MÉMOIRE D'HABILITATION À DIRIGER DES RECHERCHES

Présenté à
L'UNIVERSITÉ DE MARNE LA VALLÉE

CODAGE DE LA PAROLE À BAS ET TRÈS BAS DÉBIT

TRANSFORMATION DE LA VOIX

G. Baudoin
École Supérieure d'Ingénieurs en Électronique et Électrotechnique
Laboratoire Signaux et Télécommunications

---

20 novembre 2000

Membres du jury:

- BILDSTEIN Paul, Professeur, ESIEE, laboratoire Signaux et Télécommunications.
- CHOLLET Gérard, Directeur de Recherche, CNRS, Département traitement du signal et des Images, ENST Paris.
- LACOUME Jean-Louis, Professeur, Institut National Polytechnique de Grenoble, LIS, ENSIEG Grenoble.
- LOUBATON Philippe, Professeur, université de Marne-la-Vallée, laboratoire Systèmes de Télécommunications.
- MENEZ Jean, Professeur, Université de Nice Sophia-Antipolis, Département GE&II, IUT de Nice Côte d'Azur.
- RODET Xavier, Professeur, Université Paris 6, Équipe Analyse-Synthèse, IRCAM.
- WELLEKENS Christian, Professeur, Institut Eurécom, Sophia-Antipolis.

Ce document est structuré en 2 parties. La $1^{ère}$ partie est un curriculum vitae détaillé. Elle résume mes activités de recherche et d'enseignement et comprend une bibliographie personnelle.
La $2^{ème}$ partie présente mes travaux dans les domaines du codage de la parole et de la transformation de voix.
Ce mémoire comprend 2 bibliographies : une bibliographie personnelle à la fin de la première partie et une bibliographie générale à la fin du document.

G. Baudoin,
Version juillet 2000

Table des matières

Première partie

Curriculum vitae et résumé des activités de recherche et enseignement

Cette 1ère partie du document est structurée en 4 sections :

1. Curriculum Vitae,
2. Résumé des activités d'enseignement,
3. Résumé des activités de recherche et d'animation d'équipe,
4. Liste des publications personnelles.

Les références citées dans cette 1ère partie se réfèrent toutes à la bibliographie personnelle qui est donnée dans la section 5.2.

1 Curriculum-Vitae

1.1 État civil

Geneviève Baudoin
45 ans, mariée, 2 enfants.
ESIEE, Laboratoire Signaux et Télécommunications,
Noisy Le Grand, BP 99, CEDEX, 93162
téléphone : 01-45-92-66-46
email : baudoing@esiee.fr

1.2 Diplômes

Baccalauréat série C 1972, ingénieur télécom PARIS ENST 1977.

1.3 Expérience professionnelle

- Septembre 1977 à novembre 1979 : Assistante à l'université de Paris-Ouest.
- Novembre 1979 à novembre 1980 : Philips-LEP, ingénieur de recherche au LEP, laboratoire de Philips à Limeil-Brévannes, équipe d'imagerie médicale par ultrasons.
- Novembre 80 à aujourd'hui : Enseignant-chercheur à l'ESIEE.

1.4 Situation actuelle

- Déléguée à la recherche à l'ESIEE depuis février 2000.
- Professeur associé $2^{ème}$ degré depuis 1992.
- Disciplines d'enseignement et de recherche : traitement du signal, traitement de la parole, communications numériques.

1.5 Activités depuis l'arrivée au groupe ESIEE

- Déléguée à la recherche à l'ESIEE depuis février 2000.
- Responsable du laboratoire Signaux et Télécommunications (septembre 1997 à février 2000).
- Année sabbatique à l'université technique de Hambourg (96-97).
- Responsable du laboratoire de recherche PSI Parole, Signal et Images (94-95, 95-96).
- Période sabbatique de 3 mois chez Tektronix à Beaverton USA (1990),

- Responsable du département signaux et télécommunications (1985-1992)
- Enseignante dans la division télécommunications (1981-1984) : participation aux enseignements d'électronique et de traitement des signaux. Encadrement de projets industriels dans le domaine des télécommunications.

2 Activités d'enseignement

Ma charge annuelle d'enseignement est d'environ 110 h de cours et 70 h de TD (classiques crayon papier, Matlab, CAO) ou TP (DSP). À cette charge s'ajoute l'encadrement de plusieurs projets internes et stages en entreprise.

2.1 Formation d'ingénieurs ESIEE

À mon arrivée à l'ESIEE, j'ai participé aux enseignements d'électronique et de traitement du signal ainsi qu'à l'encadrement de projets industriels.

Puis j'ai créé en 1984, en collaboration avec F. Baillieu, la majeure[1] « Systèmes intégrés de traitement du signal, SITS ». Cette spécialisation comprenait une formation sur la conception de circuits intégrés numériques et analogiques basse-fréquences ainsi que des enseignements de traitement du signal et d'électronique.

Plus récemment, j'ai créé la majeure « Communications numériques et réseaux de télécommunications, CN » et participé à la création de la majeure « Télécommunications et traitement du signal, TTS ».

J'ai été responsable des majeures SITS et CN pendant plusieurs années.

J'interviens aujourd'hui en tronc commun et dans les majeures « télécommunications et traitement du signal » et « systèmes électroniques et micro-électroniques ».

Je participe aux enseignements suivants :

- $3^{ème}$ année :
  Théorie du signal.
- $4^{ème}$ année :
  Communications numériques,
  Filtres numériques,
  Processeurs de traitement numérique du signal,
  Étude de cas simulation d'une liaison hertzienne numérique.
- $5^{ème}$ année :
  Techniques avancées de communications numériques,
  Traitement de la parole.

Je suis responsable du bloc d'enseignement « Communications numériques et réseaux de télécommunications » (130h au total).

2.2 Formation de technologues ESTE

La formation ESTE est une formation bac + 3.

- $2^{ème}$ année :
  Je suis responsable du cours : Introduction au traitement numérique des signaux et utilisation des processeurs de traitement numérique du signal.

[1] Une majeure est une spécialisation technique, correspondant aux enseignements des 2 dernières années de l'ESIEE.

– $3^{ème}$ année :

Dans le cadre de la majeure « électronique des télécommunications », je participe au cours de communications numériques.

2.3 Participation a des formations de $3^{\text{ème}}$ cycle

J'interviens dans le DEA « systèmes de communications hautes fréquences », dans lequel l'ESIEE est cohabilité avec l'université de Marne La Vallée, le CNAM et l'INT. Je participe au cours de communications numériques (j'effectue environ 20 h de cours).
J'encadre par ailleurs régulièrement (en moyenne 1 par an) des stagiaires de ce DEA et du DEA d'électronique d'Orsay (avant la création du DEA de Marne La Vallée).

2.4 Encadrements de projets

J'encadre, tous les ans, plusieurs projets internes et stages industriels de $4^{ème}$ et de $5^{ème}$ années à l'ESIEE, 1 ou 2 stages de DEA, et des stagiaires étrangers en projets de fin d'études.

Les projets internes de $4^{ème}$ année durent 6 semaines et se font en trinôme.

Les stages industriels de $4^{ème}$ année durent 3 mois et ceux de $5^{ème}$ année 6 mois.

2.5 Formation Continue

J'ai organisé de nombreux stages de formations continues et participé aux stages de traitement numérique des signaux, communications numériques, processeurs de traitement numérique des signaux.

2.6 Interventions extérieures à l'ESIEE

2.6.1 INT

Je participe au cours de traitement de la parole à l'INT. J'ai en charge le cours de codage de la parole (durée 18h).

2.6.2 Enseignement dans des écoles d'été

Je vais participer à une école d'été intitulée « Communications et automatisme industriels » qui aura lieu en août 2000 à l'ISSAT en Syrie.

J'effectuerai le cours de communications numériques.

Je vais par ailleurs intervenir dans une école d'été sur les DSP. Cette école est organisée par l'école polytechnique et aura lien en novembre 2000.

2.7 Développement de travaux pratiques

En plus des TP d'électronique classiques auxquels j'ai participé à mon arrivée à l'ESIEE, j'ai développé 3 sortes de travaux pratiques :

1. Des TP utilisant Matlab. J'ai mis au point environ 20 séances de 4h, avec corrigé pour la plupart d'entre elles, sur les thèmes suivants :

TFD, FFT,
Convolution, corrélation,
Illustration du cours de distributions par 2 applications de traitement de signal : modulations d'amplitude et échantillonnage,
Signaux aléatoires,
Estimation spectrale et modélisation des signaux aléatoires,
Analyse des cellules élémentaires de filtrage numérique,
Calcul des filtres IIR et FIR,
Implantation d'un filtre IIR en précision finie format fixe,
Changement de fréquence d'échantillonnage, filtrage multicadence,
Analyse des signaux de parole,
Codage de parole,
Codes en ligne et récepteur optimal,
Modulations numériques QAM,
Modulations numériques à enveloppe constante,
Corps de Galois et codes BCH,
Codes de Reed-Solomon,
Codes convolutifs, algorithme de Viterbi,
Génération de séquences à bonnes propriétés d'auto et d'intercorrélation, Application à l'identification de réponses impulsionnelles, à l'étalement de spectre et au CDMA,

2. Des TP utilisant des logiciels de CAO en traitement de signal et communications numériques.

   J'ai commencé avec le logiciel SPW de Cadence, et j'utilise maintenant le logiciel ADS de HP.

   L'intérêt d'ADS est qu'il permet de concevoir et simuler aussi bien la partie RF que la partie bande de base d'une chaîne de communications numériques, car il intègre plusieurs moteurs de simulation tels que spice pour la simulation linéaire, une simulation en paramètres s, une simulation d'enveloppe, une simulation non-linéaire (*Harmonic Balance*),$\cdots$, et un simulateur *Synchronous Data Flow* pour la simulation bande de base.

   ADS permet la co-simulation analogique-numérique et s'interface avec Matlab.

   Les thèmes des TP que j'ai développés sont les suivants :
   – Simulation de la partie bande de base d'un faisceau hertzien à modulation 16 QAM. Cette étude dure 15 h et se fait en complément d'une étude de 15 h réalisée par mes collègues, sur le segment RF. La partie numérique comprend 3 parties :
     Développement du modulateur et du récepteur optimal,
     Évaluation des performances par méthode de monte-Carlo,
     Trajets multiples, égalisation (LMS).
   – Simulation d'un modulateur GMSK et d'un modulateur Edge.

3. Des TP DSP[2].

   J'ai développé 2 sortes de TP DSP :

   Des TP dans lesquels les étudiants utilisent un DSP sans avoir à le programmer en assembleur et où l'accent est mis sur l'aspect temps réel et précision des calculs en format fixe. Le TP typique consiste à calculer en filtre IIR avec Matlab et à le synthétiser sous une forme cascade puis à entrer les coefficients des différentes cellules dans un fichier pour le DSP, et enfin à relever la fonction de transfert effectivement réalisée. Ce TP se fait en 4h.

[2]DSP = *Digital Signal Processor.*

Des TP où les étudiants développent en assembleur une application simple, telle que des filtres numériques [25] ou un modulateur démodulateur FSK [30]. L'intérêt de ce dernier TP est qu'il est très visuel, qu'il peut être réalisé avec des étudiants assez jeunes (technologues ESTE par exemple) et qu'il permet d'illustrer les notions de génération de fréquence utilisant la circularité de la représentation en complément à 2, de filtrage numérique FIR et IIR, de modulation à phase continue. Mais il demande une durée minimum de 12h à 16h d'encadrement en laboratoire. Ces TP ont été présentés lors des conférences Texas-Instruments [25, 30].

3 Résumé des activités de recherche et d'animation d'équipes

3.1 Description succincte des thèmes et travaux de recherche et développement

Mon domaine de recherche est le traitement des signaux et plus particulièrement le traitement de la parole. Au cours des dernières années j'ai travaillé sur le codage de la parole à bas débit et sur la transformation de voix pour des applications à la synthèse vocale et au codage à très bas débit.

En parallèle avec cette activité de recherche, j'ai cherché à développer une expertise sur les processeurs de traitement numérique du signal (DSP). J'ai rédigé deux livres sur le sujet, en collaboration avec F. Virolleau [28, 29] et j'ai établi une collaboration suivie avec Texas-Instruments.

Depuis 97-98, tout en continuant mon travail sur le codage de la parole à très bas débit au travers d'un projet RNRT, j'ai infléchi l'orientation de mes travaux de recherche vers les communications numériques dans le cadre du nouveau laboratoire « systèmes de télécommunications » qui associe l'ESIEE, l'université de Marne La Vallée et le CNAM.

3.1.1 Traitement de la parole

J'ai choisi de présenter en détail, dans la $2^{\text{ème}}$ partie de ce document, mes travaux sur le codage de la parole et sur la transformation de voix, aussi ne les décrirai-je que d'une manière très succincte dans cette section.

3.1.1.1 Codage de la parole Je me suis intéressée au codage de parole à bas et très bas débit, typiquement inférieur à 4800 bps[3], en particulier aux 3 points suivants :

- Réduction de la complexité des codeurs CELP,
- étude de la limitation en débit des codeurs CELP,
- Développement d'un nouveau principe de codage à très bas débit (inférieur à 600 bps). Ce sujet a donné lieu à un projet RNRT (SYMPATEX) labellisé en 1999.

J'ai par ailleurs implanté plusieurs codeurs de parole sur DSP, dans le cadre de contrats industriels.

3.1.1.2 transformation de la voix J'ai étudié la transformation de la voix (timbre et prosodie) pour des applications à la synthèse de parole à partir du texte, et pour des applications de codage à très bas débit. Dans ce dernier cas, il peut en effet être intéressant de transmettre au décodeur quelques informations concernant le locuteur et d'effectuer une personnalisation de la voix décodée.

[3] j'utilise le sigle bps pour indiquer bits par seconde.

Sur ce thème général, je me suis intéressée à la décomposition source-filtre du signal vocal (thèse de Jianping Liu [85]) et à la suite de ce travail j'ai obtenu une convention CNET sur la transformation du timbre de la voix.

3.1.1.3 Comparaison de paramètres spectraux pour la reconnaissance de parole en milieu bruité J'ai participé à un projet financé par le CNRS sur la comparaison de paramètres spectraux pour la reconnaissance de parole en milieu bruité. J'ai travaillé à l'ESIEE avec une collègue P. Jardin et une étudiante slovène J. Gross en projet de fin d'études.

Nous avons travaillé avec un système de reconnaissance utilisant la technique DTW et avec une base de données de parole bruitée EUROM0.

Les résultats ont été présentés au GRETSI [15] et lors d'une école d'été OTAN [16].

3.1.2 Traitement numérique des signaux et communications numériques

Le traitement de la parole a constitué l'essentiel de mon activité, mais j'ai toutefois travaillé sur d'autres applications de traitement de signal ou de communications numériques qui sont décrites dans les paragraphes suivants.

3.1.2.1 Codage d'ECG à bas débit, étude d'un holter numérique J'ai étudié au milieu des années 80, en lien avec le CECA[4], la faisabilité d'un Holter numérique.

Un Holter est un système utilisé en cardiologie pour tester l'électrocardiogramme (ECG) au cours d'activités ordinaires et qui enregistre 24 h d'ECG sur un enregistreur à cassette magnétique porté par le patient pendant une journée.

L'ECG est un signal basse fréquence que l'on peut échantillonner à 250 Hz. En supposant que la numérisation se fasse sur 16 bits, une durée de 24 h d'ECG représente 43 Moctets. à l'époque de cette étude, les mémoires vives étaient de capacités très inférieures à celles d'aujourd'hui et il était donc indispensable pour réaliser un holter numérique portable de comprimer au moins par un facteur 10 le signal d'ECG afin de réduire la taille de la mémoire nécessaire.

J'ai travaillé sur le sujet avec un étudiant en thèse (M. Chaouche). Nous avons développé un algorithme de compression utilisant une décomposition de Karuhnen-Loëve.

La difficulté du problème venait en grande partie de la forte variabilité de la « ligne de base », c'est-à-dire de composantes basses fréquences qui se superposaient au signal utile et qui provenaient de la fixation non parfaite des électrodes de l'ECG. Il n'était pas possible d'éliminer ces composantes par simple filtrage passe-bas, cependant la forme des impulsions d'ECG restait assez reconnaissable visuellement. J'ai proposé une méthode de pré-traitement utilisant des techniques d'analyse d'images de type érosion et dilatation [13].

Nous avons testé et validé les algorithmes sur des cassettes de 24 h d'ECG fournies par le CECA ainsi que sur des cassettes d'ECG enregistrées sur des skieurs militaires en action.

Nous avons par ailleurs implanté ces algorithmes sur un des premiers processeurs de traitement de signal le TMS320C10 [51, 12, 14].

3.1.2.2 Annulation d'échos sur des images de télévision par filtrage adaptatif J'ai été invitée en 1990 à passer 3 mois dans un laboratoire de recherche et développement de Tektronix à Beaverton aux USA.

[4]CECA = Centre d'Exploratoire Cardiologique Ambulatoire.

Pendant ce séjour je me suis intéressée à l'annulation des échos sur les images de télévision. J'ai plus particulièrement étudié la convergence des algorithmes LMS sur des signaux cyclostationnaires [8] en reprenant les travaux d'O. Macchi.

3.1.2.3 Systèmes d'identification sans contact à 13.56 MHZ J'ai commencé récemment à infléchir mes travaux de recherche vers les communications numériques, car ce domaine correspond mieux aux activités du nouveau laboratoire « Systèmes de télécommunications » qui regroupe l'ESIEE, l'UMLV et le CNAM sur le site de Marne La Vallée.

Dans ce cadre, j'ai effectué un PCT (Pré Conseil Technologique) pour la société STID, et ceci en collaboration avec mes collègues P. Bildstein et C. Ripoll.

Dans ce PCT, nous avons fait un point sur les systèmes d'identification sans contact à 13,56 MHz (norme ISO/IEC 14443 en cours de définition) et nous avons identifié les principaux points critiques pour l'amélioration des lecteurs de badges, étiquettes ou cartes à puces sans contact. Les badges sont des systèmes téléalimentés par le lecteur et communiquent avec lui par couplage inductif en utilisant une modulation de charge [32].

Les lecteurs actuels sont essentiellement analogiques. j'ai proposé un nouveau principe de lecteur numérique travaillant par sous-échantillonnage des signaux reçus. J'ai effectué un premier test du principe sur un signal émis par un badge et enregistré sur un oscilloscope numérique rapide [20].

3.1.2.4 Implantation d'algorithmes de traitement de signal et de communications numériques sur DSP J'ai utilisé différents types de DSP format fixe, format flottant et plus récemment un DSP à architecture VLIW[5] le TMS320C6201.

J'ai développé plusieurs applications typiques pour l'enseignement de l'implantation temps réel d'algorithmes de traitement de signal sur DSP [30, 25] et j'ai rédigé sur le sujet en collaboration avec F. Virolleau 2 livres publiés chez Dunod [28, 29].

J'ai implanté des codeurs de parole (voir plus haut) et des algorithmes de communications numériques en particulier récemment dans la perspective du développement des systèmes CDMA[6] [71, 18].

Exemple de développement d'une application complète : Génération et Traitement des signaux d'un gyromètre vibrant à excitation magnétique

Dans le cadre d'un contrat industriel avec la société ISNAV et la DGA [26], je travaille en collaboration avec un collègue de l'ESIEE O. Venard à la numérisation de l'électronique d'un gyromètre acoustique à bol vibrant excité magnétiquement mis au point par la société ISNAV.

Au terme du projet, nous aurons remplacé l'électronique analogique qui dérive en température par un traitement numérique plus sophistiqué, plus performant et plus fiable sur un DSP orienté contrôle, le TMS320F243. Nous devons concevoir les algorithmes, les implanter sur DSP et réaliser le système matériel complet incluant le DSP et les interfaces analogiques avec le capteur.

Le gyromètre vibrant magnétique (GVM) mesure la vitesse angulaire du solide sur lequel il est fixé. Il est constitué d'un cylindre métallique contenant un système d'excitation magnétique formé de bobines enroulées sur une tôle en forme de croix à 8 branches. Quatre branches sont utilisées pour l'excitation et les 4 autres servent à la mesure. L'excitation est appliquée à 4 bobines à 90 degrés. Lorsque le cylindre est excité à sa fréquence de résonance, des nœuds et des ventres de vibration mécanique apparaissent. Sous l'influence des forces de Coriolis, quand le cylindre est en rotation, la

[5] VLIW = Very Long Instruction Word.

[6] CDMA = Code Division Multiple Access.

position relative des nœuds de vibration se déplace à une vitesse proportionnelle à la vitesse angulaire du mobile. Le principe de la mesure de vitesse angulaire consiste à asservir la position des nœuds de vibration en appliquant une contre-réaction sur les 4 bobines de mesure entrelacées avec les 4 bobines d'excitation, de façon à maintenir ces nœuds en une position fixe. L'amplitude de la contre-réaction appliquée est alors proportionnelle à la vitesse angulaire recherchée.

Le traitement implanté sur DSP est constitué de la recherche de la fréquence de résonance du cylindre, de plusieurs boucles à verrouillage de phase, de filtres, de la génération des signaux d'excitation pour les bobines (sous forme PWM)[7], et de l'acquisition des signaux sur les bobines d'excitation et de contre-réaction.

3.2 Encadrement doctoral, encadrement de projets longs

3.2.1 Thèses encadrées

J'ai reçu en 1990 de l'université d'Orsay, un agrément local pour diriger des thèses.

J'ai encadré 4 thèses.

J'ai encadré directement 3 d'entre elles (M. Chaouche, M. Mauc et J. Liu) qui ont été soutenues en 87, novembre 93 et décembre 93. Et j'ai co-encadré la thèse de J. Černocký avec G. Chollet. Il s'agissait d'une thèse en cotutelle dirigée par le P^r Sebesta en république tchèque. La thèse a été soutenue en décembre 98.

Les sujets de ces 4 thèses ont été les suivants :

- M. Chaouche : étude de faisabilité d'un système portable d'enregistrement et de traitement de 24 heures d'électrocardiogramme (Holter numérique).
- M. Mauc : Réduction de la complexité des algorithmes de codage de la parole de type CELP, recherche de l'excitation par méthode multi-étapes et sous-échantillonnage, application au standard FS-1016.
- J. Liu : Amélioration de la décomposition source-filtre du signal vocal, étude de la variabilité des paramètres de l'onde glottique, application à la transformation de voix.
- J. Cernocky : Traitement de la parole s'appuyant sur des unités segmentales déterminées automatiquement : applications au codage à très bas débit et à la vérification du locuteur.

J'encadre depuis septembre 99 un $5^{ème}$ thésard Bashar Abdulrahman.

- Sujet de la thèse de B. Abdulrahman : étude des techniques de pré-distorsion numériques adaptatives en bande de base pour le traitement des non-linéarités de l'étage d'amplification de puissance des mobiles des futurs systèmes de communications mobiles et de réseaux locaux radio.

À partir de septembre 2000, j'encadrerai en cotutelle avec le P^r Sebesta, de l'université technique de Brno, la thèse de R. Marsalek. Le sujet de cette thèse sera lié à celui de B. Abdulrahman, mais comportera un aspect architecture globale de l'émetteur.

3.2.2 Encadrement de stagiaires en projets longs

J'ai par ailleurs supervisé plusieurs étudiants en projets longs : 1 ou 2 stagiaires de DEA par an, et des stagiaires étrangers en projet de fin d'étude ou en thèse pour une durée de 4 mois en moyenne

[7] PWM = Pulse Width Modulation.

dans le cadre de programmes de collaboration européens Tempus, Proteus, Erasmus ou du réseau ENTREE[8]. Il s'agit en particulier de :

- Michal Polanski, 2000, étudiant tchèque de l'université de Brno, en projet de fin d'études, qui travaille sur le projet de gyromètre vibrant.
- Farid Moussous, 2000, étudiant du DEA « systèmes de communications hautes fréquences », qui développe un système de linéarisation pour un amplificateur de puissance, pour le standard EDGE.
- Y.-P. Nakache, 2000, étudiant en projet de fin d'études ESIEE qui travaille en lien avec Thomson-CSF et moi-même sur le projet RNRT SYMPATEX.
- Christian Konki, 1999, étudiant du DEA « systèmes de communications hautes fréquences », qui a étudié un démodulateur pour les systèmes d'identification sans contact à 13,56 MHz.
- Roman Marsalek, 99, étudiant de l'université de Brno qui a effectué son projet de fin d'études à l'ESIEE sous ma supervision. Il a travaillé sur la simulation d'une liaison UMTS et sur l'implantation d'algorithmes de synchronisation pour le standard UMTS sur DSP VLIW TMS320C6201 [10]. Je l'encadre, maintenant, à distance pendant son DEA (ou plutôt l'équivalent) à Brno. Il reviendra à l'ESIEE pour sa thèse en septembre 2000.
- J. Prokes, 99, étudiant de l'université de Brno en fin de thèse qui est venu passer 3 mois à l'ESIEE. Il a travaillé sur l'utilisation de DSP VLIW pour l'implantation d'un décodeur de Viterbi [10].
- D. Janu, 98, étudiant de l'université de Brno. Je l'ai encadré pendant son projet de fin d'étude à l'ESIEE. IL a travaillé sur la simulation d'une liaison CDMA et son implantation sur DSP [71].
- J.-P. Goldman, 96, stagiaire du DEA « Traitement du signal et des images » de l'ENSEA, qui a participé au projet CNET sur la conversion de voix.
- E. Steinbach, 95 stagiaire de l'université de Karlsruhe qui a travaillé sur le projet CNET.
- J. Gross, 94, stagiaire de l'université de Ljubljana qui a travaillé avec moi sur la comparaison de paramètres spectraux pour la reconnaissance vocale en milieu bruité dans le cadre d'un projet CNRS [15, 16].
- M. Zganec, 94, stagiaire de l'université de Ljubljana, qui a travaillé sur le projet TEMPUS [9] et a implanté sous SPW un logiciel de calcul de bac de filtres multicadence.
- M. Jelinek, 94, stagiaire de l'université de Brno qui a travaillé à l'ESIEE sur les codeurs de parole à bas débit et participé au contrat pour la société ACSYS [74], puis est parti à l'université de Sherbrooke pour une thèse sur le codage de parole.

3.2.3 Situation actuelle des thésards que j'ai encadrés

M. Chaouche a été enseignant à l'université d'Oran, puis pour des raisons personnelles a dû revenir en France où il enseigne maintenant à l'école d'ingénieurs ECE.

M. Mauc a d'abord travaillé chez Matra-communications au sein du groupe de recherche en traitement de la parole dans le prolongement direct de sa thèse. Il a développé des codeurs de parole pour des systèmes de radiocommunications. Il travaille maintenant chez Nortel communications sur le développement de systèmes d'évaluation de la qualité audio sur les liaisons de téléphonie cellulaire.

J. Liu a d'abord été employée par la société ACSYS sur des systèmes de reconnaissance vocale, puis elle a continué chez Alcatel radiotéléphone dans le groupe traitement de la parole. Elle a étudié les systèmes de communications mains libres dans les voitures. Elle travaille maintenant sur les communications numériques dans les réseaux cellulaires.

[8]ENTREE = *European Network for Training and Research in Electrical Engineering.*

J. Černocký est enseignant-chercheur à l'université technique de Brno. Il continue à travailler de manière très active sur le traitement de la parole. Il a coordonné 2 projets d'enregistrements de bases de données de la langue tchèque, l'un pour la société Siemens et l'autre en lien avec la société Matra-communications dans le cadre d'un projet européen. Il encadre actuellement un thésard sur le codage de parole à bas débit.

3.2.4 Participation à des jurys de thèses

En plus des jurys des thésards que j'ai encadrés, j'ai participé à 3 autres jurys de thèses en traitement de la parole :

- Jury de thèse de J. Stylianou sur la modélisation HNM (ENST),
- Jury de thèse de C. Gérard sur l'utilisation de la décomposition en ondelettes en reconnaissance de parole (université d'Orsay),
- Jury de thèse de K. Ouaissa sur l'optimisation des techniques de compression des signaux multimedia (CNAM).

3.3 Animation d'équipes

J'ai assuré différentes responsabilités à l'ESIEE.

- Je suis déléguée à la recherche depuis février 2000.
- J'ai été Responsable du laboratoire Signaux et Télécommunications, laboratoire d'enseignement et de recherche, de 1997 à février 2000.
- J'ai animé le laboratoire de recherche PSI (Parole, Signal et Images) en 94-95 et 95-96.
- J'ai été Responsable d'un département d'enseignement (département Télécommunications) de 1984 à 1992.

À mon arrivée à l'ESIEE, il n'y avait pas d'activité de recherche organisée, par contre les enseignants de l'école participaient à de nombreux projets de développement financés par des industriels et impliquant les étudiants de dernière année. C'est à la suite d'un de ces projets financé par le CECA (Centre d'Exploration Cardiologique Ambulatoire) que j'ai encadré ma $1^{ère}$ thèse sur l'enregistrement et le traitement de 24 heures d'électrocardiogramme.

De 1984 à 1992, j'ai assuré la responsabilité du département signaux et télécommunications, les départements étant alors avant tout des structures d'enseignement.

Lors du déménagement de l'ESIEE à Marne La Vallée en 87, la direction du groupe a décidé de lancer un programme de recherche dont P. Bildstein a pris la responsabilité. J'ai participé avec plusieurs collègues au lancement de ce programme.

J'ai animé pendant 2 ans, depuis sa création en octobre 94 jusqu'en septembre 96, le laboratoire PSI (Parole Signal et Image). à mon départ en année sabbatique (septembre 96) à l'université technique de Hambourg, mon collègue G. Bertrand a repris ce rôle.
Le laboratoire PSI était constitué d'enseignants-chercheurs de 3 départements différents (informatique, mathématiques, signaux et télécommunications) qui avaient peu travaillé ensemble jusque là, il était important lors de la première année de présenter les domaines algorithmiques développés dans l'équipe. Dans ce but nous avons organisé une série d'exposés de type tutorial. Les réunions scientifiques et techniques du laboratoire se sont ensuite centrées sur les travaux de recherche en cours.
Le laboratoire a accueilli plusieurs thésards ainsi que des chercheurs étrangers pour des durées longues (A. De Albuquerque, B. Boianov, S. Tomasic) ou des séjours courts (J. Tasic,T. Slivnick, B. Zajc).

Le groupe ESIEE a récemment été restructuré en laboratoires d'enseignement et de recherche. De septembre 97 à février 2000, j'ai été responsable du laboratoire signaux et télécommunications qui

est maintenant non seulement une structure d'enseignement mais aussi une structure de recherche. L'équipe est constituée d'une dizaine de chercheurs-enseignants, avec 2 grands domaines de compétences : les techniques RF et micro-ondes d'une part et d'autre part le traitement du signal et les communications numériques. Nous avons décidé d'utiliser cette complémentarité de compétences et de focaliser nos activités de R&D sur les communications sans fil. Ce thème couvre aussi bien les applications dans le domaine des télécommunications (mobiles, satellites, réseaux locaux radios) que les communications courtes distances (systèmes d'identification sans contact par exemple). Cette complémentarité de compétences est précieuse pour la conception de systèmes de plus en plus intégrés où les aspects analogiques et numériques sont fortement imbriqués.

Les activités de R&D du laboratoire sont centrées sur le thème « Architectures, algorithmes et circuits intégrés pour les émetteurs-récepteurs de radiocommunications ». Ce thème comprend les rubriques suivantes :

- Architectures et circuits actifs RF : conception d'amplificateurs de puissance large bande multimode multi-standard pour les systèmes de communications mobiles de $3^{\text{ème}}$ génération, étude de l'influence des non-linéarités de l'amplificateur de puissance dans une liaison de radiocommunications numériques, compromis linéarité-rendement, techniques et circuits de linéarisation.
- Fonctions hyperfréquences pour les microsystèmes autonomes : télé-alimentation (antenne, adaptation et système redresseur à très faible niveau), électronique à ultra faible consommation, rétro-modulation à ultra faible consommation.
- émetteurs-récepteurs RF faible distance, faible consommation pour communications courte distance, lecteurs de cartes à puce sans contact.
- égalisation aveugle, algorithmes et implantation temps réel.
- Implantation d'algorithmes de communications numériques et de traitement de signal sur DSP.
- Circuits intégrés de communications numériques.

Le laboratoire participe par ailleurs activement à 2 pôles de R&D : le Pôle Francilien des Microsystèmes (PFM) et l'équipe « systèmes de communications ».

Le PFM regroupe outre l'ESIEE, l'IEF à l'université d'Orsay et l'ENS Cachan. Nous y intervenons sur le thème du développement de fonctions hyperfréquences pour les microsystèmes autonomes.

L'équipe « systèmes de communications » comprend l'ESIEE, l'UMLV et le CNAM. Nous participons à 2 des projets fédérateurs, l'un portant sur l'intégration d'amplificateurs de puissance et de circuits de linéarisation pour émetteurs multistandard-multimodulation, et l'autre portant sur la propagation dans les canaux RF et incluant les questions de mesure et modélisation de la propagation et d'égalisation de canal.

3.4 Collaborations académiques nationales et internationales

3.4.1 Collaboration avec l'université de Ljubljana

Cette collaboration a commencé lors d'un projet TEMPUS « Interactive distance learning of digital signal processing ». J'ai coordonné le projet au niveau ESIEE. D'une durée de 3 ans (91 à 94), ce projet a été animé par le Pr G. Caine de l'université de Westminster. Les autres participants étaient : L'université de Ljubljana et l'université technique de Varsovie. Le programme comprenait plusieurs rubriques : création de TP en traitement de signal (avec le logiciel SPW), échanges d'enseignants-chercheurs et d'étudiants, vidéoconférences interactives par liaisons satellites entre les différents sites.

Depuis, j'ai conservé des liens avec les différents participants et en particulier avec les laboratoires de traitement du signal (Pr Tasic), d'intelligence perceptive (Pr Pavesic) et d'électronique (Pr Zajc) de l'université de Ljubljana. Nous avons reçu chaque année une aide du ministère des affaires étrangères en France et du ministère de la recherche en Slovénie pour le financement des échanges entre les 2 institutions (séjours d'enseignants-chercheurs et d'étudiants).

Pour 97, 98 et 99, nous avons obtenu une subvention au titre du programme d'actions intégrées franco-slovène PROTEUS, sur le thème de la segmentation des signaux et des images [31].

3.4.2 Collaboration avec le CNET, l'ENST et l'INRIA dans le cadre de la convention CNET sur la transformation de voix

Dans le cadre d'une convention (CTI) CNET sur la transformation de voix, j'ai travaillé en lien avec le CNET, l'ENST et l'INRIA :
CNET Lannion : B. Cherbonnel, O. Boeffard (sous la direction de C. Sorin),
ENST : E. Moulines, I. Stylianou [21]. I. Stylianou a effectué un post-doc d'un an à l'ESIEE.
INRIA : J. Levy-vehel.

3.4.3 Collaboration avec Boian Boianov du CLBE académie des sciences de Sophia Bulgarie

B. Boianov a effectué un séjour d'un mois à l'ESIEE. Nous avons travaillé ensemble sur l'analyse du formant du chanteur et sur les voix pathologiques [36, 37, 35].

3.4.4 Collaboration avec G. Chollet de l'ENST

Nous collaborons depuis plusieurs années [85, 15, 16, 43, 44, 23, 22, 40, 47, 49, 49, 48, 24] et nous avons co-encadré la thèse de J. Cernocky.

3.4.5 Collaboration avec l'université de Marne La Vallée et le CNAM

Des liens se sont établis entre l'ESIEE, l'université de Marne La Vallée (UMLV) et le CNAM en particulier au travers d'un DEA commun « Systèmes de communications hautes fréquences ». L'année dernière, il a été décidé de créer une équipe de recherche commune appelée « Systèmes de communications ». Une demande de reconnaissance officielle a été faite auprès du ministère de la recherche.

Les 3 institutions ont des compétences complémentaires dans le domaine des télécommunications : traitement du signal et communications numériques (UMLV et ESIEE), électromagnétisme (UMLV), optoélectronique (CNAM), hyperfréquences, MMIC (ESIEE, CNAM, UMLV), conception de circuits intégrés RF et numériques (ESIEE, CNAM).

Trois projets fédérateurs ont été définis. Je participe au projet « Intégration d'amplificateurs de puissance et de circuits de linéarisation pour émetteurs multistandard-multimodulation ». Je travaille avec un étudiant en thèse, Bashar Abdulrahman, sur la prédistorsion numérique adaptative en bande de base et sur l'influence des non-linéarités dans une liaison numérique.

3.4.6 Collaboration avec l'université de Brno

J'ai établi une collaboration avec le P^r Sebesta du département de radioélectricité de l'université technique de Brno. J'accueille chaque année 1 ou 2 étudiants de Brno en projet de fin d'études ou en

thèse pour une période de quelques mois [10, 71, 11].

J'ai par ailleurs encadré en cotutelle avec le P^r Sebesta, Jan Černocký qui a soutenu sa thèse fin 98. Jan Cernocky est maintenant enseignant-chercheur à l'université technique de Brno et nous continuons à travailler sur le codage de parole à très bas débit. Un de ses étudiants en thèse passera 4 mois à l'ESIEE à partir de septembre 2000 et participera au projet RNRT SYMPATEX sur le codage de parole à très bas débit.

J'encadre actuellement Roman Marsalek [10] qui va lui aussi effectuer une thèse en cotutelle à partir de septembre 2000. Sa thèse se passera à mi-temps à l'ESIEE et à mi-temps à l'université technique de Brno.

3.5 Séjours dans des laboratoires étrangers

J'ai effectué 2 séjours longs dans des laboratoires étrangers : dans le laboratoire « Video Electronic System Lab » de Tektronix aux USA et à l'université technique de Hambourg.

- Invitation de 3 mois en 1990 dans le laboratoire de recherche et développement de Tektronix à Beaverton USA, travail sur l'annulation des échos dans des images de télévision.
- Séjour sabbatique de 9 mois (1996-1997) à l'université technique de Hambourg dans le laboratoire du Pr N. Fliege. Le Pr N. Fliege avait effectué un séjour à l'ESIEE au début des années 80 et m'a accueillie dans son laboratoire. Malheureusement, pendant mon séjour il a quitté l'université de Hambourg pour celle de Mannheim, je n'ai donc pas pu vraiment travailler avec lui. J'ai profité de ce séjour pour finir le travail sur la transformation de voix (convention CNET) et la rédaction du 1^{er} ouvrage sur les DSP. J'ai participé à un enseignement sur les DSP et encadré plusieurs étudiants en projet.

J'ai, par ailleurs, effectué plusieurs séjours courts dans des universités européennes dans le cadre des programmes Tempus, Proteus et Erasmus, en particulier à l'université de Brno en république tchèque et à l'université de Ljubljana en Slovénie.

- Université de Brno : collaboration de type enseignement (programme Tempus) et recherche en collaboration avec J. Černocký et le Pr Sebesta.
- Université de Varsovie et de Westminster. Collaboration dans le cadre du projet TEMPUS « Interactive Distance Learning of Digital Signal Processing ».
- Université de Ljubljana : Collaboration dans le cadre du projet TEMPUS « Interactive Distance Learning of Digital Signal Processing ». puis du projet PROTEUS « Segmentation des signaux et des images, algorithmes et implantation multiprocesseurs ».

3.6 Participation à l'organisation de conférences, expertises, revue d'articles

Je suis membre du comité de programme du workshop international TSD, Text Speech and Dialog, qui s'est tenu en 98 et 99 en république tchèque et qui est prévu pour septembre 2000.

J'ai fait partie du comité d'organisation international du symposium IEEE "ISIE'96" (International Symposium on Industrial Electronics) qui a eu lieu à Varsovie en juin 96.

J'ai d'autre part participé à l'organisation scientifique et technique des conférences Texas Instruments qui se sont déroulées à l'ESIEE en juin 94, juin 95, septembre 96, et septembre 98. La prochaine conférence « 3^{rd} European conference on DSP Research and Education » se déroulera dans les lo-

caux de l'ESIEE en septembre 2000.

J'ai effectué 2 expertises pour l'ANVAR à la suite du travail sur la compression d'ECG. Ces expertises ont porté sur des propositions de projets de traitement de signaux biomédicaux. J'ai participé à plusieurs expertises de dossiers CIFRES ainsi qu'à celle de 2 projets de recherche soumis au ministère de la recherche slovène.

Enfin j'ai été sollicitée pour la revue d'articles : IEE en codage de parole à bas débit, pour une revue slovène, pour les workshops TSD'98, TSD'99 et TSD'2000 ainsi que la conférence IEEE ISIE'96 et les conférences Texas-Instruments.

4 Contrats et subventions de recherche et développement

4.1 Résumé

10 contrats ou subventions depuis 1989 :
- Membre d'un projet RNRT labellisé en 99 : projet SYMPATEX, dont j'assure la responsabilité pour l'ESIEE,
- 1 convention CNET sur 3 ans, 94-97,
- 3 contrats industriels (SECMAT, ACSYS, ISNAV-DGA),
- 1 subvention projet Elite Texas-Instruments,
- 1 projet PROTEUS,
- 1 projet TEMPUS,
- Participation à un projet CNRS,
- 1 PCT Pré-Conseil-Technologique ANVAR.
- Participation à plusieurs contrats industriels avant 90.

4.2 Liste des contrats

1. Projet RNRT SYMPATEX labellisé en 99 : SYstème de Messagerie unifiée avec présentation vocale des messages (PArole et TEXte), projet exploratoire d'une durée de 3 ans en collaboration avec les sociétés Thomson CSF Télécommunications, ELAN, INFO-réalités et l'ENST.
   Montant ESIEE = 340 KF.
2. Contrat avec la société ISNAV et la DGA 1999-2001 : Génération des signaux de contrôle et traitement des signaux d'un gyromètre magnétique vibrant, implantation sur DSP.
   Montant = 300 KF.
3. PCT ANVAR pour la société STID sur les systèmes d'identification sans contact à 13,56 MHz.
   Montant = 60 KF.
4. Convention de recherche (CTI) CNET $N^o 947B013$ sur 3 ans 94-97, transformation du timbre de la voix, création de voix nouvelles à partir de voix connues.
   Montant = 500 KF.
5. Projet Elite Texas-Instruments en 96-97, enregistreur vocal statique.
   Subvention = don de matériels + 50 KF.
6. Projet PROTEUS 97-98-99 (Actions Intégrées franco-slovènes), en collaboration avec l'université de Ljubljana, segmentation des signaux et des images, algorithmes et implantation multiprocesseur.

Financement de 3 à 6 séjours d'une semaine par an dans chaque pays + financement de stages longs.

7. Projet TEMPUS JEP 1326, durée 3 ans de 91-94, Interactive distance learning of Digital Signal Processing, en collaboration avec l'université de Westminster, l'université de Ljubljana, et l'université de Varsovie.
   Financement ESIEE = 180 KF + financement de plusieurs séjours longs pour des chercheurs.
8. Participation à un projet CNRS GRECO PRC CHM, 92-93 sur la reconnaissance vocale en milieu bruité.
   Montant ESIEE = 20 KF.
9. Contrat de R&D avec la société ACSYS, 92-93, Codeur CELP à bas débit.
   Montant = 250 KF.
10. Contrat de R&D avec la société SECMAT, 89-90, étude et implantation temps réel sur DSP d'un codeur de parole à 2400 bps.
    Montant = 250 KF.
11. Participation à plusieurs projets industriels avant 1990, à l'époque où les projets des étudiants de dernière année se déroulaient dans les locaux de l'école et où ces projets donnaient lieu à contrats. J'ai en particulier travaillé avec les sociétés PRESCOM (vocodeur bande de base), SECMAT (modem pour liaisons satellites, multiplexage parole données pour liaison satellite), CNET (logiciel de filtrage à capacités commutées), EDF (système temps réel d'analyse des signaux électriques), ATES AlcaTel ESpace (évaluation de la complexité et des performances des codes BCH).

5 Publications

5.1 Résumé

- 2 livres (Dunod).
- 1 conférence invitée à une conférence slovène.
- 3 articles de revues scientifiques françaises,
- 1 article de revue CCIP,
- 1 publication à l'académie bulgare des sciences,
- 4 chapitres dans des ouvrages collectifs,
- 20 publications dans des conférences ou workshops internationaux avec actes et comité de lecture,
- 9 publications dans des conférences ou workshops nationaux français ou étrangers avec actes et comité de lecture.
- 6 publications dans les conférences Texas-Instruments dont 5 avec actes et comité de lecture.
- Plusieurs rapports de contrats de recherche et développement.

5.2 Liste des publications personnelles

Livres

[1] G. Baudoin and F. Virolleau. *Les processeurs de traitement du signal, la famille TMS320C50*. DUNOD, ISBN 2 10 00 003049 3, Paris, 1997.

[2] G. Baudoin and F. Virolleau. *DSP-La famille TMS320C54x, développement d'applications*. Dunod, ISBN 2 10 004646 2, Paris, 2000.

Conférence invitée

[3] G. Baudoin. Speech coding at low and very low bit rates. In *Proc. ERK'99 conference*, Portoroz, Slovenia, Sept. 1999. 11–14.

Articles de revues

[4] G. Baudoin, J. Černocký, P. Gournay, and G. Chollet. Codage de la parole à bas et très bas débit. *annales des télécommunications*, to appear in 2000.

[5] J. Černocký, G. Baudoin, and G. Chollet. Alisp : Quelques outils pour l'analyse acoustico-phonétique de la parole. *revue parole*, to appear in 2000.

[6] P. Bildstein and G. Baudoin. La révolution du Silicium. *Chambre de Commerce et d'industrie de Paris, Le nouveau Courrier, numéro spécial de prospective*, 42 :64–66, ISSN 1162-3802, Feb. 1996.

[7] B. Boianov, S. Hadjidorov, and G. Baudoin. Method for evaluation of the energy in the singer formant. *comptes rendus de l'académie bulgare des sciences*, 48(8) :25–28, Jan. 1995.

[8] C. Gueguen, J. Leroux, J.C. Domenger, and G. Baudoin. sur l'influence du retour visuel sur la régulation de la posture. *revue agressologie*, 17 :63–66, 1976.

Chapitres dans des ouvrages collectifs

[9] G. Bazin, P. Sangouard, G. Baudoin, C. Ripoll, P. Nicole. *Revue nano-micro*, chapter Microsystèmes autonomes sans fils. Hermès, to appear in 2000.

[10] J. Černocký, G. Baudoin, and G. Chollet. *Computational models of Speech pattern processing*, chapter Towards a very low bit rate segmental speech coder. Springer-Verlag, Jersey, Great Britain, nato-asi edition, 1997.

[11] G. Baudoin, P. Jardin, J. Gross, and G. Chollet. *Speech Recognition and Coding, new advances and trends*, chapter Comparison of parametric spectral representations for voice recognition in noisy environments, pages 313–316. Springer-Verlag, nato asi serie f., edited by a. rubio & jm lopez edition, 1995.

[12] M. Jelinek and G. Baudoin. *Speech Recognition and Coding, new advances and trends*, chapter Excitation construction for the robust low bit rate CELP speech coder, pages 439–443. Springer-Verlag, nato asi serie f., edited by a. rubio & jm lopez edition, 1995.

Proceedings de conférences internationales avec actes et comité de lecture

[13] G. Baudoin, P.Jardin. A new adaptive baseband pre-distortion algorithm for linearization of power amplifiers, application to EDGE-GSM transmitters. To appear in *Proceedings of CSCC, Int. Conf. on Circuits Systems and Communications*, Greece, July 2000.

[14] J. Černocký, G. Baudoin, I. Kopecek, and G. Chollet. *Lecture notes in artificial intelligence LNCS 1692, 2nd international workshop TSD'99*, chapter Very low bit rate speech coding : comparizon of data driven units with syllables segments, pages pp 262–267. Springer-Verlag, Mariánské Lázne, czech republic, 1999.

[15] J. Černocký, G. Baudoin, D. Petrovska-Delacretaz, J. Hennebert, and G. Chollet. *lectures notes in artificial intelligence, 1rst international workshop TSD'98*, chapter Automatically derived speech units : application to very low rate coding and speaker verification, pages 183–188. Springer-Verlag, Brno, czech republic, 1997.

[16] J. Černocký, G. Baudoin, and G. Chollet. The use of alisp for automatic acoustic-phonetic transcription. In *Proc. of ESCA SPoSS workshop, workshop on sounds patterns of Spontaneous Speech*, Aix en Provence, France, 1998.

[17] J. Černocký, G. Baudoin, and G. Chollet. Segmental vocoder-going beyong the phonetic approach. In *Proc. IEEE ICASSP'98*, pages 605–608, Seattle USA, 1998.

[18] G. Baudoin, J. Černocký, and G. Chollet. Quantization of spectral sequences using variable length spectral segments for speech coding at very low bit. In *Proc. of Eurospeech 97*, pages 1295–1298, Rhodos, Greece, September 1997.

[19] J. Černocký, G. Baudoin, and G. Chollet. speech spectrum representation and coding using multigrams with distance. In *Proc. IEEE ICASSP'97*, pages 1343–1346, Munchen, Germany, April 1997.

[20] G.Baudoin and I. Stylianou. On the transformation of the speech spectrum for voice conversion. In *Proc. of ICSLP'96*, pages 1404–1408, Philadelpia, USA, October 1996.

[21] J. Černocký, G. Baudoin, and G. Chollet. Efficient method of speech spectrum description using multigrams. In *proc of 3rd slovenian-german workshop Speech and image understanding 3rd Sloveniain and 2nd SDRV Workshop*, pages 139–148, Ljubljana, Slovenia, 1996.

[22] B. Boianov and G. Baudoin. Stress detection through voice analysis. In *proc of 3rd slovenian-german workshop Speech and image understanding 3^{rd} sloveniain and 2^{nd} SDRV Workshop*, pages 149–156, Ljubljana, Slovenia, 1996.

[23] B. Boianov, S Hadjitodorov, and G. Baudoin. Acoustical analysis of pathological voices. In *proc of 3rd slovenian-german workshop Speech and image understanding 3rd Sloveniain and 2nd SDRV Workshop*, pages 157–166, Ljubljana, Slovenia, 1996.

[24] M. Mauc, G. Baudoin, and M. Jelinek. Complexity reduction for the fs1016 coder with multistagew search. In *Proc. IEEE ICASSP'94*, pages I-261–I-264, Adelaide, Australia, April 1994.

[25] M. Mauc, G. Baudoin, and M. Jelinek. Complexity reduction for the fs1016 at 4800 bps celp coder. In *Proc. of Eurospeech'93*, pages I-245–I-248, Berlin, Germany, 1993.

[26] M. Mauc and G. Baudoin. Reduced complexity celp coder. In *Proc. IEEE ICASSP'92*, pages I-53–I-56, San Francisco, USA, 1992.

[27] J. Liu, G. Baudoin, and G. Chollet. Studies of glottal excitation and vocal tract parameters using inverse filtering and a parametrised input model. In *Proc. ICSLP'92*, pages 1051–1054, 1992.

[28] M. Mauc, G. Baudoin, M. Jelinek, and P. Jardin. Reduced complexity celp coder with a multistage search. In *Proc. Eusipco'92*, pages 523–526, Brussels, Belgium, 1992.

[29] G. Baudoin and M. Chaouche. A portable digital system for recording and processing of ecg. In *Proc. Eusipco'88*, pages 1275–1278, Grenoble, France, 1988.

[30] G. Baudoin and M. Chaouche. A portable system for digital recording of electrocardiogram. In *Proc. IEEE conf. on bioeng. and med. Physic*, page 172, San Antonio, USA, 1988.

[31] M. Chaouche and G. Baudoin. A digital ECG recording system. In *Proc. 1rst Mediterranean conf. Biomedical Engineering*, pages 149–152, Sevilla, Spain, 1986.

[32] A. Marguinaud, G. Baudoin, A. Roseiro, and S. Bruel. Soft decoding in presence of chanel erasures. In *Proc. of IEEE international Conference on Information Theory*, Paris, France, 1986.

Proceedings de conférences nationales avec actes et comité de lecture

[33] G. Baudoin, R. Marsalek, and J. Prokes. Evaluation of the potential of the VLIW digital signal processor TMS320C6201 for UMTS FDD standard baseband processing implementation. In *proceedings ERK'99*, pages 113–116, Portoroz, Slovenia, 1999.

[34] G. Baudoin, J. Černocký, and G. Chollet. Quantification de séquences spectrales de longueurs variables pour le codage de parole à très bas débit. In *Proc. GRETSI'97*, pages 1093–1096, Grenoble, France, September 1997.

[35] F. Friderich, G. Baudoin, and Y. Tasic. Contour detection for image segmentation. In *Proc. conference ERK'97*, pages 253–256, Portoroz, Slovenia,, 1997.

[36] J. Černocký and G. Baudoin. Représentation du spectre de parole par les multigrammes. In *Proc. XXI-es Journées d'Etude sur la Parole*, pages 239–242, Avignon, France, June 1996.

[37] J. Černocký and G. Baudoin. Speech spectrum coding using multigram segmentation. In *proc. conf Radioelektronika*, pages 140–143, Brno, czech republic, 1996.

[38] G. Baudoin, P. Jardin, G. chollet, and J. Gross. Comparaison de techniques de paramètrisation spectrale pour la reconnaissance vocale en milieu bruité. In *Actes du quinzième colloque GRETSI*, pages 783–786, Juan les pins, France, 1993.

[39] G. Baudoin. Convergence d'algorithmes de type LMS pour l'annulation des échos sur les images de télévision. In *Actes du* $14^{ème}$ *colloque GRETSI*, pages 521–524, Juan les pins, France, 1991.

[40] G. Baudoin, M. Chaouche. Holter Numérique, un système portable pour l'enregistrement de l'électrocardiogramme. In *Actes du colloque GRETSI*, pages 635–637, Nice, France, 1987.

[41] M. Mauc and G. Baudoin. codeur celp à complexité réduite. *journal de physique IV, colloque C1, supplément du journal de physique III*, 2 :C1-327–C1-330, Apr. 1992.

Proceedings des conférences Texas-Instruments

[42] G. Baudoin, O. Venard. Digital Signal Processing for a vibrating magnetic excitation gyrometer, implementation on a DSP TMS320F243. To appear in *Proc. of* 3^{nd} *European Conference on DSP Research and education*, Noisy Le Grand, FRANCE, 2000.

[43] G. Baudoin, O. Venard. Implementation of FIR filters on fixed point DSP for communication systems. In *Proc. of* 2^{nd} *European Conference on DSP Research and education*, pages 365–371, Noisy Le Grand, FRANCE, 1998.

[44] D. Janu, G.Baudoin, J.-F. Bercher, and O. Venard. Design of a cdma system simulator and implementation on a TMS320C6201. In *Proc. of* 2^{nd} *European Conference on DSP Research and education*, pages 119–124, Noisy Le Grand, FRANCE, 1998.

[45] G.Baudoin and P. Blaha. Development of a low bit rate speech coder on a TMS320C30 based on the half rate gsm standard. In *Proc. of* 1^{rst} *European Conference on DSP Research and education*, pages 11–22, Noisy Le Grand, FRANCE, 1996.

[46] G.Baudoin, F. Virolleau, O. Venard, and P. Jardin. Teaching dsp through the case study of a fsk modem. In *Proc. of* 1^{rst} *European Conference on DSP Research and Education*, pages 260–263, Noisy Le Grand, FRANCE, September 1996.

[47] G.Baudoin and F. Virolleau. Traitement du signal et DSP. In *Proc. of DSPconf Texas-Instruments*, Noisy Le Grand, FRANCE, June 1995.

Rapport des contrats de recherche et développement

[48] G. Baudoin and O. Venard. Rapport de recherche projet isnav, génération et traitement des signaux d'un gyromètre magnétique vibrant. Technical report, ESIEE, 1999.

[49] G. Baudoin, C. Ripoll and P. Bildstein. Rapport d'étude (PCT ANVAR) pour la société STID, sur les systèmes d'identification sans contact à 13,56 MHz. Technical report, ESIEE, 1999.

[50] G. Baudoin and A. Zemva. Rapport projet proteus, segmentation des signaux et des images, algorithmes et implantation multiprocesseur. Technical report, ESIEE and University of Ljubljana, 1997 and 1998.

[51] G. Baudoin, I. Stylianou, and P. Jardin. Rapports de recherche projet CNET, transformation de la voix, création de voix nouvelles et application à la synthèse de parole à partir du texte. Technical report, ESIEE, 1995 and 1996 and 1997.

[52] G. Baudoin. Rapport du projet Elite Texas-Instruments, enrgistreur vocal statique. Technical report, ESIEE, 1997.

[53] G. Baudoin. Design, simulation and applications of multirate filter banks, proc. final meeting tempus jep 1326 project, varsovie jul. 1994. Technical report, ESIEE, 1993 and 1994.

[54] G. Baudoin and M. Jelinek. Rapport du contrat de recherche acsys, codeur de parole celp à bas débit. Technical report, ESIEE, 1993.

[55] G. Baudoin and P. Jardin. Rapport du contrat de R&D secmat, codeur de parole à 2400 bps. Technical report, ESIEE, 1990.

Deuxième partie

Codage de la parole à bas et très bas débit
Transformation de la voix

Introduction

J’ai choisi de ne décrire dans cette partie du document que mes travaux dans les domaines du codage de la parole et de la transformation de voix, ces thèmes représentant l’essentiel de mon activité de recherche dans les dernières années.

Cet exposé est divisé en 2 chapitres :

- Le codage de la parole à bas et très bas débit,
- la transformation de la voix.

Les travaux présentés ont pour la plupart été réalisés en collaboration avec des étudiants en thèse, en projets de DEA ou de fin d’études.

Notations et sigles utilisés

- Les vecteurs sont représentés par des lettres minuscules en caractères gras.
- Les matrices sont représentées par des lettres majuscules en caractères gras.
- $\mathbf{M}^T$ représente la matrice $\mathbf{M}$ transposée.
- Le produit scalaire de 2 vecteurs $\mathbf{u}$ et $\mathbf{v}$ est noté $< \mathbf{u}, \mathbf{v} >$.
  Pour des vecteurs de longueur N et de coordonnées u_i, v_i :
  $$< \mathbf{u}, \mathbf{v} >= \sum_{i=0}^{N-1} u_i v_i.$$
- Le symbole $*$ représente un produit de convolution.
  Par exemple $y(n) = x(n) * h(n) = \sum_{k=0}^{+\infty} h(k)x(n-k)$.
- CELP = Code Excited Linear Prediction
- DT = Décomposition Temporelle
- DTW = Dynamic Time Warping
- HSX = Harmonic Stochastic Coding
- HVXC = Harmonic Vector eXcitation Coding
- HNM = Harmonic plus Noise Model
- HMM = Hidden Markov Model
- LBG = algorithme de Lindo-Buzzo-Gray pour la construction d'un dictionnaire de quantification vectorielle
- LPC = Linear Predictive Coding
- LSP = Line Spectrum Pair
- MBE = MultiBand Excitation coding
- MELP = Mixed Excitation Linear Prediction
- MFCC = Mel Frequency Cepstrum Coefficient
- STC = Sinusoïdal Transform Coding
- WI = Waveform Interpolation coding

CHAPITRE I

LE CODAGE DE LA PAROLE À BAS ET TRÈS BAS DÉBIT

CE chapitre présente mes activités de R&D dans le domaine du codage de la parole à bas et très bas débit pour des applications de stockage ou de transmission.

Le domaine a connu d'importants développements ces dernières années. L'accroissement des communications radio-mobiles a conduit à transmettre le signal vocal sous forme numérique pour améliorer la robustesse au bruit du canal. Pour économiser la bande spectrale occupée différents algorithmes de codage ont été proposés et standardisés.

Je me suis intéressée au codage de parole à bas et très bas débits, typiquement inférieurs à 4800 bps[1], plus particulièrement aux 3 points suivants :
- Réduction de la complexité des codeurs CELP,
- Étude de la limitation en débit des codeurs CELP,
- Développement d'un nouveau principe de codage à très bas débit (inférieur à 600 bps).

J'ai par ailleurs implanté plusieurs codeurs de parole sur DSP, dans le cadre de contrats industriels.

Ce chapitre comprend 5 sections. La section 1 est un état de l'art sur les codeurs à bas et très bas débits. Les 4 sections suivantes sont consacrées à mon travail.
- La section 2 décrit les travaux effectués sur la réduction de complexité des codeurs CELP.
- La section 3 présente la recherche sur la limitation en débit des codeurs CELP.
- La section 4 développe les études sur le codage à très bas débit.
- La section 5, très courte, résume mes travaux d'implantations de codeurs sur DSP.

Les sections 2, 3 et 4 sont ordonnées de manière chronologique, les travaux sur le codage à très bas débit, présentés ici, se continuent dans le cadre d'un projet RNRT.

1 État de l'art

1.1 Généralités

Dans les systèmes de téléphonie filaire classiques, la parole est numérisée à 64 Kbps. De nombreux algorithmes [120, 72] ont été proposés pour diminuer ce débit tout en essayant de conserver

[1] Les sigles bps et Kbps signifient respectivement bits par seconde et kilobits par seconde.

une qualité subjective donnée fonction des exigences de l'application à laquelle le codeur est destiné. On distingue en général 3 plages de débits :

- Les hauts débits, supérieurs à 16 Kbps, correspondant à des algorithmes de codage de la forme d'onde non spécifiques à la parole,
- Les débits moyens, de 4 Kbps à 16 Kbps, correspondant à des techniques de codage hybrides utilisant des méthodes de codage de la forme d'onde et prenant en compte certaines propriétés de la parole ou de la perception auditive[2]. Le principal représentant de cette classe est le codage CELP [116].
- Les bas et très bas débits, de quelques dizaines de bits par seconde à 4 Kbps, correspondant aux vocodeurs (VOice CODER) spécifiques au codage de la parole.

Un système de codage de la parole comprend 2 parties : le codeur et le décodeur. Le codeur analyse le signal pour en extraire un nombre réduit de paramètres pertinents qui sont représentés par un nombre restreint de bits pour archivage ou transmission. Le décodeur utilise ces paramètres pour reconstruire un signal de parole synthétique.

La plupart des algorithmes de codage mettent à profit un modèle linéaire simple de production de la parole. Ce modèle sépare la source d'excitation, qui peut être quasi-périodique pour les sons voisés ou de type bruit pour les sons fricatifs ou plosifs, du canal vocal qui est considéré comme un résonateur acoustique. La forme du conduit vocal détermine ses fréquences de résonance et l'enveloppe spectrale (formants) du signal de parole.

Le signal de parole est souvent modélisé (modèle « source-filtre ») comme la sortie d'un filtre tout pôle (appelé filtre de synthèse) dont la fonction de transfert représente l'enveloppe spectrale, excité par une entrée dont les caractéristiques (en particulier la fréquence fondamentale[3]) déterminent la structure fine du spectre.

Le signal de parole n'étant pas stationnaire, les codeurs le découpent généralement en trames quasi-stationnaires de durée comprise entre 5 et 30 ms. Sur chaque trame, le codeur extrait des paramètres représentant l'enveloppe spectrale et caractérise ou modélise l'excitation de manière plus ou moins fine soit par quantification vectorielle, soit à l'aide de paramètres tels que l'énergie, le voisement et la fréquence fondamentale F_0. D'autres paramètres peuvent être calculés pour représenter plus finement l'excitation. Les paramètres les plus souvent utilisés pour l'enveloppe spectrale sont les paires de raies spectrales ou LSF « *Line Spectral Frequencies* ») qui sont déduites des coefficients de prédiction linéaire et qui possèdent de bonnes propriétés pour la quantification et l'interpolation.

De nombreux algorithmes de codage à moyen débit ont été normalisés au cours des 10 dernières années pour les systèmes de communications avec les mobiles, GSM plein débit (ou *Full Rate GSM* et demi-débit (ou *Half Rate GSM*), GSM plein débit amélioré (ou *Enhanced Full Rate GSM*), IS95 par exemple. La numérisation de la parole permet une meilleure protection contre les distorsions et les bruits introduits par les canaux radio-mobiles. Une diminution du débit en dessous de 4 Kbps, à condition de conserver une qualité de type téléphonique permettra d'augmenter la capacité des réseaux de communication avec les mobiles.

Les autres applications des codeurs à bas ou très bas débits incluent l'amélioration des systèmes de téléphonie sécurisés par cryptage, la radio-messagerie vocale, la téléphonie sur Internet, les répondeurs vocaux, les communications sur le canal HF, les communications personnelles par satellites à faible coût, et les bas débits des communications à débit adaptatif où le codeur de source et le codeur

[2]Certains codeurs à haut débit utilisent aussi les propriétés de la perception auditive.

[3]On utilise (par abus de langage) les expressions fréquence fondamentale et pitch indifféremment dans ce document.

de canal s'adaptent à la qualité du canal et à la nature de la source.

L'évaluation des codeurs à bas et très bas débits ne peut pas se faire par des critères objectifs de rapport signal à bruit. Le signal décodé doit être perçu comme proche de l'original, mais les formes d'onde peuvent être très différentes. On évalue ces codeurs par des tests subjectifs, tels que le test ACR (*Absolute Category Rating*) délivrant un score MOS (*Mean Opinion Score*) ou le test d'acceptabilité DAM (*Diagnostic Acceptability Measure*) pour la qualité, et le test de rimes DRT (*Diagnostic Rhyme Test* [1]) pour l'intelligibilité. Ces tests sont menés sous certaines conditions de bruit ambiant ou de taux d'erreurs canal. Pour qualifier la qualité d'un codeur, on utilise les termes anglais : « *broadcast* », « *toll* », « *telecommunication* », « *synthetic* ». Une qualité de type « *broadcast* » correspond à un codage large bande (audioconférence par exemple), la qualité de type « *toll* » est celle du téléphone analogique filaire. Pour une qualité de type « *telecommunication* », l'intelligibilité et le naturel sont conservés mais quelques distorsions sont audibles. Un codeur de qualité « *synthetic* » est intelligible mais le signal manque de naturel.

La limite théorique minimum de débit pour un codage conservant l'information sémantique contenue dans la parole est de l'ordre de 60 bps, si l'on compte environ 60 phones dans une langue et une vitesse d'élocution moyenne d'une dizaine de phones par seconde. Pour un débit aussi faible, les informations concernant le locuteur et ses émotions sont perdues.

Cet état de l'art s'intéresse à la catégorie de codeurs à bas et très bas débits. Il comprend cette introduction puis une section sur les codeurs à bas débit et une section sur les codeurs à très bas débit.

1.2 Les codeurs de parole à bas débit

Pour les bas débits, typiquement de 800 bps à 4800 bps, les techniques de codage de la forme d'onde ne donnent pas de bons résultats. Les codeurs doivent éliminer les informations sans pertinence pour la perception. Les vocodeurs utilisent certaines caractéristiques de la perception et de la production de la parole, aussi sont-ils généralement très peu efficaces pour les signaux autres que la parole comme les signaux DTMF[4] de numérotation téléphonique ou le bruit ambiant.

Cette section présente d'abord succinctement les codeurs CELP, puis les vocodeurs classiques à 2 états d'excitation et enfin les nouveaux algorithmes de codage à bas débit.

1.2.1 Présentation du codage CELP et de ses limitations pour le codage à bas débit

Le codage CELP (Code Excited Linear Prediction) a été introduit par Schroeder et Atal [116]. Il est très efficace pour les débits moyens de 4,8 Kbps à 16 Kbps, en témoignent les nombreuses normes qui l'utilisent. La figure I.1 représente le principe du codage CELP.

Dans chaque trame, une analyse spectrale par prédiction linéaire court terme permet d'estimer l'enveloppe spectrale et détermine le filtre de synthèse $1/A(z)$.

On découpe chaque trame en sous-trames plus courtes (durée typique 5 ms). On modélise la périodicité de l'erreur de prédiction court terme (résiduel) à l'aide d'un prédicteur linéaire long terme représenté par un filtre $B(z) = 1 - bz^{-Q}$, où Q est une estimation de la période fondamentale. Sur chaque sous-trame on effectue une quantification vectorielle du signal par une technique d'analyse par synthèse. La quantification vectorielle utilise un dictionnaire de $M = 2^k$ séquences de bruit blanc

[4]DTMF = Dual Tone Multi-Frequency.

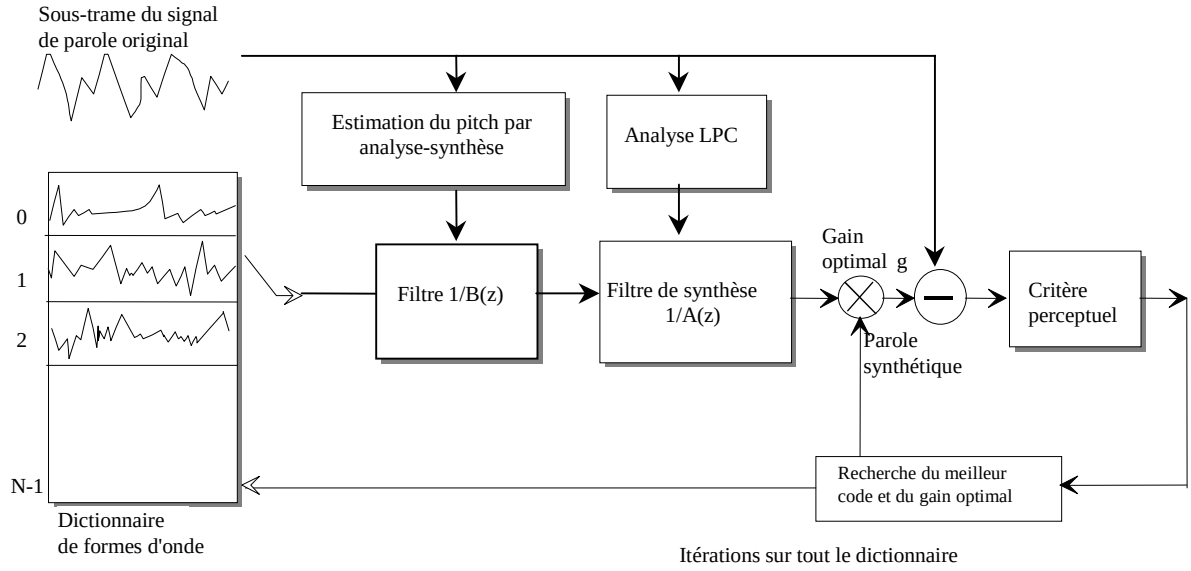


FIG. I.1 – Principe du codage CELP

normalisées en énergie. La longueur de ces séquences est égale à une sous-trame. Chaque séquence du dictionnaire est filtrée par le filtre de synthèse $1/(A(z)B(z))$ et multipliée par un gain. La sortie obtenue est le signal de parole synthétique qui est comparé au signal original. Le codeur teste toutes les séquences du dictionnaire, calcule le gain optimum pour chacune et retient celle qui minimise un critère « perceptuel[5] » de comparaison entre le signal synthétique et le signal original. Le codeur transmet l'indice de la séquence qui minimise le critère (sur k bits) ainsi que le gain associé, les paramètres spectraux et le pitch. Le critère « perceptuel » est un critère de moindres carrés calculé sur la différence entre le signal original et le signal synthétique après filtrage de cette différence par un filtre de pondération de type $A(z)/A(z/\gamma)$ où γ est compris entre 0 et 1 (typiquement $\gamma = 0.85$). Ce filtre pondère l'erreur dans le domaine fréquentiel, il atténue l'erreur dans les zones où l'amplitude de $1/|A(f)|$ est importante (zones de formants) et amplifie l'erreur dans les zones de faible amplitude de $1/|A(f)|$. Il met ainsi à profit les propriétés de masquage des bruits par les zones de fortes amplitudes du spectre, d'où le nom de critère « perceptuel ».

En pratique, pour diminuer la complexité du codeur, on remplace le filtre $1/B(z)$ par un dictionnaire qui contient les séquences de résiduel précédentes. Ce dictionnaire est appelé adaptatif, sa sortie est ajoutée à la sortie du dictionnaire de bruit blanc qui est appelé dictionnaire stochastique. Certains codeurs utilisent plusieurs dictionnaires stochastiques et forment le signal synthétique en ajoutant les sorties des différents dictionnaires.

Quelques tentatives ont été faites pour diminuer les débits obtenus avec les codeurs CELP [74]. Mais en dessous de 3 Kbps la méthode est inférieure aux approches de type vocodeurs.

La qualité subjective des codeurs CELP décroît rapidement lorsque le débit descend en dessous de 4 Kbps. En effet, le codage CELP effectue essentiellement une quantification vectorielle de la forme d'onde et pour un débit trop faible il n'est pas possible de coder cette forme précisément.

Pour les sons voisés, le signal synthétique présente parfois des harmoniques de F_0 jusqu'à $f_e/2$ même si le signal original n'a plus d'harmoniques au-delà d'une fréquence f_{max}. On parle dans ce cas

[5]On utilise le néologisme « perceptuel » pour indiquer un critère ou un filtre essayant de tenir compte de la perception auditive.

d'artefact tonal. La figure I.2 illustre ce phénomène pour un signal codé par un codeur CELP GSM demi-débit à 5600 bps.

D'une manière générale la partie hautes fréquences du spectre est mal représentée car malgré le filtre de pondération, son amplitude est très faible par rapport à la partie basses fréquences qui est de ce fait favorisée par le critère des moindres carrés.

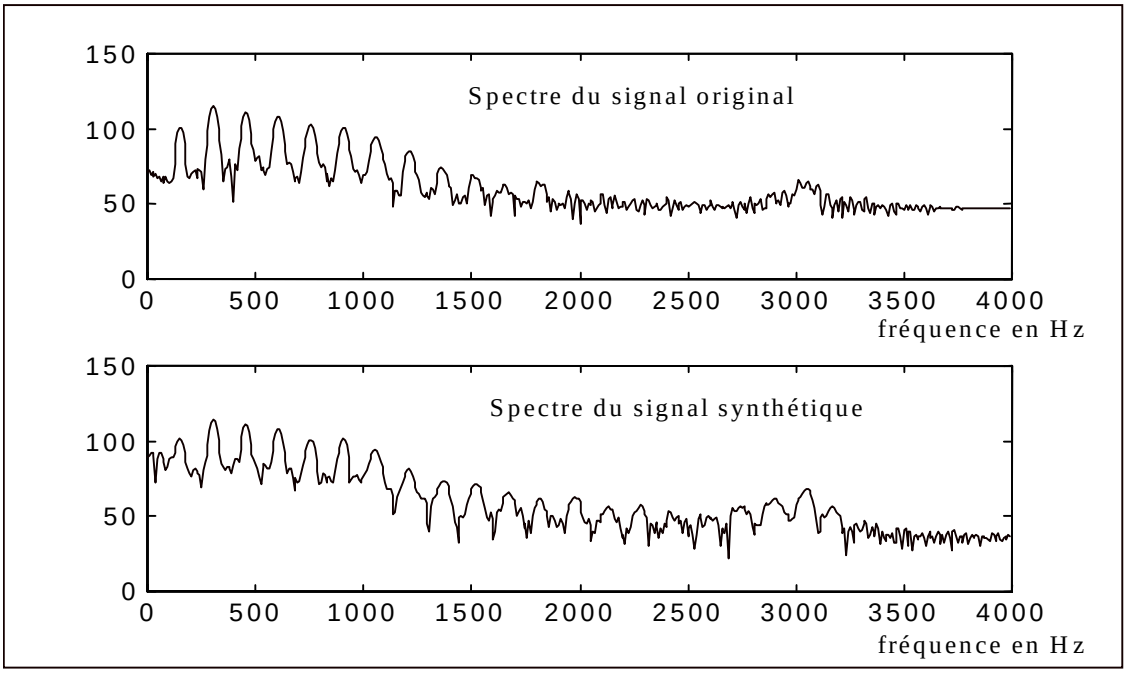


FIG. I.2 – Artefacts tonals introduits par un codage CELP à 5600 bps

1.2.2 Les vocodeurs classiques à 2 états d'excitation

Dans les vocodeurs classiques, vocodeurs à canaux, vocodeurs à formants, ou vocodeurs LPC, les différentes trames de signal sont classées en trames voisées (V) et trames non-voisées (NV). Ces vocodeurs utilisent le modèle « source-filtre ». La synthèse du signal décodé utilise un signal d'excitation reconstruit formé d'un bruit blanc pour les trames non-voisées et d'un train périodique d'impulsions à la fréquence F_0 pour les trames voisées. La figure I.3 représente le synthétiseur d'un vocodeur à 2 états d'excitation. Ces vocodeurs diffèrent essentiellement dans leur façon d'estimer et d'appliquer l'enveloppe spectrale.

Dans les vocodeurs à canaux introduits par Dudley en 1939 [61], le codeur évalue l'énergie, le voisement, F_0, et les puissances relatives du signal dans un ensemble de bandes de fréquences adjacentes (de l'ordre de 10 bandes). Le décodeur génère la parole synthétique en passant le signal d'excitation dans un banc de filtres passe-bande dont les sorties sont pondérées par les puissances relatives du signal original dans ces différentes bandes. Les sorties des filtres sont ensuite ajoutées et cette somme est mise à l'échelle en fonction de l'énergie de la trame originale. Ces codeurs ont été utilisés jusqu'à des débits de 400 bps.

Dans les vocodeurs à formants [61], le codeur détermine la position, l'amplitude et la largeur de bande des 3 premiers formants, ainsi que l'énergie de la trame, le voisement et F_0. Au décodeur, l'excitation synthétique est filtrée par 3 filtres accordés sur les formants. Le signal résultant est mis à l'échelle en fonction de l'énergie de la trame. On obtient avec cette technique un signal intelligible pour des

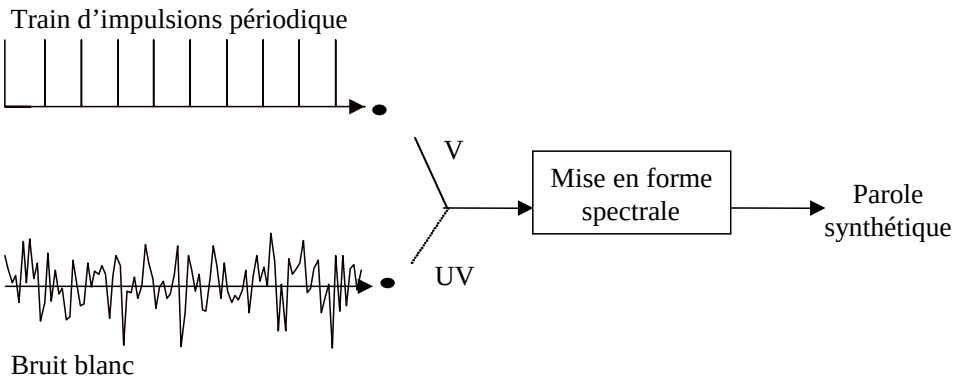


FIG. I.3 – Synthèse dans un vocodeur à 2 états d'excitation

débits de 1200 bps, mais la détermination des formants est une tâche difficile et peu robuste.
Dans les vocodeurs à prédiction linéaire LPC (Linear Predictive Coding) [7, 128], l'enveloppe spectrale du signal de parole est modélisée par l'amplitude de la fonction de transfert d'un filtre tout pôle $1/A(z)$. Les coefficients a_i du filtre sont obtenus par prédiction linéaire. Le signal de parole x_n est prédit par une combinaison linéaire $\hat{x}_n$ des échantillons précédents :

$$\hat{x}_n = -\sum_{i=1}^{p} a_i x_{n-i}.$$

L'enveloppe spectrale est très sensible à la quantification des coefficients a_i. De plus l'interpolation de ces coefficients peut conduire à des filtres de synthèse instables. Aussi les transforme-t-on souvent en un autre jeu de coefficients pour la quantification et la transmission. Les coefficients classiques sont les logarithmes de rapports d'aires (*Log Area Ratio* ou LAR), les coefficients de réflexion (ou k_i), et les paires de raies spectrales (*Line Spectrum Frequencies* ou LSF). Le nombre de coefficients a_i est compris entre 8 et 16 pour une fréquence d'échantillonnage de 8 KHz, de façon à ce que la fonction de transfert du filtre présente un nombre suffisant de résonances pour modéliser correctement les 3 à 5 premiers formants. En plus des coefficients déduits des coefficients LPC, le codeur transmet l'énergie, le voisement et la fréquence fondamentale de la trame. Le décodeur génère le signal synthétique en filtrant l'excitation reconstruite par le filtre de synthèse $1/A(z)$ et en mettant à l'échelle la sortie en fonction de l'énergie de la trame.

Les codeurs LPC à 2 états ont été développés pour des débits d'environ 2400 bps. Des débits de 600 à 800 bps ont été atteints en appliquant une quantification vectorielle aux coefficients spectraux [110, 76, 60, 75].

Le codage LPC à 2400 bps a été normalisé par L'OTAN (*Voice coding standard* STANAG 4198 [107]), et le département de la défense américain DOD (Federal Standard 1015 [128]). Plus récemment l'OTAN a normalisé un codeur LPC à 800 bps pour les communications HF [105].

Dans ces 3 codeurs, l'excitation est représentée de manière trop succincte. Pour un codeur à 2400 bps, environ 1850 bps sont dédiés à l'enveloppe spectrale et seulement 550 bps à l'excitation.

La classification de l'excitation en 2 classes (V ou NV) n'est pas adaptée aux sons mixtes comme les fricatives voisées. Elle ne peut pas représenter les sons qui présentent un spectre harmonique

jusqu'à une fréquence f_{max} puis une structure de bruit au-delà de f_{max}. Les sons plosifs ne sont pas correctement modélisés à l'aide d'un bruit blanc dont l'énergie répartie sur la trame.

Pour ces différentes raisons, le signal synthétique manque de clarté, est perçu comme bruité et présente des artefacts tonals. De plus si la classification V/NV est erronée ou si F_0 est mal estimée, la qualité du signal synthétique est fortement dégradée. Les défauts les plus audibles se produisent sur les zones voisées ou aux transitions. Ils sont essentiellement dus à une mauvaise représentation de l'évolution des paramètres de voisement.

1.2.3 Les nouveaux algorithmes de codage à bas débit

Dans les 10 dernières années, plusieurs algorithmes ont été proposés qui permettent un codage à bas débit avec une qualité de type communication (MOS autour de 3.5). Ces nouveaux algorithmes ont en commun une meilleure représentation des parties voisées du signal et de l'évolution des paramètres de voisement aux transitions entre sons. La plupart du temps, les paramètres spectraux sont codés par quantification vectorielle [64, 83] sans distorsion audible pour un débit de 1500 bps. Une pondération perceptuelle peut-être appliquée autour des formants, les paramètres LSF se prêtant bien à ce type de pondération.

Parmi les nouvelles méthodes de codage à bas débit, on peut distinguer les algorithmes de type codeurs harmoniques (MBE[6], STC[7]), les algorithmes à interpolation de forme d'onde (WI[8]) et les algorithmes à excitation mixte (MELP[9], HSX[10]).

La complexité de ces nouvelles approches est nettement supérieure à celle des codeurs LPC classiques, mais il est possible de les implanter sur un seul DSP en virgule fixe.

1.2.3.1 Les codeurs à modèles sinusoïdaux ou STC (Sinusoïdal Transform Coders) Les codeurs STC (McAulay et Quatieri [98, 95, 96, 97]) modélisent la parole par une somme de sinusoïdes dont les amplitudes, les fréquences et les phases évoluent au cours du temps. Pour les parties voisées, les fréquences sont reliées aux harmoniques de F_0 et évoluent lentement au cours du temps. Les pics de la transformée de Fourier à court terme peuvent être utilisés pour déterminer les paramètres des sinusoïdes. Le nombre de sinusoïdes dans le modèle est variable car il dépend de F_0. Il a donc fallu développer des techniques de quantification vectorielle de vecteurs de longueur variable.

Pour les sons non-voisés, l'excitation est un bruit blanc obtenu par une somme de sinusoïdes dont les fréquences sont uniformément réparties entre 0 et $f_e/2$. Différents modèles d'évolution de la phase ont été proposés [97].

Le codage STC donne de très bons résultats pour les débits moyens et pour la plage supérieure des bas débits. Un codeur sinusoïdal multi-débit a été développé aux MIT Lincoln Labs [96] avec des débits de 1.8 à 8 Kbps. Pour les débits les plus faibles, les informations de phase ne sont pas transmises.

1.2.3.2 Les codeurs à excitation multibande ou Multi-Band Excited Coders (MBE) Dans les codeurs MBE [67] et leurs variantes IMBE (Improved MBE) ou AMBE (Advanced MBE), le signal

[6] MBE = Multiband Excited Coder.
[7] STC = Sinusoidal Transform Coder.
[8] WI = Waveform Interpolation.
[9] MELP = Mixed Excitation Linear Prediction.
[10] HSX = Harmonic Stochastic Coder.

est analysé dans plusieurs bandes de fréquence adjacentes et est déclaré voisé ou non-voisé dans chacune des bandes. Le nombre de bandes d'analyse est de l'ordre du nombre d'harmoniques de F_0 entre 0 et $f_e/2$. La figure I.4 représente l'amplitude de la transformée de Fourier discrète d'une trame de signal de parole avec ses zones harmoniques (voisées) ou non harmoniques (non-voisées) représentées par les signes V et UV. L'enveloppe spectrale $H(f)$ et la structure fine $E(f)$ de la transformée

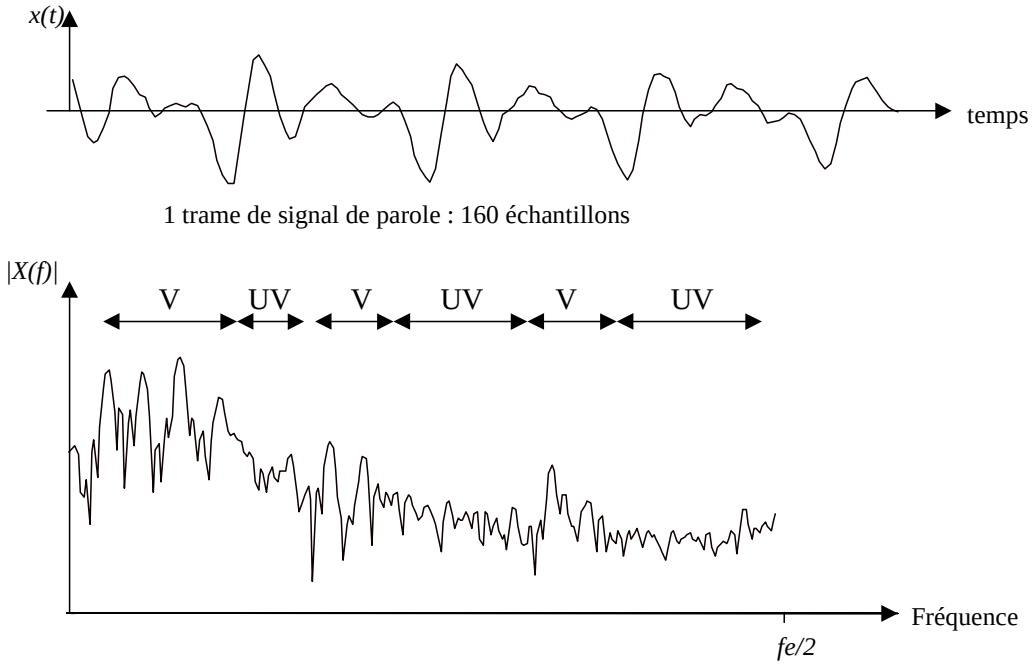


FIG. I.4 – Zones voisées (V) ou non-voisées (UV) du spectre d'une trame de parole

de Fourier discrète à court terme $X(f)$ de la trame de signal sont approchées séparément par $\hat{H}(f)$ et $\hat{E}(f)$. Le signal synthétique $\hat{x}_n$ est obtenu dans le domaine fréquentiel par

$$\hat{X}(f) = \hat{E}(f)\hat{H}(f).$$

Les paramètres transmis par le codeur sont : la fréquence fondamentale, l'information de voisement pour chaque bande, et les paramètres décrivant l'enveloppe spectrale. Pour les bas débits, le voisement est estimé par groupe de quelques harmoniques.

L'algorithme IMBE (ou Improved MBE coding) a été normalisé à 4150 bps pour le système Inmarsat-M avec 2250 bps pour la correction d'erreur, d'où un débit total de 6400 bps.

1.2.3.3 WI prototype Waveform Interpolation coders Dans les codeurs WI à interpolation de formes d'onde (WI = Waveform Interpolation coders) [77, 79, 78, 119], les paramètres spectraux correspondent aux coefficients de prédiction linéaire. La fréquence fondamentale est estimée et le résiduel de prédiction est calculé par filtrage du signal de parole par $A(z)$. Puis une forme d'onde caractéristique (CW = Characteristic Waveform) est extraite du signal résiduel à intervalles réguliers (typiquement à un rythme de 480 Hz). Cette extraction se fait en plaçant des marqueurs de pitch par détection de pics sur le signal résiduel suréchantillonné. Pour les sons voisés, la longueur des CW correspond à une période de pitch $p(t_m)$ à l'instant de calcul t_m.

La figure I.5 représente le signal résiduel d'une trame voisée de 20 ms et les CW correspondantes calculées toutes les 2,5 ms. Pour les sons non voisés, la longueur des CW est arbitraire. La longueur

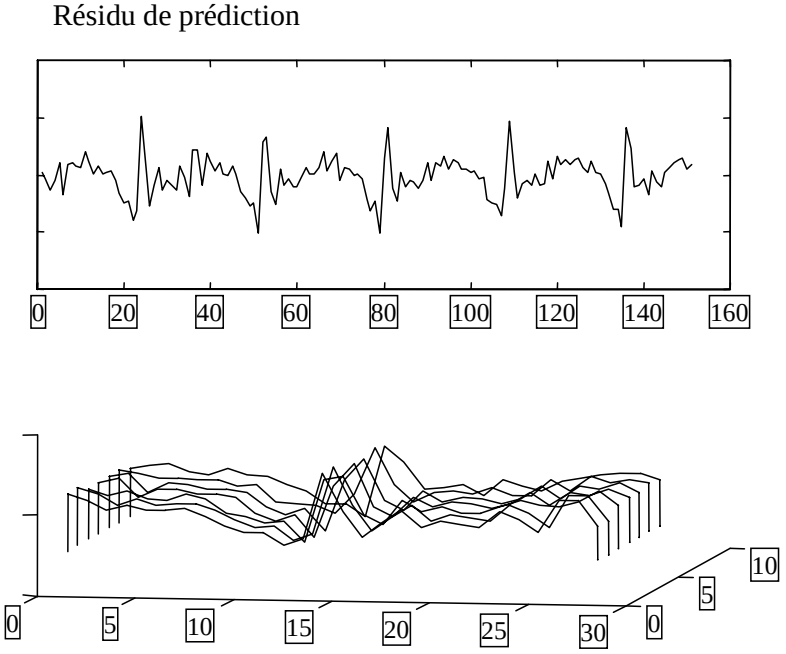


FIG. I.5 – Trame de résiduel avec les formes d'ondes caractéristiques CW correspondantes extraites toutes les 2.5 ms

de l'onde caractéristique $z(t_m, \tau)$ calculée à l'instant t_m est normalisée à 2π par la relation

$$u(t_m, \tau) = z\left(t_m, \frac{p(t_m)}{2\pi}\tau\right),$$

puis alignée en temps avec l'onde précédente $u(t_{m-1}, \tau)$. À chaque instant t est associé un signal périodique $u(t, \tau)$ de période 2π, représenté par les coefficients de sa série de Fourier. Ce signal est obtenu par interpolation linéaire (sur les coefficients de Fourier) entre 2 CW successives aux instants t_m et t_{m+1}. L'équation I.1 donne la formule d'interpolation.

$$u(t, \tau) = (1 - \alpha(t))u(t_m, \tau) + \alpha(t)u(t_{m+1}, \tau) \tag{I.1}$$

Dans l'équation I.1, $\alpha(t)$ est une fonction monotone croissante avec $\alpha(t_m) = 0$ et $\alpha(t_{m+1}) = 1$.

La longueur dénormalisée d'une période de ce signal est obtenue par interpolation linéaire du pitch par l'équation I.2 :

$$p(t) = (1 - \alpha(t))p(t_m) + \alpha(t)p(t_{m+1}) \tag{I.2}$$

Pour les segments voisés, la forme d'onde caractéristique évolue lentement tandis que pour les segments non-voisés elle évolue rapidement. Ces 2 composantes sont séparées par filtrages passe-bas et passe-haut de fréquence de coupure de 20 Hz appliqués à $u(t, \Phi)$ le long de l'axe t.

Les 2 composantes à 2 dimensions résultant de ces filtrages sont appelées SEW (*Slowly Evolving Waveform*) et REW (*Rapidly Evolving Waveform*).

Elles sont numérisées séparément de façon à exploiter au mieux la différence de perception de ces 2 signaux.

Il est en effet inutile du point de vue perception de coder précisément la composante rapide REW. Une représentation grossière de la forme de son amplitude spectrale est suffisante. Mais ce signal évoluant rapidement, il faut transmettre ces informations à un rythme suffisamment élevé (par exemple 240 Hz).

Il faut au contraire coder la composante SEW avec beaucoup de précision car l'oreille perçoit les distorsions même faibles sur ces sons périodiques. Mais on peut transmettre les paramètres de la composante SEW à un rythme lent (typiquement à 40 Hz, c'est-à-dire toutes les 25 ms). La quantification de la composante SEW est faite par quantification vectorielle des coefficients de sa série de Fourier.

Les paramètres d'analyse sont donc transmis à des rythmes différents, par exemple le pitch à 80 Hz, les paramètres LPC à 40 Hz, la puissance du signal à 80 Hz, les amplitudes des coefficients de la série de Fourier de la composante REW à 240 Hz, et les paramètres de la SEW à 40 HZ.

Le synthétiseur reconstruit les 2 composantes SEW et REW à partir de leurs coefficients de Fourier. La composante REW est obtenue en combinant les amplitudes reçues du codeur avec une phase aléatoire. À chaque instant t la forme d'onde $u(t,\tau)$ de période 2π, peut être calculée par interpolation linéaire des CW transmises (voir l'équation I.1).

La longueur dénormalisée de la forme d'onde est obtenue par interpolation linéaire sur le pitch par l'équation I.2. L'excitation synthétique correspondante $e(t)$ est obtenue par l'équation I.3 :

$$e(t) = u(t, \Phi(t)) = u\left(t, \Phi(t_m) + \int_{t_m}^{t} \frac{2\pi}{p(u)} du\right) \tag{I.3}$$

L'excitation totale reconstruite $e(t)$ est obtenue en ajoutant les coefficients de Fourier des composantes REW et SEW. Elle est ensuite filtrée par le filtre de synthèse LPC. Les paramètres LPC sont interpolés linéairement à chaque instant. Un filtre de renforcement des formants est appliqué pour améliorer la qualité subjective du signal.

Un codeur WI travaillant à 2400 bps [78] donne de meilleurs résultats subjectifs que la norme FS1016 à 4800 bps utilisant un codage CELP. Le modèle WI n'est pas limitatif, on peut obtenir une meilleure qualité en augmentant le débit.

1.2.3.4 Les codeurs LPC à excitation mixte ou MELP Mixed Excitation Linear Prediction Coders Le nouveau standard DOD à 2400 bps [99, 124, 121] est un codeur LPC à excitation mixte (MELP = *Mixed Excitation Linear Prediction*).

Il utilise une excitation mixte c'est-à-dire formée de la somme d'une composante impulsionnelle et d'une composante de bruit. La composante impulsionnelle est formée d'un train d'impulsions périodique ou non. Cette excitation est une excitation multibande avec une intensité de voisement définie pour chaque bande de fréquence.

Le codeur fait une première estimation de la fréquence fondamentale, puis il calcule l'intensité de voisement dans 5 bandes de fréquence adjacentes. L'intensité de voisement est déterminée dans chaque bande par la valeur de l'autocorrélation normalisée. Dans la norme, cette intensité est codée sur 1 bit, chaque bande est donc classée voisée ou non-voisée. Après analyse le codeur peut positionner un indicateur appelé indicateur d'apériodicité (« *aperiodic flag* ») pour indiquer au décodeur que la composante impulsionnelle doit être apériodique.

Le codeur effectue par ailleurs une analyse spectrale par prédiction linéaire et calcule les amplitudes des 10 premières harmoniques du pitch sur la transformée de Fourier du signal résiduel. Ces

amplitudes sont quantifiées de manière vectorielle.

Les paramètres transmis par le codeur sont finalement : la période fondamentale, le drapeau d'apériodicité, les 5 intensités de voisement, 2 gains (correspondant aux énergies de 2 demi-trames), les paramètres spectraux et les 10 amplitudes d'harmoniques du pitch codées par quantification vectorielle.

Le synthétiseur interpole linéairement les différents paramètres de manière synchrone au pitch. La composante impulsionnelle est obtenue sur une période de pitch par transformée de Fourier inverse sur les 10 amplitudes de Fourier. Pour les sons non-voisés ou lorsque l'indicateur d'apériodicité est positionné, une perturbation aléatoire (jitter) est appliquée à la valeur de la période fondamentale.

Cette possibilité d'excitation impulsionnelle non périodique est particulièrement intéressante pour les zones de transitions entre sons. La composante impulsionnelle et la composante de bruit sont filtrées puis ajoutées. Le filtrage appliqué à la composante impulsionnelle a pour réponse impulsionnelle la somme de toutes les réponses impulsionnelles des filtres passe-bande pour les bandes voisées.

Le filtrage de la composante de bruit est déterminé de la même façon à partir des bandes non-voisées. L'excitation globale est ensuite filtrée par un filtre adaptatif de renforcement des formants et par le filtre de synthèse LPC.

Le signal synthétique résultant est mis à l'échelle en fonction de l'énergie de la trame originale et passé dans un filtre dont le but est d'étaler l'énergie des impulsions sur une période de pitch (*pulse dispersive filter*).

La figure I.6 représente la synthèse MELP. La qualité obtenue avec cette norme correspond à la

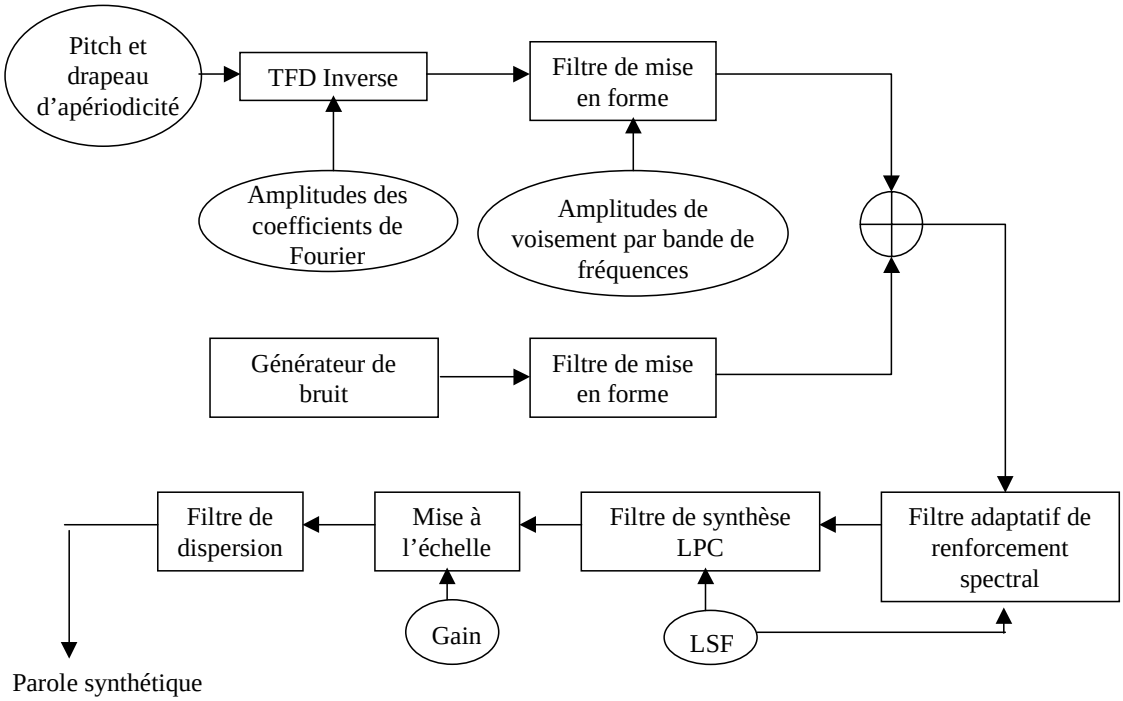


FIG. I.6 – Synthèse MELP

qualité dite de communication (MOS autour de 3,5) qui est légèrement inférieure à la qualité téléphonique classique. Cette qualité est nettement supérieure à celle du standard précédent LPC10e à 2400 bps.

1.2.3.5 Les codeurs HSX ou Harmonic Stochastic eXcitation coders Le codage HSX [82, 66] est très proche d'un point de vue conceptuel du codage MELP. La modélisation de l'excitation est plus élémentaire ce qui permet d'obtenir des débits plus bas et assure une complexité plus faible.

L'excitation synthétique d'un codeur HSX est la somme d'une composante harmonique et d'une composante stochastique. L'excitation est harmonique jusqu'à une fréquence limite f_{max} puis stochastique au-delà de cette fréquence. Le spectre de l'excitation est plat.

Le codeur détermine la fréquence fondamentale, l'énergie, les paramètres LPC, l'intensité de voisement dans 4 bandes de fréquence adjacentes. L'intensité de voisement est contrainte à être une fonction décroissante de la fréquence. Le codeur détermine la fréquence f_{max} par analyse multibande.

Le synthétiseur filtre l'excitation mixte par le filtre de synthèse LPC et par un filtre de renforcement des formants puis met à l'échelle le résultat en fonction de l'énergie de la trame originale.

Ce principe de codage permet d'obtenir des débits de 1200 à 600 bps avec une qualité subjective très supérieure au standard LPC10.

1.2.3.6 Le codeur MPEG-4 HVXC de SONY La norme MPEG4 stipule différents algorithmes pour le codage des sons et de la parole. Dans cette norme, le codage de la parole au plus bas débit (entre 2000 et 4000 bps) est effectué par l'algorithme HVXC (*Harmonic Vector eXcitation Coding*) de Sony [106]. Ce codeur applique 2 techniques différentes pour le codage des sons voisés et non-voisés. Les sons non-voisés sont traités par une méthode CELP sans dictionnaire adaptatif, et les sons voisés par une approche paramétrique proche des techniques MBE.

Ce codeur donne des résultats nettement supérieur au codeur FS1016 à 4800 bps.

1.2.4 Codage Multitrame

Un codage multitrame peut être appliqué pour diminuer le débit des codeurs précédents. Le standard OTAN à 800 bps [105] correspond à un codage LPC10 dans lequel on code globalement les paramètres de 3 trames successives.

1.3 Les codeurs à très bas débits

Pour obtenir des débits inférieurs à quelques centaines de bits par seconde, il n'est plus possible de travailler sur des trames de longueur fixe. Une approche segmentale utilisant des segments de longueur variable est nécessaire [41, 46, 52, 55, 69, 73, 86, 108, 109, 111, 112, 113, 114, 115, 117, 118, 125, 132].

On peut considérer que les codeurs à très bas débit effectuent une reconnaissance de segments acoustiques dans la phase d'analyse et une synthèse de parole à partir d'une suite d'indices de segments dans le décodeur. Le codeur réalise une transcription symbolique du signal de parole à partir d'un dictionnaire d'unités élémentaires de taille variable qui peuvent être des unités linguistiques (comme des phonèmes, des transitions entre phonèmes, des syllabes), on parle alors de vocodeurs phonétiques, ou bien des unités acoustiques obtenues automatiquement de manière non supervisée sur un corpus d'apprentissage, on utilisera par la suite l'expression vocodeurs pseudo-phonétiques pour désigner ces derniers codeurs.

On distingue 2 approches. La $1^{\text{ère}}$ segmente le signal de parole par différentes méthodes telles que l'identification de régions stables puis code les séquences de vecteurs spectraux de longueur va-

riable par des techniques comme la quantification matricielle par exemple. Dans la $2^{\text{ème}}$ approche, la segmentation et la quantification sont effectuées simultanément, à l'aide de techniques de reconnaissance d'unités de longueur variables utilisant des modèles de Markov cachés HMM (*Hidden Markov Model*) ou une technique DTW (*Dynamic Time Warping*).

1.3.1 Approche par segmentation et quantification séparées

Dans la $1^{\text{ère}}$ approche, la segmentation de la séquence de vecteurs spectraux peut se faire en comparant à un seuil une approximation de la dérivée des vecteurs spectraux. Souvent, les segments vont du milieu d'une zone stable au milieu de la zone suivante. Deux techniques sont couramment utilisées pour le codage des séquences de vecteurs spectraux, la quantification matricielle et le codage VFR (*Variable Frame Rate*).

La quantification matricielle [113, 38] code une suite de vecteurs spectraux de dimension p à l'aide d'un dictionnaire de matrices-codes, de dimension (N, p), formées de N vecteurs spectraux.

Si la longueur des séquences de vecteurs à coder est variable, on peut effectuer un alignement temporel (par DTW par exemple) entre la séquence et les matrices du dictionnaire, aussi bien lors de l'apprentissage que lors de la classification. Il faut alors transmettre une information sur la durée réelle du segment.

Dans [114], une contrainte est ajoutée sous la forme d'un réseau qui détermine quelles matrices-codes peuvent suivre une matrice-code donnée.

Dans la technique du codage VFR, on ne code et on ne transmet qu'un nombre réduit de vecteurs d'une séquence donnée. Au décodeur, les vecteurs manquants sont récupérés par interpolation à partir des vecteurs transmis [88, 76]. Le choix des vecteurs à coder est fait au codeur soit en boucle ouverte soit en boucle fermée. En boucle ouverte, on détermine les vecteurs en repérant ceux pour lesquels la dérivée des paramètres spectraux est la plus grande ou ceux qui présentent le plus grand écart par rapport à une interpolation effectuée sur les vecteurs adjacents. En boucle fermée, on choisit les vecteurs qui permettent d'obtenir la plus faible distorsion spectrale en synthèse, en testant toutes les possibilités.

1.3.2 Approche par segmentation et quantification conjointes

La $2^{\text{ème}}$ approche, par segmentation et quantification conjointes, utilise soit des matrices de longueur fixe [118], soit des matrices de longueur variable [54, 23], on parle alors de VVVQ *Variable to Variable Vector Quantization*, ou bien des modèles HMM.

L'approche phonétique ou pseudo-phonétique qui reconnaît la suite des phonèmes ou des unités acoustiques constituant le signal original est une technique de segmentation et indexation conjointes.

1.3.3 Paramètres transmis, synthèse

Quelle que soit la méthode de codage utilisée, pour chaque segment, le codeur transmet le symbole correspondant à l'unité reconnue ainsi que des paramètres auxiliaires tels que les contours de fréquence fondamentale et d'énergie, et la longueur du segment. La synthèse se fait généralement par concaténation de représentants des unités élémentaires. Elle peut utiliser les techniques PSOLA[11] (*Pitch Synchronous Overlap and Add*) ou HNM [122] (*Harmonic plus Noise Model*).

[11] PSOLA est une marque déposée CNET/France-Télécom.

Le débit moyen nécessaire pour coder la séquence d'unités reconnues est compris entre 50 et 150 bps (soit un débit moyen de 12 segments par seconde et 50 à 2000 unités). À ce débit il faut ajouter le débit des paramètres auxiliaires qui est du même ordre de grandeur.

Le retard introduit par ces codeurs est grand, de l'ordre de quelques centaines de ms.

Le dictionnaire d'unités élémentaires peut contenir des séquences de vecteurs spectraux de longueur variable, des segments de parole, des modèles HMM décrivant les unités.

Le dictionnaire d'unités acoustiques élémentaires n'est pas forcément le même pour le codeur et le décodeur.

Les vocodeurs phonétiques nécessitent la transcription phonétique du corpus d'apprentissage, tâche lourde et sujette aux erreurs qui doit être effectuée pour chaque nouvelle langue. La détermination automatique d'unités acoustiques à partir d'un corpus de parole non étiqueté est donc une approche intéressante.

C'est cette dernière approche que nous avons choisie pour nos travaux sur le codage à très bas débit (voir chapitre II).

2 Réduction de la complexité des codeurs CELP, application à la norme FS1016 à 4800 bps

La technique CELP (Code Excited Linear Prediction) introduite par Atal donne d'excellents résultats pour le codage de parole à des débits compris entre 4,8 Kbps et 16 Kbps. La complexité de ces codeurs est cependant très importante. La question de la complexité d'un algorithme est toujours relative. Il faut l'évaluer en fonction de la puissance de calcul des processeurs de traitement de signal (DSP[12]) existants à un moment donné. Lors de l'apparition de technique CELP, l'implantation d'un codeur CELP sur un seul DSP était un problème difficile. Aujourd'hui, le codeur CELP reste toujours l'algorithme le plus lourd en calcul d'un mobile GSM.

Jusqu'à fin 93, je me suis intéressée au développement de codeurs CELP de faible complexité.

J'ai encadré la thèse de Michel Mauc sur ce sujet, thèse qui a été soutenue en novembre 93.

Nous avons développé un algorithme utilisable pour tout codeur CELP qui permet de chercher la meilleure séquence d'excitation en utilisant une technique de sous-échantillonnage et une approche multi-étapes. Nous avons testé l'algorithme sur plusieurs structures de codeurs CELP et vérifié par des mesures objectives et subjectives qu'il ne dégradait par les performances en terme de qualité du codeur. Le gain en complexité dépend de plusieurs paramètres dont la taille des dictionnaires de séquences d'excitation, le facteur de sous-échantillonnage utilisé et le nombre d'étapes de l'algorithme.

Nous avons d'autre part proposé un algorithme de réduction de la complexité des codeurs CELP qui utilisent des dictionnaires de séquences d'excitation ternaires ce qui est le cas de la norme FS1016 à 4800 bps.

Nous avons finalement implanté sur un DSP TMS320C30 un codeur à 4800 bps conforme à la norme FS1016 et mettant en œuvre ces 2 algorithmes [90, 91, 94, 92, 93].

[12]DSP = Digital Signal Processor

2.1 Position du problème de la recherche de la meilleure séquence d'excitation dans un codeur CELP

Les codeurs CELP (voir section 1.2.1) effectuent un codage du signal de parole par quantification vectorielle en utilisant une technique d'analyse par synthèse.

La complexité du système est asymétrique. Le codeur (analyse) est très complexe alors que le décodeur (synthèse) est très simple.

Le codeur effectue un grand nombre de synthèses différentes et compare les signaux synthétiques obtenus avec le signal original. Le signal synthétique le plus proche de l'original, au sens d'un certain critère, est sélectionné. On parle de ce fait d'analyse par synthèse.

Les codeurs CELP, comme les codeurs LPC classiques, effectuent une analyse spectrale court-terme par prédiction linéaire sur chaque trame de signal de parole $s(n)$ et déterminent ainsi un filtre de prédiction $A(z)$. L'erreur de prédiction ou résiduel $r(n)$ peut être obtenue en filtrant $s(n)$ par $A(z)$. Le signal original $s(n)$ peut être reconstitué en filtrant le résiduel par $1/A(z)$ qui est appelé filtre de synthèse.

Les codeurs CELP ou LPC transmettent les coefficients spectraux et des paramètres caractérisant le signal résiduel.

Dans un codeur LPC, le résiduel est représenté par un modèle à 2 états voisé (V) ou non-voisé (NV) et est caractérisé par son énergie, sa classe (V ou NV) et la valeur de la période fondamentale T_0. Le résiduel ou excitation synthétique $\hat{r}(n)$ est, selon la classe, modélisé soit par un bruit blanc soit par une suite périodique d'impulsions, avec la bonne valeur d'énergie et de période.

Dans un codeur CELP, l'excitation synthétique est obtenue à partir de séquences de signal contenues dans des dictionnaires.

2.1.1 Prédiction long terme LTP (*Long Term Prediction*)

Une 1ère caractéristique du codage CELP est l'utilisation de la prédiction à long terme LTP[13]. Le résiduel $r(n)$ obtenu après prédiction court terme LPC est un signal quasi-périodique pour les sons voisés. L'objectif de la prédiction long terme est de tirer profit de cette périodicité pour le codage. Un filtre prédicteur long terme $B(z)$ possède généralement 1 à 3 coefficients. Dans le cas d'un coefficient unique, $B(z)$ s'exprime par :

$$B(z) = 1 - bz^{-Q}.$$

Pour effectuer une prédiction long terme, il faut calculer le coefficient b (problème linéaire) et la constante Q qui correspond à la période T_0 du signal en nombre d'échantillons. Généralement, la constante Q est calculée par une méthode d'autocorrélation. Pour des signaux échantillonnés à f_e = 8000 Hz, la valeur de la période est souvent estimée avec une précision meilleure que la durée d'un échantillon $T_e = 1/f_e = 125\ \mu s$. On parle alors de technique de pitch fractionnaire car la valeur de T_0 est estimée à une fraction de T_e près (jusqu'à $T_e/6$).

2.1.2 Constitution de l'excitation synthétique et boucle d'analyse-synthèse

Le résiduel $r(n)$ obtenu par application des prédictions court et long terme est un signal non périodique quasi-blanc. Le signal original $s(n)$ peut être obtenu en filtrant $r(n)$ par le filtre $\frac{1}{B(z)}\frac{1}{A(z)}$

[13]LTP = *Long Term Prediction*

Le signal de parole synthétique $\hat{s}(n)$ est généré en filtrant un signal d'excitation synthétique $\hat{r}(n)$ par le filtre $1/B(z)$ qui rend le signal périodique puis le filtre $1/A(z)$ qui met en forme l'enveloppe spectrale (voir figure I.1).

$$\hat{S}(z) = \frac{\hat{R}(z)}{B(z)A(z)}.$$

Dans un codeur CELP, le résiduel synthétique est déterminé par une technique de quantification vectorielle et d'analyse par synthèse. Cette technique est appliquée sur des durées de signal plus courtes que la trame utilisée pour l'analyse spectrale. Typiquement une trame (environ 20 ms) est découpée en 3 ou 4 sous-trames (environ 5 ms) de N échantillons.

La figure I.7 représente une boucle d'analyse-synthèse avec 2 dictionnaires d'excitation.

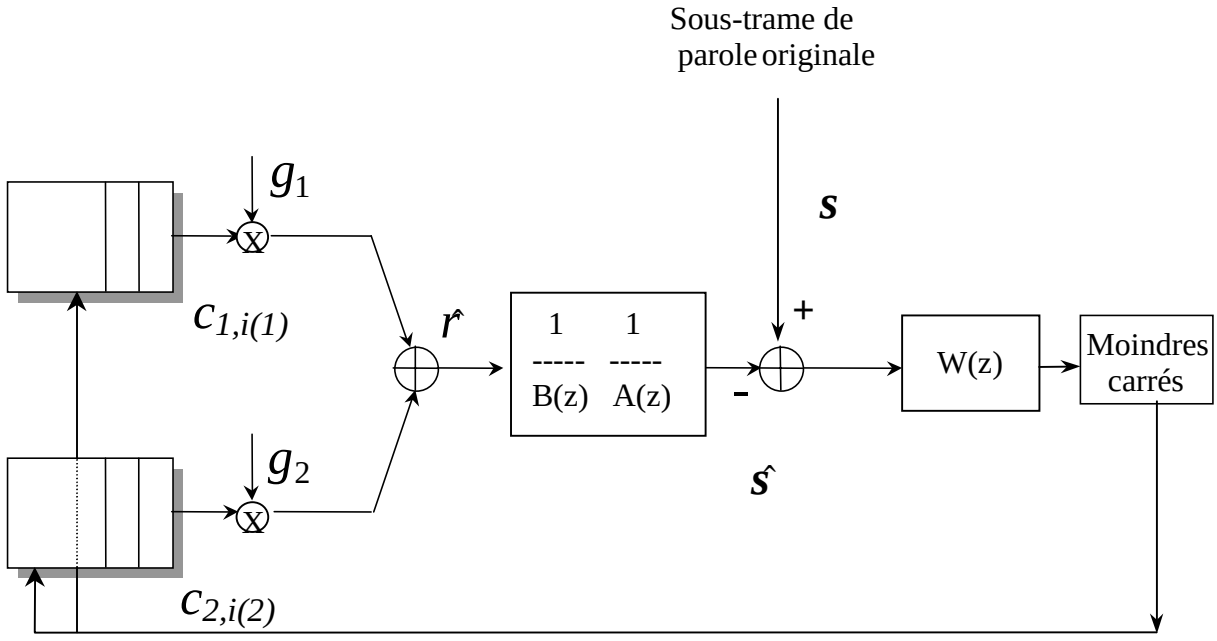


FIG. I.7 – Boucle d'analyse-synthèse dans un codeur CELP

Le codeur (voir figure I.7) utilise K dictionnaires[14] contenant des séquences d'excitation de longueur donnée N. Ces séquences sont généralement normalisées en énergie et de spectre quasi-blanc. Le signal d'excitation synthétique $\hat{r}(n)$, de N échantillons, est fabriqué en combinant un ensemble de K séquences de dictionnaires pondérées par un gain.

On note $\mathbf{c_{j,i(j)}}$ la séquence, choisie dans le dictionnaire j, et d'indice $i(j)$. On note $g_{j,i(j)}$ le gain associé à la séquence $\mathbf{c_{j,i(j)}}$. Le résiduel synthétique, correspondant à un ensemble de K séquences d'indices $i(j)$ avec $j \in [0, K-1]$ s'écrit :

$$\hat{r}(n) = \sum_{j=0}^{K-1} g_{j,i(j)} c_{j,i(j)}(n)$$

Pour simplifier les écritures, on note g_j au lieu de $g_{j,i(j)}$ quand il n'y a pas ambiguïté.

[14] K est appelé ordre de modélisation.

Le codeur teste toutes les combinaisons possibles de K séquences pour déterminer celle qui génère le signal de parole synthétique $\hat{s}(n)$ le plus proche de l'original $s(n)$ au sens d'un critère appelé « critère perceptuel ».

Le critère perceptuel est un critère de type moindres carrés pondérés dans le domaine fréquentiel. Il met à profit les propriétés auditives de masquage des bruits par des sons de plus fortes amplitudes. La différence $s(n) - \hat{s}(n)$ est filtrée par un filtre de pondération perceptuelle $W(z)$, dont la fonction de transfert a une amplitude plus forte dans les plages de fréquences où l'amplitude du spectre de $s(n)$ est faible. L'idée est d'accorder un poids plus fort aux erreurs de quantification les plus audibles.

Un filtre de pondération couramment utilisé est du type $\frac{A(z)}{A(z/\gamma)}$ où γ est une constante proche de 1. La figure I.8 représente le spectre LPC d'une trame de parole et la fonction de transfert du filtre perceptuel $W(z)$ correspondant pour $\gamma = 0.75$.

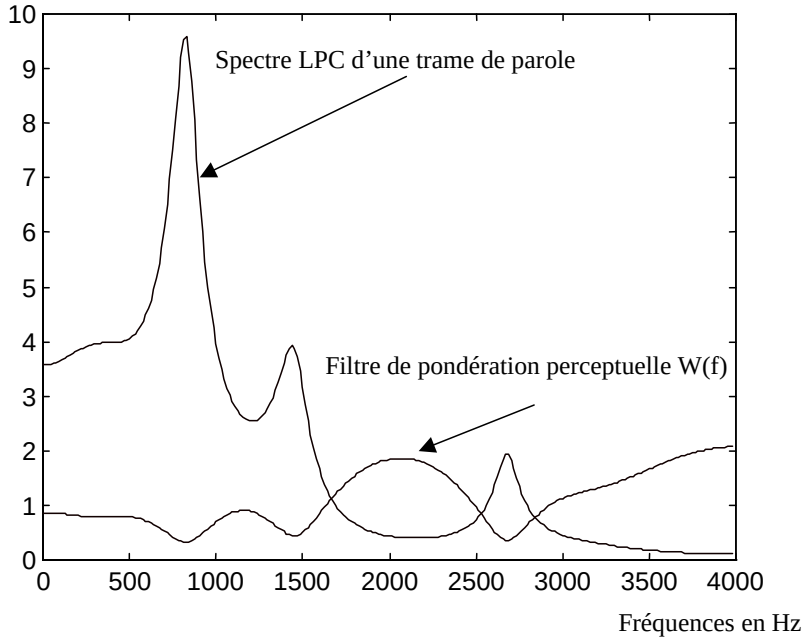


FIG. I.8 – Spectre LPC d'une trame de parole et filtre de pondération perceptuelle correspondant

En pratique le filtre perceptuel est appliqué aux 2 membres de la différence et le codeur mesure la différence entre le signal de parole dit « perceptualisé » (**s** filtré par $W(z)$) et le signal obtenu par filtrage de $\hat{\mathbf{r}}$ par le filtre :

$$H(z) = \frac{W(z)}{A(z)B(z)}.$$

Pour un filtre perceptuel $W(z) = \frac{A(z)}{A(z/\gamma)}$, le filtre $H(z)$ se réduit à

$$H(z) = \frac{1}{A(z/\gamma)}.$$

En notant $w(n)$ la réponse impulsionnelle du filtre perceptuel, on peut écrire le critère J sous la forme :

$$J = \sum_{n=0}^{N-1} e(n)^2.$$

avec :

$$e(n) = w(n) * (s(n) - \hat{s}(n)) = (s(n) * w(n) - \hat{s}(n) * w(n)) .$$

L'erreur $\mathbf{e}$ peut s'écrire en mettant à part $\hat{\mathbf{p}}_\mathbf{o}$ « la mémoire du filtre », c'est-à-dire la composante de la sortie du filtre $H(z)$ qui ne dépend que des excitations passées et qui correspond à la sortie du filtre pour une entrée nulle.
On note $h(n)$ la réponse impulsionnelle du filtre $H(z)$ et $\hat{r}(n)$ l'entrée du filtre $H(z)$ pour une combinaison donnée de séquences des dictionnaires. À l'instant n, la sortie $q(n)$ du filtre $H(z)$ s'écrit, en tenant compte des conditions initiales :

$$q(n) = \sum_{k=0}^{+\infty} h(k)\hat{r}(n-k) = \sum_{k=0}^{n} h(k)\hat{r}(n-k) + \sum_{k=n+1}^{+\infty} h(k)\hat{r}(n-k).$$

Pour $k > n$, les valeurs $n-k$ sont négatives et l'entrée $\hat{r}(n-k)$ ne dépend pas de la combinaison de séquences choisies dans la sous-trame mais seulement des séquences choisies dans les sous-trames précédentes. On appelle mémoire du filtre $\hat{\mathbf{p}}_\mathbf{0}$ la sortie correspondante :

$$\hat{p}_0(n) = \sum_{k=n+1}^{+\infty} h(k)\hat{r}(n-k).$$

L'erreur $\mathbf{e}$ peut donc s'écrire :

$$\begin{aligned} e(n) &= s_n * w_n - \hat{p}_0(n) - \sum_{k=0}^{n} h(k)\hat{r}(n-k). \\ \hat{r}(n) &= \sum_{j=0}^{K-1} g_j c_{j,i(j)} \end{aligned}$$

Dans la dernière équation $i(j)$ représente l'indice de la séquence choisie dans le dictionnaire j.

On note $\mathbf{p}$ le signal cible qui est la différence entre le signal de parole original filtré par le filtre perceptuel et la mémoire du filtre $H(z)$:

$$p(n) = s_n * w_n - \hat{p}_0(n).$$

Et on note $\hat{\mathbf{p}}$ la sortie du filtre $H(z)$ excité par $\hat{\mathbf{r}}$ avec des conditions initiales nulles :

$$\hat{p}(n) = \sum_{k=0}^{n} h(k)\hat{r}(n-k).$$

La figure I.9 représente une boucle d'analyse-synthèse implantée de cette manière. La figure permet de plus de fixer les notations utilisées.

2.1.3 Dictionnaire adaptatif

En pratique, il est possible de supprimer le filtrage par $1/B(z)$[15], à condition d'ajouter un dictionnaire supplémentaire appelé dictionnaire adaptatif. Le schéma général de la figure I.9 reste valable, seule la valeur de $H(z)$ passe de $H(z) = \frac{W(z)}{A(z)B(z)}$ à $H(z) = \frac{W(z)}{A(z)}$. Le contenu du dictionnaire varie au cours du temps, à la différence des dictionnaires stochastiques qui sont fixes. Il contient les séquences d'excitations passées, parmi lesquelles, si le signal est périodique, on doit pouvoir retrouver

[15] lorsque $1/B(z)$ ne possède qu'un coefficient

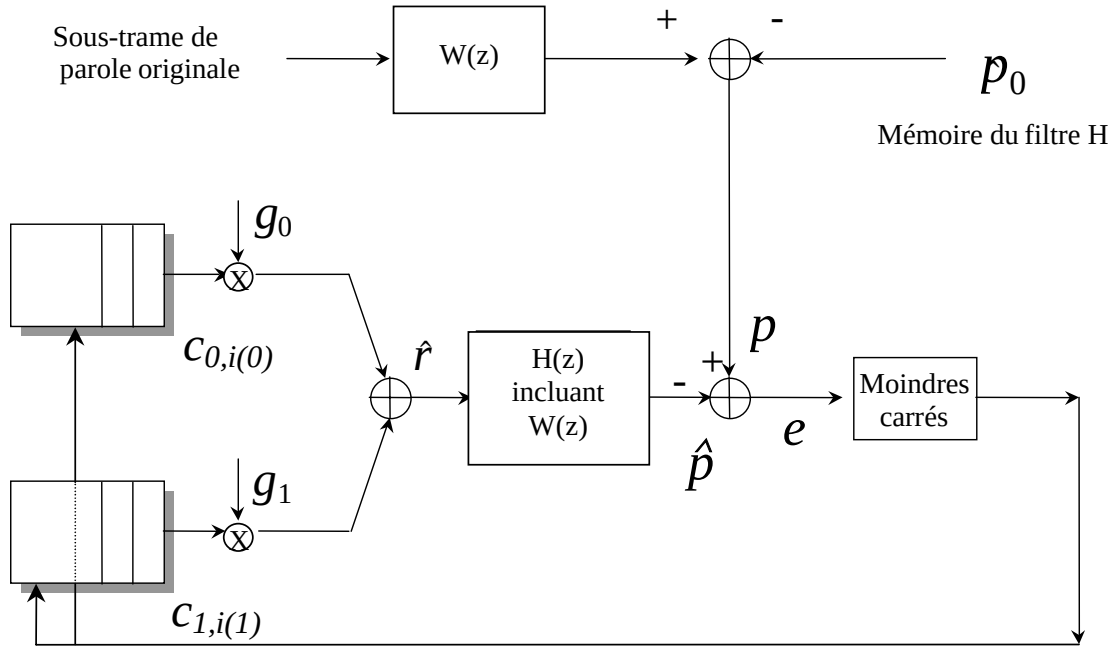


FIG. I.9 – Implantation pratique d'une boucle d'analyse-synthèse dans un codeur CELP

une séquence proche du résiduel à traiter. En appelant $r_b(n)$ la sortie du filtre $1/B(z)$ excité par $\hat{r}(n)$, on peut écrire :

$$r_b(n) = \hat{r}(n) + b\hat{r}(n-Q).$$

Si $Q \geq N$ [16], les termes $\hat{r}(n)$ sont tous connus au démarrage de la sous-trame (les indices $n-Q$ sont négatifs, il s'agit donc d'échantillons de résiduel de sous-trames précédentes). On peut dans ce cas, ne pas effectuer le filtrage par $1/B(z)$, mais plutôt calculer $r_b(n)$ en ajoutant à $\hat{\mathbf{r}}$ un vecteur $\mathbf{r_Q}$ de N échantillons de résiduel passé.

$$\begin{aligned} \forall n \in [0, N-1] \qquad & r_Q(n) = \hat{r}(n-Q)\ , \\ \hat{\mathbf{r}} \quad = \quad & \sum_{j=0}^{K-1} g_j \mathbf{c_{j,i(j)}} + b\mathbf{r_Q} \end{aligned}$$

Cette formulation suppose la constante Q connue, mais on peut l'étendre en incluant, dans la boucle d'analyse- synthèse, la recherche de la valeur de Q, c'est-à-dire en effectuant plusieurs synthèses correspondant aux différentes valeurs possibles de Q. Cela revient à utiliser un dictionnaire contenant tous les vecteurs $\mathbf{r}_Q$ correspondant à la plage de valeurs de Q que l'on veut tester. Ce dictionnaire est appelé dictionnaire adaptatif car son contenu varie au cours du temps.

Le vecteur $\hat{\mathbf{r}}$ s'écrit alors :

$$\hat{\mathbf{r}} = \sum_{j=0}^{K} g_j \mathbf{c_{j,i(j)}} \text{ où } g_K = b_{i(K)} \text{ et } c_{K,i(K)} = \mathbf{r}_{i(K)}$$

[16] ce qui est le cas au moins pour les voix graves. Par exemple à $f_e = 8000\ Hz$ avec $N = 40$, $Q > N$ signifie que la fréquence fondamentale est inférieure à $200\ Hz$.

Pour un dictionnaire adaptatif à M vecteurs, en notant Q_{min} et Q_{max} les valeurs minimale et maximale de Q et en prenant comme instant 0 le début de la sous-trame concernée, le vecteur numéro i $\mathbf{c_{adaptatif,i}}$ du dictionnaire adaptatif s'exprime par :

$$\begin{array}{ll} \forall i \in [0, M-1] & \\ \forall n \in [0, N-1] & c_{adaptatif,i}(n) = \hat{r}(n - i - Q_{min}). \end{array}$$

Et :

$$\begin{aligned} \mathbf{c_{adaptatif,0}} &= [\hat{r}(-Q_{min}), \hat{r}(1-Q_{min}), \cdots, \hat{r}(N-1-Q_{min})], \\ \mathbf{c_{adaptatif,M-1}} &= [\hat{r}(-Q_{max}), \hat{r}(1-Q_{max}), \cdots, \hat{r}(N-1-Q_{max})]. \end{aligned}$$

On peut généraliser l'approche au cas où les valeurs de retards Q à tester sont fractionnaires. Il suffit de remplir le dictionnaire adaptatif avec des vecteurs obtenus à partir du résiduel $\hat{\mathbf{r}}$ retardé d'une valeur fractionnaire.

La constitution du dictionnaire adaptatif est simple lorsque $\forall Q \in [Q_{min}, Q_{max}] \;\; Q \geq N$. Mais pour les valeurs $Q < N$, on ne connaît plus tous les termes, en début de trame :

$$\hat{r}(n - Q) \quad \forall n \in [0, N-1] \quad \forall Q \in [Q_{min}, Q_{max}].$$

En particulier, les échantillons $\hat{r}(0)$ à $\hat{r}(N - 1 - Q_{min})$ n'ont pas encore été calculés, puisque leurs indices sont positifs. Pour ces petites valeurs de retard Q, on résout le problème, en répétant périodiquement (à la période Q) les valeurs de résiduel connues, pour fabriquer un vecteur complet de longueur N. Par exemple si $N = 40$, et $Q_{min} = 25$, $F_0 = 320\ Hz$, on fabrique le premier vecteur du dictionnaire par :

$$\mathbf{c}_{\text{adaptatif,0}} = [\hat{r}(-Q_{min}), \hat{r}(1-Q_{min}), \cdots, \hat{r}(0), \hat{r}(-Q_{min}), \hat{r}(1-Q_{min}), \cdots, \hat{r}(10)].$$

2.1.4 Algorithme standard de recherche de la meilleure excitation

En conclusion, la figure I.9 représente le cas général d'une boucle d'analyse-synthèse. La valeur du filtre $H(z)$ variant selon que l'on utilise ou pas un filtre de prédiction long terme.

Le problème de la recherche de la meilleure excitation synthétique $\hat{\mathbf{r}}$ se ramène donc, dans tous les cas, à celui de l'approximation linéaire au sens des moindres carrés, du vecteur cible $\mathbf{p}$ par une combinaison linéaire de K[17] vecteurs des dictionnaires $\mathbf{c_{j,i(j)}}$ filtrés par $H(z)$ avec des conditions initiales nulles et multipliés par les gains optimaux g_j.

Par la suite, on utilise l'expression « dictionnaire filtré » pour désigner le dictionnaire obtenu en filtrant un des dictionnaires de départ par $H(z)$. chaque vecteur $\mathbf{f_{j,i(j)}}$ du dictionnaire filtré numéro j est ainsi constitué des N premiers échantillons du signal obtenu en filtrant le vecteur $\mathbf{c_{j,i(j)}}$ par $H(z)$ avec des conditions initiales nulles.

Avec ces notations, le critère J s'écrit :

$$\begin{aligned} J &= \min \|\mathbf{p} - \hat{\mathbf{p}}\|^2. \\ \hat{\mathbf{p}} &= \sum_{i=0}^{K-1} g_j \mathbf{f_{j,i(j)}}. \end{aligned}$$

[17] On inclut le dictionnaire adaptatif dans ce nombre K, si nécessaire

Pour une combinaison de K indices $i(j)$ donnés, le vecteur $\hat{\mathbf{p}}$ s'écrit :

$$\hat{\mathbf{p}} = \mathbf{F}\mathbf{g}.$$

où $\mathbf{g}$ est le vecteur des gains optimaux g_j et $\mathbf{F}$ est la matrice formée des K vecteurs colonnes $\mathbf{f_{j,i(j)}}$. la solution optimale du problème de l'approximation linéaire au sens des moindres carrés de $\mathbf{p}$ par $\hat{\mathbf{p}}$ est la projection de l'observation $\mathbf{p}$ sur l'espace vectoriel engendré par les vecteurs $\mathbf{f_{i(j)}}$ de la combinaison choisie. On montre simplement que cela revient à maximiser $\|\hat{\mathbf{p}}\|^2$. En notant $\mathbf{g_{opt}}$ le vecteur des gains optimaux, on obtient :

$$\begin{aligned} \mathbf{g_{opt}} &= \left(\mathbf{F^T F}\right)^{-1} \mathbf{F^T p} \\ \hat{\mathbf{p}} &= \mathbf{g_{opt} p} \end{aligned}$$

Et la combinaison optimale de vecteurs pour constituer $\hat{\mathbf{r}}$ est celle qui maximise l'expression :

$$\|\hat{\mathbf{p}}\|^2 = \mathbf{p^T F}(\mathbf{F^T F})^{-1}\mathbf{F^T p}.$$

La solution optimale est extrêmement lourde en calcul. Ainsi pour un ordre de modélisation $K = 3$ et des dictionnaires formés de 256 vecteurs, y-a-t'il plus de 16 millions de combinaisons d'indices à tester. Pour chaque combinaison, il faut résoudre un système d'équations linéaire à 3 inconnues pour trouver les gains optimaux.

Aussi, utilise-t-on généralement un algorithme sous-optimal, tel que l'algorithme itératif standard [103].

Cet algorithme explore les K dictionnaires l'un après l'autre de $j = 0$ à $j = K - 1$.

À l'étape j, il recherche dans le dictionnaire filtré numéro j le vecteur $f_{j,i(j)}$ tel que $\hat{\mathbf{p}}_\mathbf{j} = g_j \mathbf{f_{i(j)}}$ approche au mieux le vecteur cible $\mathbf{p_j}$. Le vecteur cible $\mathbf{p_j}$ à l'étape j est égal au vecteur cible $\mathbf{p_{j-1}}$ après suppression de la contribution du vecteur trouvé à de l'étape $j - 1$:

$$\mathbf{p_j} = \mathbf{p_{j-1}} - g_{j-1}\mathbf{f_{j-1,i(j-1)}}.$$

À la première itération, le vecteur cible est le vecteur $\mathbf{p}$.

À l'étape j, le vecteur $\hat{\mathbf{p}}_\mathbf{j}$ optimum est celui qui maximise l'expression $\|\hat{\mathbf{p}}_\mathbf{j}\|^2$. Cette norme est une fonction $PC_j(k)$ de l'indice k du vecteur dans le dictionnaire filtré, et comme la matrice $\mathbf{F}$ n'a qu'une colonne $\mathbf{f_{j,k}}$ dans ce cas là, on peut écrire :

$$PC_j(k) = \frac{\langle \mathbf{f_{j,k}}, \mathbf{p}_j \rangle^2}{\langle \mathbf{f_{j,k}}, \mathbf{f_{j,k}} \rangle} = \frac{\beta_k^2}{\alpha_k}.$$

L'indice k_{opt} correspondant au vecteur optimal dans le dictionnaire j filtré est donné par :

$$k_{opt} = \arg\max_k PC_j(k) = \arg\max_k \frac{\beta_k^2}{\alpha_k}.$$

Et le gain optimal associé est égal à :

$$g_{opt}(k_{opt}) = \frac{\beta(k_{opt})}{\alpha(k_{opt})}.$$

La notation $< \mathbf{u}, \mathbf{v} >$ représente le produit scalaire de 2 vecteurs $\mathbf{u}$ et $\mathbf{v}$. Les termes α et β s'interprètent comme des termes d'autocorrélation des vecteurs $\mathbf{f}$ et d'intercorrélation entre les vecteurs $\mathbf{f}$ et $\mathbf{p_j}$.

L'algorithme itératif standard est sous-optimal et différentes méthodes ont été proposées pour l'améliorer en calculant tous les gains à chaque étape, ou en orthogonalisant les dictionnaires filtrés. On peut citer les travaux de Moreau et Dymarski sur le sujet [103, 104, 59]..

2.1.5 Évaluation de la complexité de l'algorithme itératif standard

On utilise par la suite les notations suivantes :

N = longueur d'une sous-trame = longueur des vecteurs (ou séquences) du dictionnaire d'excitations.

M = nombre de vecteurs dans les dictionnaires d'excitation stochastique (on les suppose tous de même taille pour simplifier les notations).

M_a = taille du dictionnaire adaptatif.
p = ordre de la prédiction linéaire utilisée pour le calcul du filtre $1/A(z)$.
N_T = Longueur d'une trame.
K = nombre de dictionnaires utilisés ($K-1$ stochastiques plus 1 adaptatif).

Certains calculs sont effectués au rythme des trame (f_e/N_T), d'autres au rythme ses sous-trames (f_e/N).

1. L'analyse spectrale par prédiction linéaire et le calcul du vecteur cible $\hat{\mathbf{p}}$ sont faits à chaque trame. L'essentiel du calcul est constitué du calcul des p coefficients d'autocorrélation et du calcul de $\hat{\mathbf{p}}$ par filtrage, ce qui représente environ :

$$\frac{f_e}{N_T} 3pN_T = 3f_e p \text{ additions-multiplications par seconde.}$$

2. Le calcul des M expressions $PC_j(k)$, sans algorithme spécifique, nécessite le calcul des M séquences $\mathbf{f}_{j,k}$ des dictionnaires filtrés.

   Cas des dictionnaires stochastiques : Si les coefficients de prédiction linéaire ne sont pas mis à jour à chaque sous-trame, il suffit d'effectuer le filtrage des dictionnaires stochastiques 1 fois par trame, ce qui correspond approximativement à :

$$\frac{f_e}{N_T}(K-1)MNp \text{ additions-multiplications par seconde.}$$

   Cas du dictionnaire adaptatif : Le contenu de ce dictionnaire est actualisé à chaque sous-trame. À la $1^{\text{ère}}$ sous-trame, on filtre complètement le dictionnaire. Puis, à chaque nouvelle sous-trame, on calcule N nouveaux échantillons d'excitation et les N vecteurs les plus anciens du dictionnaire sont remplacés par les N nouveaux. Il suffit d'effectuer le filtrage pour ces Nnouveau vecteur de résiduel seulement. La charge de calcul est donc :

$$\frac{f_e}{N_T}\left(M_a NP + (N_T/N - 1)\, NNp\right) \text{ additions-multiplications par seconde.}$$

3. La recherche de la meilleure excitation est effectuée à chaque sous-trame.
   Il faut évaluer les termes α_k et β_k^2 pour chaque dictionnaire.
   Cas des dictionnaires stochastiques :
   Le calcul des $\alpha(k)$ est fait une fois par trame et nécessite :

$$\frac{f_e}{N_T}(K-1)MN \text{ additions-multiplications par seconde.}$$

   Le calcul des $\beta(k)$ est fait à chaque sous-trame car le vecteur cible qui intervient dans le produit scalaire change. Le nombre d'opérations à réaliser est :

$$\frac{f_e}{N}(K-1)MN = f_e(K-1)M \text{ additions-multiplications par seconde.}$$

   Cas du dictionnaire adaptatif : Le calcul des $\alpha(k)$ et $\beta(k)$ est fait à chaque sous-trame car le contenu du dictionnaire change. Le nombre d'opérations à effectuer pour le calcul des $\alpha(k)$ vaut :

$$\frac{f_e}{N_T}\left(M_aN + \left(\frac{N_T}{N} - 1\right)N\right) \text{ additions-multiplications par seconde.}$$

   Le calcul des $\beta(k)$ nécessite :

$$\frac{f_e}{N}M_aN \text{ additions-multiplications par seconde.}$$

   Une fois les termes $\alpha(k), \beta(k)^2$ connus, l'évaluation des M termes $PC(k)$ nécessite M divisions par dictionnaire et par sous-trame. La recherche du maximum des M valeurs $PC_j(k)$ demande M tests par dictionnaire et par sous-trame. Soit au total :

$$\frac{f_e}{N}\left((K-1)M + M_a\right) \text{ divisions et tests par seconde.}$$

En conclusion, le nombre total d'opérations $NBops$ à effectuer par seconde, est égal à :

$$NBops = \begin{cases} & 3f_ep & \text{additions-multiplications par seconde,} \\ & & \text{pour la prédiction linéaire et le calcul de } \mathbf{p}. \\ + & \frac{f_e}{N_T}(K-1)MNp & \text{additions-multiplications par seconde,} \\ & & \text{pour le filtrage des dictionnaires stochastiques.} \\ + & \frac{f_e}{N_T}\left(M_aNP + (N_T/N-1)\,N^2p\right) & \text{additions-multiplications par seconde,} \\ & & \text{pour le filtrage du dictionnaire adaptatif.} \\ + & \frac{f_e}{N_T}(K-1)MN & \text{additions-multiplications par seconde,} \\ & & \text{calcul des } \alpha(k) \text{ des dictionnaires stochastiques} \\ + & f_e(K-1)M & \text{additions-multiplications par seconde,} \\ & & \text{calcul des } \beta(k) \text{ des dictionnaires stochastiques} \\ + & \frac{f_e}{N_T}\left(M_aN + \left(\frac{N_T}{N} - 1\right)NN\right) & \text{additions-multiplications par seconde,} \\ & & \text{pour le calcul des } \alpha(k) \text{ du dictionnaire adaptatif} \\ + & \frac{f_e}{N}M_aN & \text{additions-multiplications par seconde,} \\ & & \text{pour le calcul des } \beta(k) \text{ du dictionnaire adaptatif} \\ + & \frac{f_e}{N}\left((K-1)M + M_a\right) & \text{divisions et tests par seconde,} \\ & & \text{pour le calcul des } PC(k) \text{ et les test finaux.} \end{cases}$$

Application numérique :
Par exemple, pour les valeurs suivantes :

$$\begin{cases} f_e &= 8000\ Hz \\ M &= 256 \\ M_a &= 128 \\ K &= 3,\ \text{1 dictionnaire adaptatif et 2 stochastiques.} \\ N_T &= 160 \\ N &= 40 \\ p &= 10 \end{cases}$$

On obtient un nombre d'opérations élémentaires par seconde égal à :

	240 000	Analyse LPC et calcul de **p**
+	10 240 000	Filtrage dictionnaires stochastiques
+	4 960 000	Filtrage dictionnaire adaptatif
+	1 024 000	Calcul des $\alpha(k)$ stochastiques
+	4 096 000	Calcul des $\beta(k)$ stochastiques
+	496 000	Calcul des $\alpha(k)$ adaptatifs
+	1 024 000	Calcul des $\beta(k)$ adaptatifs
+	128 000	Calcul des $PC(k)$ et tests finaux
=	**22 028 000**	Opérations élémentaires par seconde.

L'algorithme optimal demanderait dans ce cas là plus de $80\ 10^9$ opérations.

Par comparaison avec le codeur, la complexité du décodeur est très faible. Le codeur transmet les indices des meilleures séquences d'excitation trouvées ainsi que les gains associés et des paramètres décrivant l'enveloppe spectrale du signal. Le décodeur génère le signal de parole synthétique en filtrant le signal d'excitation synthétique par le filtre de synthèse. Le filtre de synthèse est déduit des paramètres spectraux. Le signal d'excitation est la combinaison linéaire des vecteurs des dictionnaires d'excitation correspondant aux indices reçus et aux gains reçus.

Le nombre d'opérations élémentaires par seconde pour la synthèse est donc environ :

$$f_e(Kp).$$

Ce qui pour l'application numérique précédente correspond à $240\ 000$ opérations élémentaires par seconde.

Pour les DSP de la fin des années 80, la complexité de l'algorithme itératif standard[18] était trop grande pour permettre une implantation directe sur un seul DSP. Aussi, de nombreux algorithmes ont-ils été proposés pour diminuer la complexité de ce calcul.

2.1.6 Algorithmes CELP rapides

On peut distinguer 2 types d'approches dans les travaux réalisés :

[18] avec les valeurs de l'application numérique précédente, environ 22 Mips. Mips = Méga instructions par seconde, une instruction correspondant à une opération élémentaire de type Addition-Multiplication.

- Des algorithmes n'utilisant pas directement le dictionnaire filtré. On peut citer par exemple, les travaux de Trancoso avec Atal aux Bell Labs [126, 127], et ceux de Kleijn [80].
- Des algorithmes utilisant des dictionnaires de structures particulières facilitant les calculs. On peut citer les travaux de J. Menez [100], le développement du codeur VSELP chez Motorola [65], ou les travaux de Adoul et Laflamme utilisant des codes algébriques [5, 81].

2.1.6.1 Algorithme sans filtrage des dictionnaires L'opération de filtrage des vecteurs[19] $\mathbf{c_k}$ peut s'écrire comme une équation de récurrence. Et pour un filtre $H(z)$ d'ordre p purement récursif, de coefficients $\tilde{a}_i$ on obtient :

$$f_k(n) = c_k(n) - \sum_{i=1}^{p} \tilde{a}_i f_k(n-i).$$

Ce qui demande p opérations par échantillons.

L'opération de filtrage (avec conditions initiales nulles) peut aussi s'écrire avec une convolution sous la forme :

$$f_k(n) = -\sum_{i=0}^{n} \tilde{a}_i c_k(n-i).$$

Ce qui s'exprime sous la forme matricielle :

$$\begin{aligned} \mathbf{f_k} &= \mathbf{H c_k} \\ \mathbf{H} &= \begin{pmatrix} h(0) & 0 & \cdots & \cdots & 0 \\ h(1) & h(0) & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ h(N-1) & h(N-2) & \cdots & h(1) & h(0) \end{pmatrix}. \end{aligned}$$

Par la suite, on appelle la matrice de filtrage, la matrice $\mathbf{H}$.

Cas du dictionnaire adaptatif :

Le cas du dictionnaire adaptatif est particulier, parce qu'il a une forme de Tœplitz . Pour simplifier les écritures, on suppose ici que $Q_{min} > N$. Après le calcul du 1^{er} vecteur filtré $\mathbf{f_0}$ (qui demande pN opérations), les vecteurs suivants se déduisent par récurrence :

$$\begin{aligned} f_k(0) &= h(o)\hat{r}(-Q_{min}-k) \\ f_k(n) &= f_{k+1}(n-1) + h(n)\hat{r}(-Q_{min}-k) \quad \forall n \in [1, N-1]. \end{aligned}$$

Le filtrage du dictionnaire adaptatif, pour la $1^{\text{ère}}$ sous-trame, peut donc se faire en :

$$pN + (M_a - 1)N \text{ opérations élémentaires.}$$

D'où :

$$\frac{f_e}{N_T}\left(pN + (M_a-1)N + \left(\frac{N_T}{N} - 1\right)(pN + (N-1)N\right) \text{ opérations par seconde.}$$

Au lieu de :

$$\frac{f_e}{N_T}\left(M_a N p + (N_T/N - 1)\, N^2 p\right).$$

La complexité est divisée par une valeur proche de p. On passe ainsi, pour l'application numérique vue plus haut de $4,96$ Mips à $0,57$ Mips.

[19] On note maintenant les $k^{\text{ème}}$ vecteurs des dictionnaires $\mathbf{c_k}$.

Cas des dictionnaires stochastiques

– Calcul des $\beta(k)$:
Pour les dictionnaires stochastiques, sans structure particulière, on peut ré-écrire les termes d'intercorrélation β sans faire intervenir les vecteurs filtrés, car :

$$\beta(k) = \langle \mathbf{f_k}, \mathbf{p_j} \rangle = \mathbf{c_k^T H^T p_j} = \left\langle \mathbf{c_k}, \mathbf{H^T p_j} \right\rangle .$$

Le produit scalaire entre $\mathbf{f_k}$ et $\mathbf{p_j}$ peut avantageusement être remplacé par un produit scalaire entre $\mathbf{c_k}$ et $\mathbf{H^T p_j}$. Il n'est pas nécessaire de calculer les M vecteurs du dictionnaire filtré, il suffit de calculer le vecteur $\mathbf{H^T p_j}$.
Toutefois cette méthode est sans intérêt s'il faut quand même calculer les dictionnaires filtrés pour obtenir les $\alpha(k)$.

– Calcul des $\alpha(k)$:
Les termes $\alpha(k)$ s'écrivent en utilisant la matrice H :

$$\alpha(k) == \langle \mathbf{f_k}, \mathbf{f_k} \rangle = \mathbf{c_k^T H^T H c_k}$$

Cette expression ne fait pas intervenir les dictionnaires filtrés, mais si on la calcule directement elle n'apporte rien. De nombreux algorithmes ont cherché à effectuer ce calcul efficacement en imposant une structure à la matrice $(H^T H)$.

Si on impose à la matrice $(H^T H)$ une structure de Tœplitz le calcul se simplifie beaucoup. Les $\alpha(k)$ s'écrivent alors uniquement en fonction des N termes de la première ligne de $\mathbf{H^T H}$ qui correspondent à l'autocorrélation de $h(n)$ et que l'on note $r_h(n)$, pour $n \in [0, N-1]$:

$$r_h(n) = \sum_{i=0}^{N-1-n} h(i)h(i+n).$$

En notant de la même façon l'autocorrélation des vecteurs $\mathbf{c_k}$:

$$r_{c_k}(n) = \sum_{i=0}^{N-1-n} c_k(i)c_k(i+n).$$

Les termes $\alpha(k)$ s'écrivent alors :

$$\alpha(k) = r_h(0)r_{c_k}(0) + 2\sum_{i=0}^{N-1} r_h(i)r_{c_k}(i).$$

Pour les dictionnaires adaptatifs les $r_{c_k}(n)$ peuvent être calculés à l'avance.
Le calcul des $\alpha(k)$ par cette formule ne demande pas le filtrage des dictionnaires. Il faut calculer les N termes $r_h(n)$ puis les $\alpha(k)$. Le nombre d'opérations à effectuer pour le calcul des $\alpha(k)$ sans avoir à filtrer les dictionnaires, est égal, pour les $(K-1)$ dictionnaires, à :

$$\frac{f_e}{N_T}(K-1)\left(M(N+1) + \frac{NN_T}{2}\right).$$

Il faut comparer cette expression à la somme du nombre d'opérations pour filtrer le dictionnaire et du nombre d'opérations pour le calcul des $\alpha(k)$ utilisant le dictionnaire filtré, à savoir :

$$\frac{f_e}{N_T}(K-1)MN(p+1).$$

Le gain en complexité est un peu inférieur à p. Ainsi pour l'application numérique précédente, on passe de 11,26 Mips à 1,37 Mips.
Mais imposer que $\mathbf{H^T H}$ soit de Tœplitz introduit une distorsion. Elle n'est toutefois généralement pas audible [127].

D'autres algorithmes [126] ont été proposés utilisant une décomposition en valeurs singulières de la matrice $\mathbf{H}$ ou bien une recherche dans le domaine fréquentiel en remplaçant les convolutions par des produits.

2.1.6.2 Algorithmes utilisant des dictionnaires stochastiques à structure particulière Ces algorithmes imposent une structure particulière aux dictionnaires stochastiques pour diminuer le nombre de calculs, sans dégradation audible de la qualité du signal synthétique.

J. Menez [100] a utilisé des dictionnaires stochastiques de forme Tœplitz, ce qui permet d'effectuer par récurrence les calculs des dictionnaires filtrés comme pour le dictionnaire adaptatif.

Gerson [65] a proposé l'algorithme VSELP *Vector Sum Excited Linear Predictive coder*, dans lequel l'ensemble des vecteurs des dictionnaires adaptatifs est généré à partir d'une base formée d'un petit nombre k_b de vecteurs. Les M vecteurs d'un dictionnaire sont constitués par combinaison linéaire à coefficients binaires (+1 ou -1) des vecteurs de cette base. Avec une base de k_b vecteurs, on génère ainsi un dictionnaire de $M = 2^{k_b}$ vecteurs. Les coefficients de la combinaison linéaire correspondant au vecteur $\mathbf{c}_k$ s'obtiennent par codage de Gray de la valeur k. Ainsi quand on passe de $\mathbf{c}_k$ à $\mathbf{c_{k+1}}$ un seul coefficient change. Le filtrage du dictionnaire se ramène alors essentiellement au filtrage des k_b vecteurs de base. Le filtrage des autre vecteurs s'en déduit par récurrence en utilisant le code de Gray.

Adoul et Laflamme [5, 81] ont proposé des dictionnaires algébriques.

Le standard américain FS1016 à 4800 bps utilise un dictionnaire stochastique ternaire, c'est-à-dire ne contenant que les valeurs $+1, 0, -1$ et possédant une structure de Tœplitz.

2.2 Premier algorithme proposé de recherche de la meilleure excitation par méthode multi-étapes et sous-échantillonnage

On a vu dans la section précédente que la complexité de l'algorithme CELP se situe à 2 niveaux :
- Pour l'algorithme optimal, le nombre de combinaisons de vecteurs à tester est très grande. Ce problème est résolu par l'algorithme sous-optimal itératif standard et ses variantes améliorées.
- Le calcul des dictionnaires filtrés, et des termes $\alpha(k), \beta(k)$.

Le 1^{er} algorithme que M. Mauc et moi-même avons développé se situe au $2^{\text{ème}}$ niveau. Son objectif est de simplifier le calcul des dictionnaires filtrés et des termes d'autocorrélation $\alpha(k)$ sans dégrader de façon audible la qualité de signal codé.

Il est de portée générale c'est à dire qu'il peut s'appliquer quelque soit le type de dictionnaire utilisé.

2.2.1 Idée de base

Les termes $\alpha(k)$ représentent l'énergie du signal synthétique $\hat{\mathbf{p}}$. Ce signal est obtenu par filtrage d'un résiduel synthétique quasi-blanc par le filtre de synthèse « perceptualisé » $H(z)$, $H(z) = \frac{W(z)}{A(z)}$.

À l'exception de certains sons non-voisés la fonction de transfert de $1/A(f)$ décroît rapidement en amplitude avec la fréquence.

De plus, pour les sons ayant une partie significative de leur énergie dans les hautes fréquences, le filtre $W(f)$ a tendance à atténuer cette énergie au bénéfice des basses fréquences.

D'autre part, l'oreille est moins sensible aux distorsions sur les sons non-voisés que sur les sons voisés.

L'algorithme que nous avons mis au point tire parti de ces arguments de la manière suivante :

– Il calcule d'abord une approximation de l'énergie $\alpha(k)$ dans la plage de fréquences $[0, \frac{f_e}{2q}]$. On note $\alpha_q(k)$ ces énergies approchées.
Pour que cette approximation soit acceptable, il faut que q ne soit pas trop grand. Pour une fréquence d'échantillonnage $f_e = 8000\ Hz$, une valeur de $q = 5$ correspond à la plage de fréquence $[0, 800\ Hz]$.

Ce calcul se fait à la cadence fe/q sur les vecteurs des dictionnaires sous-échantillonnés par q. Les dictionnaires sous-échantillonnés sont calculés a priori ce qui augmente la taille de la mémoire nécessaire pour le codeur.
La réduction de complexité provient de la réduction de la fréquence de travail et de la diminution de la longueur des vecteurs après sous-échantillonnage.

– L'algorithme effectue par ailleurs une recherche multi-étapes.
À la $1^{\text{ère}}$ étape, après le calcul des énergies $\alpha_{q_1}(k)$ avec sous-échantillonnage par q_1, on calcule le critère $PC_{q_1}(k) = \frac{\beta(k)^2}{\alpha_{q_1}(k)}$ pour tous les vecteurs du dictionnaire. On ordonne les résultats et on conserve les M_1 vecteurs du dictionnaire de départ qui donnent les meilleurs résultats pour le critère $PC_{q_1}(k)$.
On note $\mathcal{C}_1$ le sous-ensemble retenu.
On répète ensuite la méthode, en calculant à nouveau les $\alpha(k)$ sur ce sous-ensemble $\mathcal{C}_1$ avec un facteur de sous-échantillonnage q_2 inférieur à q_1, ce qui revient à élargir la largeur de bande pour le calcul de l'énergie.

L'itération $n^o i$ de l'algorithme comprend donc :
Le calcul des énergies α_{q_i} avec un facteur de sous-échantillonnage donné q_i.
La recherche, dans le dictionnaire $\mathcal{C}_{i-1}$ retenu à l'étape précédente, d'un sous-ensemble $\mathcal{C}_i$ de taille M_{q_i} correspondant aux meilleurs vecteurs pour le critère $PC_{q_i}(k)$ calculé avec les énergies α_{q_i}.

Les facteurs de sous-échantillonnage diminuent à chaque étape. Et à la dernière, les calculs se font à f_e sur le dernier sous ensemble $\mathcal{C}$ retenu.

La taille des sous-dictionnaires $\mathcal{C}_i$ a été déterminée de manière expérimentale, sur une base de données de parole, de façon à ce que la meilleure séquence pour le critère $PC(k)$ sans sous-échantillonnage ait une très bonne probabilité d'appartenir au sous-ensemble retenu.

En pratique 1 ou 2 itérations de la méthode sont suffisantes.

Les termes D'intercorrélation $\beta(k)$ sont calculés, une seule fois, de manière classique, sans faire intervenir le dictionnaire filtré (voir 2.1.6.1) par la formule $\beta(k) = \langle \mathbf{f_k}, \mathbf{p} \rangle = \left\langle \mathbf{c_k}, \mathbf{H^T p} \right\rangle$.

2.2.1.1 Description du calcul des énergies par sous-échantillonnage Le terme d'énergie $\alpha_j(k)$ est l'énergie du signal synthétique $\mathbf{\hat{p}_j}$ obtenu à partir du $k^{\text{ème}}$ vecteur du dictionnaire :

$$\alpha_j(k) = \|\mathbf{\hat{p}_{j,k}}\|^2 = \mathbf{c_{j,k}^T H^T H c_{j,k}}.$$

Dans ce paragraphe, on omettra l'indice j qui représente la $j^{\text{ème}}$ itération de l'algorithme itératif standard. On notera simplement $\mathbf{\hat{p}_k} = \mathbf{H c_k}$.

L'énergie $\alpha(k)$ est calculée sur la plage de fréquence $[0, \frac{f_e}{2}]$.

Le vecteur $\mathbf{\hat{p}_k}$ est la convolution du vecteur $\mathbf{c_k}$ avec la réponse impulsionnelle $h(n)$ du filtre $H(z)$.

Le terme $\alpha_q(k)$ correspond à l'estimation de l'énergie de $\mathbf{\hat{p}}$ dans la bande $[0, \frac{f_e}{2q}]$. Cette approximation est faite en filtrant les 2 signaux $h(n)$ et $c_k(n)$ par un filtre passe-bas coupant à $\frac{f_e}{2q}$ puis en les sous-échantillonnant par q et en calculant la norme de la convolution des 2 signaux sous-échantillonnés.

Le filtrage passe-bas est réalisé par simple moyennage de q échantillons successifs.

Les vecteurs $\hat{c}_k$ et $\mathbf{h} = \begin{pmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{pmatrix}$ sous-échantillonnés ont une longueur N/q. Et en notant $\mathbf{h_q}$ et $\mathbf{c_{k,q}}$ les vecteurs sous-échantillonnés, l'approximation $\alpha_q(k)$ de l'énergie s'écrit :

$$\alpha_q(k) = \sum_{n=0}^{\frac{N}{q}-1} h_q(n) * c_{k,q}(n)$$

La figure I.10 représente le principe du calcul des énergies par sous-échantillonnage.
La figure I.11 représente l'aspect multi-étapes de la méthode.

2.2.2 Évaluation de la complexité de la méthode et de la mémoire nécessaire

Les évaluations sont faites dans le cas d'un seul dictionnaire de taille M, mais elles se généralisent sans difficultés au cas de plusieurs dictionnaires. Le contenu des dictionnaires peut aussi être adapté pour minimiser la charge de calcul.

Le tableau I.1 donne le nombre d'opérations élémentaires pour les différentes étapes de calcul dans le cas de l'algorithme classique et dans le cas de l'algorithme multi-étapes avec sous-échantillonnage.

On considère le cas d'une implantation sur DSP. De ce fait, on ne différencie pas les 2 types d'opérations élémentaires multiplications-additions et tests, car elles s'effectuent généralement en un seul temps de cycle sur un DSP.

D'autre part, on ne compte pas les divisions du calcul des $PC(k)$, car elles sont peu nombreuses, bien que classiquement elles soient 16 fois plus complexes qu'une multiplication-addition sur un DSP format fixe 16 bits.

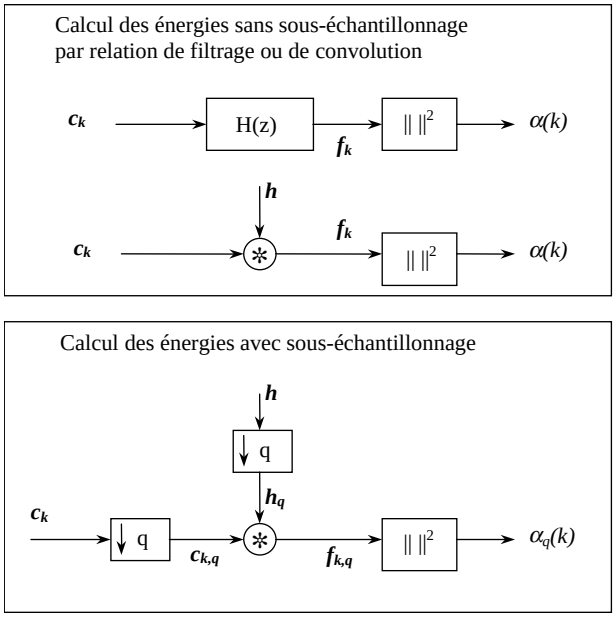


FIG. I.10 – Calcul des approximations des termes d'énergie $\alpha(k)$ dans la bande $[0, \frac{f_e}{2q}]$ avec sous-échantillonnage par Q

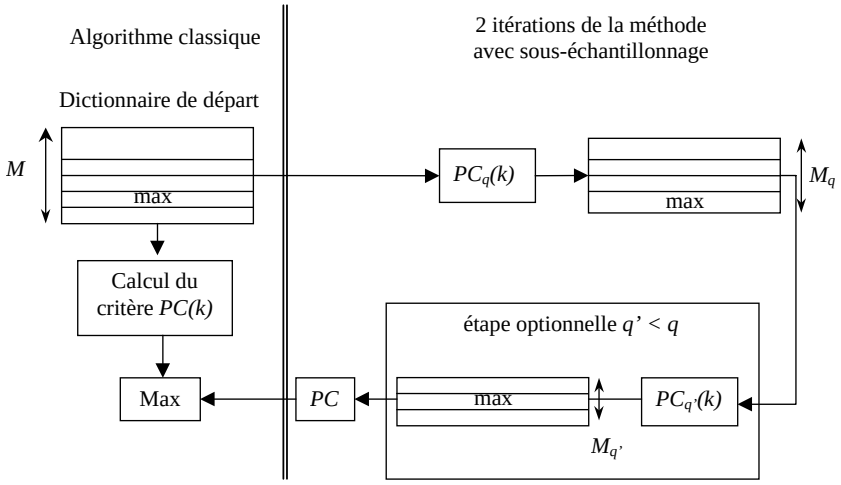


FIG. I.11 – Itération de la méthode

Algorithme classique	
Filtrage du dictionnaire	$\frac{f_e}{N_T} MNp$
Calcul des $\alpha(k)$	$\frac{f_e}{N_T} MN$
Total	$\frac{f_e}{N_T} MN\,(p+1)$
Algorithme avec sous-échantillonnage	
Calcul des convolutions	$\frac{f_e}{N_T} \frac{MN}{2q} \left(\frac{N}{q}+1\right)$
Calcul des énergies $\alpha_q(k)$	$\frac{f_e}{N_T} M \frac{N}{q}$
tri des résultats	$\frac{f_e}{N_T} M M_q$
Filtrage du sous-dictionnaire	$\frac{f_e}{N_T} M_q pN$
Total pour 1 étape de sous-échantillonnage	$\frac{f_e}{N_T} \left(\frac{M}{2} \left(\frac{N^2}{q^2}+\frac{3N}{q}\right) M_q\,(M+pN) \right)$
Total pour 2 étapes de sous-échantillonnage avec les facteurs q_1 et q_2	$\frac{f_e}{N_T} \begin{pmatrix} \frac{M}{2}\left(\frac{N^2}{q_1^2}+3\frac{N}{q_1}\right) \\ M_{q_1}\left(M+\frac{1}{2}\left(\frac{N^2}{q_2^2}+\frac{3N}{q_2}\right)\right) \\ + \\ M_{q_2}\,(Mq_1+pN) \end{pmatrix}$

TAB. I.1 – Complexité en Mips de l'algorithme classique et de l'algorithme multi-étape avec sous-échantillonnage.

On mesure le rythme des opérations en Mips.

La taille de la mémoire nécessaire est augmentée par rapport à l'algorithme classique, car il faut stocker les dictionnaires sous-échantillonnés. Dans le cas, d'un algorithme à 2 étapes de sous-échantillonnage avec les facteurs q_1 et q_2, la taille de la mémoire est multipliée par $\left(1 + \frac{1}{q_1} + \frac{1}{q_2}\right)$.

2.2.3 Détermination expérimentale de la taille des sous-dictionnaires

La taille M_q du sous-dictionnaire, pour un facteur q, a été déterminée expérimentalement. Elle a été choisie assez grande pour que le meilleur vecteur $\mathbf{c_k}$ ait une probabilité supérieure à 99% (pour la base de données de test) d'appartenir au sous-dictionnaire de taille M_q. Le meilleur vecteur $\mathbf{c_k}$ est celui qui maximise le critère $PC(k)$ calculé sans sous-échantillonnage, mais il n'est pas forcément maximum pour le critère $PC_q(k)$.

Les tests ont été faits sur un ensemble de phrases (locuteurs français masculins et féminins) de plusieurs dizaines de minutes.

Les paramètres suivants ont été utilisés dans les expériences :

f_e	=	$8000\ Hz$.
1 dictionnaire stochastique		de M vecteurs.
1 dictionnaire adaptatif		de taille $M_a = 256$.
N	=	40.
N_T	=	160.
p	=	10.

La valeur de M_q a été déterminée pour différentes valeurs de q et de M. Le tableau I.2 résume les résultats obtenus.

q / M	2	4	5	8	10
128	6	12	14	22	28
256	8	16	23	36	49
512	12	31	39	64	85
1024	15	45	66	103	155

TAB. I.2 – Valeurs de M_q pour différentes valeurs de M et de q

2.2.3.1 Application numérique Avec les paramètre précédents, et un dictionnaire stochastique de $M = 1024$ vecteurs, la complexité des algorithmes de calcul des dictionnaires filtrés et des $\alpha(k)$ est :

Algorithme classique	:	22,5 Mips
Algorithme avec 1 étape de sous-échantillonnage, $q = 5$ et $M_q = 66$	:	6,95 Mips

Le nombre d'opérations par seconde est donc divisé par 3.

Après analyse des différents termes intervenant dans l'expression de la complexité, on constate que Le gain en performances est essentiellement limité par le rapport $\frac{pN}{M_q}$.

2.2.4 Application à un dictionnaire linéaire

On appelle dictionnaire linéaire, un dictionnaire dans lequel chaque ligne de N échantillons est obtenue en prélevant N valeurs successives dans une longue séquence d'échantillons et en effectuant un décalage de S valeurs dans cette séquence pour chaque nouvelle ligne du dictionnaire.

Ainsi 2 lignes successives du dictionnaire ne diffèrent-elles que de S valeurs.

Le schéma suivant illustre la formation d'un dictionnaire linéaire, avec $S = 2$ et $N = 4$:

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	$\cdots$

$\Downarrow$

r_1	r_2	r_3	r_4
r_3	r_4	r_5	r_6
r_5	r_6	r_7	r_8
r_7	r_8	r_9	r_{10}
r_9	r_{10}	r_{11}	r_{12}
r_{11}	r_{12}	r_{13}	r_{14}
r_{13}	$\cdots$	$\cdots$	$\cdots$

TAB. I.3 – Formation d'un dictionnaire linéaire avec un décalage $S = 2$

Le dictionnaire adaptatif est un dictionnaire linéaire avec $S = 1$ formé à partir de la séquence des échantillons de résiduels synthétiques passés.

Une structure de dictionnaire linéaire est intéressante car elle permet de calculer le dictionnaire filtré par récurrence :

$$f_i(n) = f_{i+1}(n-S) + \sum_{j=0}^{S-1} h(n-j)c_i(j) \text{ où les } c_i(n) \text{ sont les composantes du vecteur } \mathbf{c_i}.$$

Cette récurrence n'est intéressante que si $S < p$.

De plus, ce type de dictionnaire occupe une place réduite en mémoire. Un dictionnaire linéaire de longueur M avec un décalage S et des vecteurs de N échantillons occupe $S(M-1)+N$ mots mémoire.

Aussi cette structure de dictionnaire a-t-elle été appliquée même aux dictionnaires stochastiques dans certaines normes. Par exemple dans la norme américaine FS1016 à 4800 bps, le dictionnaire stochastique est un dictionnaire linéaire avec $S = 2$.

La méthode avec sous-échantillonnage peut aussi s'appliquer efficacement à ce type de dictionnaire en prenant $q = S$.

Avec les notations définies précédemment, on peut comparer la complexité des 2 approches :

– Approche classique, utilisant la structure linéaire du dictionnaire, pour calculer le dictionnaire filtré :

filtrage	=	$\frac{f_e}{N_T}(pN + (M-1)NS)$.
Calcul des énergies	=	$\frac{f_e}{N_T}NM$.
Total	=	$\frac{f_e}{N_T}N(P + SM + M - S)$

– Approche avec 1 étape de sous-échantillonnage et $q = S$:

Convolution de la 1ère séquence	=	$\frac{f_e}{N_T}\frac{N}{2q}\left(\frac{N}{q}+1\right)$.
Filtrage avec récurrence de mise à jour	=	$\frac{f_e}{N_T}\frac{N}{q}(M-1)$.
Calcul des énergies	=	$\frac{f_e}{N_T}\frac{N}{q}M$.
Tri	=	$\frac{f_e}{N_T}M_qM$.
Filtrage et calcul des énergies	=	$\frac{f_e}{N_T}M_qN(p+1)$.
Total	=	$\frac{f_e}{N_T}\left(\frac{N^2}{q^2}-\frac{N}{2q}+\frac{2MN}{q}+M_q\left(M+N(p+1)\right)\right)$.

Applications numériques :

Pour $N = 60$, $M = 1024$, $q = S = 5$, la complexité est divisée par 2,7.
Pour $N = 60$, $M = 512$, $q = S = 2$, la complexité est divisée par 4/3.

2.3 Algorithme utilisant la structure des dictionnaires d'excitation ternaires

Certaines normes, comme la norme FS1016, utilisent des dictionnaires d'excitation ternaires, c'est-à-dire dont les éléments ne peuvent prendre que 3 valeurs : ± 1 et 0.

Dans le cas de la norme FS1016, ce dictionnaire contient par ailleurs 75% de valeurs nulles, ce qui permet de simplifier significativement les calculs.

Nous avons développé un algorithme tirant profit de la structure ternaire du dictionnaire pour calculer rapidement les termes d'intercorrélation $\beta(k)$.

Différentes variantes de cet algorithme ont été proposées à peu près simultanément par d'autres auteurs [62].

Le terme d'intercorrélation $\beta(k)$ s'écrit :

$$\begin{aligned}\beta(k) &= \langle \mathbf{f_k}, \hat{\mathbf{p}} \rangle = \langle \mathbf{c_k}, \mathbf{q} \rangle .\\ \mathbf{q} &= \mathbf{H^T p}.\\ \beta(k) &= \sum_{n=0}^{N-1} q(n)c_k(n).\end{aligned}$$

2.3.1 Principe de l'algorithme

L'idée de base est que les différents produits scalaires $\beta(k)$ contiennent des sections communes. Par exemple si $\beta(1) = q_1 - q_2 + q_4$ et $\beta(2) = q_1 - q_2 + q_6 - q_{12}$, il suffit de calculer une seule fois $tmp = q_1 - q_2$ et de l'utiliser pour calculer $\beta(1) = tmp + q_4$ et $\beta(2) = tmp + q_6 - q_{12}$.

Dans ce but, nous avons découpé les vecteurs $\mathbf{c_k}$ et $\mathbf{q}$ en sections de longueur $L < N$. On note $\mathbf{sq_i}$ la $i^{\text{ème}}$ section de $\mathbf{q}$ et de même $\mathbf{sc_{k,i}}$ la $i^{\text{ème}}$ section de $\mathbf{c_k}$.

$$\begin{aligned}\mathbf{q} &= \left[\mathbf{sq_1}, \mathbf{sq_2}, \cdots, \mathbf{sq_{\frac{N}{L}}}, \right].\\ \mathbf{c}_k &= \left[\mathbf{sc_{k,1}}, \mathbf{sc_{k,2}}, \cdots, \mathbf{sc_{k,\frac{N}{L}}}, \right].\end{aligned}$$

Les vecteurs $\mathbf{sc_{k,i}}$ et $\mathbf{sq_i}$ sont des vecteurs de longueur $\frac{N}{L}$. Et on peut écrire :

$$\beta(k) = \sum_{i=1}^{\frac{N}{L}} \mathbf{sc_{k,i}^T q_i}$$

Il existe 3^L vecteurs $\mathbf{sc_{k,i}}$ différents puisque les coordonnées de ces vecteurs sont ternaires. On note ces vecteurs $\mathbf{sc_m}$ avec $m \in [0, 3^L - 1]$.

On peut calculer, pour chaque section i, les 3^L produits scalaires $b_{i,m} = \mathbf{s}c_m^T \mathbf{q_i} \quad m \in [0, 3^L - 1]$ a priori, puis combiner ces résultats pour obtenir les $\beta(k)$ en $\frac{N}{L}$ opérations.

$$\beta(k) = \sum_{i=1}^{\frac{N}{L}} b_{i,m(i)}.$$

Il y a donc $\frac{N}{L}3^L$ produits scalaires partiels $b_{i,m}$ à calculer puis $\frac{N}{L}M$ opérations à effectuer pour obtenir les $\beta(k)$. Le calcul des produits scalaires partiels peut se faire efficacement.

2.3.1.1 Calcul des produits scalaires partiels Le calcul d'un produit scalaire $b_{i,m}$ demande L opérations. Et comme il y a 3^L produits scalaires $b_{i,m}$ différents, il faut $L3^L$opérations pour calculer tous les produits scalaires $b_{i,m}$.

Mais ces valeurs $b_{i,m}$ présentent des propriétés caractéristiques que l'on peut utiliser pour diminuer le nombre de calculs. Par exemple, à chaque valeur $b_{i,m}$ (sauf 0) correspond son inverse, ce qui divise par 2 le nombre de calculs.

De manière plus systématique, les $b_{i,m}$ peuvent se déduire les unes des autres par des relations simples. On néglige ici l'indice i pour simplifier l'écriture. On écrit m en base 3 :

$$m = \sum_{j=0}^{L-1} m_j 3^j \text{ avec } m_j = \pm 1 \text{ ou } 0.$$

Les termes b_m positifs vérifient les relations :

$$\begin{aligned} b_{3^j} &= q_{L-1-j} \\ b_{3^j-n} &= b(3^j) - b(n) \\ b_{3^j+n} &= b(3^j) + b(n) \end{aligned}$$

Le calcul des b_m positifs demande donc seulement 1 opération pour chaque b_m. Les termes négatifs s'en déduisent immédiatement. Il faut donc $\frac{3^L-1}{2} - L$ opérations pour calculer les 3^L termes b_m, pour chaque section i.

2.3.1.2 Complexité du calcul des $\beta(k)$ En conclusion, pour obtenir les M termes $\beta(k)$, il faut :

– Calculer les $\frac{N}{L}3^L$ produits scalaires partiels $b_{i,m}$, ce qui représente :

$$\frac{N}{L}\frac{3^L - 1}{2} - N \text{ opérations élémentaires.}$$

– Calculer les produits scalaires $\beta(k)$ à partir des $b_{m,i}$, ce qui demande :

$$M\left(\frac{N}{L} - 1\right) \text{ opérations.}$$

– D'où au total :

$$\frac{N}{L}\left(\frac{3^L-1}{2}+M\right)-N-M \text{ opérations élémentaires.}$$

Soit :

$$\frac{f_e}{N}\left(\frac{N}{L}\left(\frac{3^L-1}{2}+M\right)-M-N\right) \text{ Mips.}$$

Il faut comparer ce résultat avec la complexité de l'algorithme classique pour le calcul des $\beta(k)$, à savoir $\frac{f_e}{N}M(N-1)$ Mips.

La figure I.12 représente la complexité de l'algorithme proposé pour différentes valeurs de L et pour $M = 1024$ et $M = 512$.

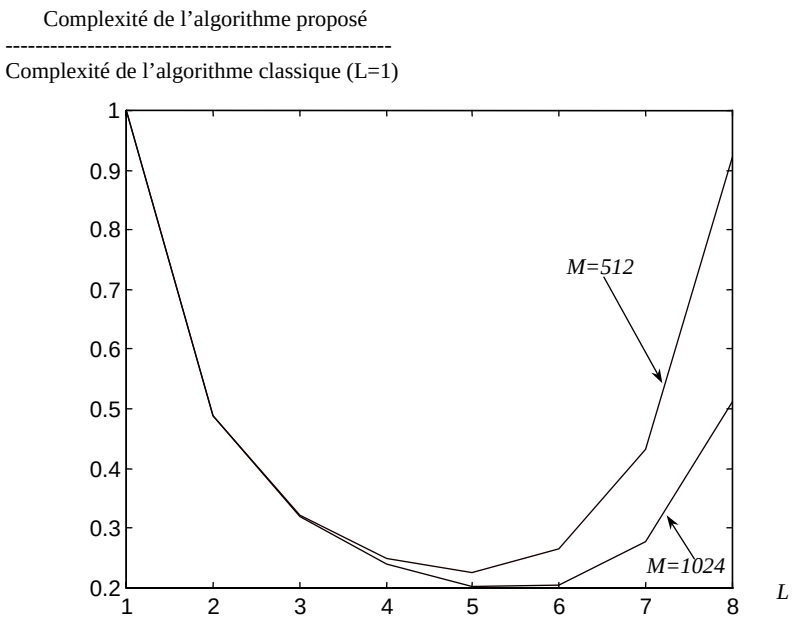


FIG. I.12 – Complexité de l'algorithme proposé pour le calcul des $\beta(k)$ dans le cas d'un dictionnaire ternaire, pour différentes valeurs de L

On constate sur cette courbe que la complexité du calcul des $\beta(k)$ est divisée approximativement par L tant que L est assez petit devant $log_3(3M)$.

2.4 Application des 2 algorithmes proposés à la norme FS1016

Nous avons appliqué les 2 algorithmes à la norme américaine FS1016 de codage CELP à 4800 bps. Et nous avons diminué la complexité globale d'un facteur 2.

Cette norme a les caractéristiques suivantes :
– Codage CELP à 4800 bps.
– $f_e = 8000\ Hz$.
– 1 dictionnaire adaptatif de 256 vecteurs.
– 1 dictionnaire stochastique, de taille $M = 512$, linéaire (avec un décalage $S = 2$), ternaire et contenant environ 75% de 0. Les séquences sont de variance unité et de moyenne nulle.
– N=60

2.4.1 Calcul des termes d'énergie $\alpha(k)$

Le calcul des termes d'énergie a été fait avec une étape de sous-échantillonnage avec $q = S = 2$ et en exploitant le fait que les dictionnaires sous-échantillonnés sont linéaires avec $S = 1$.

Nous avons trouvé qu'une valeur $M_q = 8$ était suffisante pour avoir une probabilité supérieure à 99% de conserver le meilleur vecteur parmi les M_q vecteurs du sous-dictionnaire retenu après sous-échantillonnage.

Par ailleurs, nous avons limité la durée de $h(n)$ à 20 échantillons sans dégradation notable de performances.

Nous obtenons les résultats suivants :

– Calcul des $\alpha(k)$ par l'algorithme classique : 6,9 Mips.
– Calcul des $\alpha(k)$ par l'algorithme avec sous-échantillonnage : 4 Mips.

2.4.2 Calcul des termes d'intercorrélation $\beta(k)$

Pour $L = 4, 5, 6$ nous obtenons des résultats semblables. Le gain en complexité pour le calcul des $\beta(k)$ est proche de 5.

On passe de 4 Mips pour l'algorithme classique à 0,8 Mips pour l'algorithme prenant en compte l'aspect ternaire du dictionnaire.

2.4.3 Complexité globale pour les calculs sur le dictionnaire stochastique

La complexité globale obtenue avec les 2 algorithmes proposés est de :

$$(4 + 0.8) = 4,8 \text{ Mips.}$$

Complexité qu'il faut comparer à celle de l'algorithme classique prenant en compte la structure linéaire du dictionnaire stochastique :

$$(6.9 + 4) = 10,9 \text{ Mips.}$$

La complexité a donc été divisée par 2.

2.4.4 Tests subjectifs

Nous avons effectué des tests subjectifs de la qualité du codage obtenu pour la norme FS1016 utilisant l'algorithme avec sous-échantillonnage.

Nous avons fait écouter 25 paires de phrases phonétiquement équilibrées à 30 auditeurs qui devaient choisir entre les 3 réponses : pas de préférence, préférence pour le codeur 1, préférence pour le codeur 2.

Nous avons obtenu les résultats suivants :

– Pas de préférence : 45%.
– Préférence pour le codeur 1 : 27%.
– préférence pour le codeur 2 : 28%.

La méthode avec sous-échantillonnage n'introduit pas de différence audible de la qualité, dans les conditions de test utilisées.

3 Développement d'un codeur CELP à 3200 bps

Dans le cadre d'un contrat avec la société ACSYS (en 92-93), j'ai étudié la qualité des codeurs CELP pour des débits inférieurs à 4800 bps. Ce travail a été effectué en collaboration avec Milan Jelinek (étudiant tchèque en stage à l'ESIEE) et G. Chollet.

La société ACSYS utilisait des codeurs LPC10 à 2400 bps pour transmettre de la parole sur des lignes à haute tension et souhaitait améliorer la qualité de ces liaisons tout en conservant un débit aussi faible que possible.

Nous avons mis au point un codeur à environ 3500 bps de qualité nettement supérieure à celle du codeur classique LPC10 normalisé à 2400 bps [74, 17].

Mais nous avons constaté que la qualité subjective des codeurs CELP décroît rapidement pour des débits inférieurs à 3000 bps et que la technique CELP devient alors inférieure aux approches de type vocodeur. En effet le codage CELP effectue essentiellement une quantification vectorielle de la forme d'onde et pour un débit trop faible il n'est pas possible de coder cette forme précisément. Pour les sons voisés, le signal synthétique peut présenter des harmoniques de la fréquence fondamentale jusqu'à $f_e/2$ même si le signal original n'a plus d'harmoniques au-delà d'une fréquence $f_{max} < f_e/2$, voir figure I.2.

3.1 Codage CELP à moins de 4000 bps

Le débit obtenu pour un codeur CELP résulte du codage des paramètres spectraux et des paramètres liés au codage de l'excitation.

Si on étudie le cas classique d'un codeur avec un dictionnaire adaptatif de taille M_a et un dictionnaire stochastique de taille M, on peut calculer le débit global en utilisant les notations suivantes :

N_s = nombre de bits pour coder un vecteur de paramètres spectraux.

N_g = nombre de bits pour coder un gain g_k, on suppose qu'on code sur le même nombre de bits les gains pour le dictionnaire adaptatif et le dictionnaire stochastique.

N = longueur d'une sous-trame.
N_T = longueur d'une trame.
f_e = fréquence d'échantillonnage.
N_{ST} = $\frac{N_T}{N}$ = nombre de sous-trames par trame.

Le débit global D en bits par seconde vaut :

$$D = \frac{f_e}{N_T}\left(N_S + N_{ST}\left(2N_g + log_2(M_a) + log_2(M)\right)\right) \text{ bps.}$$

Dans le codeur de la norme FS1016, 3600 bps sont alloués au codage de l'excitation et 1200 bps au codage de l'enveloppe spectrale.

Pour diminuer le débit, on peut :

– Augmenter la longueur des trames. Mais si on allonge trop cette durée, on ne code plus correctement l'enveloppe spectrale des sons rapides comme les plosives. On ne peut donc pas augmenter la durée des trames au-delà d'une trentaine de ms.

– Diminuer le nombre de sous-trames par trame. Mais lorsque la durée des sous-trames augmente, on code moins bien l'excitation, en particulier les aspects non-stationnaires.
– Diminuer le nombre et la taille des dictionnaires.
– Diminuer le nombre de bits de codage des différents paramètres (paramètres spectraux et gains). On peut par exemple, prendre en compte les dépendances inter-trames des paramètres [87].
– Classifier les trames avant de les coder, puis les coder d'une manière adaptée à leur classe avec des dictionnaires différents pour chaque classe [131, 133, 68]. Les classes utilisées sont larges, du type : voisée, non-voisées, silence, ... Cette approche permet de n'utiliser qu'un seul dictionnaire à la fois et donc de diminuer le débit. Mais la qualité diminue en présence de bruit de fond ou d'erreurs canal à cause des erreurs de classification.

3.2 Codeur proposé

Dans un codeur CELP classique à 2 dictionnaires, le dictionnaire adaptatif est surtout important pour les sons voisés et le dictionnaire stochastique pour les sons non-voisés.

Le codeur que nous avons développé utilise un dictionnaire unique de façon à diminuer le débit. Il n'effectue pas de pré-classification des trames, mais il utilise un dictionnaire de grande dimension contenant différentes catégories de vecteurs :

– des vecteurs formés d'une impulsion unique,
– des vecteurs aléatoires (équivalent du dictionnaire stochastique) blancs gaussiens,
– des séquences de résiduel passées (équivalent du dictionnaire adaptatif).

La figure I.13 représente le principe du codeur CELP développé.

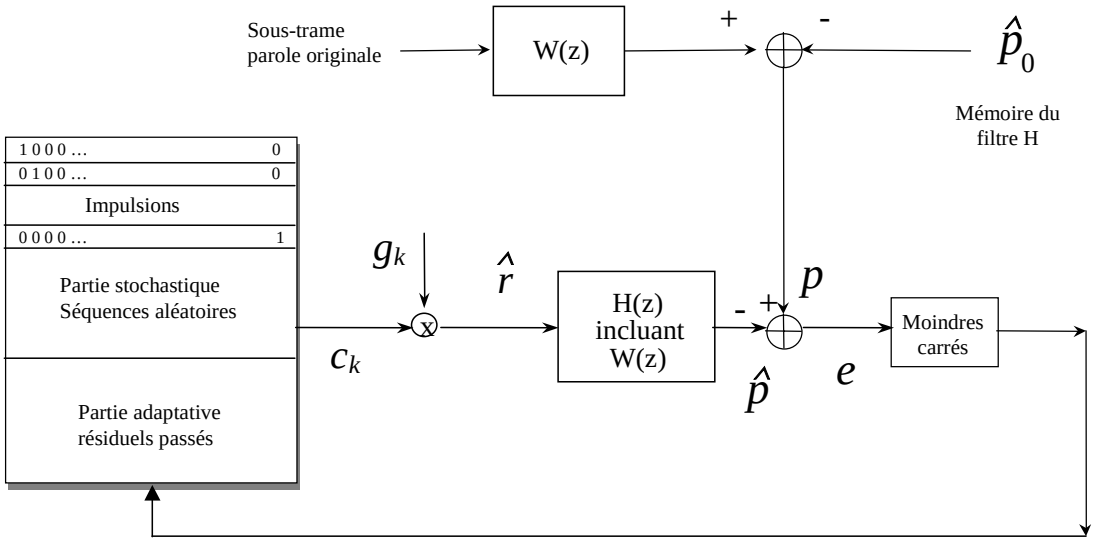


FIG. I.13 – Structure du codeur CELP proposé

L'excitation est donc formée à partir d'un seul vecteur du dictionnaire. Au lieu de pré-classifier la trame puis de la coder avec le dictionnaire adapté, on laisse le codeur CELP lui-même choisir le bon type d'excitation pour chaque trame. La complexité est plus élevée que pour un codeur à pré-classification mais on augmente la robustesse du codeur puisqu'on supprime le risque de mauvaise classification.

3.3 Conditions expérimentales et résultats

Nous avons utilisé les paramètres suivants :

f_e = 8000 Hz.
N_T = 240 (30 ms).
N = 60 4 sous-trames par trame.
M = 1024,
avec 128 vecteurs dans la partie adaptative,
836 vecteurs dans la partie aléatoire,
et 60 vecteurs dans la partie impulsions.

Les vecteurs spectraux sont les mêmes que pour la norme FS1016 (10 LSP) et sont quantifiés de la même manière.

Le débit du codeur est de 3200 bps. Il se décompose de la manière suivante :

3200 bps = (1200 + 2000) bps,

1200 bps pour les vecteurs spectraux, comme pour la norme FS1016.
Les vecteurs spectraux sont codés sur 36 bits.
1200 bps = $\frac{f_e}{N_T}36$ bps.

2000 bps pour l'excitation (au lieu de 3600 bps dans le cas de la norme FS1016).
2000 bps =$\frac{f_e}{N}(5+10)$,
avec 5 bits pour le codage du gain
et 10 bits pour le codage de l'index du dictionnaire.

Nous avons testé le codeur sur des phrases françaises équilibrées phonétiquement (environ 30 s) et prononcées par un locuteur féminin et un locuteur masculin.

le tableau I.4 précise le taux d'utilisation des différentes parties du dictionnaire.

	Impulsions	Partie adaptative	Partie stochastique
Locuteur masculin	5,2%	43,7%	51,1%
Locuteur féminin	4,9%	46,6%	48,5%

TAB. I.4 – Taux d'utilisation des différentes parties du dictionnaire

On conserve avec cette approche un codage correct de l'excitation jusqu'à des débits de 2000 bps pour l'excitation. La qualité est toutefois inférieure à celle de la norme FS1016.

D'autre part la qualité subjective obtenue est meilleure que celle du codeur travaillant avec des sous-trames 2 fois plus longues et 2 dictionnaires (un adaptatif et un stochastique).

4 Codage de la parole à très bas débit

Mes travaux portent maintenant sur le codage à très bas débit (quelques centaines de bits par seconde). Les applications envisageables sont le stockage (répondeurs statiques par exemple), la possibilité d'offrir un service de télécommunications toujours disponible même dans de très mauvaises

conditions de transmission, les « *pagers* » ou systèmes de messagerie unifiés avec présentation vocale des messages, les transmissions militaires sur les canaux HF, l'ajout d'un service vocal à une application déjà existante (système DARC/SWIFT par exemple pour les transmissions FM), certaines des applications de la synthèse vocale.

Sur ce thème, j'ai co-encadré avec Gérard Chollet et le P^r Sebesta un étudiant tchèque en thèse (Jan Černocký) (thèse en co-tutelle).

Les codeurs à très bas débit utilisent un modèle très simple de production de la parole (on parle de vocodeurs). Ce modèle permet de représenter le signal comme la convolution de deux termes généralement appelés source et filtre ou spectre, la source modélisant l'excitation vocale et le filtre la fonction de transfert du conduit vocal (liée à l'enveloppe spectrale du signal). Les paramètres de ces deux contributions sont calculés sur des trames successives de signal. Pour atteindre des débits de l'ordre de 500 bps ou moins, il faut coder la source et le spectre avec moins de 300 bps en moyenne. A ces débits, des techniques segmentales s'imposent, il faut prendre en compte les dépendances inter-trames.

Je me suis d'abord intéressée à la quantification des séquences de vecteurs spectraux de longueurs variables et j'ai développé avec J. Černocký et G. Chollet une méthode appelée MGQ (Quantification par Multi-Grammes). Mais cette méthode ne prenait pas correctement en compte la variabilité du rythme d'élocution et les valeurs de distorsions obtenues pour des débits de 200 bps étant trop grandes, nous avons introduit une pré-étape de segmentation en classes de cibles spectrales proches acoustiquement. Plus précisément, nous avons réorienté nos travaux en développant une technique de codage à très bas débit par indexation d'unités acoustiques obtenues automatiquement.

Les sections suivantes 4.1 et 4.2 présentent respectivement le travail effectué sur le codage des séquences spectrales de longueurs variables et sur le développement d'un codeur à très bas-débit par indexation d'unités acoustiques obtenues automatiquement.

4.1 Quantification de séquences spectrales de longueurs variables pour le codage de la parole à très bas débits

J'ai développé en collaboration avec Jan Černocký et G. Chollet un algorithme de codage des séquences de vecteurs spectraux à l'aide d'un dictionnaire de segments de longueurs variables que nous avons appelés multigrammes [42, 44, 43]. Nous avons appelé l'algorithme MGQ comme « *MultiGram Quantization* ».

Cet algorithme s'est révélé être très proche d'une méthode proposée précédemment par chou et Loockabaugh sous le nom de VVVQ (*Variable to Variable Vector Quantization*). J'ai proposé une nouvelle interprétation des 2 approches qui permet en particulier de donner un sens au multiplicateur de Lagrange intervenant dans le critère d'optimisation de la méthode VVVQ [23, 22].

J'ai d'autre part étudié l'influence de la limitation du retard introduit par la méthode. Et pour essayer de prendre en compte la variabilité du rythme d'élocution, j'ai proposé l'introduction de longues séquences dans le dictionnaire par interpolation linéaire des séquences courtes, en considérant que chaque séquence du dictionnaire pouvait être étirée ou au contraire comprimée en temps [23, 22].

4.1.1 Description et comparaison des méthodes VVVQ et MGQ

4.1.1.1 Méthode VVVQ Cette méthode segmente et quantifie une suite temporelle de vecteurs spectraux à l'aide d'une quantification vectorielle à dimension variable en utilisant un dictionnaire de séquences spectrales (suite de vecteurs spectraux) de longueurs variables de 1 à n vecteurs.

Les séquences du dictionnaire sont codées par un codage entropique et sont donc représentées par un nombre variable de bits dépendant de leur probabilité. Ainsi, à la fois la longueur des séquences du dictionnaire et le nombre de bits pour les coder sont variables, d'où le nom " Variable to variable Vector Quantization ". Le dictionnaire est obtenu, sur une base de données d'apprentissage, en minimisant la distorsion spectrale moyenne pour un débit spectral moyen limité. Une technique de multiplicateur de Lagrange est appliquée et le critère à optimiser s'écrit :

$$\min_{S_i \in S} \left(d_{s_i} + \lambda r_{S_i} \right). \tag{I.4}$$

Où S est l'ensemble de toutes les segmentations possibles de la base en segments de longueur inférieure à n, S_i est l'une d'elles, d_{S_i} est la distorsion correspondante, r_{S_i} le débit associé et λ le multiplicateur de Lagrange. Plus précisément :

$$d_{s_i} + \lambda r_{S_i} = \sum_{U_j \in S_i} \left(d_j + \lambda n_j \right).$$

Où U_j est le $j^{\text{ème}}$ segment de S_i, n_j le nombre de bits de codage de U_j et d_j la distorsion sur ce segment (somme des distorsions sur tous les vecteurs du segment). On suppose que le codage est réalisé de manière entropique et que :

$$n_j = -\log_2 \left(p(M_j) \right).$$

Où M_j est la séquence du dictionnaire associée à U_j et $p(M_j)$ la probabilité de cette séquence. Le dictionnaire est initialisé avec Z séquences de vecteurs et leurs probabilités.

Puis, on utilise un algorithme EM (Expectation Maximization) [58] itératif pour calculer le dictionnaire. À la $q^{\text{ème}}$ itération, le dictionnaire C_q contient Z séquences $M_{q,j}$ avec leurs probabilités $p(M_{q,j})$.

le nouveau dictionnaire C_{q+1} est calculé en 2 étapes :

- Étape 1 : Segmentation de la base de données en N segments en optimisant le critère I.4 avec l'algorithme de Viterbi. À chaque $M_{q,j}$ correspond une classe de $N_{q,j}$ séquences de la base de données codées par $M_{q,j}$.
- Étape 2 : Mise à jour du dictionnaire. $M_{q+1,j}$ est le centroïde de la classe de $M_{q,j}$ et $p(M_{q+1,j})$ est estimé par $\frac{N_{q,j}}{N}$.

4.1.1.2 Méthode de quantification par multi-grammes MGQ Comme pour la VVVQ, le principe consiste à segmenter et quantifier les séquences de vecteurs spectraux à l'aide d'un dictionnaire de segments de longueurs variables, appelés multigrammes par analogie avec les modèles de langage. La figure I.14 représente le dictionnaire de multigrammes.

La figure I.15 illustre l'analyse spectrale trame par trame et la segmentation en multigrammes.

Dans une première approche, les vecteurs spectraux étaient codés par quantification vectorielle et les multigrammes M_k étaient des séquences de 1 à n indices de quantification. Le dictionnaire était obtenu en maximisant la vraisemblance conjointe L de l'observation d'apprentissage (séquence d'indices de quantification) et de la segmentation optimale : S_{opt} [57].

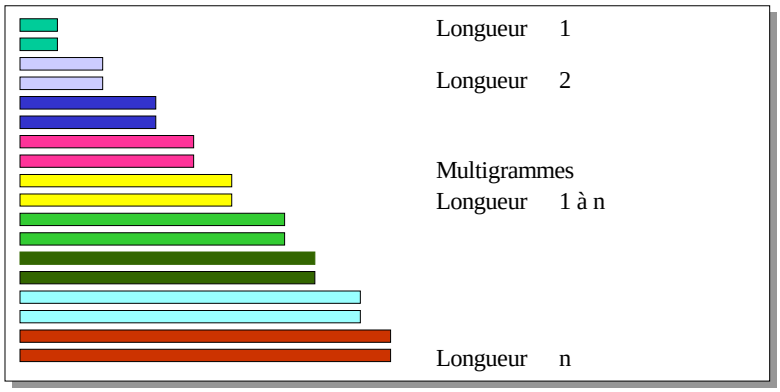


FIG. I.14 – Dictionnaire de multigrammes = séquences spectrale de longueurs variables.

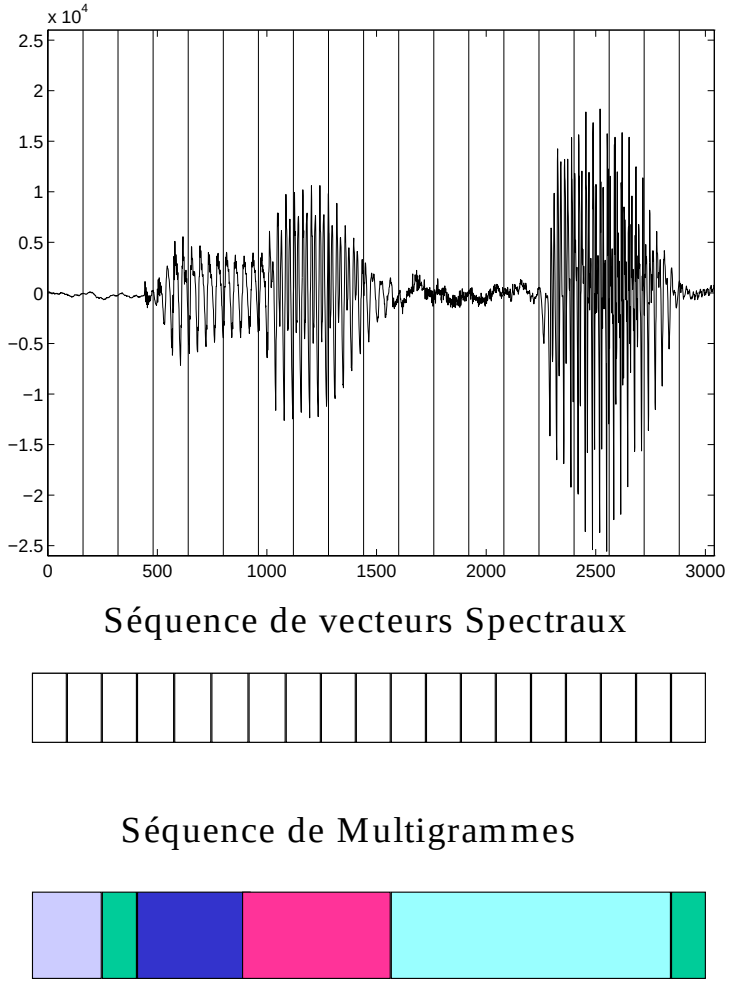


FIG. I.15 – Dictionnaire de multigrammes = séquences spectrale de longueurs variables

Les segments étant supposés indépendants, le critère consistait à maximiser :

$$L(\text{observation}, S_{opt}) = \max_{S_i \in S} \prod_{M_k \in S_i} p(M_k).$$

Le dictionnaire était initialisé avec les séquences présentes dans la base d'apprentissage et leurs nombres d'occurrences. Puis il était optimisé (probabilités seulement) à l'aide de l'algorithme EM, un codage entropique étant appliqué aux multigrammes.

Les résultats furent insuffisants pour les tailles de QV supérieures à 128, à cause de la trop grande variabilité des séquences d'indices.

Aussi, une deuxième approche a-t-elle été développée. Les vecteurs spectraux ne sont plus transformés en symboles par quantification vectorielle. Un multigramme M_k est une séquence de 1 à n vecteurs spectraux et non plus d'indices. Une chaîne de vecteurs spectraux est segmentée en segments U_k qui sont quantifiés par les multigrammes M_k de façon à maximiser le critère L' :

$$L'(\text{observation}, S_{opt}) = \max_{S_i \in S} \prod_{k} p'(M_k). \tag{I.5}$$

Où $p'(M_k)$ est la probabilité pénalisée de M_k, définie comme le produit de la probabilité de M_k avec un facteur de pénalité Q dépendant de la distance d_k entre le segment observé U_k et le multigramme qui le code M_k.

$$\begin{aligned} p'(M_k) &= p(M_k)Q(d_k) \\ d_k &= d(U_k, M_k) \\ Q(d) &= \begin{cases} 1 - \frac{d}{d_{max}} & \text{pour} \quad d \leq d_{max} \\ 0 & \text{pour} \quad d > d_{max} \end{cases} \end{aligned}$$

Où d_{max} est une constante arbitraire. Le nombre de multi-grammes de chaque longueur dans le dictionnaire initial est limité a priori. Le dictionnaire segmental est initialisé, puis il est calculé itérativement avec l'algorithme EM en optimisant le critère I.5. A chaque itération, les 2 étapes de l'algorithme EM s'effectuent comme pour la méthode VVVQ.

4.1.2 Nouvelle interprétation et comparaison des 2 méthodes

Bien que développées indépendamment, les 2 techniques sont très ressemblantes. La VVVQ est mieux formulée mathématiquement et est localement optimale en distorsion pour un débit et une structure de dictionnaire donnés.

4.1.2.1 Reformulation de la méthode MGQ L'approche MGQ apporte un éclairage différent. On peut reformuler la MGQ en considérant qu'une séquence de vecteurs spectraux est générée par une source qui émet des multigrammes (MG) de longueur variable indépendants entre eux.

On considère de plus que les vecteurs spectraux (de dimension p) de ces MG ont une densité de probabilité gaussienne de matrice de covariance $\sigma^2 I$, I étant la matrice identité de dimension (p, p).

Une autre interprétation peut être obtenue en considérant que la source émet des multigrammes constants auxquels s'ajoute un bruit blanc gaussien centré de variance σ^2.

La figure I.16 illustre cette modélisation.

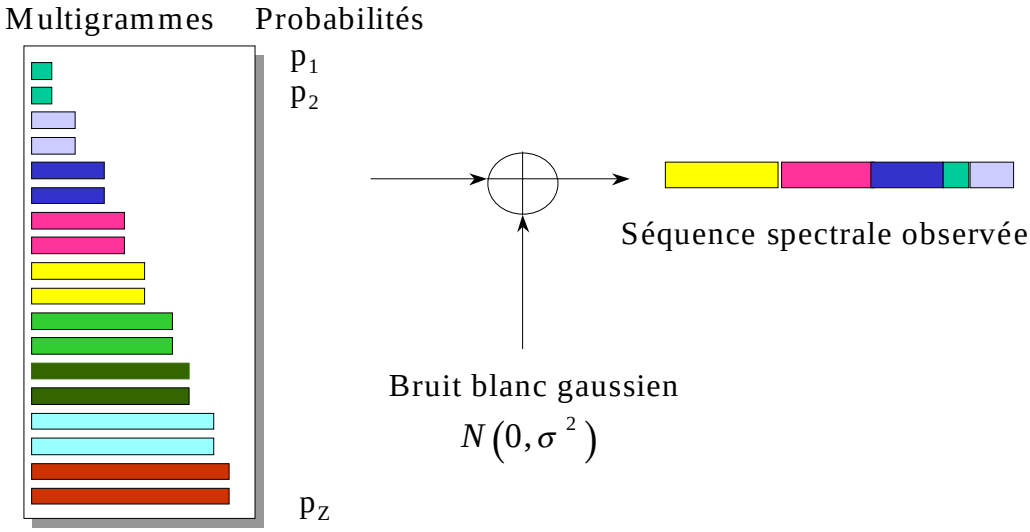


FIG. I.16 – Modèle de la source de multigramme

Les paramètres θ (multi-grammes et probabilités) de la source sont identifiés en maximisant la vraisemblance conjointe de la segmentation optimale S_{opt} et de l'observation :

$$\begin{aligned} \max_{\theta} L(\text{observation}, S_{opt}) &\iff \max_{\theta} L(S_{opt}) L(\text{observation}/S_{opt}) \\ L(S) &= \prod_k p(M_k) \\ L(\text{observation}/S) &= \prod_k p(U_k/M_k) \end{aligned}$$

Où U_k est un segment de longueur l_k de la base d'apprentissage et M_k le multigramme par lequel U_k est quantifié dans la segmentation S.

Selon le modèle gaussien proposé et en appliquant un logarithme, le critère est équivalent à :

$$\max \sum_k \left(\log\left(p(M_k)\right) - \sum_{j=1}^{l_k} \sum_{m=1}^{p} \frac{(c_{k,j,m} - m_{k,j,m})^2}{2\sigma^2} \right) \tag{I.6}$$

$$\iff \min \sum_k \left(\left(\sum_{j=1}^{l_k} d(c_{k,j}, m_{k,j}) \right) - 2\sigma^2 \log\left(p(M_k)\right) \right) \tag{I.7}$$

$c_{k,j,m}$ et $m_{k,j,m}$ sont les $m^{èmes}$ coefficients du $j^{\text{ème}}$ vecteur du segment U_k et du multigramme M_k. $d(c_{k,j}, m_{k,j})$ est la distance quadratique entre les $j^{\text{ème}}$ vecteurs de U_k et M_k.

On reconnaît dans l'équation I.7 le critère de la VVVQ avec $\lambda = 2\log(2)\sigma^2$ et une distance quadratique sur les vecteurs spectraux.

D'autre part , il est possible d'interpréter le critère arbitraire de la méthode MGQ en remarquant que pour $d \ll d_{max}$:

$$\log(p) + \log\left(1 - \frac{d}{d_{max}}\right) = \log(p) - \frac{d}{d_{max}} \text{ avec } d_{max} = 2\log(2)\sigma^2.$$

Nous avons utilisé, dans la $1^{\text{ère}}$ formulation de la méthode, une densité de probabilité triangulaire, qui est proche d'une gaussienne quand d est petit devant d_{max}.

Une différence supplémentaire entre les approches VVVQ et MGQ réside dans la mesure de distorsion spectrale utilisée. Chou & al ont travaillé avec une distance d'Itakura modifiée alors que nous avons utilisé une distance quadratique sur les coefficients cepstraux. Avec la distance d'Itakura modifiée, les interprétations précédentes doivent être appliquées au résiduel de prédiction linéaire supposé blanc et gaussien.

4.1.3 Limitation du retard

En théorie, la méthode introduit un retard égal à la durée totale du signal. Quand on limite ce retard à une valeur de k_{max} trames, les performances se dégradent. La technique classique pour limiter le retard, consiste à utiliser un buffer de k_{max} trames, à imposer des points de segmentation aux extrémités du buffer et à vider le buffer toutes les k_{max} trames .

A chaque entrée d'un vecteur spectral dans le buffer, on examinons si les n meilleures segmentations possibles du buffer depuis son origine jusqu'au dernier vecteur reçu coïncident jusqu'à une certaine position. Si un tel point existe, le buffer est vidé jusqu'à ce point. Tant que le buffer ne sature pas, la limitation du retard à k_{max} trames ne dégradent pas les performances.

Nous avons étudié les caractéristiques statistiques du remplissage du buffer pour différentes valeurs de n et de λ, de façon à obtenir une idée du retard nécessaire pour implanter la méthode.

4.1.4 Construction de longs multigrammes par interpolation

Allonger la longueur maximale possible des multigrammes entraîne l'augmentation rapide du nombre de vecteurs spectraux à estimer. Ainsi, pour 64 multigrammes par longueur, y a-t-il 8704 vecteurs à estimer pour $n = 16$ et 35088 pour $n = 32$.

Aussi, dans le but d'augmenter la longueur maximale n des segments, sans avoir à accroître la taille de la base de données d'apprentissage, avons-nous construit un dictionnaire contenant des multigrammes de longueur 1 à n à partir d'un dictionnaire de longueur maximale $n/2$, en étirant par interpolation linéaire les multigrammes de longueur $n/2$ pour obtenir les longs multigrammes de longueur $n/2 + 1$ à n, prenant ainsi en compte le fait que les mêmes séquences acoustiques peuvent être prononcées à différentes vitesses.

La figure I.17 représente la construction de longs multigrammes par interpolation.

Lors de l'apprentissage, les multigrammes de longueur $n/2$ sont actualisés à partir des segments de longueur $n/2$ qui leur ont été associés et à partir des segments de longueurs $n/2 + 1$ à n associés à ces multigrammes étirés.

Dans ce dernier cas, la mise à jour se fait par contraction linéaire des longs segments. Les probabilités sont actualisées normalement pour toutes les longueurs. On sauve à chaque itération les multigrammes de taille 1 à $n/2$ et toutes les probabilités.

La figure I.18 illustre le procédé de mise à jour des multigrammes.

Les résultats obtenus (courbes distorsion-débit), pour $n = 16$ avec étirement des multigrammes de longueur 8, sont supérieurs à ceux obtenus avec un dictionnaire non étiré de longueur maximale 12 et contenant le même nombre de vecteurs spectraux à estimer (figure I.19). Mais la complexité est augmentée.

La figure I.19 donne les courbes distorsion-débit obtenues avec et sans étirement des multigrammes.

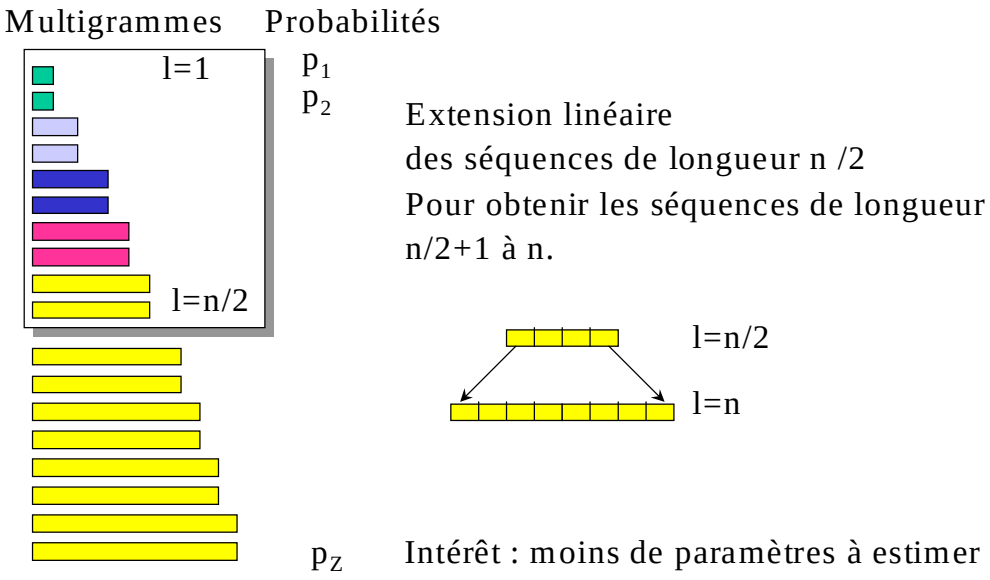
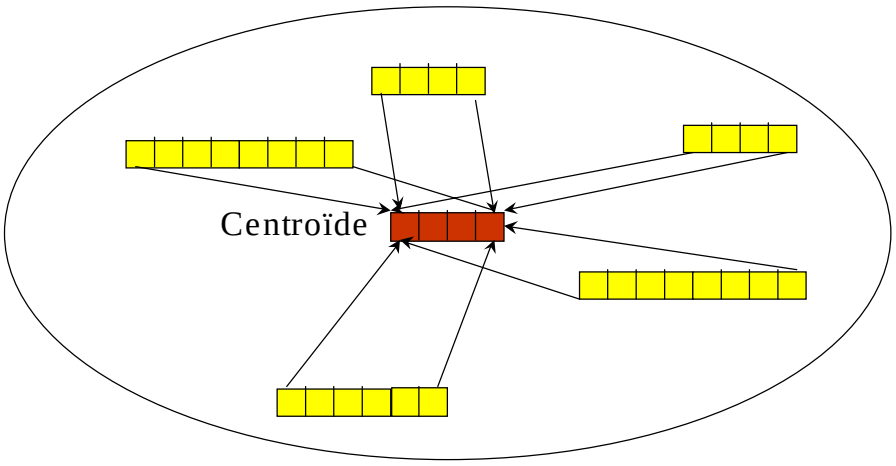


FIG. I.17 – Construction de longs multigrammes par interpolation



Classe associée à un multigramme de longueur n/2
Compression linéaire des séquences de longueur n/2+1 à n
pour actualiser les séquences de longueur n/2

FIG. I.18 – Mise à jour du dictionnaire

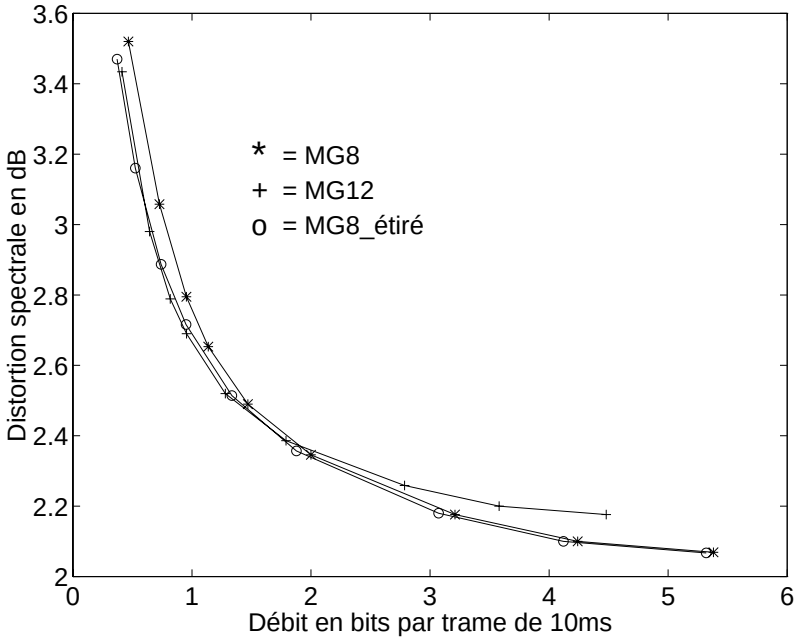


FIG. I.19 – Courbes distorsions-débits avec et sans étirement

4.1.5 Résultats expérimentaux

4.1.5.1 Définitions de la distorsion et du débit La distorsion spectrale est calculée en dB à partir d'une distance euclidienne entre les coefficients cepstraux originaux et quantifiés.

Le débit binaire est défini comme le nombre moyen de bits pour le codage d'un vecteur spectral. C'est un nombre moyen de bits par trame. Le débit binaire moyen par trame R, correspondant à dictionnaire de multigrammes avec un codage entropique, est le rapport de l'entropie H du dictionnaire à la longueur moyenne des multigrammes :

$$R = \frac{H}{\bar{l}} = -\frac{\sum_{i=1}^{Z} p(M_i)\log_2(p(M_i))}{\sum_{i=1}^{Z} p(M_i)l(M_i)}.$$

Où $l(M_i)$ et $p(M_i)$ sont la longueur et la probabilité de M_i, et Z est le nombre de multigrammes dans le dictionnaire.

4.1.5.2 Base de données utilisée Nous avons utilisé un seul locuteur de la base de données Poly-Var en français suisse. Elle contient des appels téléphoniques enregistrés sur une période de 6 mois, constitués de phrases lues, de mots épelés, de nombres, de quelques mots de contrôle et de parole spontanée. Le signal est numérisé à 8 KHz selon une loi A 8 bits. Les vecteurs spectraux sont formés de 10 coefficients LPCC calculés avec pré-emphase sur des fenêtres de Hamming de 20ms avec un recouvrement de 10ms. Le premier coefficient cepstral (lié à l'énergie) n'est pas utilisé. Le corpus a été divisé en 213270 vecteurs d'apprentissage et 122903 vecteurs de test.

4.1.5.3 Initialisation du dictionnaire Différentes initialisations du dictionnaire ont été comparées :

– Initialisation avec les multigrammes quantifiés les plus fréquents : après quantification vectorielle de la base d'apprentissage, on utilise pour chaque longueur l de multigramme les séquences quantifiées de longueur l les plus fréquentes.

- Initialisation par quantification matricielle (toutes les séquences du dictionnaire ont la même longueur) : pour chaque longueur l de multigramme, le dictionnaire de multigrammes est initialisé avec un dictionnaire de quantification matricielle à séquences de longueur l [54].
- Initialisation aléatoire naturelle : le dictionnaire de multi-grammes est initialisé avec des séquences naturelles de vecteurs spectraux choisies au hasard.

Après quelques itérations de l'algorithme EM, les 3 initialisations ont donné des résultats similaires, nous avons donc utilisé la troisième dans les expériences.

4.1.5.4 Configurations de test Nous avons testé la méthode avec différentes configurations de pour les dictionnaires de Multigrammes ou de quantification matricielle. Ces configurations sont appelées :

- MG16 :
  Quantification par multigrammes,
  $n = 16,$
  64 multigrammes par longueur,
  $0 \leq \lambda \leq 1,$
  8704 vecteurs cepstraux dans le dictionnaire.
- MQ8704 :
  Quantification Matricielle,
  avec différents dictionnaires chacun correspondant à une longueur unique de séquence l entre 2 et 20 vecteurs,
  tous les dictionnaires contenant 8704 vecteurs cepstraux.
  Ainsi, pour l=8, y a-t-il 1088 séquences de 8 vecteurs dans le dictionnaire et 544 séquences de 16 vecteurs pour $l = 16$.
- MQ1, MQ2, MQ4 :
  3 Quantifications matricielles avec codage entropique des séquences,
  et des dictionnaires de 8704 séquences de longueur respectives 1, 2, 4,
  contenant 8704, 17408 and 34816 vecteurs cepstraux,
  $0 \leq \lambda \leq 1$.
- MG8, MG12, MG8_étiré :
  quantifications par multigrammes,
  avec respectivement $n = 8, 12, 16$,
  et 113, 64, 113 multigrammes par longueur,
  $0 \leq \lambda \leq 1$.

  Les longs multi-grammes (l = 9 à 16) de MG8_étiré sont obtenus en étirant les séquences de longueur 8. Il y a 4992 vecteurs à estimer pour MG12 et 4068 pour MG8 et MG8_étiré.

4.1.5.5 Limitation du retard, remplissage du buffer La figure I.20 représente les fonctions de répartition du remplissage du buffer (nombre de trames dans le buffer), pour MG16, lorsque l'on utilise la technique décrite en 4.1.3, pour $0 \leq \lambda \leq 0.5$.
Par exemple, pour MG16, quand $0 < \lambda < 0.1$, un buffer de $k_{max} = 40$ trames (retard maximal de 400 ms) donne des résultats équivalents à ceux obtenus avec un retard illimité.
Pour $0.1 < \lambda < 0.2$, un buffer de 70 trames est suffisant.

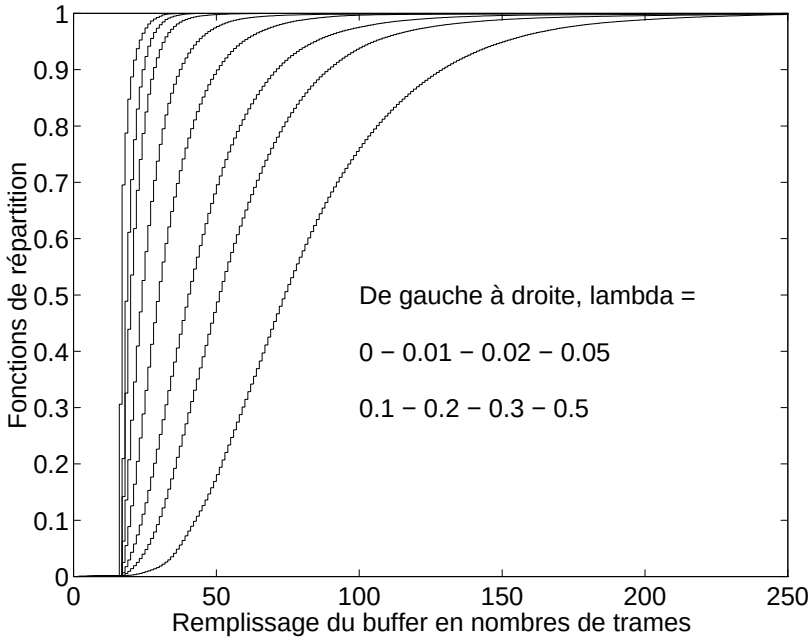


FIG. I.20 – Fonctions de répartition du remplissage du buffer pour $0 \leq \lambda \leq 0.5$, $n = 16$, MG16

4.1.5.6 Comparaison de la Quantification par multigrammes (VVVQ, MGQ) et de la Quantification Matricielle (MQ) Chou & al ont comparé la VVVQ et l'ECMQ (Entropy Constrained MQ) pour une même complexité.

Mais ils ne purent pas estimer les grands dictionnaires de MQ pour des longueurs de séquences supérieures à 4 à cause de la taille limitée de la base de données utilisée.

Aussi avons nous de plus comparé les 2 approches pour un même nombre de vecteurs spectraux dans les dictionnaires (configuration MG16 vs MQ8704).

La figure I.21 donne les courbes distorsion-débit obtenues sur la base de test avec les configurations MG16, MQ8704, MQ1, MQ2, MQ4.

Pour les petits débits spectraux (moins de 2 bits/trame, 200 bits/s), la quantification par multigrammes est supérieure à la quantification matricielle. Mais, quand la comparaison est faite pour un même nombre de vecteurs cepstraux dans les dictionnaires de MGQ ou de MQ, le gain en performance est assez faible pour une augmentation significative de la complexité.

Le principal défaut de cette approche est de ne pas tenir suffisamment compte de la variabilité du rythme d'élocution. L'introduction de la possibilité d'étirer ou de comprimer les séquences du dictionnaire n'a pas suffisamment amélioré les performances.

Aussi avons-nous pensé à appliquer la méthode non plus directement sur les séquences de vecteurs spectraux mais sur les cibles spectrales obtenus après décomposition temporelle. Nous avons ainsi réorienté notre recherche vers le développement d'un codeur de parole très bas-débit travaillant par reconnaissance d'unités acoustiques obtenues de manière non-supervisée.

4.2 Codage à très bas débit par indexation d'unités acoustiques obtenues automatiquement

Nous avons développé une technique de codage à très bas débit par indexation d'unités acoustiques obtenues automatiquement. Le codeur effectue une reconnaissance de segments acoustiques

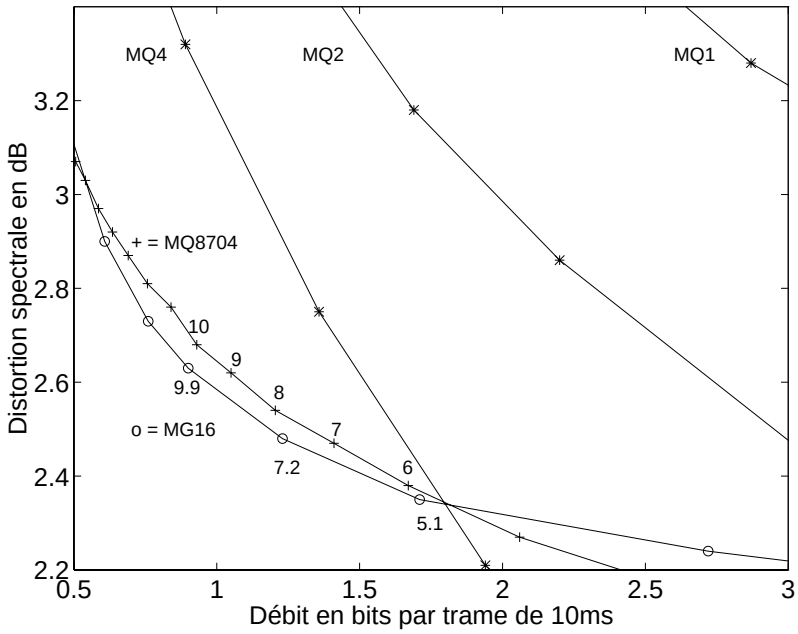


FIG. I.21 – Courbes distorsions-débits. Les nombres le long des courbes MQ8704 et MG16 représentent respectivement les longueurs des séquences ou les longueurs moyennes.

dans la phase d'analyse et une synthèse de parole à partir d'une suite d'indices de segments dans le décodeur. Le codeur réalise une transcription symbolique du signal de parole à partir d'un dictionnaire d'unités élémentaires de tailles variables. Dans les vocodeurs dits phonétiques, ces unités sont des unités linguistiques (comme des phonèmes, des transitions entre phonèmes, des syllabes). Mais ces codeurs phonétiques nécessitent la transcription phonétique du corpus d'apprentissage, tâche lourde et sujette aux erreurs qui doit être effectuée pour chaque nouvelle langue. Pour cette raison, nous avons choisi une approche utilisant des unités acoustiques obtenues de manière non supervisée sur un corpus d'apprentissage non étiqueté phonétiquement [24, 48, 50, 40, 49, 45].

L'approche consiste à déterminer de manière automatique un jeu d'unités à partir de signaux de paroles bruts sans a priori sur le contenu linguistique de ces unités, en s'intéressant uniquement à leur contenu acoustique.

On peut distinguer 2 types d'unités acoustiques :

- Les unités de codage qui sont utilisées par le codeur. Le codeur segmente le signal original en une suite d'unités de codage et transmet au décodeur les indices des unités reconnues. Ces unités de codage doivent permettre un codage à très faible débit (typiquement de 100 à 500 bps), ne pas être trop nombreuses pour faciliter la reconnaissance et ne pas être trop longues pour limiter le retard introduit par le codage.
  Ces unités sont modélisées par des modèles HMM pour faciliter leur reconnaissance.
- Les unités de synthèse qui sont utilisés par le décodeur pour fabriquer le signal de parole synthétique. Ces unités peuvent être du même type que celle de codage, mais ce n'est pas obligatoire. Si on envisage une technique de synthèse par concaténation d'unités élémentaires, il est utile

d'utiliser des unités qui se concaténent bien, c'est-à-dire, par exemple, qui commencent et se terminent par des zones stables.

Les unités de synthèse sont représentées par des modèles qui permettent des modifications simples de la prosodie. On peut envisager des unités de synthèse formées d'un signal temporel modélisé par un modèle HNM[20].

Les unités de codage étant obtenues de manière non-supervisée, on peut envisager d'appliquer la technique indépendamment de la langue et du locuteur dans différentes applications de traitement de la parole autre que le codage, par exemple la vérification du locuteur ou la reconnaissance de parole. Pour cette raison, nous avons appelé cette approche **ALISP**, comme « Automatic Language Independent Speech Processing ». Par la suite, les unités de codage obtenues automatiquement seront parfois appelée unités ALISP.

Jan Černocky a utilisé ces unités ALISP dans les expériences NIST de vérification du locuteur [49].

Pour utiliser ces unités en reconnaissance de la parole, il faut soit établir une correspondance entre les phonèmes et les unités ALISP, soit générer un dictionnaire de prononciation utilisant ces unités. Cette dernière approche a été utilisée par Fukada [63].

Cette section est décomposée en 6 parties :
- Détermination du jeu d'unités acoustiques de manière non-supervisée.
- Codeur, étape de reconnaissance.
- Décodeur, étape de synthèse.
- Expériences réalisées et résultats.
- Perspectives, projet RNRT SYMPATEX.
- Correspondances des unités acoustiques obtenues automatiquement avec les phonèmes.

4.2.1 Détermination du jeu d'unités acoustiques de manière non-supervisée, unités ALISP

Cette section présente la détermination du jeu d'unités de codage, ou unités ALISP.

La détermination du jeu d'unités ALISP se fait en 3 étapes principales :

- La $1^{\text{ère}}$ étape effectue une segmentation initiale du corpus à l'aide d'une décomposition temporelle (technique introduite par Atal [6]) qui décompose la séquence de vecteurs spectraux en cibles spectrales reliées entre elles par des fonctions d'interpolation.
- Dans la $2^{\text{ème}}$ étape nous classifions par quantification vectorielle les segments obtenus. Nous obtenons ainsi un nombre réduit de classes de cibles spectrales (typiquement 64).
- Dans la $3^{\text{ème}}$ étape, nous modélisons les classes obtenues à l'étape $n^o 2$ par des modèles stochastiques HMM[21].

  Une fois les paramètres des modèles identifiés, nous re-segmentons la base de données en utilisant ces HMM et nous utilisons cette nouvelle segmentation pour réestimer les paramètres des modèles. On effectue quelques itérations de segmentation-réestimation des modèles. L'utilisation de cette approche itérative augmente la fonction de vraisemblance des modèles et améliore la cohérence acoustique des classes obtenues. En général 5 itérations sont suffisantes.

À la fin de la phase d'apprentissage, nous disposons d'un dictionnaire de N unités acoustiques caractérisées par N modèles HMM, et d'une transcription du corpus d'apprentissage utilisant ces unités.

[20] HNM=Harmonic plus Noise Model.

[21] HMM = Hidden Markov Model.

La détermination du jeu d'unités acoustiques est lourde en calcul, mais elle n'a pas à être effectuée en temps réel.

La longueur moyenne des unités acoustiques ainsi obtenues dépend des paramètres utilisées dans la décomposition temporelle. Nous avons réglé ces paramètres de manière à obtenir en moyenne une quinzaine de cibles ou unités acoustiques élémentaires par seconde, ce qui correspond à un débit phonétique moyen.

Nous avons par ailleurs, essayé de déterminer des unités plus longues en regroupant par des techniques statistiques (multigrammes) plusieurs unités acoustiques élémentaires. Le principe utilisé est de rechercher les séquences d'unités acoustiques les plus probables dans la base de données, ces séquences étant de longueurs variables.

Ce travail peut se faire à 2 moments différents, après l'étape 2 sur les classes de quantification vectorielle de la décomposition temporelle, ou après l'étape 3 sur les indices des modèles HMM. La première solution conduit ensuite à entraîner un grand nombre de modèles HMM correspondant aux nouvelles unités allongées, ce qui demande une base de données assez grande. La $2^{\text{ème}}$ est plus intéressante de ce point de vue.

La figure I.22 résume les différentes étapes de la détermination des unités acoustiques. Les boîtes en pointillés, intitulées « Multigrammes », représentent les 2 positions possibles pour les phases d'allongement des unités acoustiques.

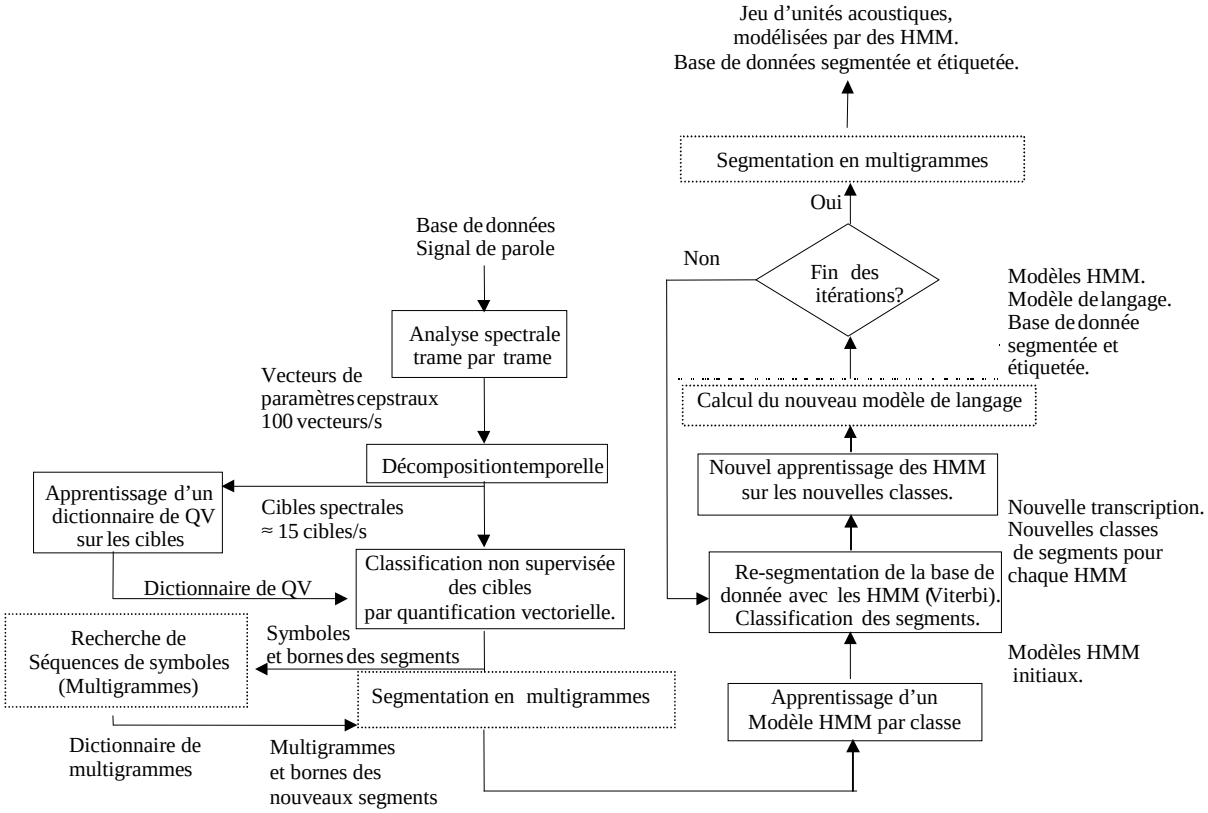


FIG. I.22 – Détermination du jeu d'unités acoustiques.

4.2.1.1 Analyse spectrale, paramétrisation La $1^{\text{ère}}$ étape du traitement est l'analyse spectrale des signaux de parole de la base de données. Elle est effectuée trame par trame.

Nous avons effectué une analyse par prédiction linéaire dont nous avons déduit des coefficients cepstraux LPCC[22].

Nous avons utilisé des coefficients cepstraux parce que la distance euclidienne entre 2 jeux de coefficients cepstraux $\mathbf{c_1}, \mathbf{c_2}$ correspond à la distance logarithmique entre les enveloppes spectrales associées :

$$\sum_{k=0}^{+\infty}(c_1(k) - c_2(k))^2 = \frac{1}{f_e}\int_{-f_e/2}^{f_e/2}(S_1(f) - S_2(f))^2 df.$$

Cette distance spectrale est assez bien corrélée avec la perception auditive. Les coefficients cepstraux sont donc intéressants puisqu'une simple distance euclidienne permet d'obtenir une bonne mesure de la distance perceptuelle de 2 spectres.

On note P la dimension des vecteurs cepstraux $\mathbf{c(n)}$, où n représente la trame numéro n.

Une détection d'activité vocale est par ailleurs réalisée, de façon à segmenter le signal en sections de durée limitée (inférieure à 1 trentaine de seconde).

Nous avons de plus soustrait le vecteur cepstral moyen fichier par fichier, pour diminuer l'influence des variations de conditions d'enregistrement.

4.2.1.2 Décomposition temporelle La décomposition temporelle est une technique de segmentation introduite par Atal [6] en 83 et affinée par F. Bimbot [34].

L'idée sous-jacente est la prise en compte des phénomènes de co-articulation. Les phonèmes successifs ne sont pas émis indépendamment les uns des autres. À un instant donné correspondant à l'émission d'un phonème, le locuteur anticipe le phonème suivant et l'articulation du phonème actuel est influencée par les phonèmes précédents et suivants.

La décomposition temporelle (DT) intègre ces phénomènes de co-articulation, en modélisant une suite de vecteurs spectraux (ou cepstraux) à l'aide de cibles spectrales reliées par des fonctions d'interpolation qui se recouvrent partiellement.

On note :

- $\mathbf{c(n)}$ le vecteur spectral correspondant à la trame numéro n.
- $\mathbf{X}$ la matrice formée de N vecteurs cepstraux successifs. Elle est de dimension (P, N)
- $\mathbf{Y}$ la matrice approximation de $\mathbf{X}$.
- $\mathbf{\Phi}$ la matrice des fonctions d'interpolation.
- $\mathbf{G}$ la matrice des cibles spectrales.

On décompose la matrice $\mathbf{X}$ en un nombre limité d'événements spectraux caractérisés par une cible spectrale et un fonction d'interpolation.

On modélise, pour cela, la matrice $\mathbf{X}$ de N vecteurs spectraux par une matrice Y, correspondant à un nombre réduit $M < N$ de cibles $\mathbf{G}$ reliées par des fonctions d'interpolation $\mathbf{\Phi}$.

La matrice $\mathbf{Y}$ s'écrit sous la forme :

$$\mathbf{Y} = \mathbf{G\Phi}.$$

La matrice $\mathbf{G}$ est de dimension (P, M) et la matrice $\mathbf{\Phi}$ de dimension (M, N).

[22] LPCC = Linear Prediction Cepstral Coefficient.

À chaque cible $\mathbf{g}_k$ (colonne k de $\mathbf{G}$) correspond une fonction d'interpolation $\Phi_k(n)$ (ligne k de la matrice $\mathbf{\Phi}$) fonction du temps, c'est-à-dire du numéro de trame n.

Les fonctions d'interpolation (FI) sont contraintes à être concentrées en temps.

La décomposition temporelle commence par le calcul des fonctions d'interpolation. Dans la méthode proposée par F. Bimbot, on répète le même traitement à chaque instant trame $n \in [0, N-1]$. On recherche à chaque temps n une fonction Φ combinaison linéaire des vecteurs spectraux. Cette fonction doit être compacte, c'est-à-dire que son énergie doit être maximale dans une sous-fenêtre $w(t)$ d'un intervalle de temps $[t_1, t_2]$ correspondant à quelques trames (typiquement 5).

Plus précisément, une décomposition en valeurs singulières est effectuée sur la matrice locale $\mathbf{X}_n$ des vecteurs spectraux :

$$\mathbf{X}_n = \mathbf{U^T D V}.$$

et les $p << P$ premières lignes $\mathbf{u}_i$ de U associées aux vecteurs propres de plus forte contribution énergétique sont conservées. La valeur p est comprise entre 3 et 5. La fonction Φ est recherchée comme une combinaison linéaire des vecteurs orthonormés $\mathbf{u}_i$.

$$\Phi(t) = \sum_{i=1}^{p} b_i \mathbf{u}_i(t).$$

La sous-fenêtre $w(t)$ est rectangulaire et centrée sur le milieu de l'intervalle $[t_1, t_2]$. On maximise le critère de compacité :

$$\sum_{t_1}^{t_2} w(t)\Phi^2(t) / \sum_{t_1}^{t_2} \Phi^2(t).$$

La fonction $\Phi(t)$ obtenue en maximisant ce critère est lissée, normalisée à 1, et tronquée en temps (seuil).

On répète ensuite la procédure. À partir de la fonction $\Phi(t)$ obtenue, on définit un nouvel intervalle de calcul $[t_1^{'}, t_2^{'}]$ plus centré sur $\Phi(t)$ et de largeur agrandie ou diminuée selon la largeur de $\Phi(t)$. Le même processus est itéré jusqu'à ce que les fenêtres trouvées n'évoluent plus. On calcule alors les intercorrélations entre fonctions d'interpolation et si cette intercorrélation dépasse un seuil prédéterminé, les fonctions d'interpolation correspondantes sont confondues.

Après l'obtention des fonctions d'interpolation, on calcule les cibles et on affine les fonctions d'interpolation de manière itérative.

On part d'une première valeur de $\mathbf{\Phi}$ et on en déduit les cibles, puis connaissant les cibles on calcule $\mathbf{\Phi}$, etc.

À partir des fonctions Φ on déduit les cibles à l'aide de la pseudo-inverse $\mathbf{\Phi}^{\#}$ de $\mathbf{\Phi}$:

$$\mathbf{G} = \mathbf{X}\mathbf{\Phi}^{\#}.$$

De même à partir de $\mathbf{G}$ on obtient une nouvelle valeur de $\mathbf{\Phi}$ par :

$$\mathbf{\Phi} = \mathbf{G}^{\#}\mathbf{X}.$$

Nous avons effectué La décomposition temporelle à l'aide des logiciels dt95 développés et fournis par F. Bimbot [33] que je remercie très chaleureusement.

La figure[23] I.23 illustre les résultats obtenus avec la DT sur un signal de parole.
Un débit moyen d'environ 15 événements par seconde, est obtenu avec la méthode, ce qui correspond à un débit phonétique moyen.

[23] J'ai emprunté cette figure à Jan Černocky que je remercie.

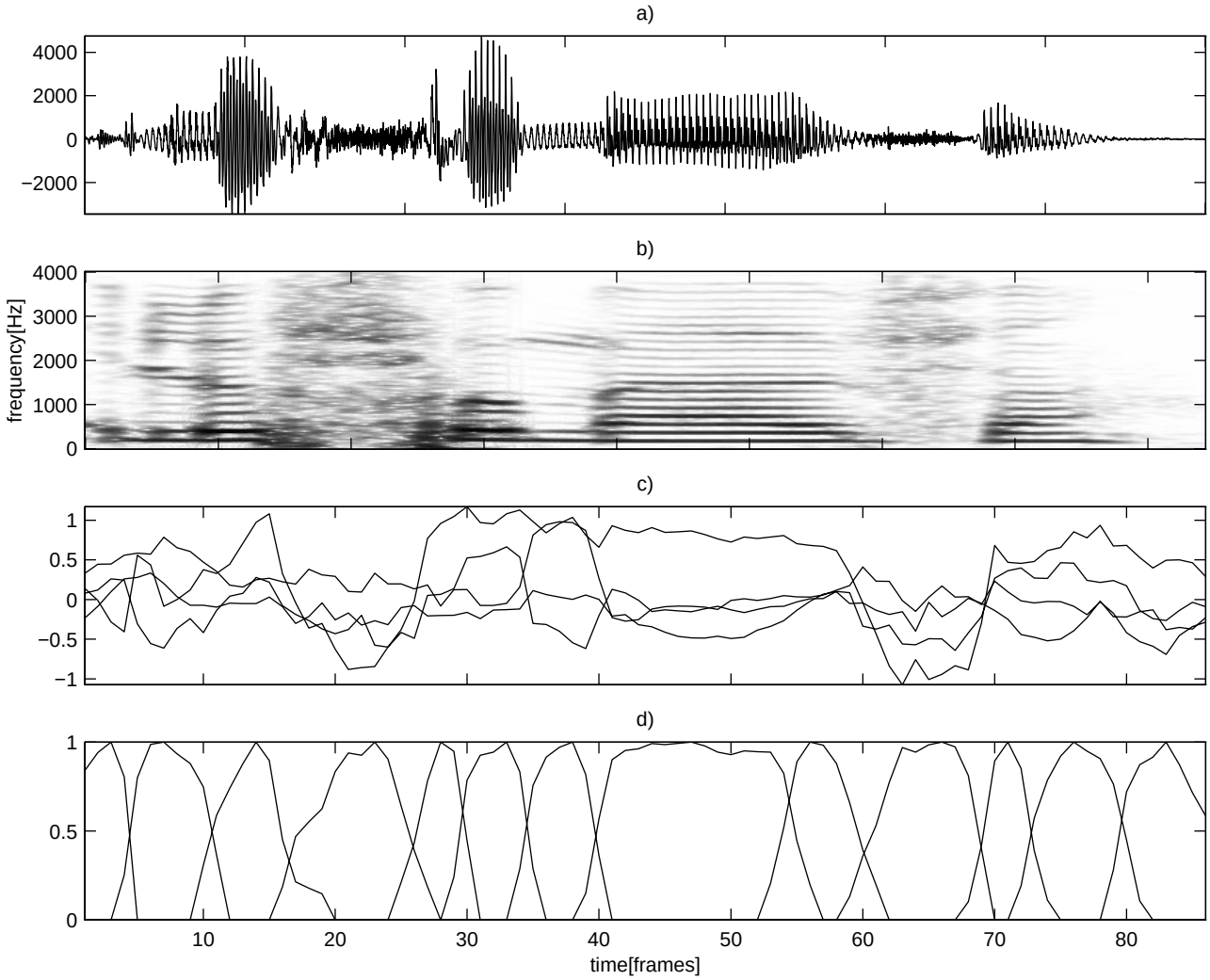


FIG. I.23 – Exemple de résultat de la décomposition temporelle. 1. Signal, 2. Spectrogramme, 3. Trajectoires des 4 premiers coefficients LPCC, 4. Fonctions d'interpolation de la DT

4.2.1.3 Classification des cibles par quantification vectorielle Après l'étape de décomposition temporelle, la base de données est segmentée en une succession d'événements acoustiques caractérisés par une cible spectrale et un fonction d'interpolation. Ces événements constituent des segments qui se recouvrent partiellement.

Nous avons ensuite cherché à regrouper ces segments en un petit nombre de classes (typiquement 64). Nous avons effectué cette classification de manière non supervisée en utilisant une simple quantification vectorielle.

Le dictionnaire de quantification vectorielle a été déterminé par l'algorithme LBG[24].

Nous n'avons pas utilisé les cibles spectrales comme ensemble d'apprentissage, mais les vecteurs cepstraux réels situés au centre de gravité des fonctions d'interpolation.

La classification de chaque événement spectral (segment $\mathbf{s}$ formé de vecteurs spectraux $\mathbf{c_{k,i}}$ avec $i \in [0, N_s - 1]$) a été faite en comparant les distances du segment avec les différentes classes de quantification vectorielle. On note $\mathbf{y_k}$ le centroïde de la $k^{\text{ème}}$ classe. Nous avons défini la distance du segment $\mathbf{s}$ au vecteur $\mathbf{y_k}$ par :

$$d(\mathbf{s}, \mathbf{y_k}) = \sum_{i=0}^{N_s-1} \|\mathbf{y_k} - \mathbf{s_{k,i}}\|^2.$$

Après cette classification, on dispose d'une première transcription de la base de données. Cette transcription comprend les bornes des segments (obtenus à partir des fonctions d'interpolation de la décomposition temporelle) et pour chaque segment un symbole correspondant au numéro de sa classe de quantification vectorielle.

4.2.1.4 Modélisation HMM, affinement du jeu d'unités ALISP Les $L = 64$ classes obtenues par décomposition temporelle et quantification vectorielle sont modélisées par des modèles de Markov cachés (HMM). Cette modélisation facilite leur utilisation dans un système de reconnaissance (codeur) et est utilisée pour affiner le jeu d'unités acoustiques. L'ensemble des L HMM constitue le jeu d'unités acoustiques qui est utilisé par le codeur. Le nombre d'unités acoustiques élémentaires est déterminé par le nombre de classes de la quantification vectorielle.

On effectue un affinement du jeu d'unités acoustiques, en répétant quelques itérations comprenant un apprentissage des HMM suivi d'une re-segmentation de la base de données avec les nouveaux modèles HMM (voir figure I.22).

On entraîne L modèles HMM , puis on utilise ces modèles pour re-segmenter la base de données, on actualise les L classes sur lesquelles on ré-apprend les paramètres des HMM. Au bout de quelques (typiquement 5) itérations les résultats se stabilisent, aussi bien la vraisemblance des modèles que la cohérence acoustique des classes obtenues.

Architecture des HMM

Le principe de la modélisation HMM est de supposer que les observations (vecteurs spectraux par exemple) sont émises par un modèle de Markov caché, c'est-à-dire que l'on n'observe pas directement. Ce modèle est une machine d'état finie caractérisée par son architecture : nombre d'états et connexions entre états, par les probabilités initiales des états et de transition entre états ($a_{i,j}$ = probabilité de transition de l'état i à l'état j). Les changements d'état dans ce modèle ont lieu au rythme

[24]LBG= Lindo-Buzzo-Gray.

trame. À l'arrivée dans un état i à l'instant t, le modèle émet un vecteur $\mathbf{o}(t)$ qui est observé. La densité de probabilité associé à l'émission de $\mathbf{o}(t)$ dans l'état i est notée $b_i(\mathbf{o}(t))$.

Le jeu de paramètres Λ d'un HMM d'architecture donnée, est donc formé de la distribution initiale de probabilités des états, des probabilités de transition $a_{i,j}$ et des paramètres des densités de probabilités $b_i(\mathbf{o})$.

Nous avons choisi d'utiliser des modèles HMM Gauche-Droite.

Pour une classe ou unité acoustique, obtenue par DT et VQ, nous utilisons un modèle HMM à 3 états émetteurs et 2 états non émetteurs (utiles pour les connexions entre HMM), avec des transitions possibles d'un état vers lui-même ou vers le suivant et pas de saut d'états.

Pour des unités plus longues (voir section 4.2.1.5) formées d'une séquence de i unités élémentaires, le nombre d'états émetteurs du modèle vaut $2i + 1$.

La figure I.24 représente l'architecture des HMM utilisés dans le cas des unités obtenues sans allongement, c'est-à-dire de HMM avec 3 états émetteurs et 2 états non émetteurs.

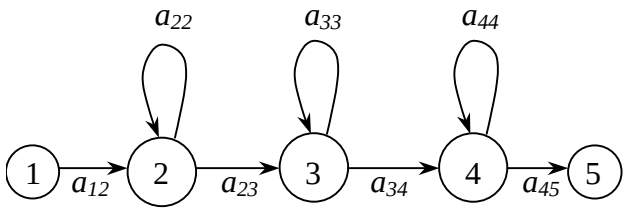


FIG. I.24 – Architecture utilisée pour le HMM modélisant une unité acoustique élémentaire

Dans la plupart de nos expériences, nous avons utilisé un modèle de langage très simple, n'utilisant que des uni-grammes et supposant que les unités acoustiques sont indépendantes et de même probabilité a priori. Nous avons effectué quelques essais avec un modèle plus sophistiqué faisant appel à des multigrammes de longueurs variables (voir section 4.2.1.5).

Par la suite, on note O l'observation sur une durée de T trames. Nous avons décomposé l'observation en 3 flux de paramètres indépendants et de même poids. les LPCC (1^{er} flux), leurs dérivées ΔLPCC ($2^{\text{ème}}$ flux), l'énergie logarithmique $\log(E)$, et sa dérivée $\Delta \log(E)$ ($3^{\text{ème}}$ flux).

À chaque flux de paramètres, est associé une densité de probabilité gaussienne simple (pas de mélange de gaussienne). Ce choix est correct pour des expériences monolocuteur, mais sera à remettre en cause pour les extensions multilocuteur.

La densité de probabilité d'émission de l'observation $o(t)$, à l'instant t dans l'état i, peut donc s'écrire :

$$\begin{aligned} b_i(\mathbf{o}(t)) &= \prod_{s=1}^{3} \mathcal{N}\left(\mathbf{o}_s(t); \mu_{\mathbf{s,j}}, \mathbf{\Sigma_{s,i}}\right). \\ \mathcal{N}\left(\mathbf{o}; \mu, \mathbf{\Sigma}\right) &= \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{o}-\mu)^T \Sigma^{-1} (\mathbf{o}-\mu)}. \\ n &= \text{dimension de } (o). \end{aligned}$$

Dans cette équation s varie de 1 à 3 car il y a 3 flux de paramètres.

Le jeu de paramètres Λ associés à un modèle est constitué des probabilités initiales des états, des probabilités de transition entre états (notées $a_{i,j}$) et des paramètres des gaussiennes (moyennes et matrices de variance) pour les 3 flux de paramètres.

Outils logiciels utilisés

Pour l'apprentissage des modèles HMM ainsi que pour leur reconnaissance, nous avons utilisé le logiciel HTK [134] (*Hidden Markov Model Tool Kit*) de la société Entropics aujourd'hui disparue.

Apprentissage des HMM

L'apprentissage des HMM est fait sur le même corpus que celui qui a été utilisé pour la DT et la VQ. L'initialisation des modèles prend en compte la transcription initiale T^0 fournie par l'association DT+VQ.

Le but de l'apprentissage est de déterminer les paramètres optimaux Λ^0 des modèles de façon à optimiser le critère de vraisemblance conjointe de l'observation et des modèles pour une transcription donnée :

$$\Lambda^0 = \left\{\lambda_i^0\right\} = \arg\max_{\Lambda} \mathcal{L}\left(O, \Lambda | T^0\right).$$

On effectue un 1^{er} apprentissage des modèles sans contexte, c'est à dire modèle par modèle en les traitant de manière indépendante. On utilise pour cela les fonction HInit et HRest de HTK.

Puis en utilisant l'apprentissage précédent comme initialisation, on effectue un $2^{\text{ème}}$ apprentissage en contexte puisque le codeur est analogue à un système de reconnaissance de parole continue. Tous les modèles sont entraînés en parallèle sur la base de données. On suppose que tout HMM peut être connecté à n'importe quelle autre HMM.

Apprentissage hors contexte, initialisation de l'apprentissage en contexte

La fonction HInit de HTK, effectue une initialisation des paramètres HMM hors contexte de la manière suivante : Pour un HMM, les observations $\mathbf{o}(t)$ sont distribuées de manière uniforme aux différents états du modèle et une $1^{\text{ère}}$ estimation de μ et $\mathbf{\Sigma}$ est obtenue pour l'état par :

$$\hat{\mu_{\mathbf{i}}} = \frac{1}{T_i} \sum_{t=t_1}^{t_{T_i}} \mathbf{o}(t). \tag{I.8}$$

$$\hat{\mathbf{\Sigma}_{\mathbf{i}}} = \frac{1}{T_i} \sum_{t=t_1}^{t_{T_i}} (\mathbf{o}(t) - \mu_i)(\mathbf{o}(t) - \mu_i)^T. \tag{I.9}$$

Dans cette équation, T_i représente la durée d'observation associé à l'état j, et t_1 et t_{T_i} sont le 1^{er} et le dernier instant associé à l'état i.

La fonction HInit affine ensuite cette estimation en cherchant, par l'algorithme de Viterbi, la séquence d'états la plus vraisemblable. Puis les observations sont ré-attribuées aux états conformément à cette séquence et les moyennes et matrices de covariance sont ré-estimées avec la formule I.9. Le procédé est itéré jusqu'à ce que les estimations ne varient plus significativement. L'estimation des probabilités de transition entre états se fait par la formule :

$$\hat{a_{i,j}} = \frac{A_{i,j}}{\sum_{k=2}^{N} A_{i,k}}.$$

$A_{i,j}$ représente le nombre de transitions entre les états i et j.

La fonction HRest utilise l'estimation des paramètres effectuée par HInit comme initialisation de l'algorithme de Baum-Welch (ou forward-Backward) qui estime les paramètres au sens du maximum de vraisemblance, par les formules :

$$\hat{\mu_{\mathbf{i}}} = \frac{\sum_{t=1}^{T} L_i(t)\mathbf{o}(t)}{\sum_{t=1}^{T} L_i(t)} \tag{I.10}$$

$$\hat{\mathbf{\Sigma}}_{\mathbf{i}} = \frac{\sum_{t=1}^{T} L_i(t)(\mathbf{o}(t)-\mu_i)(\mathbf{o}(t)-\mu_i)^T}{\sum_{t=1}^{T} L_i(t)} \tag{I.11}$$

Dans cette équation $L_i(t)$ représente la probabilité d'être dans l'état i à l'instant t. Ces probabilités sont calculées de manière efficace par l'algorithme de Baum-Welch.

Les fonctions HInit et HRest travaillent sur un modèle à la fois, en utilisant les données des segments qui ont été classés dans la classe du modèle en question.

Apprentissage en contexte

Le jeu d'unités acoustiques est utilisé dans un codeur qui fonctionne comme un système de reconnaissance de parole continue. Aussi est-il préférable d'effectuer l'apprentissage des HMM en contexte, c'est-à-dire en tenant compte des HMM environnants, d'autant plus que la $1^{ère}$ segmentation-transcription de la base de données n'est pas parfaite.

La fonction HERest de HTK effectue cet apprentissage en contexte en utilisant l'algorithme de Baum-Welch. Tous les HMM sont entraînés en parallèle sur toute la base de données. La fonction utilise les L initialisations des HMM obtenues par HInit et HRest et les fichiers de transcription de la base de données. Dans ces fichiers de transcription, seuls les symboles sont utiles pour HERest, les bornes des segments ne sont pas utilisées. Pour chaque fichier d'apprentissage, HERest construit un grand modèle HMM composite par assemblage des HMM correspondant aux symboles donnés dans la transcription du fichier. Les états non-émetteurs des HMM facilitent ces connexions. Ce grand modèle est ensuite entraîné par l'algorithme de Baum-Welch.

La routine HERest est itérée quelques fois (typiquement 5) jusqu'à stabilisation des estimations.

Re-segmentation de la base de données à partir d'un jeu de HMM et affinement des modèles HMM

Après l'apprentissage d'un jeu de HMM, on utilise ces HMM pour segmenter et transcrire la base de données.

On utilise pour cela la fonction HVite de HTK. cette fonction effectue une reconnaissance de la suite des HMM présents dans un signal test. Comme son nom l'indique, cette fonction utilise l'algorithme de Viterbi. La meilleure transcription T (ou suite de modèles reconnus) est celle qui maximise le critère de vraisemblance $\mathcal{L}$:

$$T = \arg\max_{M_1^N} \mathcal{L}\left(O|M_1^N\right)\mathcal{L}\left(M_1^N\right).$$

Où M_1^N représente une suite de modèles reconnus.

Et comme on utilise un modèle de langage simple, n'utilisant que des uni-grammes, avec un facteur de langage γ, on peut écrire le critère sous la forme :

$$\max_{M_1^N} \prod_i p(M_i)^\gamma p\left(O|M_i\right).$$

La fonction HVite a besoin d'un fichier décrivant le réseau de connexions possibles entre HMM. Dans notre cas, nous avons supposé que toutes les connexions étaient possibles.

on utilise les résultats obtenus sur la base de données pour re-calculer un modèle de langage dans lequel on supprime les HMM qui sont apparus trop peu de fois dans la reconnaissance (seuil fixé ici à 20).

Après la phase de reconnaissance, on dispose d'une nouvelle transcription de la base de données.

On recommence ensuite la même procédure : apprentissage-segmentation-transcription un petit nombre de fois. Dans nos expériences nous avons itéré 5 fois les procédures.
En résumé les étapes de l'itération m peuvent se formuler sous la forme :

- Segmentation et transcription de la base de données avec les modèles Λ^{m-1}, la transcription T^{m-1} et le modèle de langage LM^{m-1} de l'étape $m-1$. La nouvelle transcription T^m s'obtient par :
$$T^m = \arg\max_{M_1^N} \mathcal{L}\left(O, M_1^N | \Lambda^{m-1}, LM^{m-1}\right).$$
- Ré-estimation des paramètres des modèles HMM :
$$\Lambda^m = \arg\max_{\Lambda} \mathcal{L}\left(O, \Lambda | T^{m-1}\right).$$

À la fin de la phase d'affinement des modèles, on dispose des éléments suivants :

- Un ensemble de L unités acoustiques élémentaires, modélisées par L modèles HMM.
- Pour chaque unité acoustique, un ensemble de segments de parole de la base de données associés à cette unité par la phase de reconnaissance.
- Une transcription de la base de données avec ces unités acoustiques.

4.2.1.5 Allongement des unités acoustiques, multigrammes Les $L = 64$ unités acoustiques élémentaires obtenues par la technique précédente (DT+VQ+HMM itéré) ont une durée moyenne de l'ordre de 1/15s, c'est-à-dire une durée moyenne comparable à celle d'un phone. Cela provient du choix des paramètres de la DT.

On peut allonger la longueur des unités acoustiques en déterminant des séquences caractéristiques d'unités élémentaires de longueur variable comprise entre 1 et n. On appelle multigramme, ces séquences de longueur variable.

On fixe a priori la longueur maximale n.

La détermination de ces unités acoustiques caractéristiques longues, devrait permettre de réduire le débit de codage ou d'améliorer la synthèse en diminuant le nombre de transitions. Mais cette approche augmente le retard introduit par le traitement. D'autre part cet allongement des unités acoustiques va de pair avec l'augmentation de leur nombre, on note Z le nombre d'unités acoustiques longues.

La détermination des unités longues de longueur variable ou multigrammes peut se faire après la phase de transcription de la base de données par Décomposition temporelle et quantification vectorielle ou après la phase de transcription par les HMM. Dans les 2 cas la démarche est la même.

On part d'une suite de symboles $d(k)$ correspondant à la transcription de la base de données par les unités acoustiques élémentaires. L'observation O est constituée de cette chaîne de symboles. On cherche à déterminer un ensemble X de Z multigrammes $\{x_i\}$ formé chacun de 1 à n symboles et

de probabilité $p(x_i)$ qui maximise la vraisemblance de la segmentation optimale S_{opt} de l'observation par ces multigrammes :

$$X_{opt} = \arg\max_{X} \mathcal{L}\left(S_{opt}|X\right).$$

On suppose les multigrammes indépendants.

Pour une segmentation donnée S formée d'une séquence de multigrammes s_k et un dictionnaire X de multigrammes :

$$\mathcal{L}(S|X) \quad = \quad \prod_{x_k \in S} p(x_k).$$

Pour un dictionnaire X donné, la recherche de la meilleure segmentation se fait par l'algorithme de Viterbi.

La recherche du dictionnaire de multigrammes, se fait de manière itérative en utilisant l'algorithme EM [56]. À l'itération m, le dictionnaire de multigrammes est modifié en mettant à jour les probabilités $p(x_i)$ par :

$$p(x_i)^{(m)} = \frac{N_i}{N_{S_{opt}}}.$$

Où $N_{S_{opt}}$ est le nombre total de segments multigrammes dans la segmentation optimale et N_i est le nombre de multigrammes égaux à x_i. D'autre part les multigrammes trop rares sont supprimés du dictionnaire.

L'initialisation du dictionnaire se fait avec toutes les séquences possibles de longueur 1 à n et leur probabilités d'occurrence dans la base de données d'apprentissage.

Si l'étape d'allongement des unités est appliquée avant la modélisation HMM, le nombre de HMM à entraîner sera Z au lieu de L, avec $Z \gg L$. Il faudra donc utiliser une base de données d'apprentissage plus grande, ou bien lier certains des paramètres des gaussiennes. On peut par exemple lier les distributions de probabilités des états représentés par la même classe de QV au départ.

Si l'étape de détermination des multigrammes est appliquée après le calcul des HMM, on peut modifier le modèle de langage. Nous ne l'avons pas fait jusqu'à maintenant.

Nous avons utilisé une longueur maximale $n = 5$ dans nos expériences.

4.2.2 Codeur, étape de reconnaissance

Le codeur de parole travaille comme un système de reconnaissance de parole continue et utilise les mêmes techniques pour reconnaître une suite de segments acoustiques caractéristiques dans le signal à coder.

Les segments caractéristiques sont définis par des modèles HMM et la reconnaissance s'effectue sur un HMM composite formé de modèles HMM élémentaires connectés en séquence. Dans nos expériences nous avons utilisé la fonction HVite de HTK pour cette reconnaissance. Cette fonction utilise une alternative à l'algorithme de Viterbi appelée méthode de passage du jeton. [134].

Le codeur transmet au décodeur les indices des unités acoustiques reconnues ainsi que des informations sur la prosodie.

Le synthétiseur devra, à partir de ces données, générer un segment de parole pour chaque unité acoustique reconnue. Mais les unités acoustiques étant décrites par des HMM, il n'est facile de les utiliser directement pour la synthèse.

Aussi avons nous associé à chaque classe acoustique, un ensemble de 8 représentants (signaux de parole) qui sont les unités qui seront utilisées en synthèse.

Ces représentants sont les 8 segments les plus longs de l'ensemble des segments de la base de données d'apprentissage appartenant à la classe de l'unité acoustique en question. Ce choix est arbitraire, et j'étudie actuellement comment choisir ces représentants.

Les unités acoustiques de synthèse sont donc, dans cette $1^{ère}$ version du codeur, les 8 segments de parole les plus longs de la classe associée à une unité acoustique dans la base d'apprentissage.

le codeur, après avoir reconnu la suite des unités de codage, détermine pour chacune d'elles l'unité de synthèse parmi les 8 possibles qui sera utilisée par le décodeur pour re-synthétiser le signal.

Le choix de l'unité de synthèse se fait en comparant par DTW[25] le segment réel avec les 8 représentants de l'unité acoustique reconnue.

Le codeur transmet finalement au décodeur les indices des séquences reconnues, le numéro de l'unité de synthèse ainsi qu'une information supplémentaire sur la prosodie dans le segment. Pour le moment nous n'avons pas travaillé sur le codage segmental des paramètres de prosodie.

4.2.3 Décodeur, étape de synthèse

Le décodeur effectue la synthèse à partir de ces informations en utilisant une technique de concaténation de segments et un modèle LPC. Nous n'avons pas encore beaucoup travaillé sur la synthèse et nous envisageons de l'améliorer en appliquant des méthodes similaires à celles des systèmes de synthèse de parole à partir du texte, telles que les synthèse PSOLA[26] et HNM[27].

J'encadre actuellement un étudiant de dernière année de l'ESIEE, en stage chez Thomson-CSF partenaire du projet RNRT. Cet étudiant a remplacé la synthèse LPC par une synthèse la HSX de Thomson et a ainsi amélioré la qualité du signal synthétique.

Les segments à concaténer (unités de synthèse) doivent être modifiés pour leur appliquer les caractéristiques de prosodie transmises par le codeur. Pour le moment nous avons utilisé une synthèse LPC très rudimentaire. Mais une modélisation HNM des unités de synthèse faciliterait ce travail.

Le modèle HNM modélise une trame de parole, comme la somme d'un signal harmonique jusqu'à une fréquence maximum f_{max} et d'un bruit au-delà de f_{max}. Les modifications de durée et de période sont donc très simples à réaliser [122, 21, 123].

Pour le moment, nous avons peu travaillé sur le codage de la prosodie et la synthèse est effectuée à partir des paramètres de prosodie non codés. Ces paramètres sont :

– La période fondamentale, L'énergie, Le chemin d'alignement temporel (DTW) entre le segment à coder et le représentant de synthèse choisi.

La figure I.25 résume les opérations effectuées par le codeur et le décodeur.

4.2.4 Expériences réalisées et résultats

4.2.4.1 Bases de données Nous avons testé ce codeur en mono-locuteur sur 3 bases de données : en français suisse (base de données téléphonique Polyvar), en anglais (base de données Boston Uni-

[25] DTW = Dynamic Time Warping.
[26] PSOLA =Pitch Synchronous Overlap and Add.
[27] HNM = Harmonic plus Noise Model.

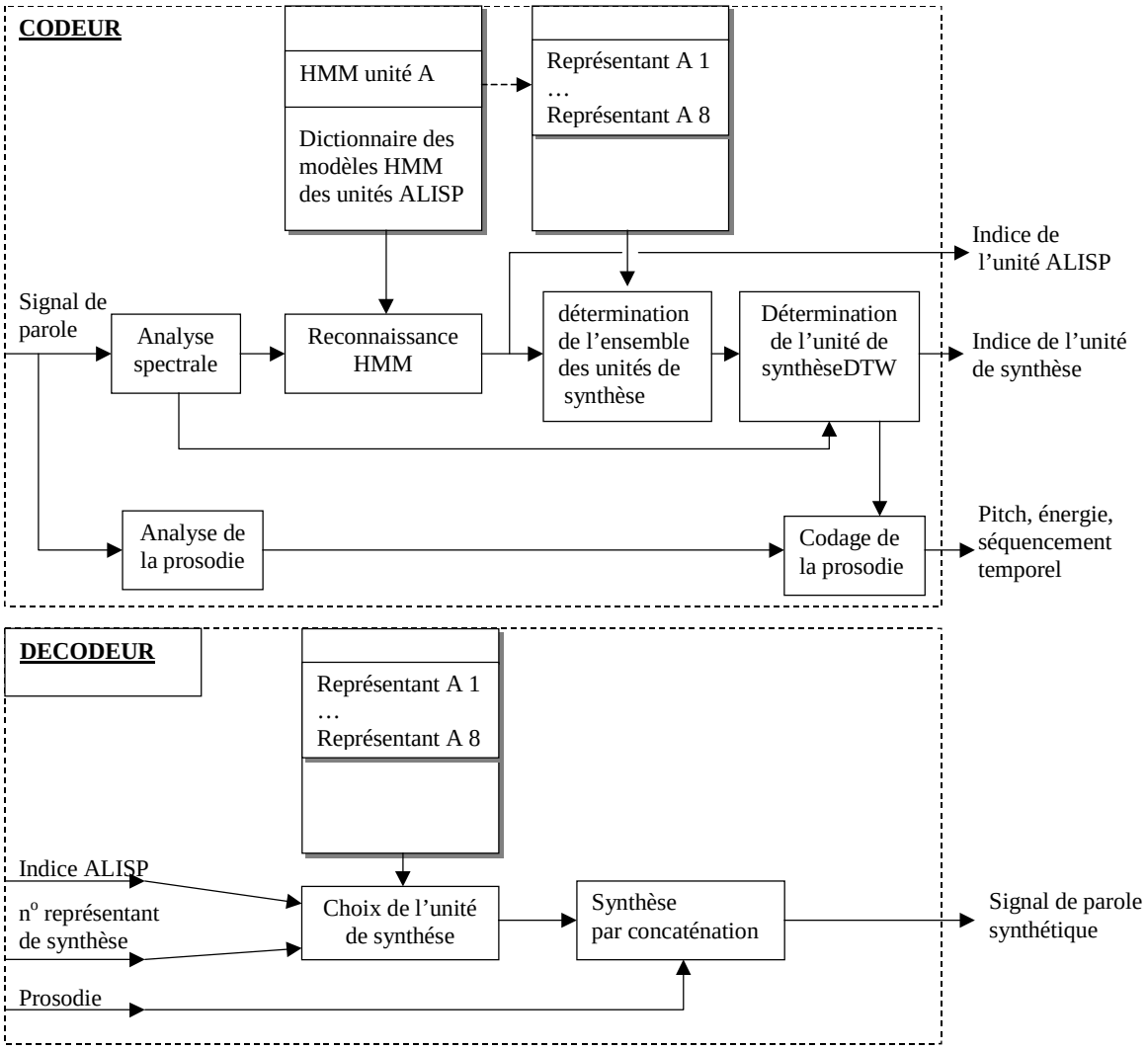


FIG. I.25 – Codeur-Décodeur de parole utilisant les unités acoustiques obtenues de manière non supervisée

versity Radio Speech Corpus distribuée par LDC[28], notée BU), et en tchèque (locuteur professionnel Martin Ruzěk, notée MR).

La $1^{\text{ère}}$ base de données n'était pas de qualité suffisante. La durée de signal par locuteur était trop courte et certains mots clés étaient répétés très souvent.

La base BU est échantillonnée à 16 KHz. Elle contient des enregistrements de 7 présentateurs radio professionnels de la station WBUR. Nous avons travaillé avec 1 locuteurs masculin (M2B, 78 min) et un locuteur féminin (F2B, 83 min).

La base de données contient 2 types d'enregistrement : des enregistrements faits à la radio et des enregistrements effectués par les mêmes locuteurs en studio. Nous avons utilisé les enregistrements radio pour le test et les enregistrements studio pour le test.

La base de données tchèque a été enregistrée en collaboration entre l'université technique de Brno, l'université Masaryk de Brno et la station de Brno de la radio tchèque. Cette base est échantillonnée à 11,025 KHz. Elle contient des enregistrements de plus d'une heure de textes lus par l'acteur Martin Ruzek. Les signaux ont été segmentés en segments de 6 à 18 s sur les minima d'énergie. Une partie du corpus ($7/8^{\text{ème}}$) a servi à l'apprentissage et le reste au test.

4.2.4.2 Paramétrisation Pour la base de données BU à 16 KHz d'échantillonnage, nous avons utilisé 16 coefficients LPCC, calculés sur des trames de 20 ms se recouvrant de 10 ms. Nous avons soustrait la moyenne cepstrale fichier par fichier d'enregistrement.

Pour la base polyphone et la base MR, nous n'avons utilisé respectivement que 10 et 12 coefficients LPCC.

La décomposition temporelle a été ajustée pour obtenir un débit moyen d'une quinzaine d'événements par seconde.

Nous avons utilisé $L = 64$ classes pour la quantification vectorielle qui suit la décomposition Temporelle.

4.2.4.3 Résultats Les résultats obtenus pour le codage des unités acoustiques sont comparables pour les 3 base de données.

Le codage des indices des unités de synthèse est fait simplement. Pour L unités, on utilise $\log_2(L)$ bits pour coder le numéro de l'unité. On pourra remplacer cette approche par un codage entropique prenant en compte les probabilités des unités.

Le débit binaire moyen R est obtenu par la formule :

$$R = \frac{\log_2(L) \sum_{i=1}^{L} c(M_i)}{T_f \sum_{i=1}^{L} c(M_i) l(M_i)}. \tag{I.12}$$

Dans cette formule, $c(M_i)$ est le nombre d'occurrences de l'unité M_i sur la base de donnée transcrite, $l(M_i)$ est la longueur de M_i en nombre de trames, et T_f est le décalage entre trames acoustiques en secondes.

[28]Linguistic Data Consortium - University of Pennsylvania, http ://www.ldc.upenn.edu/

Pour un débit moyen de 120 bps pour ces unités, la parole synthétique est intelligible (sauf pour la base polyphone de qualité insuffisante).

À ce débit, il est nécessaire d'ajouter le débit correspondant au codage des unités de synthèse (en moyenne 60 bps, 3 bits par unité) et au codage des informations de prosodie.

Le tableau I.5 résume les résultats obtenus pour la base BU.

Sur cette base, nous avons testé l'allongement des unités en segmentant la séquence d'indices de HMM en multigrammes de longueur maximale égale à 6. Cette étape a diminué le nombre de transitions en synthèse et amélioré légèrement la qualité du signal synthétique. Le nombre de multigrammes obtenus est de 722 pour le locuteur féminin et de 972 pour le locuteur masculin.

Locuteur	F2B		M2B	
Débit binaire sur les données en :	Apprentissage	Test	Apprentissage	Test
HMM 6$^{\text{ème}}$ génération	189,27	190,28	189,75	195,51
HMM 6$^{\text{ème}}$ génération+MG	135,91	145,09	141,86	156,02

TAB. I.5 – Débits binaires obtenus sur la base "BU radio speech corpus". (Ce débit ne prend en compte que le codage des unités acoustiques de codage et de synthèse).

Pour la base de données tchèque, le débit moyen obtenu est d'environ 175 bps.

L'intelligibilité de la parole synthétique démontre que la représentation symbolique obtenue représente correctement le contenu acoustico-phonétique du message. Le débit binaire en bps et la taille du dictionnaire représentent l'efficacité de la description, la qualité de la parole synthétique étant liée à la précision de cette représentation.

4.2.5 Correspondances des unités ALISP avec les phonèmes

Une partie de la base de données « Boston University Radio Corpus » étant étiquetée phonétiquement, nous avons essayé d'analyser le lien entre les unités de codage ALISP et les phonèmes. L'étiquetage phonétique de la base BU a été réalisé par un système de reconnaissance de phonèmes et d'unités sub-phonétiques. On appelle par la suite ce jeu d'unités UP pour le distinguer du jeu d'unités ALISP.

Nous disposons donc d'une double segmentation-transcription de certains fichiers, par les unités ALISP et par les unités UP.

On note n_a et n_p le nombre d'unités ALISP et UP.

Pour le locuteur féminin F2B, nous avons calculé les recouvrements entre les 2 types d'unités.

Pour chaque occurrence $p_{i,k}$ de l'unité UP p_i, nous avons observé avec quelles unités ALISP elle coïncidait, et mesuré son recouvrement relatif avec chacune de ces unités ALISP. La grandeur $r(p_{i,k}, a_j)$ représente le nombre de trames de recouvrement relatif entre l'unité UP et l'unité ALISP numéro j a_j et la $k^{\text{ème}}$ occurrence de p_i.

Le recouvrement relatif est le rapport entre le recouvrement absolu $R(p_{i,k}, a_j)$ et la longueur $l(p_i)$ de l'unité x_i.

$$r(p_{i,k}, a_j) = \frac{R(p_{i,k}, a_j)}{l(p_i)}.$$

Pour établir la correspondance entre p_i et a_j nous avons ensuite moyenné ces valeurs de recouvrement sur les $c(p_i)$ occurrences de p_i. Nous obtenons ainsi une matrice de confusion $\mathbf{X} = \{x_{i,j}\}$ avec :

$$x_{i,j} = \frac{\sum_{k=1}^{c(p(i))} r(p_{i,k}, a_j)}{c(p(i))}$$

Les unités ALISP utilisées sont celles obtenues sans allongement. Il y a donc $L = 64$ unités ALISP.

Pour les unités UP, nous n'avons pas distingué les voyelles accentuées ou non accentuées. Il y a, de ce fait, 57 unités UP. Elles sont listées dans le tableau I.6.

classe phonétique	abréviations.	phonèmes
clôtures	CLO	BCL, DCL, GCL, PCL, KCL, TCL
occlusives	OCC	B, D, G, P, T, K, DX
affricatives	AFF	JH, CH
fricatives	FRI	S, SH, Z, ZH, F, TH, V, DH
nasales	NAS	M, N, NG, EM, EN, NX
demi-voyelles et glides	DVG	L, R, W, Y, HH, HV, EL
voyelles	VOY	IY, IH, EH, EY, AE, AA, AW, AY, AH AO, OY, OW, UH,UW,ER,AX,AXR
autres	AUT	PAU, H#, H, brth

TAB. I.6 – Jeu d'unités UP, phonèmes utilisés dans la comparaison unités ALISP — unités phonétiques.

L'alphabet utilisé pour les étiquettes phonétiques est ARPABET défini dans la base de données TIMIT.

Pour les noms des 64 unités ALISP, nous avons utilisé les 26 lettres de l'alphabet en majuscule, plus les 26 lettres minuscules, les chiffres de 0 à 9, et les 2 symboles $ et @. La figure[29] I.26 représente les résultats obtenus, après arrangement pour obtenir une forme quasi-diagonale.

La matrice de confusion n'est pas diagonale, mais on peut observer des correspondances fortes entre certaines unités ALISP et UP. Par exemple, l'unité ALISP $ correspond essentiellement à l'unité UP SH, mais elle est aussi liée à son équivalent voisé ZH et aux unités affriquées JH et CH.

Ce type de correspondance pourrait être utilisé dans un système de reconnaissance de la parole utilisant les unités ALISP.

4.2.6 Perspectives, projet RNRT SYMPATEX

Nous n'avons pas encore travaillé sur les questions d'adaptation au locuteur, au canal et à la langue. Elles seront étudiées, de même que le codage segmental des paramètres prosodiques et l'amélioration de la synthèse (voir la section « Conclusions et perspectives »), pendant le projet RNRT SYMPATEX[30]. Ce projet exploratoire de 3 ans qui a été labellisé en 99 regroupe les sociétés Thomson CSF, ELAN, Info Télécom, l'ENST et l'ESIEE. La technique de codage proposée sera intégrée dans un démonstrateur de messagerie unifiée intégrant les messages vocaux et textuels.

[29] J'ai emprunté cette figure à J. Černocký que je remercie.

[30] SYstème de Messagerie Unifiée avec présentation vocale des messages (PArole et TEXte).

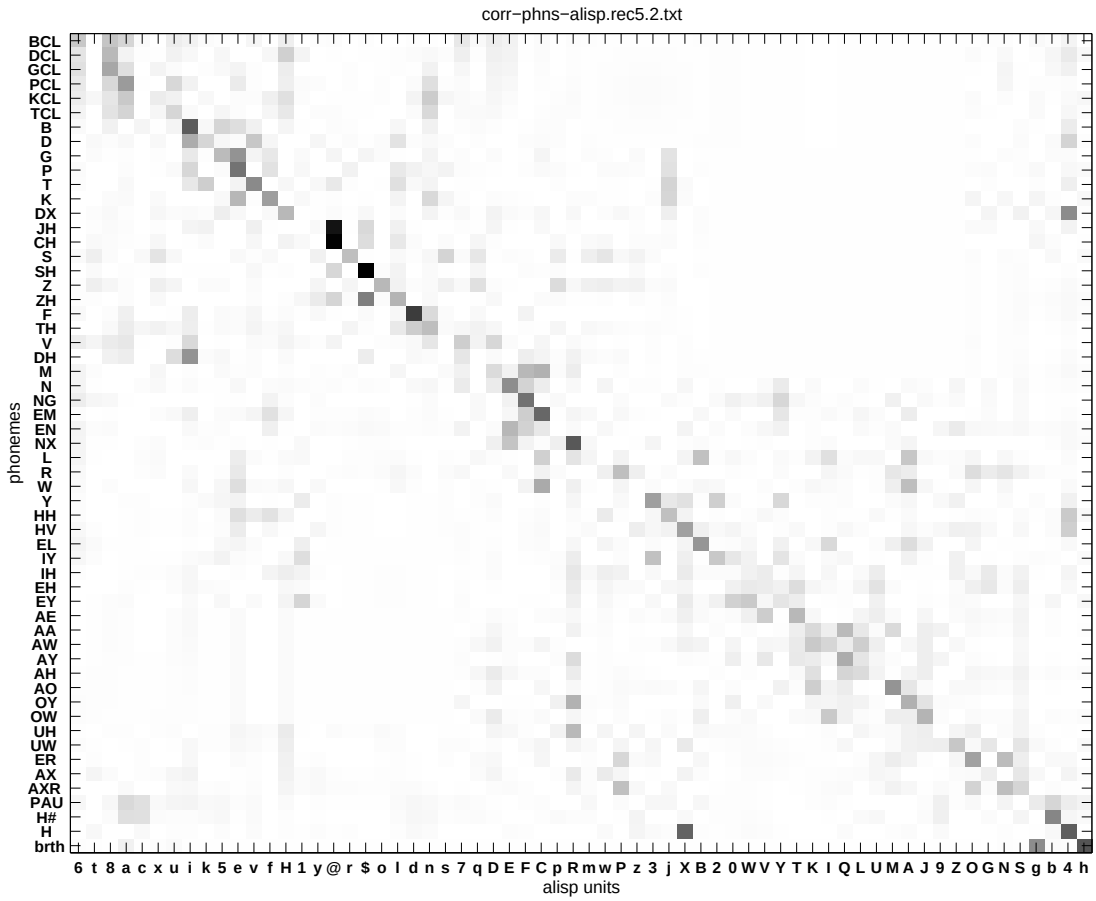


FIG. I.26 – Correspondance de la segmentation ALISP avec un alignement phonétique pour le locuteur F2B de BU corpus. Le gris-clair correspond à une corrélation nulle, le noir à la valeur maximale de $x_{i,j}$=0.806

5 Implantation de codeurs de parole sur DSP

J'ai implanté, en collaboration avec des étudiants, plusieurs codeurs de parole sur processeurs de traitement de signal : un codeur LPC10 à 2400 bps sur TMS320C25 (pour la société SECMAT), un codeur en sous-bandes sur TMS320C50 (pour la société SECMAT), un codeur CELP à 4800 bps et un codeur demi débit GSM sur TMS320C30 (en liaison avec le projet Elite de Texas Instruments).

Texas-Instruments a retenu l'ESIEE pour un de ses projets appelés Elite. Dans ce cadre, nous avons bénéficié d'une aide matérielle (don de cartes et de logiciels) et financière pour des participations à diverses conférences.

Le thème que nous avions proposé à Texas était le développement d'un démonstrateur de répondeur vocal statique sur TMS320C30 (processeur format flottant). Comme base des algorithmes de codage, nous avons retenu la norme demi-débit GSM à 5600 bps. Mais pour une application de stockage, nous pouvions nous affranchir des contraintes de délai et augmenter ainsi les performances de codage.

Plusieurs étudiants ont participé à ce projet qui a été présenté lors de la conférence Texas « First European conference on DSP research and education » qui s'est déroulée à l'ESIEE en 96 [11].

CHAPITRE II

TRANSFORMATION DE VOIX

J'ai étudié la transformation de la voix (timbre et prosodie) pour des applications à la synthèse de parole à partir du texte. Mais ces techniques peuvent aussi être utiles pour des applications de codage à très bas débit, dans lesquelles, il peut être intéressant de transmettre au décodeur quelques informations concernant le locuteur et d'effectuer une personnalisation de voix décodée.

Sur ce thème général, je me suis intéressée à la décomposition source-filtre du signal vocal (thèse de Jiangping Liu [85]) et à la suite de ce travail j'ai obtenu une convention CNET sur la transformation du timbre de la voix.

Ce chapitre comprend 3 sections. La section 1 définit la transformation de voix et précise ses applications. Les sections 2 et 3 exposent mes travaux dans le domaine. La section 2 présente, de manière très succincte, l'étude sur la décomposition source-filtre avec modélisation de l'onde glottique. La section 3 commence par un état de l'art puis présente mon travail sur la conversion de timbre de la voix.

1 Définition et intérêt de la transformation de voix

Après avoir porté principalement sur la reconnaissance du locuteur et l'adaptation au locuteur pour les systèmes de reconnaissance , les études sur la personnalisation de la voix sont appliquées depuis quelques années à des systèmes de synthèse et de conversion de voix. On peut citer les travaux de l'ENST et du CNET (H. Valbret, E. Moulines, F. Charpentier, ...), les travaux menés au Japon par ATR, TOYOCOM, KDD Labs (Shikano, Abe, Kuwabara, Sagasika...), par STL (Wakita, Matsumoto), ainsi que par KTH (Grandström, Carlson), par Childers, et par Savic et Nam.

Modéliser les caractéristiques d'un locuteur par quelques paramètres est un objectif autant pour les systèmes de reconnaissance et de vérification d'identité que pour les systèmes de synthèse vocale. Des applications potentielles sont la personnalisation des systèmes de synthèse s'appuyant sur la concaténation d'unités acoustiques, de même que les dispositifs de traduction automatique.

La personnalisation de la voix des synthétiseurs peut se faire en utilisant des dictionnaires d'unités acoustiques différents pour chaque voix. Pour chaque nouvelle voix, il faut alors enregistrer et segmenter un corpus spécifique pour créer le dictionnaire correspondant. Une autre approche consiste à générer des voix nouvelles à consonance naturelle à partir de voix existantes, on parle alors de conversion de voix. Dans ce chapitre nous ne considérons que ce dernier schéma.

Les systèmes de conversion de voix proposés aujourd'hui tentent d'imiter une voix cible A à partir d'une voix source B. Ces systèmes utilisent généralement une décomposition source filtre du signal vocal. Le filtre de synthèse représentant l'enveloppe spectrale est souvent calculé par prédiction linéaire. Les approches diffèrent par le type de signal d'excitation (source) utilisé : modèles d'onde

glottique (Childers, Bimbot, Hedelin), bruit et impulsions unitaires pseudo-périodiques (premiers travaux d'Abe, Savic et Nam), résiduel de prédiction linéaire codé ou non (Moulines, Valbret). Plus récemment une modélisation de type HNM[1] a été utilisée [123].

La transformation de voix se fait trame par trame ou segment par segment, en appliquant aux paramètres spectraux et prosodiques des transformations qui ont été calculées (apprises) au préalable.

Les modifications de l'excitation (en particulier la fréquence fondamentale) sont réalisées en modifiant l'espacement des impulsions élémentaires dans le cas où ce type de modèle est utilisé, ou en utilisant des techniques PSOLA, ou en modifiant les paramètres d'un modèle HNM.

Les transformations spectrales sont généralement différentes pour chaque classe de sons, ces classes pouvant être obtenues par quantification vectorielle. Les transformations spectrales sont calculées par différentes techniques : régression linéaire multiple, alignement fréquentiel dynamique (DFW), modèle connexionniste...

Les résultats déjà obtenus sont prometteurs, mais ces techniques restent à améliorer et les évaluations perceptives sont encore insuffisantes.

2 Amélioration de la décomposition source-filtre du signal de parole, extraction et modification de l'excitation glottale

J'ai encadré sur ce thème, la thèse de J. Liu [84], thèse qui a consisté à développer des méthodes permettant d'améliorer la décomposition source-filtre du signal vocal.

L'excitation glottique et le conduit vocal sont représentés par des modèles paramétriques dont on identifie les paramètres à partir de signaux de parole uniquement. Ces signaux doivent être enregistrés à l'aide de microphones aux caractéristiques de phase bien linéaires pour que les modèles utilisés soient appropriés.

Pour réaliser l'étude, nous avons enregistré une base de données incluant des enregistrement d'une dizaine de personnes, dans une salle anéchoïque avec un microphone à phase linéaire. Nous avons demandé aux locuteurs de répéter les phrases en parlant à différentes vitesses, et avec une voix normale, aiguë ou grave

Nous avons étudié et comparé différentes façons d'extraire les paramètres des modèles : le filtrage inverse ainsi que les techniques ARX et ARMAX [85]. La dernière approche permettant d'améliorer les performances pour les sons nasaux.

Nous avons appliqué les techniques développées à l'estimation des formants et de leur largeur de bande. Nous avons effectué une analyse statistique des paramètres de l'onde glottique obtenus sur notre base de données. Puis nous avons réalisé quelques expériences de transformation de voix, cette approche permettant de modifier séparément la prosodie en jouant sur les paramètres glottiques et le timbre en jouant sur les paramètres du conduit vocal.

Les principaux défauts de cette approche sont la nécessité d'utiliser un microphone à caractéristiques de phase linéaires et la non prise en compte dans le modèle des couplages entre la glotte et le

[1] HNM= *Harmonic plus Noise Model.*

conduit vocal.

3 Transformation du timbre de la voix, génération de voix nouvelles, application à la synthèse

La conversion de voix consiste à modifier la voix d'un locuteur de référence A pour qu'elle ressemble à celle d'un locuteur cible B, tout en conservant un aspect naturel à la voix transformée.

La conversion de la voix porte sur la prosodie et le timbre.

Les techniques de modification de la prosodie sont aujourd'hui assez performantes, au moins pour des rapports de fréquence fondamentale et de vitesse d'élocution raisonnables. Le problème, pour la prosodie, n'est pas la conversion mais plutôt la génération de prosodies naturelles.

La conversion du timbre de la voix consiste à modifier l'enveloppe spectrale du signal pour changer sa structure formantique. Les techniques de modification de timbre sont moins au point que les techniques de modification de prosodie, en particulier en ce qui concerne la qualité de la voix transformée.

L'objectif général est la création de voix situées entre deux locuteurs : on ne cherche pas seulement à imiter un locuteur, mais aussi à créer des voix intermédiaires qui devront être perçues comme celles de locuteurs différents des locuteurs de départ

Cette section s'intéresse à la conversion du timbre de la voix. Elle dresse un état de l'art du domaine et présente mes travaux sur le sujet effectués dans le cadre d'une convention CNET.

3.1 état de l'art sur la conversion du timbre de la voix

Les différents travaux procèdent globalement de façon assez proche. On distingue la phase d'apprentissage et la phase de transformation.

Pendant la phase d'apprentissage deux corpus lexicalement identiques prononcés par les deux locuteurs sont analysés pour calculer les vecteurs spectraux des trames ou segments successifs. Puis ces corpus sont alignés en temps par une technique DTW, afin de mettre en correspondance les vecteurs spectraux de sons similaires. Enfin les transformations spectrales sont apprises de façon à minimiser un certain critère sur le corpus d'apprentissage qui est en général constitué de mots isolés ou de phrases courtes.

Pendant la phase de transformation, une analyse spectrale est effectuée sur le signal source. Les paramètres spectraux sont modifiés à l'aide de la transformation apprise dans la phase d'apprentissage. Le signal d'excitation du filtre de synthèse ainsi obtenu est modifié pour adapter la prosodie de la source à celle de la cible.

3.1.1 Travaux d'Hélène Valbret [129] LMR, DFW "locales"

H. Valbret a comparé deux approches pour la transformation spectrale :

– la Régression Linéaire Multidimensionnelle,

– l'Alignement Dynamique en Fréquence (DFW : Dynamic Frequency Warping).

Ces transformations sont appliquées sur les coefficients du filtre de synthèse qui est excité par le résiduel de prédiction. Les modifications prosodiques sont appliquées sur le signal d'excitation en utilisant l'algorithme TD-PSOLA (Time Domain Pitch-Synchronous Overlap and Add) [130]. Les paramètres du filtre sont obtenus en utilisant une technique cepstrale discrète qui fournit une meilleure

représentation de l'enveloppe spectrale que la prédiction linéaire classique en particulier pour les sons à fréquence fondamentale élevée.

Les transformations spectrales dépendent des classes phonétiques des sons. Le classement est effectué lors d'une étape de pré-traitement par quantification vectorielle qui partage l'espace acoustique en quelques sous-espaces (typiquement 4). Une transformation est déterminée pour chaque sous-espace lors de l'apprentissage, d'où le terme "local" utilisé pour ces transformations.

3.1.1.1 L'Alignement fréquentiel non-linéaire (DFW) Le principe de la DFW a été proposé par Matsumoto et Wakita [89]. Il s'agit d'un alignement spectral par distorsion fréquentielle. L'ajustement fréquentiel est non-linéaire et est réalisé par comparaison dynamique. Les spectres logarithmiques de deux locuteurs sont échantillonnés et des distances $d^{c,s}$ locales sont calculées :

$$d^{c,s}(i,j) = |S_i^c - S_j^s|^2.$$

S_i^c représente l'échantillon i du locuteur cible et S_j^s est l'échantillon j du locuteur de source. Un chemin optimum à travers la matrice $d^{c,s}$ est calculé à l'aide de l'algorithme de Viterbi. Ce chemin représente l'alignement fréquentiel des spectres du locuteur de référence et du locuteur cible.

H.Valbret a proposé deux méthodes d'estimation de la transformation :

- Pour chaque sous-espace, un chemin médian (la moyenne de tous les chemins) est déduit de tous les chemins obtenus lors de l'alignement fréquentiel du corpus d'apprentissage. ce chemin médian peut être utilisé pour transformer un spectre en un autre lors de la conversion de la voix.
- Le calcul d'un chemin optimal qui minimise une distance sur l'ensemble des couples de vecteurs acoustiques (référence et cible) d'un sous-espace.

3.1.1.2 La Régression Linéaire Multidimensionnelle (LMR) La LMR cherche la fonction linéaire g qui transforme les vecteurs spectraux issus de l'analyse du corpus source pour qu'ils deviennent le plus proches possible des vecteurs spectraux du corpus cible, au sens d'un critère quadratique .

$$\mathbf{X}_j^c = g(\mathbf{X}_j^s) + \mathbf{e}_j \quad j \in [0, M-1].$$
$$\min \sum_j \|e_j\|^2$$

avec $\mathbf{X}_j^s$ vecteur du locuteur de référence, et $\mathbf{X}_j^c$ vecteur du locuteur cible (test), ces vecteurs ayant été mis en correspondance par la DTW.

Si g est linéaire, nous obtenons :

$$\begin{aligned} g(\mathbf{X}_j^s) &= \mathbf{\Gamma}\mathbf{X}_j^s \\ \mathbf{X}^c &= \mathbf{\Gamma}\mathbf{X}^s + \mathbf{E} \\ \min \mathbf{E}^T\mathbf{E} & \end{aligned}$$

Cette équation peut être résolue en utilisant les techniques statistiques de régression linéaire.

Une comparaison des deux techniques de modification du spectre par des tests d'écoute formelle a montré que les auditeurs préfèrent les résultats obtenus par la LMR à ceux de la DFW. Mais ces tests ont été faits sur des bases de données de taille assez limitée.

3.1.2 Travaux de Childers et al [53] : homothétie de l'axe fréquentiel et excitation glottique

Pendant l'analyse, les segments de signal sont regroupés en 5 classes (silence, non-voisé, voisé et deux segments de transition). Une analyse LPC est ensuite effectuée sur ces segments avec des paramètres (ordre, longueur de la fenêtre d'analyse, etc.) qui dépendent de la classe du segment.

L'excitation des filtres LPC utilise un modèle dépendant de la nature du segment (5 signaux d'excitation différents).

La conversion de la voix utilise un facteur de modification du pitch constant fonction des fréquences fondamentales des deux locuteurs.

Une compression/expansion spectrale moyenne compense la différence entre les longueurs des conduits vocaux des deux locuteurs. Il est donc supposé qu'une simple homothétie de l'axe des fréquences du spectre du locuteur source suffit pour imiter le spectre du locuteur cible.

Les facteurs de qualité de la conversion de la voix, étudiés par l'équipe de Childers, donnent une bonne idée des paramètres du signal de la parole qui déterminent l'individualité d'un locuteur. Pour obtenir une bonne qualité du signal converti il est essentiel d'avoir :

1. une bonne mesure du spectre (formants, largeur de bande des formants, énergie),
2. une détermination précise du pitch pour les zones voisées
3. une bonne modélisation du signal d'excitation.

Le problème principal de ce système reste l'expansion/compression du spectre par un facteur constant. Une transformation linéaire sur tous les phonèmes n'est pas suffisante pour la transformation des spectres. Le décalage des formants pour deux locuteurs différents dépend de plusieurs facteurs : du numéro du formant, du pitch, ...

3.1.3 Travaux de Iwahashi et al [70] conversion de voix par interpolation des voix de plusieurs locuteurs

Il s'agit d'une méthode de transformation spectrale qui crée un spectre interpolé à partir d'un certain nombre de locuteurs de référence. Les spectres à court-terme des corpus d'apprentissage de plusieurs locuteurs mémorisés lors de l'étape d'apprentissage permettent la création d'une nouvelle série de vecteurs par interpolation la plus proche possible du locuteur cible. Tous les enregistrements des M locuteurs de référence sont alignés temporellement par l'algorithme DTW. Le spectre interpolé est ensuite calculé de la façon suivante :

$$\begin{aligned} Y_{i,j} &= \sum_{k=1}^{M} w_k X_{k,i,j} \\ \sum_{k=1}^{M} w_k &= 1. \end{aligned}$$

Où $X_{k,i,j}$ est le paramètre spectral X_j de la trame i du locuteur k.

Les paramètres utilisés dans ce système sont les coefficients cepstraux et les "Log Area Ratio" (LAR).

L'imitation d'un nouveau locuteur à partir des locuteurs mémorisés, se fait de la manière suivante :

1. l'Enregistrement d'un mot du nouveau locuteur
2. l'Alignement temporel (DTW) de ce mot avec le même mot du locuteur de référence

3. la Minimisation de la fonction :

$$F(w_1, w_2, \cdots, w_M) = \sum_{i,j} \left(Y_{i,j} - y_{i,j}\right)^2 .$$

$Y_{i,j}$ étant le paramètre spectral j de la trame i du nouveau locuteur. Les w_i représentent donc la transformation spectrale.

4. L'utilisation des w_i pour la synthèse d'une élocution quelconque qui ressemble au nouveau locuteur.

Seuls des tests sur des mots isolés ont été effectués. La distance spectrale entre le locuteur cible et le locuteur synthétique est de 40% inférieure à la distance spectrale entre le locuteur cible et le locuteur de référence le plus "proche". Un seul mot du nouveau locuteur est utilisé pour calculer les w_i représentant la transformation spectrale.

3.1.4 Travaux de Abe et al

3.1.4.1 Conversion de voix par quantification vectorielle et dictionnaire de correspondances « *Mapping Codebook* » [2] Dans ces premiers travaux, Abe a utilisé un synthétiseur vocal de type vocodeur LPC : la parole est synthétisée en excitant un filtre tout pôles d'ordre 12 par une entrée constituée d'un bruit blanc pour les sons non voisés ou d'une entrée formée d'impulsions élémentaires répétées périodiquement pour les sons voisés. Cette approche vocodeur limite la qualité du signal obtenu. Par contre les modifications de fréquence fondamentale, de durée ou d'enveloppe spectrales sont simples à mettre en œuvre.

Le principe de transformation de voix proposé par Abe est le suivant :

- Dans la phase de transformation, Le signal source est analysé par prédiction linéaire. On applique une quantification vectorielle aux paramètres spectraux, en utilisant un dictionnaire spécifique au locuteur source. Puis chaque vecteur quantifié est remplacé par un nouveau vecteur censé copier au mieux le locuteur cible. Ce nouveau vecteur est lu dans un dictionnaire appelé par Abe « *Mapping Codebook* ».
  D'autre part, la fréquence fondamentale et l'énergie de chaque trame sont transformées par une technique similaire mais en utilisant une simple quantification scalaire. Enfin le signal transformé est synthétisé par la technique du vocodeur LPC.
- Dans la phase d'apprentissage deux corpus lexicalement identiques prononcés par chacun des deux locuteurs (A=source et B=cible) sont analysés trame par trame. L'enveloppe spectrale est estimée par prédiction linéaire. La fréquence fondamentale et l'énergie sont elles aussi calculées. Les vecteurs spectraux sont quantifiés vectoriellement (taille des dictionnaires de quantification vectorielle = 256 vecteurs).
  Il a bien sûr fallu auparavant calculer les dictionnaires de quantification vectorielle pour chacun des deux locuteurs.

Les vecteurs spectraux des deux corpus sont alignés temporellement par DTW. Cet alignement est fait mot par mot. De cette façon on établit des correspondances entre les vecteurs spectraux des corpus de A et de B. Pour chaque vecteur du dictionnaire de A on compte combien de fois ce vecteur a été mis en correspondance par DTW avec chacun des vecteurs du dictionnaire de B. On calcule donc ainsi des histogrammes de correspondances entre vecteurs des dictionnaires de A et B.

On construit de même des histogramme de correspondances (après quantification scalaire) de la fréquence fondamentale et de l'énergie.

Les dictionnaires de transformation : « *Mapping Codebook* » sont déterminés, pour la fréquence fondamentale et l'énergie, en faisant correspondre à chaque valeur de ces paramètres quantifiés, la valeur du maximum de l'histogramme de correspondance.

Le dictionnaire de correspondances pour les vecteurs spectraux associe à chaque vecteur de A soit le vecteur de B correspondant au maximum de l'histogramme de correspondance, soit un vecteur obtenu par combinaison linéaire des vecteurs de B pondérés par les valeurs de l'histogramme de correspondance. Abe a testé les 2 approches qui se sont révélées assez proches, la deuxième solution étant toutefois légèrement supérieure.

Des tests objectifs de distorsion spectrale et des tests d'écoute ont été réalisés pour évaluer le système de conversion. La distance spectrale entre 2 locuteurs est diminuée de 49% dans le cas de 2 locuteurs mâles, de 27% pour 2 locuteurs femmes, et de 66% pour une conversion homme femme.

Dans des travaux plus récents, Abe a amélioré cette approche en effectuant une quantification vectorielle « floue ».

Les tests ont été réalisés sur du japonais uniquement.

3.1.4.2 Approche segmentale [3] Cette méthode est une extension du système de conversion de la voix par quantification vectorielle et "dictionnaire de correspondances". La conversion de la voix par quantification vectorielle est capable de transformer les caractéristiques stationnaires d'un locuteur (l'enveloppe spectrale). Elle travaille sur des vecteurs spectraux obtenus sur des trames successives ayant une longueur fixe. L'analyse ne prend pas en compte la durée des unités acoustiques du signal de la parole.

Pour améliorer le système, M. Abe propose de ne pas utiliser des trames de longueur fixe comme unités de conversion mais des segments de longueur variable. Les segments extraits correspondent à 39 phonèmes.

Pendant l'étape d'apprentissage, un ensemble de modèles HMM est construit pour chaque locuteur (1 pour chaque phonème). Un tableau de correspondances entre les segments des deux locuteurs en est déduit.

Le signal de parole du locuteur A est analysé (LPC) et l'utilisation du HMM permet l'extraction des segments qui sont, par la suite, comparés avec un dictionnaire de segments préalablement appris pour le locuteur A. Chaque segment est ensuite remplacé par un segment de locuteur B. Le choix du segment du locuteur cible est déterminé dans le tableau de correspondances entre les segments des deux locuteurs. Le signal transformé est finalement synthétisé par concaténation des segments du locuteur B choisis.

Le système est très complexe et il n'est pas évident de l'utiliser pour générer des voix nouvelles intermédiaires.

Lors de tests subjectifs, la qualité de la voix obtenue et l'efficacité de la transformation sont jugées supérieures à la méthode utilisant la quantification vectorielle et un dictionnaire de correspondances. Cependant, en ce qui concerne la distorsion spectrale cette dernière méthode donne de meilleurs résultats.

3.1.4.3 Synthèse de parole par concaténation de segments acoustiques avec modification des formants [102] La structure désirée des formants d'un spectre à court-terme est spécifiée par un jeu de paramètres qui sont :

1. les fréquences des formants,
2. la largeur de bande des formants,
3. l'intensité des formants (l'amplitude).

La méthode proposée, permet de modifier la structure des formants de façon contrôlée. Il s'agit d'une technique itérative qui effectue d'abord le décalage nécessaire des fréquences des formants et ajuste ensuite leur intensité en changeant leur largeur de bande. Les modifications obtenues sont par la suite effectuées sur le signal original dans le domaine spectral (calculé par FFT). La synthèse se fait simplement par FFT inverse.

Cette technique est utile pour diminuer les discontinuités des formants lors de la concaténation des segments de parole et peut s'appliquer dans un système de conversion de la voix. La structure de formants désirée est alors celle du locuteur cible.

Les transformations sont appliquées sur la représentation originale du spectre, et non pas sur un spectre quantifié vectoriellement, ce qui est important pour la qualité de la voix synthétisée.

3.1.4.4 Conversion de voix par transformation à l'aide de règles de transformation de formants et de pente spectrale [101] Mizuno et Abe ont développé un algorithme de conversion de voix qui modifie les paramètres spectraux en respectant des règles de transformation des formants et de la pente spectrale. Cette méthode permet de modifier le timbre de la voix tout en gardant une bonne qualité du signal de parole.

Lors de la phase d'apprentissage, l'espace acoustique est divisé en 256 classes par quantification vectorielle des vecteurs spectraux des corpus d'apprentissage. Pour chaque classe on calcule les formants et la pente spectrale. La détermination des formants se faisant de façon automatique avec correction manuelle. Un dictionnaire de correspondances est calculé. Les correspondances entre les deux locuteurs sont ensuite étudiées : valeur de décalage des formants et de la pente spectrale. Les règles de conversion des formants et de la pente spectrale sont ainsi déterminées pour chaque classe.

Pendant l'étape de conversion de la voix, les vecteurs spectraux du signal d'entrée sont classés par quantifiés vectorielle. Les formants et la pente spectrale sont extraits en s'aidant des valeurs de référence pour cette classe. Les règles de transformation apprises au préalable sont appliquées à ces formants et à cette pente spectrale, et non à leur valeur quantifiée. La quantification vectorielle ne sert ici qu'à classer la trame de signal pour déterminer la transformation à appliquer à ces formants.

La conversion n'est appliquée que sur les zones voisées ce qui diminue le nombre de sous-espaces nécessaires.

Lors des tests subjectifs avec notation sur une échelle à cinq niveaux, la voix synthétique a été jugée de très bonne qualité et très proche de la voix du locuteur cible.

3.1.4.5 Conversion de la langue d'un locuteur [4] La conversion de la voix multilingue est une extension des techniques de transformation de voix, intéressante pour les systèmes de traduction automatique .

Abe et al ont étudié la conversion Japonais-Anglais en essayant de conserver le timbre de la voix du locuteur original.

En plus des problèmes de conversion de voix et de contrôle de l'individualité de la voix, certaines difficultés spécifiques apparaissent lorsque l'on passe d'une langue à l'autre. Certains phonèmes présents dans une langue, n'existent pas dans l'autre, les fréquences d'utilisation des phonèmes sont

différentes. Même si un locuteur maîtrise les deux langues, ses habitudes linguistiques varient en fonction de la langue.

En multipliant les tests sur un locuteur bilingue, les auteurs ont conclu que la différence spectrale Japonais-Anglais pour un locuteur est plus petite que la variabilité inter-locuteur de deux personnes de même langue maternelle. Des évaluations subjectives ont montré que les deux espaces acoustiques japonais et anglais pour le même locuteur se recoupent.

Toutefois le système de conversion de la voix inter-langue proposé introduit une diminution de la qualité de la voix. Les dictionnaires utilisés pour la conversion de la voix sont en effet calculés pour une langue et ne sont pas optimisés pour une autre langue.

3.2 Travaux personnels réalisés sur la conversion de voix pour des applications en synthèse

Mes travaux sur la conversion de voix ont été réalisés dans le cadre d'une convention CNET d'une durée de 3 ans (1994-1996) en collaboration avec l'ENST (E. Moulines), l'INRIA (J.Levy-Vehel) et le CNET (O.Boeffard, B. Cherbonnel, sous la direction de C. Sorin) .

3.2.1 Position du problème

La synthèse de parole à partir du texte se décompose en 2 phases : une étape linguistique et une étape acoustique. L'étape linguistique analyse le texte et génère la suite des phonèmes du message à émettre ainsi que les paramètres prosodiques associés.

Deux méthodes sont couramment utilisées pour l'étape acoustique : la synthèse par règles et la synthèse par concaténation d'unités linguistiques telles que les diphones. La seconde approche donne les meilleurs résultats. Elle consiste à assembler des segments de parole (diphones par exemple) qui sont choisis dans un répertoire. Ce dernier est obtenu en segmentant des enregistrements de signal prononcés par un locuteur unique. Cette segmentation est fastidieuse et nécessite l'aide d'un phonéticien. Le synthétiseur parle avec la voix du locuteur qui a servi à la création du répertoire.

L'objectif de l'étude s'inscrivait dans l'ensemble des recherches relatives au développement de nouvelles voix de synthèse, pour des synthétiseurs utilisant la concaténation d'unités acoustiques. Il s'agissait d'étudier la possibilité de créer de nouvelles voix conservant un timbre naturel à partir de voix existantes. La recherche devait porter sur les transformations spectrales et se faire en complément d'autres travaux menés au CNET et à l'ENST sur les modifications de prosodie. J'ai travaillé à l'ESIEE avec P. Jardin (enseignant-chercheur ESIEE), Y. Stylianou (post-doc) et plusieurs étudiants en stage long dont E. Steinbach, J.-P. Goldman et F. Tonelli. Y. Styliannou est arrivé à l'ESIEE après avoir terminé sa thèse sous la direction d'Éric Moulines à l'ENST.

Lors de ce travail, nous avons essayé de convertir la voix d'un locuteur A en celle d'un locuteur B, dans le but de créer une ou plusieurs voix intermédiaires entre A et B.

3.2.2 Travaux réalisés

J'ai développé, en collaboration avec P. Jardin et les étudiants, 4 techniques de conversion de voix, par transformations spectrales utilisant les méthodes suivantes :

– Réseaux de neurones multicouches (notée NNETS[2]),

[2]NNETS=Neural NETworks.

- Mise en correspondance de dictionnaires de quantification vectorielle (notée VQM[3])
- Alignement fréquentiel avec compensation d'amplitude (notée DFWA[4]),
- Régression linéaire multi-dimensionnelle (notée LMR[5])

Ces techniques ont été comparées avec une méthode développée à l'ENST utilisant une modélisation des vecteurs spectraux par mélange de gaussiennes (notée GMM)[6] . Les résultats de ces comparaisons ont été présentés à la conférence ICSLP en 1996 [21].

Les techniques NNETS, VQM, LMR et GMM ont été développées complètement et testées sur les bases de données du CNET.

La technique DFWA, ayant donné après les 1^{ers} tests des résultats inférieurs à ceux de la LMR, n'a pas été testée sur les bases de données CNET mais seulement sur une petite base de voyelles.

Les recherches précédentes ont montré qu'une conversion linéaire de l'axe des fréquences n'est pas suffisante pour transformer le spectre d'un locuteur en celui d'un autre locuteur et que la transformation devrait dépendre du type de son à modifier.

Nous avons donc développé plusieurs méthodes de transformation non-linéaires, et nous les avons comparé avec une méthode linéaire de régression linéaire multi-dimensionnelle.

Pour prendre en compte la dépendance en fonction du son, nous avons développé des méthodes en sous-classes, c'est-à-dire dans lesquelles les vecteurs cepstraux sont pré-classifiés par quantification vectorielle avant la transformation qui est spécifique à chaque classe. Le cas limite de cette approche en sous-classes est constitué de 2 sous-classes correspondant aux trames voisées et non-voisées.

Par ailleurs, pour les méthodes NNETS et LMR, nous avons mis au point une version prenant en compte la coarticulation. C'est-à-dire que la transformation d'un vecteurs dépend des 2 vecteurs adjacents. Nous avons appelé ces méthodes NNETS et LMR en contexte.

3.2.2.1 Bases de données, type d'analyse spectrale et modélisation des signaux utilisés

Bases de données

Les travaux étaient destinés à être intégrés dans un système de synthèse à partir du texte développé au CNET et travaillant par concaténation de diphones et de polyphones.

L'idée était d'appliquer les méthodes de transformation développées au répertoire d'unités acoustiques (diphones et polyphones[7]) du synthétiseur. Le CNET nous a donc fourni une base de données formée de logatomes (1280 logatomes) à partir desquels on pouvait extraire les diphones du synthétiseur. Nous avons reçu un corpus de données pour deux locuteurs masculins OB et RG constitué (pour chacun des locuteurs) de :

- 33 fichiers de 40 logatomes au format PRL[8].
- 33 fichiers de segmentation associés aux précédents, qui donnent en particulier :
  la description phonétique des logatomes, l'emplacement des diphones au cœur de ceux-ci, les frontières (en numéros d'échantillons) de ces logatomes et des phonèmes les constituant.
- 1 fichier de 10 phrases phonétiquement équilibrées et le fichier de segmentation (en phrases) correspondant.

[3]VQM=Vector Quantization Mapping.

[4]DFWA=Dynamic Frequency Warping with Amplitude.

[5]LMR = Linear Multidimensionnal Regression.

[6]GMM = Gaussian Mixture Model.

[7]Par la suite j'utilise le terme diphone pour désigner les unités acoustiques du synthétiseur, aussi bien les diphones que les polyphones.

[8]Le format PRL est un format CNET.

La base contenait une occurrence de chaque diphone.

Dans un premier temps nous avons utilisé cette base pour apprendre et appliquer les transformations proposées. Les premiers résultats obtenus étant peu satisfaisants nous avons mis en cause la mauvaise répartition phonétique présentée par cette base. On trouvait plus de 1700 occurrences du phonème le plus représenté, c'est à dire EU, alors qu'une quinzaine de phonèmes apparaissaient moins de 85 fois.

Nous avons donc créé une base de données de diphones en ne gardant que le diphone utile compris dans chaque logatome. On a perdu en quantité de données mais la répartition phonétique de la base de diphones est très équilibrée (35 à 37 occurrences de chaque phonème).Cette propriété à priori intéressante peut en fait défavoriser les méthodes basées sur les réseaux de neurones qui vont accorder le même poids aux coûts provenant de l'association de vecteurs de probabilités naturelles très diverses et d'importances (en terme de discrimination auditive) différentes. Il est difficile de partager la base obtenue en base d'apprentissage et base de test puisqu'il n'y a qu'un exemplaire de chaque diphone. Nous avons choisi d'extraire un vecteur sur 5 de la base pour constituer une petite base de test qui nous permette de contrôler l'apparition éventuelle (et rarement effective dans nos essais) d'un sur-apprentissage des réseaux de neurones et de tester les méthodes sur une base différente de celle d'apprentissage.

Le CNET nous a ensuite fourni une $2^{\text{ème}}$ base de données, formée de 89 phrases phonétiquement équilibrées, pour 2 locuteurs masculins OB et un $2^{\text{ème}}$ locuteur que l'on note RG bien qu'il soit différent de celui de la base de diphones.

Types d'analyse spectrale et de modélisation des signaux utilisés

Au démarrage de l'étude, nous avons utilisé une simple analyse-synthèse LPC. Cette approche repose sur le modèle source-filtre de production de la parole pour lequel on admet que le conduit vocal se comporte comme un filtre excité par un signal source qui est l'onde glottique dans le cas des sons voisés. On détermine la fonction de transfert (modèle auto-régressif) du filtre par prédiction linéaire sur des segments successifs du signal de parole et le signal source optimal est alors le résiduel de prédiction correspondant.

L'utilisation de ce modèle pour la conversion de voix est très simple dans son principe puisqu'il suffit de modifier l'un ou (et) l'autre de ses éléments (résiduel ou filtre) pour synthétiser une voix différente de l'originale. La modification du résiduel doit essentiellement entraîner des transformations sur la prosodie du message et celle du filtre qui correspond approximativement à la fonction de transfert du conduit vocal permet théoriquement un ajustement des formants modifiant ainsi le timbre du locuteur. Le schéma général d'une telle conversion est donné par la figure II.1 :

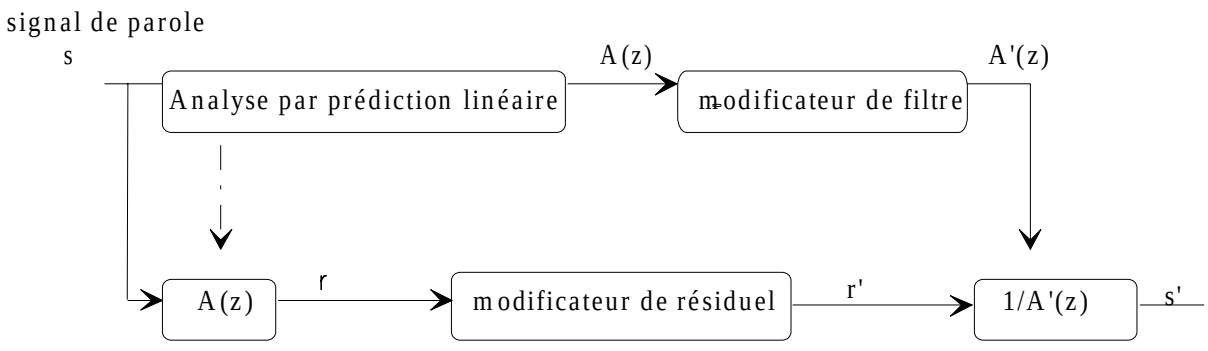


FIG. II.1 – Principe de la conversion de voix.

Cette approche est discutable, puisqu'il est bien connu que le résiduel LPC est généralement quasiment compréhensible, c'est-à-dire qu'il contient encore beaucoup d'informations sur l'enveloppe spectrale du signal. Nous avons toutefois préféré cette approche à celle utilisée par Abe qui excitait le filtre de synthèse par des pulses périodiques ou du bruit, pour obtenir un signal synthétique de meilleure qualité.

Les coefficients que nous avons utilisés pour apprendre et réaliser les transformations spectrales sont les coefficients cepstraux LPCC, car la distance euclidienne sur ces coefficients est bien corrélée avec la perception auditive.

Les coefficients a_i du filtre de synthèse peuvent être calculés de deux manières, à partir des coefficients cepstraux :

- Soit en utilisant la relation de récurrence inverse de celle qui permet de calculer les LPCC à partir des a_i. Mais dans ce cas, le filtre obtenu $1/A(z)$ n'est pas forcément stable.
- Soit en calculant le logarithme de la densité spectrale de puissance du signal par FFT sur les coefficients cepstraux c_i, ce qui permet d'obtenir p coefficients d'autocorrélation et finalement p coefficients de prédiction linéaire a_i. Dans la mesure où on utilise une méthode d'autocorrélation pour la prédiction linéaire, la stabilité du filtre1/A(z) est garantie. C'est cette méthode que nous avons utilisée.

Lorsque le logiciel HNM[9], développé à l'ENST par Y. Styliannou et É. Moulines, a été disponible, nous avons remplacé cette analyse-synthèse LPC par une modélisation HNM [123].

La technique HNM modélise le signal de parole $x(t)$ comme la somme de 2 composantes : une composante purement harmonique $x_h(t)$ (jusqu'à une fréquence f_{max}) et une composante de bruit $x_n(t)$ modulé en amplitude.

$$x(t) = x_h(t) + x_n(t)$$

Pour les sons voisés le spectre est divisé en 2 bandes limitées par une fréquence variable f_{max} appelée fréquence maximale de voisement. La partie basse du spectre est une somme de sinusoïdes de fréquences harmoniques de la fréquence fondamentale. La partie haute du spectre est un bruit dont l'énergie passe par des maxima de manière synchrone au pitch.

La partie harmonique est modélisée par une somme de sinusoïdes de fréquences multiples de la fréquence fondamentale. Si on note $a_k(t)$ et $\Phi_k(t)$ l'amplitude et la phase de la $k^{\text{ème}}$ harmonique à l'instant t :

$$x_h(t) = \sum_{k=1}^{K(t)} a_k(t) \cos(\Phi_k(t)).$$

Le nombre total d'harmoniques $K(t)$ à l'instant t dépend de la fréquence fondamentale $F_0(t)$.

$$\frac{d\Phi_k(t)}{dt} = 2\pi\ k\ F_0.$$

Les paramètres du modèle sont mis à jour régulièrement en des instants d'analyse $t_{a,i}$ synchrones au pitch ou espacés d'une longueur fixe.

Les amplitudes des sinusoïdes qui constituent la partie voisée sont déterminées par la minimisation d'un critère temporel de moindres carrés pondérés entre le signal original et le signal synthétique formé de la somme des sinusoïdes.

[9] HNM = Harmonic plus Noise Model.

Pour les composantes non-voisées (trames non-voisées ou parties hautes des spectres de trames voisées), l'enveloppe du spectre est obtenue en calculant les énergies à la sortie d'un banc de filtres. Le contour temporel d'énergie est modélisé par une fonction linéaire par morceau.

La synthèse HNM se fait de manière additive pour la partie harmonique $x_h(t)$. On assure une évolution continue des paramètres par interpolation entre les instants d'analyse. La partie bruit est synthétisée par une technique OLA[10] pour la composante bruit $x_n(t)$.

La modélisation HNM se prête bien aux modifications prosodiques, il suffit de modifier les paramètres du modèle. En fonction des modifications de pitch et de vitesse d'élocution désirées on génère une suite d'instants de synthèse $t_{s,i}$ à partir des instants d'analyse $t_{a,i}$, et on crée le signal synthétique en interpolant les paramètres entre les instants de synthèse.

L'analyse-synthèse HNM peut être faite de manière synchrone ou asynchrone à la période fondamentale du signal. L'analyse synchrone donne de meilleurs résultats, mais on ne peut l'appliquer dans la phase d'apprentissage des transformations car les 2 locuteurs n'ont pas la même fréquence fondamentale.

L'analyse est donc asynchrone lors de l'apprentissage et synchrone lors de la phase de transformation.

À partir des coefficients du modèle décrivant la partie harmonique et la partie bruit du spectre, on déduit des coefficients cepstraux discrets et réciproquement [123].

Un modèle continu de l'enveloppe spectrale qui connecte les amplitudes des raies harmoniques et les énergies du banc de filtre pour la composante de bruit $x_n(t)$, est estimé par une méthode de cepstre discret régularisé [39].

On détermine une enveloppe spectrale continue définie par des coefficients cepstraux c_i. La valeur des coefficients cepstraux est déterminé par un critère de moindres carrés, appliqué sur les différences entre l'enveloppe continue et les valeurs sur l'ensemble discret de fréquences. Soit $S(f, \mathbf{c})$ l'enveloppe spectrale pour un vecteur de coefficients cepstraux $\mathbf{c}$:

$$\log\left(S(f, \mathbf{c})\right) = c_0 + 2\sum_{i=1}^{P} c_i cos(2\pi i f).$$

le critère s'écrit :

$$\min \sum \| \log a_k - \log\left(|S(f_k, \mathbf{c})|\right) \|^2.$$

Ce critère est régularisé pour éviter d'obtenir des spectres présentant de nombreuses oscillations quand le problème est mal conditionné, c'est-à-dire que le nombre de coefficients cepstraux est proche du nombre d'harmoniques. De plus, pour améliorer le critère dans les basses fréquences, où l'oreille présente une plus grande sensibilité aux défauts, on transforme l'axe des fréquences de manière non-linéaire en utilisant une échelle de Bark.
L'enveloppe spectrale est ainsi décrite par un jeu de paramètres qui sont analogues au coefficients MFCC standards. On les note c_i dans la suite du document. On a utilisé des vecteurs cepstraux de longueur $p = 16$. Dans nos expériences de transformation du timbre de la voix, nous avons modifié ces vecteurs de coefficients cepstraux.

[10] OLA = *OverLap and Add.*

Alignement des vecteurs cepstraux par DTW :
Même si deux locuteurs prononcent le même corpus, il y peut y avoir une grande différence entre leurs débits d'élocution. Pour trouver une correspondance entre deux locuteurs il est donc d'abord nécessaire d'aligner temporellement les deux corpus d'apprentissage. Cet alignement temporel se fait logatome par logatome ou phrase par phrase, selon le corpus, et nécessite la segmentation du corpus d'apprentissage.

3.2.2.2 Transformation par réseaux de neurones multi-couches, méthode NNETS Cette méthode utilise un réseau de neurones pour effectuer la conversion des vecteurs cepstraux.

L'idée est d'entraîner un réseau de neurones à transformer les vecteurs $\mathbf{c_n^S}$ d'un locuteur source en vecteurs $\mathbf{c_n^C}$ correspondant au locuteur cible sur la base d'apprentissage formée d'un ensemble de couples ($\mathbf{c_n^S}$, $\mathbf{c_n^C}$) puis d'utiliser ce réseau appris pour effectuer la transformation pour un vecteur $\mathbf{c_n^S}$ quelconque (en comptant sur les capacités de généralisation du réseau).

Le réseau utilisé pour la conversion de voix est un perceptron multicouches. Ce choix est guidé par la simplicité de mise en œuvre et par l'adéquation au problème posé qui est d'opérer une fonction non linéaire entre les couches d'entrée et de sortie.

Toutefois le nombre de couches cachées et le nombre de neurones sur ces couches sont des paramètres à déterminer. Il n'existe pas de règle exacte pour effectuer ce choix qui sera guidé par l'expérience et par le souci d'avoir un rapport " nombre de données d'apprentissage"/"nombre de paramètres inconnus " suffisamment élevé.

La sortie d'un neurone des couches cachées du réseau est une fonction non linéaire sigmoïde (ou tangente hyperbolique)de l'activité du neurone, l'activité du neurone étant la somme pondérée de ses entrées et d'un biais.

La couche de sortie du réseau utilise une fonction linéaire de façon à pouvoir générer des vecteurs de sortie non bornés.

L'équation de la fonction sigmoïde, pour un biais b et pour une valeur x de la somme pondérée des entrées du neurone, est la suivante :

$$f(x,b) = \frac{1}{1+\exp^{-(x+b)}}.$$

Où $f(x,b)$ est la sortie du neurone.
L'équation de la fonction de type tangente hyperbolique s'écrit :

$$f(x,b) = \alpha \frac{1-e^{\lambda(x+b)}}{1+e^{\lambda(x+b)}}. \tag{II.1}$$

Cette dernière fonction est comprise entre $-\alpha$ et $+\alpha$. Les valeurs α et λ sont des constantes scalaires.

Les couches d'entrée et sortie du réseau sont respectivement constituées de vecteurs de coefficients cepstraux des locuteurs source et cible.

C'est l'ensemble W des poids sur les connections entre neurones et des biais B sur les neurones qui détermine la relation entre l'entrée I et la sortie O du réseau (O=F(I,W,B)).

Cet ensemble (W, B) doit être appris lors de la phase d'entraînement supervisé durant laquelle on minimise l'erreur quadratique entre les vecteurs de sortie du réseau et les vecteurs cibles.

L'algorithme d'apprentissage est celui de rétropropagation du gradient de l'erreur.

(les notations sont volontairement incomplètes, non indicées, par souci de simplifier la lecture).

Nous avons testé plusieurs variantes de l'algorithme de rétropropagation de gradient : global, stochastique, semi-stochastique.

On dispose d'une base de N couples ($\mathbf{c_n^S}$, $\mathbf{c_n^C}$), n=1...N, obtenus par alignement temporel, constituant les vecteurs d'entrée (source) et de sortie désirée correspondante (cible).

On peut définir une erreur élémentaire pour chaque présentation d'un couple ($\mathbf{c_n^S}$, $\mathbf{c_n^C}$) :

$$E_n = \frac{1}{p}\left(\mathbf{c_n^C} - F\left(\mathbf{c_n^S}, W, B\right)\right)^2$$

L'erreur totale sur l'ensemble de la base vaut alors :

$$E = \frac{1}{N}\sum_{n=0}^{N-1} E_n.$$

Dans l'algorithme global, on modifie les poids du réseau à chaque passage de la base entière.

Cet algorithme est amélioré en adaptant le pas de l'algorithme du gradient à l'évolution de l'erreur entre 2 itérations.

Dans l'algorithme stochastique, on modifie les poids à chaque présentation d'un couple ($\mathbf{c_n^S}$, $\mathbf{c_n^C}$). Le temps de convergence est plus faible dans ce cas.

Dans l'algorithme semi - stochastique, on effectue une classification de la base en N_c classes (par un algorithme de quantification vectorielle par exemple). On modifie les poids à chaque passage de N_c couples (1 par classe choisi aléatoirement). La complexité de l'algorithme est N_c/N fois celle de la méthode globale et l'algorithme est celui qui converge le plus rapidement. Une itération correspond à une minimisation de l'erreur sur un ensemble de vecteurs représentatif de toute la base.

Modification des pas au cours de l'apprentissage

Les pas utilisés par l'algorithme du gradient pour les poids et pour les biais, sont modifiés au cours de l'apprentissage de la manière suivante :

- Après une itération, si le rapport entre la nouvelle valeur du critère et l'ancienne est supérieur à un seuil noté s, les pas sont diminués.
- Inversement si le rapport des critères avant et après l'itération est inférieur à $2 - s$, les pas sont augmentés.
- Dans l'intervalle, c'est a dire si le rapport est proche de 1, les pas ne sont pas modifiés.

D'autre part, les biais et les poids du réseau ne sont modifiés que si l'itération n'a pas trop dégradé les performances, c'est à dire si le rapport entre le nouveau et l'ancien critère est inférieur à s.

Initialisation du réseau

Nous avons constaté qu'une initialisation aléatoire des poids et biais du réseau n'était pas très efficace (temps de convergence très grand), aussi avons nous imaginé une nouvelle initialisation qui utilise le fait que l'entrée et la sortie du réseau sont assez proches dès le début. L'idée est d'essayer d'initialiser le réseau de façon que sa fonction de transfert globale soit pratiquement l'identité. Pour cela, nous avons initialisé le réseau de la manière suivante :

Les biais sont initialisés par une valeur aléatoire uniformément répartie entre $+\epsilon$ et $-\epsilon$, où $-\epsilon$ est une petite valeur, en pratique égale à 0.005.

Les poids (dans le cas de la fonction de transfert tangente hyperbolique II.1) sont initialisés soit par $\frac{1}{pente} + \epsilon$, soit par ϵ , où $pente$ est la pente de la fonction de transfert des neurones à l'origine (le gain à l'origine de la fonction sigmoïde). Les poids initiaux valent donc soit $\frac{1}{pente}$ soit 0 à une petite constante près.

Plus précisément, en appelant :

- N_i et N_{i-1} le nombre de neurones sur les couches i et $i-1$.
- $S_{i-1,j}$ la sortie du neurone j de la couche $i-1$.
- $X_{i,k}$ l'activité du neurone k de la couche i.
- $W_{k,j,i}$ le poids reliant la sortie du neurone j de la couche $i-1$ à l'entrée du neurone k de la couche i.
- $B_{i,k}$ le biais du neurone k de la couche i.
- f_s la fonction de transfert des neurones (sigmoïde).

on peut écrire :

$$X_{i,k} = \sum_{j=1}^{N_{i-1}} S_{i-1,j} W_{k,j,i} + B_{i,k}.$$

Et on initialise avec :

$$\begin{aligned} W_{k,k,i} &= \frac{1}{pente} + \epsilon. \\ W_{k,j,i} &= \epsilon \text{ si } j \neq k \end{aligned}$$

On peut alors écrire :

$$\begin{aligned} S_{i,k} &= f_s(X_{i,k}). \\ S_{i,k} &\simeq pente\ X_{i,k}, \text{ développement limité à l'ordre 1} \\ \text{si } k \leq N_{i-1}\ S_{i,k} &\simeq pente\ W_{k,k} S_{i-1,k} \simeq pente \frac{1}{pente} S_{i-1,k} \simeq S_{i-1,k}, \\ \text{si } k > N_{i-1}\ S_{i,k} &\simeq 0. \end{aligned}$$

Le réseau converge plus vite avec cette initialisation qu'avec une initialisation complètement aléatoire.

Normalisation des vecteurs

L'apprentissage du réseau s'effectue plus rapidement si l'on travaille avec des vecteurs normalisés.

On estime la moyenne et l'écart type des vecteurs cepstraux de la source et de la cible sur la base de données et on utilise ces estimations pour normaliser les vecteurs cepstraux la source et de la cible.

Soit $\mathbf{m_S}$, $\sigma_\mathbf{S}$, $\mathbf{m_C}$, $\sigma_\mathbf{C}$ les moyennes et écarts types de la source et de la cible.

Pour l'apprentissage, la base de données de coefficients cepstraux de la source est normalisée avec la moyenne et l'écart type de la source, de même la cible est normalisée avec la moyenne et l'écart type de la cible.

Dans la phase de transformation, on normalise les vecteurs de coefficients cepstraux de la source avec la moyenne et l'écart type de la source, on les transforme avec le réseau de neurones travaillant sur vecteurs normalisés, puis on dénormalise le résultat en utilisant la moyenne et l'écart type de la cible.

Étude de la structure à employer pour le réseau de neurones

Nous avons utilisé un réseau de neurones multicouches : MLP[11].

[11] MLP = *Multi-Layer Perceptron.*

Un réseau 2 couches, formé d'une couche cachée et d'une couche de sortie, suffit a priori pour approximer une fonction quelconque. Toutefois, un réseau comportant plus de couches cachées peut être plus efficace en nombre de neurones.

Pour choisir une structure, nous avons fait différents tests, sur le nombre de couches et le nombre de neurones par couche.

Les tests ont été réalisés avec la base de données de diphones et avec l'algorithme semi-stochastique utilisant un dictionnaire de taille 256.

On a comparé les résultats obtenus avec différentes structures au bout de 1000 itérations (1 itération correspond ici au passage de 256 vecteurs dans le réseau), en partant toujours d'un fichier de poids et de biais initiaux calculé de la manière décrite précédemment 3.2.2.2.

Pour caractériser la structure d'un réseau on note le nombre de neurones en entrée et sur les couches successives, de la manière suivante :
Le réseau 15 12 12 12 15 représente un réseau ayant 15 entrées, puis 4 couches de respectivement 12, 12, 12, 15 neurones.

Les meilleurs résultats ont été obtenus avec les structures 15 12 12 12 15 et 15 15 15 15 qui ont respectivement 648 et 675 neurones, 51 et 45 biais, pour 3 et 2 couches cachées. Nous avons utilisé la première structure dans les expériences réalisées avec une seule classe de vecteurs cepstraux, et la deuxième structure pour les expériences en sous-classes. En effet dans ce dernier cas, nous avons utilisé les routines Matlab qui ne supportaient pas les réseaux à plus de 2 couches cachées. Dans le cas à 1 seule classe, nous ne pouvions pas utiliser Matlab, car la taille de la base de données (de l'ordre de 30 000 vecteurs) dépassait la capacité de Matlab.

3.2.2.3 Transformation de dictionnaire de quantification vectorielle, méthode VQM La méthode est inspirée des premiers travaux de Abe, elle consiste à construire un dictionnaire de quantification vectorielle sur chacun des ensembles de vecteurs cepstraux du locuteur source $\mathbf{c_n^S}$ et du locuteur cible $\mathbf{c_n^C}$, puis à associer à chaque vecteur quantifié source un vecteur quantifié cible. Les correspondances en nombre fini ainsi créées sont stockées dans un *mapping codebook*, ou dictionnaire de correspondances.

Les modifications apportées par rapport à la démarche de Abe sont les suivantes :

- le signal est analysé et synthétisé de manière différente. Abe excitait le filtre de synthèse $1/A(z)$ par une suite d'impulsions périodiques pour un son voisé, ou par un bruit blanc pour un son non voisé. Dans notre système, nous utilisons une modélisation HNM où la modification de l'enveloppe spectrale se répercute sur l'amplitude des sinusoïde du modèle. La prosodie est facilement modifiée, en modifiant directement les paramètres (fréquence fondamentale par exemple) dans le modèle.
- Nous avons utilisé un dictionnaire de correspondances contenant des vecteurs naturels au lieu de vecteurs quantifiés transformés. Ce qui améliore la qualité subjective du signal transformé.
- nous avons utilisé des dictionnaires de quantification et de correspondance différents pour différentes classes de sons. Nous avons par exemple séparé les sons voisés ou non-voisés, ou bien pré-classifié chaque trame de signal en N classes (avec N entre 4 et 64) formées chacune de vecteurs proches acoustiquement.

Quantification vectorielle
La quantification vectorielle est effectuée sur les coefficients cepstraux. Le coefficient $c(0)$ lié à l'énergie de la trame n'est pas pris en compte dans la quantification. Il est soit directement recopié dans la voix transformée, soit modifié à part.

La taille des corpus dont nous disposions, ne nous a pas permis de construire des dictionnaires de taille supérieure à 1024 vecteurs. Cette taille est encore plus limitée quand on travaille en sous-classe. Pour la classe non voisée, par exemple, qui contenait moins de 5000 vecteurs nous nous sommes limité à un dictionnaire de taille 128.

Dictionnaire de mise en correspondance « *Mapping codebook* »
Un dictionnaire de correspondances est un dictionnaire qui met en correspondance un vecteur de paramètres (ici cepstraux) d'un locuteur source avec des paramètres modifiés qui sont censés représenter ce que seraient les paramètres du locuteur cible.

Une fois les corpus d'apprentissage segmentés, analysés et alignés temporellement, les correspondances entre les deux locuteurs sont accumulées dans un histogramme. C'est à dire que pour chaque couple de vecteurs (A_i, B_i), où A_i est un vecteur cepstral du dictionnaire du locuteur A et B_i un vecteur cepstral du dictionnaire du locuteur B, nous comptons le nombre de fois $N_{A_iB_j}$ où ces vecteurs ont été mis en correspondance par la DTW.

Après le calcul de l'histogramme, on peut créer un tableau qui contient, pour chacun des vecteurs du dictionnaire du locuteur A, un vecteur "correspondant du locuteur B" . Ce tableau est le dictionnaire de correspondances entre les deux locuteurs. Ce dictionnaire de conversion contient les règles de transformations spectrales. Il a la même taille que les dictionnaires de quantification vectorielle.

Nous avons testé les deux types de "dictionnaire de correspondances" proposés par Abe, à savoir :

– associer à chaque vecteur A_i, le vecteur B_j correspondant au nombre de mises en correspondance avec A_i maximale ($N_{A_iB_j}$ max).

$$A_i \longrightarrow B_j \text{ tel que } N_{A_iB_j} \text{ max}$$

on parlera alors de "max-mapping".

– associer à chaque vecteur A_i, la combinaison linéaire de tous les vecteurs B_j qui ont été mis en correspondance avec A_i ($N_{A_iB_j}$ non nul), pondérée en fonction du nombre d'occurrences $N_{A_iB_j}$:

$$A_i \longrightarrow \frac{\sum_j N_{A_iB_j} B_j}{\sum_j N_{A_iB_j}}.$$

on parlera de weighted-mapping.

Nous avons ensuite essayé plusieurs modifications de cette méthode en vue d'améliorer ses performances. Nous avons testé les modifications suivantes :

– Utilisation de dictionnaires de quantification vectorielle et de correspondances différents selon le voisement ou la classe du segment de signal,
– Itération de l'alignement temporel,
– Utilisation d'un dictionnaire de correspondances « naturalisé »,

Itération de l'alignement temporel
Nous avons étudié la possibilité d'améliorer la transformation en améliorant l'alignement temporel entre voix source et voix cible. Pour ce faire nous avons réitéré la méthode en calculant une première transformation (un mapping codebook) puis en effectuant cette transformation sur la base d'apprentissage source. Nous disposions alors d'une base intermédiaire pour laquelle nous avons recommencé

l'alignement temporel avec la base cible. Nous espérions de la sorte réaliser un meilleur alignement phonétique.

Nous avons alors construit l'histogramme des correspondances entre cible et source originale (obtenue par transformation inverse de la source utilisée pour l'alignement).

L'utilisation du rebouclage de l'alignement temporel n'a pas suffisamment amélioré les résultats pour que nous retenions cette approche.

Utilisation d'un dictionnaire de correspondances naturalisé
L'idée de cette transformation est d'obtenir des vecteurs transformés plus "naturels" que ceux obtenus après la transformation (max ou weighted mapping). Ces vecteurs sont en effet quantifiés (max mapping) ou formés par moyennage de vecteurs quantifiés.

Nous avons donc construit un nouveau dictionnaire de correspondances constitué de "vrais" vecteurs cibles (non quantifiés) de la base d'apprentissage cible. Ces vecteurs sont choisis comme les plus proches des vecteurs fournis par le mapping normal.

la distance source transformée - cible est bien sûre légèrement meilleure avec le dictionnaire « weighted mapping » qu'avec le dictionnaire naturel.

Les tests subjectifs ont montré que la qualité de la voix synthétique est meilleure avec le dictionnaire naturel.

3.2.2.4 Méthode de transformation de l'enveloppe spectrale par régression linéaire multidimensionnelle LMR De façon à vérifier la pertinence de l'utilisation d'une transformation non linéaire par réseau de neurones ou autre méthode décrite précédemment, nous avons effectué le même travail en utilisant une transformation linéaire (notée par la suite LMR : Linear Multiple Regression). La transformation linéaire a été calculée sur des coefficients cepstraux normalisés.

Principe de la LMR
Soit un vecteur de coefficients cepstraux normalisés de la source $\mathbf{c_S}$, il est transformé linéairement par la matrice $\mathbf{M}$ en un vecteur $\mathbf{c_T}$:

$$\mathbf{c_T} = \mathbf{M}\ \mathbf{c_S}$$

Dans notre cas les dimensions utilisées sont (15,1) pour $\mathbf{c_S}$ et $\mathbf{c_T}$, et(15,15) pour $\mathbf{M}$.

Étant données une base de données source formée de N vecteurs cepstraux normalisés de dimension p, organisés en une matrice $\mathbf{C_S}$ de dimension (p, N) et une base de données cible formée de N vecteurs cepstraux normalisés de dimension p, organisés en une matrice $\mathbf{C_C}$ de dimension (p, N), le problème est de trouver la matrice $\mathbf{M}$ qui minimise un critère J des moindres carrés entre la base de données source transformée par $\mathbf{M}$ et la base de données cible.

$$\min J = \min \sum_{i=1}^{N} \sum_{j=1}^{p} \left(\mathbf{c_C}(\mathbf{i},\mathbf{j}) - \mathbf{c_T}(\mathbf{i},\mathbf{j})\right)^2 .$$

La solution optimale pour M est :

$$\mathbf{M} \ = \ \mathbf{C_C}{\mathbf{C_S}}^T \left(\mathbf{C_S}{\mathbf{C_S}}^T\right)^{-1} .$$

3.2.2.5 Apprentissage et transformation en sous-classes Nous avons testé certaines des transformations spectrales en utilisant une pré-classification des vecteurs cepstraux puis une transformation spécifique à chaque classe.

Nous avons étudié le cas de 2 classes obtenues par classification selon le voisement (classe voisée et classe non-voisée).
Taille de la classe voisée $\leq$ 25000 vecteurs
Taille de la classe non voisée $\leq$ 5000 vecteurs

Nous avons aussi appliqué ce principe avec 4, 16 et 64 classes.

La classification des vecteurs s'effectue en utilisant un dictionnaire de quantification vectorielle obtenu avec l'algorithme LBG (Lindo, Buzzo, Gray), une classe étant constituée par l'ensemble des vecteurs les plus proches d'un des représentants du dictionnaire.

La classification est réalisée en prenant en compte les vecteurs de la source uniquement. Après l'alignement temporel, on affecte les vecteurs de la cible à la même classe que les vecteurs de la source avec lesquels ils sont alignés.

Un apprentissage est effectué sur chaque classe. L'ensemble d'apprentissage pour une classe est constitué par les paires de vecteurs (source, cible) alignés temporellement, pour lesquelles le vecteur de la source appartient à la classe en question.

Dans le cas des méthodes où on utilise des vecteurs normalisés, pour la transformation d'un vecteur il faudra d'abord le normaliser avec les moyennes et écarts type de la source, puis le classer et le transformer avec la transformation correspondant à sa classe et enfin le dénormaliser par les moyennes et écarts type de la cible pour cette classe.

3.2.2.6 Transformation de l'enveloppe spectrale utilisant le contexte Aussi bien pour les réseaux de neurones que pour la transformation LMR, nous avons essayé d'influencer la transformation d'un vecteur de coefficients cepstraux par le contexte de ce vecteur. Nous avons appelé contexte d'un vecteur l'ensemble formé par ses 2 plus proches voisins (le vecteur qui le précède et celui qui le suit).

Nous avons construit des bases d'apprentissage formées pour la source de triplets de vecteurs cepstraux, la cible restant la même qu'avant.

Dans le cas où on utilise une seule classe, dans le fichier source les triplets sont assemblés diphone par diphone. Lorsqu'un vecteur manque pour constituer un triplet, on répète le vecteur central. Ainsi, pour un diphone, le premier vecteur cepstral (noté $\mathbf{c_0}$) n'a pas de précédent et de même le dernier vecteur cepstral (noté c_{last}) n'a pas de suivant.

Le premier triplet est	$\mathbf{c_0}$	$\mathbf{c_0}$	$\mathbf{c_1}$
le deuxième triplet est	$\mathbf{c_0}$	$\mathbf{c_1}$	$\mathbf{c_2}$
Le dernier triplet est	$\mathbf{c_{last-1}}$	$\mathbf{c_{last}}$	$\mathbf{c_{last}}$

Cas de la LMR avec contexte (1 seule sous-classe)
La matrice source devient la matrice $\mathbf{C_{S,contexte}}$ de triplets de vecteurs cepstraux normalisés obtenus pour les diphones. Sa dimension est $(3p, N)$. Le cœur de cette matrice est formé de l'ancienne matrice $\mathbf{C_S}$.

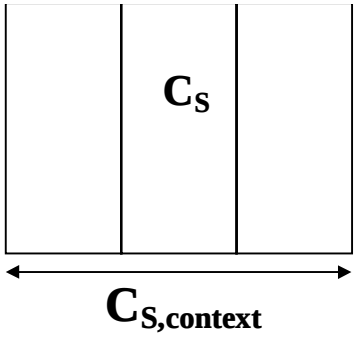


FIG. II.2 – Matrice $\mathbf{C_S}$ avec contexte

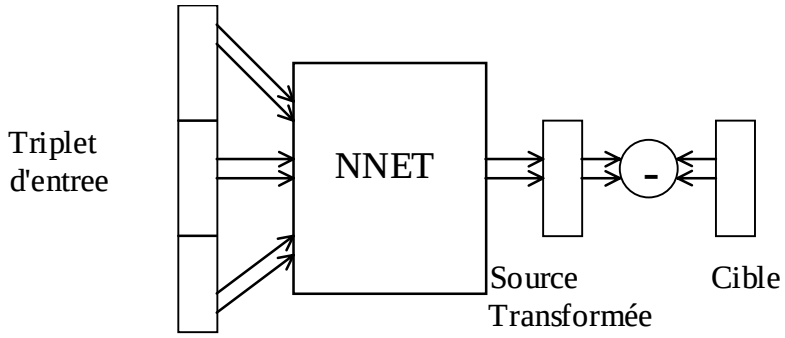


FIG. II.3 – Réseau de neurones utilisant le contexte

Le critère utilisé est le même qu'avant, mais la transformation se fait sur des vecteurs de dimension $(3p, 1)$:

$$\mathbf{c_T} = \mathbf{M_{context}} \mathbf{C_{S,context}}.$$

La matrice $\mathbf{M_{context}}$ est de dimension $(p, 3p)$. La solution optimale pour cette matrice est :

$$\mathbf{M_{context}} = \mathbf{C_C} \mathbf{C_{S,context}}^T \left(\mathbf{C_{S,context}} \mathbf{C^T_{S,context}} \right)^{-1}.$$

Cas d'une transformation par réseau de neurones avec contexte
L'entrée du réseau est constituée dans ce cas de 3 vecteurs de coefficients cepstraux consécutifs (un des triplets). La sortie générée par le réseau représente le vecteur central du triplet transformé. le critère est le même que dans le cas sans contexte.

3.2.2.7 Alignement fréquentiel avec compensation d'amplitude, méthode DFWA Principe de la méthode de transformation spectrale DFWA

Le principe de la transformation spectrale DFWA consiste à déformer l'axe des fréquences et à corriger les amplitudes spectrales. Soit S_S(f) la densité spectrale de puissance de la source, elle est transformée en $\hat{S}(f)$ par :

$$\log \hat{S}(f) = \log \left(S_S(D(f)) \right) + A_m(f)$$

Où $D(f)$ est une fonction de déformation de l'axe des fréquences et $A_m(f)$ une compensation en amplitude.
- Si D(f)=f alors :

$$\log \hat{S}(f) = \log \left(S_S(f) \right) + A_m(f)$$

et la transformation est un simple filtrage du signal temporel par un filtre de fonction de transfert $H(f)$ telle que $|H(f)|^2 = e^{A_m(f)}$.
- Si $A_m(f) = 0 \, \forall f$, alors :

$$\log \hat{S}(f) = \log \left(S_S \left(D \left(f \right) \right) \right).$$

Et il s'agit d'une transformation DFW (Dynamic Frequency Warping).

Justification : Il nous est apparu que la méthode DFW arrivait à aligner les pics des enveloppes spectrales, mais qu'elle ne pouvait pas corriger les différences d'amplitude. On peut interpréter les différences de position des pics d'enveloppes spectrales comme dues à des différences de dimension anatomique des conduits vocaux et les différences d'amplitude de ces pics par des différences d'excitation glottale et de constitution des parois du conduit vocal.

Remarque sur l'environnement d'analyse synthèse
Le travail n'a pas été fait dans le contexte du modèle HNM, aussi la technique d'analyse synthèse utilisée (type OLA[12]) est-elle décrite dans la suite.

Apprentissage de la transformation
L'apprentissage consiste à déterminer les fonctions $D(f)$ et $A_m(f)$.

Dans un premier temps, les deux bases de données sont analysées spectralement et alignées en temps. Une première analyse spectrale est faite qui calcule les coefficients MFCC (FFT puis banc de filtres triangulaires) qui sont ensuite utilisés par la DTW.

Sur les couples source-cible des trames alignées, une seconde analyse spectrale est faite par prédiction linéaire.

Calcul et transformation de la pente spectrale
Sur chacune de ces trames, on calcule par ailleurs, le coefficient g appelé coefficient de pente spectrale (Abe, 1988). Ce coefficient est censé représenter la contribution moyenne de la source glottale à l'enveloppe spectrale. Il est considéré comme un paramètre important pour le timbre. Ce coefficient est transformé à part de manière linéaire.

Le coefficient de pente spectrale g est obtenu par prédiction linéaire AR d'ordre 2 de fonction de transfert :

$$\frac{1}{A(z)} = \frac{1}{\left(1 + gz^{-1}\right)^2}.$$

Les équations de prédiction s'écrivent donc :

$$\hat{x}_n = -2gx_{n-1} - g^2 x_{n-2}.$$

Et par minimisation d'un critère de moindres carrés, g est solution de l'équation de degré 3 :

$$g^3 + 3g^2 \frac{r_1}{r_0} + g\left(2 + \frac{r_1}{r_0}\right) + \frac{r_1}{r_0}.$$

Où r_0 et r_1 sont les coefficients d'autocorrélation d'ordre 0 et 1.

Ce coefficient est lié au coefficient $c(1)$. On montre que : $g \simeq -c(1)$ quand g est petit devant 1.

[12]OLA = OverLap and Add.

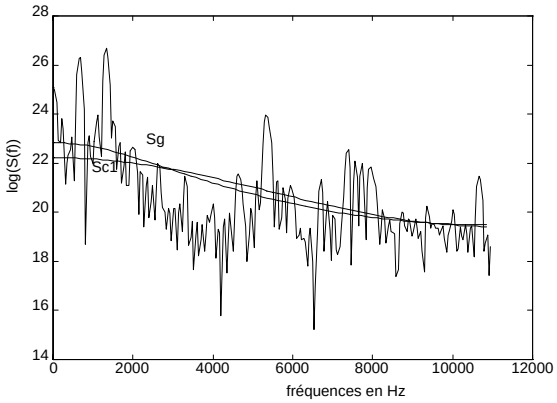


FIG. II.4 – Décroissance globale du spectre (« Spectral tilt ») estimée avec g et c_1

.

La figure II.4 suivante illustre l'approximation de la pente spectrale ou « spectral tilt » obtenue avec g (notée S_g) et avec $c(1)$ (notée S_{c_1}) :

Une fois le coefficient g calculé, on corrige le spectre en fonction de cette pente avant d'apprendre la transformation spectrale, une transformation sur g étant apprise séparément.

Les coefficients g_k des trames successives (d'indices $k \in [1, N]$) sont regroupés en deux vecteurs : g_S pour la source et g_C pour la cible.

Une régression linéaire est effectuée sur ces 2 vecteurs pour calculer les coefficients a et b qui servent à transformer la pente spectrale g_n d'une trame de la source d'indice n, par :

$$\hat{g}_n = a g_n + b$$

Correction de la pente de l'enveloppe spectrale

Pour chaque trame, l'enveloppe spectrale $S(f)$ obtenue par prédiction linéaire est corrigée pour tenir compte de la pente spectrale. Le spectre corrigé est noté $\tilde{S}(f)$:

$$S(f) = \frac{1}{|A(f)|^2}$$

$$\tilde{S(f)} = S(f)|1 + ge^{-2i\pi ft}|^4$$

Les fonctions $D(f)$ et $A_m(f)$ sont ensuite calculées sur ces spectres corrigés.

Rappel du principe de la DFW

Le calcul des fonctions $D(f)$ et $A_m(f)$ se fait de manière très similaire à une DFW (Dynamic Frequency Warping), aussi rappelle-t-on d'abord le principe de la DFW.

Soient 2 trames alignées de densités spectrales de puissance corrigées respectives $\tilde{S}_S(f)$ et $\tilde{S}_C(f)$, où f est discrétisée sur NF points f_k entre 0 et $f_e/2$. La DFW consiste à trouver une fonction $D(f)$, satisfaisant à certaines contraintes, qui minimise le critère J :

$$J = \min \frac{\sum_{i \in Ch} |\log \tilde{S}_C(f_i) - \log \tilde{S}_S(D(f_i))|^2 w(f_i)}{\sum_{i \in Ch} w(f_i)}$$

où Ch est le chemin considéré et $w(f)$ est une fonction de pondération le long de ce chemin. On note $f_j = D(f_i)$.
Le problème revient à trouver le chemin optimal :

$$\{(i,j), \text{ avec } f_j = D(f_i); i \in [0, NF-1], j \in [0, NF-1]\}$$

dans un carré de NF x NF points, parmi tous les chemins possibles vérifiant certaines contraintes.

La recherche du chemin optimal est faite par programmation dynamique.

Calcul des fonctions $D(f)$ et de $A_m(f)$, de transformation de l'axe des fréquences et de correction d'amplitude

Deux approches ont été utilisées pour calculer $D(f)$ et $A_m(f)$, approches que l'on a appelées respectivement approche globale et approche par moyennage.

1. L'approche globale consiste à calculer un seul chemin d'alignement spectral en utilisant globalement les spectres de toutes les trames. C'est à dire que $D(f)$ et $A_m(f)$ sont calculés en même temps sur toutes les trames. Le problème consiste à minimiser la distance cumulée moyenne (pour toutes les trames). Ce critère s'écrit :

$$\min \frac{\sum_{n=1}^{N} \sum_{i \in Ch} |\log \tilde{S}_S(D(f_i);n) - \log \tilde{S}_C(f_i;n) - A_m(f_i;n)|^2 w(f_i)}{\sum_{n=1}^{N} \sum_{i \in Ch} w(f_i)}$$

   La fonction A_m qui minimise le critère globalement doit minimiser le critère en chaque fréquence. Ainsi, $A_{m,opt}$ vérifie :

$$A_{m,opt}(f_i) = \frac{1}{N} \sum_{n=1}^{N} \log \tilde{S}_C(f_i;n) - \log \tilde{S}_S(D(f_i);n)$$

   On intègre cette connaissance de $A_m(f)$ dans la recherche du chemin optimum, c'est à dire dans la recherche de $D(f)$. La recherche de $D(f)$ se fait par programmation dynamique avec le même algorithme que pour la DFW entre deux trames. La seule différence porte sur la définition des distances élémentaires $d(i,j)$. Ici la distance élémentaire $d(i,j)$ est obtenue en prenant en compte $A_m(f_i)$ et en faisant une moyenne sur toutes les trames des distances élémentaires de chaque trame. Ainsi, $d(i,j)$ s'écrit :

$$d(i,j) = \sum_{n=1}^{NF-1} |\log \tilde{S}_S(D(f_i)) - \log \tilde{S}_C(f_i) - A_{m,opt}(f_i)|^2 w(f_i)$$

   On obtient ainsi le chemin optimal correspondant à $D(f)$ et le long de ce chemin optimal la correction d'amplitude $A_m(f)$.

2. La deuxième approche, ou approche par moyennage de chemin, consiste à calculer un chemin optimal pour chaque trame puis à moyenner ces chemins pour obtenir le chemin optimal. Cette seconde approche permet d'éliminer les trames aberrantes. Une fonction $A_m(f)$ est ensuite calculée le long du chemin final.

   Pour chaque trame, les distances élémentaires s'écrivent :

$$d(i,j) = \log \tilde{S}_S(D(f_i)) - \log \left(\tilde{S}_C(f_i)\right).$$

   Deux moyennages différents de chemin ont été essayés. Ils prennent tous deux en compte les histogrammes de correspondances entre les fréquences de la source et de la cible, obtenus sur les

N trames. Ainsi pour la fréquence source f_i, on note $p_i(f_j)$ l'histogramme de correspondance avec les fréquences f_j de la cible. $p_i(f_j)$ est égal au nombre de fois où la fréquence source f_i a été associée avec la fréquence cible f_j dans l'ensemble des chemins optimaux obtenus trame par trame.

Le premier type de moyennage essayé utilise l'histogramme complet et :

$$D(f_i) = \frac{1}{\sum_j p_i(f_j)} \sum_j p_i(f_j) f_j$$

Le deuxième type de moyennage essayé n'utilise que le maximum de l'histogramme et :

$$D(f_i) = \arg(\max(p_i(f_j))$$

On a obtenu de meilleurs résultats avec le premier type de moyennage.

Mise en œuvre de la transformation DFWA
transformation spectrale

Le spectre d'une trame de la source est transformé par les opérations suivantes :

- Calcul de la pente spectrale g,
- Transformation de la pente spectrale :

$$\hat{g} = ag + b$$

- Correction de la pente de l'enveloppe spectrale par :

$$\tilde{S}(f) = \frac{S(f)}{H(f)^2} = S(f)|1 + ge^{-2i\pi ft}|^4$$

- Transformation par $D(f)$ et $A_m(f)$

$$\log \hat{\tilde{S}}(f) = \log \tilde{S}(D(f)) + A_m(f, D(f))$$

Le spectre obtenu a en général moins de points que le spectre de départ. Les valeurs manquantes sont obtenues par interpolation linéaire.

- Ajout de la pente spectrale transformée par :

$$\hat{S}(f) = \frac{\hat{\tilde{S}}(f)}{|1 + \hat{g}e^{-2i\pi ft}|^4}$$

Mise en œuvre sur le signal temporel

La transformation DFWA est appliquée de manière synchrone au pitch à travers une technique de type OLA (OverLap and Add).

On découpe le signal source en trames de longueur égale à deux périodes pitch dans le cas voisé ou au double d'une valeur fixe L dans le cas non voisé, et se superposant d'une période pitch ou de L.

Par ailleurs, on considère que le signal vocal $s(n)$ de transformée de Fourier $S(f)$ sur une trame, vérifie :

$$S(f) = E(f)G(f)C(f)$$

où C(f) représente la contribution du conduit vocal, G(f) la contribution de l'excitation glottale à la pente spectrale et E(f) une excitation blanche.

Pour chaque trame $S(f)$ est calculé par FFT, puis l'enveloppe et la pente spectrale sont calculées par prédiction linéaire comme décrit précédemment. On obtient ainsi $|C(f)|$ et $|G(f)|$. On en déduit un signal d'entrée I(f) par :

$$I(f) = \frac{S(f)}{|C(f)||G(f)|}$$

L'enveloppe spectrale $|C(f)||G(F)|$ est modifiée comme expliqué précédemment. Puis la FFT de la trame transformée est obtenue par :

$$\hat{S}(f) = I(f)|\hat{G(f)}||\hat{C(f)}|$$

Par FFT inverse on en déduit un signal temporel $\hat{s}(n) = \mathit{IFFT}(s(n))$, pour chaque trame. Puis on reconstruit le signal temporel transformé en ajoutant la trame transformée $\hat{s}(n)$ multipliée par une fenêtre de Hanning avec un recouvrement entre trames successives égal à une période pitch ou à une valeur constante L.

Les trames sont superposées en faisant la somme des échantillons qui se recouvrent (OLA).

Résultats de la DFWA

Les résultats de cette méthode sont présentés séparément des résultats des autres méthodes car la méthode DFWA n'a pas été testée sur les bases de données du CNET. En effet, les 1^{ers} résultats obtenus étaient inférieurs à ceux d'une simple transformation linéaire LMR.

Données expérimentales

Pour un premier test des algorithmes, nous avons constitué un corpus très réduit formé des sons [a], [ou], [i] prononcés plusieurs fois de façon soutenue par 2 locuteurs, un masculin et un féminin.

Les signaux ont été échantillonnés à 8000 Hz sur 16 bits.
Pour la DTW, on a utilisé des trames de 20 ms avec un chevauchement de 10 ms, 16 filtres entre 0 et 4000 Hz et 10 coefficients MFCC (Mel Frequency Cepstrum Coefficient).
L'analyse spectrale a utilisé une prédiction linéaire d'ordre 10.
Les spectres sur lesquels ont été calculés les fonctions D et A_m comportaient 512 points de fréquence entre 0 et 4000 Hz.
On a utilisé les valeurs extrêmes de pitch : 400 Hz et 80 Hz.

Comparaison entre l'approche globale et l'approche par moyennage de chemins

Approche globale

La recherche du chemin d'alignement se fait à partir d'un tableau des distances élémentaires obtenu comme somme des distances élémentaires de chaque trame. Le fait de calculer le chemin une fois seulement diminue le temps de calcul. L'alignement en fréquence obtenu est assez bon (on peut remarquer que les formants sont bien déplacés), mais la transformation n'est pas toujours satisfaisante en amplitude.

Approche par moyennage de chemins

Le chemin d'alignement est la moyenne des chemins trouvés pour chaque trame. Les résultats semblent parfois meilleurs que ceux obtenus avec l'approche globale, le déplacement des formants est plus précis, la correction d'amplitude reste peu satisfaisante. L'algorithme est plus lent, car on calcule le chemin d'alignement un nombre de fois égal au nombre de trames.

Les 2 approches donnent la même qualité de signal reconstruit sur la base de test. Sur les voyelles, cette qualité est plutôt bonne et la voix obtenue est une voix "moyenne" entre source et cible.

Distances spectrales

On a calculé les distances quadratiques sur les logarithmes des spectres de la source et de la cible le long de la diagonale (pas d'alignement en fréquence) et le long du chemin optimal, ceci avec et sans compensation d'amplitude. Les résultats sont les suivants :
On donne ici les résultats obtenus avec l'approche globale pour les sons a et i.

Pour le son [a]

DFW sans correction d'amplitude	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,337$
DFW avec correction d'amplitude (DFWA)	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,274$
Correction d'amplitude seule	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,376$

Pour le son [i]

DFW Sans correction d'amplitude	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,462$
DFW avec correction d'amplitude (DFWA)	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,3119$
Correction d'amplitude seule	$\frac{\text{distance}_{chemin\_optimal}}{\text{distance}_{diagonale}} = 0,500$

Quelques tracés spectraux

La figure II.5 donne quelques tracés de spectres obtenus pour le son [a] avec les 2 méthodes.

3.2.2.8 Transformation statistique par modélisation avec un mélange de gaussiennes, méthode GMM

Cette méthode a été développée à l'ENST par Y. Styliannou et E. Moulines.

Dans la phase d'apprentissage, on ajuste un modèle de mélange de gaussiennes sur les vecteurs de la source $\mathbf{c_S}$. La densité de probabilité des vecteurs cepstraux s'écrit alors :

$$p\left(\mathbf{c_S}\right) = \sum_{i=1}^{m} \alpha_i \mathcal{N}\left(\mathbf{c_S}; \mu_\mathbf{i}, \mathbf{\Sigma_i}\right).$$

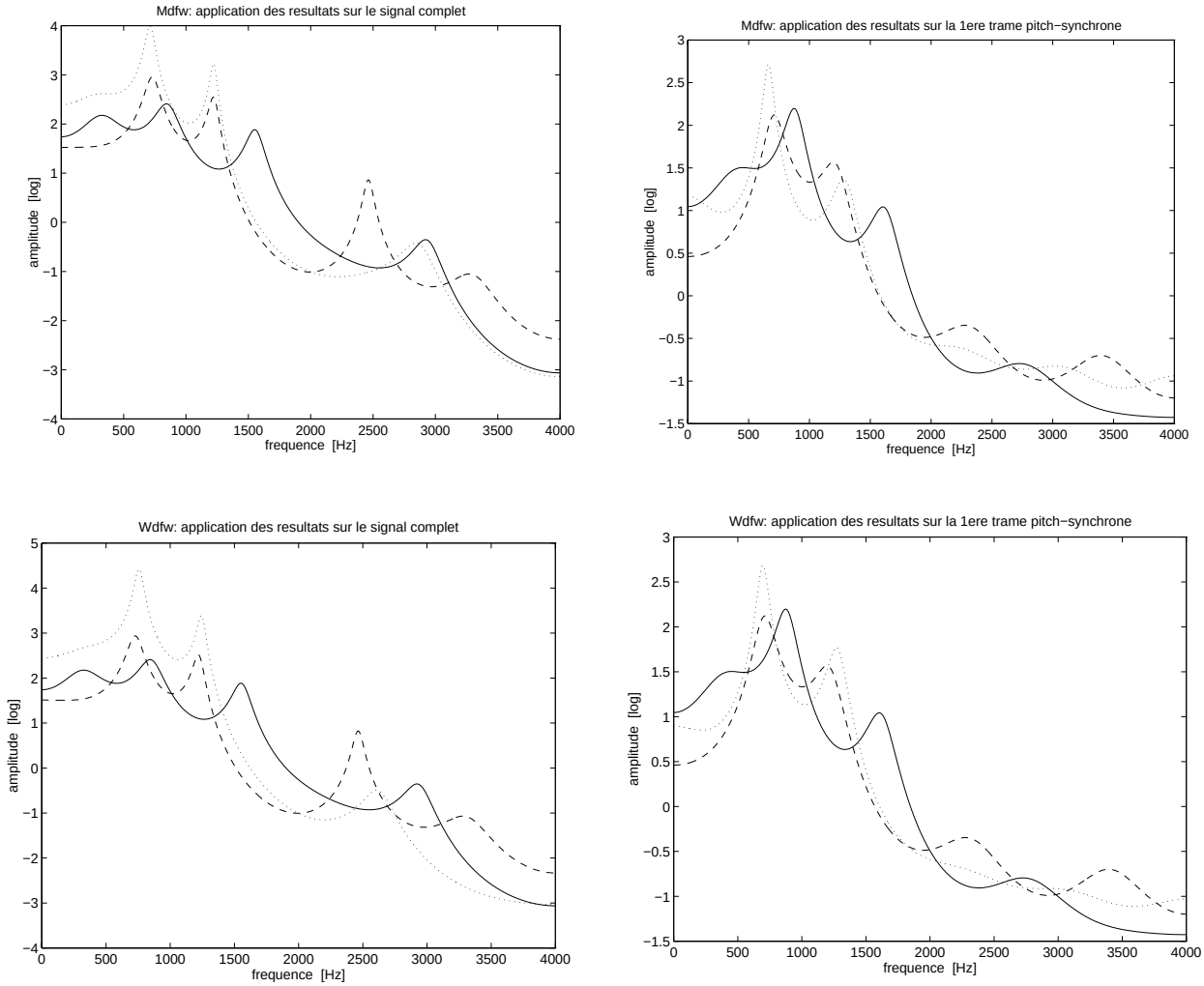
Où $\mathcal{N}\left(\mathbf{c}; \mu, \Sigma\right)$ représente la loi gaussienne de dimension$p$ de moyenne μ et de matrice de covariance $\mathbf{\Sigma}$, et où les α_i sont des scalaires positifs normalisés.

Le modèle de mélange de gaussiennes est une technique très utilisée en vérification du locuteur. Il est capable de représenter l'espace acoustique d'un locuteur comme une combinaison linéaire de plusieurs composantes Ω_i ($i = 1, \cdots, m$) où m est le nombre de composantes du mélange.

La densité de probabilité conditionnelle $p\left(\Omega_i|\mathbf{c_S}\right)$ qu'une observation $\mathbf{c_S}$ appartienne à l'une des classes Ω_i est donnée par :

$$p\left(\Omega_i|\mathbf{c_S}\right) = \frac{\alpha_i \mathcal{N}\left(\mathbf{c_S}; \mu_\mathbf{i}, \mathbf{\Sigma_i}\right)}{\sum_{j=1}^{m} \alpha_j \mathcal{N}\left(\mathbf{c_S}; \mu_\mathbf{j}, \mathbf{\Sigma_j}\right)}.$$

Les paramètres du mélange de gaussiennes sont estimés par l'algorithme EM (Expectation-Minimisation) [58].



Ligne continue = Source
Ligne en tirets = Cible
Ligne pointillée = Source Transformée

Mdfw = Méthode globale,
Wdfw = Méthode par moyennage de chemins

FIG. II.5 – Spectres orignaux et transformés

La conversion de voix est effectuée par la transformation suivante :

$$\mathbf{C_C} = F\left(\mathbf{c_S}\right) = \sum_{i=1}^{m} p\left(\Omega_i|\mathbf{c_S}\right)\left[\nu_\mathbf{i} + \Gamma_i\Sigma_i^{-1}\left(\mathbf{c_S} - \mu_\mathbf{i}\right)\right].$$

La fonction de conversion est complètement définie par les vecteurs $\nu_\mathbf{i}$ de dimension p et les matrices $\mathbf{\Gamma_i}$, de dimension (p, p), pour $i \in [1, m]$.

Les paramètres de la fonction de conversion sont obtenus par la minimisation d'un critère de moindres carrés E sur la base d'apprentissage mesuré entre les N vecteurs cibles $\mathbf{c_{C,k}}$) et les N vecteurs transformés $F(\mathbf{c_{S,k}})$.

$$E = \sum_{k=1}^{N} d\left(\mathbf{c_{C,k}}, F(\mathbf{c_{S,k}})\right).$$

Trois types de conversion ont été comparées : la conversion avec des matrices $\mathbf{\Sigma_i}$ et $\mathbf{\Gamma_i}$ complètes, avec des matrices $\mathbf{\Sigma_i}$ et $\mathbf{\Gamma_i}$ diagonales, et un $3^{\text{ème}}$ type appelé *VQM-type* dans lequel ces matrices $\mathbf{\Sigma_i}$ et $\mathbf{\Gamma_i}$ sont omises. Ce dernier type se rapproche de la méthode de mise en correspondance de dictionnaires de quantification vectorielle, mais elle travaille d'une manière moins brutale.

3.2.2.9 Transformation de l'enveloppe spectrale par simple normalisation dénormalisation Afin de valider l'utilisation de méthodes compliquées, nous les avons comparé à une méthode de transformation très simple, consistant à normaliser les vecteurs de la cible avec les moyennes et les écarts type de la source, puis à les dénormaliser avec les moyennes et les écarts type de la cible.

Cette transformation utilisée sur la base de diphones sur des vecteurs n'ayant pas servi à l'apprentissage des méthodes dites élaborées, donne de nettement moins bons résultats que les transformations plus élaborées.

3.2.2.10 Expérience de transformation par simple substitution Afin d'avoir une idée du meilleur résultat possible avec les différentes méthodes développées, nous avons réalisé une expérience de substitution spectrale, consistant à substituer aux vecteurs cepstraux de la source, les vecteurs de la cible.

Les différentes étapes de cette substitution sont les suivantes :

On travaille sur deux exemples de signaux, l'un pour la source, l'autre pour la cible, correspondant au même logatome ou à la même phrase, selon la base de données utilisée.

Les fichiers de coefficients cepstraux sont déduits des résultats d'une analyse HNM asynchrone.

Ils sont alignés temporellement. On obtient un tableau de correspondances entre les instants d'analyse de la source et ceux de la cible.

Puis on transforme le fichier source de coefficients cepstraux par substitution avec la cible. Deux cas peuvent se présenter, soit le vecteur source a été aligné avec un vecteur de la cible, il est alors remplacé par ce vecteur de la cible, soit il n'a pas été aligné avec un vecteur de la cible et dans ce cas on le remplace par une combinaison linéaire des vecteurs cepstraux adjacents de la cible.

En ce qui concerne le coefficient $c(0)$, nous avons testé deux solutions : le conserver, ou le remplacer par celui de la cible.

On recalcule ensuite les nouveaux paramètres de synthèse HNM, à partir des anciens paramètres HNM et des nouveaux coefficients cepstraux. Enfin on synthétise par HNM, le signal transformé.

Nous avons, par ailleurs, effectué des tests avec et sans modification de la prosodie.

Les résultats obtenus sont présentés, pour chaque base de données, dans les sections 3.2.3 et 3.2.4

3.2.3 Résultats obtenus sur la base de données de diphones OB, RG

L'analyse est faite sur des trames de 20 ms se recouvrant de 10 ms. Les vecteurs cepstraux sont de longueur 16 et leur nombre est environ 35 000.

Rappel sur la distance utilisée :
$\mathbf{C_1}$ et $\mathbf{C_2}$ étant 2 matrices de dimension (p, N) formées de N colonnes de vecteurs spectraux à p composantes.

$$D(\mathbf{C_1}, \mathbf{C_2}) = \frac{1}{pN} \sum_{i=1}^{N} \sum_{j=1}^{p} \left(\mathbf{C_1}(i,j) - \mathbf{C_2}(i,j)\right)^2 .$$

La base de test a été obtenue en prenant un vecteur sur 5 dans les fichiers complets.

3.2.3.1 Tableau de résultats et de comparaison des différentes méthodes Le tableau II.1 effectue une comparaison des résultats obtenus avec les différentes méthodes.

Toutes les distances données dans le tableau sont normalisées par rapport à la distance quadratique entre la source et la cible alignées $d(OB, RG)$, classe voisée ou non-voisée. Les distances normalisées suivantes sont données pour les différentes configurations :

D_{tc} V = distance normalisée entre la source transformée et la cible, pour la classe voisée,
D_{ts} V = distance entre la source transformée et la source originale, pour la classe voisée,
D_{tc} NV = distance normalisée entre la source transformée et la cible, pour la classe non-voisée,
D_{ts} NV = distance entre la source transformée et la source originale, pour la classe non-voisée,
On indique dans le tableau les résultats obtenus sur la base de test.

On notera pour compléter le tableau, les valeurs des distances quadratiques absolues :

Base de test	d(OB,RG)	=	0,0347
Base de test, cas voisé :	d(OB,RG)	=	0,0334
Base de test, cas non-voisé :	d(OB,RG)	=	0,0269

La distorsion de quantification vectorielle mesurée avec la même distance, avec 2 dictionnaires, un dictionnaire de taille 512 pour les vecteurs voisés et un dictionnaire de taille 128 pour les vecteurs non voisés, vaut pour le locuteur OB : 0.0037.

Par souci de comparaison, les distances utilisées d_{tc}, d_{ts} pour la méthode VQM sont calculées par rapport aux vecteurs non quantifiés.

Pour les 2 méthodes utilisées en sous-classes : NNETS ou LMR, on constate que les performances s'améliorent jusqu'à 16 classes, mais que les résultats obtenus avec 16 ou 64 classes sont quasiment les mêmes. Ceci est sans doute dû à la taille limitée de la base de données.

On peut remarquer que les réseaux de neurones non linéaires ne donnent pas de meilleurs résultats que la LMR lorsque ces transformations sont effectuées en sous-classes.

	D_{tc} V	D_{ts} V	D_{tc} NV	D_{ts} NV
VQM WeightedMapping	0.30	0.77	0.20	0.84
VQM MaxMapping	0.41	0.88	0.28	0.92
GMM complète (64 composantes)	0.28	0.75	0.20	0.83
GMM matrices diagonales (128 composantes)	0.30	0.70	0,20	0,79
GMM *VQM-type* (256 composantes)	0,30	0,77	0,20	0,82
NNETS (1 classe)	0,35	0,65	0,23	0,85
NNETS (64 classes)	0,32	0,79		
LMR (1 classe)	0,36	0,64	0,22	0,77
LMR (64 classe)	0,31	0,74	0,18	0,62
LMR (1 classe avec contexte)	0,34	0,66		

TAB. II.1 – Comparaison des différentes méthodes. Les distances sont des distances normalisées par rapport à la distance originale entre la source et la cible, en distinguant la classe voisée et la classe non-voisée.

Résultats subjectifs Les tests subjectifs effectués sur les phrase pour évaluer la qualité de la parole transformée, ont donné le résultat suivant :
GMM > NNETS $\simeq$ LMR > VQM.

Mais la qualité de parole obtenue est insuffisante pour un système de synthèse à partir du texte. Le principal défaut est le manque de clarté dû au lissage entre sons introduit par les différentes méthodes.

Ce défaut pourrait être atténué en utilisant un filtre de renforcement des formants après la transformation spectrale.

3.2.3.2 Test de transformation effectuée par simple substitution sur un logatome On a travaillé sur deux logatomes, en effectuant les opérations décrites au paragraphe 3.2.2.10 :

- Analyse HNM asynchrone d'un logatome de la source et de la cible,
- Alignement temporel des fichiers de coefficients cepstraux des 2 logatomes par DTW,
- Substitution des vecteurs de coefficients cepstraux de la source par ceux de la cible avec lesquels ils sont alignés.
- calcul des paramètres HNM à partir des nouveaux paramètres,
- Synthèse HNM du logatome source transformé.

Les tests d'audition ont montré que :

Sans modification de la prosodie :
La qualité de la parole synthétique est bonne.

Le timbre est transformé significativement (au moins pour la base de logatomes) et se rapproche de la cible, ce qui est encourageant, mais la fréquence fondamentale est très importante pour l'identification du locuteur. Le meilleurs résultats sont obtenus lorsqu'on utilise le $c(0)$ de la cible.

Avec modification de la prosodie :
La qualité de la voix est dégradée. La transformation de la prosodie a été faite en même temps que celle de l'enveloppe spectrale. Peut-être aurions nous dû refaire une analyse synchrone avant de modifier la prosodie,

La qualité de la transformation est plutôt bonne, ce qui est surtout dû à la transformation de la fréquence fondamentale.

3.2.3.3 Quelques conclusions générales sur les résultats obtenus sur la base de données de diphones Quelque soit la méthode utilisée, le travail en sous classes améliore les résultats quantitatifs de distances spectrales.

Les résultats obtenus avec une transformation non linéaire par réseau de neurones sont légèrement supérieurs à ceux obtenus avec une transformation linéaire LMR. Les résultats quantitatifs obtenus par transformation de dictionnaire de quantification vectorielle sont à peu près équivalents à ceux obtenus avec les réseaux de neurones, mais la qualité subjective de la parole synthétique est inférieure, sauf si on utilise un dictionnaire dit naturel (les distances se dégradant légèrement dans ce cas). Les meilleurs résultats sont obtenus avec la méthode GMM.

Les spectrogrammes de parole transformée sont beaucoup moins contrastés que ceux de parole originale.

l'évaluation subjective est difficile. Une évaluation sur des phrases serait intéressante, mais les phrases dont nous disposions ne correspondent pas aux mêmes conditions d'enregistrements que les logatomes qui ont servi à l'apprentissage.

Les périodes fondamentales des deux locuteurs sont très différentes et il est difficile d'évaluer la transformation spectrale seule, la transformation de la prosodie est essentielle.

Une amélioration sur les distances spectrales ne se traduit pas forcément par une amélioration audible de la transformation spectrale.

Toutefois l'expérience de simple substitution spectrale a montré que le timbre de la voix peut être changé de façon audible sans trop de dégradation de la qualité. Cette expérience nous donne une limite de ce qu'on peut espérer obtenir avec les différentes méthodes que nous avons essayées.

3.2.4 Résultats sur la base de données de phrases OB et RG

La $2^{\text{ème}}$ base de données est constituée de 89 phrases phonétiquement équilibrées pour le locuteur OB et un locuteur RG qui n'est pas le même que dans l'ancienne base.

3.2.4.1 Segmentation manuelle de la base La nouvelle base de données ne comportait que les fichiers de signal (extension .sig). Pour améliorer les performances de l'alignement temporel par DTW, nous avons segmenté manuellement cette nouvelle base en zones assez courtes pour la DTW. Plus précisément, à l'aide d'un éditeur de signaux et de tests d'écoute, nous avons segmenté les phrases en segments syllabiques contenant de 2 à 5 syllabes.

3.2.4.2 Résultats des tests

Remarque générale :

On a remarqué que la distance moyenne entre OB et RG, calculée sur les vecteurs cepstraux, est très inférieure à celle calculée sur la base de logatomes.

d(OB,RG) = 0,033 avec la base de logatomes,
d(OB,RG) = 0,018 avec la base de phrases et le nouveau locuteur RG.

Il n'est pas possible de savoir si cette différence provient du nouveau locuteur RG, ou si elle provient du fait que les logatomes ont été remplacés par des phrases.

Par la suite, on notera toujours $\hat{OB}$, pour indiquer OB transformé par une des méthodes, De même que V représentera le cas voisé et NV le cas non voisé.

On a de plus constaté, avec la nouvelle base, que quelque soit la méthode utilisée, le rapport des distances source-cible avant et après transformation est toujours supérieur à 0,5.

$$\frac{d(\hat{OB}, RG)}{d(OB, RG)} > 0,5$$

Ce rapport varie de 54% à 87% selon la méthode, alors qu'il variait de 28% à 42% avec la base de diphones.

Les résultats obtenus sont donnés dans les tableaux suivants, où chaque case représente la distance normalisée entre les 2 éléments indiqués en bout de ligne et de colonne. Par distance normalisée, on entend distance exprimée en pourcentage de la distance source_cible originale.

Pour la méthode VQM, nous avons classé les trames en voisées et non-voisées. Pour la classe voisée nous avons utilisé un dictionnaire de 512 vecteurs et pour la classe non-voisée un dictionnaire de 256 vecteurs.

cas Voisé

distances normalisées	RG_{test}	OB_{test}
OB_{test}	(0,0179) 100%	
$\hat{OB}$,GMM full 64 composantes	54,6%	45,3%
$\hat{OB}$,GMM diagonal 128 composantes	59,9%	40,1%
$\hat{OB}$, VQM weighted mapping	70,3%	44,4%
$\hat{OB}$, VQM max mapping	87,2%	57,1%
$\hat{OB}$, NNETS 1 classe	68,2%	33,9%
$\hat{OB}$, LMR 1 classe	68,4%	31,5%

cas Non Voisé

distances normalisées	RG_{test}	OB_{test}
OB_{test}	(0,0159) 100%	
$\hat{OB}$, VQM weighted mapping	65,1%	42,1%
$\hat{OB}$, VQM max mapping	92,6%	71,1%

Commentaires des résultats quantitatifs

les distances spectrales entre OB transformé et RG représentent toujours plus de 55% de la distance originale entre OB et RG, ceci quelque soit la méthode.

Les deux locuteurs ont des enveloppes spectrales proches à l'origine. Il est donc plus difficile percevoir les transformations de timbre que dans le cas des deux locuteurs précédents (base de logatomes) qui avaient des spectres plus différents au départ.

La hiérarchie des performances des méthodes est à peu près conservée, même si certaines méthodes qui donnaient des résultats quasiment identiques, se différencient avec la nouvelle base (GMM

diagonal 128 et VQM par exemple).

Transformation d'une phrase et évaluation subjective de la qualité de la phrase transformée Une des 89 phrases a été transformée par les différentes méthodes.

La qualité du signal synthétique transformé est bien meilleure que celle obtenue avec la base de logatomes mais les effets des transformations ne sont pas convaincants, comme le laissaient prévoir les résultats sur les distances.

En ce qui concerne l'efficacité subjective des transformations, il est difficile de classer les méthodes.

Test de substitution effectué sur une phrase
Pour la même phrase, nous avons effectué un test de substitution spectrale des vecteurs de OB avec ceux de RG, de la manière décrite en 3.2.2.10.

Les différentes méthodes de transformation qui sont construites par apprentissage sur des fichiers de vecteurs cepstraux alignés ne peuvent sans doute pas faire mieux que cette substitution. Or en écoutant le signal synthétique obtenu par cette substitution, la substitution est peu audible et on ne reconnaît pas la cible. Ceci semble montrer que pour ces deux locuteurs, le timbre ne joue pas un rôle prépondérant dans l'identification de ces 2 locuteurs, la prosodie étant pour eux plus importante.

3.2.4.3 Conclusions sur les résultats obtenus avec la base de données de phrases La méthode de transformation GMM matrice pleine (64 classes) a donné les meilleurs résultats quantitatifs sur les deux bases de données. Mais c'est aussi la plus lourde en calcul.

les méthodes NNETS et VQM (weighted) sont à peu près équivalentes en complexité et en performances. La méthode LMR, qui est de loin la moins complexe, donne des résultats comparables aux deux précédentes si on l'utilise en sous classes avec 64 classes.

Avec la base de phrases, les méthodes LMR, NNETS, VQM(weighted) sont équivalentes.

D'un point de vue subjectif, pour la base de logatomes, la qualité du signal synthétique est insuffisante et de ce fait, il est très difficile de juger de l'efficacité des transformations. Pour la base de phrases, la qualité du signal synthétique après transformation est tout à fait acceptable, quelque soit la méthode, mais les transformations spectrales sont très peu audibles, car les enveloppes spectrales originales des 2 locuteurs sont proches.

3.2.5 Conclusion générale pour les 2 bases de données

Les résultats obtenus dépendent fortement de la base de données utilisée.

Pour la première base, les transformations spectrales testées, ont conduit à une transformation quantitativement efficace (évaluation par les distances spectrales moyennes entre locuteurs source, cible et transformé). Mais la qualité subjective des signaux transformés est insuffisante, les distorsions étant plus ou moins importantes selon les diphones.

Pour la deuxième base de données, l'efficacité des transformations spectrales est beaucoup moins grande et de ce fait les signaux transformés restent de bonne qualité subjective. Dans ce cas, la distance spectrale moyenne originale entre les 2 locuteurs était très inférieure à celle existant entre les deux locuteurs de la première base. Sans doute, pour cette deuxième base, la prosodie jouait-elle un rôle plus important que le timbre dans la distinction entre les deux locuteurs. Les différences de résultats s'expliquent aussi par la nature différentes des 2 bases (logatomes vs phrases).

Conclusions et perspectives

Au cours des dernières années, j'ai essentiellement travaillé en traitement de la parole. Mais ce domaine ne correspond pas actuellement, au moins en France, à un secteur d'embauche très important. Compte tenu de la mission première de l'ESIEE, à savoir la formation d'ingénieurs et de technologues, il m'a semblé nécessaire d'infléchir mes activités de recherche vers un domaine plus porteur d'un point de vue industriel et en meilleure adéquation avec les activités de R&D sur le site de Marne La Vallée.

J'ai choisi le domaine des radiocommunications, et plus particulièrement la conception de techniques de prédistorsion adaptative en bande de base pour les émetteurs des mobiles de radiocommunications et pour les réseaux locaux radio. Une première publication sur ce sujet a été acceptée à la conférence CSCC'2000 [19]. Ce travail s'effectue avec un étudiant en thèse et en collaboration directe avec ma collègue P. Jardin. Il se fait dans le cadre du pôle Électronique Hautes Fréquences de Marne La Vallée pour lequel le thème « Intégration d'amplificateurs de puissance et de circuits de linéarisation pour émetteurs multistandards-multimodulations » constitue un projet fédérateur regroupant les activités de plus d'une dizaine de chercheurs. Ce projet comporte de nombreux aspects tels que :

- Aspects technologiques : modélisation de composants de puissance en technologie SiGe,
- Aspects électroniques : conception d'amplificateurs de puissance et de circuits de linéarisation analogiques, conception de convertisseurs analogiques numériques rapides,
- Aspects architecture : étude de la séparation et du traitement séparés de l'amplitude et de la phase des signaux,
- Aspects traitement du signal : prédistorsion adaptative en bande de base, étude des caractéristiques nécessaires pour les convertisseurs analogiques numériques et numériques analogiques,
- Aspects système : étude de l'influence des non linéarités sur les performances des systèmes selon les modulations numériques et les types d'accès multiples utilisés, étude du compromis linéarité-rendement des amplificateurs.

Je continue toutefois à m'intéresser au codage de parole à très bas débit utilisant les techniques ALISP. Je participe, sur ce thème, au projet RNRT SYMPATEX. Je cherche en particulier à améliorer la méthode de détermination des unités de synthèse (US) et la qualité de la méthode de synthèse du décodeur. Il a été décidé de représenter les unités de synthèse par un modèle HNM pour faciliter les modifications prosodiques.

Pour le moment, la détermination des US (voir chapitre 1, les sections 4.2.2 et 4.2.3) est faite d'une manière très rudimentaire. Pour chaque classe acoustique ou unité ALISP (UA), il y a 8 US qui sont les 8 segments les plus longs de la base d'apprentissage étiquetés par cette UA.

Le codeur après avoir reconnu une unité ALISP, choisit le meilleur représentant parmi les 8 possibles, en appliquant une technique de comparaison dynamique (DTW) sur les séquences cepstrales des représentant et du segment de parole à coder. Le numéro du représentant ou US choisi est transmis au décodeur. Cette approche est critiquable à plusieurs points de vue :

- Le choix les 8 segments les plus longs comme US n'a pas de justification théorique précise,
- Le choix du meilleur représentant parmi les 8, par comparaison des séquences cepstrales des représentants et du segment à coder ne prend pas en compte la capacité du segment retenu à se concaténer avec ses voisins.

Je développe actuellement une nouvelle méthode qui cherche à apporter une meilleure solution aux 2 points précédents :

- Les US sont plus nombreuses (ce qui augmente le débit). Différentes techniques sont étudiées pour leur détermination. Je développe une approche qui consiste à effectuer une classification des segments de la base de données pour une unité ALISP, et à choisir un représentant dans chaque classe. Différentes classifications sont envisageables, s'appuyant sur le contenu cepstral ou la prosodie des segments.
- Le choix de l'unité de synthèse pour un segment de parole à coder, utilisera un critère prenant en compte la ressemblance cepstral du segment avec l'US, mais aussi la proximité de la fréquence fondamentale et la bonne concaténation avec les segments voisins.

Je m'intéresserai ensuite à la généralisation du codeur au cas multilocuteur.

Enfin, je continue à développer une compétence sur l'utilisation des DSP pour diverses applications de traitement de signal. Les applications que j'envisage à court terme sont :

- Les systèmes de prédistorsion adaptative en bande de base pour les émetteurs de radiocommunications, où des DSP rapides et faible consommation seront nécessaires.
- Les capteurs intelligents. Je vais, en particulier, continuer le travail sur le gyromètre vibrant à excitation magnétique, en coopération avec mon collègue O. Venard et en lien avec la société ISNAV et la DGA. L'objectif est d'améliorer les performances du capteur existant en précision et étendue de la plage de mesure. Pour y réussir une modélisation fine des sources des défauts est nécessaire. Une première publication [27] sur ce thème a été aceptée à la conférence Texas-Instruments « 3^{rd} European Conference on DSP Research and Education », qui aura lieu à l'ESIEE en septembre 2000.

Je continuerai par ailleurs à enseigner le sujet, aussi bien en formation d'ingénieurs que de technologues. L'utilisation de DSP permet en effet aux étudiants d'effectuer une synthèse de plusieurs disciplines, telles que le traitement du signal, le temps réel, la programmation, l'analyse numérique.

Enfin, un dernier projet à court terme est la coordination d'un ouvrage intitulé : « Radiocommunications numériques, modélisation et simulation », qui sera publié chez Dunod. La rédaction de ce livre sera effectuée par l'ensemble des 9 enseignants du laboratoire signaux et télécommunications de l'ESIEE. Elle s'appuyera sur les polycopiés de cours et sur les études de cas que nous avons développées au cours des dernières années. Ces études de cas sont aujourd'hui mises en œuvre sur le logiciel de CAO HP-ADS, permettant la cosimulation analogique et numérique de circuits et systèmes de radiocommunications numériques.

On peut espérer que ce travail de synthèse effectué en commun nous aidera à dégager des orientations pertinentes pour nos futurs travaux de R&D.

Références générales

[1] Le test de diagnostic par paires minimales, adaptation au français du diagnostic rythm test de w.d. voiers. *Revue d'acoustique*, (27), 1973.

[2] M. Abe. Voice conversion through vector quantization. In *Proc. ICASSP 88*, pages 655–658, April 1988.

[3] M. Abe. A segment-based approach to voice conversion. In *Proc. ICASSP 91*, pages 765–768, 1991.

[4] M. Abe, K. Shikano, and H. Kuwabara. Cross-language voice conversion. In *Proc. ICASSP 90*, pages 345–348, Albuquerque, 1990.

[5] J. P. Adoul, P. Mabilleau, and S. Morissette. Fast celp coding based on algebraic codes. In *Proc. IEEE ICASSP 87*, pages 1954–1956, 1987.

[6] B.S. Atal. Efficient coding of LPC parameters by temporal decomposition. In *Proc. IEEE ICASSP 83*, pages 81–84, 1983.

[7] B.S. Atal and S. L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *J. Acoust. Soc. Amer.*, 50(2) :637–657, 1971.

[8] G. Baudoin. Convergence d'algorithmes de type lms pour l'annulation des échos sur les images de télévision. In *Actes du* $14^{ème}$ *colloque GRETSI*, pages 521–524, Juan les pins, France, 1991.

[9] G. Baudoin. Design, simulation and applications of multirate filter banks,proc. final meeting tempus jep 1326 project. Technical report, ESIEE and university of Varsovy., 1994.

[10] G. Baudoin. Speech coding at low and very low bit rates. In *Proc. ERK'99 conference*, pages 11–14, Portoroz, Slovenia, Sept. 1999.

[11] G. Baudoin and P. Blaha. Development of a low bit rate speech coder on a tms320c30 based on the half rate gsm standard. In *Proc. of* 1^{rst} *European Conference on DSP Research and education*, pages 11–22, Noisy Le Grand FRANCE, 1996.

[12] G. Baudoin and M. Chaouche. Holter numérique, un système portable pour l'enregistrement de l'électrocardiogramme. In *Actes du colloque GRETSI*, pages 635–637, Nice, France, 1987.

[13] G. Baudoin and M. Chaouche. A portable digital system for recording and processing of ecg. In *Proc. Eusipco'88*, pages 1275–1278, Grenoble, France, 1988.

[14] G. Baudoin and M. Chaouche. A portable system for digital recording of electrocardiogram. In *Proc. IEEE conf. on bioeng. and med. Physic*, page 188, San Antonio, 1988.

[15] G. Baudoin, P. Jardin, G. chollet, and J. Gross. Comparaison de techniques de paramètrisation spectrale pour la reconnaissance vocale en milieu bruité. In *Actes du quinzième colloque GRETSI*, pages 783–786, Juan les pins, France, 1993.

[16] G. Baudoin, P. Jardin, J. Gross, and G. Chollet. *Speech Recognition and Coding, new advances and trends*, chapter Comparison of parametric spectral representations for voice recognition in noisy environments, pages 313–316. Springer-Verlag, nato asi serie f., edited by a. rubio & jm lopez edition, 1995.

[17] G. Baudoin and M. Jelinek. Technical report for research contract acsys, codeur de parole celp à bas débit. Technical report, ESIEE, Noisy Le Grand, 1993.

[18] G. Baudoin, R. Marsalek, and J. Prokes. Evaluation of the potential of the vliw digital signal processor tms320c6201 for umts fdd standard baseband processing implementation. In *Proceedings ERK'99*, pages 113–116, Portoroz, Slovenia, 1999.

[19] G. Baudoin and P.Jardin. A new adaptive baseband pre-distortion algorithm for linearization of power amplifiers, application to edge-gsm transmitters. In *To appear in* Proceedings of CSCC, Int. Conf. on Circuits Systems and Communications, Greece, July 2000.

[20] G. Baudoin, C. Ripoll, and P. Bildstein. Rapport d'étude (pct anvar) pour la société stid, sur les systèmes d'identification sans contact à 13,56 mhz. Technical report, ESIEE, Noisy Le Grand, 1999.

[21] G. Baudoin and Y. Styliannou. On the transformation of the speech spectrum for voice conversion. In *Proc. of ICSLP'96*, pages 1404–1408, Philadelpia, USA, October 1996.

[22] G. Baudoin, J. Černocký, and G. Chollet. Quantification de séquences spectrales de longueurs variables pour le codage de parole à très bas débit. In *Proc. GRETSI'97*, Grenoble, France, September 1997.

[23] G. Baudoin, J. Černocký, and G. Chollet. Quantization of spectral sequences using variable length spectral segments for speech coding at very low bit. In *Proc. of Eurospeech 97*, pages 1295–1298, Rhodos, Grece, September 1997.

[24] G. Baudoin, J. Černocký, P. Gournay, and G. Chollet. Codage de la parole à bas et très bas débit. *To appear in Annales des télécommunications.*, 2000.

[25] G. Baudoin and O. Venard. Implementation of fir filters on fixed point dsp for communication systems. In *Proc. of* 2^{nd} *European Conference on DSP Research and education*, pages 365–371, ESIEE, Noisy Le Grand, FRANCE, 1998.

[26] G. Baudoin and O. Venard. Research report for project isnav, génération et traitement des signaux d'un gyromètre magnétique vibrant. Technical report, ESIEE, Noisy Le Grand, 1999.

[27] G. Baudoin and O. Venard. Digital signal processing for a vibrating magnetic excitation gyrometer, implementation on a dsp tms320f243. In *Proc. of* 3^{nd} *European Conference on DSP Research and education*, ESIEE, Noisy Le Grand, FRANCE, Sept. 2000.

[28] G. Baudoin and F. Virolleau. *Les processeurs de traitement du signal, la famille TMS320C50*. DUNOD, ISBN 2 10 00 003049 3, Paris, 1997.

[29] G. Baudoin and F. Virolleau. *DSP-La famille TMS320C54x, développement d'applications*. DUNOD, ISBN 2 10 004646 2, Paris, 2000.

[30] G. Baudoin, F. Virolleau, O. Venard, and P. Jardin. Teaching dsp through the case study of a fsk modem. In *Proc. of* 1^{rst} *European Conference on DSP Research and Education*, Noisy Le Grand, FRANCE, September 1996.

[31] G. Baudoin and A. Zemva. proteus technical report, segmentation des signaux et des images, algorithmes et implantation multiprocesseur. Technical report, ESIEE and University of Ljubljana, 1997 and 1998.

[32] G. Bazin, P. Sangouard, G. Baudoin, C. Ripoll, and P. Nicole. *Revue nano-micro*, chapter Microsystèmes autonomes sans fils. Hermès, to appear in 2000.

[33] F. Bimbot. An evaluation of temporal decomposition. Technical report, Acoustic Research department, AT&T Bell Labs, 1990.

[34] F. Bimbot, G. Chollet, P. Deleglise, and C. Montacie. Temporal decomposition and acoustic-phonetic decoding of speech. In *Proc. IEEE ICASSP 88*, pages 445–448, New York, 1988.

[35] B. Boianov and G. Baudoin. Stress detection through voice analysis. In *Speech and image understanding* 3^{rd} *sloveniain and* 2^{nd} *SDRV Workshop*, University of Ljubljana, Slovenia, April 1996.

[36] B. Boianov, S. Hadjitodorov, and G. Baudoin. Method for evaluation of the energy in the singer formant. *Comptes rendus de l'académie bulgare des sciences*, 48(8), January 1995.

[37] B. Boianov, S Hadjitodorov, and G. Baudoin. Acoustical analysis of pathological voices. In *Proc of 3rd slovenian-german workshop Speech and image understanding 3rd Sloveniain and 2nd SDRV Workshop*, pages 157–166, Ljubljana, Slovenia, 1996.

[38] S. Bruhn. Matrix product vector quantization for very low bit rate speech coding. In *Proc. ICASSP-95*, pages 724–727, 1995.

[39] O. Cappé, J. Laroche, and E. Moulines. Regularised estimation of cepstrum enveloppe from discrete frequency points. In *IEEE ASSP workshop on AP. Of SIG. Proc. to audio and accoustics*, Mohouk, 1995.

[40] J. Cernocký, G. Baudoin, and G. Chollet. Segmental vocoder-going beyong the phonetic approach. In *Proc. IEEE ICASSP'98*, pages 605–608, Seattle USA, 1998.

[41] J. Cernocký. *Speech Processing Using Automatically Derived Segmental Units : Applications to Very Low Rate Coding and Speaker Verification,*. PhD thesis, Université Paris XI Orsay, 1998.

[42] J. Cernocký and G. Baudoin. Représentation du spectre de parole par les multigrammes. In *Proc. XXI-es Journéees d'Etude sur la Parole*, pages 239–242, Avignon, France, June 1996.

[43] J. Cernocký, G. Baudoin, and G. Chollet. Efficient method of speech spectrum description using multigrams. In *Speech and image understanding* 3^{rd} *sloveniain and* 2^{nd} *SDRV Workshop*, University of Ljubljana, Slovenia, April 1996. 139–148.

[44] J. Cernocký, G. Baudoin, and G. Chollet. speech spectrum representation and coding using multigrams with distance. In *Proc. IEEE ICASSP 97*, pages 1343–1346, Munich, Germany, April 1997.

[45] J. Cernocký, G. Baudoin, and G. Chollet. Towards a very low bit rate segmental speech coder. In *Proc. NATO ASI Summer school,Computational Models of Speech Pattern Processing*, Jersey, Great Britain, July 1997.

[46] J. Cernocký, G. Baudoin, and G. Chollet. Segmental vocoder - going beyond the phonetic approach. In *Proc. ICASSP98*, pages 605–608, Seattle, 1998.

[47] J. Cernocký, G. Baudoin, and G. Chollet. The use of alisp for automatic acoustic-phonetic transcription. In *Proc. of SPoSS-ESCA Workshop on Sound Patterns of Spontaneous Speech*, pages 149–152, Aix en Provence France, 1998.

[48] J. Cernocký, G. Baudoin, and G. Chollet. Alisp : Quelques outils pour l'analyse acoustico-phonétique de la parole. *to appear in Revue parole*, 2000.

[49] J. Cernocký, G. Baudoin, D. Petrovska-Delacrétaz, J. Hennebert, and G. Chollet. Automatically derived speech units : Applications to very low bit rate coding and speaker verification. In *Proc. of Workshop on Text Speech and Dialogue (TSD'98)*, Lecture notes in computer science, pages 182–188, Brno, Czech Republic, September 1998. Springer Verlag.

[50] J. Cernocký, I. Kopeček, G. Baudoin, and G. Chollet. Very low bit rate speech coding : comparison of data-driven units with syllable segments. In *Proc. of Workshop on Text Speech and Dialogue (TSD'99)*, Lecture notes in computer science, Mariánské Lázně, Czech Republic, September 1999. Springer Verlag.

[51] M. Chaouche and G. Baudoin. A digital ecg recording system. In *Proc. 1rst Mediterranean conf. Biomedical Engineering*, pages 149–152, Sevilla,Spain, 1986.

[52] Y. M. Cheng and D. O'Shaughnessy. A 450 bps vocoder with natural sounding speech. In *Proc. ICASSP-90*, pages 649–652,, 1990.

[53] D. G. Childers, Ke Wu, D.M. Hicks, and B.Yegnanarayama. Voice conversion. *Speech Communication*, pages 147–158, 1989.

[54] P. A. Chou and T. Lookabaugh. Variable dimension vector quantization of linear predictive coefficients of speech. In *Proc. IEEE ICASSP 94*, pages I–505–508, Adelaide, June 1994.

[55] J. R. Crosmer and T.P. Barnwell. A low bit rate segment vocoder based on line spectrum pairs. In *Proc. ICASSP-85*, pages 240–243, 1985.

[56] S. Deligne. *Modèles de séquences de longueurs variables : Application au traitement du langage écrit et de la parole*. PhD thesis, École nationale supérieure des télécommunications (ENST), Paris, 1996.

[57] S. Deligne and F. Bimbot. Language modelling by variable length sequences : Theoretical formulation and evaluation of multigrams. In *Proc. IEEE ICASSP 95*, pages 169–172, Detroit, USA, 1995.

[58] P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data with the em algorithm. *J. Roy. Stat.*, 39(1) :1–38, 1977.

[59] P. Dymarski, N. Moreau, and A. Vigier. Optimal and sub-optimal algorithms for selecting the excitation in linear predictive coders. In *Proc. IEEE ICASSP 90*, pages 41–44, 1990.

[60] B. Fette and C. Jaskie. A 600 bps lpc voice coder. In *proc. MILCOM-91*, pages 1215–1219, 1991.

[61] J. L. Flanagan. *Speech Analysis, Synthesis and Perception*. Springer Verlag New York, 1965 second ed. 1972.

[62] R. Di Francesco. Codage algébrique de la parole : prédiction linéaire à excitation par code ternaire. *Annales des Télécommunications*, 47(5-6), 1992.

[63] T. Fukada, M. Bacchiani, K. Paliwal, and Y. Sagisaka. Speech recognition based on acoustically derived segment units. In *Proc. of ICSLP'96*, pages 1077–1080, Philadelphia, USA, September 1996.

[64] A. Gersho. *Vector Quantization and Signal Compression*. Kluwer Academic Publisher, 1996.

[65] L. A. Gerson and M. A. JAsiuk. Vector sum excited linear prediction (vselp) speech coding at 8 kbps. In *Proc. IEEE ICASSP 90*, pages 461–464, 1990.

[66] P. Gournay and F. Chartier. A 1200 bps hsx speech coder for very low bit rate communications. In *IEEE Workshop on Signal Processing System SiPS'98*, Boston, 1998.

[67] D. Griffin and J. Lim. Multiband excitation vocoder. *IEEE trans. ASSP*, 36(8) :1223–1235, 1988.

[68] J. Haagen, H. Nielsen, and S. D. Hansen. 2,4 kbps speech coding : A new strategy for coding voiced residual. In *Proc. IEEE ICASSP 88*, pages 151–154, 1988.

[69] M. Ismail and K. Ponting. Between recognition and synthesis 300 bit/s speech coding. In *Proc. Eurospeech-97*, pages 441–444, Rhodos, 1997.

[70] N. Iwahashi and Y. Sagisaka. Speech spectrum transformation by speaker interpolation. In *Proc. ICASSP 94*, pages 461–464, 1994.

[71] D. Janu, G.Baudoin, J.-F. Bercher, and O. Venard. Design of a cdma system simulator and implementation on a tms320c6201. In *Proc. of* 2^{nd} *European Conference on DSP Research and education*, pages 119–124, ESIEE, Noisy Le Grand, FRANCE, 1998.

[72] C. Jaskie and B. fette. A survey of low bit rate vocoders. In *DSP & Multimedia Technology*, pages 26–40, April 1994.

[73] P. Jeanrenaud and P. Peterson. Segment vocoder based on reconstruction with natural segment. In *Proc. ICASSP-91*, pages 605–608, 1991.

[74] M. Jelinek and G. Baudoin. *Speech Recognition and Coding, new advances and trends*, chapter Excitation construction for the robust low bit rate CELP speech coder, pages 439–443. Springer-Verlag, nato asi serie f., edited by a. rubio & jm lopez edition, 1995.

[75] G.S. Kang and I.J. Fransen. Application of line spectrum pairs to low-bit rate speech encoders. In *Proc. ICASSP-85*, pages 244–247, 1985.

[76] D. P. Kemp, Collura J. S., and Tremain. T. E. Multiframe coding of lpc parameters at 600-800 bps. In *Proc. ICASSP-91*, pages 609–612, 1991.

[77] W. Kleijn. Encoding speech using prototype waveforms. *IEEE Trans. Speech Audio Processing*, 1(4) :386–399, 1993.

[78] W. Kleijn and J. Haagen. *Speech coding and synthesis*, chapter Waveform interpolation for Coding and Synthesis. W.B. Kleijn and K.K. Paliwal Editors, Elsevier., 1995.

[79] W. B. Kleijn and J. Haagen. A speech coder based on decomposition of characteristic waveforms. In *Proc. ICASSP-95*, pages 508–511, 1995.

[80] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum. Fast methods for the celp speech coding algorithm. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(8) :1330–1342, August 1990.

[81] C. Laflamme, J. P. Adoul, S. Morissette, and P. Mabilleau. 16k̃bps wideband celp speech coding technique based on algebraic celp. In *Proc. IEEE ICASSP 91*, pages 13–16, 1991.

[82] C. Laflamme, R. Salami, R. MAtmti, and J.P. Adoul. Harmonic-stochastic excitation (hsx) speech coding below 4 kbps. In *Proc. ICASSP-96*, pages 204–207, 1996.

[83] Y. Linde, A. Buzo, and R.M. Gray. Algorithm for vector quantization design. In *IEEE tran. commun., vol. COM-28*, pages 84–95, 1980.

[84] J. Liu. *Amélioration de la décomposition source-filtre du signal vocal, étude de la variabilité des paramètres de l'onde glottique, application à la transformation de voix.* PhD thesis, Université d'Orsay, France, 1993.

[85] J. Liu, G. Baudoin, and G. Chollet. Studies of glottal excitation and vocal tract parameters using inverse filtering and a parametrised input model. In *Proc. ICSLP'92*, pages 1051–1054, 1992.

[86] Y. J. Liu and J. Rothweiler. A high quality speech coder at 400 bps. In *Proc. ICASSP-89*, pages 204–206, 1989.

[87] Y. P. Liu. On reducing the bit rate of a celp-based speech coder. In *Proc. IEEE ICASSP 92*, pages I–49–I–52, 1992.

[88] J. M. López-Soler and N. Farvardin. A combined quantization-interpolation scheme for very low bit rate coding of speech LSP parameters. In *Proc. IEEE ICASSP 93*, pages II–21–24, Minneapolis, 1993.

[89] H. Matsumoto and H.Wakita. Vowel normalization by frequency warped spectral matching. *Speech Communication*, (5) :239–251, 1986.

[90] M. Mauc and G. Baudoin. Codeur celp à complexité réduite. In *journal de physique IV, colloque C1, supplément du journal de physique III*, pages 2 :C1–327–C1–330, France, April 1992.

[91] M. Mauc and G. Baudoin. Reduced complexity celp coder. In *Proc. of IEEE ICASSP'92*, pages I–53–I–56, San Francisco, USA, 1992.

[92] M. Mauc, G. Baudoin, and M. Jelinek. Complexity reduction for the fs1016 at 4800 bps celp coder. In *Proc. of Eurospeech'93*, pages I–245–I–248, Berlin, Germany, 1993.

[93] M. Mauc, G. Baudoin, and M. Jelinek. Complexity reduction for the fs1016 coder with multistagew search. In *Proc. IEEE ICASSP 94*, pages I–261–I–264, Adelaide, Australia, April 1994.

[94] M. Mauc, G. Baudoin, M. Jelinek, and P. Jardin. Reduced complexity celp coder with a multistage search. In *Proc. of Eusipco'92*, pages 523–526, Brussels, Belgium, 1992.

[95] R. McAulay and T. Champion. Improved interoperable 2.4 kbps lpc using sinusoidal transform coder techniques. In *Proc. ICASSP-90*, pages 641–643, 1990.

[96] R. McAulay and T. Quatieri. Multirate sinusoidal transform coding at rates from 2.4 kbps to 8kbps. In *Proc. ICASSP-87*, Dallas, 1987.

[97] R. McAulay and T. Quatieri. Sine-wave phase coding at low data rates. In *Proc. ICASSP-91*, pages 577–580, 1991.

[98] R. McAulay and T.Quatieri. Speech analysis-synthesis based on a sinusoidal representation of speech. *IEEE trans. ASSP*, 34(4) :744, 1985.

[99] A. McCree, George K. Truong, E. B., T. P. Barnwell, and V. Viswanathan. A 2,4 kbits/s melp coder candidate for the new u.s. federal standard. In *Proc. ICASSP-96*, pages 200–203, 1996.

[100] J. Menez, C. Galand, M. Rosso, and F. Bottau. Adaptive code excited linear predictive coder (acelpc). In *Proc. IEEE ICASSP 89*, pages 132–135, 1989.

[101] H. Mizuno and M. Abe. Voice conversion based on piecewise linear conversion rules of formant frequency and spectrum tilt. In *Proc. ICASSP 93*, pages 469–472, 1994.

[102] H. Mizuno, M. Abe, and T. Hirokawa. Waveform-based speech synthesis approach with a formant frequency modification. In *Proc. ICASSP 93*, pages 195–198, 1993.

[103] N. Moreau. *Techniques de compression des signaux*. Masson, Paris, 1995.

[104] N. Moreau and P. Dymarski. Successive orthogonaliations in the multistage celp coder. In *Proc. IEEE ICASSP 92*, pages I–61–I–64, 1992.

[105] B. Mouy, P. de la Noue, and G.Goudezeune. Nato stanag 4479 : a standard for an 800 bps vocoder and channel coding in hf-eccm system. In *Proc. IEEE ICASSP-95*, pages 480–483, 1995.

[106] M. Nishiguchi, A. Inoue, Y. Maeda, and J. Matsumoto. Parametric speech coding-hvxc at 2.0-4.0 kbps. In *Proc IEEE Workshopon Speech Coding*, Munich, 1997.

[107] Parameters and coding characteristics that must be common to assure interoperability of 2400 bps linear predictive encoded speech. Technical report, NATO Standard STANAG-4198-Ed1, 13 February 1984.

[108] P. Peterson, P. Jeanrenaud, and J. vandegrift. Improving intelligibility at 300b/s segment vocoder. In *Proc. ICASSP-90*, pages 653–656, 1990.

[109] Picone and G. R. Doddington. A phonetic vocoder. In *Proc. ICASSP-89*, pages 580–583, 1989.

[110] J. Potage, D. Rochette, and G. Mathevon. Speech encoding techniques for low bit rate coding applicable to naval communications. *Rev. Tech. Thomson-CSF, Vol.18, No1*, 18(1) :171–205, March 1986.

[111] C. Ribeiro and M. Trancoso. Phonetic vocoding with speaker adaptation. In *Proc. Eurospeech-97*, pages 1291–1294, Rhodes, 1997.

[112] J. Rothweiler. Performances of a real time low rate voice coder. In *Proc. ICASSP-86*, pages 3039–3042, 1986.

[113] S. Roucos, R. Schwarz, and J. Makhoul. Segment quantization for very-low rate speech coding. In *Proc. ICASSP-82*, pages 1565–1568, Paris, 1982.

[114] S. Roucos, R. Schwarz, and J. Makhoul. A segment vocoder at 150 b/s. In *Proc. ICASSP-83*, pages 61–64, 1983.

[115] S. Roucos and A.M. Wilgus. The waveform segment vocoder : A new approach for very low rate speech coding. In *Proc. ICASSP-85*, pages 236–239, 1985.

[116] M.R. Schroeder and B. Atal. Code-excited linear prediction(celp) : High quality speech at very low bit rates. In *Proc. IEEE ICASSP 85*, pages 937–940, Tamp., 1985.

[117] R. M. Schwartz and Roucos. R. M. A comparison of methods for 300-400 b/s vocoders. In *Proc. ICASSP-83*, pages 69–72, 1983.

[118] Y. Shiraki and M. Honda. LPC speech coding based on variable length segment quantization. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(9) :1437–1444, September 1988.

[119] Y. Shoham. Very low complexity interpolative speech coding at 1.2 to 2.4 kbps. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1599–1602, Munich, 1997.

[120] Spanias. Speech coding : A tutorial review. *Proc. IEEE*, 82(10) :1541–1582, October 1994.

[121] Specification for the analog to digital conversion of voice by 2400 bit/s mixed excitation linear prediction. Technical report, Federal Information Processing Standards Publication (FOPS PUB) draft, 1998.

[122] I. Stylianou, T. Dutoit, and J. Schroeter. Diphone concatenation using a harmonic plus noise model of speech. In *Proc. of Eurospeech'97*, Rhodes, Greece, September 1997.

[123] Y. Stylianou. *Harmonic plus Noise Model for Speech, combined with statistical methods sor speech and speaker modification*. PhD thesis, École nationale supérieure des télécommunications (ENST), Paris, 1996.

[124] L. M. Supplee, R.P. Cohn, J.S. Collura, and A.V. McCree. Melp : The new federal standard at 2400 bits/s. In *Proc. ICASSP-97*, pages 1591–1594, Munich, 1997.

[125] K. Tokuda, T. Masuko, J. Hiroi, T. Kobayashi, and T. Kitamara. A very low bit rate speech coder usinghmm-based speech recognition/synthesis techniques. In *Proc. ICASSP-98*, pages 609–612, 1998.

[126] I. Trancoso and B. Atal. Efficient procedures for finding the optimum innovation sequence in stochastic coders. In *Proc. IEEE ICASSP 86*, pages 2379–2382, Tokio, Japan, 1986.

[127] I. Trancoso and B. Atal. Efficient search procedures for selecting the optimum innovation sequence in stochastic coders. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(3) :385–396, March 1990.

[128] T. E. Tremain. the government standard linear predictive coding algorithm : Lpc10. *Speech Technology*, 1(2) :40–49, 1982.

[129] H. Valbret. *Système de conversion de voix pour la synthèse de parole*. PhD thesis, Thèse de Doctorat de Telecom Paris, 1992.

[130] H. Valbret, E. Moulines, and J.P. Tubach. Voice transformation using psola technique. *Speech Communication*, (11) :175–187, 1992.

[131] S. Wang and A. Gersho. Improved phonetically segmented vector excitation coding at 3,4 kbps. In *Proc. IEEE ICASSP 92*, pages I–349–I–352, 1992.

[132] D.Y. Wong, B.H. Juang, and D.Y. Cheng. Very low data rate speech compression using lpc vector and matrix quantization. In *Proc. ICASSP-83*, pages I–65–68, 1983.

[133] Z. Xiongwei and C. Xienzki. A new excitation model for lpc vocoder at 2,4 kb/s. In *Proc. IEEE ICASSP 92*, pages I–65–I–68, 1992.

[134] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *The HTK book*. Entropics Cambridge Research Lab., Cambridge, UK, 1996.