œuvre de technique de raccourcissement et de poinçonnage pour adapter finement la taille de la trame au besoin de l'application.
- l'utilisation de la version étendue du code ou la version expurgée

Ce chapitre présente rapidement les différents algorithmes mis en œuvre pour le décodage pondéré des TCB, à la fois de type BCH (Chase-Pyndiah, Chase-Hartmann-Rudolph-Nazarov) et de type RS (Chase-Pyndiah-Berlekamp-Massey), dans le cadre des thèses que j'ai co-encadrées et de différents travaux réalisés dans le cadre de contrats (voir Chapitre 5 : Applications des Turbo Codes). Les différents algorithmes ont été initialement développés en langage C en format flottant. Les implémentations<sup>1</sup>, en format fixe, sur processeurs de signaux ont été optimisées en langage d'assemblage (DSP Motorola 56xxx, DSP Texas Instruments TMS320C6201)

## 4.2 Décodage itératif des codes produits

Les codes produits furent introduits par Elias en 1954 [5]. Ils résultent de la concaténation en série de deux codes en blocs linéaires systématiques  $\mathcal{C}_1(n_1, k_1, \delta_1)$  et  $\mathcal{C}_2(n_2, k_2, \delta_2)$  où  $n_i$ ,  $k_i$ , et  $\delta_i$  désignent respectivement la longueur (nombre de symboles codés), la dimension (nombre de symboles d'information) et la distance minimale de Hamming du code  $\mathcal{C}_i$ . Le code  $\mathcal{P} = \mathcal{C}_1 \otimes \mathcal{C}_2$  est alors ainsi obtenu (figure 4.1) :

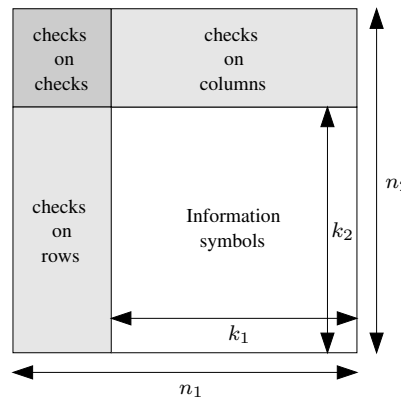


FIG. 4.1 — Code produit  $\mathcal{P} = \mathcal{C}_1 \otimes \mathcal{C}_2$

<sup>1</sup>l'implémentation des codes RS est aujourd'hui en cours de développement dans le cadre d'un nouveau projet

1. on place les  $k_1 \times k_2$  symboles d'information à coder dans une matrice de  $k_1$  colonnes et  $k_2$  lignes.
2. les  $k_2$  lignes sont codées par le code  $\mathcal{C}_1$
3. les  $n_1$  colonnes sont codées par le code  $\mathcal{C}_2$

Les paramètres de  $\mathcal{P}$  sont  $n = n_1 \times n_2$ ,  $k = k_1 \times k_2$ , et  $\delta = \delta_1 \times \delta_2$ . Le rendement  $R$  de  $\mathcal{P}$  est égal à  $R_1 \times R_2$ , produit des rendements des codes  $\mathcal{C}_1$  et  $\mathcal{C}_2$ . Il est important de noter que les deux dernières étapes sont interchangeables. Ceci traduit une propriété intéressante du code produit : les  $k_2$  lignes sont des mots de codes de  $\mathcal{C}_1$  et les  $k_1$  colonnes sont des mots de codes de  $\mathcal{C}_2$ , ce qui peut être éventuellement exploitable dans un schéma de détection d'erreur. Précisons enfin que cet encodage peut être étendu à 3 dimensions (ce qui permet d'atteindre des rendements plus faibles). De plus, les codes constitutifs du code produits sont généralement étendus ou expurgés (ce qui permet d'obtenir des distances minimales plus élevées).

### 4.3 Décodage des Turbo Codes en blocs BCH

#### 4.3.1 Décodage des Turbo Codes en blocs avec l'algorithme de Chase Pyndiah

Ramesh Pyndiah en 1994 [20] a proposé un nouvel algorithme de décodage des codes produits, basé sur un décodage pondéré, proche des performances optimales de décodage des codes produits, en terme de Taux d'Erreur Binaire TEB. Après codage, modulation et bruitage par un canal Gaussien à bruit additif. Soit  $E = (e_1, \dots, e_i, \dots, e_n)$  le mot de code émis,  $N = (n_1, \dots, n_i, \dots, n_n)$  le bruit additif gaussien d'écart type  $\sigma$ , le mot de code reçu  $R = (r_1, \dots, r_i, \dots, r_n)$  est de la forme eq. (4.17) :

$$R = E + N \quad (4.1)$$

Le décodage optimal de la décision  $D$  du mot  $R$  prend la forme suivante eq. (4.2) :

$$D = C^i \text{ si } P_r\{E = C^i/R\} > P_r\{E = C^j/R\} \forall j \neq i \quad (4.2)$$

ou :

$$\|R - C^i\|^2 = \sum_{l=1}^n (r_l - c_l^i)^2 \quad (4.3)$$

Pour les codes en blocs à rendements élevés, le nombre de mot de code  $2^k$  est relativement important et le décodage optimal demande des temps de calcul incompatibles avec des implémentations temps réels. En 1972 Chase [4] propose un algorithme pour réduire le temps de calcul avec une réduction de performance limitée. Au lieu de passer en revue tous les mots de code, Chase réduit sa recherche à la distance de Hamming dans une sphère de rayon  $(\delta - 1)$ , centrée sur le mot  $Y = (y_1, \dots, y_i, \dots, y_n)$  ou  $y_i = 1$  si  $r_i > 0$  et  $y_i = 0$  si  $r_i \leq 0$ . En utilisant les informations issues du canal, seuls les mots de codes les plus probables sont sélectionnés et il est possible de décomposer la procédure de recherche en trois étapes :



- étape 1 : détermination des  $m$  bits les moins fiables de  $Y$  en utilisant le mot reçu  $R$  ( $m=3$ ) pour des contraintes temps réels.
- étape 2 : formation de l'ensemble  $T^q$  pour  $q = 0, \dots, 2^m$  de l'ensemble des mots de code avec des "0" et des "1" aux  $m$  positions les moins fiables.
- étape 3 : après avoir formé les  $q$  mots de code  $Z^q = Y \oplus T^q$  procéder au décodage des  $Z^q$ . Cette procédure inclue le calcul des syndromes pour l'ensemble des  $2^m$  mots à tester. Le décodeur algébrique détermine  $2^m$  mots de codes<sup>2</sup>, l'un d'entre eux étant sélectionnés par une décision pondérée

L'algorithme de Chase permet d'obtenir une décision  $D$  pour une information  $R$ . Le logarithme du rapport de vraisemblance pour  $d_j$  notée  $(LLR)_j$  s'exprime de la façon suivante eq. (4.4) :

$$(LLR)_j = \ln \left\{ \frac{Pr\{e_j = +1/R\}}{Pr\{e_j = -1/R\}} \right\} \quad (4.4)$$

En considérant les différents mots de code, le numérateur de eq. (4.4) peut s'exprimer de la façon suivante eq. (4.5) :

$$Pr\{e_j = +1/R\} = \sum_{C^i \in S_j^{+1}} Pr\left\{E = C^i/R\right\} \quad (4.5)$$

$S_j^{+1}$  étant l'ensemble des mots de code tel que  $c_j^i = +1$ . De même nous avons eq. (4.6) :

$$Pr\{e_j = -1/R\} = \sum_{C^i \in S_j^{-1}} Pr\left\{E = C^i/R\right\} \quad (4.6)$$

avec  $S_j^{-1}$  l'ensemble des mots de code tel que  $c_j^i = -1$ . En appliquant la règle de Bayes aux équations eq. (4.5) et eq. (4.6) et considérant que les différents mots de code sont uniformément distribués, le  $LLR$  de  $d_j$  défini par eq. (4.4) l'expression suivante eq. (4.7) :

$$(LLR)_j = \ln \left\{ \frac{\sum_{C^i \in S_j^{+1}} p\{R/E = C^i\}}{\sum_{C^i \in S_j^{-1}} p\{R/E = C^i\}} \right\} \quad (4.7)$$

ou la densité de probabilité  $p\{\}$  est donné par eq. (4.8) :

$$p\{R/E = C^i\} = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp \left\{ -\frac{|R - C^i|^2}{2\sigma^2} \right\} \quad (4.8)$$

En combinant eq. (4.7) et eq. (4.8), on obtient eq. (4.9) :

$$(LLR)_j = \ln \left\{ \frac{\sum_{C^i \in S_j^{+1}} e^{-\frac{|R - C^i|^2}{2\sigma^2}}}{\sum_{C^i \in S_j^{-1}} e^{-\frac{|R - C^i|^2}{2\sigma^2}}} \right\} \quad (4.9)$$

L'évaluation du  $LLR$  utilisant eq. (4.9) est trop complexe pour des codes en bloc à haut rendement et évidemment pour des implémentations temps réel. En notant  $C^{min(+1)}$ , le

<sup>2</sup>uniquement valable pour un pouvoir de correction de  $t = 1$

mot de code à la distance Euclidienne minimale de  $R$  avec  $c_j^{min(+1)} = +1$  et  $C^{min(-1)}$ , le mot de code à la distance Euclidienne minimale de  $R$  avec  $c_j^{min(-1)} = -1$ , la valeur du  $LLR$  peut être approximée par l'expression suivante eq. (4.10) :

$$(LLR)_j = \frac{1}{\sigma^2} \left( |R - C^{min(-1)}|^2 - |R - C^{min(+1)}|^2 \right) \quad (4.10)$$

En développant l'équation eq. (4.10) en utilisant l'équation eq. (4.3) on obtient eq. (4.11) :

$$(LLR)_j = \frac{1}{\sigma^2} \left( r_j + \sum_{l=1, l \neq j}^n r_j c_l^{min(+1)} p_l \right) \quad (4.11)$$

ou :

$$p_l = \begin{cases} 0 & \text{si } c_l^{min(+1)} = c_l^{min(-1)} \\ 1 & \text{si } c_l^{min(+1)} \neq c_l^{min(-1)} \end{cases}$$

et  $c_l^{min(+1)}$  est le  $l^{th}$  symbole appartenant à  $C^{min(+1)}$ . Si on normalise la valeur du  $LLR$  de  $d_j$  par rapport à  $2/\sigma^2$ , on obtient eq. (4.12) :

$$r'_j = r_j + w_j \quad (4.12)$$

ou eq. (4.13) :

$$w_j = \sum_{l=1, l \neq j}^n r_j c_l^{min(+1)} p_l \quad (4.13)$$

Pour calculer le  $LLR$  normalisé, on suppose que l'on a identifié les 2 mots de code  $C^{min(+1)}$  et  $C^{min(-1)}$ . Pour chaque mot de code  $C^q$  identifié à l'étape 3, on calcule sa distance Euclidienne à  $R$  eq. (4.14) :

$$M^q = |R - C^q|^2 \quad (4.14)$$

Comme dans l'algorithme de Chase, on sélectionne le mot de code  $C^d$  à la distance Euclidienne minimale de  $R$ . Puis on recherche le mot de code  $C^c$  situé à la distance minimale de  $R$  de telle façon que  $c_j^c \neq c_j^d$ . Si l'on trouve un tel mot de code, on pondère la décision sur chaque bit  $d_j$  de la séquence décodée par l'expression suivante eq. (4.15) :

$$r'_j = \left( \frac{M^c - M^d}{4} \right) c_j^d \quad (4.15)$$

dans le cas contraire eq. (4.16) :

$$r'_j = \beta c_j^d \quad (4.16)$$

ou  $\beta$  est une constante dépendant du numéro de l'itération.

Le décodage itératif consiste à décoder successivement les lignes puis les colonnes du code produit  $P$  et à répéter cette opération un nombre de fois prédéfini ou jusqu'à ce qu'un critère d'arrêt soit satisfait. De plus afin d'obtenir les meilleures performances, les décodeurs associés aux codes constitutifs des lignes et des colonnes du code produit doivent être à entrées et sorties pondérées :

- Un récepteur fonctionne avec des entrées pondérées lorsque le récepteur situé entre la sortie du canal de transmission et le décodeur de canal ne prend pas de décision ferme et définitive sur le symbole émis (e.g ; dans le cas d'un récepteur cohérent pour une modulation binaire type Q-PSK, l'information transmise au décodeur de canal correspond à la sortie du filtre adapté échantillonnée à la cadence du temps symbole). Sur un canal gaussien, le gain de codage peut être de l'ordre de 2 dB par rapport à un décodage dit à entrées fermes.
- Afin de pouvoir itérer le processus de décodage, les décodeurs relatifs aux codes élémentaires constitutifs du schéma de codage (ici un code produit) doivent associer à chacune de leur décision une mesure de fiabilité. De cette sortie pondérée sera alors extraite, pour chaque symbole, une information propre à la structure du code élémentaire indépendante de l'entrée du décodeur, appelée information extrinsèque qui sera transmise au décodeur suivant.

À la réception d'une matrice  $[R]$  correspondant à une matrice transmise  $[E]$ , le décodeur les lignes (ou colonnes), estime le  $LLR$  et génère la matrice de sortie du décodeur élémentaire  $[W]$ . Le processus de décodage décrit ci-dessus est alors réitéré. La (Fig. 4.2) ci-dessous décrit le processus itératif :

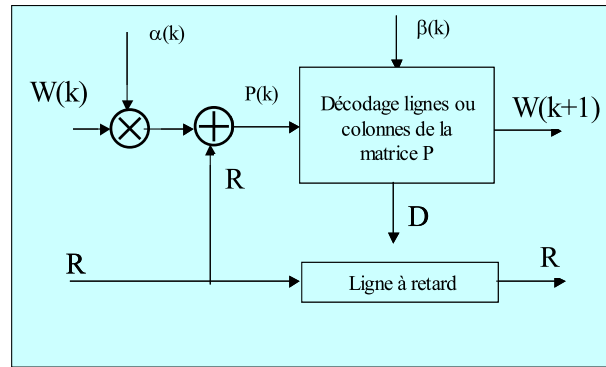


FIG. 4.2 — Schéma du décodeur élémentaire

- $k$  est le nombre d'itération
- $W$  représente la matrice d'information extrinsèque
- $\alpha(k)$  est un paramètre permettant de pondérer l'influence de l'information extrinsèque lors des premières itérations (sa valeur croît de la valeur zéro à la première itération pour atteindre la valeur de un à la dernière)
- $\beta(k)$  est une valeur affectée à l'information extrinsèque en présence de concurrent non trouvé, sa valeur dépend du rapport signal à bruit et est optimisé par simulation.
- l'entrée du décodeur élémentaire d'indice  $k$  répond à la relation suivante eq. (4.17) :

$$R(k) = R + \alpha(k)W(k) \quad (4.17)$$

### Conditionnement temps réel

L'algorithme de Chase Pyndiah a fait l'objet des premières implémentations et validations sur DSP. Compte tenu de la relative complexité et surtout du processus itératif mis en œuvre dans les systèmes turbo, l'obtention de bonnes performances passent par différentes étapes :

- l'élimination de toutes les redondances logicielles et calculatoires pour ce qui concerne une version en langage C de l'application.
- la réécriture en langage d'assemblage des parties critiques du code, notamment des boucles qui devront inévitablement faire appels aux parallélismes matériels (utilisation des différentes unités de calcul, selon le processeur utilisé) et logiciels (utilisation du "pipe lining" et des options "SIMD"<sup>3</sup>. Les boucles sont ainsi éclatées en trois parties à savoir le prologue, le cœur de boucle et l'épilogue.

La première implémentation a été réalisée sur une des premières générations de processeurs (Motorola 56002 : fréquence d'horloge 40 MHz, 20 MIPS). Le turbo décodeur a été entièrement écrit en langage d'assemblage et optimisé manuellement. Dédié au traitement du signal, ce processeur a des possibilités limitées en matière de parallélisme. Comparé à la version flottante en langage C, les dégradations obtenues sur la version fixe n'excèdent pas 0.5 dB pour un taux d'erreur de  $10^{-5}$ . Compte tenu de la fréquence d'horloge les performances restent modestes (160 Kbits/s après une demi-itération). Cette première implémentation a fait l'objet d'une publication à GLOBECOM'97 à Phoenix [12] À partir de 1997, à la suite d'une intégration au programme "Elite European"<sup>4</sup>, nous avons opté pour la famille de processeur TMS320C62xx. Ces nouveaux processeurs possèdent plusieurs unités de calcul (8 pour le 6201) travaillant en parallèle et une fréquence d'horloge de 200Mhz dans la version de base. La puissance de calcul crête offerte est de 1600 MIPS<sup>5</sup>. La plate forme de développement offre également des outils logiciels très performants autorisant une première validation des systèmes en langage C. Les outils de "monitoring" permettent facilement de détecter les parties critiques du code afin de procéder à des optimisations en langage d'assemblage. Au parallélisme matériel s'ajoute un parallélisme logiciel relativement délicat à mettre en œuvre. L'algorithme de Chase Pyndiah a été par la suite implémenté sur ce nouveau processeur dans le cadre de différents travaux et notamment dans le cadre de la thèse de Nadine Chapalain [3] soutenue le 14 octobre 2002.

Le turbo décodeur reçoit des blocs de 1024 échantillons, quantifiés sur 8 bits. le processus de turbo décodage passe par l'exécution de 4 tâches principales :

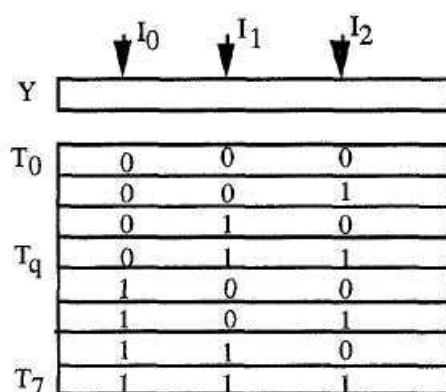
- **Décision Classement Parité** : à partir des échantillons  $R$  reçus, évaluer le vecteur de décision  $Y$ . la même boucle de traitement permet de trouver les  $m$  échantillons les moins fiables, de les classer et de calculer la parité du mot reçu après décision (Fig. 4.3).
- **Syndrome** : A partir d'une table pré calculée et stockée en mémoire, le décodeur détermine pour l'ensemble  $T^q$ , la position des bits faux et les corrige, en n'utilisant qu'une seule boucle de traitement<sup>6</sup>
- **Distance Euclidienne** : Cette procédure se réduit dans la pratique aux calculs des distances uniquement pour les bits différents. Celui à la distance minimale est le candidat, le second devient le concurrent.

<sup>3</sup>Single Instruction Multiple Data, opération permettant de multiplier le nombre d'opérations par temps cycle suivant la taille des opérandes

<sup>4</sup>Dans le cadre de ce programme, les Universités partenaires développent des applications sur DSP, les cartes et logiciels étant mis à disposition par Texas Instruments

<sup>5</sup>Une nouvelle version à 1.2 Giga Instructions par seconde est d'ores et déjà disponible

<sup>6</sup>valide uniquement pour le pouvoir de correction de  $t=1$

FIG. 4.3 — Création de  $Z^q = Y \oplus T^q$ 

- **Information extrinsèque** : le turbo décodeur ne fait les calculs que pour les positions ou les bits sont différents.

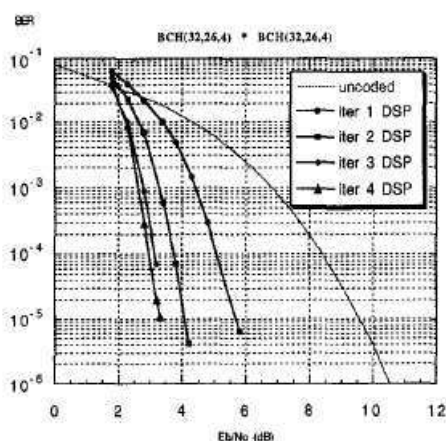


FIG. 4.4 — Taux d'erreur binaire en fonction du rapport Signal à Bruit après chaque itération

Les performances de l'algorithme ont été évaluées en utilisant les simulations de Monte Carlo pour différents  $E_b/N_0$ , le Taux d'Erreurs Binaires (TEB-BER) est évalué après chaque itération (nombre d'itération fixée à 8 soit 4 décodages ligne-colonne)(Fig. 4.4). Pour évaluer les performances sur DSP, le décodeur en langage C a été conditionné aux contraintes du DSP (Fig. 4.5) :

- échantillon d'entrée quantifié sur 8 bits
- prise en compte de la simplification algorithmique

Une version en langage C entièrement optimisée a été implémentée atteignant des débits de l'ordre de 400 Kbits/s (après 4 itérations). Une autre version, n'utilisant pas de parallélisme logiciel "Pipe lining", a été partiellement optimisée<sup>7</sup> en langage d'assemblage et atteint un débit de 500 kbits/s. Ces travaux ont fait l'objet de plusieurs publications [14], [15], [16].

<sup>7</sup>les nouveaux travaux en cours de développement utilisent ces remarquables possibilités

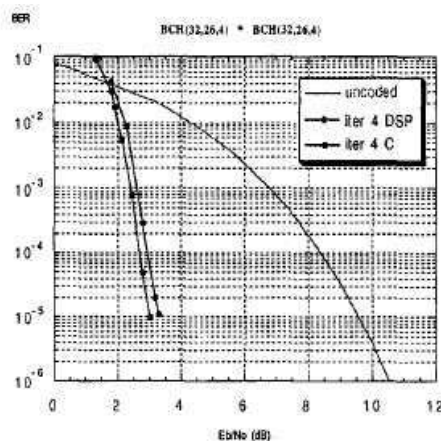


FIG. 4.5 — Comparaison des Performances entre les C et DSP versions

### 4.3.2 Décodage des Turbo Codes en blocs avec l'algorithme de Hartmann-Rudolph-Nazarov HRN

Dans le prolongement des travaux sur le turbo décodage des Codes en Blocs, le département Système et Communications s'est intéressé à une autre méthode de décodage des Codes en Bloc utilisant l'algorithme de Hartmann-Rudolph-Nazarov. Alors que l'algorithme de Chase-Pyndiah, qui est sous-optimal, génère après décodage une fiabilité sur les mots de codes, l'algorithme de Hartmann-Rudolph-Nazarov permet de calculer la fiabilité de chacun des bits des mots. L'objectif de l'étude était de comparer les deux algorithmes à la fois en terme de performances (Taux d'Erreur Binaire en fonction du rapport Signal à Bruit) et également en terme de débit binaire, facteur dépendant directement de la complexité. Comme on pouvait s'y attendre L'algorithme de Hartmann Nazarov converge plus rapidement que celui de Chase-Pyndiah mais son conditionnement temps réel est beaucoup plus lourd et demande un développement en langage d'assemblage pour approcher les résultats atteints par celui de Chase-Pyndiah. Cette étude a fait l'objet d'une présentation à ICC2002 (International Conference on Communications ) New York [17].

Cet algorithme optimal au sens de la minimisation de la probabilité d'erreur par symbole a été proposé par Nazarov [9] et récemment adapté au décodage itératif [17]. Le rapport de vraisemblance est calculé pour chaque décision  $d_j$  en balayant de manière exhaustive l'ensemble des mots de code de l'espace dual du code constituant. L'intérêt de cette méthode est d'utiliser la transformée rapide de Hadamard dont le nombre de points est déterminé par la dimension du code dual.

Les codes BCH binaires primitifs capables de corriger 1 erreur (i.e. les codes de Hamming) sont particulièrement bien adaptés à ce type de décodage en raison des bonnes propriétés de la matrice de contrôle de parité  $H$ . On réalise en effet des transformées de Hadamard sur  $2^{n-k}$  valeurs. Ces valeurs sont partout nulles sauf aux  $n$  points correspondants aux représentations binaires des colonnes de  $H$ . Or, pour un code de Hamming ces  $n$  colonnes sont chacune une représentation binaire de 1 à  $2^{n-k} - 1$ . L'algorithme est alors très efficace et d'une complexité de calcul supérieure quoique de l'ordre de celle de l'algorithme précédent. Surtout, c'est un algorithme exact au sens de la propagation des

probabilités et ne nécessite plus de coefficient empirique.

### L'algorithme de Hartmann-Rudolph-Nazarov

Cet algorithme permet d'obtenir un décodage optimal symbole par symbole des Turbo Codes en blocs en utilisant le code dual L'utilisation de la transformée rapide de Hadamard donne de bonnes performances dans des environnements temps réel.

Soit  $m = (m_i; 0 \leq i < k)$  les symboles d'informations du mot de code  $B = (b_l; 0 \leq l < n)$  d'un code élémentaire  $\mathcal{C}$  de matrice génératrice  $G = (g_{il}; 0 \leq i < k, 0 \leq l < n)$ , et de matrice duale  $H = (h_{il}; 0 \leq i < n - k, 0 \leq l < n)$  et  $R = (r_i; 0 \leq i < n)$  le mot reçu. Les composantes des mots de valeurs  $\{0, 1\}$  sont transformées en  $\{-1, +1\}$  avant transmission.

Le décodeur calcule la probabilité *a posteriori*  $\Pr(b_l|R)$  pour chaque composante  $b_l$  du mot transmis. Sur un canal sans mémoire, l'algorithme de décodage optimal symbole par symbole peut être écrit sous la forme suivante :

$$C(l) = \Pr(b_l = 0|R) - \Pr(b_l = 1|R) \quad (4.18)$$

Nazarov a exprimé  $C(l)$  sous la forme suivante :

$$C(l) = \frac{1}{\sum_{\gamma=0}^{2^{n-k}-1} D_\gamma} \sum_{\gamma=0}^{2^{n-k}-1} D_\gamma \times \Lambda_l(\gamma) \quad (4.19)$$

ou

$$\Lambda_l(\gamma) = \begin{cases} \rho_l & \text{if } t_\gamma(l) = 0, \\ \frac{1}{\rho_l} & \text{if } t_\gamma(l) = 1. \end{cases} \quad (4.20)$$

$$D_\gamma = \prod_{l=0}^{n-l} \rho_l^{t_\gamma(l)} \quad (4.21)$$

et

$$t_\gamma(l) = \sum_{m=0}^{n-k-1} \gamma_m \times h_{ml} \text{ if } y = \sum_{m=0}^{n-k-1} \gamma_m \times 2^m \quad (4.22)$$

L'ensemble  $h_{ml}$  représente les coefficients de la matrice du code dual  $H$ , avec  $m$  comme indice de ligne et  $l$  comme indice colonne. La composante  $\rho_l$  est définie de la façon suivante :

$$\rho_l = \frac{p(r_l, b_l = 0) - p(r_l, b_l = 1)}{p(r_l, b_l = 0) + p(r_l, b_l = 1)} \quad (4.23)$$

Il faut noter sur ce point que cette dernière expression diffère de celle  $\rho_l$  donné dans la référence [?]. En effet c'est la densité de probabilité conjointe  $p(r_l, b_l)$  qui apparaît maintenant au lieu de la densité de probabilité conditionnelle. Pour prendre en compte la ré estimation de la probabilité *a priori* des bits au cours du décodage, l'expression des  $\rho_l$

pour prendre en compte à la fois la contribution de la sortie du canal et de l'information *a priori*.

Donc, en introduisant le logarithme du rapport de vraisemblance conditionnelle,  $LLRi_l^{(0)}$  défini de la façon suivante :

$$LLRi_l^{(0)} = L(r_l|b_l) = \ln \left( \frac{p(r_l|b_l = 1)}{p(r_l|b_l = 0)} \right) = 2r_l/\sigma^2 \quad (4.24)$$

ou  $\sigma$  représente l'écart type de l'observation  $R$ , l'expression de  $\rho_l$  avant décodage peut être réécrite comme suit (les bits sont supposés équiprobables) :

$$\rho_l = \frac{1 - \exp(L(r_l|b_l))}{1 + \exp(L(r_l|b_l))} = -\tanh \left( \frac{LLRi_l^{(0)}}{2} \right) \quad (4.25)$$

En remplaçant  $LLRi_l^{(0)}$  par sa valeur  $(2/\sigma^2) \times r_l$ , on obtient

$$\rho_l = -\tanh \left( \frac{r_l}{\sigma^2} \right) \quad (4.26)$$

La modification de l'expression de  $\rho_l$  permet maintenant d'introduire l'information extrinsèque comme une information *a priori*. Après le premier décodage de lignes et de colonnes, on ajoute à l'argument de la fonction  $\tanh()$  l'information extrinsèque. Les composantes  $\rho_l$  étant connues ( $l = 0, \dots, n-1$ ) deux transformées rapides de Hadamard (FHT : Fast Hadamard Transform)  $F_\rho$  and  $F_q$  définies ci-dessous peuvent être calculées.

– Calcul de  $F_\rho$

$$F_\rho(v) = \sum_{l=0}^{n-1} \ln(|\rho_l|) \exp \left( j\pi \sum_{m=0}^{n-k-1} v_m h_{ml} \right) \quad (4.27)$$

$F_\rho$  peut également être écrite suivant l'expression suivante :

$$F_\rho(v) = \sum_{p=0}^{2^{n-k}-1} f_\rho(p) (-1)^{\langle p, v \rangle} \quad (4.28)$$

avec,

$$f_\rho(p) = \begin{cases} \ln(|\rho_l|) & \text{if } p = \sum_{m=0}^{n-k-1} h_{ml} 2^m, \\ 0 & \text{otherwise.} \end{cases} \quad (4.29)$$

$p$  représente la somme modulo 2 des différentes colonnes de la matrice du code dual, la fonction,  $F_\rho$  peut ainsi être obtenue en calculant la FHT de la fonction  $f_\rho$ .

– Calcul de  $F_q$

$$F_q(v) = \sum_{l=0}^{n-1} q(l) \exp \left( j\pi \sum_{m=0}^{n-k-1} v_m h_{ml} \right) \quad (4.30)$$

ou  $\text{sign}(\rho_l) = (-1)^{q_l}$ .  $F_q$  peut également être écrite en utilisant la définition de la FHT,

$$F_q(v) = \sum_{p=0}^{2^{n-k}-1} f_q(p) (-1)^{\langle p, v \rangle} \quad (4.31)$$



avec,

$$f_q(p) = \begin{cases} q_l & \text{if } p = \sum_{m=0}^{n-k-1} h_{ml} \times 2^m, \\ 0 & \text{otherwise.} \end{cases} \quad (4.32)$$

Ainsi,  $F_q$  est la FHT de la fonction  $f_q$ .

A partir des deux FHT précédemment calculées, un paramètre intermédiaire  $D_\gamma$  peut être calculé par la formule suivante,

$$D_\gamma = \exp \left[ \frac{1}{2} (F_\rho(0) - F_\rho(\gamma)) \right] \cdot \exp \left[ \frac{1}{2} j\pi (F_q(0) - F_q(\gamma)) \right] \quad (4.33)$$

alors,

$$F_D(h_l) = \sum_{\gamma=0}^{2^{n-k}-1} D_\gamma \exp \left( j\pi \sum_{m=0}^{n-k-1} \gamma_m \times h_{ml} \right) \quad (4.34)$$

Par la suite, le paramètre  $C(l)$  utilisé pour calculer l'information extrinsèque à la sortie du décodeur peut être déterminé à partir de la dernière FHT  $F_D(h_l)$  de la façon suivante :

$$C(l) = \frac{1}{2} \left[ \rho_l \left( 1 + \frac{F_D(h_l)}{F_D(0)} \right) + \frac{1}{\rho_l} \left( 1 - \frac{F_D(h_l)}{F_D(0)} \right) \right] \quad (4.35)$$

avec  $l$  la position du bit décodé dans le mot de code  $B$ . Finalement l'information extrinsèque est définie par :

$$LLRo_l = \ln \left( \frac{P_r(b_l = 1|R)}{P_r(b_l = 0|R)} \right) = \ln \left( \frac{1 - C(l)}{1 + C(l)} \right) \quad (4.36)$$

### Le décodage itératif

On peut montrer que la sortie pondérée  $LLRo_l^k$  du décodeur à l'itération  $k$ -th peut être réécrite de la façon suivante :

$$LLRo_l^k = \ln \left( \frac{p(r_l, b_l = 1)}{p(r_l, b_l = 0)} \right) + W_l^{k+1} = LLRi_l^k + W_l^{k+1} \quad (4.37)$$

ou les symboles  $b_l$  sont supposés être équiprobables,

$$W_l^{k+1} = \ln \left( \frac{\rho_l (F_D(0) + F_D(h_l)) - (F_D(0) - F_D(h_l))}{\rho_l (F_D(0) + F_D(h_l)) + (F_D(0) - F_D(h_l))} \right) \quad (4.38)$$

Le terme additionnel  $W_l^{k+1}$ , concrètement l'information extrinsèque, peut être obtenue soit en soustrayant l'information pondérée d'entrée  $LLRi_l^k$  à l'information pondérée de sortie  $LLRo_l^k$  ou directement calculé à partir de eq. (4.38). Ce terme joue un rôle prépondérant dans le décodage itératif. En effet l'index  $k + 1$  indique qu'il sera utilisé comme information d'entrée *a priori* dans l'étage suivant de décodage des colonnes (ou lignes). Donc l'information pondérée d'entrée correspondant à la composante  $l$  de la colonne (ou de la ligne) devant être décodé à la prochaine itération ( $(k + 1)^{ieme}$ ) est donnée par  $LLRi_l + W_l^{k+1}$  (notons que  $W_l^0 = 0, \forall l \in \{1, \dots, n\}$ ). Généralement le processus de décodage est obtenu en cascade de décodeurs élémentaires décrits à la Fig. 4.6.

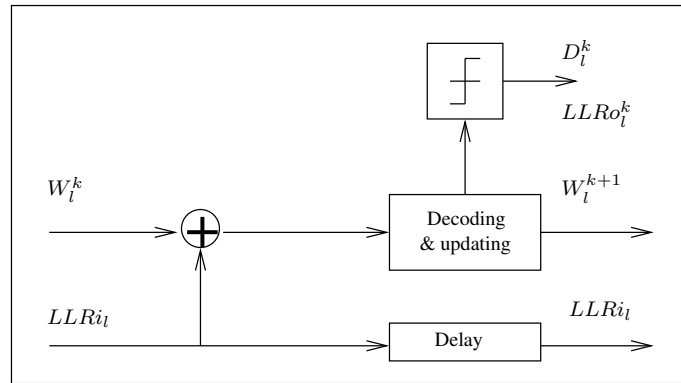


FIG. 4.6 — le décodage itératif

### Le conditionnement temps réel

L'algorithme de décodage symbole par symbole inclus un certain nombre de fonctions mathématiques pour générer l'information extrinsèque. Parmi celles-ci, un certain nombre peuvent poser quelques problèmes dans le cadre de fortes contraintes temps réel :

- la tangente hyperbolique,
- le logarithme Népérien,
- la fonction exponentielle,
- la division

Cet algorithme peut être relativement facilement conditionné en virgule fixe si des contraintes temps réel l'exigent. Différentes techniques peuvent être mises en œuvre pour résoudre ces problèmes de conditionnement :

- **approximation linéaire de la fonction** : Cette technique peut être utilisée pour le calcul de la fonction  $\tanh$ . Cela permet un calcul simple et rapide, évitant du même coup de longues recherches dans des tables pré calculées. L'algorithme HRN [9] demande une bonne estimation de la fonction à la fois pour les petites valeurs voisines de zéro et pour les grandes valeurs positives ou négatives, donnant des valeurs proches de +1 ou de -1. Notre conditionnement temps réel utilise l'approximation suivante :

$$\tanh(x) = \begin{cases} 0.8|x| & \text{si } |x| \leq 1.0 \\ 0.165|x| + 0.635 & \text{si } 1 < |x| \leq 2.0 \\ 0.005|x| + 0.9555 & \text{si } |x| > 2.0 \end{cases}$$

- **tables pré-calculées et calcul direct d'offset** : Pour éviter de longs calculs de fonctions, de longues recherches dans des tables et maintenir des performances proches de celles de l'algorithme tournant en langage de haut niveau, on peut utiliser des tables pré-calculées. On peut calculer  $2^n$  points de la fonction (n étant le nombre de bits de quantification). Cette technique est utilisée pour le calcul du logarithme Népérien. L'offset peut être calculé rapidement et nécessite l'emploi de deux tables suivant l'amplitude de l'entrée :
  - $0 < x \leq 1$  : le segment  $[\epsilon, \dots, 1]$  ( $\epsilon \neq 0$ ) est divisé en  $2^n$ , donnant ainsi  $2^n$  niveaux de quantification. Les niveaux sont pré calculés et stockés dans un tableau. Comme la valeur de x est bornée à un, le calcul de l'offset se fait simplement en multipliant x par  $2^n$  et en prenant la partie entière.

- $x > 1$  dans ce cas il faut connaître la dynamique de  $x$ . Si  $x_{\max}$  est la valeur maximum de  $x$ , il suffit alors de normaliser la valeur de  $x$  par  $x_{\max}$  pour retrouver le cas précédent.
- **Réécriture partielle de l'algorithme pour éviter la division** : L'obtention de  $C(l)$  résulte d'une division et donc pose des problèmes dans le cadre d'applications à fortes contraintes temps réel. Dans ce cas il est possible de réécrire l'expression de l'information extrinsèque  $LLRo_l$  eq. (4.15) de la façon suivante :

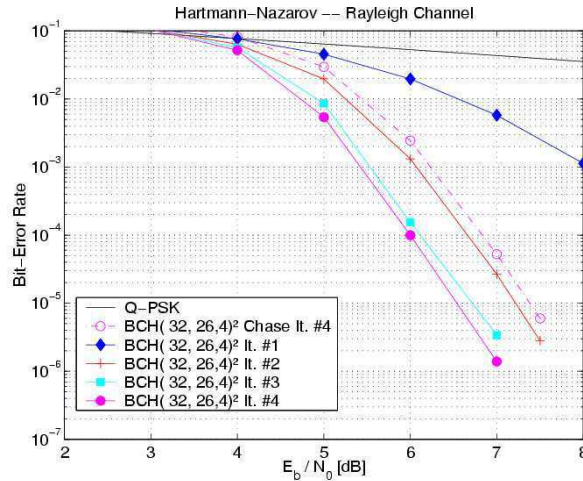
$$\begin{aligned} LLRo_l &= \ln \left( \frac{P_r(b_l = 1|R)}{P_r(b_l = 0|R)} \right) = \ln \left( \frac{1 - C(l)}{1 + C(l)} \right) = \ln \left( \frac{a(l) - b(l)}{a(l) + b(l)} \right) \\ &= \ln(|a(l) - b(l)|) - \ln(|a(l) + b(l)|) \quad (4.39) \end{aligned}$$

avec

$$a(l) = 2\rho_l F_D(0) \quad (4.40)$$

$$b(l) = \rho_l^2 (F_D(h_l) + F_D(0))(F_D(h_l) - F_D(0)) \quad (4.41)$$

Le calcul de l'information extrinsèque se ramène à la différence de 2 logarithmes.



**FIG. 4.7** — Comparaison des algorithmes de Hartmann-Rudolph-Nazarov et de Chase Pyndiah sur canal de Rayleigh. 4 itérations.

## Performances

Avec l'évaluation symbole par symbole de l'information extrinsèque, l'algorithme de Hartmann-Rudolph-Nazarov surpasse l'algorithme de Chase-Pyndiah [17] (Fig. 4.7). L'effet turbo démarre plus tôt et la vitesse de convergence est deux fois plus rapide mais objectivement la complexité calculatoire est supérieure. De plus quand on compare deux algorithmes, il est également intéressant de comparer les consommations énergétiques susceptibles de limiter le nombre d'itérations. Le Turbo Code choisi est le code produit BCH(32,26,4) × BCH(32,26,4) de rendement 0.66, comprenant donc 676 bits d'information sur un total de 1024 bits. Ce choix apporte une souplesse au niveau de la taille du bloc élémentaire. Le nombre d'itérations sur la courbe suivante est de 4 (une itération

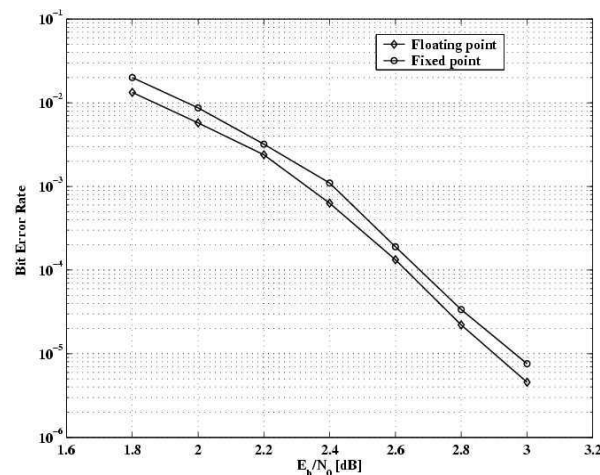


Figure 10-4 - Taux d'erreur binaire en fonction du Rapport Signal à Bruit sur canal de Gauss. 4 itérations.

FIG. 4.8 — Taux d'erreur binaire en fonction du Rapport Signal à Bruit sur canal de Gauss. 4 itérations.

comporte un décodage ligne suivi d'un décodage colonne). Dans le but d'évaluer les performances de l'algorithme en terme de taux d'erreur binaire, le programme a été adapté pour simuler un processeur à virgule fixe (entrées sur 8 bits signés, approximation linéaire, tables pré calculées et suppression de la division). La (Fig. 4.8) montre que l'implémentation à virgule fixe n'est qu'à 0.2 dB de l'implémentation à virgule flottante. Une version non entièrement optimisée atteint le débit binaire de  $550\text{ kbits/s}$  sur un processeur TMS320C6201 (puissance crête 1600 MIPS. Cette puissance est portée à 12000 MIPS sur la nouvelle série de processeur TMS320C64xx. Ceci permet d'envisager des applications à des débits relativement élevés  $5\text{ Mbits/s}$  avec en plus l'évolutivité rapide du système via le téléchargement.

## Conclusions-Perspectives

Malgré les relatives complexités des algorithmes étudiés, de très bonnes performances ont été obtenues en terme de TEB, proches des versions en langage C avec calcul flottant. Il est évident que des efforts important sont à mettre en pour passer d'une version C à une version implémentable sur un processeur à virgule fixe. Ils passent la plupart du temps par un re conditionnement de l'algorithme : suppression de toutes les redondances calculatoires, utilisation de tables pré calculées, linéarisation des fonctions, etc. L'optimisation de l'implémentation passe également par la prise en compte de l'architecture du processeur :

- possibilité de parallélisme matériel, logiciel,
- présence d'unité multiple de traitement, d'instructions spécifiques type SIMD<sup>8</sup>

nos performances en matière de débit obtenu, nous incitent à l'optimisme. Avec  $550\text{ kbits/s}$  à 1600 MIPS, le chiffre de  $5\text{ Mbits/s}$  sera atteint à 12000 MIPS d'autant plus que les registres du CPU sont doublés et que les instructions SIMD étendus. L'option DSP est un atout dans le contexte évolution rapide des systèmes de Télécommunications par la possibilité de téléchargement des dernières versions logicielles.

<sup>8</sup>Single Instruction Multiple Data

## 4.4 Les Turbo Codes en Blocs de type RS Reed-Solomon

Le travail de thèse de Rong ZHOU a porté sur l'étude des Turbo Codes en Blocs construits à partir de codes de Reed-Solomon. Les premiers schémas de Turbo Codes en Blocs employaient initialement des codes constituants de type BCH. L'emploi de codes RS en lieu et place des codes BCH permet d'approcher de très près la limite de Shannon et se révèle particulièrement intéressant pour les applications nécessitant des rendements de codage élevés ( $> 0.8$ ) ou bien des blocs d'information de petite taille (100 bits ou moins). Entre autres avantages, ce schéma de codage présente une faible complexité ce qui en fait un bon candidat pour une implantation matérielle capable d'opérer à très haut débit. Ce travail de thèse était initialement motivé par la remarque suivante : à paramètres équivalents (longueur de code et rendement), les codes RS présentent une plus grande distance minimale que leurs homologues BCH, et donc un gain de codage plus important. En contrepartie, le décodage s'avère plus coûteux (en terme de nombre d'opérations à effectuer) dans le cas des codes RS. Il nous est donc apparu qu'il y a un compromis à rechercher entre bonnes performances et complexité raisonnable.

### 4.4.1 Les Codes de Reed-Solomon

Les codes de Reed-Solomon forment l'une des familles de codes algébriques parmi les plus puissantes connues à ce jour. On les retrouve utilisés à ce titre dans de nombreux systèmes de traitement (transmission ou stockage) numérique.

Parmi les propriétés les plus remarquables, nous pouvons notamment relever leur appartenance à la famille des codes à distance maximale séparable (codes MDS). Ainsi, à longueur de code et rendement de codage fixé, les codes RS offrent la distance minimale la plus grande possible. Leur capacité à lutter efficacement contre les rafales d'erreurs constitue un autre avantage clé de ces codes. Pour toutes ces raisons, les codes RS ont trouvé de nombreuses applications dans les systèmes existants : disque compact, DVD, communications satellites, missions en espace lointain, transmissions sans fils, technologies xDSL.

Cette large diffusion des codes RS est également due en grande partie à l'existence d'algorithmes de décodage à entrées fermes efficaces et simples à implémenter, pouvant opérer aujourd'hui à de très haut-débits (de l'ordre de plusieurs Giga bits par seconde). Le décodage se ramène à la résolution d'un système d'équations linéaires dont les solutions sont à valeur dans le corps de Galois. En pratique, la résolution de ce système fait intervenir plusieurs étapes successives :

1. calcul des syndromes
2. recherche de la position des erreurs
3. calcul de la valeur des erreurs et application de la correction

La recherche de la position des erreurs représente généralement l'opération la plus coûteuse dans le décodeur. Celle-ci peut être menée efficacement à l'aide de l'algorithme de Peterson pour les faibles pouvoirs de correction, l'algorithme itératif proposé par E. Berlekamp et J. Massey offre une extension pour des pouvoirs de corrections plus élevés

mais au prix d'une complexité nettement plus importante. Des modifications sont à apporter à la procédure de décodage en présence d'effacements sur le message reçu.

Ces algorithmes garantissent la correction d'un nombre limité d'erreurs, au plus égal à la capacité de correction du code. Une manière d'améliorer les performances globales du décodage consiste à opérer non plus sur des décisions fermes à la sortie du démodulateur, mais sur des décisions souples (ou pondérées). Les algorithmes optimaux proposés jusqu'à ce jour sont généralement trop complexes à mettre en œuvre dans le cas des codes RS. On doit alors se tourner vers des solutions sous-optimales mais de moindre complexité, à l'exemple de l'algorithme de décodage proposé par D. Chase.

#### 4.4.2 Décodage à Entrée Souple des Codes RS par l'algorithme de Chase

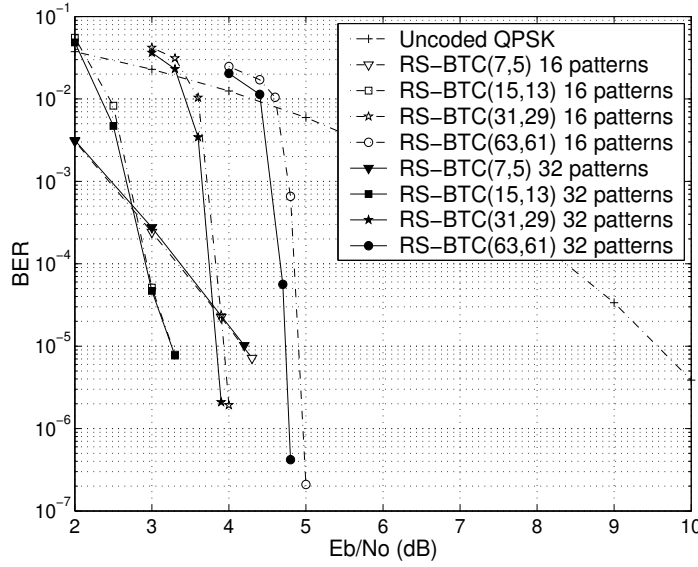
La chaîne de transmission se compose principalement d'un codeur RS suivi d'un modulateur linéaire. Nous nous intéressons ici non seulement à une modulation binaire (MDP-4), mais également à des modulations d'ordre plus élevé (modulations MAQ- $M$ ), dans un souci d'optimisation de l'efficacité spectrale du système. Dans ce dernier cas de figure, l'association entre le code et la modulation est réalisée de manière pragmatique, en découplant l'optimisation du code de l'optimisation de la règle de codage des symboles modulés, ce qui nous conduit naturellement à retenir un code de Gray. L'algorithme de Chase consiste à rechercher le mot de code à distance euclidienne minimale de l'observation reçue en procédant par décodages fermes successifs d'une série de séquences candidates judicieusement choisies. On peut envisager plusieurs niveaux d'implémentation de cet algorithme, suivant la nature des séquences candidates considérées. Trois solutions sont possibles : une implémentation au niveau binaire, une implémentation au niveau symbole de modulation, et enfin une implémentation au niveau des symboles du code. Au final, il ressort de l'étude des trois implémentations possibles de l'algorithme de Chase que le décodage au niveau binaire offre le meilleur compromis entre complexité et performance. C'est la solution qui a été retenue dans la suite des travaux.

#### 4.4.3 Les Turbo-Codes en Blocs Reed-Solomon

Dans ce qui suit on se limite aux cas où les deux codes constituants sont identiques  $\mathcal{C}_1 = \mathcal{C}_2 = \text{RS}(n, k, \delta)$  définis sur un corps de Galois à  $Q = 2^q$  éléments. Deux types de concaténation : la concaténation bit et la concaténation symbole sont proposés. Dans le cas de la concaténation bit la matrice codée est de longueur  $(nq)^2$  tandis qu'elle est égale à  $(n^2q)$  dans le cas symbole.

L'algorithme de décodage itératif des codes produits à éléments binaires Chase-Pyndiah est ensuite étendu au cas des codes à éléments non-binaires de type RS. Le décodage de Chase au niveau bit est retenu car il offre le meilleur compromis entre complexité et performance et de plus il simplifie l'extension de l'algorithme Chase-Pyndiah aux codes non-binaires.

Pour un pouvoir de correction  $t = 1$ , il n'est pas avantageux d'utiliser la version étendue (ou expurgée) des codes constituants RS contrairement aux codes BCH.



**FIG. 4.9** — RS( $t=1$ )-TCB avec concaténation bit modulation MDP-4 canal BBAG, Bruit Blanc Additif Gaussien

En effet, dans le cas de codes RS( $t = 1$ ) le nombre de symboles de redondances est égal à  $2t = 2$  et le rajout d'un symbole de redondance supplémentaire augmente de 50% la redondance ce qui se traduit par une réduction significative du rendement de codage. Ceci se traduit par une diminution de la limite de Shannon d'environ 0.6 dB au lieu de 0.2 dB dans le cas BCH. D'autre part, les deux codes donnent les mêmes performances pour un  $TEB > 10^{-7}$  à même complexité de décodage ce qui se traduit par un écart plus important par rapport à la limite de Shannon. Enfin, le code produit RS présente un gain asymptotique de codage plus élevé que pour le code BCH grâce à la propriété MDS. Ainsi, le TCB-RS dispose d'une plus grande robustesse au phénomène de plancher d'erreur, contrairement aux TCB-BCH, ce qui conduit à utiliser les codes constituants RS non-étendus (et non-expurgés). Les performances de différents TCB-RS associés à la modulation MDP-2 (voire MDP-4) sur canal BBAG sont présentées à la figure (Fig. 4.9). Une comparaison des codes produits RS avec concaténation bit et symbole montre que la concaténation symbole est plus intéressante car elle est moins complexe à décoder et pour des blocs codés  $q$  fois plus petits. Les figures (Fig. 4.10), (Fig. 4.11) et (Fig. 4.12) présentent respectivement les performances sur le canal de Rayleigh, sur le canal (BSC) et sur le canal à effacement (BEC). Le TCB-RS est très efficace sur les canaux de type CBS et effacement ce qui ouvre des perspectives intéressantes pour des applications comme la transmission sur fibre optique ou le stockage de données pour lesquelles la sortie du canal est généralement binaire voire à effacement. Par contre, on peut envisager des améliorations du décodage itératif dans le cas de codes produits RS sur le canal de Rayleigh. Le tableau 4.1 montrent les écarts entre les TCB RS à entrées pondérées et les entrées de type binaires (Canal Binaire Symétrique BSC) ou ternaires (Canal à Effacement BEC). Les écarts sont mesurés en calculant les différences des  $E_b/N_o$  au TEB de  $= 10^{-5}$ , ils sont notés  $\Delta_{BSC}$  ( $\Delta_{BEC}$ ) pour les écarts entre les entrées pondérées et les

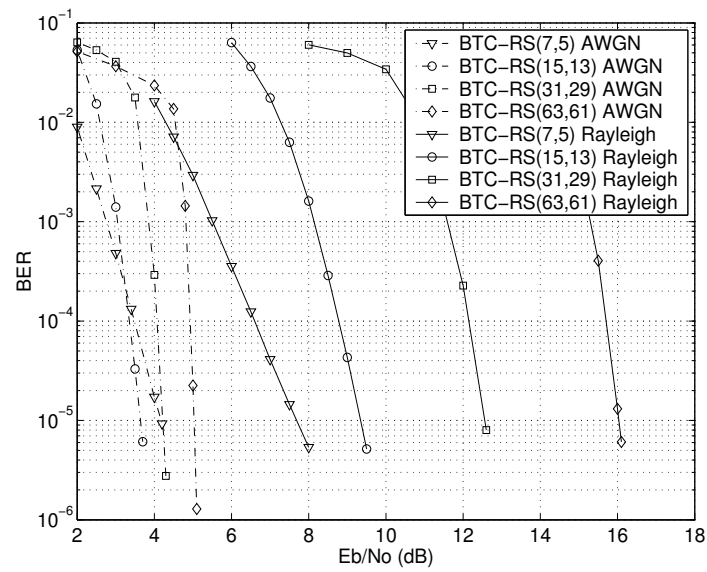


FIG. 4.10 — RS( $t=1$ )-BTC avec concaténation au niveau symbole  $Q$ -ary utilisant une modulation MDP4 sur canaux BBAG et Rayleigh à évanouissements

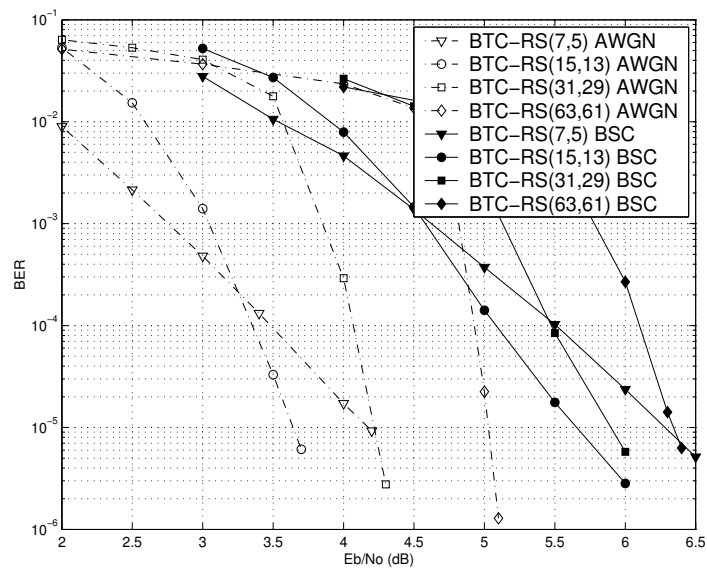


FIG. 4.11 — RS( $t=1$ )-TCB sur canal BBAG et canal CBS utilisant 16 patterns de test



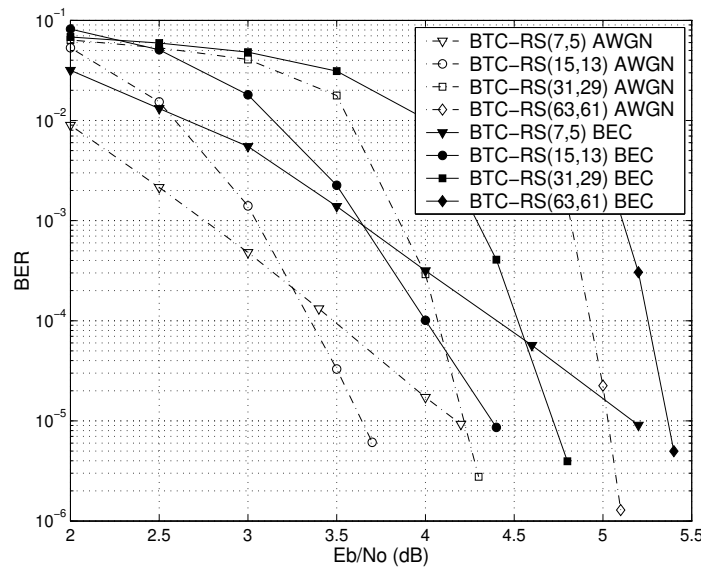


FIG. 4.12 — RS-BTC ( $t=1$ ) sur canal BBAG et Canal Binaire à Effacement (CBE) utilisant 16 patterns de test

entrées binaires et ternaires. Les faibles dégradations du TBC RS ( $t=1$ ) à la fois sur le canal binaire et ternaire les rendent tout à fait adaptés pour les systèmes à hauts rendements tels que les transmissions optiques et le stockage magnétique

TAB. 4.1 — Ecart entre entrées pondérées et entrées binaires et ternaires

	$(Eb/No)_{AWGN}$	$(Eb/No)_{BSC}$	$\Delta_{BSC}$	$(Eb/No)_{BEC}$	$\Delta_{BEC}$
RS-BTC(7, 5)	4.19 dB	6.30 dB	2.11 dB	5.18 dB	0.99 dB
RS-BTC(15, 13)	3.64 dB	5.66 dB	2.02 dB	4.38 dB	0.74 dB
RS-BTC(31, 29)	4.23 dB	5.90 dB	1.67 dB	4.72 dB	0.49 dB
RS-BTC(63, 61)	5.03 dB	6.36 dB	1.33 dB	5.37 dB	0.34 dB

Ces différents résultats ont fait l'objet de cinq publications en conférence : [28], [27], [29], [30], [32].

## 4.5 Comparaison TCB-BCH et TCB-RS

Les performances des TCB-RS ( $t = 1$ ) avec concaténation symbole sont comparées au TCB-BCH de même rendement [21]. Cette comparaison utilise un bloc d'information de taille  $K$ , un rendement  $R$  et l'écart à la limite de Shannon  $\Delta Sh$  comme paramètres. Le tableau 4.2 [29] présente les différentes valeurs.

Il apparaît que tous les TCB du tableau (BCH et RS) peuvent assurer des communications fiables à moins de 1 dB de la limite de Shannon avec une modulation MDP4 sur

TAB. 4.2 — Comparaison TCB BCH et TCB-RS modulation MDP4 Canal Gaussien

	$K$ (bits)	$R$	$\Delta SH$ (dB)
RS-BTC(15, 13)	676	0.75	0.80
BCH-BTC(64, 57)	3249	0.79	0.88
RS-BTC(31, 29)	4205	0.88	0.80
BCH-BTC(128, 120)	14400	0.88	0.83
BCH-BTC(256, 239)	57121	0.87	0.90
RS-BTC(63, 61)	22326	0.94	0.80
BCH-BTC(256, 247)	61009	0.93	0.70

canal Gaussien.

Une comparaison entre TCB-RS et TCB-BCH pour un rendement de codage équivalent (0.94) montre un net avantage en faveur des TCB-RS car la longueur des codes est plus faible dans un rapport supérieur à 2.5 (24,000 bits au lieu de 64,000 bits). Ceci se traduit par une diminution de la mémoire du circuit turbo décodeur qui peut représenter jusqu'à 80% de la complexité du circuit. Ainsi la dimension des mémoires et le délai de codage/décodage est réduit dans un facteur 2.5 ce qui est un très net avantage en faveur du TCB-RS. Ceci constitue une première contribution originale du travail.

### 4.5.1 Conclusions

Les TCB utilisant les codes de type BCH à éléments binaires sont devenus une technologie éprouvée dix ans après leur introduction en 1994. C'est une solution complémentaire aux TCC et sont très efficaces pour des rendements de codage élevés ( $R > 0.8$ ). Ils sont déjà adoptés pour le WiMax et ils sont considérés pour de nombreuses applications telles que le stockage de données, le Digital Subscriber Line et la transmission haut débit sur fibre optique. Le désavantage majeur des TCB-BCH demeure la très grande dimension des blocs codés ( $>60,000$  bits) pour atteindre les rendements élevés ( $R > 0.9$ ) et l'écart croissant par rapport à la limite de Shannon lorsqu'on augmente le nombre d'états de la modulation.

La première motivation pour étudier les TCB-venait de sa propriété MDS (Maximum Distance Separable) qui stipule que pour une longueur de code et une distance minimale (Hamming) données, le RS offre le plus grand rendement. En utilisant l'algorithme Chase/Pyndiah nous avons simulé les performances de différents TCB construits avec différents codes constituants RS afin de trouver le meilleur compromis entre complexité et performance. La complexité du turbo décodeur a été limitée à 16 séquences tests dans le Chase/Pyndiah et 8 itérations. En utilisant l'écart des performances par rapport à leur limite de Shannon (avec bloc fini), on peut faire les remarques suivantes :

1. Il n'y avait aucun avantage à utiliser les version étendus (ou expurgés) des codes constituants RS comme dans le cas de BCH binaires.
2. La concaténation bit et symbole Q-aire donnaient des écarts très proches par rapport

à leur limite de Shannon.

Une autre caractéristique des TCB-RS vient du fait qu'ils sont capables de délivrer des performances proches de la limite de Shannon sur des canaux à sorties non-continues (CBS et à effacement). Bien que les entrées soient binaires (ou ternaires), la convergence de l'information extrinsèque des bits d'information et des bits de parité vers une distribution Gaussienne procure des performances de décodage quasi-optimales. C'est une caractéristique très intéressante pour les communications à très haut débit ( $> \text{Gbps}$ ) où les données pondérées en sortie du démodulateur ne sont pas disponibles pour des raisons techniques et/ou économiques. Les applications potentielles sont : les communications optiques à très haut débit ou le stockage de données.

les TCB-RS sont plus efficaces que les TCB-BCH pour des rendements de codage équivalents et associés à des modulations à grand nombre d'états. D'autre part, la taille des codes produits RS est 2.5 fois plus petite et donc le TCB-RS est un excellent candidat pour les applications à grande efficacité spectrale telles que : le DSL, le DVB, le WLAN et autres.

En conclusion le TCB-RS, utilisant des codes constituants RS avec ( $t = 1$ ) et une concaténation symbole, est une solution efficace pour les applications à rendements élevés. Les applications potentielles sont : le DSL, les communications optique, le DVB, le WLAN et le stockage de données ...

## 5.1 Introduction

Les avancées technologiques récentes en électronique ont permis une augmentation considérable de la puissance de calcul des systèmes numériques de traitement de l'information. Ces avancées ont ouvert et ouvrent la voie à de nouvelles applications, comme les communications mobiles (Inmarsat, GSM et UMTS), les réseaux d'accès sans fils (WiFi et WiMax), ou encore de l'internet haut-débit sur la ligne téléphonique (ADSL), ainsi que la télévision numérique terrestre (TNT, ou DVB-T en anglo-américain).

A ce jour, il est évident que tous les systèmes ou toutes les normes n'ont pas intégré les possibilités de correction d'erreur des Turbo Codes. De nouvelles applications hors standards sont présentées, elles proposent d'étendre le champ d'application des Turbo codes.

1. Réseau Indoor - Canal 60 Ghz [37], pour cette application nous avons proposé l'utilisation de l'algorithme Hartmann-Rudolph-Nazarov (voir Chapitre 4) pour le décodage des Turbo codes produits de type BCH. Cette étude s'est déroulée dans le cadre des projets GET incitatifs<sup>1</sup>. Pour cette application, c'est une implémentation circuit qui doit être envisagée et éventuellement l'utilisation de l'algorithme de Chase-Pyndiah.
2. Proposition d'introduction de Code BCH et Turbo Codes en blocs dans BLUE-TOOTH [39], liaison radio courte distance destinée à remplacer les câbles de connexions réseaux. Le cadre de l'étude est le même que le cas précédent, les options circuits et processeurs de signaux peuvent ici être envisagées.
3. Codage canal Pour les Nouvelles Modulations VLF/LF [38]. Cette application s'est déroulée dans le cadre d'un contrat d'étude avec un partenaire extérieur. Les débits utilisés vont de 300 à 1500 bits/s. L'exigence du client (absence de brevet) nous a conduit à utiliser l'algorithme Hartmann-Rudolph-Nazarov pour le décodage des Turbo codes produits de type BCH. Le système de décodage a été implémenté et validé sur un processeur à virgule flottante<sup>2</sup>

---

<sup>1</sup>Groupe des Ecoles des Télécommunications, projets innovants développés sur plusieurs écoles

<sup>2</sup>Texas Instruments TMS320C6701

## 5.2 Réseau Indor - Canal 60 GHz

### 5.2.1 Introduction

Nous nous proposons d'introduire un schéma de codage de canal dans la chaîne de transmission afin d'améliorer les performances du système. La définition des principaux paramètres d'un code correcteur d'erreurs (rendement, longueur de trame) et l'évaluation des performances obtenues doit pouvoir être faite en référence à un standard prédéfini ou en cours d'élaboration. Malheureusement, il n'existe pas à ce jour de norme pour les réseaux locaux d'accès sans fil large bande en onde millimétriques. Nous ferons donc référence par la suite aux deux standards les plus proches : IEEE 802.11a et ETSI/BRAN HIPERLAN/2.

Le choix du meilleur schéma de codage de canal pour notre étude doit prendre en compte la capacité du réseau à offrir une qualité de service propre aux applications multimédia, telles que la transmission vidéo. Or, le standard MPEG-2 nécessitant un taux d'erreur binaire (TEB) inférieur à  $10^{-7}$  [10], une solution de type turbo code s'impose naturellement. De plus, les spécificités de notre système nous incitent à considérer plus particulièrement l'insertion d'un turbo code en blocs (TCB)<sup>3</sup> : très haut débit, rendement de codage élevé, trames de faible taille (e.g. 54 octets pour un couplage au réseau ATM). A cet égard, la sélection d'un turbo code en blocs, acquise pour le standard IEEE 802.16.1<sup>4</sup>, et probable pour IEEE 802.16.3<sup>5</sup>, est assez significative. Quoique différents des standards IEEE 802.11a et ETSI/BRAN HIPERLAN/2, ces réseaux d'accès sans fils présentent des similitudes en termes de débits et rendement de codage.

### Performance sur canaux 60 Ghz

Les premiers travaux nous ont permis d'évaluer les performances de l'algorithme d'Hartmann-Nazarov sur les canaux de Rayleigh. La figure 4.7 montre que cet algorithme a une vitesse de convergence deux fois plus rapide que l'algorithme de Chase Pyndiah. Malgré sa plus grande complexité sur le plan mathématique avec l'utilisation d'un certain nombre de fonction comme la tangente hyperbolique, la fonction logarithme, la fonction exponentielle et même la division, cet algorithme peut relativement facilement être conditionné dans une optique en temps réel et être implémenter sur processeur de signaux [18]. Nous avons par la suite testé notre algorithme de décodage sur les canaux 60 Ghz. Le modèle utilisé est un canal multitrajet avec trajet direct dans un système CDMA avec récepteur RAKE. Dans ce modèle, le temps de cohérence du canal est fixé par deux paramètres à savoir, d'une part le déplacement de l'utilisateur, pour lequel le changement des profils des retards est réalisé tous les 512 symboles BPSK, et d'autre part la variation du canal. Pour un emplacement donné le canal varie tous les 100 symboles. La figure 5.1 montre les premiers résultats obtenus. Le gain de codage obtenu après 4

---

<sup>3</sup>Turbo Product Code (TPC) selon la terminologie anglaise

<sup>4</sup>IEEE 802.16.1 est responsable du développement de l'interface air du standard LMDS (*Local Multipoint Distribution Service*). ETSI/BRAN HIPERACCESS est l'homologue européen.

<sup>5</sup>IEEE 802.16.3 est responsable du développement de l'interface air du standard MMDS (*Multichannel Multipoint Distribution Service*), ETSI/BRAN HIPERMAN est l'homologue européen.

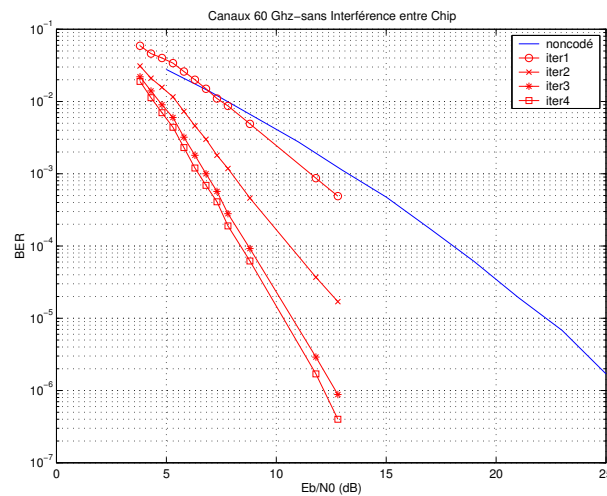


FIG. 5.1 — Taux d'erreurs Binaire sur canaux 60 Ghz, sans interférence entre chip. Nombre de finger égal à 3

itérations est de l'ordre de 12 dB pour un taux d'erreurs binaires de  $10^{-5}$ , ceci dans le cas d'absence d'interférence entre chips. Le flux de données est découpé en trames de 1024 symboles comprenant 676 symboles d'informations (rendement de codage : 0.66). Ce choix est purement arbitraire et avait pour seul but de valider le turbo décodage avec l'algorithme d'Hartmann-Rudolph-Nazarov sur les canaux 60 Ghz. L'étude ne sait pas non plus intéresser à la validation temps réel du système.

## 5.2.2 Conclusions - Perspectives

Dans ce paragraphe, nous avons montré l'importance de la mise en place d'un codage correcteur d'erreurs efficace, dans l'optique d'une exploitation de ces canaux 60 GHz dans le cadre d'une transmission CDMA. le gain de codage est relativement important (12 dB à  $10^{-5}$ ). Il est certain que le modèle devra être affiné pour prendre en compte toutes les contraintes du canal. Avant de poursuivre les travaux sur ces canaux et envisager des configurations réelles, il est important d'apporter des précisions sur les différentes tailles de blocs devant être traitées, les contraintes éventuellement temps réel à prendre en compte etc.. Différentes possibilités existent pour adapter les turbo codes aux tailles exigées pour les différentes applications (codes BCH différents, utilisation de codes étendus ou expurgés, recours aux techniques de raccourcissement et de poinçonnage...).

## 5.3 Turbo Codes pour BLUETOOTH

### 5.3.1 Introduction

BLUETOOTH [39] est une liaison radio courte distance destinée à remplacer les câbles de connexion réseaux pour les appareils électroniques portables ou fixes. Les caractéristiques principales sont la robustesse, la faible complexité, la faible consommation

et le bas prix. Bluetooth utilise la bande ISM (Industrial, Scientific & Medical) à 2.4 GHz. Un étalement de spectre par saut de fréquence est utilisé pour combattre à la fois les interférences et les évanouissements, le débit symbole est de 1 Ms/s. Sur le canal, l'information est échangée par paquets, chaque paquet étant transmis sur un saut de fréquence différent, chacun est constitué de trois entités :

- le code d'accès de taille fixe 72 bits,
- l'entête, également de taille fixe 54 bits,
- la charge utile, de taille variant de 0 à 2745 bits.

Dans son environnement, Bluetooth est perturbé par du bruit d'origine électromagnétique. Pour améliorer la qualité de la communication, Bluetooth possède 3 schémas de correction d'erreurs :

- un correcteur de type FEC (Forward Error Correction) de rendement 1/3
- un correcteur de type FEC de rendement 2/3 une possibilité de répétition ARQ (Automatique Repeat Request).

### 5.3.2 Turbo Codes Programmables

L'objectif de notre intervention dans le projet était de proposer des solutions à base de Turbo-codes pouvant être intégrées dans la norme Bluetooth. Tous les paquets ne sont pas protégés et nous avons retenu 3 configurations d'intégrations possibles, pour les charges utiles de 15, 1500 et 2746 bits. Notre choix s'est porté sur des Codes et Turbo Codes en blocs utilisant des codes élémentaires BCH. Le décodeur retenu utilise l'algorithme de Chase-Pyndiah. La taille de ces Codes et Turbo Codes n'étant pas quelconque, nous avons eu recours aux techniques de bourrage et de raccourcissement pour adapter le système de correction d'erreurs aux contraintes de l'application. Les solutions retenues utilisent un code élémentaire  $BCH(16, 11)$  pour la charge utile de 15 bits, des Turbo Codes pour les charges utiles de 1500 et 2746 bits. Les codes utilisés pour ces 2 charges sont des codes  $BCH(64, 57)$ ,  $BCH(64, 51)$  et  $BCH(32, 26)$ . Les configurations ont été validées à la fois sur les canaux de Gauss et de Rayleigh. Les gains de codage obtenus varient de 7 à 8 dB sur canaux de Gauss à 27 dB sur canaux de Rayleigh.

#### Raccourcissement et poinçonnage

Les codes produits sont, ainsi que nous l'avons vu, constitués à partir des codes élémentaires  $BCH(n, k, d)$  avec la contrainte suivante pour la création du code :

- Longueur des mots de code  $N = 2^m - 1$
- Nombre de symboles de redondance  $N - K = mt$ , où  $t$  est le pouvoir de correction
- Distance minimale  $d_{min} = 2t + 1$ .

Ce qui veut dire que les tailles des blocs pouvant être traitées dépendent des codes choisis. Dans la pratique bien évidemment, la taille des blocs est fixée par l'application, on a donc recours aux techniques de raccourcissement et de poinçonnage pour adapter le turbo code aux tailles exigées par l'application.

- On utilise le raccourcissement pour réduire le nombre de bits d'information
- On utilise le poinçonnage des bits de redondance, après l'opération de raccourcissement pour obtenir le nombre de bits à émettre.

Le problème à résoudre, quand on utilise de telles techniques, est de deux sortes :

- Quelle est la distribution optimale des bits raccourcis et poinçonnés ?
  - Comment traiter les données correspondant aux bits raccourcis et poinçonnés ?
- Les positions des bits raccourcis et poinçonnés sont connues du décodeur. Celui-ci donne une forte fiabilité aux bits raccourcis et une fiabilité faible aux bits poinçonnés. Pour préciser les idées prenons l'exemple suivant :
- - objectif :  $N = 380, K = 190$
  - - code produit  $(32, 26, 4)^2, n^2 = 1024, k^2 = 676$
  - - nombre de bits raccourcis  $X = (k^2 - K) = (676 - 190) = 486$
  - - nombre de bits poinçonnés  $Y = (n^2 - k^2) - (N - K) = 158$ .

Une propriété intéressante des codes permet de réduire de façon importante le nombre de bits à poinçonnés. Au vecteur d'information, constitué de symboles nuls, correspond des symboles de redondance également nuls. Pour obtenir  $X = 486$ , il suffit de mettre 12 lignes et colonnes à zéro soit 480 bits à zéro et d'en rajouter 6 sur la diagonale. Compte tenu du fait qu'un grand nombre de bits à poinçonnés se retrouve à zéro avec une grande fiabilité, sur les 158 précédents il n'en reste plus que 14 qui recevront une fiabilité faible.

### Configuration Des Turbo-Codes pour BLUETOOTH

Pour répondre aux différents services proposés par Bluetooth, la charge utile des paquets prend des valeurs différentes, avec des types et des rendements de codage différents.

- 1. DM1 : Nombre total de bits par paquets : 240, rendement de codage 2/3, soit donc 160 bits d'information .
- 2. DM3 : Nombre total de bits par paquets : 1500, rendement de codage 2/3, soit donc 1000 bits d'information Le turbo code produit utilisé est le suivant  $BCH(64, 57, 4) * BCH(32, 26, 4)$  avec 2 colonnes et 7 lignes à zéro ainsi que 45 bits sur diagonale. La configuration finale est obtenue avec 5 bits poinçonnés. Les deux codes élémentaires ont un pouvoir de correction  $t = 1$ .
- 3. DM5 : Nombre total de bits par paquets : 2736, rendement de codage 2/3, soit donc 1824 bits d'information Le turbo code produit utilisé est le suivant  $BCH(64, 57, 4) * BCH(64, 51, 6)$  avec 11 colonnes et 11 lignes à zéro ainsi que 16 bits sur diagonale. La configuration finale est obtenue avec 57 bits poinçonnés. Le premier code élémentaire a un pouvoir de correction  $t = 1$ , le second  $t = 2$ .

### 5.3.3 Performances obtenues

#### Canal de GAUSS

La figure 5.2 présente les performances d'un simple code BCH(16, 11, 4). Le gain de codage obtenu est de 4 dB pour un taux d'erreur de  $10^{-4}$ . Le décodeur utilise un décodage pondéré avec 8 séquences tests. pour une transmission par modulation MDP-4 sur canal de Gauss. Les figures 5.3 et 5.4 présentent les résultats obtenus avec utilisation d'un turbo décodage avec l'algorithme de Chase-Pyndiah. Le nombre de séquences test est fixé à 8. pour une transmission par modulation MDP-4 sur canal de Gauss. Le turbo décodeur des trames de charge utile 1500 bits utilise deux codes élémentaires avec des pouvoirs



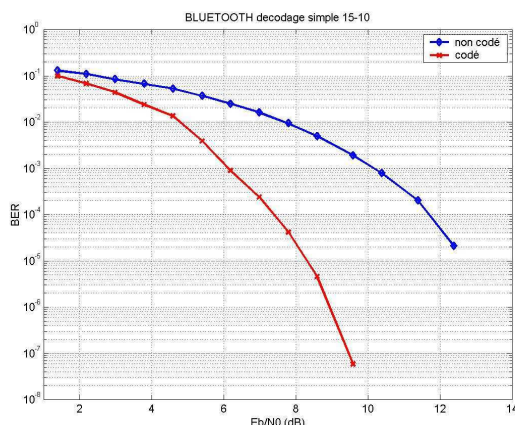


FIG. 5.2 — Décodage pondéré code BCH(16, 11) raccourci à BCH(15,10)

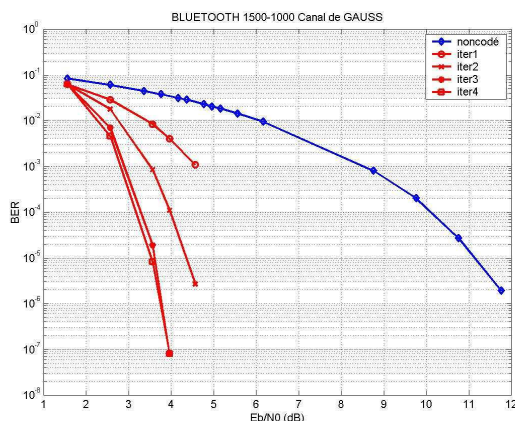


FIG. 5.3 — Turbo décodage des paquets de charge utile 1500 bits Avec un rendement de codage de 2/3 sur canal de gauss.

de correction de  $t = 1$  alors que le turbo décodeur des trames de charge utile 2736 bits utilisent deux codes élémentaires de pouvoir de correction  $t = 1$  pour le premier et  $t = 2$  pour le second. La figure 5.5 ; les gains de codage obtenus pour un taux d'erreurs binaires de  $10^{-5}$  sont respectivement de  $8.5dB$  pour la configuration 1500 – 1000 et de  $8dB$  pour la configuration 2736 – 1824 après 4 itérations

### Canal de RAYLEIGH

Les différentes configurations des turbo codes ont également été validées sur le canal théorique de RAYLEIGH. Pour la configuration 1500 – 1000 (Fig. 5.5), le gain de codage atteint  $27dB$  pour un taux d'erreur de  $10^{-4}$  après 3 itérations. Pour la configuration 2736 – 1824 (Fig. 5.6) celui-ci est du même ordre de grandeur.

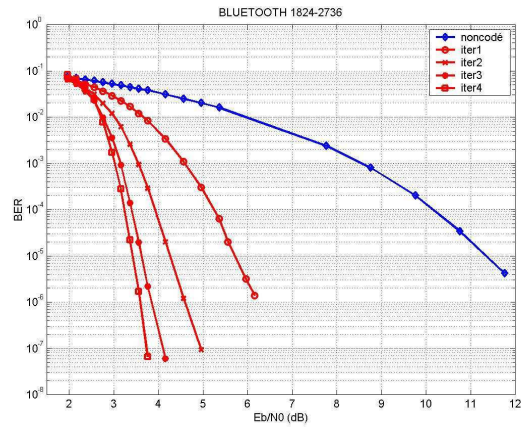


FIG. 5.4 — Turbo décodage des paquets de charge utile 2736 bits Avec un rendement de codage de 2/3 sur canal de gauss

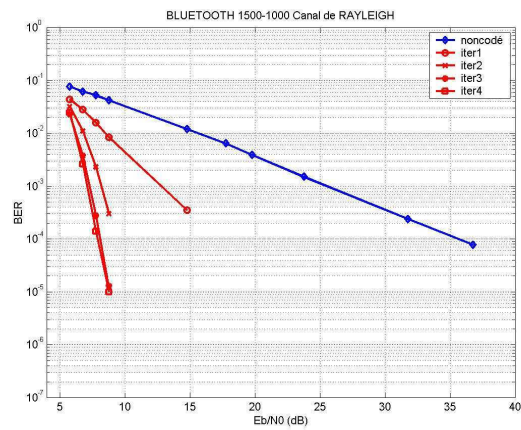


FIG. 5.5 — : Turbo décodage des paquets de charge utile 1500 bits Avec un rendement de codage de 2/3 sur canal de Rayleigh

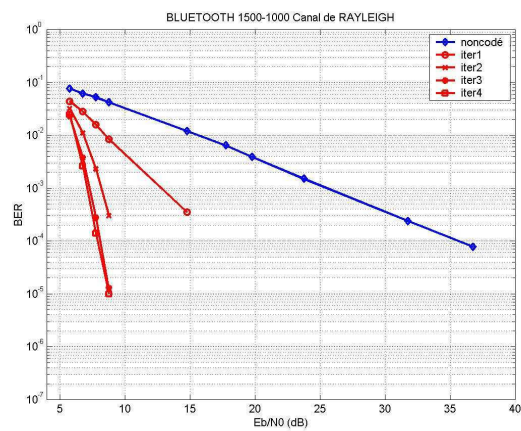


FIG. 5.6 — : Turbo décodage des paquets de charge utile 2736 bits Avec un rendement de codage de 2/3 sur canal de Rayleigh

### 5.3.4 Conclusion

Au terme de cet pré étude sur la possibilité d'introduction de codes ou turbo codes dans la norme Bluetooth un certain nombre de remarques peuvent être faites. Il est clair qu'au niveau des performances exigées, l'apport d'un système turbo ne peut être que bénéfique. L'adaptation des turbo codes en bloc à la taille des paquets de Bluetooth se fait sans problème majeur et se traduit par une faible dégradation des propres performances des turbo codes de base (non raccourcis et poinçonnés). Deux avantages intéressants peuvent découler de l'intégration d'un turbo code dans la norme. Le premier concerne la puissance d'émission, la capacité de correction d'erreurs étant accrue de façon importante, celle ci peut être réduite évitant ainsi d'éventuels effets secondaires sur les personnes soumises à de tels rayonnements, le deuxième avantage étant l'élargissement de la zone couverte à puissance constante. Malgré la plus grande complexité des systèmes turbo les débits utilisés par Bluetooth ne posent pas de problème, les contraintes temps réel, notamment pour les transmissions vocales, sont facilement vérifiées dans les implémentations Circuit et même DSP. Les configurations des codes et turbo codes n'ont pu être validées que sur les canaux de Gauss et de Rayleigh théoriques. Il est clair que préalablement à tous travaux visant à intégrer ces systèmes dans la norme Bluetooth, ceux-ci devront être validés sur le canal radio à 2.4 Ghz.. Le choix pourra être affiné concernant le type de turbo codes et éventuellement un entrelaceur pourrait intégrer en cas d'important problèmes d'évanouissement.

## 5.4 Étude des Nouvelles Modulations VLF/LF Codage de Canal

### 5.4.1 Introduction

Dans le spectre radioélectrique, l'exploitation des gammes de fréquences très basses (VLF) est très coûteuse. Les propriétés physiques des ondes électromagnétiques à basses fréquences et notamment leur profondeur de pénétration dans l'eau de mer ainsi que le faible débit de ces liaisons en ont restreint l'usage aux communications militaires, essentiellement de la terre vers les sous-marins. L'accroissement de la quantité et des formes d'informations à transmettre conduit à rechercher des solutions pour optimiser l'utilisation du canal car les gains en bande passante restent limités par les contraintes physique. Techniquement plusieurs degrés de liberté existent :

- Compression et codage de l'information à la source : transformation de l'information (message, fichier,...) en train binaire.
- Codage de canal (informations de synchronisation pour la démodulation, de recherche et de correction d'erreurs, d'entrelacement pour l'adaptation au canal de transmission,...) ·
- Modulation.

Des traitements conjoints sont possibles, dans ce cas le codage canal et la modulation sont intimement liés. L'objectif de l'étude est d'optimiser la chaîne de transmission pour offrir le maximum de souplesse à l'utilisateur du point de vue type d'information transmise et

résistance au brouillage, tout en gardant un débit utile au moins égal aux débits offerts par les systèmes actuels (typiquement 200 bits/s).

Le choix du codage canal repose sur un certain nombre de critères concernant l'application :

- Contraintes temps réel plus ou moins sévères
- Débit binaire devant être atteint
- Compromis complexité/performance
- Puissance énergétique disponible.
- Critères économiques pouvant orienter le choix
- Etc.. .

Dans le cadre de cette application, les contraintes temps réel et les débits binaires à atteindre ne constituent pas des arguments de choix. D'autre part de très bonnes performances étant attendues, le débit binaire (300 ou 1500 bits/s) autorise l'utilisation d'algorithmes relativement complexes. Compte tenu de ces arguments et en prenant en compte l'aspect économique non négligeable quand il s'agit de faire un choix, nous avons opté pour un turbo code en bloc. En ce qui concerne le décodeur, nous avons opté pour un algorithme symbole par symbole proposé par Hartmann-Rudolph-Nazarov (HRN). Malgré sa complexité il affiche de très bonnes performances. Par la suite nous proposons quelques méthodes pour, le cas échéant, optimiser les performances en terme de débit binaire, et nous présentons les performances obtenues à la fois sur le canal de Gauss et de Rayleigh. Pour cette application, l'ensemble de l'application a été implémenté sur un processeur à virgule flottante<sup>6</sup>. C'est également la version flottante de l'algorithme de Hartmann-Rudolph-Nazarov (voir Chapitre 4) qui a été implémenté.

### 5.4.2 Effet d'un Turbo Code

On examine l'effet de l'ajout d'un turbo-code BCH(31+1,26,4) en mode VLF, à une profondeur d'immersion de 6 mètres (atténuation de 26.7 dB), en bruit gaussien. La puissance de bruit atmosphérique est de  $42dB\mu V/m$  dans 200 Hz. Le bruit équivalent du système de réception est de  $-33dB\mu V/m/Hz$  (soit  $+16.7dB\mu V/m$  au dessus de la mer dans 200 Hz, négligeable devant le bruit atmosphérique). Le bruit total équivalent est de  $42dB\mu V/m$  dans 200 Hz. Il n'y a pas de vagues simulées. Le débit brut est de  $200bits/s$ .

Le débit est réduit d'un tiers (132 bits/s), mais le taux d'erreur chute brutalement. Le taux d'erreur bit est de  $10^{-3}$  pour un niveau de signal d'environ  $45.5dB\mu V/m$ , soit un rapport  $S/B$  de  $3.5dB$ . Il devient virtuellement nul à partir d'un rapport signal à bruit de  $5dB$ . Compte tenu de la perte de débit ( $2/3 = -1.8dB$ ), ce seuil se ramène à  $6.8dB$  pour comparaison avec le MSK non codé, qui possède alors un taux d'erreur bit supérieur à  $10^{-3}$ .

### 5.4.3 Conclusions

Malgré sa complexité, cet algorithme affiche des résultats intéressants même dans le cadre d'implémentation temps réel qui répétons-le n'était pas un argument de choix. Un

---

<sup>6</sup>Texas Instruments DSP TMS320C6701

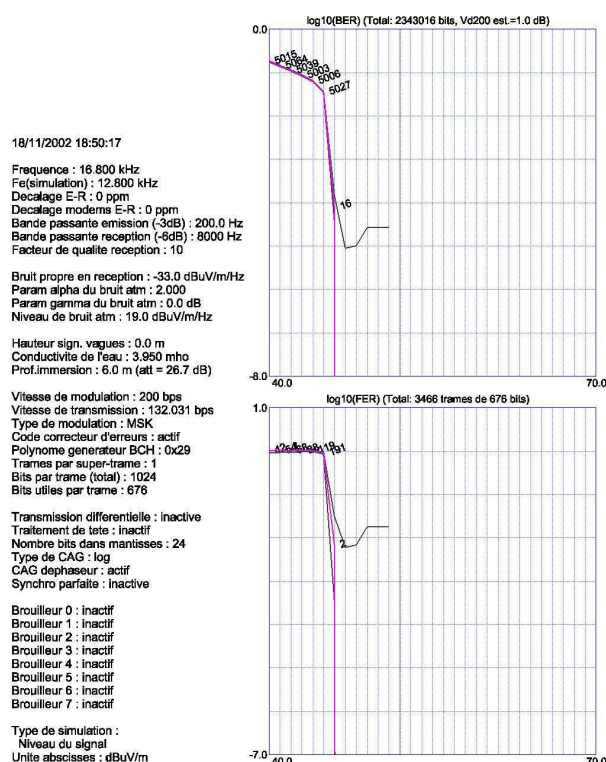


FIG. 5.7 — Effet de l'ajout d'un turbo-code BCH(31+1,26,4) en mode VLF, à une profondeur d'immersion de 6 mètres (atténuation de 26.7 dB), en bruit gaussien

point également intéressant pour cet algorithme de décodage itératif optimal, entièrement basé sur la propagation des probabilités des symboles, est l'absence de paramètres empiriques utilisés par d'autres algorithmes sous-optimaux [20]. Cet algorithme converge également deux fois plus rapidement.

Le codage canal entraîne une réduction du débit utile de 30%, mais apporte une amélioration de la fiabilité de la liaison. Pour un rapport signal à bruit générant un taux d'erreur bit de  $10^{-3}$  sur la liaison non codée, le turbo-code offre une liaison virtuellement sans erreurs. Compte tenu de la réduction de débit utile, le gain de codage, en bruit blanc gaussien, est de l'ordre de 3 dB pour un taux d'erreur binaire de  $10^{-3}$ . Il est de l'ordre de 4 dB pour un taux d'erreur binaire de  $10^{-4}$ .

## **6.1 Introduction**

C'est une activité très valorisante car elle conduit, soit à une plate forme de démonstration temps réel, soit à un produit pré-industriel. Bien souvent elle n'est pas prise à sa juste valeur dans la mesure où elle est très exigeante en terme de temps de travail. Dans la plupart des applications, les contraintes sont telles qu'il est indispensable de faire du développement en langage d'assemblage. Malgré tout ce que l'on peut dire, c'est qu'elle est un passage obligé pour valider tout algorithme. Suivant les applications cette validation peut être réalisée soit sur circuit (débit élevé exigé) soit sur DSP (fonctionnalité et évolutivité exigée). Deux activités sont présentées dans ce chapitre. Il s'agit tout d'abord de la téléphonie acoustique sous marine puis de la présentation d'un banc d'essai de turbo codes en blocs.

## **6.2 Téléphonie acoustique sous-marine**

### **6.2.1 Introduction**

Les besoins actuels en communications acoustique sous-marine, à la fois pour des applications civiles et militaires, sont en croissance continue. Durant ces dernières années le GESMA (Groupe d'Etudes Sous-Marines de l'Atlantique) et l'ENST Bretagne ont développé un système appelé TRIDENT (TRansmission de Données EN Temps réel). C'est le premier lien acoustique capable de transmettre des images, du texte et des données, utilisant une égalisation aveugle multicanaux. En 1994, un codeur de parole à bas débit (5.45 kbit/s) était mis au point et validé dans le contexte réel de la rade de Brest [8] pour transmettre de la parole par onde acoustique. Plus récemment, en juillet 2003, une transmission de parole a également été réalisée avec succès en rade de Brest en utilisant le lien acoustique GESMA-ENST-Bretagne. Comparé à un système analogique testé en parallèle, le passage au numérique améliore la qualité de la parole restituée, en offrant les outils du traitement de signal. Le cœur du lien acoustique est constitué d'un égaliseur autodidacte multicapteurs [25]. Les premières transmissions d'images, de données et de texte datent de 2002 et 2003. À la suite de ces essais Le GESMA a décidé d'étendre les possibilités de la plate forme à la transmission de parole à bas débit (2400 Bits/s). Dans la suite nous présentons les principales caractéristiques du codeur/décodeur de parole dans

le contexte temps réel avant de présenter les principaux résultats.

### 6.2.2 Codeur/décodeur CELP Ternaire (TCELP codeur/décodeur)

le signal de parole à l'entrée est segmenté en trame de 20 ms, la contrainte temps réel est donc de 20 ms. Le prototype du téléphone est présenté à la figure (Fig. 6.1) :

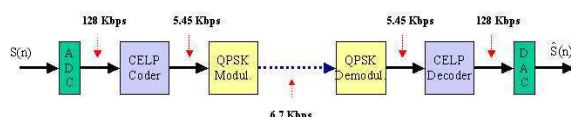


FIG. 6.1 — Téléphone acoustique numérique sous marin

le fonctionnement temps réel inclue les trois tâches suivantes :

- tâche d'acquisition : échantillonnage du signal de parole à 8000 Hz et quantification sur 16 bits sur un Convertisseur Analogique Digital (CAD) soit 128 kbits/s.
- tâche de fond : le flux binaire d'entrée (160 échantillons, quantifiés sur 16 bits soit 2560 bits) est compressé en 23 paramètres de 109 bits. Les 23 paramètres permettent de reconstruire les trois étages du codeur/décodeur CELP modélisant le système phonatoire humain, des poumons à la bouche, il comprend :
  1. le système d'excitation : il consiste en un ensemble de 1024 formes d'onde stockées dans un dictionnaire.
  2. les cordes vocales : l'effet des cordes vocales est modélisé par une fonction de transfert à un seul coefficient.
  3. le canal vocal : la fonction de transfert est modélisée à l'ordre 10 par les paires de raies spectrales
- tâche de transmission : une modulation de phase à 4 états est utilisée pour transmettre le flux de données binaires sur le canal de transmission. Une synchronisation trame de 11 bits est placée en tête de trame, donnant un débit de 6 kbits/s.

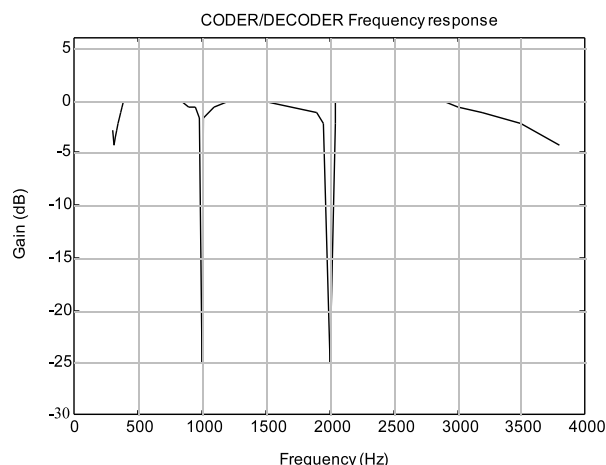


FIG. 6.2 — Réponse en fréquence

À l'issu des premiers travaux, notre première plate-forme <sup>1</sup> avait permis de tracer la courbe de réponse en fréquence du système complet (ensemble codeur/décodeur)(Fig. 6.2), le délai de traitement a également été mesuré (70 ms). Compte tenu des caractéristiques du dictionnaire des trous de fréquence apparaissent à 1000, 2000 Hz (Fig. 6.3) Les me-

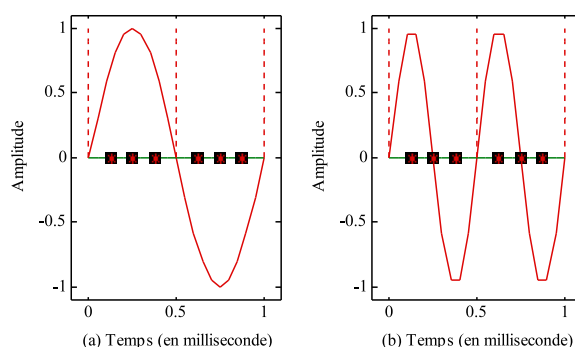


FIG. 6.3 — Trous de fréquence

sures au banc d'essai ont permis également de contrôler que notre codeur satisfait bien la contrainte<sup>2</sup> temps réel (Fig. 6.1). En juillet 2003, nous avons procédé a des essais de trans-

Analyse	Durée(ms)	Pourcentage
LPC	2.125	10.6%
LTP	2.75	13.75%
Excitation	13.5	67.5%
Total Codeur	18	90%

TAB. 6.1 — Temps de traitement codeur

mission de parole sur site dans la rade de Brest. le dispositif mis en œuvre est présenté à la figure (Fig. 6.4). les essais se sont déroulés en eau peu profonde et ne concerne que le canal horizontal. Après compression le débit du codeur de parole est de 5.45 kbits/s. Avant transmission ce débit est porté à 6.7 kbits/s (mot de synchronisation et bourrage), la modulation utilisée est une modulation de phase à 4 états et la porteuse de 20 khz. la distance entre l'émetteur et le récepteur est de l'ordre de 2000 mètres. Les essais consistent en une émission successive de signaux de parole (durée 5 mn). L'émission se fait à partir d'un fichier de référence, préalablement conditionné et stocké dans la base d'émission. Ceci donne une idée de l'effet du canal acoustique sous-marin sur la parole synthétisée. En réception l'espace entre 2 hydrophones voisins sur les 4 est approximativement égal à 25 cm

Les performances du récepteur sont évalués par l'évaluation de l'erreur quadratique moyenne (Mean Square Error MSE), à la sortie de l'égaliseur. Les constellations d'entrée

<sup>1</sup>Cette première version de l'ensemble CodeurDécodeur a été implémenté sur des processeurs Motorola DSP 56xxx (fréquence d'horloge 27 Mhz, 13.5 MIPS)

<sup>2</sup>le temps correspond au temps maximum mesuré sur la base de donnée



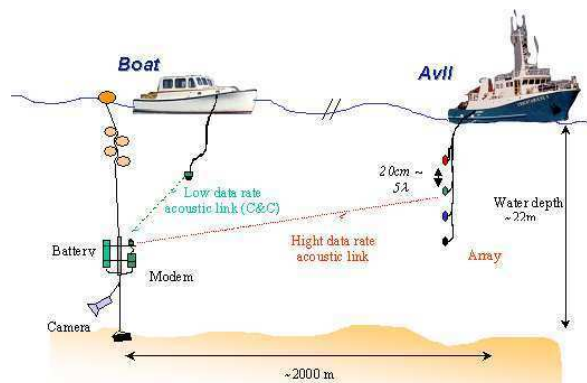


FIG. 6.4 — Essai en mer

et de sortie de l'égaliseur donne une idée du bon fonctionnement du récepteur. L'égaliseur est constitué de filtres transverses à 21 coefficients et d'un filtre récursif à 35 coefficients commun à chaque dans le mode de convergence et unique en mode de poursuite, permettant de prendre en compte la durée de la réponse impulsionnelle du canal. La figure (Fig. 6.5) montre l'évolution du MSE. Il faut noter le bon comportement de l'égaliseur sur la durée de la séquence mais du fait des 2 perturbations, le signal de parole n'est pas en permanence correctement synthétisé. En cas de perturbation, l'égaliseur passe de la

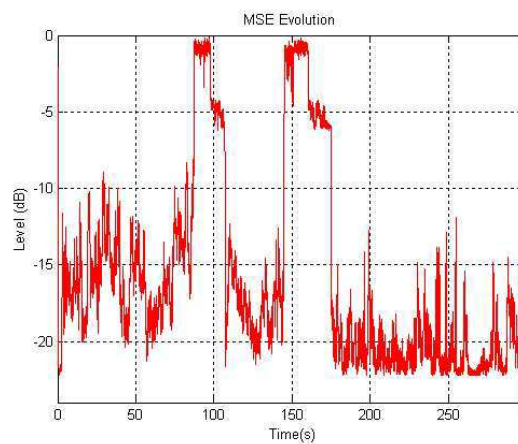


FIG. 6.5 — Erreur quadratique moyenne

phase de poursuite à la phase de convergence pour y revenir dès que les décisions sont suffisamment fiables. Cette possibilité de passage d'un mode à l'autre prouve la robustesse de l'égaliseur face aux sévères perturbations du canal. Dans la figure (Fig. 6.6, les constellations d'entrée (capteur 1) et de sortie sont représentées pour une séquence de 20 secondes (entre la 12<sup>ème</sup> et la 32<sup>ème</sup> secondes du fichier d'entrée). La constellation de sortie prouve à la fois l'efficacité et la justesse de l'égaliseur. La constellation d'entrée prouve la présence de très fortes interférences intersymboles dues aux chemins multiples.

la figure (Fig. 6.7) montre la parole synthétisée après transmission acoustique sous-marine, l'effet de la perturbation apparaît très clairement. Pour l'instant notre système ne dispose pas de codage canal.

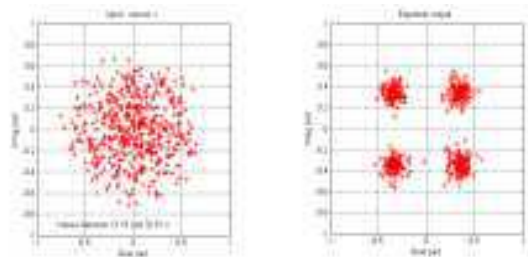


FIG. 6.6 — Constellations d'entrée et sortie de l'égaliseur entre la 12<sup>eme</sup> et 32<sup>eme</sup> s

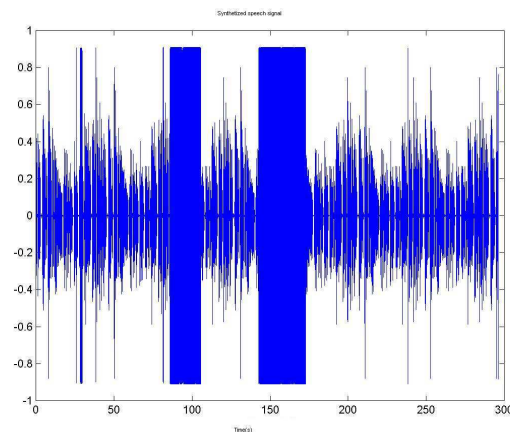


FIG. 6.7 — parole synthétisée après transmission acoustique

### 6.2.3 Conclusions-Perspectives

A la suite de ces essais qui ont concerné les transmissions d'images, de données, de textes et de signaux de parole, on a pu noter la robustesse de l'égaliseur grâce à ses 2 modes de fonctionnement et la possibilité de commuter d'un mode à l'autre de façon automatique. Des communications d'images, de textes, de données et aussi de paroles peuvent être réalisés sur des canaux très sévères comme le canal acoustique sous-marin. La prochaine introduction du codage canal devrait permettre d'apporter encore une plus grande fiabilité aux communications acoustiques sous-marines. De plus pour augmenter la portée opérationnelle du système, la plate forme va être équipée d'un nouveau codeur de parole à très bas débit (MELP 2400 Bits/s), permettant l'utilisation d'une porteuse à une fréquence plus basse.

## 6.3 Banc d'essai de Turbo Codes

### 6.3.1 Introduction

Aujourd'hui les chercheurs disposent d'outils très performants, qui leurs permettent de présenter et de valider leurs travaux dans un contexte temps réel. C'est dans un tel souci que j'ai lancé l'idée de la réalisation d'une plate forme d'essai de turbo codes en temps réel. Les premiers résultats sont présentés. Ces travaux se sont déroulés et se

déroulent dans le cadre d'un programme "ELITE EUROPEAN PROGRAM"<sup>3</sup>. A la suite des nombreux travaux temps réels réalisés, l'ENST Bretagne a été intégrée à ce programme. L'objectif est de développer une plate forme (Fig. 6.8) conviviale et interactive de démonstration temps réel pour présenter et analyser les performances des Turbo Codes en blocs (le système utilise le TCB BCH(32,26,4)\*BCH(32,26,4), voir Chapitre 4. La plate forme inclue tous les outils matériels (PC, DSP et la liaison RTDX<sup>4</sup>, elle comprend 2 parties :

- la première partie est installée sur l'ordinateur hôte, elle exécute les tâches suivantes :
  1. génération de la matrice de bits d'information.
  2. codage BCH des lignes et des colonnes.
  3. après bruitage, émission des échantillons bruités vers le décodeur.
  4. après réception de la matrice corrigée, calcul du taux d'erreur binaire et tracé de courbes en temps réel.
- la seconde partie est implémentée sur le DSP, elle assure le décodage de la matrice transmise par le codeur. Après décodage, la matrice est retransmise à l'ordinateur hôte pour lui permettre de réaliser la tâche 4 précédente.

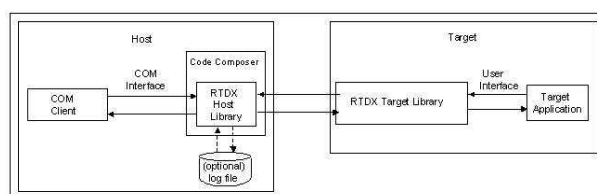


FIG. 6.8 — Plate forme Temps réel

### 6.3.2 Gestion de l'interface graphique

Un certain nombre de fonctionnalités ont été mises en place sur cette plate forme :

1. Taux d'Erreur Binaire (BER Bit Error Rate) en fonction du rapport Signal à Bruit (SNR Signal to Noise Ratio). La figure (Fig. 6.9) présente la courbe de taux d'erreur en fonction du rapport Signal à Bruit (SNR Signal to Noise ratio). À chaque fois qu'un nouveau point est disponible, il est immédiatement tracé sur le graphique. Il est également possible de lancer le calcul pour un point isolé.
2. Suivi graphique du décodage. Les figures (Fig. 6.10), (Fig. 6.11) et (Fig. 6.12) présentent l'état de la matrice après chaque itération suivant le SNR.
3. Turbo Décodage d'images bruitées, la figure (Fig. 6.13) est un exemple de débruitage d'une image.

<sup>3</sup>Dans ce cadre Texas Instruments met à la disposition d'un certain nombre d'Université Européenne, des moyens techniques pour développer des projets innovants

<sup>4</sup>Real Time Data eXchange, communication temps réel entre l'ordinateur hôte et la cible DSP

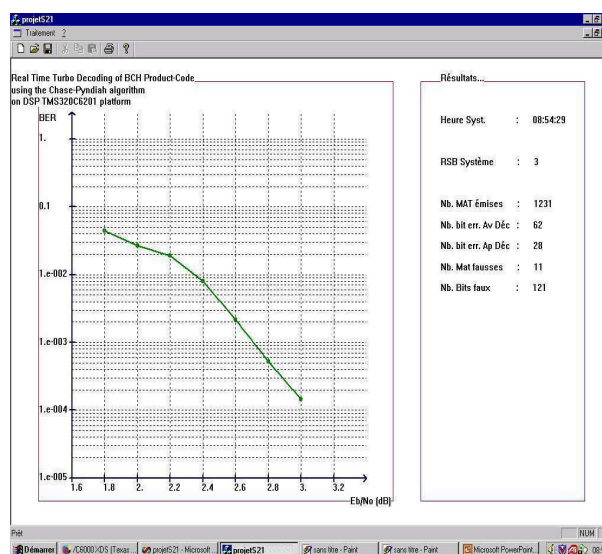


FIG. 6.9 — BER vs SNR

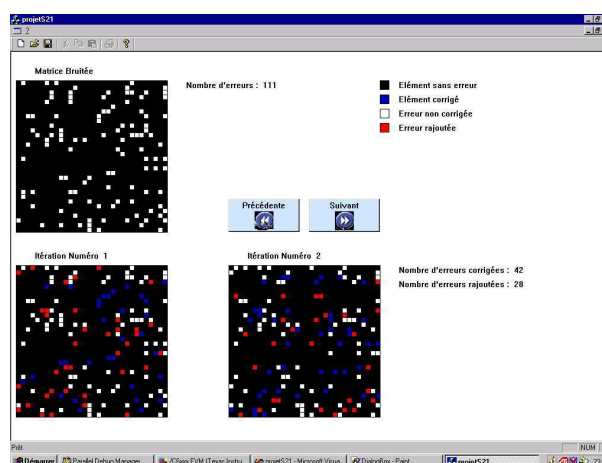


FIG. 6.10 — Décodage d'une matrice

## 6.4 Conclusions-Perspectives

Dans un premier temps, l'ambition de cette plate-forme était de réaliser un outil essentiellement pédagogique. Elle permet de valider le fonctionnement temps réel de l'ensemble codeur-transmission-décodeur. Concernant la vitesse d'exécution, un certain nombre de points peuvent être signalés :

- la liaison RTDX (Real Time Data Exchange) a un débit de transmission limité, car elle était prévu pour des simples échanges d'information entre l'ordinateur hôte et la cible sur DSP.
- le traitement au niveau du codeur et du décodeur reste séquentiel et ne bénéficie donc des nouveaux outils disponibles sur les plates formes à savoir le BIOS<sup>5</sup>, moniteur multi tâches temps réels désormais disponible dans l'environnement DSP

<sup>5</sup>Built In Operating System

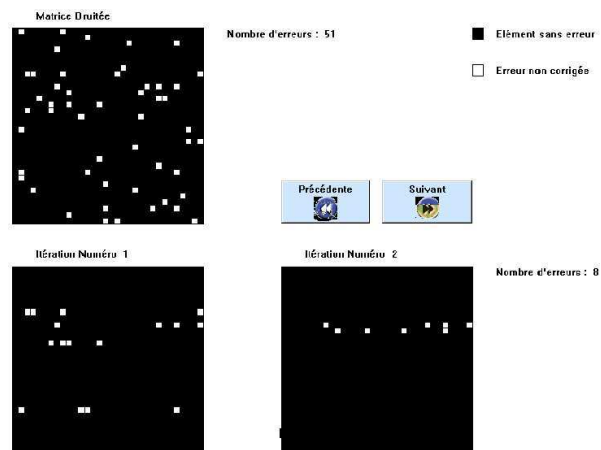


FIG. 6.11 — Plate forme de demonstration

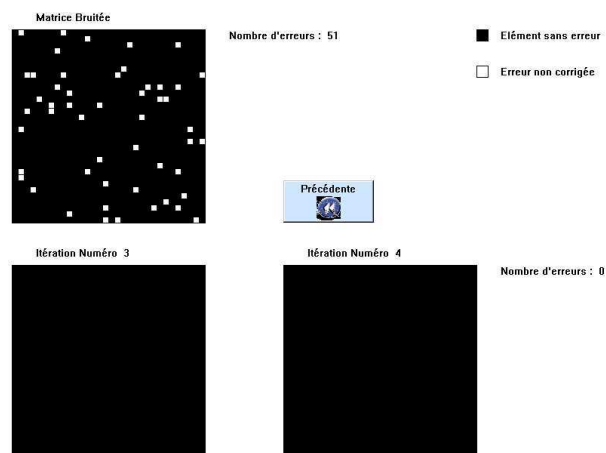


FIG. 6.12 — Plate forme de demonstration



**FIG. 6.13** — Décodage d'image : en haut image originale, en bas image bruitée et au milieu image après débruitage



# CHAPITRE 7 Conclusion

---

Après quatorze années passées au sein des départements Mathématiques et Communications puis Signal et Communications de L'ENST-Bretagne, le moment est venu, après ma thèse soutenue en 1994, de présenter mes travaux pour être Habilité à Diriger des Recherches (HDR). C'est dans le cadre des télécommunications que je suis aujourd'hui amené à présenter cette habilitation. Ingénieur physicien de formation, c'est dans ce domaine que j'ai le sentiment de pouvoir apporter le plus à cette économie bretonne, mise récemment en difficulté, avec la crise mondiale des télécommunications. Mes expériences professionnelles dans des unités délocalisées <sup>1</sup> de l'Université de Bretagne Occidentale (UBO), ne m'avaient objectivement pas permis d'entreprendre des travaux de recherche. Trois axes importants marquent ces quinze dernières années dans les télécommunications :

- Le codage source avec la compression de la parole, principal objet de mes travaux de thèse. Une première maquette de démonstration temps réel a été réalisée au laboratoire. La version n'incluant que le décodeur a été validé sur canal sous-marin en rade de Brest en 1993, cette version n'incluait pas le codage canal destiné à corriger les éventuelles erreurs de transmission. La version met en œuvre également les travaux réalisés au département SC sur l'égalisation. Un véritable prototype de téléphone acoustique sous-marin aurait pu être mis au point dans la région brestoise au milieu des années 90 en avance sur d'autres centres de recherches, les circonstances ne l'ont pas permis.
- Le codage canal avec les turbo codes, spécialité mondiale reconnue de l'ENST-Bretagne. C'est tout naturellement que des nouveaux travaux ont été développés au département SC pour proposer d'intégrer les TCB dans différentes normes et dans différentes applications (Hiperlan/2, Canaux LF/VLF, Canaux 60 GHz, Bluetooth etc..). Une activité plus spécifique a consisté à valider en temps réel différents algorithmes de turbo-décodage de codes en blocs sur processeurs de signaux.
- Les travaux de mise au point d'algorithmes ne sont terminés que si il ont été validés sur des plates formes temps réel soit sur des circuits, soit sur des processeurs de signaux. Bien souvent les algorithmes validés en simulation doivent être reconditionnés pour faire face aux contraintes temps réel. Les travaux résultants des points précédents ne peuvent évidemment pas échapper à la règle.

---

<sup>1</sup>Génie Thermique IUT de Lorient, Département Génie thermique 1975  
Département Biologie Appliquée option Industrie Agroalimentaire IUT de Quimper



C'est sur ces mêmes thèmes que j'entends poursuivre mes activités à l'ENST-Bretagne <sup>2</sup>. Parmi les travaux d'ores et déjà programmés citons :

- Extension de la fonction parole au système de communication acoustique sous-marine TRIDENT. La Direction Générale de l'Armement avec le GESMA a récemment relancé cette activité parole en souhaitant l'adjoindre au prototype de communication acoustique déjà réalisé et permettant de transmettre des images, du texte et des données en temps réel
- Intégration d'une fonction de codage de canal aux canaux acoustiques sous-marins (système TRIDENT), complément au point précédent.
- Poursuite des travaux sur l'intégration des différentes familles de Turbo Codes (BCH, RS) aux différentes normes ou applications de télécommunications.
- Poursuite et extension des activités temps réels, projet "Elite European", réalisation de véritables plate-forme de validation temps réel.
- Acquisition des compétences dans les domaines de l'égalisation et turbo égalisations se rapprochant beaucoup de ma formation initiale.
- Intégration des fonctionnalités multi-tâches temps réel sur les applications DSP. Mise en œuvre des BIOS (Built In Operating System).
- Etc.

---

<sup>2</sup>Situation de chercheur-associé confirmée en 2002 par l'UBO.

---

# Bibliographie

## 7.1 Publications

- [1] Omar Aitsab *Turbo codes et codage conjoint source-canal : Application à la transmission d'images* Thèse de doctorat, Université de Bretagne Occidentale, Février 1998
- [2] C. Berrou, A. Glavieux, P. Thitimajshima *Near Shannon limit error-correcting coding and decoding : Turbo-codes* Proc IEEE International Conference on Communications (ICC'93), Geneva 1993, pp1064-1070
- [3] Nadine Chapalain-Le Floc'h *Application des turbo codes en blocs pour les réseaux locaux sans fil à haut débit* Thèse de doctorat, Université de Bretagne occidentale, Octobre 2002
- [4] D. Chase *A class of algorithm for decoding block codes with channel measurements information* IEEE trans.inform.Theory. Vol .IT-18,Jan.1972, pp170-182.
- [5] P.Elias *Error-free coding* IRE trans on Inf. Theory, vol. IT-4, pp. 29-37,Sept.1954
- [6] J. Hagenauer, E. Offer and L. Papke, *Iterative decoding of binary block and convolutional codes*, IEEE Trans. Inform. Theory, vol. 42, no. 2, pp. 429-445, 1996.
- [7] Carlos R.P. Hartmann and Luther D. Rudolph, *An optimum symbol-by-symbol decoding rule for linear codes*, IEEE Trans. Inform. Theory, vol. IT-22, no. 5, pp. 514-517, 1976.
- [8] Sylvie Jacq *Décodage itératif des codes produits : "turbo-codes en blocs", et évaluation de leurs performances pour des modulations MDP et MAQ sur canal de Gauss et de Rayleigh* Thèse de doctorat, Université de Limoges, Mai 1996
- [9] L.E. Nazarov and V.M. Smolyaninov, *Use of Walsh-Hadamard transformation for optimal symbol-by-symbol binary block-code decoding*, Electron. Letters, vol. IT-34, no. 3, pp. 261-262, 1998.
- [10] G. Wu and M. Inoue and H. Murakami and Y. Hase", *Broadband radio access integrated network (BRAIN) in millimeter-wave band* Proceedings of the ISSSE'01,2001, pp 126-130
- [11] A.Goalic and R. Pyndiah *Real time Turbo Decoding of Product Codes on a Digital Signal Processor* INTERNATIONAL SYMPOSIUM ON TURBO CODES AND RELATED TOPICS, pp 267-270, Brest, France, September 1997.

- [12] A.Goalic and R. Pyndiah *Real time Turbo Decoding of Product Codes on a Digital Signal Processor* GLOBECOM'97, Phoenix, USA, Vol.II, pp624-628.
- [13] A. Goalic, N. Chapalain and R. Pyndiah, *Implementation of a low complexity algorithm for BCH block turbo decoding on Digital Signal Processor*, ICSPAT'99, nov., Orlando, Florida, USA.
- [14] A.Goalic and R. Pyndiah *Implementation of a low complexity algorithm for BCH Block Turbo Decoding on a Digital Signal Processor* ICSPAT, November 1-4, 1999, Orlando, Florida, USA.
- [15] A.Goalic, N. Chapalain and R. Pyndiah *Real time Turbo Decoding of BCH product code on the DSP Texas TMS320C6201* INTERNATIONAL SYMPOSIUM ON TURBO CODES AND RELATED TOPICS Brest, France, 4-7 September 2000.
- [16] A.Goalic, N. Chapalain and R. Pyndiah *Real time Turbo Decoding of BCH product code on the DSP Texas TMS320C6201* The Third European DSP Education and research Conference, ESIEE Paris 20-22 September 2000.
- [17] A. Goalic, Karine Cavalec-Amis and Vincent Kerbaol *Real time Turbo Decoding of Block Turbo Codes using the Hartmann-Nazarov algorithm on the DSP TEXAS TMS320C6201* ICC2002, New York, USA 29 Avril - 3 Mai 2002.
- [18] A. Goalic *Implantation en temps réel d'un codeur de parole CELP pour la téléphonie acoustique sous-marine* Séminaire Compression et indexation parole et audio, 11 Juin 2003 ENSSAT Lannion
- [19] A. Goalic, W. Sawaya, R. Okouyi and L. Clavier *Performance Of Block Turbo Codes For A Multiusers - Cdma Link On The 60ghz Channel* 2nd ISIVC, ENST-B Brest France 7-9 July 2004
- [20] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq *Near optimum product codes* Globecom'94, san francisco, Nov. , Dec. 1994, pp339-943
- [21] R. Pyndiah, *Near-optimum decoding of product codes : Block Turbo Codes*, IEEE Trans. Commun., vol. 46, no. 8, pp. 1003-1010, 1998.
- [22] W. Tapsoba, A. Goalic advisor *Use of fast Hadamard transform for Block Turbo Codes Decoding and Real Time Implementation on a Texas TMS320C6201 Platform* Projet de recherche Mastère International en Télécommunications ENST-Bretagne et Texas Instruments "DSP and Analog Challenge 2001"
- [23] G Lapierre, J Labat, J Trubuil *Evaluation of high data rate acoustic link : contribution of blind spatio-temporal equalization* Proc. ECUA' 2000, Lyon, France, Vol.1, pp549-554
- [24] Gérard LAPIERRE, Nicolas BEUZEULIN, Joël LABAT, Joel TRUBUIL, Andre GOALIC, Samir SAOUDI, Gerard AYELA, Pascal COINCE, Stephane COATELAN, *1995-2005 : Ten years of active research on underwater acoustic communications in Brest* OCEANS'05 EUROPE, June 2005, Brest, Brittany.
- [25] J Trubuil, T Gall, G Lapierre, J Labat *Development of a real-time high data rate acoustic link* Proc. OCEANS 2001, Hawaii, Vol 4, pp 2159-2164.
- [26] Joël Trubuil, Gerard Lapierre, Joël Labat, Nicolas Beuzelin, André Goalic and Christophe Laot *Improved AUV autonomy provided by an underwater acoustic link* ISOPE-2006 San Francisco, California, USA, May 28-June 2

- [27] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *Performance of low complexity high code rate Reed-Solomon block turbo codes* Second International Symposium on Image/video Communications over fixed and mobile network, Brest, France, July 2004.
- [28] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *Small block Reed-Solomon btc for reliable transmission* International Conference on Information and Communications Technologies : From Theory to Applications, Damascus, Syria, April 2004.
- [29] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *Achieving channel capacity with low complexity RS-BTC using QPSK over AWGN Channel* International Conference on Communications, Fortaleza, Brazil, Aug. 2004.
- [30] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *reliable transmission with low complexity Reed-Solomon block turbo codes* First International Symposium on Wireless Communications Systems, Mauritius, Sept. 2004.
- [31] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *Potential application of low complexity non binary high code rate turbo codes* Milcom 2004, Monterrey, USA, Oct. 2004.
- [32] R. Zhou, A. Picart, R. Pyndiah and A. Goalic *Performance of low complexity RS-BTC with M-ary phase shift keying* Global Mobile Congress 2004, Shanghai, Oct. 2004.
- [33] Rong Zhou *Etude des Turbo Codes en Blocs Reed-Solomon et leurs Applications* Thèse de doctorat, Université de Rennes 1, Avril 2005

## 7.2 Rapports de Recherche

- [34] Lotfi AKRIMI, Estelle GABARRON, Guillaume GALATAUD Yiqun GE, Pascal KOENIG, Sandrine PASQUALINI, Nicolas WIMMER *Real-Time Demonstrator of Block Turbo-Codes Decoding on the Texas Instruments DSP TMS320C6201 Platform* Elite Européen program, Texas Instrument, septembre 2003
- [35] K. Amis, A. Goalic et R. Pyndiah *Recensement bibliographique des Turbo Codes* contrat de recherche ENST-Bretagne N° 02CS006, lot No 1, novembre 2002
- [36] A. Goalic et R. Pyndiah *Étude d'un Turbo Code en Bloc pour Matra Communication, amélioration des performances du GSM* contrat d'étude No 3ZC1832 entre Matra Communication et ENST Bretagne avril 1998-Mai 1999
- [37] A. Goalic et V. Kerbaol *Réseau indoor large bande, canaux 60GHz* contrat de recherche ENST-Bretagne N° 3ZRE409, Mars 2002
- [38] A. Goalic *Étude des nouvelles modulations VLF/LF, codage de canal* contrat de recherche ENST-Bretagne N° 3ZCP155 novembre 2002
- [39] A. Goalic et K. Amis *Configurations des Turbo-Codes en blocs pour le projet de radio logicielle BLUETOOTH* contrat de recherche ENST-Bretagne N° 3ZRE410, mars 2003
- [40] A. Goalic *Recensement des différentes solutions à base de turbo-codes ayant fait l'objet d'intégration dans des standards* contrat de recherche ENST-Bretagne N° 02CS006, lot No 3, Mars 2004

- [41] D. Le Roux et A. Goalic *Étude du codage de canal en acoustique sous-marine* contrat de recherche ENST-Bretagne N° 00-59-910, décembre 2003
- [42] R. Pyndiah, K. Cavalec-Amis and A. Goalic *Block Turbo Code for Voice And Data Services in UMTS* Contrat de recherché Texas Instrument France
- [43] Motorola semiconducteurs *DSP56000/56001 User's manual*

### **A.1 Real time low bit rate speech transmission through underwater acoustic channel**

Goalic, A. ; Trubuil, J. ; Lapierre, G. ; Labat, J. ; Oceans 2005 - Europe Volume 1, 20-23 June 2005 Page(s) :319 - 321 Vol. 1 Digital Object Identifier 10.1109/OCEANSE.2005.1511733

### **A.2 Potential applications of low complexity non-binary high code rate block turbo codes**

Zhou, R. ; Picart, A. ; Pyndiah, R. ; Goalic, A. ; Military Communications Conference, 2004. MILCOM 2004. IEEE Volume 3, 31 Oct.-3 Nov. 2004 Page(s) :1694 - 1699 Vol. 3 Digital Object Identifier 10.1109/MILCOM.2004.1495192

### **A.3 Real-time turbo decoding of block turbo codes using the Hartmann-Nazarov algorithm on the DSP Texas TMS320C6201**

Goalic, A. ; Cavalec-Amis, K. ; Kerbaol, V. ; Communications, 2002. ICC 2002. IEEE International Conference on Volume 3, 28 April-2 May 2002 Page(s) :1716 - 1720 vol.3 Digital Object Identifier 10.1109/ICC.2002.997142

### **A.4 Real-time turbo-decoding of product codes on a digital signal processor**

Goalic, A. ; Pyndiah, R. ; Global Telecommunications Conference, 1997. GLOBE-COM '97., IEEE Volume 2, 3-8 Nov. 1997 Page(s) :624 - 628 vol.2 Digital Object Identifier 10.1109/GLOCOM.1997.638405

## **A.5 An intrinsically reliable and fast algorithm to compute the line spectrum pairs (LSP) in low bit rate CELP coding**

Goalic, A. ; Saoudi, S. ; Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on Volume 1, 9-12 May 1995 Page(s) :728 - 731 vol.1 Digital Object Identifier 10.1109/ICASSP.1995.479797

## **A.6 Toward a digital acoustic underwater phone**

Goalic, A. ; Labat, J. ; Trubuil, J. ; Saoudi, S. ; Rioualen, D. ; OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings Volume 3, 13-16 Sept. 1994 Page(s) :III/489 - III/494 vol.3 Digital Object Identifier 10.1109/OCEANS.1994.364247

# REAL TIME LOW BIT RATE SPEECH TRANSMISSION THROUGH UNDERWATER ACOUSTIC CHANNEL

André Goalic <sup>\*</sup>, Joël Trubuil <sup>\*</sup>, Gerard Lapierre <sup>\*</sup> and Joël Labat <sup>\*</sup>

<sup>\*</sup>ENST-Bretagne, BP83818, 29238 Brest Cedex 3, France

<sup>\*</sup>GESMA. (Groupe d'Etudes Sous-Marines de l'Atlantique), BP 42, 29240 Brest Armees, France  
andre.golic@enst-bretagne.fr

**Abstract:** Current needs in underwater acoustic communication, both for civil and military applications, are continuously increasing. During these last years, GESMA (Groupe d'Etudes Sous-Marines de l'Atlantique) and ENST-Bretagne developed the TRIDENT system (TRIDENT : TRansmission de Données EN Temps reel. This is the first real time acoustic link able to transmit images, text and data based on multichannel blind equalization. In 1994, a low bit rate speech coder (5.45 kbit/s) was designed and validated for speech acoustic wave transmission in Brest bay [1]. More recently, a speech transmission was also successfully realized in July 2003 at sea, using the new GESMA-ENST-Bretagne acoustic link. This paper provides an overview of this project. The context of the study and the implementation of a very low speech acoustic link are presented. Last Brest harbor speech transmission experiments (2003) are presented. The real time horizontal acoustic link performances are evaluated. Two carriers frequencies are available (20, 35 kHz). Acoustic communications for bit rate ranging from 6 to 20 kbps and range from 500 to 4000 m have been conducted successfully over several hours in shallow water environments.

## I. INTRODUCTION

A high data rate acoustic link has been developed by the GESMA. The Main objective of this project was to create an acoustic communication link between an AUV (Autonomous Underwater Vehicle) and a surface vessel. This acoustic link must be sufficiently robust and improve underwater vehicles autonomy.

The first years of this project were devoted to upstream studies [2]. The goal was to choose the best solution able to deal with different perturbations generated by underwater acoustic channel. An equalizer developed and patented by the ENST Bretagne stood for a very good solution for this application [3] [4].

An acoustic link was then developed to validate the whole communication between source data emitter and receiver. Transmission part as well as reception stage (hydrophones, amplifier) were designed by ORCA Instrumentation (SERCEL), a french company specialized in underwater acoustics modems. The receiver platform was developed by ENST Bretagne [5]. The spatio-temporal equalizer called SOC-MI-DFE (Self Optimized Configuration Multiple Input Decision Feedback Equalizer) constitutes the receiver core.

Some experiments were carried out in 2002 and 2003 to show some images, data and text transmissions. From now, GESMA plans to extend TRIDENT possibilities to low bit rate speech digital transmission. Existing and commercial products are mostly based on analog communications. Digital transmission is expected to provide improved

reliability. During our first sea-trail, digitized speech by code excited linear predictive coding (CELP) clearly shown a better listening quality.

This paper aims to highlight some aspects of the TRIDENT system. Firstly, we introduce the receiver platform with different implemented functions. Some characteristics of different algorithms are described. Secondly, we present main characteristics of speech coder/decoder under consideration for TRIDENT extension.

The next part describes some sea trials, and finally, some results are presented.

## II. TRIDENT system

Most processing stages are realized with digital signals. There are only signal amplification, calibration and prefiltering left which remain to be performed analogically. Then, each input signal is synchronously sampled. Demodulation, timing-recovery and equalization are processed by Digital Signal Processor (DSP). Demodulation is carried out by sampling every input signal at 4 times the clock frequency  $f_0$  ( $f_0$ : carrier frequency). Symbol duration  $T$  was chosen in order to keep the product  $f_0.T$  to an integer value. Timing recovery is performed with the Gardner algorithm [6]. Then T-spaced equalization can be performed.

Both multipath propagation and noise affect acoustic communication reliability. Spatio-temporal equalizer SOC-MI-DFE uses input signals sampled on several hydrophones coming from the same emission source. This spatial diversity provides a better SNR (Signal Noise Ratio) compared to single channel case.

The SOC-MI-DFE is able to run according to two modes: a convergence or starting mode in which the equalizer uses blind adaptive criteria and a tracking mode where it is decision directed. Switching between these two modes is carried out in an automatic and reversible way by comparison of the MSE (Mean Square Error) with a threshold  $J_0$  (-6dB in our system). Differences between these two modes concern as structural aspects as adaptive criteria.

Receiver platform is based on an acquisition board plugged into a personal computer (PC). Board architecture is based on a Texas Instruments (TI) TMS320C6201 DSP. This DSP is a high performance 32 bits fixed point able to reach a maximum value of 1600 MIPS (Million



Instructions Per Second). The minimum cycle time lasts 5 ns corresponding to a 200 MHz clock frequency.

Decoded data are then decoded and displayed into a GUI (Graphical User Interface). Acquisition, demodulation, synchronization, equalization are applied to a symbol. Real time is achieved when all treatments are realized faster than a symbol duration. Most signal processing functions are written in assembly language in order to reduce the processing time. In fact, there is an important gap that can reach more than 90 % in execution time depending on whether functions are written in assembly language or in C-language. Real-time constraints require to process simultaneously the acquisition of the  $(n+1)^{th}$  symbol while processing the  $n^{th}$  symbol and making the restitution of the  $(n-1)^{th}$  one.

### III. TERNARY CODE EXCITED LINEAR PREDICTION (TCELP coder/decoder)

In order to design a digital acoustic underwater phone, the input speech signal is segmented into frames having a length of 20 ms, thus the real time constraint is equal to 20 ms. The phone prototype is presented on fig. 3.

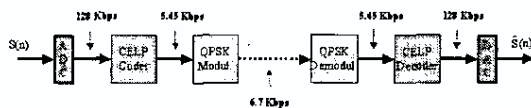


Fig. 3. Digital acoustic underwater phone prototype

Real time process includes the three following tasks:

- acquisition task : it consists in sampling and quantifies signals with sampling frequency of 8 kHz and 16 bits Analog Digital Converter (128 kbits/s).
- background task : the input stream (160 samples, coded on 16 bits - 2560 bits) is reduced to 23 parameters, corresponding to 109 bits. It includes the three stages of the phonatory system. From lungs to mouth, it is made of the excitation system, the vocal folds and the vocal tract. Excitation system only consists into ternary waveforms stored in a dictionary. Vocal folds are represented by a single coefficient transfer function. The last stage, called vocal tract, uses the line spectrum pairs to design a 10 coefficients transfer function [7].
- Transmission task: a QPSK modulation is used to transmit the bit stream over the acoustic channel, a 11 bits frame synchronization is put at the beginning of the coded frame giving a 6 kbits/s emitted rate.

### IV. EXPERIMENTAL RESULTS

Sea-trials were carried out in various areas around Brest bay in 2002 and 2003 in order to evaluate acoustic transmission reliability of the Trident system.

Sea-trials concern the real time validation of the TRIDENT system and its performance evaluation. The receiver is placed on a boat. The transmitter can be fixed either on a buoy, or for mobile tests, placed on a ROV (Remotely Operated Vehicle). Based on coding and space diversity, the system is mainly composed of two units. An autonomous underwater modem with camera dedicated to images transmission and an omni directional transducer, deal with acoustic signal emission. After transmission, a surface unit including 4 hydrophones antenna (omni directional), allows to get back acoustic signals.

Acoustic downlink is used to control the underwater unit (compression ON/OFF, compression factor, source level, bit rate, modulation...). Acoustic uplink is used to transmit information and other parameters at various programmable bit rates.

The TRIDENT system can use two carrier frequencies (20 kHz or 35 kHz) with a QPSK modulation (Quadrature Phase Shift Keying). Several data bit rate thus might be tested. Data are transmitted with a bit rate ranging from 8.75 kbps to 23.3 kbps (35 kHz carrier frequency) or 6.7 kbps to 40 kbps (20 kHz carrier frequency). In reception, acoustic signals are received using 4 hydrophones. Fig. 4 illustrates the sea trials system for communication between the underwater unit and a surface vessel (AvII).

The shown result concerns a transmission of a speech signal during five minutes. Speech signal is compressed down to 5.45 kbps, giving a raw data bit rate of 6.7 kbps, with the QPSK modulation and the 20 kHz carrier frequency. Distance between emitter and receiver is about 2000 meters. Sea trials consist in successive emissions of speech signals. Emitted from a reference file, they give an idea of underwater channel effects on the synthesized speech. In reception, spacing between two adjacent of the 4 hydrophones is approximately equal to 25 cm.

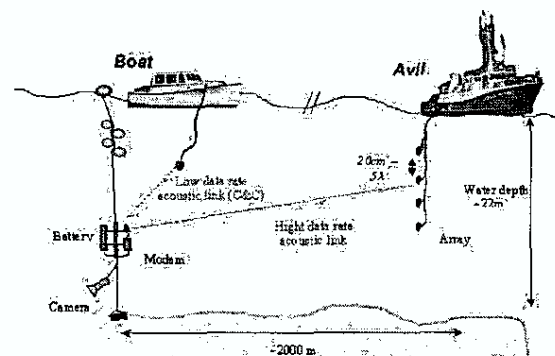


Fig. 4. Sea trials configuration

Receiver performance is evaluated by the MSE measurement, at equalizer output, and input and output equalizer constellation. Equalizer filters are designed as

follows: transversal filters: 11 taps, recursive filter : 35 taps taking into account channel impulse response duration. Fig. 5 shows the MSE evolution. We can note a good behavior over the sequence duration. Due to two perturbations, the speech signal is not correctly synthesized all the time.

The equalizer switches when these channel perturbations occur. When the trust in decided data is sufficient, the equalizer evolves towards its tracking mode. The possibility to switch from one structure to another according to the channel severity proves equalizer robustness.

In fig. 6 input (sensor 1) and output constellations are plotted for a sequence of 20 seconds (between the 12<sup>th</sup> and 32<sup>nd</sup> seconds of the recorded file). Output constellation proves both receiver working effectiveness and equalizer correctness. Input constellation obviously states a strong ISI (Inter Symbol Interference) created by multipath within time-dispersive channel.

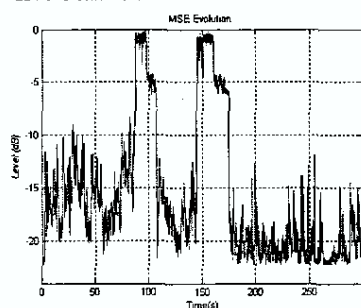


Fig. 5. MSE evolution during 300 s

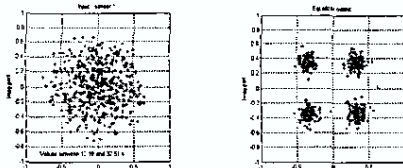


Fig. 6. Input and output equalizer constellations between the 12<sup>th</sup> and 32<sup>nd</sup> s

Fig. 7 shows the synthesized speech after underwater acoustic transmission. We can note perturbation effects on synthesized speech. Let us recall that, for the moment, information bits are transmitted without channel coding.

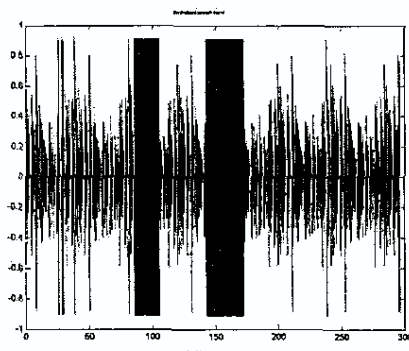


Fig.7 Synthesized speech after acoustic transmission

## V. CONCLUSION AND PERSPECTIVES

This paper presented a high data rate acoustic link named TRIDENT, developed by GESMA, ENST Bretagne and ORCA Instrumentation. After some recalls of the TRIDENT system, sea trials and test configuration are described in order to validate the system for different kinds of information.

According to the results obtained from 10 hours of recording, one will have noted the robustness of this multi-channel equalizer. For strongly non-stationary channels, diversity combining allows to increase receiver performances.

Various tests show that real time transmission of information (images, speech, text and data...) can be achieved in harsh channels such as the underwater acoustic channel. In this context, next introduction of a channel coding should improve the transmission robustness and protect the data transmitted from possible errors. To enhance the speech reached range, the use of a lower bit rate speech coder (Mixed Excited Linear Prediction working at 2400 bits/s) and lower carrier frequency are under consideration.

## REFERENCES

- [1] Goalic, A.; Labat, J.; Trubuil, J.; Saoudi, S.; Rioualen, D. 'Toward a digital acoustic underwater phone', OCEANS '94. Proceedings, Volume: 3, 13-16 Sept. 1994, Pages : III/489 - III/494 vol.3
- [2] G Lapiere, J Labat, J Trubuil, ' Evaluation of high dated spleen acoustic link: contribution of blind spatio-temporal equalization', Proc. ECUA'2000, Lyon, France, Vol.1, pp. 549-554.
- [3] J. Labat et al, " Adaptive device of equalization for numerical communication systems ", France Telecom Patent n°9510832, 15/09/95.
- [4] J. Labat and C. Laot, 'Blind adaptive multiple input decision feedback equalizer with a self optimized configuration', IEEE trans on Comm., Vol. 49, N°4, April 2001.
- [5] J. Trubuil, T. Le Gall, G. Lapiere, J. Labat, 'Development of a real-time high data rate acoustic link', in Proc Oceans 2001, Hawaii, Vol. 4, pp 2159-2164.
- [6] F. Gardner, 'A BPSK, QOSK timing-error detector for sampled receivers', IEEE Trans. on Comm., vol COM-34, N°5, May 1986.
- [7] Goalic A., Saoudi S, 'An intrinsically reliable and fast algorithm to compute the line spectrum pairs (LSP) in low bit rate CELP coding', Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on , Volume:1, 9-12 May 1995 Pages : 728 - 731 vol.1

# POTENTIAL APPLICATIONS OF LOW COMPLEXITY NON-BINARY HIGH CODE RATE BLOCK TURBO CODES

Rong Zhou, Annie Picart, Ramesh Pyndiah and André Goalic  
GET-ENST de Bretagne, Département SC, TAMCIC(CNRS-FRE 2658 )  
Technopôle de Brest Iroise, CS 83818 - 29238 Brest Cedex, FRANCE

## ABSTRACT

*High code rate Block Turbo Codes (BTC) using Bose-Chaudhuri-Hocquenghem (BCH) codes have already demonstrated near Shannon performances for Quadrature Phase-Shift Keying (QPSK) over Additive White Gaussian Noise (AWGN) channel. We show here that very low complexity Reed-Solomon (RS) BTC can also achieve reliable transmission at less than 1 dB from Shannon limit with the same transmission conditions. Performances of these RS-BTC over Binary Symmetric Channel (BSC) and Binary Erasure Channel (BEC) are also investigated in this paper. Potential applications for this study are high speed optical transmission and data storage.*

## INTRODUCTION

Since the introduction of Convolutional Turbo Codes (CTC) [1], turbo principle has been extended to different functions of a digital communication system. The general concept of iterative Soft-Input-Soft-Output (SISO) decoding has been extended to Block Turbo Codes (BTC) [2] and also to Low Density Parity Check (LDPC) codes [3]. BTC are very efficient for high code rate ( $R > 0.7$ ) applications. BTC using concatenated Bose-Chaudhuri-Hocquenghem (BCH) codes can achieve reliable transmission with Quadrature Phase-Shift Keying (QPSK) over AWGN channel at less than 1 dB from Shannon limit with very high decoding speed and reasonable decoding complexity [4]. The main limitation for BCH-BTC is that very large coded blocks (>60,000 bits) are required to achieve high code rates ( $R > 0.9$ ).

Our motivation here is to investigate the performance of non-binary BTC and its potential applications. The most widely used non-binary code is Reed-Solomon (RS) code which has the MDS (Maximum Distance Seperable) property. Thanks to this property, RS code exhibits a higher code rate than BCH codes for a given Minimum Hamming Distance (MHD) and code length. Thus it is a very attractive candidate for high code rate BTC. The performances of RS-BTC using extended (or expurgated) RS component codes were declared not as good as BCH-BTC with respect to Shannon limit [5]. After a theoretical analysis and com-

puter simulations, we propose to use single error-correcting RS component codes to construct RS product codes. The so constructed RS product codes can achieve reliable transmission at less than 1 dB from Shannon limit when using QPSK modulation over AWGN channel. The block size of the adopted RS product codes is almost three times smaller than that of BCH-BTC of similar high code rate while still maintaining very low decoding complexity. The performances of these RS-BTC over Binary Symmetric Channel (BSC) and Binary Erasure Channel (BEC) are also investigated in this paper. The results show that they are very attractive for applications such as high speed optical transmission and data storage.

The paper is organized as follows. In Section II we recall the basic concepts of RS product code, turbo decoding and transmission over AWGN, BSC and BEC channel. Section III is dedicated to the performance investigation of RS product codes based on different component codes. Simulation results of the chosen RS product codes over three above Channels are shown in Section IV. Section V draws some general conclusions.

## ITERATIVE DECODING OF RS-BTC

The concept of product code introduced by P. Elias in 1954 is a simple and efficient method to construct powerful codes. Product codes are obtained by means of serial concatenation of two (or more) linear block codes  $e^1$  having parameters  $(n_1, k_1, \delta_1)$  and  $e^2$  having parameters  $(n_2, k_2, \delta_2)$ .  $n_i$ ,  $k_i$  and  $\delta_i$  ( $i=1,2$ ) stand respectively for code length, code dimension and Minimum Hamming Distance (MHD) of each component code. The parameters of the constructed product code  $P$  are given by  $n = n_1 \times n_2$ ,  $k = k_1 \times k_2$ ,  $\delta = \delta_1 \times \delta_2$ . The code rate of  $P$  is  $R = R_1 \times R_2$  where  $R_i$  is the code rate of its component codes. It is shown [6] that all  $n_1$  rows of the matrix are code words of  $e^2$  just as all  $n_2$  columns are code words of  $e^1$  by construction.

RS codes are a subclass of non-binary BCH codes.  $RS(n, k, \delta)$  is a Maximum Distance Seperable (MDS) code which has the minimal code length for a given code rate and error correcting capability. RS product codes can be constructed based on either  $Q$ -ary symbol concatenation or bit concatenation. In order to get a smaller block size,  $Q$ -ary symbol

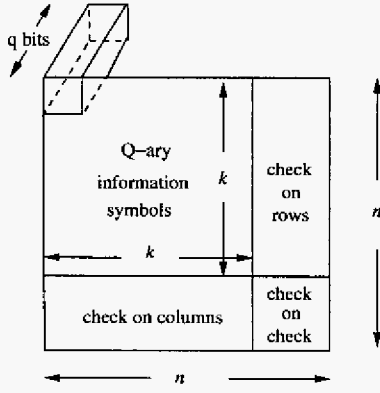


Figure 1: RS product codes with  $Q$ -ary symbol concatenation

based RS product code is used here. It is constructed from  $(k_1 \times k_2)$   $Q$ -ary ( $Q = 2^q$ ) information symbols, which means  $(k_1 \times k_2 \times q)$  data bits. The  $q$  binary elements corresponding to a  $Q$ -ary symbol are at the intersection of a row and column. Serial concatenation is realized by coding  $Q$ -ary symbols along the rows and columns successively. Our RS product code  $P$  is constructed by using two identical component codes  $e^1 = e^2 = RS(n, k, \delta)$  and is shown in Fig. 1. By using the galois field generator polynomial [7], the above RS product codes can be represented by a matrix  $[E]$  of  $n$  rows and  $(n \times q)$  columns after the  $Q$ -ary to binary conversion.

The transmission with QPSK modulation over AWGN channel can be regarded as the superposition of two independent Binary Phase-Shift Keying (BPSK) modulations applied respectively to the in-phase and orthogonal carrier. Each bit of the RS product code is associated with one binary value according to the mapping rule ( $0 \rightarrow -1, 1 \rightarrow +1$ ). At the channel output, the bit level Log Likelihood Ratio (LLR) is computed and fed to the following turbo decoder.

The transmissions over BSC channel is modeled by the transmission with BPSK modulation over AWGN channel and hard decision with one threshold applied to the bit level LLR. Thus two level LLR values  $+A$  and  $-A$  are sent to the turbo decoder. As for BEC channel, the only difference when compared to the BSC channel is that two symmetric thresholds are used in the hard decision stage and thus three level LLR values  $\{+A, 0, -A\}$  are fed to the following turbo decoder.

The turbo decoder is made up of several cascaded Soft-Input-Soft-Output (SISO) decoder. Each SISO decoder is used for decoding rows and columns. A SISO decoder can be divided into a Soft-Input-Hard-Output (SIHO) decoder and a reliability estimation of each decoded bit at the output of the SIHO decoder. The Chase's algorithm [8] is extended to construct the SIHO decoder for  $RS(n, k, \delta)$  over

$GF(Q)$ . Let the vector  $R = (r_1, r_2, \dots, r_i, \dots, r_{qn})$  be the normalized bit level LLR sequence after soft-demodulation for one received word and  $Y = (y_1, y_2, \dots, y_i, \dots, y_{qn})$  be the hard-decision estimation of the transmitted sequence with the relationship  $y_i = 0.5 \times [\text{sign}(r_i) + 1]$  for  $1 \leq i \leq (q \times n)$ . Firstly the  $s$  least reliable positions corresponding to the smallest absolute values in vector  $R$  are determined. Test patterns of length  $qn$  of weight 0 to  $s$  with the "1" restricted to the least reliable positions are added to the sequence  $Y$  to obtain new binary sequences. A  $Q$ -ary sequence  $H = (h_1, \dots, h_i, \dots, h_n)$  is then obtained based on each new binary sequence. The algebraic RS decoder processes these  $Q$ -ary sequences and produces the Maximum-Likelihood (ML) RS codewords. These ML codewords are then represented by binary sequences of length  $qn$  after the  $Q$ -ary to binary decomposition and noted as  $C^i$  for  $1 \leq i \leq 2^s$ . For each codeword  $C^i$ , the Chase algorithm computes the Euclidean Distance (ED) to the input sequence  $R$  and selects code word  $D$  at minimum ED from  $R$ . For each bit  $d_j$  ( $1 \leq j \leq qn$ ) of  $D$ , we search for the concurrent code word  $C$  at minimum ED from  $R$  such that  $c_j \neq d_j$  at position  $j$ . If such a code word is available, the soft output for bit  $d_j$  is given by [2] :

$$r'_j = \left( \frac{M^C - M^D}{4} \right) \times d_j \quad (1)$$

where  $M^D$  is ED between  $D$  and  $R$  and  $M^C$  is ED between  $R$  and  $C$ . Else we use the predefined value  $\beta$  and the following relation :

$$r'_j = r_j + \beta \times d_j \quad (2)$$

The extrinsic information  $W$  which is the difference between soft output and soft input for each component is computed and sent to the next SISO decoder. Input to the next SISO decoder is given by  $R + \alpha W$  where  $\alpha$  is used to reduce the influence of extrinsic information in the first iterations.

The complexity of this RS SISO decoder depends on parameter  $s$ . Given that the soft decoding criterion is to find the RS binary code word of length  $qn$  at minimum ED from observation vector  $R$ , then  $s$  is given by  $\lceil \delta_b/2 \rceil$  where  $\delta_b$  is the minimum Hamming distance of RS binary code words. In the case of single-error correcting codes,  $\delta_b = \delta = 3$  and thus  $s = 1$  is sufficient for SIHO decoding.

A simple stopping criterion based on the syndrome check is integrated in turbo decoding process to reduce the average number of iterations [9]. Note that if the algorithm has converged to a wrong code word, these errors are taken into account to determine the number of errors in the subsequent iterations.

## PERFORMANCES OF RS PRODUCT CODES

Prior to turbo codes, the Asymptotic Coding Gain (ACG) was regarded as the significant parameter to evaluate the error correction capability of a code. It is defined as follows [10]:

$$G_a = 10 * \log_{10}(R\delta_b) \quad (3)$$

where  $R$  is the code rate and  $\delta_b$  is the MHD of the code. In order to maximize ACG of a code, it is necessary to maximize the product  $R\delta_b$ . Although this parameter is important, it has the disadvantage of being dependent of the code itself. Shannon limit is a more general parameter which depends on  $n$ ,  $k$  and the used modulation which is independent of the code [11]. Thus Shannon limit will be considered to compare the theoretical performances of RS product codes.

Theoretical Shannon limit ( $SH$ ) of different RS product codes are given in Table 1. We have considered product codes using classical, extended and expurgated RS component codes over GF(16), GF(32) and GF(64). The differences in theoretical Shannon limit ( $\Delta SH$ ) are calculated by taking classical RS( $t = 1$ ) based product code as a reference.

Table 1: Shannon Limit with QPSK Over AWGN Channel

	$K(bits)$	$MHD$	$R$	$SH(dB)$	$\Delta SH(dB)$
$RS(15, 13)^2$	676	9	0.7511	2.739	0
$RS(16, 13)^2$	676	16	0.6602	2.089	-0.65
$RS(15, 12)^2$	576	16	0.6400	2.049	-0.69
$RS(31, 29)^2$	4205	9	0.8751	3.379	0
$RS(32, 29)^2$	4205	16	0.8213	2.729	-0.65
$RS(31, 28)^2$	3920	16	0.8158	2.699	-0.68
$RS(63, 61)^2$	22323	9	0.9375	4.139	0
$RS(64, 61)^2$	22326	16	0.9084	3.569	-0.57
$RS(63, 60)^2$	21600	16	0.9070	3.549	-0.59

From Table 1, we observe that when replacing the classical RS( $t = 1$ ) component codes by other RS component codes having MHD greater than 3, there is a significant reduction in code rate at product code level. This significant reduction in code rate translates into a significant reduction in Shannon limit. When using expurgated (or extended) RS product codes, the only way to achieve the optimum performance is to increase the number of test patterns in the SISO decoder, which requires a more complicated turbo decoder. Furthermore RS codes can correct several erroneous bits in a  $Q$ -ary symbol at one time, the error floor phenomena can be mitigated in the low error rate region. To summarize, RS product codes with MHD equal to 9 can achieve a good trade-off between complexity and performance.

In order to confirm this theoretical analysis, we simulate product codes based on RS( $t = 1$ ) codes under the same simulation conditions as in [5] and compare these performances to those of expurgated RS product codes in [5]. The difference between signal to noise ratio per information bit  $E_b/N_o$  for  $BER = 10^{-5}$  and the corresponding Shannon Limit is calculated and noted as  $\Delta SH$  in Table 2.

Table 2: Classical RS-BTC vs. Expurgated RS-BTC

	$\Delta SH = E_b/N_o - SH$
$RS - BTC(15, 13)$	1.05 dB
$RS - BTC(15, 12)$	1.90 dB
$RS - BTC(31, 29)$	1.10 dB
$RS - BTC(31, 28)$	1.70 dB
$RS - BTC(63, 61)$	1.20 dB
$RS - BTC(63, 60)$	1.45 dB

With the above theoretical analysis and simulation results, we can conclude that the penalty in terms of code rate and Shannon limit is too high for using expurgated (or extended) RS product codes. Thus we propose here to use single error-correcting RS component codes in the construction of RS product codes.

## SIMULATION RESULTS

Following the above analysis, four product codes  $RS(7, 5)^2$ ,  $RS(15, 13)^2$ ,  $RS(31, 29)^2$  and  $RS(63, 61)^2$  based on four different Galois Fields  $GF(8)$ ,  $GF(16)$ ,  $GF(32)$ ,  $GF(64)$  have been considered. These four product codes are transmitted by QPSK modulation over AWGN channel.

The number of test patterns in the simulations are limited to 16 as for BCH-BTC in order to have a low complexity turbo decoder. Extrinsic information is normalized as described in [12] and coefficients  $\alpha$  and  $\beta$  are respectively:  $\alpha = (0, 0.1, 0.2, 0.25, 0.3, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.9, 1.0, 1.0)$  and  $\beta = (0.2, 0.3, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 1.0, 1.0, 1.0, 1.0)$ . The BER curves are shown in Fig. 2 for 8 iterations with stopping criterion. The BER is given for 400 matrices having at least one erroneous  $Q$ -ary symbol after decoding down to a  $BER$  of  $10^{-5}$ . In Fig. 2, we have also plotted the performance of RS-BTC with 32 test patterns based on five least reliable bits. The coding gain is improved by less than 0.15 dB which shows that 16 test patterns is a good trade-off between complexity and performance.

We have also calculated Shannon limit ( $SH$ ) as mentioned in the previous Section. The gaps  $\Delta Sh$  between Shannon Limit and  $E_b/N_o$  for  $BER$  equal to  $10^{-5}$  when using 16 and 32 test patterns are listed in Table 3. We can see that with a proper choice of component codes,

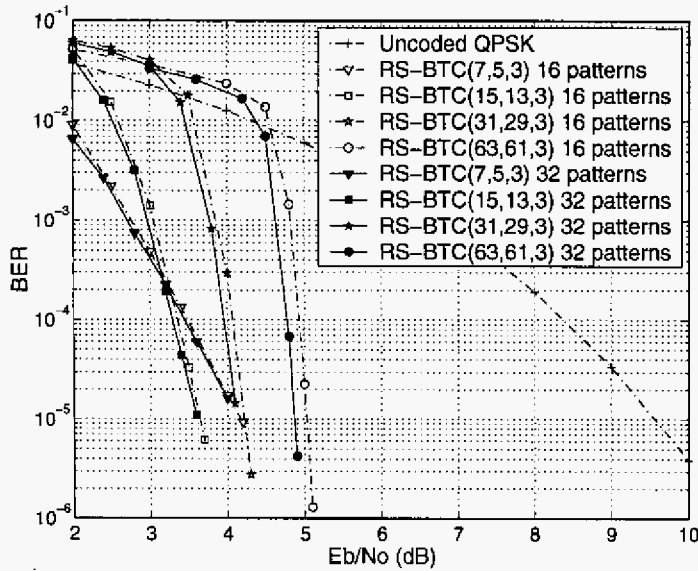


Figure 2: RS-BTC using QPSK over AWGN channel with 16 and 32 test patterns (a maximum of 8 iterations)

these above RS-BTC can achieve reliable transmission using QPSK modulation over AWGN channel at less than 1 dB from Shannon limit. Furthermore these near Shannon performances are obtained by using a very low complexity turbo decoder with only 16 test patterns.

Table 3:  $\Delta SH$  for RS( $t=1$ ) based BTC using QPSK over AWGN channel

	16 patterns	32 patterns
$RS - BTC(7, 5)$	0.90 dB	0.90 dB
$RS - BTC(15, 13)$	0.90 dB	0.85 dB
$RS - BTC(31, 29)$	0.82 dB	0.72 dB
$RS - BTC(63, 61)$	0.84 dB	0.70 dB

The performances of some above RS-BTC are compared to those of BCH-BTC with similar code rates under the condition of QPSK over AWGN channel. The comparison parameters are : information block size  $K$ , code rate  $R$  and the gap to Shannon limit  $\Delta SH$ . They are shown in Table 4.

We can see from Table 4 that all above BCH-BTC and RS-BTC can achieve reliable communication with QPSK modulation over AWGN channel. However RS-BTC require data block size  $K$  at least three times smaller than BCH-BTC of similar code rate. For high code rate ( $R > 0.9$ ) applications, the problem of requiring very large coded blocks in BCH-BTC case can be fully solved by using a RS-BTC of similar code rate. Furthermore, this gain in hardware complexity (memory size) is obtained by using the same

Table 4: BCH-BTC vs. RS-BTC When Using QPSK over AWGN channel

	$K$ (bits)	$R$	$\Delta SH$ (dB)
$RS - BTC(15, 13)$	676	0.75	0.80
$BCH - BTC(64, 57)$	3249	0.79	0.88
$RS - BTC(31, 29)$	4205	0.88	0.80
$BCH - BTC(128, 120)$	14400	0.88	0.83
$BCH - BTC(256, 239)$	57121	0.87	0.90
$RS - BTC(63, 61)$	22326	0.94	0.80
$BCH - BTC(256, 247)$	61009	0.93	0.70

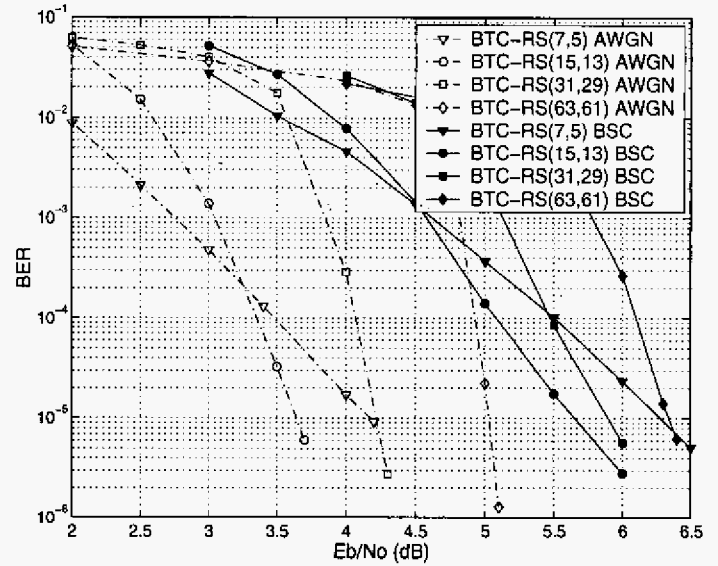


Figure 3: RS-BTC with QPSK over AWGN channel and over BSC channel using 16 test patterns (a maximum of 8 iterations)

low complexity turbo decoder.

Now we consider the performance of these RS-BTC over BSC and BEC channel. These channels correspond to the case where soft outputs to the turbo decoder are not available due to economical or technological reasons. For BSC channel case, the inputs to the turbo decoder are binary values  $+A$  or  $-A$ . While for BEC channel, one of three possible states  $\{+A, 0, -A\}$  is fed to the following turbo decoder. The above RS product codes are simulated over BSC and BEC channel with the same conditions as before and the performance curves are shown in Fig. 3 and Fig. 4.

The gaps between soft inputs and hard inputs for a BER of  $10^{-5}$  are computed. In Table 5, we have shown the performance gaps  $\Delta_{BSC}$  for BSC channel and  $\Delta_{BEC}$  for BEC channel by taking those performances over AWGN channel as a reference. We can observe that these gaps decrease as

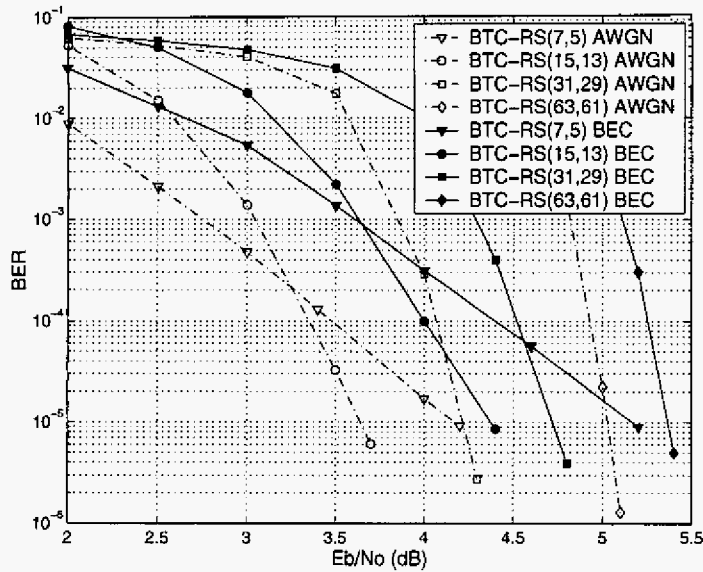


Figure 4: RS-BTC with QPSK over AWGN channel and over BEC channel using 16 test patterns (a maximum of 8 iterations)

we increase the code length. For a given code, these gaps decrease when we increase the state number of LLR values at the entrance of the turbo decoder. For example, with the longest code  $RS(63,61)^2$ , the gap between soft input and binary input is 1.31 dB while the gap between soft input and ternary input is 0.35 dB. These gaps are practically constant for BER lower than  $10^{-4}$ . This is due to the fact that the extrinsic information tends to a Gaussian distribution after the first iterations thanks to the central limit theorem. For long code with high code rate, these gaps are significantly less than the 2-3 dB gap usually observed with other classical codes [9].

Table 5: Gaps between soft inputs and hard inputs for  $RS(t=1)$  based BTC

	$\Delta_{BSC}$	$\Delta_{BEC}$
$RS - BTC(7,5)$	2.10 dB	0.99 dB
$RS - BTC(15,13)$	2.03 dB	0.74 dB
$RS - BTC(31,29)$	1.67 dB	0.49 dB
$RS - BTC(63,63)$	1.31 dB	0.35 dB

## CONCLUSIONS

In this paper we show that product codes based on single error-correcting RS component codes can achieve reliable transmission at less than 1 dB from Shannon limit

with QPSK modulation over AWGN channel using very low complexity turbo decoder. Previous contribution [5] on RS-BTC considered the extended (or expurgated) RS component codes which introduce a significant penalty in terms of code rate and requires the use of very large number of test patterns in the SISO decoder to achieve optimum performance. At the same time, we show here that high code rate RS-BTC codes have also good performances over BSC and BEC channel.

The proposed RS product codes can exhibit a much smaller block size than BCH product codes of similar code rates. As the coding/decoding delay and memory size is proportional to the size of data block, this gives a serious advantage to the proposed RS-BTC over BCH-BTC for high code rate applications. In addition, the RS encoder/decoder is based on  $Q$ -ary symbol processing instead of bit processing. The coding/decoding speed is thus  $\log_2 Q$  times higher.

The small performance degradations of RS-BTC over BSC and BEC channel with respect the performance over AWGN channel make them very attractive for high data rate systems. Potential applications are high speed optical transmission and data storage.

## References

- [1] C.Berrou, A.Glavieux and P.Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", IEEE Int. Conf. on Comm. ICC'93, vol.2/3, May 1993, pp.1064-1071.
- [2] R.Pyndiah, A.Glavieux, A.Picart and S.Jacq, "Near optimum decoding of product codes", in Proc. of IEEE Globecom '94, vol.1/3, Nov.-Dec.1994, pp.339-343.
- [3] R.G. Gallager, "Low Density Parity Check Codes", IRE Trans. on Information Theory, Jan. 1962, pp.21-28.
- [4] J.Cuevas, P.Adde, S.Kerouedan and R.Pyndiah, "New architecture for high data rate turbo decoding of product code", Globecom '02, vol.2, Nov. 2001, pp.339-343.
- [5] O.Aitsab and R.Pyndiah, "Performance of Reed-Solomon block turbo code", Globecom'96, vol.1, Nov. 1996, pp.121-125.
- [6] F.J.MacWilliams and N.J.A.Sloane, "The theory of error correcting codes", North-Holland publishing company, 1978, pp.567-580.
- [7] E.R. Berlekamp, "Algebraic coding theory", McGraw-Hill Book Company, New York, 1968
- [8] D.Chase, "A class of algorithms for decoding block codes with channel measurement information", IEEE Trans. Inform. Theory, vol IT-8, Jan. 1972.



- [9] R.Pyndiah and P.Adde, "Performance of high code rate BTC for non-traditional applications", 3<sup>rd</sup> International symposium on turbo codes & related topics, Sep. 2003.
- [10] G.C.Clark Jr. and J.Bibb Cain, "Error-Correction coding for digital communications", Plenum Press, New York, 1981.
- [11] R.G. Gallager, "Information Theory and Reliable Communication", John Wiley & Sons, New-York 1968.
- [12] R.Pyndiah, A. Picart and A. Glavieux, "Performance of block turbo coded 16-QAM and 64-QAM modulations", Globecom'95, Sigapore, Nov. 1995, pp. 1039-1043.



# Real-Time Turbo Decoding of Block Turbo Codes using the Hartmann-Nazarov algorithm on the DSP TEXAS TMS320C6201

André Goalic, Karine Cavalec-Amis and Vincent Kerbaol

Département Signal et Communications

ENST Bretagne, Technopôle Brest Iroise, BP 832, 29285 BREST Cedex, BRITTANY

**Abstract**—Product codes were introduced in 1954 by P. Elias and consist of a serial concatenation of two linear block codes with an uniform interleaver. The coding process only uses two linear block codes and creates a matrix in which both columns and rows are codewords. The decoding process can use different algorithms. After first implementing the Chase algorithm, we present in this paper another decoding process including the *Hartmann-Nazarov* algorithm and the Fast Hadamard Transform (FHT). This algorithm uses a symbol-by-symbol decoding rather than a whole word decoding. The decoding process is implemented on the fixed point DSP TMS320C6201. It communicates, via the RTDX link, with the coder, written in C++ language and running on the PC. The Hartmann-Nazarov algorithm exhibits better performance on the Rayleigh channel. A first evaluation of the bit rates gives us a value close to 500 Kbits/s on the DSP clocked at 250 MHz.

## I. INTRODUCTION

Channel coding is one of the main part of a digital communication system. The redundancy bits help the decoder to get back the emitted sequence. The DSP implementation gives us both flexibility and adaptability. While using product codes based for example on a BCH(32,26,4) scheme, the coding gain at  $10^{-5}$  Bit Error Rate slightly exceeds 6 dB on a Gaussian channel. Before the encoding process, the information bits are stored in a matrix. Then the Block Turbo Code (BTC) encoder adds redundancy bits to rows and columns, such that the sets of rows and columns are codewords. An iterative process is used to decode the product code. Section II describes the decoding algorithm. The problem of real time decoder implementation is detailed in section III. Section IV presents the demonstration platform using the Code Composer Studio and the RTDX link. Section V shows some promising results obtained in assembly language. Finally, conclusion and future perspectives of the real time TCB implementation on a DSP chip are given.

## II. SOFT DECODING OF BTC

### A. The Product Code

Let us consider a linear block code  $\mathcal{C}$  having parameters  $(n, k, \delta)$  where  $n, k, \delta$  stand for codeword length, number

of information bits and minimum Hamming distance respectively. The code rate is defined by the ratio  $r = k/n$ . The product code  $\mathcal{P} = \mathcal{C} \otimes \mathcal{C}$  is then obtained by :

- placing  $(k \times k)$  information bits in an array of  $k$  rows and  $k$  columns
- coding  $k$  rows and  $k$  columns using code  $\mathcal{C}$ .

The parameters of the resulting product code  $P$  are given by:  $N = n \times n$ ,  $K = k \times k$ ,  $\Delta = \delta \times \delta$  and the code rate  $R = r \times r$ . The  $(n - k)$  last rows and  $(n - k)$  last columns also are codewords of  $\mathcal{C}$ . A symbol-by-symbol soft decision algorithm introduced in 1976 by Hartmann *et al.* [1] and modified by Nazarov *et al.* [2] is used to compute extrinsic information.

### B. The Hartmann-Nazarov soft-decision algorithm

This algorithm offers an optimal symbol-by-symbol decoding of binary block codes using the dual code. Moreover the use of the Fast Hadamard Transform (FHT) allows good performance in a real time environment. For each row or column of the product code, let  $m = (m_i; 0 \leq i < k)$  denotes information symbols for codeword  $B = (b_l; 0 \leq l < n)$  of an elementary binary code  $\mathcal{C}$  with a generator matrix  $G = (g_{il}; 0 \leq i < k, 0 \leq l < n)$ , parity check matrix  $H = (h_{il}; 0 \leq i < n - k, 0 \leq l < n)$  and finally, the received word  $R = (r_i; 0 \leq i < n)$ . It is also assumed that codeword components are mapped from  $\{0, 1\}$  to  $\{-1, +1\}$  before transmission. The decoder computes the *a posteriori* probability  $\Pr(b_l|R)$  for each component  $b_l$  of the transmitted codeword. Considering the memoryless channel, the optimum symbol-by-symbol decoding algorithm can be written as  $C(l) = \Pr(b_l = 0|R) - \Pr(b_l = 1|R)$ . Nazarov expressed  $C(l)$  as:

$$C(l) = \frac{1}{\sum_{\gamma=0}^{2^{n-k}-1} D_{\gamma}} \sum_{\gamma=0}^{2^{n-k}-1} D_{\gamma} \times \Lambda_l(\gamma) \quad (1)$$

where

$$\Lambda_l(\gamma) = \begin{cases} \rho_l & \text{if } t_{\gamma}(l) = 0, \\ \frac{1}{\rho_l} & \text{if } t_{\gamma}(l) = 1. \end{cases} \quad (2)$$

$$D_{\gamma} = \prod_{l=0}^{n-k} \rho_l^{t_{\gamma}(l)} \quad (3)$$

(email: andre.golic@enst-bretagne.fr), Tel.: +33 2 29 00 13 78, Fax: +33 2 29 00 10 12

and

$$t_\gamma(l) = \sum_{m=0}^{n-k-1} \gamma_m \times h_{ml} \text{ if } y = \sum_{m=0}^{n-k-1} \gamma_m \times 2^m \quad (4)$$

The set  $h_{ml}$  represents the coefficients of the dual code matrix  $H$  where  $m$  is the row index and  $l$  the column index. The component  $\rho_l$  is defined as :

$$\rho_l = \frac{p(r_l, b_l = 0) - p(r_l, b_l = 1)}{p(r_l, b_l = 0) + p(r_l, b_l = 1)} \quad (5)$$

It must be stressed at this point that the last expression significantly differs from the expression of  $\rho_l$  given in [2]. Indeed, joint probability density functions (pdf)  $p(r_l, b_l)$  now appear instead of conditional pdfs. As already stated by Hagenauer *et al.* [3], derivations presented in [1], [2] in case of a hard output symbol-by-symbol MAP decoder, logically assumed equally probable codeword bits  $b_l$ . To account for the re-estimation of the *a priori* bit probabilities during the iterative decoding process, the expression of  $\rho_l$  is now modified to include both contributions of channel output measurement and *a priori* information. Thus, introducing the conditioned log-likelihood ratio  $LLRi_l^{(0)}$  defined as,

$$LLRi_l^{(0)} = L(r_l|b_l) = \ln \left( \frac{p(r_l|b_l = 1)}{p(r_l|b_l = 0)} \right) = 2r_l/\sigma^2 \quad (6)$$

where  $\sigma$  represents the standard deviation of the observation  $R$ , the expression of  $\rho_l$  before decoding (i.e. assuming equiprobable bits) can be written as:

$$\rho_l = \frac{1 - \exp(L(r_l|b_l))}{1 + \exp(L(r_l|b_l))} = -\tanh \left( \frac{LLRi_l}{2} \right) \quad (7)$$

Replacing  $LLRi_l$  by its value  $(2/\sigma^2) \times r_l$ , we obtain

$$\rho_l = -\tanh \left( \frac{r_l}{\sigma^2} \right) \quad (8)$$

It is clear that the modification of  $\rho_l$  expression now enables to introduce extrinsic information as *a priori* information. After the first row (or column) decoding, arguments of the  $\tanh()$  function in equations (7) and (8) will be augmented with the extrinsic information as detailed in the next paragraph. With the knowledge of all the components  $\rho_l$  ( $l = 0, \dots, n-1$ ) two FHT  $F_\rho$  and  $F_q$  defined below can be computed.

- Computation of  $F_\rho$

$$F_\rho(v) = \sum_{l=0}^{n-1} \ln(|\rho_l|) \exp \left( j\pi \sum_{m=0}^{n-k-1} v_m h_{ml} \right) \quad (9)$$

$F_\rho$  can also be written by the expression below,

$$F_\rho(v) = \sum_{p=0}^{2^{n-k}-1} f_\rho(p) (-1)^{\langle p, v \rangle} \quad (10)$$

with,

$$f_\rho(p) = \begin{cases} \ln(|\rho_l|) & \text{if } p = \sum_{m=0}^{n-k-1} h_{ml} 2^m, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

where  $p$  represents the sum modulo two of the different columns of the dual matrix code. As a result,  $F_\rho$  can be obtained by calculating the FHT of the function  $f_\rho$ .

- Computation of  $F_q$

$$F_q(v) = \sum_{l=0}^{n-1} q(l) \exp \left( j\pi \sum_{m=0}^{n-k-1} v_m h_{ml} \right) \quad (12)$$

where  $\text{sign}(\rho_l) = (-1)^{q_l}$ .  $F_q$  can be written as seen previously by the definition of the FHT,

$$F_q(v) = \sum_{p=0}^{2^{n-k}-1} f_q(p) (-1)^{\langle p, v \rangle} \quad (13)$$

with,

$$f_q(p) = \begin{cases} q_l & \text{if } p = \sum_{m=0}^{n-k-1} h_{ml} \times 2^m, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

As a result,  $F_q$  is the FHT of the function  $f_q$ .

From those two FHT an intermediate parameter  $D_\gamma$  is calculated by the formula,

$$D_\gamma = \exp \left[ \frac{1}{2} (F_\rho(0) - F_\rho(\gamma)) \right] \cdot \exp \left[ \frac{1}{2} j\pi (F_q(0) - F_q(\gamma)) \right] \quad (15)$$

Then,

$$F_D(h_l) = \sum_{\gamma=0}^{2^{n-k}-1} D_\gamma \exp \left( j\pi \sum_{m=0}^{n-k-1} \gamma_m \times h_{ml} \right) \quad (16)$$

Furthermore, the parameter  $C(l)$  used to compute the soft output can be calculated from  $F_D(h_l)$  as follows :

$$C(l) = \frac{1}{2} \left[ \rho_l \left( 1 + \frac{F_D(h_l)}{F_D(0)} \right) + \frac{1}{\rho_l} \left( 1 - \frac{F_D(h_l)}{F_D(0)} \right) \right] \quad (17)$$

with  $l$  being the position of the decoded bit in the codeword  $B$ . Finally, the soft output is defined as:

$$LLRo_l = \ln \left( \frac{P_r(b_l = 1|R)}{P_r(b_l = 0|R)} \right) = \ln \left( \frac{1 - C(l)}{1 + C(l)} \right) \quad (18)$$

### C. The iterative decoding

It can be shown that the soft decoder output  $LLRo_l^k$  at the  $k$ -th iteration can be rewritten as:

$$LLRo_l^k = \ln \left( \frac{p(r_l, b_l = 1)}{p(r_l, b_l = 0)} \right) + W_l^{k+1} = LLRi_l^k + W_l^{k+1} \quad (19)$$

where symbol  $b_l$  is supposed to be *a priori* equally distributed and,

$$W_l^{k+1} = \ln \left( \frac{\rho_l (F_D(0) + F_D(h_l)) - (F_D(0) - F_D(h_l))}{\rho_l (F_D(0) + F_D(h_l)) + (F_D(0) - F_D(h_l))} \right) \quad (20)$$

The additional term  $W_l^{k+1}$ , namely the extrinsic information, can be either obtained by subtracting the soft input  $LLRi_l^k$  to the soft output  $LLRo_l^k$  or directly computed from eq. (20). This term plays a major role in the iterative decoding. Indeed, the index  $k+1$  indicates that it will be used as *a priori* information input in the following decoding process of columns (or rows). Thus, the soft input corresponding to the  $l$ -th component of the column (or row) to be decoded at the next  $(k+1)$ -th iteration step is given by  $LLRi_l + W_l^{k+1}$  (note that  $W_l^0 = 0, \forall l \in \{1, \dots, n\}$ ). Eventually, the decoding process is obtained by cascading elementary decoders described in Fig. 1. The block diagram of Fig. 1 is basically the same as in [4] except that empirical parameters  $\alpha(k)$  and  $\beta(k)$  no longer exist.

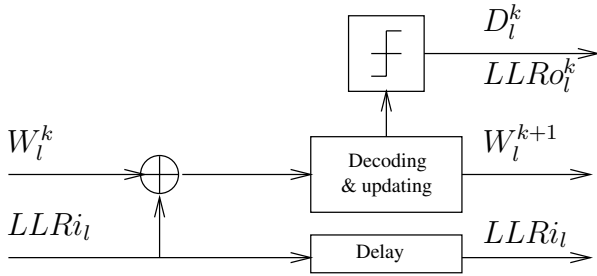


Fig. 1. The iterative process

### III. REAL TIME IMPLEMENTATION

#### A. Mathematical functions computation

The symbol-by-symbol algorithm includes a lot of mathematical functions to generate the extrinsic information. Let us recall their names to check eventual computational complexities. Among them we find the hyperbolic tangent, the Neperian logarithm, the exponential function and also the single division. Of course we have the possibility to use such functions in a fixed point real time implementation but it is clear that the bit rate would be out of interest. Different kinds of technics are usually used to solve these problems:

**Function linearization:** Sometimes the function can easily be linearized using straight segments. It makes possible, easy and fast computation avoiding long search in lookup tables. It is the case for the hyperbolic tangent (Fig. 2). which can be linearized as follows.

$$\tanh(|x|) = \begin{cases} 0.8 |x| & \text{if } |x| \leq 1.0, \\ 0.165|x| + 0.635 & \text{if } 1 < |x| \leq 2, \\ 0.005|x| + 0.955 & \text{if } |x| > 2. \end{cases} \quad (21)$$

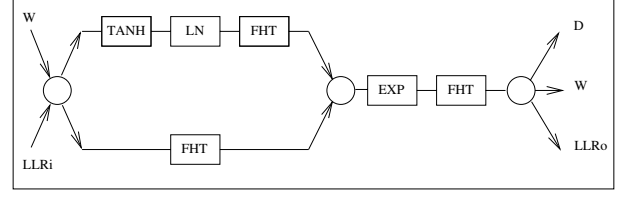


Fig. 2. The decoding process

**Lookup tables and direct offset computation:** In order to keep good performance, as close as possible of the floating point implementation in C language, It is possible to precompute the values of the function for  $2^n$  discrete points ( $n$  will be in this case the quantization bit,  $2^n$  the quantization table). The computation of the Neperian logarithm  $\ln(x)$  function uses this method and it includes two lookup tables according to magnitude of the input.

–  $0 < x \leq 1$  : the segment  $[\epsilon, \dots, 1] (\epsilon \neq 0)$  is divided into  $2^n$  giving  $2^n$  quantization levels. For each level, the pre-computed  $\ln(x)$  value is stored in a table. It is easy to get back the logarithm of  $x$  during the decoding process. As  $x$  is bounded to one in this table, it is enough to multiply its value by  $2^n$  and take the integer part. This value only is the offset to get  $\ln(x)$  in this lookup table.

–  $x > 1$  : in this case we need to have a knowledge of the range of the value of  $x$ . Let us assume that  $x_{\max}$  is the maximum value of  $x$ , then after normalization, the previous method applies.

**Rewriting algorithm to avoid division computation:**

The computation of  $C(l)$  includes a division. It is easy to solve this problem. Introducing the value of  $C(l)$  (17) in the  $LLRo_l$  expression (18), it is possible to rewrite this expression as follows:

$$\begin{aligned} LLRo_l &= \ln \left( \frac{1 - C(l)}{1 + C(l)} \right) = \ln \left( \frac{a(l) - b(l)}{a(l) + b(l)} \right) \\ &= \ln(|a(l) - b(l)|) - \ln(|a(l) + b(l)|) \end{aligned} \quad (22)$$

where,

$$a(l) = 2 \rho_l F_D(0) \quad (23)$$

$$b(l) = \rho_l^2 (F_D(h_l) + F_D(0)) + (F_D(0) - F_D(h_l)) \quad (24)$$

Then,  $LLRo_l$  becomes the difference between two logarithms.

#### B. Assembly Language Implementation

This optimum symbol-by-symbol algorithm offers exact evaluation of the extrinsic information as opposed to the sub-optimal Chase-Pyndiah algorithm [4]. As a consequence the computational complexity is a little bit higher. To efficiently implement such an algorithm, and obtain a significant bit rate, it is of main importance, firstly to choose the well-suited parameter formats and secondly to exploit as often as possible the different possibilities offered by the DSP architecture :

**Format parameter:** The fixed point DSP TMS320C6201 only allows the use of the integer format coded on 8, 16, 32, 40, 64, both signed or unsigned. The small dynamics of a lot of parameters requires the use of mixed numbers, The set of bit of an integer is split into two parts, the most significant bits being for the integer part and the remaining for the fractional part. Our implementation uses signed and unsigned integer on 8, 16 bits and also mixed number 4-12 both signed or unsigned (4 and 12 bits respectively for the integer and fractional part). Moreover this format avoids a lot of memory accesses. All operations directly accept these formats, except the multiply instruction which requires shifting.

**DSP resources:** The DSP architecture offers materials to speed up the decoding process. The implementation systematically makes use of them:

- Single Instruction Multiple Data (SIMD)

To decrease the number of mathematical operations, the DSP offers the SIMD possibility. LDW, STW, ADD2 and SUB2 instructions allow the realization of two operations in a single instruction. Moreover, taking into account the eight functional units and the parallelism, more than eight operations can be executed in a single clock cycle. For example the Fast Hadamard Transform (FHT)(length 32) which only uses the load, store, addition and subtraction requires 640 operations. Only 100 cycles are required to achieve this FHT ( without using multiply units).

- Software pipelining

This technique allows to execute in parallel multiple iterations of the loop. To do so, two possibilities are used:

- \* Loop enrolling

Algorithms often use loops to process data. Usually processor resources are unused, both at the beginning and at the end of a loop. You have to load/store data from/to memory. The way to solve this problem is to enroll the loop, splitting it into three parts : prolog, loop, epilog. In this case functional units are not efficiently used. When processing data sequentially, multiple operations are not performed, so that only some functional units are busy.

- \* “Multi-Tasking”

After preparation in the prolog we have the possibility to process multiple iterations or part of iteration in a single loop. Usually the datum processing includes a few operations separated by waiting periods during which you can deal with another datum. The datum processing is then splitted into a few miniprocessings. Thus the main loop is able to carry on a kind of multitasking. Each loop inherently includes a branch instruction which requires 5 delay slots to be processed. So the most efficient way to process it, is to split the algorithm into miniloops only including six packets of instructions. As previously said, each miniloop includes a few miniprocessings. Of course the epilog has to close correctly the sequence. It is often possible to do so but sometimes the loops are bigger.

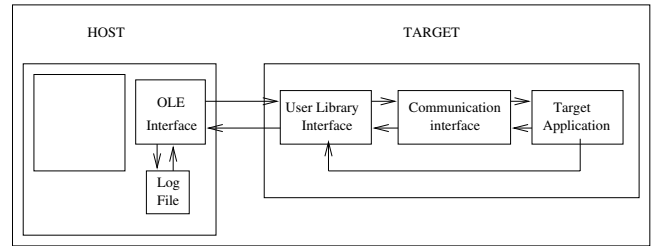


Fig. 3. Demonstration Platform

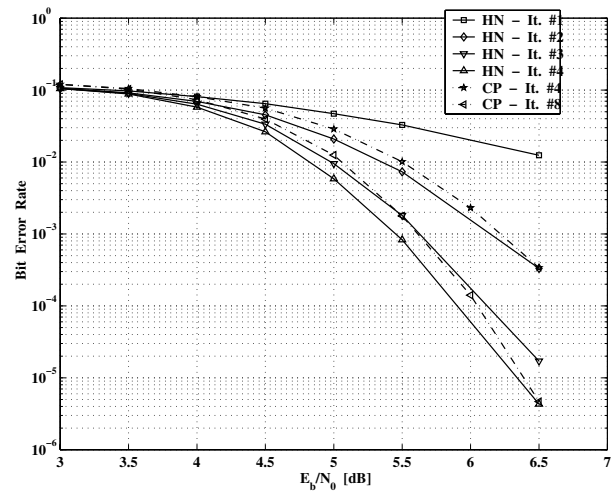


Fig. 4. Performance comparison Hartmann-Nazarov (HN) and Chase-Pyndiah (CH) on Rayleigh Channel

#### IV. CODE COMPOSER STUDIO - RTDX LINK

Designed for the TI high performance DSP TMS320C6201, Code Composer Studio tightly integrates the capabilities:

- real-time data exchange between host and target
- real-time analysis and data visualization.

The demonstration platform (Figure 3) includes two parts:

- the first is implemented in the host computer and, it also includes two tasks:

- bit generation, channel coding

After being generated, the information bit matrix is coded. Then noise is added to the set of bits (information and parity) to introduce channel transmission effects, before being sent to the decoder, via the RTDX link.

- monitoring and statistics, this part also includes Bit Error Rate computation, curves drawing and global test platform monitoring.

- the second part is implemented in the DSP, the decoding process. Each noisy frame is decoded and sent back to the host via the RTDX link, then statistics and curves can be carried out.

#### V. RESULTS

- Floating-point implementation

Due to the symbol-by-symbol extrinsic information eval-

uation, the Hartmann-Nazarov algorithm exhibits better performance than the Chase-Pyndiah one (Fig.4). The turbo-effect starts earlier and the convergence speed appears twice faster. But obviously, in return the complexity is higher. So when comparing both algorithms, it is of great interest to know the comparison criteria including also the energy consumption likely to limit the number of iterations.

- Gaussian Channel-Fixed point implementation

In order to check the performance in terms of Bit Error Rate (BER), the program has been adapted to simulate DSP constraints (input on 8 bits and inner parameters according to their dynamics). Figure 5 shows that the performance obtained for the fixed point implementation are only at 0.2 dB from the floating-point simulation.

- Bit rate

The algorithm is cut up into routines written in assembly language. Table I summarizes the number of cycles required to compute the different routines called by the main program. A first evaluation shows that a total of 1280 cycles is needed to process one row/column. This leads us to a bit rate close to 500 kbits/s with a DSP TEXAS TMS320C6201 clocked at 250 Mhz (2000 MIPS). It is important to note that the main program remains written in C which does not handle mixed numbers. Next expected released should provide higher bit rate.

TABLE I  
CYCLES BY ROUTINES

Routines	Number of cycles
$(W_l^k + LLRi_l^k)$	120
$\text{Log}(\text{abs}(\tanh))$	200
$F_q$	100
$F_p$	100
$D_\gamma$	300
$F_D$	140
$\text{sign}(LLRo_l^k)$	320
TOTAL	1280

## VI. CONCLUSIONS - PERSPECTIVES

In spite of a complexity greater than the Chase-Pyndiah algorithm, the Hartmann-Nazarov gives promising results in real-time environment. So far we cannot compare the two implementations because the Chase-Pyndiah algorithm [5] was not completely optimized. Moreover, an interesting feature of the optimal iterative decoding scheme presented in this paper, being entirely based on the propagation of symbol probabilities, is the removal of the empirical parameters  $\alpha$  and  $\beta$  as needed when using SISO decoder based on the sub-optimal Chase-Pyndiah algorithm [4]. As a consequence, the coding gain at a BER of  $10^{-5}$  is approximately of 0.75 dB on Rayleigh

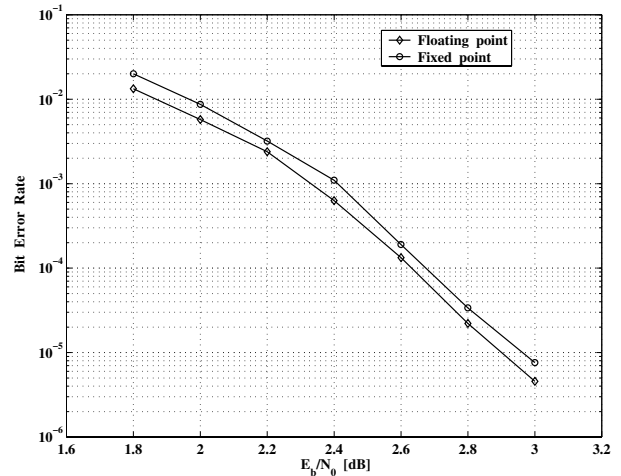


Fig. 5. Bit Error Rate vs SNR

channel. Furthermore, the Hartmann-Nazarov algorithm response-time is shorter. The waterfall region of the BER curve of the Hartmann-Nazarov algorithm occurs earlier than when using Chase-Pyndiah algorithm. The convergence speed turns out to be twice greater. Indeed, figure (4) shows that performance of the Hartmann-Nazarov algorithm at the  $2^{nd}$  iteration are slightly better than Chase-Pyndiah performance obtained at  $4^{th}$  iteration. It is also important to note that the Hartmann-Nazarov algorithm is particularly suitable for decoding one-error correcting ( $t = 1$ ) binary primitive BCH codes (i.e. Hamming codes) in terms of computational gain. Indeed each column of the parity check matrix has the interesting property of being the binary representation of integers between 1 and  $2^{n-k}$ . However, in case of more powerful codes (e.g.  $t = 2$ ), functions  $f_p$  and  $f_q$  as defined in (11) and (14) become *sparse* functions. In other words, the immediate computation of spectral components  $F_p$  and  $F_q$  might be more efficient. Eventually, a first evaluation of the bit rate is very promising, namely 500 kbits/s after 4 iterations. Up to now, we do not use stop criteria which would enable to enhance this rate.

## REFERENCES

- [1] Carlos R.P. Hartmann and Luther D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 5, pp. 514-517, 1976.
- [2] L.E. Nazarov and V.M. Smolyaninov, "Use of Walsh-Hadamard transformation for optimal symbol-by-symbol binary block-code decoding," *Electron. Letters*, vol. IT-34, no. 3, pp. 261-262, 1998.
- [3] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429-445, 1996.
- [4] R. Pyndiah, "Near-optimum decoding of product codes: Block Turbo Codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003-1010, 1998.
- [5] A. Goalic, N. Chapalain and R. Pyndiah, "Implementation of a low complexity algorithm for BCH block turbo decoding on Digital Signal Processor," *ICSPAT'99*, Orlando, Florida, USA, Nov. 1999.

# REAL-TIME TURBO-DECODING OF PRODUCT CODES ON A DIGITAL SIGNAL PROCESSOR

André GOALIC and Ramesh PYNDIAH

TELECOM BRETAGNE, Technopôle Brest Iroise BP-832, 29285 BREST, BRITTANY  
E-mail : Andre.goalic@enst-bretagne.fr

**Abstract** - Recently, a new "Block Turbo Code" algorithm was proposed for decoding product codes. High coding rate with high coding gain can be obtained. Compared to convolutional turbo codes, performances are comparable. To suit different applications, it is desirable to easily modify the component codes parameters. For obtaining both flexibility and adaptability, the DSP option was chosen for implementation. Using product code based on BCH(32, 26, 4) block codes, the coding gain, for a Bit Error Rate of  $10^{-5}$ , slightly exceeds 6 dB. Bit rate reaches 160 Kbit/s on 20 Mips DSP decoders in a cascaded configuration.

## I. INTRODUCTION

The concept of product codes is a simple and relatively efficient method to construct powerful codes using simpler linear block codes. Let us consider a linear block code  $C$  having parameters  $(n, k, \delta)$  where  $n, k, \delta$  stand for code word length, number of information bits and minimum Hamming distance respectively. The code rate is defined by the ratio  $r = k/n$ . In our scheme, we use the same code to construct the product code  $P = C \otimes C$  which is obtained by :

- placing  $(k \times k)$  information bits in an array of  $k$  rows and  $k$  columns.
- coding the  $k$  rows using code  $C$ .
- coding the  $n$  columns using code  $C$ .

One interesting property of product codes is that all the rows and columns of the  $n \times n$  matrix are codewords of  $C$  [1]. The  $(n - k)$  last rows and last columns are both code words of  $C$  by construction. Furthermore the parameters of the resulting product code  $P$  are given by :

$$\begin{aligned} N &= n \times n \\ K &= k \times k \\ \Delta &= \delta \times \delta \end{aligned}$$

and the code rate  $R = r \times r$ .

At the moment, there is no general method to perform direct decoding of product codes decoding. A new iterative algorithm based on soft decoding and soft decision output which yields near optimum decoding of product codes in terms of Bit Error Rate (BER) was proposed by R. Pyndiah [1] in 1993. This decoding algorithm applies to any product code based on linear block codes. In section II of this paper we describe the soft decoding algorithm and give the soft decision algorithm derived from the theoretical Log-Likelihood-Ratio (LLR). Section III deals with the implementation of the

decoder while section IV gives the performance achieved by the DSP turbo decoder. In section V we give conclusions and perspectives.

## II. SOFT DECODING OF BLOCK CODES

If we consider the transmission of block coded binary symbols  $[-1, +1]$  using a QPSK modulation over a Gaussian channel, the sample vector  $R = (r_1, \dots, r_i, \dots, r_n)$  at the output of the coherent demodulator, conditionally to a transmitted code word  $E = (e_1, \dots, e_i, \dots, e_n)$  can be written as :

$$R = E + N \quad (1)$$

where  $N = (n_1, \dots, n_i, \dots, n_n)$  are Additive White Gaussian Noise (AWGN) samples of standard deviation  $\sigma$ . From the received samples  $R$ , the optimum decoded sequence is given by :

$$D = C^i \text{ if } \Pr\{E = C^i / R\} > \Pr\{E = C^j / R\} \quad \forall j \neq i \quad (2)$$

where  $C^i = (c_1^i, \dots, c_l^i, \dots, c_n^i)$  is the  $i^{th}$  code word of code  $C$  with parameters  $(n, k, \delta)$  and  $D = (d_1, \dots, d_l, \dots, d_n)$  is the decision corresponding to maximum likelihood transmitted sequence conditionally to  $R$ . For received samples corrupted by AWGN, decoding rule is simplified into :

$$D = C^i \text{ if } |R - C^i|^2 < |R - C^j|^2 \quad \forall j \neq i \quad (3)$$

$$\text{where : } |R - C^i|^2 = \sum_{l=1}^n (r_l - c_l^i)^2 \quad (4)$$

For block codes with high code rate  $R$ , the number of code words  $2^k$  is relatively large and optimum decoding requires important computation time, incompatible with high bit rate real time implementations. In 1972, Chase [2] proposed an algorithm to reduce the computation with small performance degradation. Instead of reviewing all the code words, the Chase algorithm searches for the code word at Hamming distance within a sphere of radius  $(\delta - 1)$ , centered on  $Y = (y_1, \dots, y_l, \dots, y_n)$  where  $y_l$  is equal to 1 if  $r_l > 0$  and to 0 if  $r_l \leq 0$ . To further reduce the number of reviewed code words, only the most probable code words within the sphere are selected by using channel information  $R$ . This search procedure can be decomposed in three steps :

step 1 : determine the position of the  $m$  least reliable binary symbols of  $Y$  using  $R$ . For real time constraints  $m$  is set to 3.

step 2 : form test patterns  $T^q$  for  $q = 0, \dots, 2^m - 1$  defined as all the combination of patterns with "0" or "1" in the  $m$  least reliable positions, thus in our application we considered 8 test patterns (Fig.2).

step 3 : decode  $Z^q = Y \oplus T^q$  using an algebraic decoder. It includes syndrome computation for the 8 test sequences. The algebraic decoder determines 8 code words  $C^q$ ,  $q = 0, \dots, 2^m - 1$ , one of them will be selected for soft decision.

To estimate the maximum likelihood transmitted sequence, the Chase algorithm applies decision rule (3) with the reviewed code words restricted to those found at step 3 above.

The Chase algorithm yields binary decision  $D$  for a given input  $R$ . The Log-Likelihood-Ratio of the decision  $d_j$ , noted  $(LLR)_j$ , is defined as :

$$(LLR)_j = \ln \left( \frac{\Pr\{e_j = +1 / R\}}{\Pr\{e_j = -1 / R\}} \right) \quad (5)$$

By considering the different authorized code words, the numerator of (5) can be expressed as :

$$\Pr\{e_j = +1 / R\} = \sum_{C^i \in S_j^{+1}} \Pr\{E = C^i / R\} \quad (6)$$

where  $S_j^{+1}$  is the set containing code words such that  $c_j^i = +1$  and the denominator of (5) can be expressed as :

$$\Pr\{e_j = -1 / R\} = \sum_{C^i \in S_j^{-1}} \Pr\{E = C^i / R\} \quad (7)$$

where  $S_j^{-1}$  is the set containing code words such that  $c_j^i = -1$ . By applying BAYES rule to (6) and (7) and assuming that the different code words are uniformly distributed, we obtain for the  $LLR$  of  $d_j$  defined by (5) the following relation :

$$(LLR)_j = \ln \left( \frac{\sum_{C^i \in S_j^{+1}} p\{R / E = C^i\}}{\sum_{C^i \in S_j^{-1}} p\{R / E = C^i\}} \right) \quad (8)$$

where the probability distribution function  $p\{\}$  above is given by :

$$p\{R / E = C^i\} = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp \left( -\frac{|R - C^i|^2}{2\sigma^2} \right) \quad (9)$$

By combining (8) and (9) we obtain :

$$(LLR)_j = \ln \left( \frac{\sum_{C^i \in S_j^{+1}} \exp \left( -\frac{|R - C^i|^2}{2\sigma^2} \right)}{\sum_{C^i \in S_j^{-1}} \exp \left( -\frac{|R - C^i|^2}{2\sigma^2} \right)} \right) \quad (10)$$

The computation of the  $LLR$  using (10) is too complex for block codes with a high code rate and of course for real time implementation. If we note  $C^{\min(+1)}$  the code word at minimum Euclidian distance from  $R$  with  $c_j^{\min(+1)} = +1$  and  $C^{\min(-1)}$  the code word at minimum Euclidian distance from  $R$  with  $c_j^{\min(-1)} = -1$ , the  $LLR$  can be approximated by :

$$(LLR)_j = \frac{1}{\sigma^2} \left( |R - C^{\min(-1)}|^2 - |R - C^{\min(+1)}|^2 \right) \quad (11)$$

By expanding relation (11) using (4) we obtain :

$$(LLR)_j = \frac{2}{\sigma^2} \left( r_j + \sum_{l=1, l \neq j}^n \eta c_l^{\min(+1)} p_l \right) \quad (12)$$

where :

$$p_l = \begin{cases} 0 & \text{if } c_l^{\min(+1)} = c_l^{\min(-1)} \\ 1 & \text{if } c_l^{\min(+1)} \neq c_l^{\min(-1)} \end{cases} \quad (13)$$

and  $c_l^{\min(+1)}$  is the  $l^{th}$  symbol belonging to code word  $C^{\min(+1)}$ . If we normalize the approximated  $LLR$  of  $d_j$  with respect to  $2/\sigma^2$  we obtain :

$$r_j' = r_j + w_j \quad (14)$$

$$\text{where : } w_j = \sum_{l=1, l \neq j}^n \eta c_l^{\min(+1)} p_l \quad (15)$$

The estimated normalized *LLR* of decision  $d_j$ ,  $r'_j$  is given by  $r_j$  plus  $w_j$  which is a function of the code words at minimum Euclidian distance from  $R$  and  $\{\eta\}$  with  $l \neq j$ . The *LLR*  $r'_j$  is an estimation of the soft decision of the block decoder. It has the same sign as  $d_j$  and its absolute value indicates the reliability of the decision.

To compute the normalized *LLR* we suppose that we have identified the two code words  $C^{\min(+1)}$  and  $C^{\min(-1)}$ . For each code word  $C^q$  found at step 3, we compute its Euclidian distance from  $R$  :

$$M^q = |R - C^q|^2 \quad (16)$$

As in the Chase algorithm we select the code word  $C^d$  at minimum Euclidian distance from  $R$ . Then we search for the code word  $C^c$  at minimum Euclidian distance from  $R$  such that  $c_j^c \neq c_j^d$ . If we find such a code word then we compute the soft decision for each bit  $d_j$  of the decoded sequence using the relation given below :

$$r'_j = \left( \frac{M^c - M^d}{4} \right) c_j^d \quad (17)$$

else we use the relation :

$$r'_j = \beta \times c_j^d \quad (18)$$

where  $\beta$  is a constant which is a function of the iteration number.

### III REAL TIME IMPLEMENTATION

Implementation of such a decoder on a DSP chip does not present any particular difficulty from a theoretical point of view. On the other hand if we wish to achieve high data rates we have to generate a code with minimum computation redundancy.

For an input block of 1024 samples, each sample is coded on 8 bits. The turbo code decoder has to perform then four main tasks :

- Compute the vector  $Y$  using the received samples  $R$ . The  $m$  least reliable binary symbols are computed at the same time and ordered according to the  $Y$  vector bit number.  $Y$  parity is easily checked at the same time.

- Compute syndromes of vectors  $Z^q$ , this is performed using an algebraic decoder. This decoder uses 1024 values precomputed and stored in a transcoding table. According to the different bit values of the vectors  $Z^q$  and using code properties, the decoder checks the wrong bit position if the

input vector is not a code word. After eventual correction the word  $C^q$  belongs to the set of code words.

- Compute the Euclidian distances between the code words  $C^q$  and the sample vector  $R$ . In fact to reduce computation redundancy, the procedure only computes Euclidian distances between different code words  $C^q$ . The code word with the minimum value becomes the selected code word and the next nearest one will become the concurrent code word. In this implementation, we only consider one concurrent code word to simplify the weighting algorithm.

- compute the soft decision for each bit  $d_j$

#### III.1 Syndrome computation

The algebraic decoder uses the transcoding table  $SYND[S][j]$  to check the eventual wrong bit. For the BCH(32,26,4) code, the initial value of  $S$  is set to the value 31. The words  $Z^q$  are 31 bits long,  $z_j^q$  is the value of the  $j^{th}$  bit of the word  $Z^q$ . Let us consider the wrong bit checking procedure.

- For  $j = 0, \dots, 30$

if  $z_j^q = 1$  then  $S = SYND[S][j]$

The last value of  $S$  indicates the wrong bit position if  $S < 31$ . Thus the procedure only moves through the table according to the different bits of the word. The procedure has to be repeated for each word  $Z^q$ ,  $q = 0, \dots, 2^m - 1$ . By construction, word  $Z^q$  has only a few different bits corresponding to  $m$  least reliable bits. If the position of bits are  $(I_0, I_1, I_2)$  (fig. 1), an efficient and fast method can be used to compute in a single loop the  $(2^m - 1)$  syndromes.

		↓ $I_0$	↓ $I_1$	↓ $I_2$
$Y$				
$T_0$	0	0	0	
	0	0	1	
	0	1	0	
$T_q$	0	1	1	
	1	0	0	
	1	0	1	
	1	1	0	
$T_7$	1	1	1	

Fig. 1 : Creation of  $Z^q = Y \oplus T^q$

The new procedure, implemented for real time processing, can be resumed as follows ( $y_j$  is the value of the  $j^{th}$  bit of the word  $Y$ ) :

- For  $j=0, \dots, I_0 - 1$



if  $y_j = 1$  then  $S = \text{SYND}[S][j]$

of course, this value  $S$  is the same for the all the words  $Z^q$

-  $j = I_0$

if  $y_j = 1$  then  $S_0 = \text{SYND}[S][j]$

$S_1 = S$

else  $S_0 = S$

$S_1 = \text{SYND}[S][j]$

$S_0$  is the updated value of the set  $\{Z^q\}_{q=0,\dots,3}$  and  $S_1$  the

updated value of the set  $\{Z^q\}_{q=4,\dots,7}$ .

-  $I_0 < j < I_1$

if  $y_j = 1$  then  $S_0 = \text{SYND}[S_0][j]$

$S_1 = \text{SYND}[S_1][j]$

At the bit position  $I_1$  two nodes are added to the tree allowing computation of  $S_{00}, S_{01}, S_{10}, S_{11}$  which are updated according to  $y_j$  value until the bit position  $I_2$ . Then four new nodes are added to the tree to finish at the bit number 30 with the 8 syndrome values :

$S_{000}, S_{001}, S_{010}, S_{011}, S_{100}, S_{101}, S_{110}, S_{111}$

The set  $\{S_{lmn}\}_{l,m,n \in [0,1]}$  corresponds to the wrong bit positions into the code words  $\{C^q\}_{q=0,\dots,7}$ , if  $S_{lmn} < 31$ .

### III.2 Euclidian distance computations

Prior to updating the bit reliability in the code word  $C^d$ , the decoder needs to compute the extrinsic information resulting from decoding the input information  $r_j$ .

The Chase algorithm selects 8 code words  $C^q$  in our case. The distance between the sample vector  $R$  and a code word  $C^q$  can be expressed from (4) as following :

$$M^q = |R - C^q|^2 = \sum_{l=1}^n \left( r_l^2 + (c_l^q)^2 \right) - 2 \sum_{l=1}^n r_l c_l^q \quad (19)$$

For all the code words  $C^q$ , the first term has the same value. The code word  $C^d$  at minimum distance from  $R$  also maximizes the correlation term  $\sum_{l=1}^n r_l c_l^q$ . The code words

$\{C^q\}$  only have a few bits different. If we consider the set of bits  $B$ , including the  $m$  least reliable bits and the set of bits corresponding to the syndromes:

$$B = \left\{ I_0, I_1, I_2, \{S_{lmn}\}_{l,m,n \in [0,1]} \right\}$$

the following term  $\sum_{l=1, l \notin B}^n r_l c_l^q$  takes the same value for each

code word of the set  $\{C^q\}$ . So, in a real time environment it has not to be computed. The real time algorithm performs the following steps :

- using syndrome computation strategy (III.1), we begin by computing contributions of the  $m$  least reliable bits to the correlation for each code word according to  $y_j$  value (addition or subtraction).

- for each code word we compute the contribution corresponding to the syndrome. To avoid each code word sum updating, it is enough to multiply the contribution by two before adding or subtracting according to  $y_j$  value.

The code word at minimum Euclidian distance from  $R$  is then chosen as  $C^d$  and we also look for one concurrent code word  $C^c$  in our case to update the soft decision of each bit.

### III.3 Soft decision updating

Once the code word  $C^d$  found, the decoder looks for one concurrent code word  $C^c$ . From relation (17) the value  $r_j$  can be expressed as following :

$$r_j = 0.5 \left( \sum_{l=1}^n r_l c_l^d - \sum_{l=1}^n r_l c_l^c \right) c_j^d \quad (20)$$

then for all bits such as  $c_j^d \neq c_j^c$  the extrinsic information for bit  $d_j$  takes the following value :

$$w_j = 0.5 \left( \sum_{l=1}^n r_l c_l^d - \sum_{l=1}^n r_l c_l^c \right) c_j^d - r_j \quad (21)$$

otherwise, for  $c_j^d = c_j^c$ ,

$$w_j = \beta c_j^d \quad (22)$$

where  $\beta$  is a constant which depends on the iteration number.

On receiving the matrix  $[R]$  corresponding to a transmitted code word  $[E]$  of the product code, the decoder decodes the rows (or columns) of the matrix, estimates the LLR and gives the output  $[W]$ . Then the decoder performs the same operations on the columns (or rows) using as input :

$$[R''] = [R] + \alpha [W] \quad (23)$$

where  $\alpha$  is used to reduce the influence of  $[W]$  in the first iterations where the BER is relatively high and thus  $[W]$  is not

absolutely reliable. The decoding procedure described above is then iterated.

#### IV PERFORMANCES OF DSP DECODER

The performances of the algorithm have been evaluated using Monte Carlo simulations for different signal to noise ratio ( $E_b/N_0$ ), the BER was computed after each iteration. The number of iterations was set to eight as said previously.

To estimate the DSP implementation performances, the c language decoder was adapted to DSP constraints :

- input samples quantized on 8 signed bits,
- simplified decoding algorithm.

Figure 3 presents the results of BER versus Signal to Noise Ratio (SNR) after each iterations. The coding gain at a BER of  $10^{-5}$  exceeds 6 dB after 4 iterations

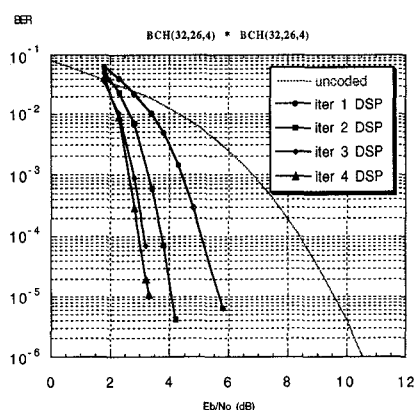


Fig. 3 : Bit Error Rate versus SNR after each of the first four iterations

Figure 4 only shows performance degradation when we take into account DSP constraints: less than 0.5 dB at a BER of  $10^{-5}$  after 4 iterations

The decoder is implemented on a MOTOROLA DSP56002 (clock frequency 40 Mhz). To reach acceptable performances in term of bit rate, the DSP version of the decoder is fully optimized for real time applications.

To improve decoder capacities, we chose a bit processing strategy. Coder (Decoder) parameters can easily be modified. The data rate mainly depends on the number of iterations and of course DSP computation capacity. Our first implementation reaches 160 kbits/s for half an iteration. The decoder requires 1300 words of program memory and 4300 words of data memory.

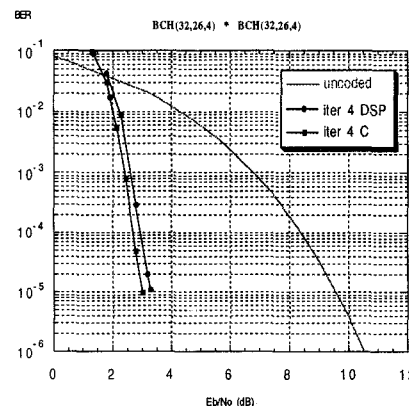


Fig. 4 : Comparison between C version and 'DSP' version

#### V CONCLUSIONS

In this paper we have presented an optimized implementation of block turbo decoding for product codes based on BCH(32,26,4) block codes. The 'DSP' version decoder achieves performances close to those of the near optimum decoder [1]. Fully optimized for real time implementation, the coder/decoder parameters can be easily modified to suit different applications. The bit rate mainly depends on iteration number and of course on DSP computation capacity. With the fast improvement in computation capacity (1600 Mips on a Texas Instrument DSP chip), the implementation of Block Turbo codes becomes a realistic alternative to VLSI circuit solutions for many applications.

#### REFERENCES

- [1] R. PYNDIAH, A. GLAVIEUX, A. PICART and S. JACQ, "Near optimum product codes", Globecom'94, San Francisco, Nov., Dec. 1994, pp 339-343.
- [2] D. CHASE, "A class of algorithms for decoding block codes with channel measurements information", IEEE Trans. Inform. Theory, Vol IT-18, Jan. 1972, pp170-182.
- [3] S. JACQ, "Décodage itératif des codes produits : "Turbo-codes en bloc", et évaluation de leurs performances pour des modulations MDP et MAQ sur canal de Gauss et de Rayleigh", thesis Limoges University, May 1996

# AN INTRINSICALLY RELIABLE AND FAST ALGORITHM TO COMPUTE THE LINE SPECTRUM PAIRS (LSP) IN LOW BIT RATE CELP CODING

A. GOALIC and S. SAOUDI

Department of Signals and Communications  
ENST-Br BP 832 -29285 Brest -BRITTANY, FRANCE  
Andre.Goalic@ENST-BRETAGNE.fr

## ABSTRACT

The Code Excited Linear Prediction Coder (CELP) makes it possible to synthesize good quality speech at low bit rates. In such a case, speech quality mainly depends on spectral envelope design accuracy. Different kinds of parameters belonging to the parametrical domain (linear prediction coefficients  $\{a_i\}$ -LPC - parameters), to the time domain (PARTIAL CORrelation - PARCOR - parameters) and to the frequency domain (Line Spectrum Pairs - LSP -parameters) are used to design the vocal tract transfer function. The latter present interesting properties both for quantization and interpolation, confirming their increasing utilization in low bit rate speech coding. In real time processing, efficient methods must be used to compute this set of parameters. The purpose of this paper is to show attractive properties in real time processing for Line Spectrum Pair computing (LSP). The possibility of computing each LSP parameter independently characterizes the intrinsic reliability of this method based on the use of the Split Levinson Algorithm. This method is compared to the one using the Chebyshev polynomials.

## 1. LINE SPECTRUM PAIRS

In the speech coding domain, a human vocal tract function transfer can be designed using different sets of parameters. LPC parameters are too dynamic to be used efficiently in speech coding. The bounded property of PARCOR parameters make them very interesting in the coding process; unfortunately they lead to spectral impairment as soon as interpolation is required, for example at very low bit rates. LSPs seem to be well-suited parameters at low and very low bit rate speech coding, where they are now widely used. Belonging to the frequency domain, they require complicated algorithms in order to be obtained. Let us recall the computing process.

The human vocal tract  $p$ th-order inverse transfer function can be expressed using LPC parameters  $\{a_i\}$ :

$$A_p(z) = 1 + \sum_{i=1}^p a_i z^{-i}$$

$A_p(z)$  may be decomposed into a set of two inverse transfer functions, corresponding respectively to complete opening

$[P(z)]$  and complete closing  $[P^*(z)]$  of the glottis. The inverse transfer function can be expressed as :

$$A_p(z) = [P(z) + P^*(z)]/2$$

The two inverse transfer functions roots directly give the Line Spectrum Pairs  $w_i$  (for an even value of  $p$ ) :

$$P^*(z) = (1 - z^{-1}) \prod_{i \text{ even}} (1 - 2\cos w_i z^{-1} + z^{-2})$$

$$P(z) = (1 + z^{-1}) \prod_{i \text{ odd}} (1 - 2\cos w_i z^{-1} + z^{-2})$$

with the ordering properties :

$$0 < w_1 < w_2 < \dots < w_p < \pi$$

The LSPs parameter computing process begins with :

- Hamming time filtering window.
- Autocorrelation function computation of the first  $p+1$  values.

According to the method used, the computing process follows different steps :

- the first method (section 2), widely used in coders and vocoders, first computes the Linear Prediction Coefficients using the Levinson Durbin algorithm, and secondly, after developing  $P(z)$  and  $P^*(z)$  in cosine functions, uses CHEBYSHEV polynomials to make possible real-time processing. Finally a dichotomy method is used to compute  $P(z)$  and  $P^*(z)$  roots - KABAL et al. method - [1].

- the second (section 3), recently introduced, uses the Split Levinson algorithm to compute only one inverse transfer function coefficient. The recurrence relation between different transfer function orders makes it possible to design two tridiagonal matrices, each eigenvalue of which depends only on an LSP function. Matrice elements property allows the utilization of the bisection method to compute rapidly the LSPs - SAOUDI et al. method - [2]. With the same initial conditions and computation accuracy, the two methods give the same results. The 10<sup>th</sup> prediction order is usually used in low bit rate speech coding ( $p=10$ ).

Finally, before concluding, section 4 compares the two methods and give results achieved in real time implementation.

## 2. KABAL et al. method

This is probably the most widely used method of computing LSP parameters. It includes the following steps:

### 2.1 LPC parameter Computation

Linear Prediction Coefficients  $\{a_i\}$  are computed using the Levinson Durbin algorithm.

### 2.2 The development of a transfer function cosine

Expressed as a LPC parameter function, each transfer function is as follows :

$$P(z) = a_0(z^0 + z^{-(p+1)}) + \sum_{i=1}^p (a_i + a_{p+1-i})z^{-(p+1-i)}$$

$$P^*(z) = a_0(z^0 - z^{-(p+1)}) + \sum_{i=1}^p (a_i - a_{p+1-i})z^{-(p+1-i)}$$

One root of each transfer function is known, so they can be expressed as :

$$P(z) = (1 + z^{-1}) \sum_{k=0}^p l_k z^{-k} = (1 + z^{-1})Q(z)$$

$$P^*(z) = (1 - z^{-1}) \sum_{k=0}^p l_k^* z^{-k} = (1 - z^{-1})Q^*(z)$$

Relations between parameters  $l_k$ ,  $l_k^*$  and the property of the located roots on the unit circle leads to  $Q(z)$  and  $Q^*(z)$  cosine development as follows :

$$Q(w) = e^{-j\frac{pw}{2}} \left( \sum_{k=0}^{\frac{p-1}{2}} 2l_k \cos\left(\frac{p-k}{2}w\right) + l_{\frac{p}{2}} \right) = Q(z)|_{z=e^{jw}}$$

$$Q^*(w) = e^{-j\frac{pw}{2}} \left( \sum_{k=0}^{\frac{p-1}{2}} 2l_k^* \cos\left(\frac{p-k}{2}w\right) + l_{\frac{p}{2}}^* \right) = Q^*(z)|_{z=e^{jw}}$$

Finally,  $R(w)$  and  $R^*(w)$  polynomial roots give the LSPs :

$$R(w) = \sum_{k=0}^{\frac{p-1}{2}} 2l_k \cos\left(\frac{p-k}{2}w\right) + l_{\frac{p}{2}}$$

$$R^*(w) = \sum_{k=0}^{\frac{p-1}{2}} 2l_k^* \cos\left(\frac{p-k}{2}w\right) + l_{\frac{p}{2}}^*$$

Requiring trigonometric function evaluation, this method seems to be difficult and time consuming in real time processing. To solve this problem, Kabal and Ramachandran use the Chebyshev polynomials.

### 2.3 Chebyshev Polynomials

Chebyshev polynomials satisfy the following recursion order :

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

where  $T_k(x) = \cos(kw)$  is a  $k^{\text{th}}$  -order Chebyshev polynomial in  $x$  ( $x = \cos(w)$ ). The two previous polynomials can now be expressed in terms of Chebyshev polynomials :

$$R(w) = \sum_{k=0}^{\frac{p}{2}} c_k T_k(x)$$

$$R^*(w) = \sum_{k=0}^{\frac{p}{2}} c_k^* T_k(x)$$

$c_k$ ,  $c_k^*$  are expressed using LPC parameters  $\{a_i\}$ . The following recurrence relationship makes it possible to compute polynomials without evaluating the cosine function :

$$b_k(x) = 2xb_{k+1}(x) - b_{k+2}(x) + c_k$$

with  $b_6(x) = b_7(x) = 0$ . This recursion is used to calculate  $b_0(x)$  and  $b_2(x)$  before computing the two following polynomials :

$$R(w) = \sum_{k=0}^{\frac{p}{2}} (b_k(x) - 2xb_{k+1}(x) + b_{k+2}(x))T_k(x)$$

$$= \frac{b_0(x) - b_2(x) + c_0}{2}$$

$$R^*(w) = \sum_{k=0}^{\frac{p}{2}} (b_k^*(x) - 2xb_{k+1}^*(x) + b_{k+2}^*(x))T_k(x)$$

$$= \frac{b_0^*(x) - b_2^*(x) + c_0^*}{2}$$

## 2.4 Line Spectrum Pair computation

The last steps of this method include  $R(w)$  and  $R^*(w)$  root evaluation. To understand the computing process well, let us recall that  $R(w)$  and  $R^*(w)$  roots interlace on the unit circle. The dichotomy method used to compute the LSPs has to take this fact into account. As roots lie in the  $[-1, +1]$  range, the dichotomy method is applied between these two limits. The process begins to evaluate  $R(w)$  the first root of the polynomial and the first root search for  $R^*(w)$  polynomial starts from the position of the  $R(w)$  root just found. The process continues as before but interchanges polynomial roles as each root is found. To be successfully performed and not to miss any roots, the dichotomy method has to take into account :

- the minimal distance between two roots of the same polynomial.
- the minimal distance between pairs of roots (one belonging to  $R(w)$ , the other to  $R^*(w)$ ). This is to avoid missing a root when switching the search from one polynomial to the other. Let the roots be denoted by  $\{r_i\}$  and ordered  $r_i > r_{i-1}$ . According to their parity they belong either to  $R(w)$  or  $R^*(w)$ .

To make possible real time processing on existing processors two iterative steps ( $\Delta, \delta$ ) must be taken :

- the first must be smaller than the minimal distance between two roots of the same polynomial ( $\Delta < (\min(r_{i+2} - r_i)) ; i = 0, 1$ ).
- the second must be smaller than the minimum spacing between pairs of roots ( $\delta < (\min(r_{i+1} - r_i)) ; i = 0, 1$ ). Both  $\Delta$  and  $\delta$  depend on the properties of the speech database statistics.

## 3. SAOUDI et al. method

The classical Levinson Durbin algorithm is known to be redundant in complexity[3]. It can be split into two parts (symmetric and antisymmetric forms), and only one of them has to be computed to evaluate the human vocal tract inverse transfer function. The two inverse transfer functions previously defined, can be expressed as :

$$P(z) = \sum_{i=0}^{P+1} p_{p+1,i} z^{-i}$$

$$P^*(z) = \sum_{i=0}^{P+1} p_{p+1,i}^* z^{-i}$$

Both are computed recursively using the Split Levinson algorithm symmetric and antisymmetric form. The two algorithms also compute the reflection coefficients if required.

## 3.1 Recurrence relations

The two forms make it possible to establish a three term recurrence relation between different orders of the inverse transfer function of course for each one :

$$P_{k+1}(z) - (1 + z^{-1})P_k(z) + \alpha_k P_{k-1}(z) = 0$$

$$P_{k+1}^*(z) - (1 + z^{-1})P_k^*(z) + \alpha_k^* P_{k-1}^*(z) = 0$$

The two set of inner parameters are evaluated while computing inverse transfer functions. Using a single change of variable, two real polynomials linked by a three-term recurrence relation, can be designated :

$$H_{k+2}(x) - (x - \alpha_k - \alpha_{k+1} + 2)H_k(x) + \alpha_{k-1}\alpha_k H_{k-2}(x) = 0$$

$$H_{k+2}^*(x) - (x - \alpha_k^* - \alpha_{k+1}^* + 2)H_k^*(x) + \alpha_{k-1}^* \alpha_k^* H_{k-2}^*(x) = 0$$

These relations lead to two tridiagonal matrices, each eigenvalue of which depends only on an LSP function :

$$\begin{bmatrix} 2\alpha_1 + \alpha_2 - 2 & 1 & 0 \\ \alpha_2 \alpha_3 & \alpha_3 + \alpha_4 - 2 & 1 \\ 0 & & \\ & & 0 & \alpha_{p-2} \alpha_{p-1} & \alpha_{p-1} + \alpha_p - 2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_2^* - 2 & 1 & 0 \\ \alpha_2^* \alpha_3^* & \alpha_3^* + \alpha_4^* - 2 & 1 \\ 0 & & \\ & & 0 & \alpha_{p-2}^* \alpha_{p-1}^* & \alpha_{p-1}^* + \alpha_p^* - 2 \end{bmatrix}$$

## 3.2 LSP computation

The eigenvalues of the two matrices only depend on the values of the set of parameters  $\alpha$  and  $\alpha^*$ . Each Split Levinson algorithm form can compute reflection coefficients, creating an easy link between the two sets of parameters. Thus only one Split Levinson algorithm makes it to compute possible  $\alpha$  and  $\alpha^*$ . Therefore the antisymmetric Split Levinson algorithm form (the symmetric form includes a larger number of instructions) is the fastest way to compute the two sets of parameters leading to the Line Spectrum Pairs.

## 3.3 Bisection method

Once the problem has been posed in matricial form, fast and powerful methods can be used to compute eigenvalues. Responding to utilization conditions, the fastest eigenvalue

computation method - the bisection method - is successfully used here to compute the Line Spectrum Pairs. Indeed matrix coefficients  $\alpha$  and  $\alpha^*$  respond to the following relation :

$$0 < \alpha, \alpha^* < 4$$

Let us recall the main advantages of the bisection method for eigenvalue computation, which makes it possible :

- to know the number of eigenvalues included in an interval, and using an iterative processing to isolate the  $i^{\text{th}}$  value independently of the others.
- to compute the  $i^{\text{th}}$  eigenvalue with the required accuracy in this interval.

#### 4. COMPARISON WITH REAL TIME PROCESSING

##### 4.1 Intrinsic reliability

The main advantage of the SAOUDI et al. method is its intrinsic reliability. Due to use of the matrix formulation and bisection method, this method evaluate each LSP parameters independently without taking into account any distances between consecutive LSPs. On the contrary KABAL et al. method must take into account these distances not to miss any roots. To avoid possible divergences, tests even if time consuming, must be implemented.

##### 4.2 Signal Statistics Independence

This features comes directly from the first point, in order not to miss roots. As far as the distance between consecutive roots has to be taken into account, the method remains linked with the speech database. This is the case for the KABAL et al. method.

##### 4.3 Computation speed

Table 1 shows the computational requirements measured on the test bench. Both coder and decoder use only a single fixed point Digital Signal Processor MOTOROLA 56001 (clock frequency : 27 MHz) [4]. The number in brackets represents whole coder computational requirements in terms of percentage. To avoid divergences, the dichotomy method often has to compute the polynomials though computational time is drastically reduced by using two incremental steps. Accuracy in LSP determination varies from 5 Hz in the middle frequencies to 20 Hz at low and high frequencies. Our process only computes the Line Spectrum Pair cosine function (CLSP) because the decoder only uses the CLSP.

##### 4.4 Digital acoustic underwater phone

The new CLSP computing process is included in a new digital acoustic underwater phone we are designing at our institute. The coder bit rate is presently 5.45 Kbit/s and the

whole system including QPSK modulation has been tested successfully both in a testpool and in BREST bay using acoustic waves.

Algorithm	LPC analysis M.I.P.S.	CLSP extraction only (M.I.P.S.)
SAOUDI et al.	1.43 (10.6%)	1.1 (8.1%)
KABAL et al.	2.1 (15.6%)	1.77 (13.1%)

Table 1 : Computational requirements

#### 5. CONCLUSION

This paper has compared two algorithms for computing the Line Spectrum Pairs parameters in a real time environment. Without increasing the complexity, the new one based on the Split Levinson algorithm and the bisection method shows interesting properties and appears to be well-suited for real time applications : intrinsic reliability, independence from the signal statistics and fast computation.

#### ACKNOWLEDGEMENTS

The authors thank Janet OMROD for her kind assistance with the english of this paper.

#### MAIN REFERENCES

- [1] P. Kabal and R.P. Ramachandran, "The computation of line Spectral Frequencies using Chebyshev polynomials", Vol. ASSP-34, N° 6, pp. 1419-1425, December 1986.
- [2] S. Saoudi, J.M. Boucher and A. Le Guyader, "A new efficient algorithm to compute the LSP parameters for speech coding", Signal Processing, Vol. 28, No 2, pp.201-212, August 1992.
- [3] P. Delsarte and Y. Genin, "The split Levinson algorithm", Proc. ICASSP'86, June 1986, pp. 470-478.
- [4] A. Goalic "Etude et réalisation de Codeur/décodeur de parole à bas débit pour la téléphonie acoustique sous-marine". Thèse de doctorat, Université de Bretagne Occidentale BREST, Juin 1994 (Study and development of a low bit rate speech coder/decoder for an acoustic underwater phone)(Doctoral thesis...).

# TOWARD A DIGITAL ACOUSTIC UNDERWATER PHONE

A. GOALIC, J. LABAT, J. TRUBUIL, S. SAOUDI, D. RIOUALEN

Département Signal et Communications

ENST-Br BP 832

29285 BREST FRANCE

Tel : 98 00 13 65 Fax : 98 00 10 12

**Abstract** - This paper deals with the design of a digital acoustic underwater phone prototype. Digital techniques allow to achieve a synthesized speech with a quality close to the telephonic one. The input speech signal is compressed down to 5.45 kbit/s using a CELP coder. The bit rate is 6 kbit/s before channel coding (including frame synchronization's word) and expected to be about 8 kbit/s after channel coding. A QPSK modulation with differential encoding was chosen to transmit the useful signal. For the receiver we used a scheme where synchronization and equalization (FSE+DFE) were jointly optimized. The whole system (unidirectional link) has been implemented on single DSPs (Motorola 56001) and tested successfully in a very difficult environment (IFREMER pool).

## I. INTRODUCTION

The purpose of this paper is to describe the real-time processing of a low bit rate unidirectional digital underwater phone on a single fixed point DSP MOTOROLA 56001 (fig.1). Both transmitter and receiver are made up of two main parts. Into the transmitter, the input speech is compressed by the source coder while the compressed bit stream is sent to the receiver on an acoustic carrier wave, using QPSK signaling. In the receiver, synchronization and equalization are both achieved according to Falconer's scheme [1], while speech synthesis is performed by the decoder.

The paper is organized as follows : section II presents the transmitter which includes the Code Excited Linear Prediction (CELP) coder and the modulator. Able to produce high quality speech at low bit rate, the CELP coder scheme was chosen as speech coding algorithm. The speech signal is sampled at 8 kHz, on a 16 bit ADC (128 kbit/s)

and allows the determination of the 23 parameters per speech frame (20 ms).

Once computed, these parameters are sent to the receiver using a QPSK modulation. Section III presents the receiver which includes the demodulator and the speech decoder. After synchronization and equalization, the receiver gets back the speech coding parameters which are fed to the decoder to synthesize the speech signal. Section IV presents the real time processing of the unidirectional digital phone running on the test bench. This section also presents the real time processing of the receiver tested in the IFREMER pool. Section V presents the system performances in term of speech coding and modulation. Section VI contains conclusions and perspectives.

## II. TRANSMITTER

The input is segmented into frames having a length of 20 ms. Therefore all the computing process must be achieved in less than 20 ms. It includes the 3 following tasks :

- Acquisition task : Using a fast interrupt, the input speech is sampled at 8 kHz and digitized by an Analog to Digital Converter (ADC).

- Background job : the input stream (160 samples coded on 16 bits : 2560 bits per frame ) is compressed to design the 23 coder parameters (109 bits) constituting the frame code. As a consequence, we obtain a data compression ratio equal to 23.5. It is the main part of the transmitter performed by the source coder. The speech compression rate involve a high degree of computational complexity, particularly on a fixed point digital signal processor.

- Transmission task : Using a long interrupt, the compressed bit stream is sent to the decoder via the Serial

Communication Interface (SCI) and the data transmission line. This task is performed by the modulator.

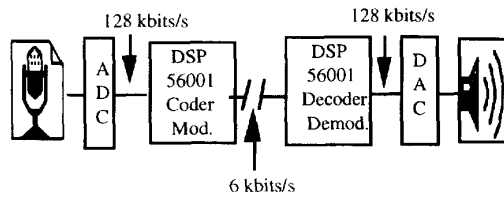


Figure 1 Phone prototype

## II.1 ACQUISITION TASK

The input speech, recorded on a Digital Audio Tape, is sampled at 8 kHz on a 16-bit ADC. The Synchronous Serial Interface (SSI) is the communication link between the converter and the DSP. Each sample is read and stored in the circular buffer by means of one fast interrupt.

## II.2 BACKGROUND JOB

The background job is the coding task because of its large computational complexity. It includes three parts (fig.2):

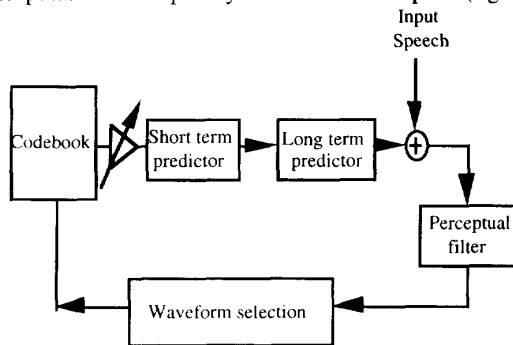


Figure 2 : TCELP coder

### II.2.1 Short term predictor : vocal tract model

An all pole time varying filter models the short-time spectral envelope of the speech waveform. It is often called short term predictor filter because its coefficients are computed by predicting a speech sample from few previous samples (10 in our coder). Different sets of parameters can be used to design the vocal tract, however the Line Spectrum Pairs (LSP) representation has shown its ability to synthesize good quality speech at low bit rate.

The new algorithm [3], only using the anti-symmetric version of the split LEVINSON algorithm, with its matrix representation, leads to an efficient method with an

intrinsic reliability particularly well-suited for real-time processing. Determination accuracy depends on the frequency : from 5 Hz (middle frequencies) to 20 Hz (low and high frequencies).

Parameters	Block(ms)	Bits/frame	Rate(bits/s)
CLSP(10)	20	30 x 1	1500
Glpc	20	7 x 1	350
B ltp	10	3 x 2	300
Pitch	10	7 x 2	700
Gq	5	3 x 4	600
I index	5	10 x 4	2000
	TOTAL	109	5450

Table 1 : Bit allocation

### II.2.2 Long term predictor : voice source model

It is designed to remove periodic speech signal redundancy. According to the voiced or unvoiced nature of the signal, a speech sample can be predicted from the linear combination of the past version signal. To decrease the computational load, a single coefficient predictor and a simplified method, called "Correlation-peak-picking", are used to compute pitch and voice gain.

### II.2.3 Waveform selection

The last part of the Code Excited Linear Prediction involves a huge amount of computations. The synthesized analysis consists in finding a waveform in the codebook which is optimum with respect to a subjective criterion. Each waveform, scaled by an optimal gain factor is processed through the short and long term inverse filter to produce synthesized speech. Then the difference between original and synthesized speech is processed through the perceptual weighting filter. The best waveform is chosen to minimize perceptual error signal energy. The large amount of computations mainly comes from the optimum waveform search procedure and particularly from all waveform filtering through short and long term predictors. Moreover the excitation waveform is renewed four times a frame, that is to say every 5 ms. To allow real time implementation and decrease the computational time, a modified structure for seeking the optimal excitation waveform was implemented and a ternary codebook used. Each sample, belonging to the waveform, takes its value in the set : [-1,0,+1]. The modified structure consists of splitting into two parts the waveform seeking procedure. The first, memoryless one, is performed once a frame



(every 20 ms), the second one must be performed four times a frame (every 5 ms). It includes the recent past process. The ternary codebook utilisation is interesting. It decreases the number of arithmetical operations, simplifies the filtering process and avoids the memory storage. The codebook contains 1024 waveforms, thus each one is coded on 10 bits (table 1).

### II.3 TRANSMISSION TASK

For this purpose we have chosen a QPSK modulation. As a matter of fact, in underwater communications, two main problems are encountered. The first one is the multipath propagation involved by multiple scattering on surface and bottom. The second one is the Doppler effect arising from relative motion between the transmitter and the receiver. One way to efficiently combat these effects consists in using a coherent receiver where equalization and synchronization are jointly optimized according to the LMS criterion [1], [2].

Moreover, in order to solve the phase ambiguity problem, the data were differentially encoded. All these functions are achieved by a DSP in conjunction with a Numerically Controlled Oscillator (NCO). The carrier frequency was set to 60 kHz and the bit rate to 6 kbit/s, including the synchronisation word. In addition to this the data were first scrambled to avoid synchronization loss (timing recovery) in the receiver.

## III. RECEIVER

In the transmitter, the biggest computational time is taken by the CELP coder to process the input speech. On the contrary, in the receiver, the demodulation operations perform the biggest one. It includes two others tasks named as following :

- Reception task : The received signal has first to be filtered and correctly amplified. Then it is demodulated using a local oscillator and the in-phase and quadrature signals are both sampled using two ADCs. Then all the receiver functions, that is to say adaptive equalization, timing and carrier recovery as well as differential decoding and descrambling, are both processed by the DSP.

- Background job : with the frame code, the 23 parameters are restored, thus allowing filter design and speech synthesisation.

- Speaking task : using fast interrupts, the speech samples are written via the SSI interface on the Digital-Analog-Converter (DAC), at 8 kHz, to drive the speakers.

### III.1 Reception task

The receiver has mainly to achieve timing and carrier recovery as well as adaptive equalization. For this purpose we have designed a receiver according to a scheme where equalization and carrier recovery were jointly optimized. In our receiver timing recovery is also optimized with respect to the LMS criterion, using the so-called Stochastic Gradient Least Mean Square algorithm (SGLMS). The adaptive equalizer includes a forward fractionally spaced filter (Fractionnally Spaced Equalizer) coupled with a feedback filter which is fed with previous decisions (Decision Feedback Equalizer). A fractionally spaced equalizer proves to be partly insensitive to a sampling phase error whereas the DFE is chosen to strongly combat the intersymbol interference due to multipath propagation.

In such an environment (IFREMER pool), the impulse response spans over tens of symbol. So the equalizer needs a large number of taps preventing the use of the RLS algorithm because of its too large computational complexity. Figure 3 shows us results about one experiment where distance between the emitter and receiver was about 40 meters. On the left plot, we can observe the complex baseband received signal represented in XY mode while on the right plot we can see the eye patterns of both in-phase and quadrature signals.

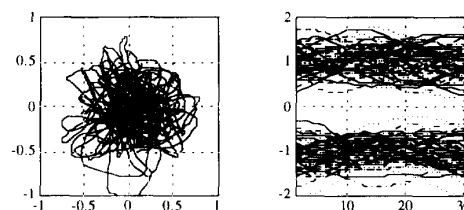


Figure 3 Analog input signal and eye patterns

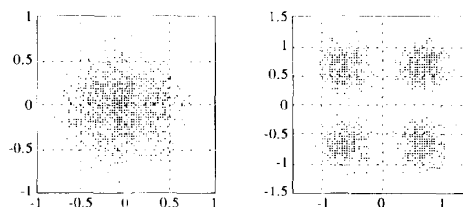


Figure 4 Equalizer input and output

On figure 4, we can see the forward equalizer input and the decision device input, showing that an important part of ISI has been removed with this strategy. The number of taps was 67, 32 for the forward equalizer and 35 for the feedback one. All the results are presented using MATLAB graphs instead of oscillographs in order to get better plots. The files were created by the processor Development System and then converted into MATLAB files.

The SNR computed on the basis of 1 000 points equals 12.3 dB. On figure 5 we can see the squared error versus the number of iterations.

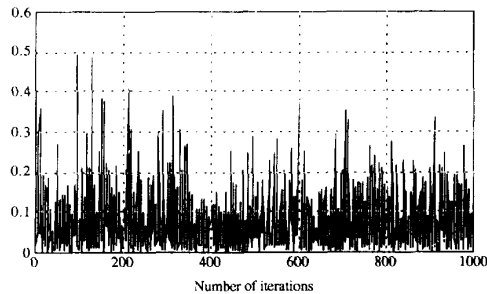


Figure 5 Squared error

The next figure gives us the estimation, in the mean square sense, of the channel impulse response. This response typically spans over 30 symbol durations and its imaginary part reveal a second path arriving with a delay of 5T (1.6 ms), the magnitude of which is roughly the same as the main path one.

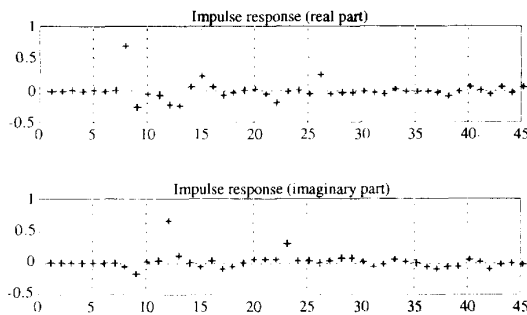


Figure 6 Impulse response

### III.2 Background job

The TCELP decoder (figure 7) includes the following operations :

- Excitation waveform determination
- Long-term filter computation
- Short term filter computation

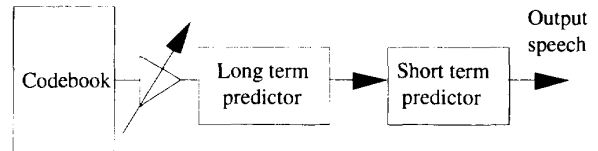


Figure 7 TCELP decoder

#### 1) Excitation waveform

The following method is used to create the waveform :

- The waveform index developed in binary expression.
- The sample is set to +1 when the bit value is 0, -1 when it is 1.
- Three zero samples are added after each significant sample.

Once created, the excitation gain makes it possible to set the sampling level. This is done four times a frame.

#### 2) Long term filtering

The two LTP parameters (pitch and gain predictor) allow the long term filter design twice a frame. The excitation samples go through this filter to rebuild the glottal signal.

#### 3) Short term filtering

The set of CLSP parameters allows the predictor filter computation. Each parameter is computed using its quantized CLSPs function. After short term filtering, the glottal signal leads to the speech samples.

### III.3 Speaking task

It only consists of writing the speech samples, at 8 kHz, on the Digital-Analog-Converter via the SSI interface.

## IV. UNIDIRECTIONAL DIGITAL PHONE

The unidirectional digital phone now runs, in real-time, on the test-bench (figure 1). Two DSP56001 are used to create the unidirectional digital phone. Wire lines were used for transmission between the transmitter and the receiver in order to validate the system.

Speech samples acquisition is performed using a microphone and a 16-bit Sigma-Delta Analog-to Digital

Converter (ADC). To run the transmitter needs a little memory extension (9k words).

A Digital-to-Analog Converter (DAC) is connected to one of the DSP interface. It is followed by a reconstruction low pass filter to drive the speakers. The receiver does not need memory extension. After validation on the test bench, a test pool experiment (figure 8) was carried out in order to test our receiver. The experiment did not include the CELP coding. The binary stream, corresponding to a sentence was previously coded and stored into the DSP memory. It was continuously emitted using unidirectional and omnidirectional emitters. As demodulation and speech synthesisization were performed in real-time by the receiver, a listen monitoring made it possible to check the transmitted and synthesized speech quality. Moreover the signal was recorded on Digital Audio Tape to allow further studies in laboratory. The same experiment was carried out in Brest harbour and gave good results.

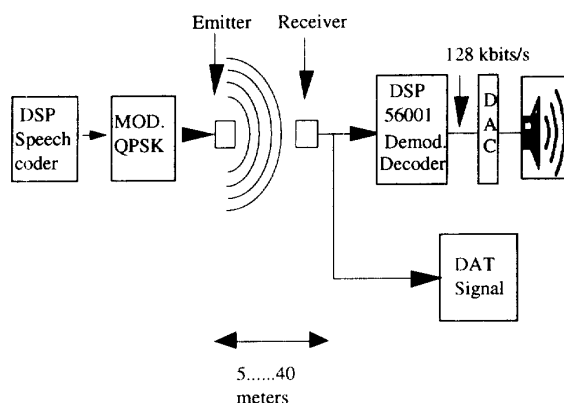


Figure 8 Test pool experiment

## V. PERFORMANCES

### V.1 Frequency response

The coder/decoder has been tested in the speech frequency band [300 .. 3600 Hz][3]. Because of the codebook structure the CELP coder/decoder cannot synthesize two frequencies (1 kHz, 2 kHz). In spite of this drawback, the system synthesizes good quality speech, any way better than analog system can do.

### V.2 Time processing

As stated previously, the computational time must be lower than 20 ms. The testing-bench [3] allows easy measurements. They have been done for silent, normal

speech (1500 speech frames). The whole computational time, in the transmitter, fluctuates between 15 and 18 ms, allowing real time processing. The computational time for the speech decoder is lower, close to 1.5 ms. For the receiver this time only reaches 10.5 ms. Table 2 shows the needs in MIPS, it clearly appears than real time processing can be achieved on DSP chip whose computational capacities overrun 12.15 MIPS.

in MIPS	Computational requirement
Transmitter	12.15
Receiver	6.9

Table 2 : Computational requirement

### V.3 Signal to noise ratio

The coder/decoder runs also in C language (floating point version). For the same sentence In spite of rounding effects, performances obtained are close to those performed in C language.

### V.4 Memory allocation

The memory allocation is the following. The Coder uses 2.5 Kwords of program space and 6 kwords of data space. The receiver only uses 2.11 kwords of program space and 2.827 kwords of data space split between X and Y memory map.

### V.5 Diver's speech

To adapt the CELP coder/decoder to diver's speech, a set of sentences pronounced by 5 divers (11 minutes of speech) were recorded in diver's mask. Quantization levels were computed and updated using these new data. An improvement in speech quality was noted. The bubble effect does not affect the speech spectral envelope because inspiration and speech are produced alternately.

## VI. CONCLUSIONS / PERSPECTIVES

This paper has presented the real time processing of a unidirectional digital phone on a test bench and in real environment. The real time processing is achieved in spite of computational complexity. Good quality speech is obtained by the Line Spectrum Pairs utilization and the whole system gave us good results in spite of the extreme severity of the environment.

#### ACKNOWLEDGMENT

We would like to thank IFREMER, Thomson Sintra, Orca Instrumentation and Subatlantic for their respective contribution. This work was partly supported by the DRIRE (Direction Régionale de l'Industrie, de la Recherche et de l'Environnement de Bretagne) as well as the CRB (Conseil Régional de Bretagne).

#### REFERENCES

- [1] D. D. Falconer, " Jointly Adaptive Equalization and Carrier Recovery in Two-dimensionnal data communication systems " Bell Syst. Tech. J., vol 55, pp. 317-334, March 1976.
- [1] A. Glavieux, J. Labat , " Transmission d'images sur canal acoustique sous-marin " , GRETSI-NICE, pp.161-176, 1-5 Juin 1987.
- [3] A. Goalic and S. Saoudi, "Real time processing of a 5.45 kbits/sec TCELP coder/decoder on DSP MOTOROLA56001," ICSPAT'93, Santaclara, USA, pp. 1400-1406, October 1993.
- [4] S. Saoudi, J.M. Boucher and A. Le Guyader, "A new efficient algorithm to compute the LSP parameters for speech coding", Signal Processing, Vol. 28, No 2, pp.201-212, August 1992.