ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE
M.Ing.

PAR
RÉMI PREISS

ÉTUDE DU CANAL TÉLÉPHONIQUE DANS UN SYSTÈME DE RECONNAISSANCE ROBUSTE DE LA PAROLE

MONTRÉAL, LE 5 AVRIL 2006

CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

M. Gheorghe Marcel Gabrea, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Christian Gargour, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Jean-Marc Lina, membre du jury
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 29 MARS 2006
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ÉTUDE DU CANAL TÉLÉPHONIQUE DANS UN SYSTÈME DE RECONNAISSANCE ROBUSTE DE LA PAROLE

Rémi Preiss

SOMMAIRE

Les applications commerciales de la reconnaissance de la parole sont en grande majorité établies via un média téléphonique fixe ou mobile.

De tels systèmes de reconnaissance ne sont pour l'heure pas parfaits car il subsiste encore plusieurs contraintes d'ordre linguistiques, marketing et techniques. Ce sont ces dernières qui nous intéressent et plus particulièrement l'effet du canal téléphonique sur les performances des systèmes de reconnaissance.

Ils existent des techniques pour compenser l'effet du canal que l'on peut regrouper en deux catégories, les techniques paramétriques qui tentent de corriger les coefficients des vecteurs de parole bruitée, et les techniques dites d'adaptation des modèles qui cherchent à modifier les caractéristiques des modèles d'entraînement pour maximiser l'appariement entre les données de test et d'entraînement. Dans cette maîtrise, nous avons étudié plusieurs techniques d'adaptation des paramètres, et aussi proposé une méthode d'adaptation des modèles. Elle consiste à construire des modélisations de canaux téléphoniques pour créer une nouvelle base d'apprentissage multiréférences. Le système est alors appris avec la base de données TIMIT bruitée par ces modélisations, NTIMIT a servi pour la phase de test. Au final, cette méthode s'est révélée concluante car le taux d'erreurs a été réduit de manière significative.

STUDY OF THE VOICE CHANNEL IN ROBUST SPEECH RECOGNITION

Rémi Preiss

ABSTRACT

The telephone channel triggers, by the reduction of the signal bandwidth, a drop of the performances of most of the recognition systems which belong to speaker identification or continuous speech recognition.

Many compensation techniques have been developed to reduce the unmatching issue between the training and the test databases which is supposed to be the main cause of the results decrease. We can gather these techniques in two categories: (1) *feature compensation* in which the representation of the acoustic vector is adjusted, and (2) *model adaptation* in which the HMM parameters are modified to get closer of the testing environments.

In this master, we evaluated several feature compensation techniques and we also developed a model adaptation method. The main idea is to develop a modeling of the PSTN channel. Then, we trained HMM with the TIMIT database passed through different PSTN channel modeling in order to get a new multireferences training database adapted to the convolutive effect of the reduction bandwidth. The NTIMIT database has been used for testing. Finally, the word error rate has been significantly decreased.

REMERCIEMENTS

Je tiens à remercier particulièrement mon directeur de recherche, Monsieur Gheorghe Marcel Gabrea, professeur au département de génie électrique à l'École de technologie supérieure de Montréal, qui m'a guidé et encouragé dans ce projet.

Je remercie également les membres du laboratoire LATIS (Laboratoire du traitement de l'information et des signaux) pour leur bonne humeur quotidienne et l'ambiance chaleureuse qu'ils m ont apportée.

Enfin, merci à mon amie, Julia, sans qui je ne serais probablement pas au Canada.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS ET SIGLES

HMM	Hidden Markov Model
HTK	Hidden markov models ToolKit
RAP	Reconnaissance Automatique de la Parole
TIMIT	Texas Instrument, Massachusetts Institute of Technology
NTIMIT	Network TIMIT
MFCC	Mel Frequency Cepstral Coefficients
CMN	Cepstral Mean Normalization
ACN	Augmented Cepstral Normalization
VAD	Voice Activity Detector
RASTA	RelAtive SpecTrAl
BMC	Bruité par des Modélisations de Canaux téléphoniques
PSTN	Public Switch Telephone Network
MLE	Maximum Likelihood Estimation
WER	Word Error Rate
RIF	Réponse Impulsionnelle Finie
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform

INTRODUCTION

L'augmentation des capacités de calcul et de mémoire des ordinateurs a favorisé l'émergence de la reconnaissance vocale. Le nombre d'enseignants-chercheurs dans le domaine s'est véritablement multiplié en vingt ans.

Une des raisons permettant d'expliquer l'effervescence dans ce domaine est l'attrait de plus en plus grandissant que suscite la reconnaissance vocale dans les applications commerciales : portail téléphonique des grandes compagnies (Bell Canada par exemple), applications biométriques (reconnaissance du locuteur) et aussi des applications à buts purement lucratifs (la possibilité de connaître le nom de la musique qui est en train de se jouer dans un bar en appelant depuis son cellulaire un numéro). En 1994, l'annonce de la commercialisation par IBM du logiciel de dictée vocale a fait coulé beaucoup d'encre, d'une part, pour la nouvelle ère futuriste que ce produit laissait présager, et d'autre part, pour les performances annoncées par le fabriquant mais contestées par les experts du domaine [1].

Introduite à la fin des années 60, la théorie de la reconnaissance vocale a suscité beaucoup de controverses à l'époque. Par exemple, M. Pierce en 1969 dans le *Journal of the Acoustical Society of America* déclara : « il est impossible de reconnaître de la parole, que ce soit mot par mot ou son par son, et les gens qui prétendent y parvenir sont soit des fantaisistes soit des imposteurs! » [2]. De nos jours, des taux de reconnaissance avoisinant les 99 % sont observables sur des petits vocabulaires (reconnaissance de mots isolés, reconnaissance de digits) indépendamment du locuteur. Toutefois, à l'heure actuelle, l'impact des systèmes basés sur la reconnaissance de la parole est tout à fait négligeable dans notre vie de tous les jours.

Les applications commerciales de la reconnaissance de la parole sont en grande majorité établies via un média téléphonique fixe ou mobile. De tels systèmes de reconnaissance ne sont pour l'heure pas parfaits car il subsiste encore plusieurs contraintes d'ordre

linguistiques, marketing et techniques. Ce sont ces dernières qui vont nous intéresser et plus particulièrement l'effet du canal téléphonique sur les performances des systèmes de reconnaissance.

En effet, la voix téléphonique ne présente pas les mêmes caractéristiques physiques que la parole directe et cela a des conséquences néfastes pour l'application. Si la phase d'apprentissage du système de reconnaissance est réalisée sur une base de données propre, le taux de reconnaissance des appelants téléphoniques sera mauvais à cause des nombreuses dégradations qu'apportent l'acquisition et le cheminement du signal téléphonique. Il semble donc nécessaire d'apporter au système des corrections pour le rendre plus robuste, plus résistant aux bruits. D'une manière très simpliste, entraîner un système de reconnaissance signifie lui apprendre quelles sont les phrases, quels sont les mots et les accents les plus probables qu'il risque de rencontrer lors de la phase de reconnaissance.

La théorie de la reconnaissance de la parole continue fait l'objet d'une étude détaillée au Chapitre 1 où nous rappelons le fondement théorique des techniques d'apprentissage des Modèles de Markov Cachés (HMM). Nous verrons également comment extraire du signal de parole brut les représentations les plus adéquates à fournir au système.

L'objet de ce mémoire est aussi le canal téléphonique et ses conséquences sur un système de reconnaissance de la parole continue à grand vocabulaire indépendamment du locuteur. Au Chapitre 2, une étude du canal téléphonique sera présentée ainsi qu'une liste des différents bruits qu'il occasionne sur le signal vocal. Nous détaillerons ensuite, dans ce même chapitre, quelques techniques pour compenser l'effet nuisible du canal sur un système de reconnaissance où une chute des performances est systématiquement observée. Ces techniques se regroupent en deux grandes catégories : l'adaptation des paramètres et l'adaptation des modèles de HMM.

Le Chapitre 3 explique la construction du système de référence à l'aide de la boite à outils HTK développée par l'Université de Cambridge.

Au Chapitre 4, nous commencerons par présenter la base de données TIMIT qui nous servira comme base de référence pour apprendre et tester nos modèles. Ensuite, nous présenterons les expérimentations effectuées ainsi que les résultats obtenus tout au long de l'étude, à commencer par l'explication de la modélisation du canal téléphonique utilisée pour bruiter la base TIMIT afin de disposer d'un nouveau corpus d'apprentissage multiréférences. Grâce à cette modélisation, nous proposerons une méthode nouvelle de compensation pour améliorer la robustesse des systèmes de reconnaissance via le téléphone fixe. La création de cette nouvelle base d'apprentissage représente la contribution principale apportée dans ce projet. Elle à d'ailleurs fait l'objet d'une publication [3].

Il est important de préciser que cette étude du canal téléphonique vise à corriger l'effet convolutif du canal en termes de restriction de largeur de bande. L'effet du microphone et des différents bruits additifs apportés lors de l'acquisition du signal n'ont pas été étudiés.

CHAPITRE 1

LA RECONNAISSANCE DE LA PAROLE

Le but de la Reconnaissance Automatique de la Parole (RAP) est d'extraire d'un signal vocal, par le biais d'un ordinateur, l'information lexicale contenue dans ce signal de parole.

Les applications liées à la RAP ont réellement pris leur essor sous l'ère de la micro-électronique qui a fournit aux chercheurs les moyens physiques pour tester les théories développées longtemps auparavant. Ainsi, les méthodes les plus performantes actuellement sont des méthodes statistiques utilisant le formalisme des modèles de Markov développé au milieu des années 70 [4, 5]. Ces techniques rendent concevable la reconnaissance de la parole continue à grands vocabulaires et indépendamment du locuteur. L'information sur le signal vocal en termes de traitement du signal est connue depuis plus longtemps encore. On peut trouver encore de nos jours des ouvrages datant des années 50.

Le signal de parole « à l'état brut » n'est pas exploitable directement par un système de RAP à cause de sa grande redondance. Nous verrons qu'il a fallu trouver des représentations plus compactes du signal. Celui-ci sera classiquement représenté comme une suite de vecteurs de paramètres calculée de manière à maximiser la discrimination entre ces paramètres. Les représentations les plus efficaces sont les représentations spectrales tenant compte de certaines connaissances acquises sur la production, la perception et la variabilité de la parole. En effet, un symbole donné peut avoir plusieurs réalisations acoustiques différentes en fonction de l'identité de la personne (variabilité inter-locuteur), de l'état de la personne (variabilité intra-locuteur), et de l'environnement (facteurs contextuels).

1.1 Complexité et variabilité du signal vocal

La conception d'un système de reconnaissance de la parole est rendue difficile par la complexité et la variabilité du signal vocal. D'une manière générale, la parole peut être vue comme la concaténation d'un signal acoustique (physique) et d'un espace de communication linguistique (abstrait). Cela fait apparaitre deux problèmes importants à prendre en compte :

- Le message doit être suffisant clair pour être intelligible, en effet, il ne doit pas comporter trop de bruits induits par l'émetteur ou encore par le support de transmission (canal téléphonique, bruit ambiant dans l'air, etc.).

- Une écoute parfaite du signal vocal ne signifie pas pour autant une compréhension du contenu du message. Il faut, pour cela, tout un ensemble de règles linguistiques, grammaticales, syntaxiques qui, pour l'homme, sont acquises tout au long de l'apprentissage de la langue. Mais pour un système de reconnaissance, il faut lui préciser ces règles par le biais d'une grammaire. Une grammaire, au sens formel du terme, est un ensemble de règles de production de suites de mots appartenant à un vocabulaire préalablement connu.

Les propriétés intrinsèques du signal vocal varient énormément d'un locuteur à l'autre, en fonction de son sexe, de son âge, de son état émotif, de son attitude, et même des contraintes environnantes qui l'entrainent à modifier sa voix pour être mieux compris par son interlocuteur (phénomène connu sous le nom de l'effet Lombard [6, 7]).

1.2 Redondance du signal vocal

Le signal de parole utilise essentiellement les fréquences comprises entre 100 et 5000 Hz, mais on peut évincer une partie de ces fréquences et l'intelligibilité demeure. Pour mettre en évidence l'aspect redondant du signal vocal, on peut, par exemple, l'écrêter voire le discrétiser complètement (remplaçant les valeurs positives par +1 et les valeurs négatives par -1) et il demeure encore compréhensible.

Le signal transporte énormément d'informations (le fondamental, la prosodie, le timbre, les phonèmes, etc..) dont toutes ne sont pas forcément nécessaires à la compréhension basique du message. Par conséquent, ceci impose aux systèmes de reconnaissance vocale de n'extraire que l'information nécessaire à son application.

Par exemple, un signal de parole échantillonné à 16 kHz sur 16 bits équivaut à un débit de 256 kbit/s. Sachant qu'une tâche de reconnaissance phonétique recherche classiquement une dizaine de phonèmes à la seconde, cela représente une compression de près de 10^4 du débit initial. Ce constat motive la recherche d'une représentation plus compacte du signal en choisissant une transformation qui rende les paramètres plus pertinents à la tache de reconnaissance car la redondance rend la tache de reconnaissance impossible dans le domaine temporel.

1.3 Extraction des paramètres

L'extraction et le calcul des paramètres a pour objectif de fournir au système de reconnaissance une représentation du signal vocal plus adéquate que les échantillons temporels non adaptés au domaine. Ils existent plusieurs types de transformations utilisables en reconnaissance mais nous nous attacherons qu'à décrire celle la plus utilisée : la transformation cepstrale.

1.3.1 Échantillonnage

Considérons un signal vocal analogique $x_A(t)$. Après échantillonnage de fréquence f_e, le signal discret $x_D(n)$ est obtenu par

$$x_D(n) = x_A(nT_e) \tag{0.1}$$

Où T_e est l'inverse de la fréquence f_e. Le théorème d'échantillonnage de Shannon [8] impose à f_e de valoir au moins le double de la largeur de bande du signal analogique. On peut considérer que la majeure partie des constituants de la voix (fréquence fondamentale, premiers formants) se situe entre 100 et 5000 Hz. La considération précédente et l'optique d'un gain des temps dans les calculs incitent les applications de traitement de la parole à limiter le spectre de la voix à 4 kHz par le biais d'un filtrage passe-bande. Elles utilisent donc couramment une fréquence d'échantillonnage de 8 kHz.

1.3.2 Pré-accentuation

L'expérience montre que, dans le signal de parole, la partie haute du spectre est moins importante. Pour accentuer la partie haute fréquence du spectre, il est conseillé d'effectuer une pré-accentuation du signal par le biais d'un filtrage de transmittance $1-\mu z^{-1}$ avec μ compris entre 0.9 et 1 (0.97 est la valeur usuelle).

1.3.3 Fenêtrage

Il est nécessaire de découper le signal en tranches pour appliquer par la suite la transformée de Fourier discrète. On peut trouver dans [9] un rappel des méthodes d'analyse de Fourier. Comment définir la durée N de ces tranches (ou trames) ?

La parole est un phénomène non stationnaire, ses propriétés statistiques changent continuellement dans le temps. Cependant, l'observation du signal de la parole indique qu'il n'évolue pas ou peu sur des durées de quelques millisecondes. On peut donc considérer ce signal comme étant stationnaire durant ce temps (stationnarité à court terme).

En pratique (Figure 1), le signal est découpé souvent en trames d'analyse de 25 ms et 2 trames successives se chevauchent sur 15 ms. Il y a donc 10 ms d'information propre à la trame sur les 25 ms que dure la tranche. Ces 25 ms sont en fait un compromis : de trop longues tranches engloberaient des parties non stationnaires du signal, et de trop courtes ne permettraient par une correcte analyse spectrale du signal. Finalement, si la fréquence d'échantillonnage du signal est 8 kHz, $N = 25 \times 8 = 200$ échantillons par trame.

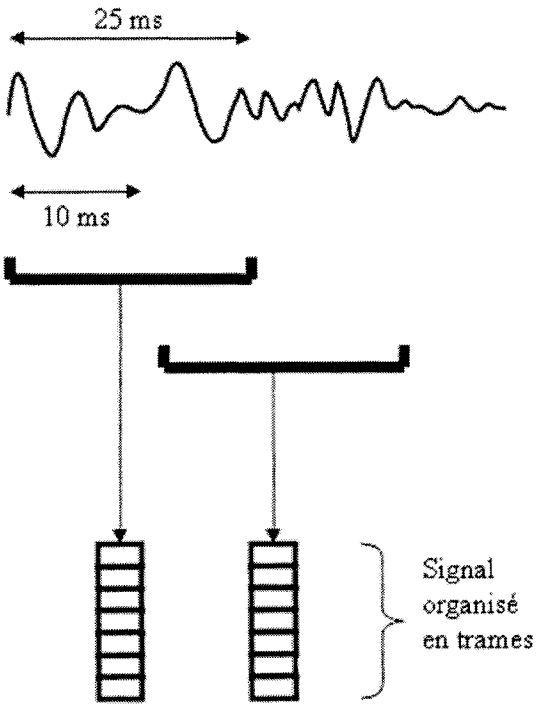

Figure 1 Exemple de fenêtrage [10]

Ensuite, on applique sur chaque trame la pondération de Hamming. Chaque échantillon $s(n)$, $1 \leq n \leq N$, de chaque trame est pondéré par le facteur suivant :

$$s(n) \rightarrow \left\{0.54 - 0.46\cos\left(\frac{2\pi(n-1)}{N-1}\right)\right\} \cdot s(n) \qquad (0.2)$$

La pondération de Hamming vient atténuer les fortes discontinuités introduites par le découpage rectangulaire aux extrémités des tranches (apparition de lobes secondaires lors du calcul de la FFT de la fenêtre).

1.3.4 Calcul des coefficients cepstraux

Le calcul des coefficients modélisant chaque trame est le stade ultime de l'analyse acoustique. À l'issue de cette phase, les coefficients seront directement utilisés par l'analyse probabiliste des algorithmes de reconnaissance. Les coefficients les mieux adaptés dans ce domaine sont les coefficients issus des méthodes paramétriques. Les méthodes paramétriques reposent sur une étude *a priori* de la synthèse vocale dans laquelle le modèle de production de la parole est considéré comme linéaire résultant de la convolution d'une source et d'un filtre. Le cepstre [11], le codage prédictif linéaire [8] et l'analyse prédictive linéaire perceptuelle [12] sont les trois méthodes paramétriques les plus répandues en traitement de la voix. Nous nous attacherons à décrire la première car mieux adaptée aux environnements bruités ce qui est précisément l'objet de cette étude.

L'appareil phonatoire humain peut être modélisé comme un système composé d'une excitation et d'un filtre [13]. Pour la parole voisée, la source, générée par le larynx, est modélisée comme un train d'impulsions à la fréquence du fondamental. Le filtre est quant à lui assimilé à l'ensemble constitué du conduit vocal, de la cavité résonante et du

rayonnement des lèvres. L'analyse cepstrale découle de ce modèle de production de la parole par une déconvolution de la source et du filtre. Le cepstre est défini comme la transformée de Fourier inverse (FFT^{-1}) du logarithme du module du spectre.

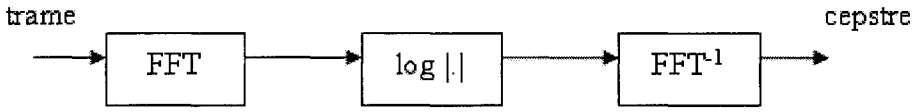

Figure 2 Processus de calcul du cepstre

Comme le spectre, le cepstre est une transformation du signal vocal qui permet d'extraire une propriété importante : les composants constituants de la source et du filtre seront combinés linéairement dans le cepstre et donc, si c'est le but, il sera possible d'éliminer un des ces composants.

Toutefois, l'oreille humaine n'est pas sensible selon une échelle linéaire d'intensité mais de type logarithmique. En effet, après 500 Hz, l'oreille perçoit moins d'une octave pour un doublement de la fréquence. Des expériences psycho-acoustiques ont alors permis d'établir la loi qui relie la fréquence et la hauteur perçue d'un son : l'échelle des Mels où le *Mel*. Voici la correspondance entre la fréquence en Hertz f_H et la valeur en *Mel* *mel(f)* [9]:

$$mel(f) = 2595 \cdot \log_{10}(1 + \frac{f_H}{700}) \tag{0.3}$$

Pourtant, l'utilisation de cette unité n'est pas encore suffisante. Pour avoir une largeur de bande relative qui reste constante, le banc de filtres Mel est construit à partir de filtres triangulaires positionnés uniformément sur l'échelle Mel donc non uniformément sur l'échelle fréquentielle.

Les coefficients cepstraux MFCC (*Mel Frequency Cepstral Coefficients*) peuvent être calculés directement à partir du logarithme des énergies sortant d'un banc de filtres en échelle Mel par la transformée de Fourier inverse. La Figure 3 résume le procédé d'obtention des paramètres MFCC.

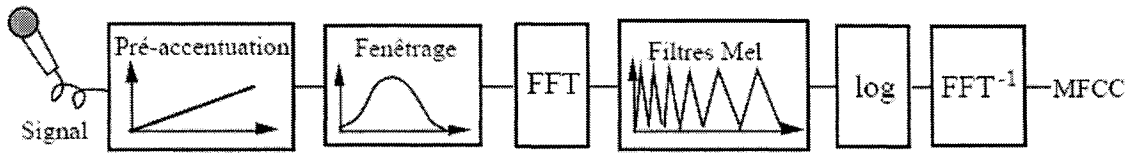

Figure 3 Calcul des coefficients MFCC [14]

On préférera en pratique utiliser la transformée en cosinus discret (DCT) au lieu de la transformée de Fourier inverse. Chaque coefficient cepstral est donc obtenu par l'équation suivante [10] :

$$c_i = \sqrt{\frac{2}{P}\sum_{j=1}^{P} m_j \cos\left(\frac{\pi \cdot i}{P}(j-0.5)\right)} \tag{0.4}$$

Où P est le nombre de filtres MEL et $\{m_j\}$ sont les log-énergies en sortie des filtres MEL.

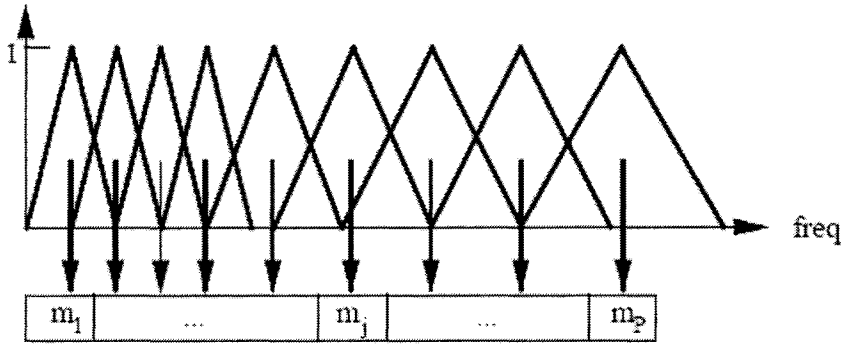

Figure 4 Échelle MEL et banc de filtres [10]

A partir d'une vingtaine de filtres en échelle Mel (Figure 4), une dizaine de coefficients cepstraux est généralement considérée comme suffisamment discriminante pour la reconnaissance de la parole. Il en résulte une diminution importante du nombre des données par vecteur de paramètres.

Ajoutons aussi que l'on ne conserve seulement les premiers coefficients (c_1 à c_{12} par exemple) car ils correspondent aux variations lentes de la densité spectrale, à l'enveloppe générale du spectre contenant les informations sur les formants. Les échantillons d'ordre plus élevé correspondent, quant à eux, aux variations rapides du spectre. Ces derniers sont en général caractéristiques des harmoniques de la fréquence fondamentale des cordes vocales et donc de l'intonation. Ils ne sont pris en compte que dans une moindre mesure dans le processus de reconnaissance.
Le principal avantage des coefficients cepstraux réside dans le fait qu'ils sont généralement décorrélés et cela permet de manipuler des matrices de covariances diagonales pour leurs utilisations ultérieures dans les modèles de Markov.

Enfin, dans l'idée de tenir compte par la suite d'un canal de transmission, l'effet du canal sur les données de parole non bruitées se traduit par une multiplication du spectre du signal vocal par la fonction de transfert du canal. Dans le domaine cepstral, cette multiplication devient une simple addition et le bruit apporté par le canal peut être enlevé par soustraction.

1.4 Les modèles de Markov cachés (HMM)

Les modèles de Markov cachés (*Hidden Markov Models* ou HMM) sont devenus la solution par excellence au paradigme de la reconnaissance de la parole. Ils s'inscrivent dans une approche statistique de la reconnaissance vocale assimilant les unités acoustiques de la parole à un processus aléatoire. Cette théorie fut introduite par Bahl,

Jelinek et Baker en 1975 [4] [15] et Jelinek en 1976 pour la reconnaissance de la parole continue [5].

Un modèle de Markov *M* est un automate probabiliste à *N* états. Un processus aléatoire parcourt l'automate en émettant à chaque sortie d'état une observation appartenant à un nombre fini d'observations. L'état dans lequel se trouve le processus à la date *t* n'est pas connu, d'où l'appellation de modèles cachés. Seule l'observation émise nous revient. Les états sont reliés entre eux par des transitions. Dans l'application de la reconnaissance de la parole, les modèles de Markov sont des automates gauche-droite, c'est à dire que les transitions sont dans un seul sens, de la gauche vers la droite, modélisant ainsi le fait que la parole est un processus irréversible. On envisage également des processus de Markov dit du premier ordre. Cela signifie que la probabilité de passer de l'état *i* vers l'état *j* en émettant le symbole *v* ne dépend ni du temps ni des états parcourus aux instants antérieurs.
Pour résumer, un modèle de Markov est défini par les éléments suivants :

- $S = \{ s_i \}_{1 \le i \le N}$, les N états de l'automate,
- $A = \{ a_{ij} \}_{1 \le i,j \le N}$, les probabilités de transition entre les états i et j,
  $a_{ij} = P(q_t = j / q_{t-1} = i)$ avec q_t l'état occupé à l'instant t,
- $V = \{ v_k \}_{1 \le k \le n_V}$, où n_V est le nombre de symboles observables v_k,
- $B = \{ b_j(k) \}_{\substack{1 \le k \le n_V \\ 1 \le j \le N}}$, les probabilités d'émission du symbole v_k après passage par l'état j, $b_j(k) = P(o_t = v_k / q_t = j)$

Certains des paramètres définis précédemment sont fixes. En effet, le nombre d'états, les transitions autorisées entre états et l'alphabet émis doivent être fixés et déterminés avant l'apprentissage des modèles. Le nombre d'états dépend généralement de la durée moyenne de l'unité acoustique considérée. L'alphabet est bien souvent déjà connu car

fonction du domaine de l'application (secteur d'activité) et du type de reconnaissance souhaité (mots isolés, phrases courtes, reconnaissance en direct). Les probabilités d'émission $b_j(k)$ sont de natures connues. On suppose, en effet, l'utilisation de gaussiennes dans l'espace R^d. Les transitions autorisées entre états sont suggérées par le modèle de Bakis [13] qui est le modèle habituellement utilisé en reconnaissance de la parole. Celui-ci permet des sauts entre états modélisant ainsi la pluralité et la variabilité de la parole humaine où notamment, le rythme et la vitesse d'élocution diffère d'un locuteur à l'autre. La figure suivante illustre le fonctionnement d'un HMM régit par le modèle de Bakis. On peut remarquer que les retours arrière sont interdits pour respecter l'aspect temporel, que les sauts entre états sont représentés, et enfin que l'automate commence et termine par des états non émetteurs par lesquels il faut obligatoirement transiter.

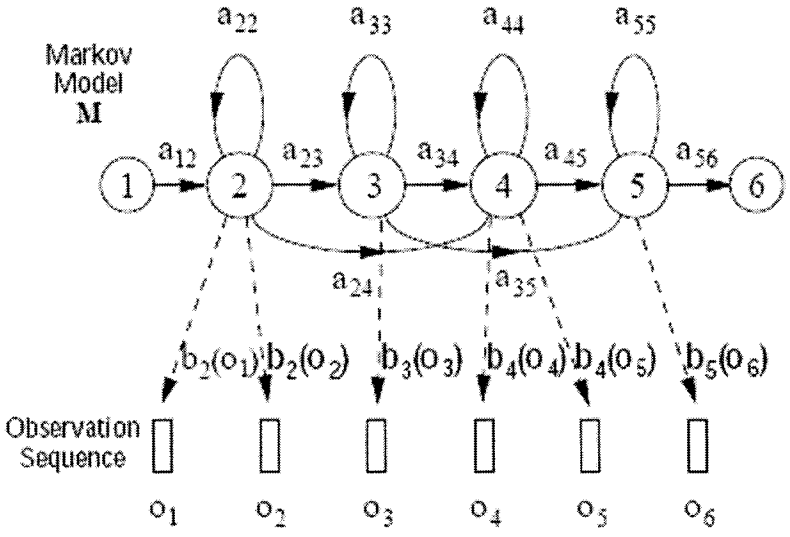

Figure 5 Exemple de processus de Markov [10]

Un HMM modélise généralement l'unité acoustique la plus couramment utilisée en RAP : le phonème. On modélise ainsi le son constituant le phonème par une suite d'événements élémentaires (les états composites de l'automate).

La segmentation de la parole en unités acoustiques, les phonèmes, permet la reconnaissance élargie à de grands vocabulaires. Au lieu de considérer chaque mot individuellement, il vaut mieux considérer les éléments acoustiques qui le constituent. En effet, sachant que le nombre de phonèmes est limité dans chaque langue (quelques dizaines), il sera plus facile d'apprendre le système avec des phonèmes qu'avec les quelques milliers de mots que recense un dictionnaire. Cette approche représente une première solution vers la reconnaissance de la parole continue.

Nous verrons par la suite qu'il est possible d'ajouter des éléments contextuels aux phonèmes pour constituer des automates plus robustes. Ainsi, un allophone ou triphone est constitué d'un phonème et de son contexte gauche (phonème prononcé antérieurement) et de son contexte droit (phonème prononcé après).

1.5 Les outils statistiques de la reconnaissance de la parole

Reconnaître une unité acoustique ou lexicale, c'est chercher lequel de nos modèles de HMM est le plus probable à émettre ce mot. Cela sous-entend de savoir la probabilité d'émission de ce mot pour chaque modèle. Pour ce faire, on utilise une base d'apprentissage qui va nous servir de connaissance à priori. Une fois nos modèles entraînés, on peut passer à la phase de reconnaissance.

La phase d'entraînement ou d'apprentissage consiste à estimer les diverses probabilités de transition et d'émission de nos modèles à partir de l'ensemble des versions enregistrées de ce mot. Il existe deux grandes méthodes pour entraîner et reconnaître un modèle [13], la méthode dite du maximum de vraisemblance (*Maximum Likelihood Estimation* ou MLE) basée sur la notion de probabilité totale et réalisée par l'algorithme de Baum-Welsh [16] et celle dite de l'algorithme de Viterbi [17] basée sur la notion de probabilité maximale. Généralement, la procédure de Baum-Welsh est préférée pour

l'entraînement (moins sensible à l'établissement des conditions initiales) alors que celle de Viterbi est préférée pour la phase de reconnaissance (car elle seule permet de connaître la suite des états parcourus).

1.5.1 Entraînement d'un modèle : Méthode du maximum de vraisemblance

On veut entraîner le modèle M associé à un mot X à partir des versions connues de ce mot $\{X_k\}$ contenues dans la base d'apprentissage.

On définit les probabilités suivantes :

- $P(M/X_k)$, la vraisemblance du modèle M à produire la version X_k.
- $P(X_k/M)$, la probabilité d'obtenir la version X_k avec le modèle M.

L'entraînement du modèle M revient à modifier ses paramètres pour rendre maximum le produit :

$$P = \prod_{k=1}^{K} P(M/X_k) \tag{0.5}$$

Selon le théorème de Bayes :

$$P(M/X_k) \cdot P(X_k) = P(X_k/M) \cdot P(M) \tag{0.6}$$

Où $P(X_k)$ et $P(M)$ sont des probabilités *a priori*. Optimiser l'équation (1.5) revient finalement à trouver les paramètres de M pour rendre maximale la probabilité d'émission par le modèle M de l'ensemble des observations X_k du corpus d'apprentissage:

$$\hat{M} = \underset{M}{ArgMax}\left[\prod_{X_k} P(X_k/M)\right] \tag{0.7}$$

Malheureusement, la résolution directe de l'équation précédente n'est pas possible car trop coûteuse en temps de calculs. Pour ce faire, nous avons recours à l'algorithme de Baum-Welsh (*forward-backward procedure*) qui permet une réestimation par récurrence des paramètres $\{ a_{ij} \}_{1 \le i,j \le N}$ (transitions entre états) et $\{ b_j(k) \}_{\substack{1 \le k \le n_V \\ 1 \le j \le N}}$ (probabilités d'émissions). Une démonstration détaillée de l'algorithme *forward-backward* se trouve dans [13].

À chaque nouvelle version du mot X, le nouveau modèle M' vérifie l'amélioration de la vraisemblance:

$$P(X/M') \ge P(X/M) \tag{0.8}$$

Cette procédure doit être reprise pour une (ou plusieurs) nouvelle itération à partir de toutes les versions du mot X jusqu'à ce que les paramètres du modèle M soient stabilisés. La convergence est démontrée en [18].

Les performances de la réestimation de Baum-Welsh sont fortement conditionnées par les conditions initiales (initialisation de la récurrence) associées à chacun des modèles. Plusieurs méthodes existent comme l'initialisation par alignement acoustique et phonétique ou encore par l'initialisation par moyenne globale. L'initialisation par moyenne globale consiste à calculer les moyennes et les variances des vecteurs de paramètres de façon globale sur tout l'ensemble des données d'apprentissage. De cette façon, les probabilités de transition et les probabilités d'émission de tous les modèles de Markov sont initialisées de la même façon. L'initialisation par alignement acoustique utilise l'algorithme de Viterbi pour aligner les trames d'observations au modèle et recalculer en conséquence les probabilités.

1.5.2 La reconnaissance d'un modèle : Méthode de l'algorithme de Viterbi

Prenons comme point de départ un vecteur d'observations que l'on souhaite reconnaître. Les paramètres précédents définissant nos HMM ont été évalués durant la phase d'apprentissage. La phase de reconnaissance se doit d'évaluer les grandeurs suivantes pour chaque HMM du système :

- La probabilité que l'observation à reconnaître puisse avoir été émise par le HMM. On retiendra le modèle présentant la plus forte probabilité d'émission.
- La recherche de la séquence optimale d'états d'un modèle ayant produit les observations.

1.5.2.1 Probabilité d'émission des observations

La probabilité qu'une suite d'observations $O = (o_1, ..., o_T)$ soit émise par une machine M n'est pas directement calculable, car la séquence d'états parcourus est cachée. On y parvient en calculant toutes les probabilités conjointes de O avec un chemin Q appartenant à l'ensemble E_Q des chemins possibles de longueur T.

Ainsi la probabilité que le modèle M ait émis l'observation O s'écrit :

$$P(O/M) = \sum_{Q \in E_Q} P(O,Q/M) \tag{0.9}$$

On sait que

$$P(O,Q/M) = P(Q/M) \cdot P(O/Q,M) \tag{0.10}$$

Avec

$$P(Q/M) = \prod_{t=1}^{T} a_{q_{t-1}\, q_t} \tag{0.11}$$

et

$$P(O/Q,M)=\prod_{t=1}^{T} b_{q_t}(o_t) \tag{0.12}$$

D'où

$$P(O/M)=\sum_{Q\in E_Q}\prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \tag{0.13}$$

Finalement, le modèle $\tilde{M}$ retenu parmi l'ensemble E_M des modèles est celui qui satisfait l'équation suivante :

$$\tilde{M}=\underset{M\in E_M}{ArgMax}\, P(O/M)\cdot P(M) \tag{0.14}$$

Toutefois, avec une machine à N états, le nombre de chemins possibles est de l'ordre de N^T, et l'ensemble des chemins E_Q devient très rapidement impossible à décrire. Par exemple, pour un modèle simple à 5 états et 100 observations à reconnaître, le nombre de calculs nécessaires s'élèverait à 10^{72}! L'algorithme de Viterbi propose une méthode itérative qui, à chaque nouvelle observation, va calculer la probabilité conjointe de l'équation (1.10) maximale obtenue à ce stade.

1.5.2.2 Algorithme de Viterbi

L'algorithme de Viterbi permet de trouver le chemin le plus probable qui a pu produire l'observation à reconnaître. L'algorithme avance en négligeant les chemins les moins probables.

On définit la grandeur $\Phi(t,i)$ comme étant la probabilité maximale que les observations observées jusqu'à l'instant t aient été émises par le modèle M en suivant un chemin qui arrive à l'état i :

$$\Phi(t,i) = \max_{q_1 \cdots q_{t-1}} P(o_1 \ldots o_t, q_1 \ldots q_{t-1} q_t = i/M) \tag{0.15}$$

Itérativement, à chaque unité de temps, on recalcule cette grandeur parmi toutes les transitions possibles partant de l'état *i*. On retient la plus grande tout en conservant en mémoire l'état précédent.

Ainsi on a

$$\Phi(t,j) = \max_{1 \le i \le N} \left\{ \Phi(\text{t-1,i}) \cdot a_{ij} \cdot b_j(o_t) \right\} \tag{0.16}$$

Si on dispose de T observations à reconnaître, l'étape finale est obtenue après le calcul de $\Phi(t,N)$ où N est l'état final.

On obtient ainsi

$$P(O,Q/M) = \Phi(T,N) \tag{0.17}$$

La probabilité d'émission sur le meilleur chemin peut être utilisée pour la reconnaissance comme une approximation de la probabilité d'émission par le modèle mais cette méthode de résolution est sous-optimale puisqu'elle néglige les chemins de plus faible probabilité.

1.6 La reconnaissance de la parole continue

L'apprentissage tel que décrit à la section précédente utilise chaque mot de la base d'entraînement pour estimer le modèle de HMM lié à ce mot. Or, dans les systèmes de reconnaissance où les modèles représentent des unités acoustiques comme le phonème (reconnaissance de la parole continue à grand vocabulaire), il faudrait pouvoir isoler chaque phonème du corpus d'apprentissage pour entraîner le modèle s'y rapportant.

Bien entendu, cette opération n'est pas concevable puisqu'il n est pas possible de prononcer un phonème de manière isolée. Heureusement, un apprentissage des modèles sans nécessité de segmentation est possible en connectant les HMM bout à bout. C'est l'objet du paragraphe suivant.

1.6.1 Modèles connectés

Soit une phrase Φ dont la transcription phonétique est connue: $\Phi = (\varphi_1 .....\varphi_T)$. On construit un modèle $\Sigma = (\varepsilon_1 .....\varepsilon_T)$ de la phrase Φ où ε_t représente le phonème φ_t. Le modèle de la phrase est crée en reliant l'état final du model ε_t à l'état initial du modèle ε_{t+1}.

Le supermodèle ainsi créé peut être réestimé (sur la phrase tout entière) avec la procédure décrite au paragraphe 1.5.1 . Celui qui obtient la plus forte probabilité donne la phrase reconnue. Enfin, la présence dans chaque modèle d'états non-émetteurs aux extrémités sert notamment à faire la jonction d'un mot à l'autre.

1.6.2 Apprentissage

L'apprentissage de mots connectés utilise la même procédure de Baum-Welsh dans le cas de mots isolés vue au paragraphe 1.5.1, à la différence que, au lieu d'entraîner chaque modèle individuellement, ici tous les modèles sont entraînés en parallèle. La procédure suit les étapes suivantes :

1) Prendre une phrase du corpus d'apprentissage.
2) Construire un supermodèle composite de HMM en joignant en séquence les HMM correspondant à la transcription de la phrase d'entraînement.

3) Calculer les probabilités *forward* et *backward* pour le modèle créé. À noter que l'inclusion dans le supermodèle d'états non émetteurs requiert quelques modifications mineures dans les algorithmes.
4) Utiliser les probabilités calculées en 3) pour calculer les probabilités d'occupation d'états à chaque unité de temps.
5) Répéter le processus jusqu'à ce que toutes les phrases aient été traitées.

1.6.3 Décodage

La méthode de reconnaissance vue au paragraphe 1.5.2 est surtout valable pour la reconnaissance de mots isolés où la démarche de tester chacun des mots possibles est encore possible. Ici, tester toutes les phrases possibles par enchainement de mots est impensable.

Le chemin intra-modèle le plus probable est déterminé itérativement étape par étape parmi toutes les transitions possibles. Comme le chemin inter-modèle ne peut être choisi en testant tous les chemins possibles, on joint à l'algorithme de Viterbi un réseau de mots qui va simplifier le nombre de transitions possibles en respectant la syntaxe définie. Il va ainsi interconnecter tous les modèles syntaxiquement correctes.

Le réseau de mots est un arbre avec des nœuds, des branches, des étiquettes qui décrit les combinaisons possibles à priori de mots (ou de l'unité lexicale utilisée) à partir du dictionnaire des unités utilisées. La suite d'états optimale trouvée par l'algorithme de Viterbi dans ce réseau fournit un décodage de la phrase en mots ou en phonèmes, ainsi qu'une segmentation du signal acoustique.

Cependant, l'algorithme recherche la meilleure suite d'états dans le réseau et non pas la meilleure suite de modèles. En effet, toutes les phrases syntaxiquement correctes

admises par le réseau sont à priori équiprobables ce qui n'est pas le cas dans la réalité où des phrases apparaissent plus souvent que d'autres. Nous verrons au chapitre consacré à la construction de notre système de reconnaissance, une technique permettant d'identifier les suites de phonèmes les plus probables d'un point de vue syntaxique.

CHAPITRE 2

ÉTUDE DU CANAL TÉLÉPHONIQUE

Nous venons de voir au chapitre précédent quelques techniques utilisées pour reconnaître la parole humaine. Mais nous n'avons pas encore évoqué comment ces techniques œuvrent dans des conditions réelles où le signal vocal comporte bien souvent autre chose que de la voix. En effet, le bruit est à considérer pour parfaire un système de reconnaissance ayant pour but une utilisation commerciale car le signal téléphonique ne possède pas les mêmes caractéristiques que le signal d'origine. Les conditions d'acquisition et de transmission du signal de parole vont entraîner de grands changements entre l'apprentissage et le test. La parole transmise par le réseau PSTN présente de nombreux bruits qui viennent dégrader les taux de reconnaissance. Quels sont ces bruits ? Comment y faire face dans un système de reconnaissance ?

2.1 Bruits liés à l'acquisition et au transport de la parole au travers du réseau téléphonique commuté

Principalement, 2 types de bruits interviennent depuis l'émetteur jusqu'au récepteur : les bruits additifs et les bruits convolutifs. Il faut ajouter également comme sources de distorsion les éléments non linéaires apportés par les équipements du réseau [19]. Ces derniers, très difficiles à corriger et généralement assez faibles, sont rarement pris en compte dans le cadre de l'étude du canal téléphonique.

2.1.1 Les bruits additifs

Les bruits additifs ne sont pas propres au canal téléphonique en lui-même. Ils sont engendrés principalement par les conditions d'acquisitions de la voix au niveau de l'émetteur. On dit de ces bruits qu'ils sont additifs car le signal enregistré est la somme du signal vocal et du bruit. Ainsi, le bruit d'ambiance (voiture en marche, bruit de foules) est considéré comme du bruit additif.

2.1.2 Les bruits convolutifs

Les bruits convolutifs viennent perturber le signal original au niveau de ses propriétés spectrales. On peut citer les sources suivantes :

- Le microphone apporte son lot de distorsions suivant la marque du microphone utilisé (bonne au mauvaise qualité) mais aussi suivant la position la personne en train de parler compte tenu de la directivité du microphone (omnidirectionnel, cardioïde, bidirectionnel, hyper cardioïde, canon, [20]).

- Les câbles téléphoniques et les commutateurs dégradent le signal. La restriction de la largeur de bande par le canal vient considérablement perturber le signal à transmettre et les propriétés fréquentielles de la parole. Ce dernier point fait l'objet d'une étude approfondie au paragraphe 2.2.

- Enfin, il ne faut pas oublier les phénomènes d'échos et de réverbération sur les lignes téléphoniques qui modifient également le spectre du signal.

Le schéma général de distorsion d'un signal téléphonique est illustré à la Figure 6.

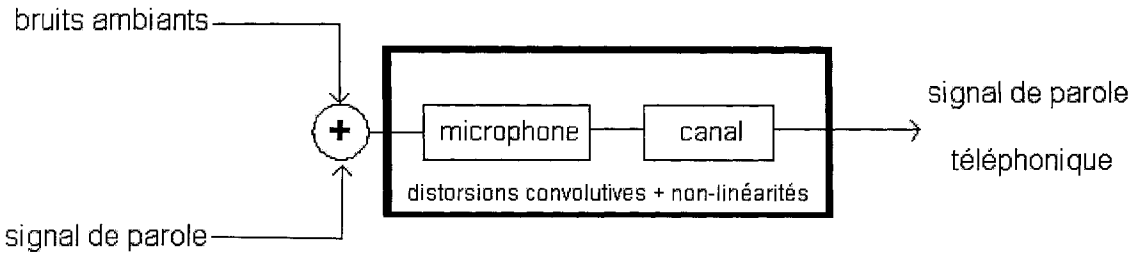

Figure 6 Les différents bruits constituants du signal téléphonique

2.2 Effets du canal téléphonique sur la parole

Le spectre d'un signal de parole est constitué de composantes fréquentielles très spécifiques. Outre la fréquence fondamentale F_0 (l'énergie) correspondant à la fréquence de vibration des cordes vocales, les résonnances du conduit vocal (cavité pharyngale, cavité buccale, positions des lèvres) font apparaître des harmoniques renforcés aux fréquences F_i, ce sont les formants (Figure 7). Ils sont numérotés de F_1 (premier formant) jusqu'à F_5 (cinquième formant). Par exemple, le fondamental F_0 se situe entre 100 et 150 Hz pour les hommes, 140 à 240 Hz pour les femmes. Quant à F_1, il se situe entre 270 et 730 Hz pour les hommes, entre 310 et 850 Hz pour les femmes [9].

Le spectre de la parole humaine peut s'étendre jusqu'à 12 kHz. La bande utile du téléphone [300 Hz, 3400 Hz] ne peut donc contenir tout le spectre et la totalité des formants. Elle permet seulement de faire passer les 3 premiers formants avec comme conséquence la dégradation du signal original. On aura ainsi de la difficulté à distinguer les sons (prononcés de manière isolée au cours d'une conversation téléphonique) puisant leur énergie (non exclusivement) au delà de 3 premiers formants comme par exemple les fricatives /s/ ou /f/. Toutefois, l'intelligibilité demeure tout à fait acceptable car [21] démontre qu'elle est portée par les premiers formants, plus précisément, que la sensation naturelle de la voix se situe au premier formant et l'intelligibilité au second formant.

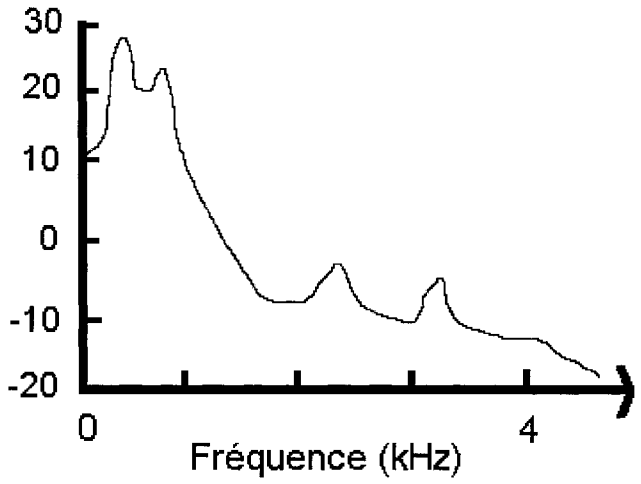

Figure 7 Observation du fondamental et des formants avec la représentation spectrale de la voyelle /o/ du mot "obey"

En conclusion, l'effet de la restriction de la largeur de bande par le canal sur la parole humaine a comme conséquences la non transmission de la fréquence fondamentale et des hautes fréquences ainsi qu'une atténuation du premier formant (pour les hommes surtout).

2.3 Effets du canal en terme de performances de reconnaissance

Une des conditions nécessaires pour avoir des bons scores de reconnaissance demeure l'existence d'une certaine similarité entre le corpus d'apprentissage et les données de test effectivement rencontrées. Ainsi, pour une application de reconnaissance vocale via le téléphone, il est clair que si la phase d'apprentissage est réalisée sur une base de données propres, le taux de reconnaissance sera assurément bas à cause des nombreuses dégradations qu'apportent l'acquisition et le cheminement du signal. Il y aurait, dans ce cas, ce qu'on appelle un phénomène de non-appariement des données d'apprentissage avec celles de test (de l'anglais *unmatching*).

Moreno et Stern [22] ont montré qu'un système de reconnaissance appris et testé sur TIMIT (voir description au Chapitre 4) atteint 52.7 % de phonèmes reconnus alors que s'ils renouvellent l'expérience en apprenant le système sur TIMIT et en le testant sur NTIMIT (le corpus TIMIT passé via le canal téléphonique), le taux chute à 31.3%. Une diminution des performances est systématiquement observée lorsque la base de test est enregistré dans des conditions d'environnement différentes du corpus d'entrainement [23, 24].

Il semble donc important d'élaborer des techniques visant à compenser l'effet du canal sur les données du corpus de test pour que l'appariement avec les données d'apprentissage soit meilleur.

2.4 Techniques d'égalisation du canal PSTN

Le paragraphe précédent a mis en lumière la nécessité d'apporter une correction aux systèmes de reconnaissance de la parole bruitée. D'une règle générale, les méthodes visant à réduire le non appariement des données d'entraînement avec celles de test peuvent se classer en 2 catégories [25] :

Adaptation des paramètres

La correction apportée va donner plus de robustesse aux paramètres (les coefficients cepstraux ici en l'occurrence) corrompus par le système de transmission de la voix sur le réseau commuté. Il ne s'agit pas de refaire toute l'analyse acoustique en trouvant une nouvelle transformation pour représenter autrement les échantillons du signal vocal. Le but est de garder les mêmes paramètres mais de les modifier grâce aux connaissances *a priori* acquises sur le corpus d'apprentissage ou sur le canal proprement dit. On peut

citer par exemple la technique d'égalisation cepstrale [26] ou la technique de filtrage RASTA [27].

Adaptation des modèles

La correction s'opère ici non plus au niveau des données mais sur les modèles de HMM et leurs constituants. Les lois d'observations caractérisant les états d'un modèle ont tendance à ne plus respecter une loi gaussienne lorsque le bruit devient trop fort. Ainsi, on se sert de corpus d'apprentissage bruité pour apprendre les modèles (Apprentissage multiréférences [28]) ou bien encore d'imaginer une double représentation pour chaque modèle, une pour le signal propre, une pour le bruit (Décomposition des modèles [29]) .

Rappelons que le sujet de cette maîtrise se cantonne à l'étude des méthodes pour réduire l'effet convolutif du canal téléphonique nuisible aux performances d'un système de RAP. Nous nous attacherons donc à ne décrire que des techniques compensatoires du bruit convolutif. Le lecteur pourra se référer à [22, 29-31] pour de plus amples renseignements sur les méthodes de correction des bruits additifs et à [32, 33] pour les bruits relatifs aux microphones.
Les techniques développées et implémentées dans le cadre ce mémoire concernent en majorité (sauf mention) la première catégorie (adaptation des paramètres), elles sont détaillées dans les sections suivantes.

2.4.1 La normalisation par moyenne cepstrale (CMN)

La soustraction cepstrale, de l'anglais *Cepstral Subtraction*, est une technique courante pour compenser l'effet convolutif du canal dans un processus de reconnaissance. Dans sa forme la plus simple, la technique de normalisation (ou égalisation) par moyenne

cepstrale (*Cesptral Mean Normalization*) requiert un coût de calcul quasiment négligeable et procure des résultats tout à fait acceptables.

En sortie du canal de réponse impulsionnelle *h(t)*, l'entrée *x(t)* et la sortie *y(t)* sont liées par la relation suivante :

$$y(t) = [x(t) + n(t)] \otimes h(t) \tag{2.1}$$

Où $\otimes$ dénote le produit de convolution et *n(t)* est un bruit ambiant additif.
On pose *s(t)*=[*x(t)* + *n(t)*], en effet, dans le cadre de cette maîtrise, la compensation des bruits additifs ne sera pas étudiée. On ne s'intéressera donc pas à annihiler les effets de *n(t)*. On considérera ainsi *s(t)* comme notre signal d'entrée, et non pas *x(t)*.

Dans le domaine spectral,

$$Y(\omega) = S(\omega) \cdot |H(\omega)|^2 \tag{2.2}$$

Où *Y(ω)* est le spectre de puissance du signal *y(t)* et *H(ω)* la fonction de transfert du canal téléphonique.

Après passage au logarithme et dans le domaine cepstral, il vient :

$$C_y(\tau) = C_s(\tau) + C_h(\tau) \text{ avec } \tau \in [1;T] \tag{2.3}$$

Où *T* est le nombre de trames de l'enregistrement.

Ainsi, si l'on parvient à évaluer correctement une estimation du cepstre du canal $C_h(\tau)$, on va soustraire cet estimé au cepstre du signal de sortie $C_y(\tau)$, d'où le nom de soustraction cepstrale.

Considérons un intervalle de temps Δt durant lequel la réponse impulsionnelle du canal *h(t)* peut être vue comme un filtre à fonction de transfert linéaire et invariante sur Δt. On dit d'un signal *s(n)* qu'il est cohérent après *L* échantillons si 2 trames séparées de *L*

échantillons sont corrélées [34]. Le corollaire dit qu'un signal *s(n)* est incohérent si 2 trames séparées par $K > L$ échantillons sont décorrélées.

Selon Mokbel [34], si on considère un signal *x(t)* tel que $y(t) = s(t) \otimes h(t)$ où *h(t)* est la fonction de transfert linéaire et constante du canal sur Δt, il est démontré que la moyenne des vecteurs cepstraux sur Δt représente le cepstre du canal si le signal *y(t)* est non cohérent après $\frac{\Delta t}{2} \cdot f_e$ échantillons.

En pratique, la fonction de transfert du canal n'est pas constante mais varie lentement dans le temps.
Théoriquement, Δt devrait être choisi tel que couvrant les zones de parole et évitant les zones de silence. Ainsi le meilleur estimé est celui qui utilise toutes les portions de parole de la communication téléphonique pour recouvrir un nombre important de phonèmes.

En effet, [35] montre que les résultats avec Δt pris exclusivement sur les zones de parole sont supérieurs à ceux obtenus à partir d'une estimation basée sur toute la trame de communication.

Dans un premier temps, on se contente d'implémenter la technique de normalisation cepstrale (CMN), la plus simple, celle qui considère Δt comme la durée de tout le signal de parole, voix et silences compris. On parle de moyenne long-terme.

Ainsi, une estimation du cepstre du canal est :

$$C_h(\tau) = \frac{1}{T} \sum_{t=1}^{T} C_y(t) \tag{2.4}$$

Où T est le nombre de trames de l'enregistrement.

2.4.2 La normalisation par moyenne cepstrale améliorée (CMN Best)

Le problème avec la technique CMN est qu'elle ne fait pas la différence entre les sons voisées et les zones de silence souvent recouverts par le bruit et pris en compte dans l'estimé du cepstre de *h(t)*. Mokbel [34] a montré que la meilleure estimation pour $C_h(\tau)$ est la moyenne des vecteurs cepstraux des trames de parole.

Nous allons préalablement décrire l'algorithme qui a été utilisé pour concevoir un détecteur d'activité de la voix qui permettra de savoir si chaque trame constituante du signal vocal est de la parole ou du silence.

2.4.2.1 Algorithme utilisé de détection d'activité de la voix

Il existe de nombreuses méthodes pour détecter l'activité de la voix [36] [37]. Dans le cadre de ce projet où l'on s'intéresse exclusivement aux bruits convolutifs, on peut aisément supposer que le SNR des phrases bruitées par notre modélisation du canal est suffisamment élevé pour considérer des techniques simples de détection de la voix.

La technique mise en œuvre ici est inspirée de l'étude de Prasad & al [38] et considère l'énergie de la trame comme critère de discrimination. Toutes les trames dont l'énergie sera supérieure à un certain seuil seront considérées comme de la parole, celles dont l'énergie ne le dépasse pas seront considérées comme du silence.

2.4.2.1.1 Energie d'une trame

Soit $S = \{s_i\}$, le signal de parole échantillonné répartit en trames de k échantillons.
Soit M le nombre de trames dans le signal considéré.

On note E_i, $0 \leq i \leq M$ l'énergie de la trame i définie par :

$$E_i = \sum_{j=(i-1)k+1}^{ik} s_j^2 \tag{2.5}$$

2.4.2.1.2 Méthode simple de détection

Valeur du seuil discriminant

Soit E_r le seuil discriminant. En première estimation, on peut considérer E_r comme la moyenne des énergies calculées uniquement sur les trames de bruit présent au tout début du signal de parole.

Ainsi, si on dénote υ le nombre de trames au début du signal, E_r a pour expression :

$$E_r = \frac{1}{\upsilon} \cdot \sum_{m=0}^{\upsilon} E_m \tag{2.6}$$

Dans l'expérimentation, sachant que les 150 premières millisecondes des signaux de parole TIMIT ne présentent aucune phase d'activité, et sachant que 10 ms est la durée effective des trames, υ fut donc choisi égal à 15.

Prise de décision

Soit α est un réel supérieur à 1.

Pour chaque trame i du signal,

Si $E_i > \alpha \cdot E_r$

Alors

i est une trame de parole

Sinon

i est une trame de silence

Fin

Sur la Figure 8, on peut voir un exemple de prise de décision simple. Tout ce qui est au dessus du seuil est considéré comme de la parole, tout ce qui en dessous est vu comme du silence.

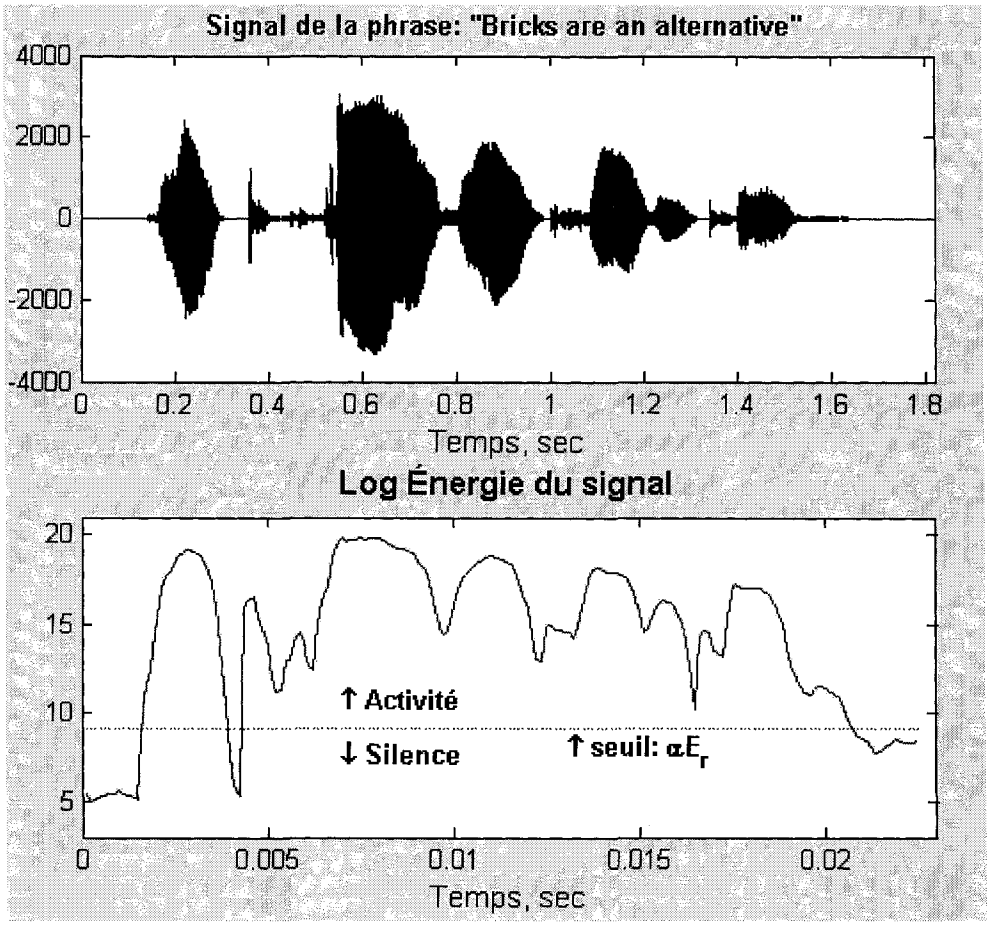

Figure 8 Exemple de détection simple d'activité

Toutefois, cette méthode avec son seuil fixé à chaque nouvel enregistrement n'est pas satisfaisante. En effet, l'énergie obtenue sur les 150 premières millisecondes n'est pas forcement très adéquate pour établir le seuil car son calcul est indépendant de l'énergie du reste du signal. Il peut donc arriver dans certains cas, que le seuil de décision αE_r soit trop élevé et que la discrimination ainsi réalisée soit complètement faussée (Figure 9).

Nous allons améliorer la prise de décision en réévaluant le seuil à chaque nouvelle trame de silence rencontrée. C'est l'objet du paragraphe suivant.

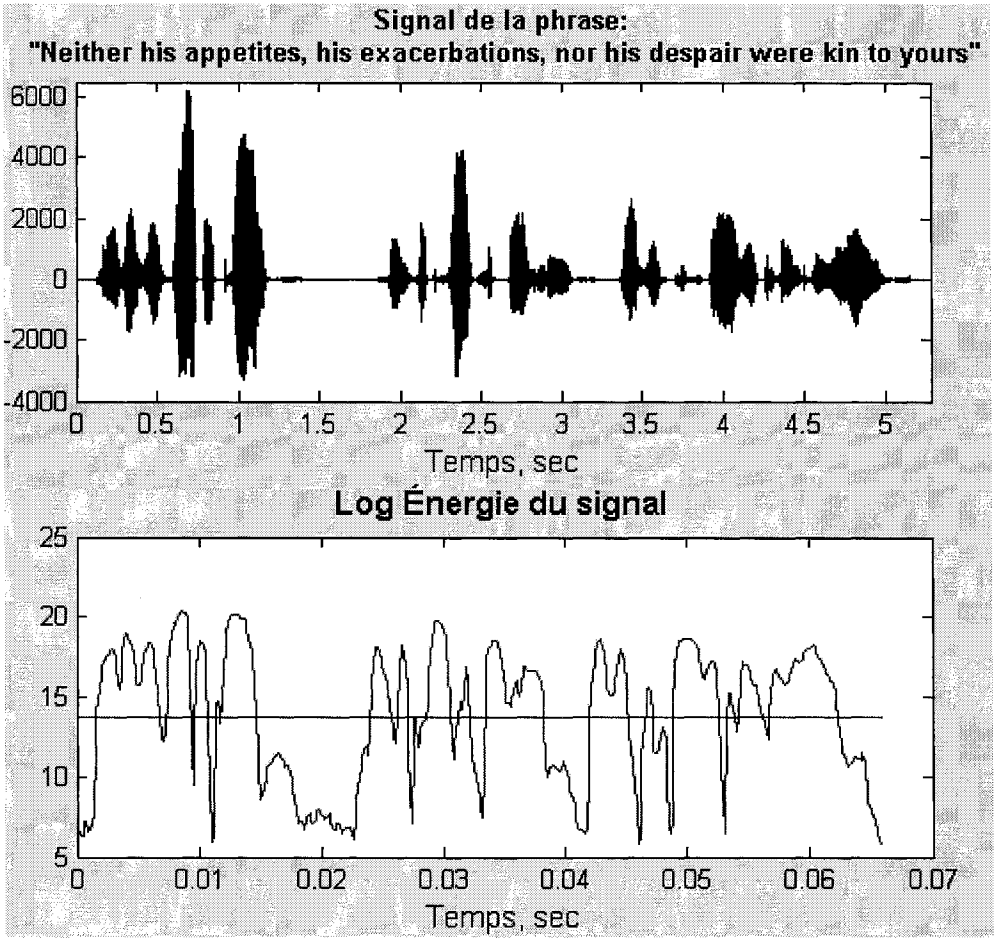

Figure 9 Problème de la détection simple d'activité

2.4.2.1.3 Méthode plus élaborée de détection

Toujours développée dans [38], l'idée consiste à modifier le seuil E_r (calculé précédemment) à chaque fois que l'on rencontre une trame de silence. Voici l'algorithme à considérer.

On note E_r^{NEW}, la nouvelle valeur du seuil.

Prise de décision

Initialisation : $E_r^{NEW} = E_r$

Pour chaque trame *i* du signal,

Si $E_i > \alpha \cdot E_r^{NEW}$

Alors

i est une trame de parole

Sinon

i est une trame de silence

$$E_r^{NEW} = \sigma \cdot E_r^{NEW} + (1-\sigma) \cdot E_i \tag{2.7}$$

Fin

La constante σ est choisie en considérant la réponse impulsionnelle de l'équation (2.9) :

$$E_r(z) = (1-\sigma) \cdot z^{-1} \cdot E_r(z) + \sigma \cdot E_i(z) \tag{2.8}$$

La fonction de transfert $H(z)$ vaut alors :

$$H(z) = \frac{E_r(z)}{E_i(z)} = \frac{\sigma}{1-(1-\sigma) \cdot z^{-1}} \tag{2.9}$$

La constante σ est choisie de manière à influencer le seuil uniquement si l'on est réellement dans une zone de silence. Prasad et al montrent que pour $\sigma = 0.2$, *H(z)* atteint 5% de sa valeur maximum (*fall-time* - 95%) pour un retard de 15 trames, i.e. 150 ms. Donc, 15 trames de silence consécutives vont influencer le calcul de E_r^{NEW}. En effet, sachant que le temps de pause entre 2 syllabes est généralement de l'ordre de 100 ms et que ces pauses n'ont pas à être considérées comme des silences, le *fall-time* doit être choisi au delà des 100 ms. Nous fixerons donc $\sigma = 0.2$.

Sur la Figure 10, nous avons appliqué l'algorithme sur l'enregistrement qui présentait des difficultés au paragraphe précédent. Avec la nouvelle technique, le seuil a été revu à la baisse pour mieux discriminer les zones de silence des zones de parole.

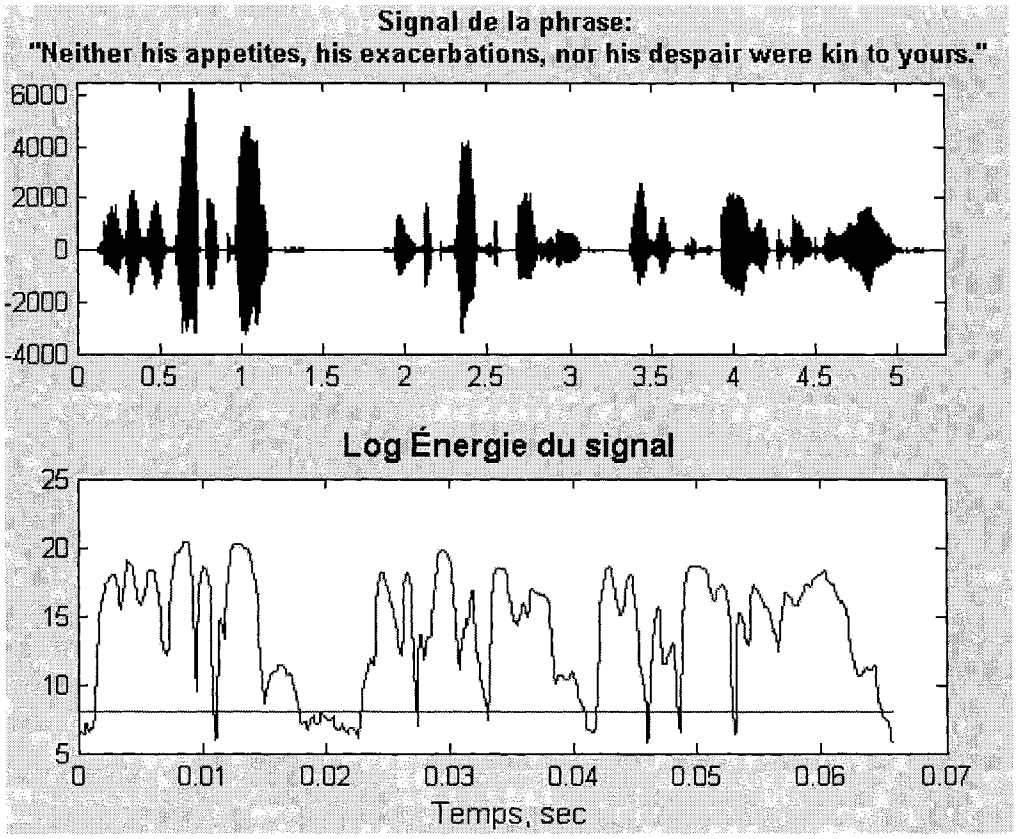

Figure 10 Exemple de détection plus élaborée d'activité

2.4.2.2 Le calcul de la nouvelle valeur du cepstre du canal

Le VAD mis en place à la section précédente nous permet de calculer la nouvelle valeur de l'estimé du cepstre du canal :

$$C_h^{BEST}(\tau) = \frac{1}{card(Parole)} \sum_{t \in Parole} C_y(t) \tag{2.10}$$

Nous appellerons « CMN Best » ce cas particulier de normalisation cepstrale.

2.4.3 La normalisation cepstrale augmentée (ACN)

Pour améliorer les performances de la technique précédente, Acero et Huang ont proposé une technique qui calcule des moyennes différentes selon que la trame est un silence ou un son voisé [39]. Cette technique s'appelle : *Augmented Cepstral Normalization* (ACN).

Soit,

- $X = \{x_i\}$, $0 \leq i \leq N$, les vecteurs cepstraux d'une phrase donnée.
- η le vecteur cepstral moyen calculé sur les trames de bruit (silence) de la phrase X.
- ς le vecteur cepstral moyen calculé sur les trames de parole de la phrase X.
- η_{AVG} le vecteur cepstral moyen calculé sur les trames de bruit (silence) de tout le corpus d'apprentissage.
- ς_{AVG} le vecteur cepstral moyen calculé sur les trames de parole de tout le corpus d'apprentissage.
- $p_i = P(Silence \ / \ x_i)$, $0 \leq i \leq N$, est la probabilité *a posteriori* que la trame i soit une trame de bruit.

La technique proposée consiste à soustraire à chaque vecteur cepstral x_i un vecteur correctif r_i.
En appelant z_i le nouveau vecteur cepstral, l'algorithme s'écrit ainsi, pour $0 \leq i \leq N$:

$$z_i = x_i - r_i \tag{2.11}$$

$$\text{avec } r_i = p_i \cdot (\eta - \eta_{AVG}) + (1 - p_i) \cdot (\varsigma - \varsigma_{AVG})$$

Toute la difficulté de cette méthode réside, d'une part, dans le processus de discrimination de la phrase en trames de parole et trames de silence. Pour ce faire, on

aura recours à l'algorithme de détection de voix décrit au paragraphe 2.4.2.1. D'autre part, le calcul de la probabilité *a posteriori* p_i nécessite aussi une attention particulière.

2.4.3.1 Calcul de la probabilité *a posteriori* p_i

$p_i = P(silence / x_i)$, $0 \leq i \leq N$, est la probabilité *a posteriori* que la trame i soit une trame de silence.

2.4.3.1.1 Estimation discrète

Tel que dans la partie précédente, l'estimation discrète repose sur le principe suivant : on définit un seuil fixé, si l'énergie de la trame est supérieure au seuil, alors elle est considérée comme de la parole et $p_i = 0$, sinon, la trame est considérée comme un silence et $p_i = 1$.

2.4.3.1.2 Estimation continue

Le théorème de Bayes assure que

$$P(silence / x_i) = \frac{P(silence) \cdot P(x_i / silence)}{P(silence) \cdot P(x_i / silence) + P(parole) \cdot P(x_i / parole)} \quad (2.12)$$

Les probabilités *a priori* $P(silence)$ et $P(parole)$ sont obtenues sur la base d'apprentissage en utilisant l'algorithme décrit précédemment dans sa version améliorée. Pour idée, nous avons obtenu les probabilités suivantes :

$$P(silence) = 0.1392$$

$$P(parole) = 0.8608$$

Les probabilités $P(x_i / speech)$ et $P(x_i / silence)$ sont supposées avoir une loi gaussienne, elles sont estimées par l'outil de reconnaissance HTK.

Pour les expérimentations, nous avons considéré la probabilité *a posteriori* p_i dans le cas de l'estimation discrète.

2.4.4 La méthode du filtrage RASTA

Il s'agit d'une autre technique de normalisation des coefficients cepstraux. Cette méthode a été développée par Hermansky dans [27]. Il y est montré que la méthode RASTA, pour **R**el**A**tive Spec**TrA**l, permet d'améliorer les performances de reconnaissance en présence de bruits additifs et convolutifs.

L'idée est d'éliminer au moyen d'un filtrage passe-haut toutes variations des coefficients cepstraux trop lentes pour être dues aux variations du signal de parole.
Le filtre passe-haut est de la forme :

$$H(z) = \frac{1 - z^{-1}}{1 - C \cdot z^{-1}} \tag{2.13}$$

La constante C définit la constante de temps du filtre RASTA. Cette constante doit être telle que les variations moyennes des coefficients cepstraux soient transmises mais que les variations trop lentes soient bloquées. Typiquement, la valeur usuelle est $C = 0.97$.

Appliqué sur les coefficients cepstraux *x(n)* de chaque trame *n*, le filtre peut aussi s'écrire :

$$y(n) = x(n) - x(n-1) + 0.97 \cdot y(n-1) \tag{2.14}$$

CHAPITRE 3

MISE EN PLACE DU SYSTÈME DE RECONNAISSANCE AVEC HTK

La plate-forme HTK, pour ***H**idden Markov Model **T**ool**K**it* ("boite à outils de modèles de Markov cachés") est développée par l'Université de Cambridge par Steeve Young et son équipe [10]. Elle est constituée d'un ensemble de librairies qui permettent notamment de construire des systèmes de reconnaissance de la parole continue à base de modèles de Markov cachés (HMM).

La démarche qui a été adoptée pour entraîner et tester la base de données TIMIT est en grande partie inspirée du tutorial HTK développée au Chapitre 3 du livre *The HTK Book.*

L'environnement de travail requiert idéalement un système d'exploitation UNIX et les fonctions HTK sont appelées par des commandes en langage C-shell. Cygwin [40], un émulateur de Linux sur Windows, a été utilisé pour ce projet.

3.1 Calcul des coefficients mel-cepstraux (MFCC)

À l'aide du fichier de configuration, le module `HCopy` va transformer les enregistrements TIMIT (.wav) en fichiers paramétrés (.mfc). Comme déjà évoqué au Chapitre 1, nous choisissons la transformation la plus utilisée dans le domaine : les coefficients cepstraux MFCC (*Mel-Frequency Cepstral Coefficients*).
Ainsi la commande est la suivante :

```
$ HCopy -S $trainlist -C $config
```

Le fichier `trainlist` contient la liste de tous les enregistrements de la base d'apprentissage. Le fichier de configuration `config` apporte les paramètres nécessaires au calcul des coefficients cepstraux (Figure 11).

```
SOURCEKIND          =  WAVEFORM
SOURCEFORMAT        =  TIMIT
SOURCERATE          =  625
TARGETKIND          =  MFCC_E_D_A
TARGETRATE          =  100000
SAVECOMPRESSED      =  TRUE
SAVEWITHCRC         =  TRUE
WINDOWSIZE          =  250000.0
USEHAMMING          =  TRUE
NUMCHANS            =  26
NUMCEPS             =  12
```

Figure 11 Le fichier de configuration `config`

Les paramètres retenus pour représenter le signal sont désignés par le paramètre `TARGETKING`. Ainsi 12 coefficients MFCC et l'énergie du signal (log-énergie) sont calculés, auxquels sont ajoutées les dérivées premières et secondes de ces coefficients, soit au total 39 coefficients. Chaque tranche de 10 ms (`TARGETRATE`) est échantillonnée à 16 kHz (`SOURCERATE`) après pondération par une fenêtre de Hamming d'une durée de 25 ms (`WINDOWSIZE`). Les 12 (`NUMCEPS`) premiers coefficients cepstraux, obtenus à partir d'un banc de 26 filtres (`NUMCHANS`) en échelle fréquentielle Mel, sont conservés.

3.2 Apprentissage par monophone

Comme déjà précisé, la reconnaissance s'opère au niveau phonétique, permettant à celle-ci de s'étendre sur de grands vocabulaires. En effet, l'apprentissage de HMM de mots aurait demandé un corpus d'apprentissage bien supérieur à celui que l'on dispose avec la

base TIMIT. Tandis qu'avec l'approche phonétique, l'ajout d'un nouveau mot à reconnaitre n'est pas une contrainte étant donné que chaque fragment de ce mot a été appris par le modèle. Enfin, cette approche rend l'espace mémoire nécessaire moindre car le nombre de distributions devant être conservé est inferieur à celui qu'aurait demandé l'approche par modèle de HMM par mot du lexique.

3.2.1 Création d'un prototype de HMM

On choisit un modèle de HMM initial à 3 états émetteurs (5 états au total avec l'état initial et l'état final non émetteurs) selon un modèle de Bakis (cf. Chapitre 1). En outre, chaque état est représenté par sa probabilité d'émission supposée uni-gaussienne (à ce stade). Le fichier `sim.pcf`, regroupant les informations précédentes, est appelé par le script Perl `MakeProtoHMMSet.prl` fourni par HTK :

```
$ perl ../script/MakeProtoHMMSet.prl sim.pcf
```

On peut visualiser le prototype créé sur la Figure 12 où les vecteurs `<Mean>` et `<Variance>` représentent respectivement le vecteur de moyenne et le vecteur de variance constituants de la probabilité d'émission de chaque état. La matrice de transition entre état est dénotée par `<TransP>`. Pour l'instant, les densités d'émission des observations sont représentées par une seule Gaussienne (`<NumMixes> 1`).

```
  ~o <VecSize> 39 <MFCC_E_D_A> <StreamInfo> 1 39
  ~h "proto"
<BeginHMM>
  <NumStates> 5
  <State> 2 <NumMixes> 1
  <Stream> 1
  <Mixture> 1 1.0000
    <Mean> 39
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 39
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 3 <NumMixes> 1
  <Stream> 1
  <Mixture> 1 1.0000
    <Mean> 39
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 39
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 4 <NumMixes> 1
  <Stream> 1
  <Mixture> 1 1.0000
    <Mean> 39
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 39
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <TransP> 5
   0.000e+0   1.000e+0   0.000e+0   0.000e+0   0.000e+0
   0.000e+0   6.000e-1   4.000e-1   0.000e+0   0.000e+0
   0.000e+0   0.000e+0   6.000e-1   4.000e-1   0.000e+0
   0.000e+0   0.000e+0   0.000e+0   6.000e-1   4.000e-1
   0.000e+0   0.000e+0   0.000e+0   0.000e+0   0.000e+0
<EndHMM>
```

Figure 12 Le prototype pour les futurs HMM

3.2.2 Initialisation des modèles de HMM

On crée 49 prototypes de HMM (48 phonèmes à considérer + un modèle de silence [`sil`]) similaires au prototype élaboré au paragraphe précédent. Pour l'instant vide, nous allons les initialiser en utilisant la méthode dite des *Flat Start Monophones*.

La fonction `HCompV` va calculer la moyenne et la covariance sur tout l'ensemble d'apprentissage. Ces résultats sont utilisés pour initialiser les paramètres de chaque HMM. Ainsi, chaque HMM se voit assigner comme moyenne et covariance, la moyenne et la covariance globale calculées sur toute la base d'apprentissage. Tous les modèles sont donc initialisés avec les mêmes valeurs.

```
$ HCompV -S $trainlistmfc -o $protoinitialisé $proto
```

```
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
 -1.012254e+001 -7.264226e+000 -5.370609e+000 -9.712223e+000 -8.047407e+000 -5.681805e+000 -5.715530e+000
 -8.866713e-001 -3.691204e+000 -1.679087e+000 -2.372116e+000 -1.992831e+000 -3.870823e+000 3.500786e-003
  3.751630e-003 -3.257445e-004 8.666397e-004 4.723764e-003 -5.870885e-004 2.242961e-004 -5.077728e-003
 -4.136859e-003 -6.560616e-004 -4.315113e-003 -2.692402e-003 4.300712e-004 -9.592683e-005 -2.103158e-004
  1.351117e-004 -1.661481e-005 -3.384908e-004 -2.177467e-004 -1.438305e-004 1.636888e-004 2.628795e-004
  5.652815e-006 2.228666e-004 2.034386e-004 -3.868835e-005
<VARIANCE> 39
  1.019786e+002 5.942492e+001 7.969102e+001 8.208926e+001 7.801923e+001 7.487221e+001 7.644022e+001
  6.621936e+001 6.555122e+001 4.695240e+001 4.796211e+001 3.524432e+001 1.121502e+001 4.562054e+000
  3.524029e+000 3.336633e+000 4.190800e+000 4.308127e+000 4.198907e+000 4.401058e+000 4.303515e+000
  3.945027e+000 3.243079e+000 2.985245e+000 2.440658e+000 4.580305e-001 6.298034e-001 5.834209e-001
  5.080934e-001 6.770245e-001 7.052047e-001 7.098504e-001 7.490557e-001 7.594697e-001 6.945505e-001
  5.957313e-001 5.417506e-001 4.463942e-001 6.905811e-002
<GCONST> 1.309636e+002
```

Figure 13 L'initialisation des états : l'état n°2 du fichier `protoinitialisé`

Sur la Figure 13, on peut voir un état initialisé, les 2 autres états ayant aussi les mêmes valeurs. A noter l'apparition du champ `<GCONST>`, dernier constituant de la loi de probabilité uni-gaussienne à savoir la multiplication du déterminant de la matrice de covariance par $(2\pi)^n$.

3.2.3 Apprentissage

Les 49 modèles sont réestimés à l'aide de la fonction `HERest` qui utilise l'algorithme de Baum-Welsh évoqué au Chapitre 1.

```
$ HERest -I $transphone -S $trainlistmfc -H hmm0 -M hmm1 $listphone
```

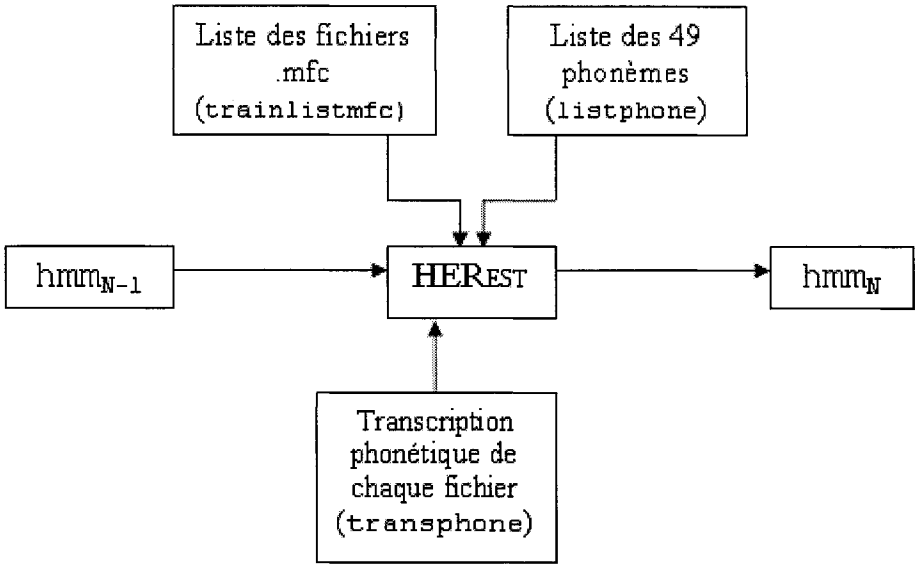

Figure 14 Le module `HERest`

On fait appel à la fonction `HERest` 4 fois. À chaque nouvelle itération, les nouveaux HMM recalculés sont stockés dans un nouveau fichier : hmm$_N$. La Figure 14 illustre le fonctionnement du module `HERest`. Le fichier `transphone` recense la liste de tous les fichiers de transcription phonétique TIMIT (*.phn, cf. Chapitre 4).

3.2.4 La reconnaissance des modèles

Pour la phase de reconnaissance, avec HTK, il faut disposer d'un réseau de mots (*word level network*), d'une grammaire et d'un ensemble de HMMs. Le réseau de mots est un ensemble de nœuds connectés entre eux par des arcs. Chaque nœud représente un modèle de HMM ou une fin de mot. Ainsi, une fois compilé, on dispose de supermodèles de HMM connectés.

La fonction `HParse` se charge de créer le réseau de mot à partir de la grammaire utilisée dans le système. On entend par grammaire (`gram`) la liste des mots utilisés, en

l'occurrence, ici, la liste des phonèmes. Le réseau de mot résultant (`wdnet`) sera utilisé par la suite dans le module de décodage `HVite`.

```
$ HParse gram wdnet
```

Le module de décodage, `HVite`, utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable correspondant aux paramètres observés dans un modèle et en déduire les unités acoustiques correspondantes (Figure 15).

Le résultat du décodage est comparé aux étiquettes de référence par un alignement dynamique réalisé par `HResults`, afin de compter les étiquettes identifiées, omises, substituées, et insérées, et de calculer le taux de reconnaissance. Il est possible d'indiquer au module `HResults` de confondre certains phonèmes entre eux que l'on jugerait très proches phonétiquement afin d'augmenter le pourcentage de reconnaissance.

Ainsi par exemple, on confondra les phonèmes suivants (Tableau I).

Tableau I

Module `HResults` : Liste des phonèmes à confondre

ih	ix
ah	ax
aa	ao
er	axr
l	el
m	em
sh	zh
cl	vcl
cl	sil
!ENTER	sil

```
$ HVite -S $testlistmfc -H $hmm4 -i $result.mlf -w $wdnet $listphone
```

```
$ HResults -t -T 1 -C $confus -I $transphonetest $result.mlf > tauxreco
```

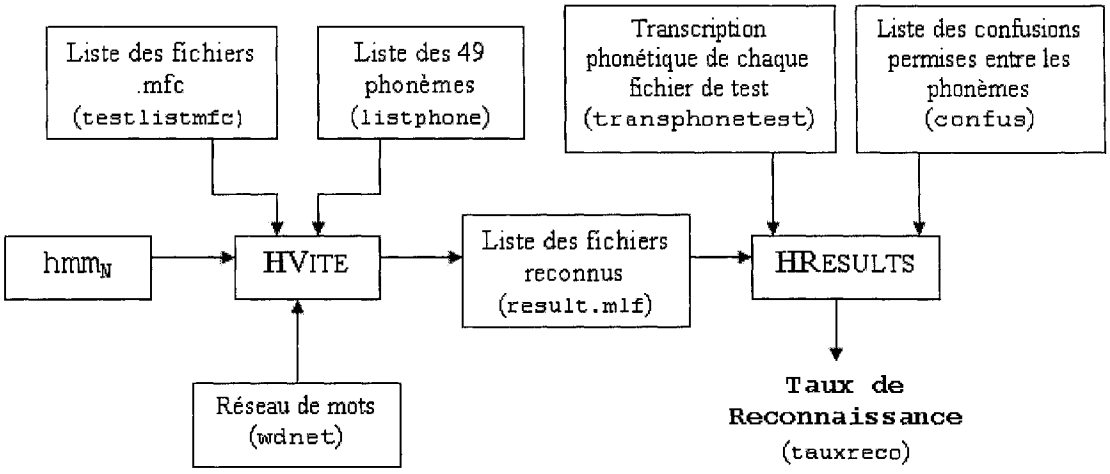

Figure 15 Les modules `HVite` et `HResults`

3.2.5 L'obtention du taux de reconnaissance

Le module `HResults` compare les alignements phonétiques des enregistrements de test générés par `HVite` (*.rec) avec les labels connus de ces enregistrements (*.lab) (Figure 16), et nous fournit le taux de reconnaissance en pourcentage ainsi que d'autres informations plus ou moins utiles pour ce projet (Figure 17).

```
Aligned transcription:
 c:/Users/rpreiss/Timit/test/dr3/mcsh0/si919.lab
 LAB: sil aa vcl v iy ix s  l iy dh ix vcl b r ay        dx el cl
      p eh   axr hh ae z sp m ix n iy eh vcl jh ah s cl m    ax
      n cl  s  cl t ix m ey cl    sp t uw dh eh r   n uw
      s ih cl ch    uw ey sh        en       sil
 vs
 c:/Users/rpreiss/Timit/test/dr3/mcsh0/si919.rec
 REC: sil aa vcl b iy ih sh l iy dh ix vcl b r ay ih vcl b  el cl
      p eh r axr hh ae z    m ih n iy ih vcl jh ah s ch m dh ax
        vcl jh cl t ix m ey cl dh cl t uw dh    axr n uw
      s ih cl ch ax w  ey sh ix vcl en cl ax sil
```

Figure 16 Exemple de comparaison des transcriptions phonétiques

Sur la Figure 17, la première ligne des résultats n'est pas très pertinente dans le cas de la reconnaissance de la parole continue. En effet, la ligne `SENT` donne le pourcentage d'enregistrements reconnus entièrement et correctement. Il est donc assez logique qu'il soit très faible. La seconde ligne est celle qui nous délivre les performances de notre système (`WORD`).

Le résultat qui nous intéresse est l'*Accuracy* (`Acc` sur la Figure 17), c'est-à-dire le pourcentage de reconnaissance des phonèmes reconnus correctement. Il se calcule ainsi :

$$\text{Accuracy} = \frac{N - D - S - I}{N} \times 100 \tag{3.1}$$

N : nombre total de labels

D : nombre d'erreurs de suppression

S : nombre d'erreurs de substitution

I : nombre d'erreurs d'insertion

A noter que la statistique `Corr` est semblable à `Acc` sauf que les erreurs d'insertions ne sont pas comptabilisées. HTK préconise d'utiliser l'*Accuracy* qui reflète mieux les performances du système.

$$\text{Correct} = \frac{N - D - S}{N} \times 100 \tag{3.2}$$

```
====================== HTK Results Analysis ==========
  Date: Sat Jan 07 21:54:33 2006
  Ref : lablist/testtri.mlf
  Rec : work/hmm20/result0.0.mlf
------------------------ Overall Results -------------
SENT: %Correct=0.06 [H=1, S=1679, N=1680]
WORD: %Corr=72.60, Acc=56.45,
          [H=45668, D=3406, S=13827, I=10159, N=62901]
=========================================================
```

Figure 17 Les statistiques générées par `HResults`

Ainsi, pour cet exemple, on peut donc dire que sur 62901 phonèmes présents dans le corpus de test, 45668 ont été reconnus correctement (72.60 %). Toutefois, en comptabilisant les erreurs d'insertions (ici, au nombre de 10159), l'*Accuracy* chute à 56.45 %.

3.3 Nouvel apprentissage suite à l'amélioration du modèle de silence

On renforce le modèle [`sil`] en lui ajoutant une transition bidirectionnelle entre les états 2 et 4 (voir Figure 19 tirée de [10]). L'idée ici, est de rendre le modèle plus robuste en permettant à chaque état d'absorber les différents bruits impulsionnels des données d'entraînement. Le retour de l'état 4 vers l'état 2 permet cette prise en compte en empêchant le model de transiter vers un autre phonème à la fin de l'état 4.

On crée également un autre automate à un seul état émetteur que l'on appelle "courte pause" ([`sp`] pour *short pause*). Comme montré à la Figure 19, [10], ce modèle possède une transition directe de l'entrée vers la sortie. Également, son seul état émetteur est partagé avec l'état émetteur n°2 du modèle de silence [`sil`].
La fonction `HHEd` va modifier la version apprise `hmm4` à l'aide du script `sil.hed` et contenir le tout dans la nouvelle version `hmm5`.

```
HHEd -H $hmm4 -M $hmm5 $sil.hed $listphone
```

Ensuite, on réestime 2 fois nos modèles avec `HERest` (Figure 18).

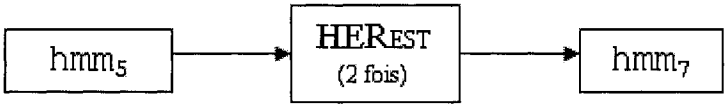

Figure 18 Réestimation des modèles `hmm5` et `hmm6`

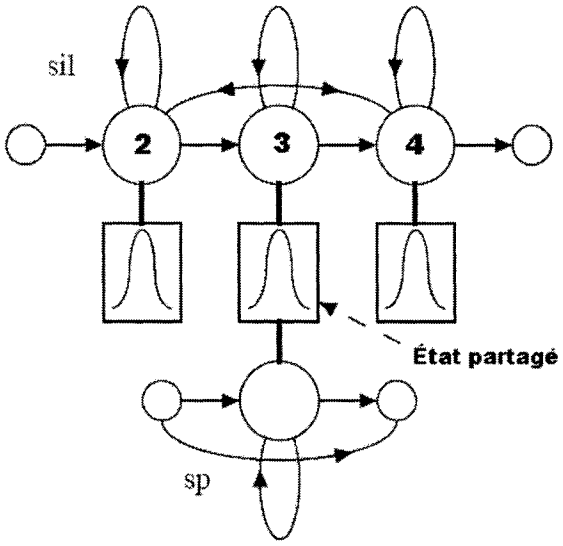

Figure 19 Améliorations apportées au modèle [sil]

3.4 Nouvel apprentissage suite au réalignement des données d'entraînement

L'objectif de l'alignement des données acoustiques et phonétiques est de déterminer les suites de phonèmes les plus « réalistes » possibles parmi toutes les combinaisons de phonèmes possibles. Ainsi cette partie a pour but d'identifier toutes les prononciations possibles d'une phrase et également de trouver la prononciation optimale en considérant les données acoustiques du corpus d'apprentissage.

```
$ HVite -H $hmm7 -I $aligned -I $transphone -S $trainlistmfc
$listephone50
```

Ensuite, on réestime 2 fois nos modèles avec `HERest`, on obtient `hmm9`.

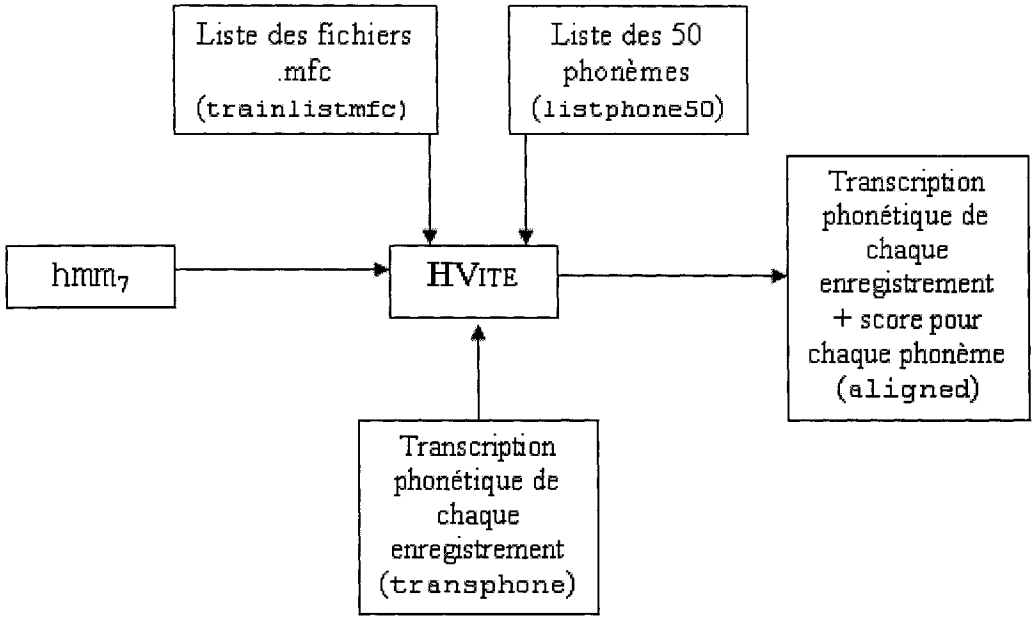

Figure 20 Fonctionnement du module `HVite` pour l'alignement

```
"c:/Users/rpreiss/DataClean/train/dr1/fcjf0/sa1.rec"
          0   1800000   sil  -1159.234741
    1800000   2700000    sh   -632.443970
    2700000   3600000    ix   -795.403442
    3600000   4300000    hh   -528.474426
    4300000   5200000    eh   -626.407471
    5200000   5800000   vcl   -533.564819
    5800000   6400000    jh   -525.688416
    6400000   7100000    ih   -540.527588
    7100000   7600000   vcl   -431.776978
    7600000   8000000     d   -338.871460
    8000000   9100000    ah   -786.127380
    9100000   9600000    cl   -415.071838
    9600000  10200000     k   -476.190796
   10200000  11200000     s   -731.749207
   11200000  12900000    uw  -1232.812622
   12900000  13800000    en   -719.964661
   13800000  14100000   vcl   -261.525848
   14100000  14700000     g   -494.726166
```

phonème **score du phonème** **durée du phonème**

Figure 21 Exemple d'obtention des scores d'alignement

3.5 Apprentissage par triphones

Les modèles phonétiques indépendants du contexte ne permettent pas une bonne représentation des caractéristiques acoustiques. Comme déjà spécifié au Chapitre 1, le phonème est un bon choix d'unité acoustique, principalement parce que son nombre est limité dans chaque langue. Toutefois, en raison des phénomènes de coarticulation, il est préférable de travailler avec des modèles tenant compte des contextes phonétiques gauche et droit : les triphones.

L'apprentissage de modèles contextuels est cependant plus difficile, car il n'est pas toujours possible de disposer d'un nombre suffisant de représentants pour l'apprentissage de chaque modèle d'allophone. En effet, un ensemble de N phonèmes génère jusqu'à C_N^3 triphones.

3.5.1 Création de triphones à partir de phonèmes

La fonction `HLEd` va copier (dans `transtri`) le fichier précédemment créé `aligned` en remplaçant chaque phonème par un triphone constitué du phonème précédent, de l'actuel et du phonème suivant (Figure 22). La liste de tous les triphones créés est affichée dans le fichier `triphones`. Avec la prise en compte des contextes gauche et droit, le nombre de modèles peut s'élever à plusieurs dizaines de milliers!

```
      0  1800000        sil -1159.234741
1800000  2700000 sil-sh+ix  -632.443970
2700000  3600000  sh-ix+hh  -795.403442
3600000  4300000  ix-hh+eh  -528.474426
4300000  5200000 hh-eh+vcl  -626.407471
5200000  5800000 eh-vcl+jh  -533.564819
5800000  6400000 vcl-jh+ih  -525.688416
6400000  7100000 jh-ih+vcl  -540.527588
7100000  7600000  ih-vcl+d  -431.776978
7600000  8000000  vcl-d+ah  -338.871460
8000000  9100000   d-ah+cl  -786.127380
9100000  9600000   ah-cl+k  -415.071838
9600000 10200000    cl-k+s  -476.190796
10200000 11200000    k-s+uw  -731.749207
11200000 12900000   s-uw+en -1232.812622
12900000 13800000 uw-en+vcl  -719.964661
13800000 14100000  en-vcl+g  -261.525848
14100000 14700000   vcl-g+r  -494.726166
```

Figure 22 Création de triphones (fichier `transtri`)

```
HLEd -n $triphones -i $transtri xwrd.hled $aligned
```

3.5.2 Clonage

Le nombre de modèles de HMM à entraîner passe désormais de 50 à plusieurs centaines ! Il est difficile de disposer de suffisamment de données pour apprendre de manière robuste la totalité des modèles de triphones possibles. En effet, certains n'apparaissent que quelquefois dans toute la base d'apprentissage, il est prévisible que ces modèles ne seront pas estimés correctement. Pour contourner cette difficulté, il est possible de regrouper plusieurs contextes produisant un effet similaire sur la réalisation acoustique du phonème. Toutefois, le partage des composants peut avoir un effet nuisible s'il s'opère de manière aléatoire. C'est pourquoi, il faut mieux partager des données qui ne varient pas de trop selon le contexte acoustique gauche-droit, c'est le cas des matrices de transition.

```
HHEd -H $hmm9 -M $hmm10 mktri.hed $listphone50
```

Sur la Figure 23, le fichier de commande `mktri.hed contient` les instructions à passer à la fonction de clonage `HHEd`.

Commande `CL` : Pour chaque modèle de la forme `a-b+c` dans le fichier de triphones `triphone`, la fonction `HHEd` regarde le phonème `b` et fait une copie de celui-ci.

Commandes `TI` : chaque commande `TI` prend comme arguments le nom d'une macro (`T_b`) suivi d'une liste de paramètres de HMM. `HHEd` va alors dupliquer le HMM `b` autant de fois qu'il y a des triphones de la formes `a-b+c` en remplaçant le « nom » `b` par le triphone considéré. Ensuite, au lieu de recopier dans chaque nouveau modèle la matrice de transition, il va la placer en tête du fichier et cette matrice sera désignée dans chaque nouveau modèle par `~t "T_b"`. Ainsi tous les modèles de la forme `a-b+c` partagent la même matrice de transition.

```
CL triphone
TI T_aa {(*-aa+*,aa+*,*-aa).transP}
TI T_ae {(*-ae+*,ae+*,*-ae).transP}
TI T_ah {(*-ah+*,ah+*,*-ah).transP}
TI T_ao {(*-ao+*,ao+*,*-ao).transP}
TI T_aw {(*-aw+*,aw+*,*-aw).transP}
TI T_ax {(*-ax+*,ax+*,*-ax).transP}
TI T_axr {(*-axr+*,axr+*,*-axr).transP}
TI T_ay {(*-ay+*,ay+*,*-ay).transP}
```

Figure 23 Une partie du fichier de commande `mktri.hed`

La Figure 24, tirée de [10] , résume le clonage opéré par `HHEd`.

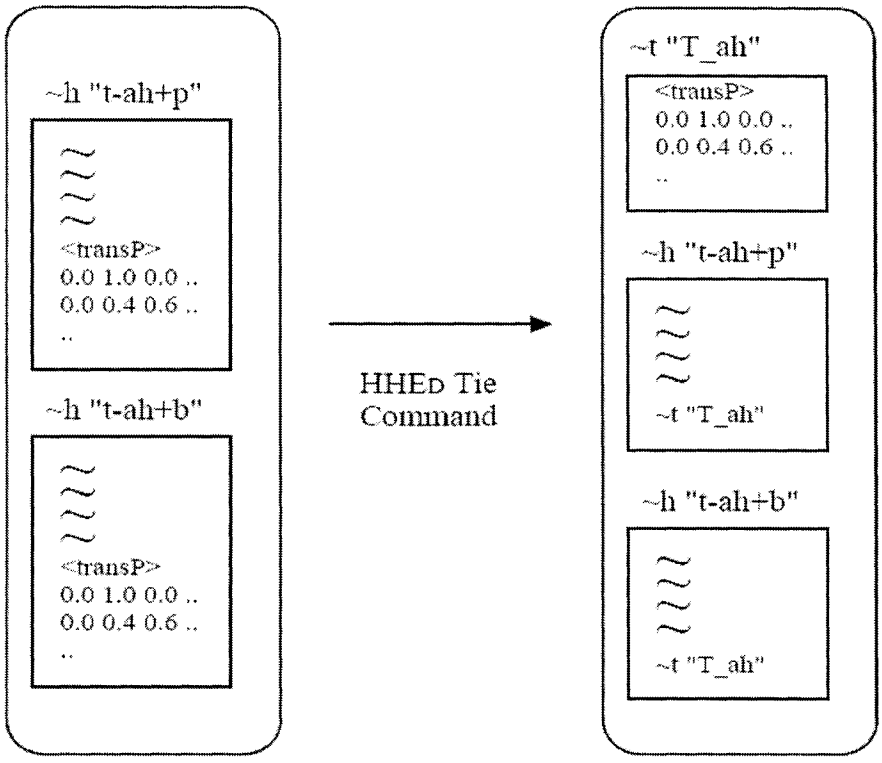

Figure 24 Mécanisme de clonage avec `HHEd`

Ensuite, on réestime 2 fois nos modèles avec `HERest`, on obtient `hmm12`.

3.5.3 Création de triphones avec états partagés

On va essayer d'affiner un peu nos modèles car considérer uniquement le partage des matrices de transitions manque un peu de subtilité. Nous allons choisir au cas par cas quel modèle va partager ses paramètres avec un autre. Ainsi des HMM de triphones vont se partager des états similaires, d'où le nom de *tied-state triphones*. C'est l'objet de la dernière partie du paragraphe 3.5.

```
HHEd -H $hmm12 -M $hmm13 tree.hed $triphones > log
```

Le fichier `tree.hed` (Figure 25) est également un fichier de commande mais beaucoup plus complexe que `mktri.hed`. Il est tout d'abord constitué d'une série de questions (`QS`). Par exemple, la troisième, (`QS "R_Stop"`) retourne `Vrai` si le contexte droit (`R` pour Right) est un phonème de la classe des occlusives ([p], [b], [t], [d], [k], [g]). Les questions progressent ainsi en s'intéressant à toutes les étiquettes (Affriquées, Fricatives, Nasales, Liquides, Voyelles, Diphtongues, etc.).

```
RO 100.0 hmm12/stats
QS  "R_NonBoundary"     { *+* }
QS  "R_Silence"         { *+sil }
QS  "R_Stop"            { *+p,*+b,*+t,*+d,*+k,*+g }
...
QS  "R_aa"              { *+aa }
QS  "R_ae"              { *+ae }
QS  "R_ah"              { *+ah }
...
QS  "L_Silence"         { sil-* }
QS  "L_Stop"            { p-*,b-*,d-*,d-*,k-*,g-* }
QS  "L_Nasal"           { m-*,n-*,en-*,ng-* }
...
QS  "L_aa"              { aa-* }
QS  "L_ae"              { ae-* }
QS  "L_ah"              { ah-* }
...
TB 350.0 "ST_aa_2_" {("aa","*-aa+*","aa+*","*-aa").state[2]}
TB 350.0 "ST_ae_2_" {("ae","*-ae+*","ae+*","*-ae").state[2]}
TB 350.0 "ST_ah_2_" {("ah","*-ah+*","ah+*","*-ah").state[2]}
TB 350.0 "ST_ao_2_" {("ao","*-ao+*","ao+*","*-ao").state[2]}
...
TB 350.0 "ST_aa_3_" {("aa","*-aa+*","aa+*","*-aa").state[3]}
TB 350.0 "ST_ae_3_" {("ae","*-ae+*","ae+*","*-ae").state[3]}
TB 350.0 "ST_ah_3_" {("ah","*-ah+*","ah+*","*-ah").state[3]}
TB 350.0 "ST_ao_3_" {("ao","*-ao+*","ao+*","*-ao").state[3]}
...
TB 350.0 "ST_aa_4_" {("aa","*-aa+*","aa+*","*-aa").state[4]}
TB 350.0 "ST_ae_4_" {("ae","*-ae+*","ae+*","*-ae").state[4]}
TB 350.0 "ST_ah_4_" {("ah","*-ah+*","ah+*","*-ah").state[4]}
TB 350.0 "ST_ao_4_" {("ao","*-ao+*","ao+*","*-ao").state[4]}
...
ST "trees"
AU "unseen.list"
CO "treeg.list"
```

Figure 25 Aperçu du fichier `tree.hed`

Ensuite, le partage des états s'effectue par l'entremise des commandes `TB`. Considérons un exemple, cela sera beaucoup plus clair. Sur la Figure 25, la commande `TB` (`TB 350.0`

"ST_ao_3_") agit sur tous les états centraux (state[3]) des modèles ayant le phonème [ao] comme noyau (phonème central).

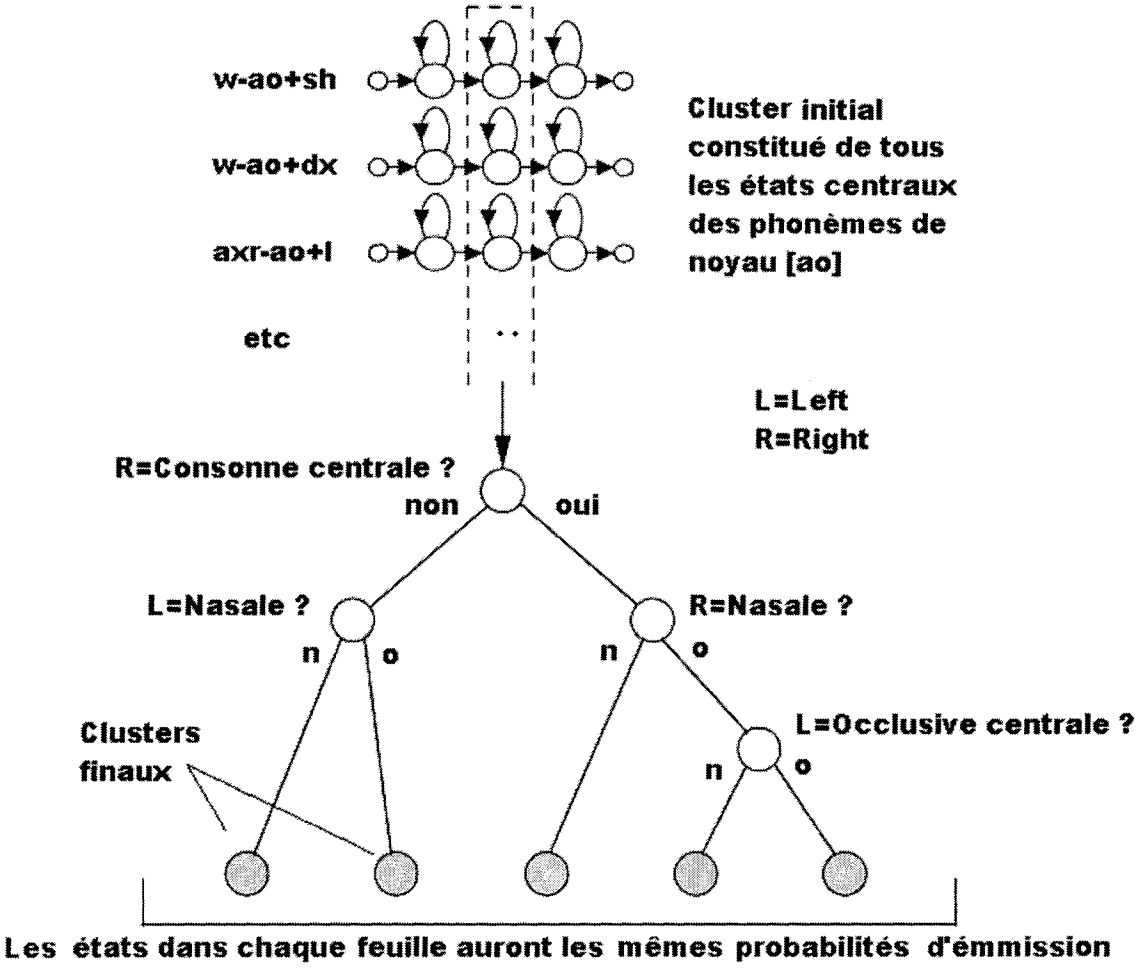

Figure 26 Un exemple de partitionnement (*Tree Base Clustering*)

On place tous les éléments (l'état n°3 de chaque déclinaison du phonème [ao]) dans un nœud source (Figure 26 inspirée par [10]). À la première question QS, l'ensemble est dupliqué en deux. Si la question permet d'augmenter la vraisemblance de l'ensemble d'apprentissage, la question est sélectionnée pour être la première branche de l'arbre. Si la vraisemblance obtenue à chaque question à n'importe quel nœud de chaque ensemble

est au dessus d'un certain seuil (ici `350.0`), le processus est répété. Au final, les états dans chaque cluster `i` vont se partager un unique état nommé `ST_ao_3_i`.

```
TB 350.00 ST_ao_3_ {}
 Tree based clustering
  Start aw[3] : 327  have  LogL=-86.899 occ=864.2
  Via   aw[3] : 22   gives LogL=-84.421 occ=864.2
  End   aw[3] : 22   gives LogL=-84.421 occ=864.2
TB: Stats 327->22 [6.7%]  ( 17873->722 [4.0%] total )
```

Figure 27 Aperçu du fichier `log` pour l'exemple considéré

On peut voir sur la Figure 27, le fichier `log` qui recense les résultats du clustering pour tous les états de tous les modèles. Nous poursuivons l'explication du fonctionnement avec l'exemple `"ST_ao_3_"`. Au commencement du processus, 327 modèles étaient dans le cluster initial avec une vraisemblance moyenne de `-86.9`. Après le clustering, les 327 modèles se sont classés en 22 clusters finaux.

Ensuite, on réestime 2 fois nos modèles avec `HERest`, on obtient `hmm15`.

3.6 Augmentation du nombre de Gaussiennes par état

Des très bons scores peuvent être obtenus en utilisant des mixtures de gaussiennes. En effet, les lois gaussiennes uni-modales, régissant les probabilités d'émissions des observations ne suffisent pas à modéliser les données émises par un état dans un automate probabiliste. On va préférer utiliser une combinaison linéaire de gaussiennes dans l'espace R^d :

$$b_j(o) = \sum_{k=1}^{G} g_k \cdot \mathrm{N}(o, \mu_k, \Sigma_k) \tag{3.3}$$

Chaque gaussienne est caractérisée par son poids g_k, un vecteur moyen μ_k, une matrice de covariance Σ_k et a pour expression :

$$\mathrm{N}(o,\mu_k,\Sigma_k)=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(o-\mu)^T\,\Sigma^{-1}(o-\mu)\right) \tag{3.4}$$

Jusqu'à présent, ($k=1$) une probabilité d'observation n'était représentée que par une seule gaussienne. Une des méthodes pour augmenter le nombre de gaussiennes par état est développée dans le manuel HTK. On commence d'abord par apprendre des modèles avec $k=1$ puis le nombre de gaussiennes est augmenté par clonage en apportant à chaque clone une petite perturbation proportionnelle au vecteur écart type $\sigma_k = diag(\Sigma_k)$:

$$g_k\cdot\mathrm{N}(\mu_k,\Sigma_k)\rightarrow\begin{cases}\frac{g_k}{2}\cdot\mathrm{N}(\mu_k-\alpha\sigma_k,\Sigma_k)\\ \frac{g_k}{2}\cdot\mathrm{N}(\mu_k+\alpha\sigma_k,\Sigma_k)\end{cases} \tag{3.5}$$

HTK préconise de faire cette étape à la fin car le processus de partage des états n'est pas implémenté pour des mélanges de gaussiennes. Encore une fois, nous allons utiliser la fonction `HHed` mais avec la commande `MU`.

```
HHEd -H $hmm15 -M $hmm16 split.hed.sim treeg.list
```

```
MU 2 {
zh-t+w.state[2].mix,
uw-ih+v.state[4].mix,
r-dx+z.state[3].mix,
n-vcl+ch.state[2].mix,
n-vcl+ch.state[3].mix,
k+ax.state[3].mix,
....
w-ix+w.state[4].mix,
b-ae+sh.state[2].mix,
ey-n+aw.state[2].mix,
sil.state[2].mix}
MU 3 {
zh-t+w.state[3].mix,
vcl-l+vcl.state[4].mix,
th-cl+th.state[3].mix,
...
d-r+ae.state[2].mix,
l-ah+dx.state[3].mix,
sil.state[2].mix}
```

Figure 28 Aperçu du fichier `split.hed.sim`

La commande MU est de la forme :

```
MU N élément.state[n].mix
```

`N` est le nombre de gaussiennes désiré pour l'état `n` du triphonème `élément`. La démarche est celle décrite à l'équation (3.5) avec $\alpha = 0.2$ et est répétée autant de fois que nécessaire en choisissant à chaque nouvelle itération la gaussienne ayant le poids le plus fort. Par exemple, la première commande de la Figure 28 va incrémenter le nombre de gaussiennes à 2 de la probabilité d'émission de l'état 2 du modèle `zh-t+w`.

Toutefois, tous les états ne reçoivent pas les mêmes traitements et n'ont pas à l'arrivée le même nombre de gaussiennes. Il est en de même avec les modèles où certains triphones ont 30 gaussiennes alors que d'autres n'en ont que 2 ou 3.

Ensuite, on réestime 4 fois nos modèles avec `HERest`, on obtient `hmm20`.

3.7 Conclusion

Avec l'obtention de `hmm20`, nous avons terminé l'apprentissage. On peut donc une nouvelle fois tester le système sur le corpus de test de TIMIT tel que décrit au paragraphe 3.2.4. Pour ce qui est des résultats, nous aurons naturellement de meilleurs taux en testant la base de test sur `hmm20` que sur `hmm9` ou `hmm15`, le traitement sur les HMM y étant plus élaboré.

Il important de préciser que l'apprentissage tel que nous l'avons décrit et effectué n'est pas l'apprentissage qui apporte les taux de reconnaissance les plus élevé, la recherche d'un taux de reconnaissance toujours plus élevé est l'activité principale de nombreux chercheurs dans la communauté.

Toujours pour comparer par rapport à ce qu'il se fait dans les articles, nous pouvons considérer notre système comme standard et suffisamment robuste pour mesurer les effets du canal téléphonique sur un système de reconnaissance de la parole.

CHAPITRE 4

MÉTHODE PROPOSÉE ET RÉSULTATS EXPÉRIMENTAUX

L'objectif de cette maîtrise est de développer des techniques visant à compenser l'effet du canal téléphonique pour rendre plus robuste un système de reconnaissance via le téléphone fixe. Dans le cadre de ce projet, nous avons développé une méthode pour adapter les modèles d'entraînement par le biais d'un apprentissage multiréférences. Pour ce faire, il a fallu disposer de modélisations de canaux téléphoniques pour bruiter la base d'apprentissage en se focalisant exclusivement sur l'origine des bruits propres au canal, en l'occurrence, la limitation de la bande passante.

4.1 Modélisation du canal téléphonique

On a vu au Chapitre 2 que le canal opère comme un filtre sur le signal vocal (Figure 29). Ainsi, soit

- *e(t)* désignant la parole numérisée,
- *h(t)* désignant la réponse impulsionnelle du canal,
- *s(t)* la parole en sortie du canal résultant de la convolution de h(t) avec l'entrée.

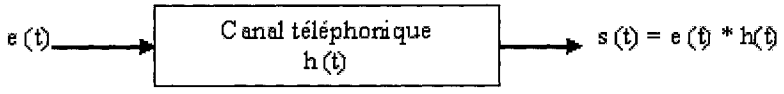

Figure 29 Le canal téléphonique opère comme un filtre

Il convient de se demander quel genre de filtres modélise au mieux le canal téléphonique. À cette interrogation, John G. Proakis [41] apporte des éléments de réponse.

Le canal téléphonique agit comme un filtre passe-bande sur la plage [300Hz ; 3400 Hz]. La réponse impulsionnelle moyenne du canal est finie et de durée d'environ 10ms. Le retard n'est pas constant dans toute la bande utile mais varie en fonction de la fréquence. On peut donc en déduire l'existence d'une phase non linéaire.

Finalement, *h(t)* peut être modélisé comme un filtre passe-bande à réponse impulsionnelle finie à phase non linéaire.

4.1.1 Construction de filtres modélisant le canal téléphonique

[41] fournit les gabarits de 4 types de canaux téléphoniques. Ces 4 gabarits sont le 3002, 3002-C1, 3002-C2 et le 3002-C4. Les 3 derniers utilisent des équaliseurs analogiques pour améliorer les caractéristiques en fréquence du canal. Ces canaux sont représentés sur la Figure 31.

Le but de cette partie est de générer des filtres, non identiques, modélisant des canaux téléphoniques pour faire en sorte que chaque enregistrement TIMIT soit bruité par un canal téléphonique différent (Figure 30). Ceci afin de refléter le caractère tous différents des canaux téléphoniques que nous pouvons rencontrer.

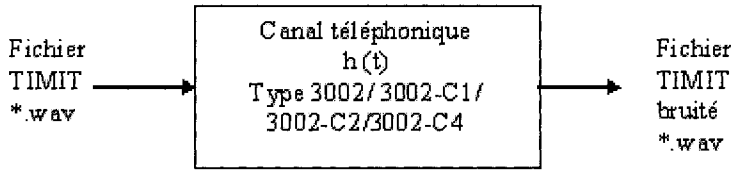

Figure 30 Méthodologie pour bruiter la base TIMIT

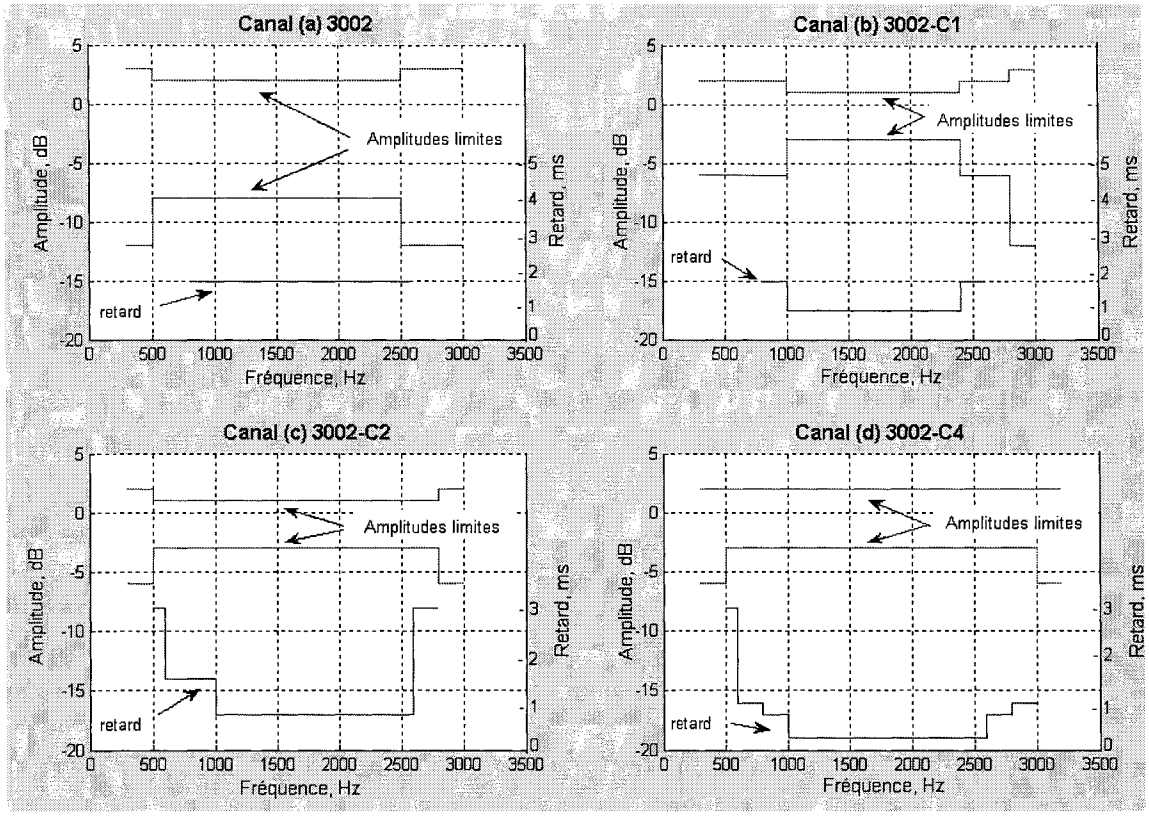

Figure 31 Gabarit des 4 filtres prototypes

La méthode utilisée à ces fins fut de construire à partir des 4 filtres prototypes (issus des 4 gabarits proposés dans le livre) une multitude de filtres dans chaque catégorie moyennant une petite perturbation pour respecter le gabarit qui leur est dicté par le type du canal d'origine. Matlab fut l'outil utilisé pour cette réalisation.

4.1.1.1 Exemple avec le canal de type 3002

La méthode employée pour construire le filtre est, disons-le, un peu grossière mais le résultat est probant ! Nous allons générer séparément la phase et l'amplitude à partir de techniques d'interpolation telles que les polynômes de Lagrange et la méthode des splines cubiques [42].

Un enregistrement TIMIT est échantillonné à 16 kHz, le théorème de Shannon nous assure donc que son spectre ne dépasse pas 8 kHz. Ainsi, toutes les modélisations seront étudiées, dans un premier temps, sur la plage [0 – 8 kHz].

4.1.1.1.1 Génération de la réponse en amplitude

On reproduit la réponse fréquentielle en amplitude en utilisant la méthode des splines cubiques. À partir de N points prélevés et mesurés (de coordonnées (x_i , y_i)) sur la courbe [41] représentant l'allure de la réponse fréquentielle en amplitude du canal, la méthode va nous interpoler N polynômes (Figure 33 et Figure 33).
Chaque polynôme relie deux points prélevés consécutivement.

Le spline cubique va ainsi se composer des N équations suivantes:

$$\forall i \in [1,\ N],\ \ g_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i \qquad (4.1)$$

Avec a_i, b_i, c_i, d_i, les coefficients de la spline à calculer.

L'amplitude $|H(f)|$ s'obtient ainsi : $|H(f)| = g_i(f)$ si $f \in [x_i, x_{i+1}]$

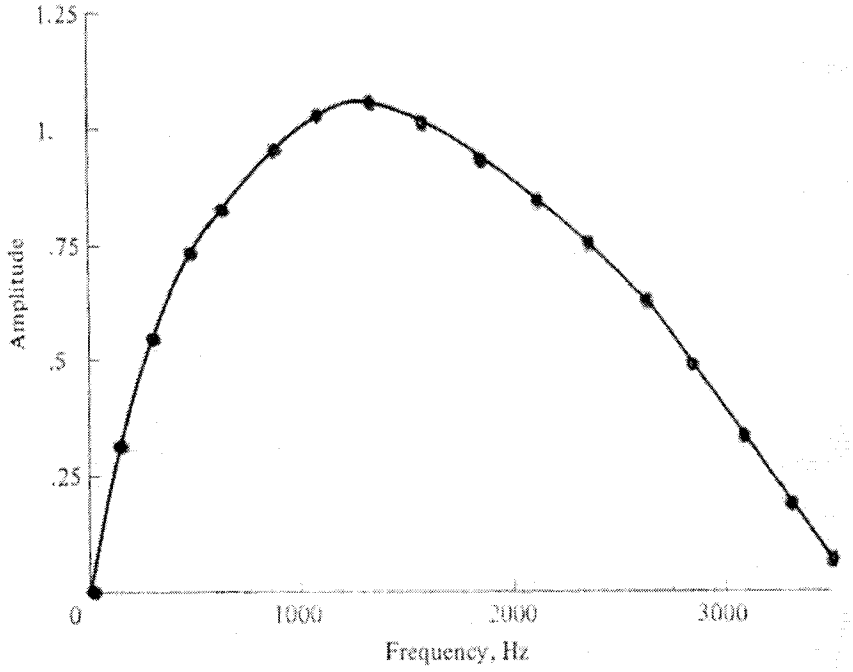

Figure 32 Amplitude moyenne d'un canal téléphonique [41]

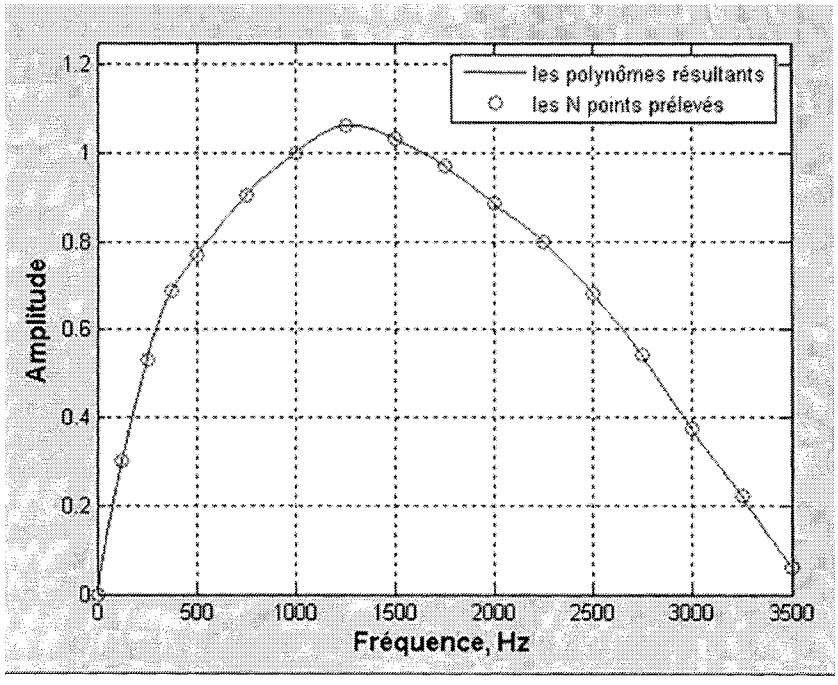

Figure 33 Interpolation de l'amplitude par la méthode des splines cubiques

4.1.1.1.2 Génération de l'enveloppe du retard

On reproduit la réponse fréquentielle du retard présente dans [41] en utilisant cette fois la méthode du polynôme de Lagrange qui convient mieux, dans ce cas, que la méthode du spline cubique.

L'expression du retard $\tau(f)$ s'obtient par la formule générale du polynôme de Lagrange :

$$\tau(f) = \sum_{j=0}^{N} y_j \prod_{\substack{i=0 \\ i \neq j}}^{N} \frac{f - x_i}{x_j - x_i} \tag{4.2}$$

Où les (x_i, y_i) sont les coordonnées des points prélevés et mesurés sur la courbe de [41] (Figure 34 et Figure 35).

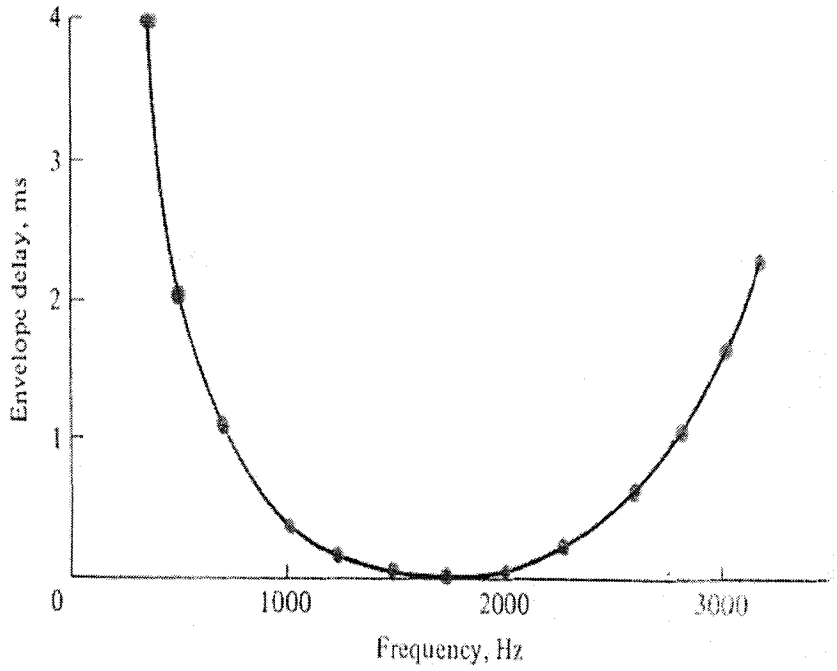

Figure 34 Enveloppe du retard moyen induit par un canal téléphonique [41]

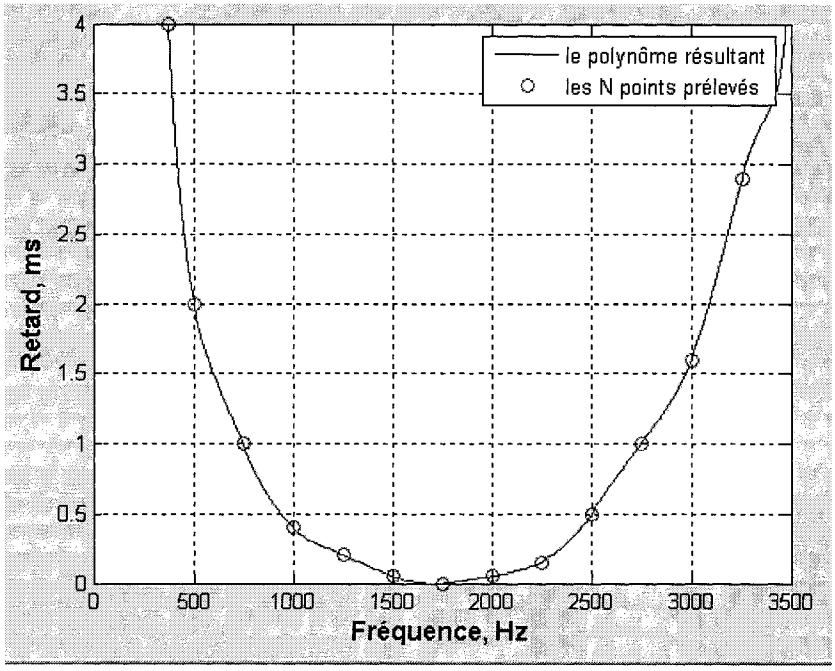

Figure 35 Interpolation de l'enveloppe du retard par les polynômes de Lagrange

4.1.1.1.3 Génération du filtre

Le retard $\tau(f)$ et la phase $\theta(f)$ sont liés par la relation suivante :

$$\tau(f) = -\frac{1}{2\pi} \cdot \frac{d\theta(f)}{df} \tag{4.3}$$

Ainsi, nous pouvons obtenir l'expression de la phase $\theta(f)$.

Une fois la phase $\theta(f)$ et l'amplitude $|H(f)|$ générées, il nous reste à effectuer une transformée de Fourier discrète inverse sur le filtre pour obtenir la réponse impulsionnelle désirée. Enfin, on ne garde que la partie réelle du filtre.

$$H(f) = |H(f)| \cdot \exp(j \cdot \theta(f)) \tag{4.4}$$

$$h(n) = real\left[TF^{-1}\left[H(f)\right]\right] \tag{4.5}$$

avec $$TF^{-1}\left[H(f)\right](n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) \cdot \exp(\frac{2\pi nk}{N}) \tag{4.6}$$

On affiche la réponse en amplitude de h (Figure 36) et s'assurer qu'elle « rentre » bien dans le gabarit associé.

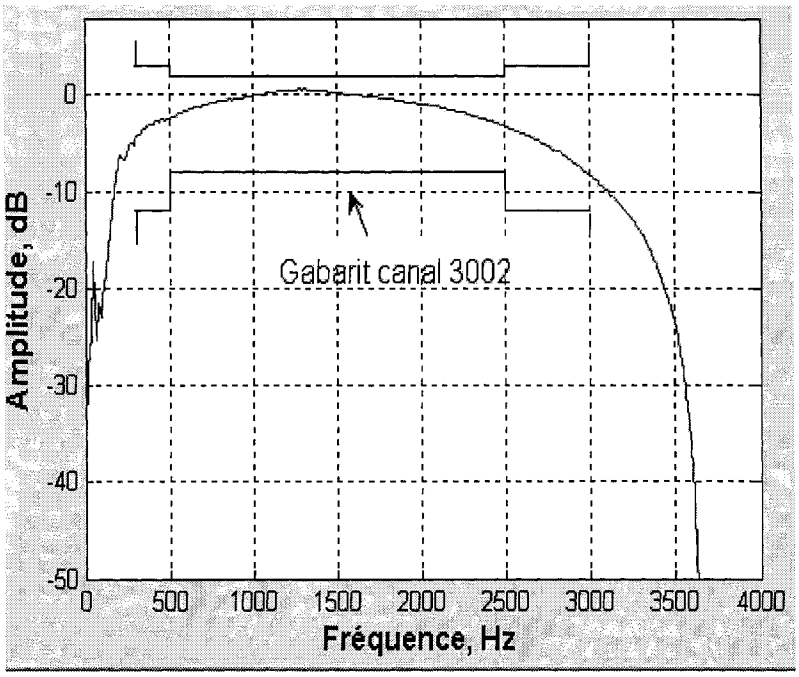

Figure 36 Réponse fréquentielle en amplitude du filtre généré

La phase de *h* est aussi représentée sur la Figure 37.

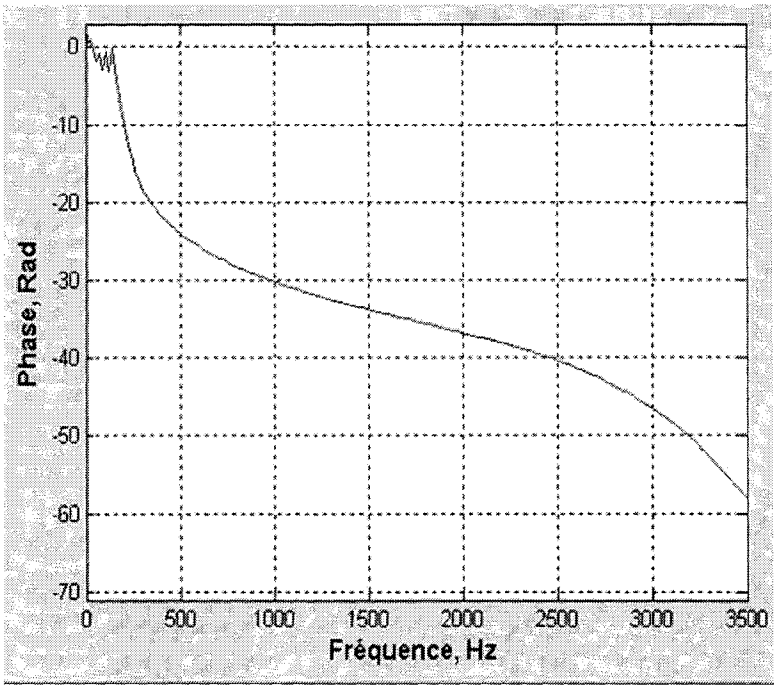

Figure 37 Phase du filtre généré

On affiche la réponse de l'enveloppe du retard (Figure 38) et s'assurer qu'elle « rentre » bien dans le gabarit associé.

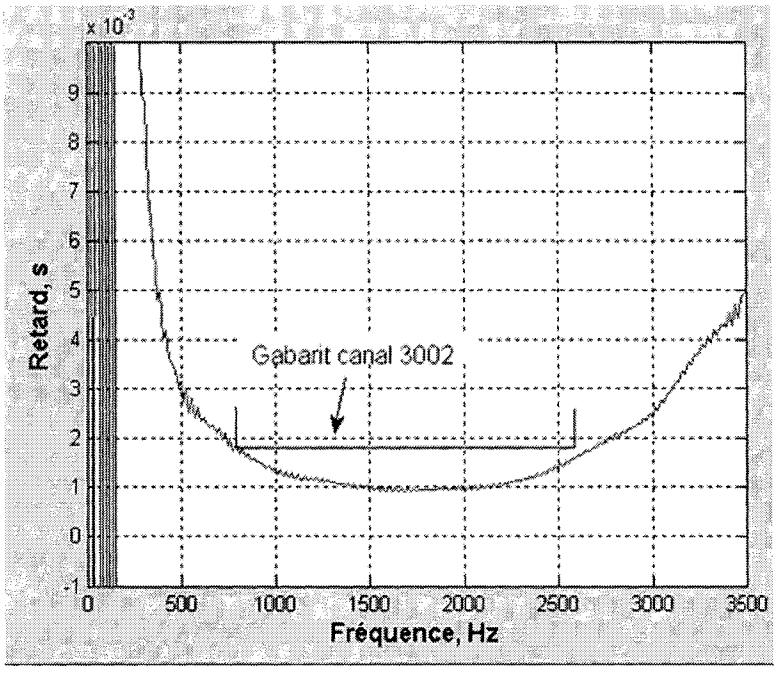

Figure 38 Retard du filtre généré

Nous pouvons comparer la réponse impulsionnelle obtenue avec celle qui figure dans [41], l'allure est la même (Figure 39 et Figure 40).

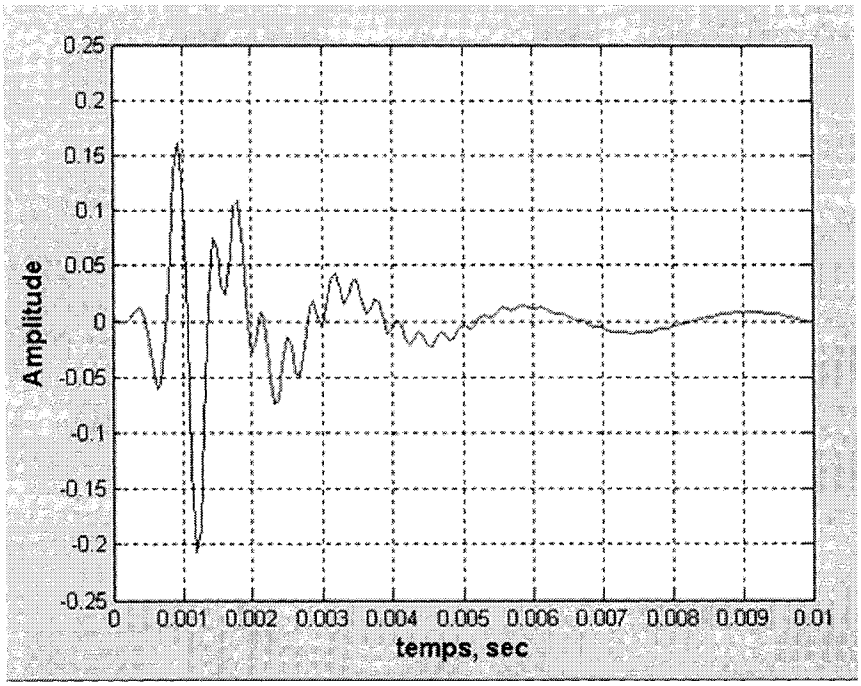

Figure 39 Réponse impulsionnelle du filtre généré

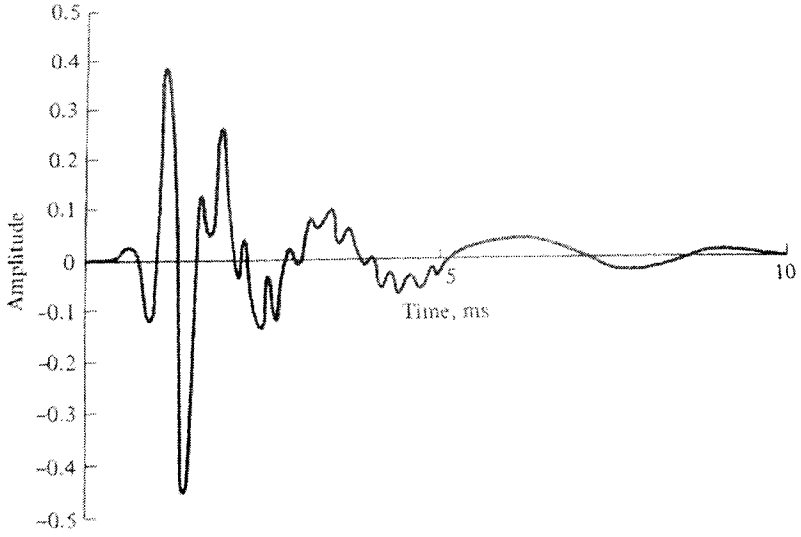

Figure 40 Réponse impulsionnelle du [41]

Nous avons donc généré une réponse impulsionnelle de modélisation de canal téléphonique. Il est toutefois abusif de parler de « filtre » pour designer cette

modélisation car la méthode utilisée ne permet pas *a priori* de connaître la fonction de transfert de cette modélisation et donc de préciser les propriétés intrinsèques du filtre (stabilité notamment).

4.1.1.2 Génération des 3 autres canaux

Le processus demeure identique pour les 3 autres filtres prototypes. La seule chose différente est bien sur l'allure de la réponse fréquentielle qui est construite de manière à respecter le gabarit associé au filtre.

4.1.1.3 Généralisation à une multitude de canaux

A partir des 4 filtres prototypes réalisés, on génère, un grand nombre de filtres pour s'assurer que chaque fichier TIMIT soit bruité par un canal différent. Ceci, afin de refléter au mieux l'hétérogénéité des canaux téléphoniques existants.

Pour ce faire, les 4 filtres sont modifiés par l'ajout d'un gain variable pour couvrir tout l'espace disponible dans le gabarit (type 3002 / 3002-C1 / 3002-C2 / 3002-C4). Pour chaque prototype de filtres, 1100 filtres sont dérivés, soit au total 4400 filtres différents ce qui correspond environ au nombre d'enregistrements du corpus d'entraînement de TIMIT. Pour chaque canal, la Figure 41 montre le filtre moyen obtenu ainsi que le « minimum » et « maximum » des réponses générées.

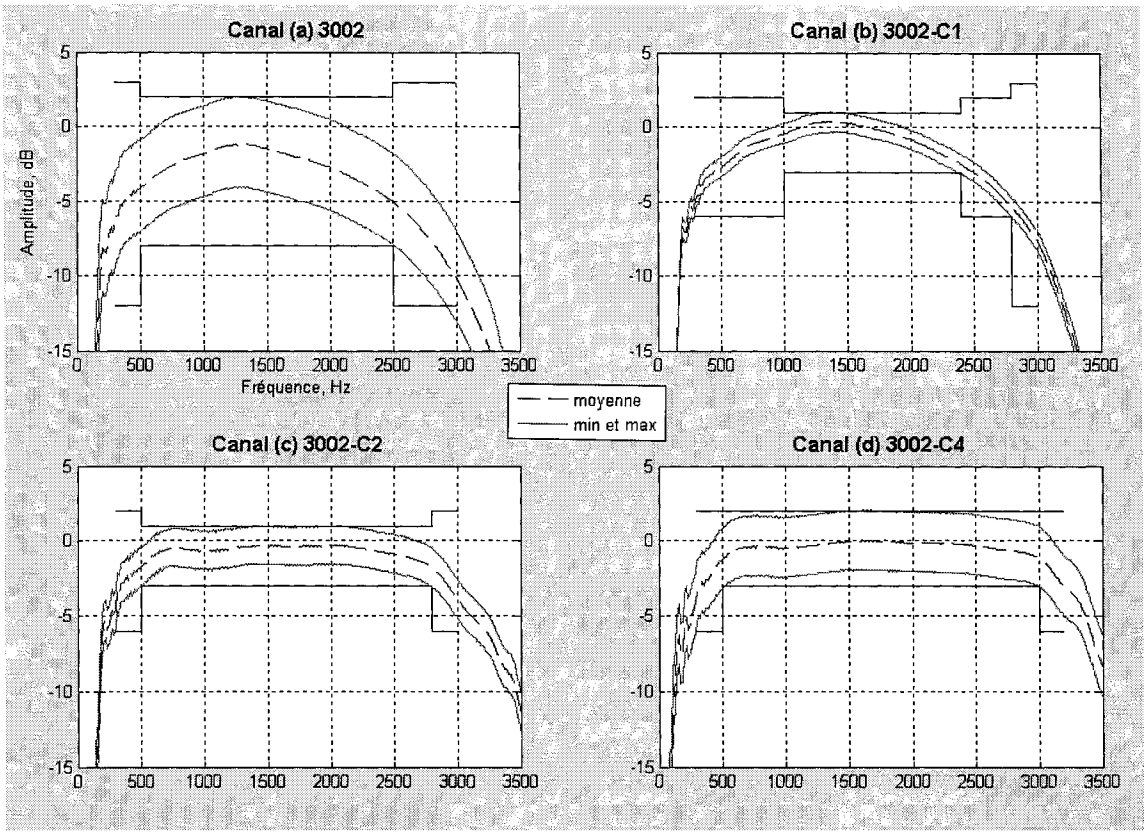

Figure 41 Réponse en amplitude des filtres générés avec leur gabarit respectif

Avant de décrire la méthode développée dans le cadre de cette maîtrise, attardons-nous un moment sur la base de données TIMIT qui a servi comme corpus d'apprentissage et de test pour évaluer les performances du système de reconnaissance.

4.2 Description de la base de données utilisée

Une des raisons expliquant la popularité de TIMIT réside dans le fait que les chercheurs y ont trouvé une base de référence commune pour comparer leurs résultats.

La base de données TIMIT [43] est très utilisée dans le domaine de la reconnaissance de la parole. TIMIT illustre très correctement la variabilité de la langue américaine en

prenant notamment en compte 8 dialectes (8 accents) communément utilisés dans diverses régions des États-Unis. La provenance des 630 personnes qui ont participé à l'étude est représentée sur le Tableau II.

Les différentes régions (dr pour *Dialect Region*) sont :

- dr1: New England
- dr2: Northern
- dr3: North Midland
- dr4: South Midland
- dr5: Southern
- dr6: New York City
- dr7: Western
- dr8: Army Brat

Tableau II

Distribution des dialectes [43]

Dialect Region(dr)	#Male	#Female	Total
1	31 (63%)	18 (27%)	49 (8%)
2	71 (70%)	31 (30%)	102 (16%)
3	79 (67%)	23 (23%)	102 (16%)
4	69 (69%)	31 (31%)	100 (16%)
5	62 (63%)	36 (37%)	98 (16%)
6	30 (65%)	16 (35%)	46 (7%)
7	74 (74%)	26 (26%)	100 (16%)
8	22 (67%)	11 (33%)	33 (5%)
8	438 (70%)	192 (30%)	630 (100%)

La base TIMIT a été enregistrée par *Texas Instrument* (TI), transcrit par le *Massachusetts Institute of Technology* (MIT) et a été maintenue, vérifiée et mise en CD-ROM par le *National Institute of Standards and Technology* (NIST).

TIMIT est une base multi-locuteurs contenant une phrase par enregistrement. Au total, 630 américains ont chacun lu 10 phrases, pas forcément les mêmes. Le choix des phrases s'est opéré comme suit :

- 2 phrases dites de calibration, prononcées par tous, servant à illustrer les variations dialectiques régionales (identifiées `sa1.wav` et `sa2.wav`);
- 5 phrases tirées au sort parmi 450 phrases et prononcés par 7 locuteurs différents. Ces phrases présentent un intérêt particulier au niveau de leur construction phonétique (identifiées `sx3.wav` à `sx452.wav`);
- 3 phrases choisies pour mettre en évidence la diversité acoustique et phonétique (identifiées `si453.wav` à `si2342.wav`).

On recense 6100 mots différents dans la base.
La répartition globale homme-femme est la suivante : 438 hommes et 192 femmes.

- dans le corpus d'apprentissage: 326 hommes et 136 femmes.
- dans le corpus de test: 112 hommes et 56 femmes.

Chaque locuteur est identifié par une lettre indiquant son genre ("m" pour les hommes et "f" pour les femmes), ses 3 initiales et un chiffre.

Pour chacun des 6300 enregistrements, nous disposons, en outre, de 3 autres fichiers du même nom avec les extensions suivantes :

- `.txt` : Transcription orthographique de la phrase prononcée + durée totale de l'enregistrement (Figure 43);
- `.phn` : Transcription orthographique de la phrase en phonèmes + durée de chaque phonème (Figure 44);
- `.wrd` : Transcription orthographique de la phrase en mots + durée de chaque mot (Figure 45).

A noter que la durée ici est exprimée en numéro de coefficients. Ainsi l'enregistrement `sa1.wav` (Figure 42) contient 46797 échantillons. Donc comme la base est échantillonnée à 16 kHz, le fichier `sa1.wav` dure 46797 / 16000 = 2.925 secondes.

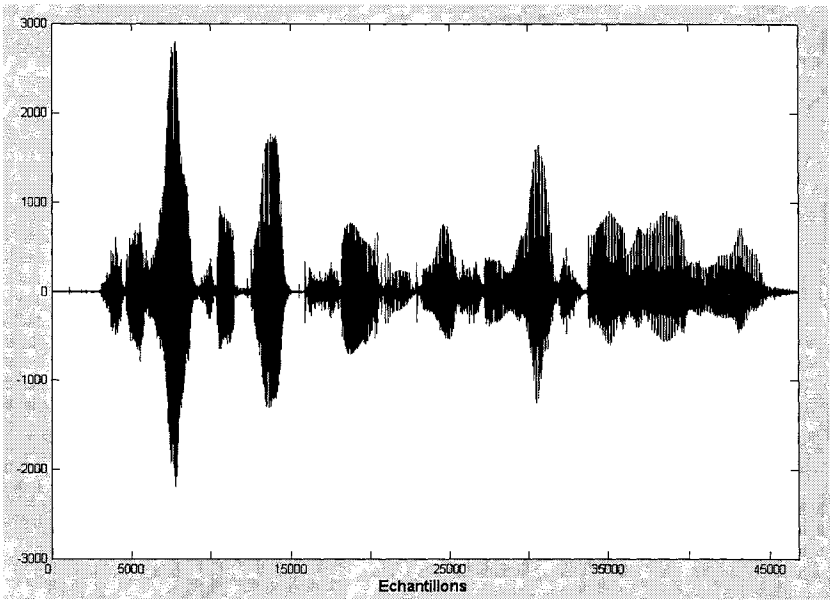

Figure 42 Exemple avec le fichier `sa1.wav`

```
0 46797 She had your dark suit in greasy wash water all year.
```

Figure 43 Le fichier `sa1.txt`

```
0 3050 h#              22920 23271 g
3050 4559 sh           23271 24229 r
4559 5723 ix           24229 25566 ix
5723 6642 hv           25566 27156 s
6642 8772 eh           27156 28064 ix
8772 9190 dcl          28064 29660 w
9190 10337 jh          29660 31719 ao
10337 11517 ih         31719 33360 sh
11517 12500 dcl        33360 33754 epi
12500 12640 d          33754 34715 w
12640 14714 ah         34715 36080 ao
14714 15870 kcl        36080 36326 dx
15870 16334 k          36326 37556 axr
16334 18088 s          37556 39561 ao
18088 20417 ux         39561 40313 l
20417 21199 q          40313 42059 y
21199 22560 en         42059 43479 ih
22560 22920 gcl        43479 44586 axr
                       44586 46720 h#
```

Figure 44 Le fichier `sa1.phn`

```
3050 5723 she
5723 10337 had
9190 11517 your
11517 16334 dark
16334 21199 suit
21199 22560 in
22560 28064 greasy
28064 33360 wash
33754 37556 water
37556 40313 all
40313 44586 year
```

Figure 45 Le fichier `sa1.wrd`

Au niveau de la transcription phonétique, la base TIMIT utilise 61 phonèmes différents. Le Tableau III, tiré de [14] et inspiré par [44], recense tous les phonèmes utilisés dans TIMIT avec leur correspondant dans l'Alphabet Phonétique International (API) suivi d'un exemple de la langue anglaise.

Tableau III

Les 61 phonèmes présents dans la base TIMIT

TIMIT	API	Exemple	TIMIT	API	Exemple	TIMIT	API	Exemple
Occlusives:			*Nasales*:			*Voyelles*:		
pcl p	p	pea	m	m	mom	iy	i^y	beet
tcl t	t	tea	em	m_i	bottom	ih	ɪ	bit
kcl k	k	key	n	n	noon	ix	I	debit
bcl b	b	bee	nx	\$	winner	eh	E	bet
dcl d	d	day	en	n_i	button	ae	æ	bat
gcl g	g	gay	ng	N	sing	aa	A	bott
dx	\	muddy	eng	N_i	washington	ao	O	bought
q	?	bat	*Liquides:*			uh	Ʊ	book
Affriquées:			l	l	lay	uw	u	boot
dcl jh	j&	joke	el	l_i	bottle	ux	ʉ	toot
tcl ch	c&	choke	r	r	ray	ax	ə	about
Fricatives:			*Semi-voyelles*:			ax-h	ə̥	suspect
f	f	fin	w	w	way	ah	ʌ	but
th	Q	thin	y	y	yacht	er	ɝ	bird
s	s	sea	*Fricative glottale*:			axr	ɚ	butter
sh	s&	she	hh	h	hay	*Diphtongues*:		
v	v	van	hv	H	ahead	ey	e^y	bait
dh	D	then	*Silences*:			ay	A^y	bite
z	z	zone	h#			oy	O^y	boy
zh	z&	azure	pau			aw	A^w	bout
			epi			ow	o^w	boat

Toutefois, l'utilisation de 61 phonèmes est jugée trop détaillée pour son utilisation avec HTK. Celui-ci nous suggère de passer à un ensemble de 48 phonèmes où les arrêts glottaux sont supprimés, où les occlusives précédant un arrêt voisé sont remplacées par une occlusive voisée générique [vcl], où toutes les occlusives précédant un arrêt non voisé sont remplacées par une occlusive non voisée générique [cl] et où enfin les différents types de silences sont regroupés dans une seule étiquette de silence [sil].

Nous présentons dans le Tableau IV [14], des statistiques sur les 48 classes phonétiques. Les regroupements décrits précédemment y sont illustrés. Aussi, pour chaque classe phonétique, on donne le nombre de représentants dans l'ensemble d'apprentissage ainsi

que sa durée moyenne. Enfin, les phonèmes différents mais considérés équivalents pendant la phase de reconnaissance sont regroupés par une accolade.

Tableau IV

Les 48 phonèmes utilisés dans la reconnaissance

Etiquette(s)	Nombre	Durée (ms)	Etiquette(s)	Nombre	Durée (ms)
Occlusives:			*Semi-voyelles*:		
b	2181	17	w	2216	60
d	2432	24	y	995	54
g	1191	27	*Fricative glottale*:		
p	2588	44	hh,hv	1660	67
t	3948	49	*Voyelles*:		
k	3794	52	iy	4626	95
dx	1864	29	ih	4248	78
Affriquées:			ix	7370	51
jh	1013	61	eh	3277	93
ch	820	86	ae	2292	136
Fricatives:			aa	2256	123
f	2215	103	ao	1865	123
th	745	92	uh	500	76
s	6176	113	uw,ux	1952	100
sh	1317	118	ax,ax-h	3892	47
zh	149	81	ah	2266	89
v	1994	60	er,axr	4138	95
dh	2376	36	*Diphtongues*:		
z	3682	84	ey	2271	127
Nasales:			ay	1934	155
m,em	3566	65	oy	304	168
n,nx	6896	52	aw	728	161
en	630	78	ow	1653	128
ng,eng	1220	61	*Silences*:		
Liquides:			sil=(h#,pau)	8283	191
l	4425	61	cl=(pcl,tcl,kcl)	12518	58
el	951	90	vcl=(bcl,dcl,gcl)	7219	54
r	4681	56	epi	908	42

La base NTIMIT quant à elle, possède le même contenu que la base de données TIMIT mais les enregistrements ont eu lieu, dans des conditions réelles, sur de vrais réseaux téléphoniques commutés. En fait, NTIMIT c'est TIMIT mais avec de la parole téléphonique.

4.3 Description des tests de reconnaissance

Les bases de données TIMIT et NTMIT sont échantillonnées à 16 kHz. Chaque enregistrement a donc un spectre compris entre 0 et 8 kHz. Puisque la bande passante du canal téléphonique est limitée à 3.4 kHz, les bases TIMIT et NTIMIT seront sous-échantillonnées à 8 kHz. Pour respecter les conditions de Shannon pour le sous-échantillonnage de TIMIT, il faut normalement faire précéder l'échantillonnage d'un filtre anti-repliement (Figure 46). Nous avons utilisé, pour le filtre anti-repliement, les travaux de Grassi dans [45] : le filtre en question est un filtre RIF d'ordre 158 à phase linéaire avec une atténuation de +97 dB dans la bande d'arrêt.

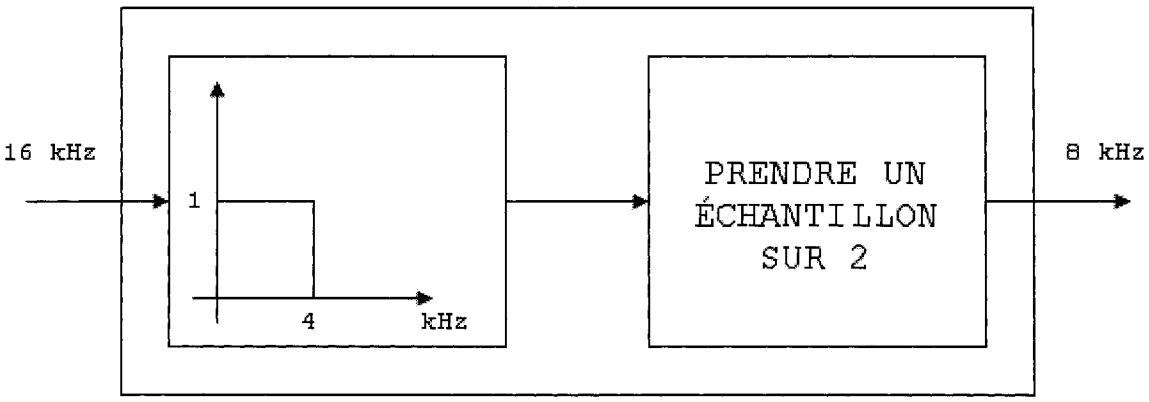

Figure 46 Principe du sous-échantillonnage de TIMIT

Pour la suite, la base TIMIT originale à 16 kHz sera référencée par TIMIT 16k, tandis que la base sous échantillonnée le sera par TIMIT 8k.

La démarche et les tests effectués sont synthétisés sur la Figure 47. On utilise comme type de résultat le pourcentage d'erreur (WER pour *Word Error Rate*) obtenu grâce au taux de phonèmes correctement reconnus (*Accuracy Rate*). Les erreurs d'insertions y sont également prises en compte comme décrit au Chapitre 3. Le taux d'erreurs est préféré à l'*Accuracy* dans la littérature lorsqu'on cherche à améliorer la robustesse des systèmes.

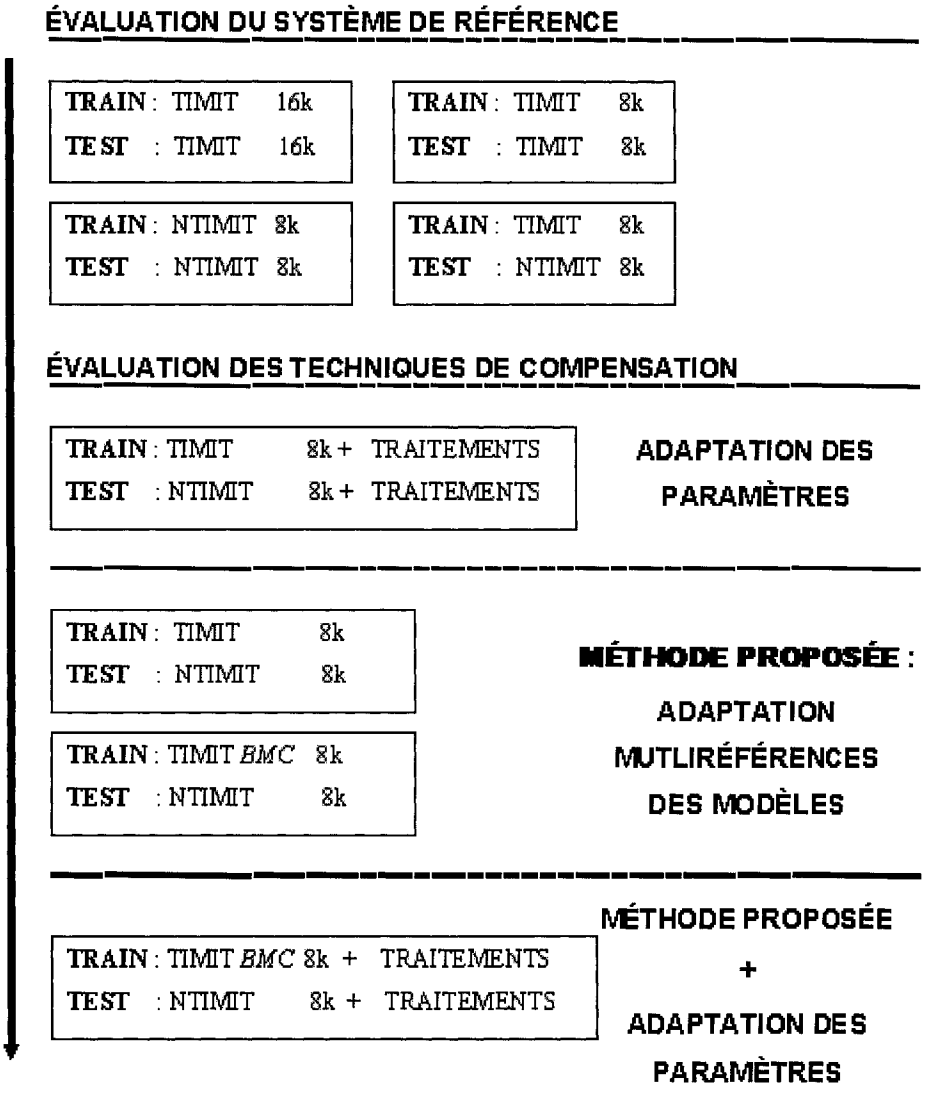

Figure 47 Déroulement des opérations de tests

4.3.1 Evaluation du système de référence

On rappelle que le système de reconnaissance développé est celui décrit au Chapitre 3. Le Tableau V contient les résultats de notre système de référence. Sur TIMIT 16k, on obtient 35.45 % d'erreurs. On peut rencontrer dans la littérature [39, 46] des taux tout à fait similaires pour la reconnaissance de la parole continue à grands vocabulaires indépendante du locuteur.

Les deux premières rangées du tableau illustrent l'effet de la réduction de la bande passante suite au sous-échantillonnage. Seulement 0.95 % de différence entre TIMIT 16k et TIMIT 8k.

On observe une augmentation relative du taux d'erreurs si le système est entraîné avec de la parole téléphonique (48.11 %). Le résultat devient mauvais (72.53 %) s'il est entraîné avec de la donnée propre et testé sur la parole téléphonique. Cela illustre bien les généralités évoquées en introduction lorsque les données d'apprentissage et de test présentent de trop grandes disparités.

Tableau V

Résultats du système de référence

TRAIN	TEST	WER (%)
TIMIT 16k	TIMIT16k	35.45
TIMIT 8k	TIMIT 8k	36.40
NTIMIT 8k	NTIMIT 8k	48.11
TIMIT 8k	NTIMIT 8k	72.53

4.3.2 Evaluation des techniques de compensation

Nous avons implémenté les techniques de compensation décrites au Chapitre 2 pour tenter d'améliorer le taux d'erreurs obtenue sur TIMIT/NTIMIT. Nous avons également joint à ces techniques une méthode nouvelle d'adaptation des modèles.

4.3.2.1 Adaptations des paramètres

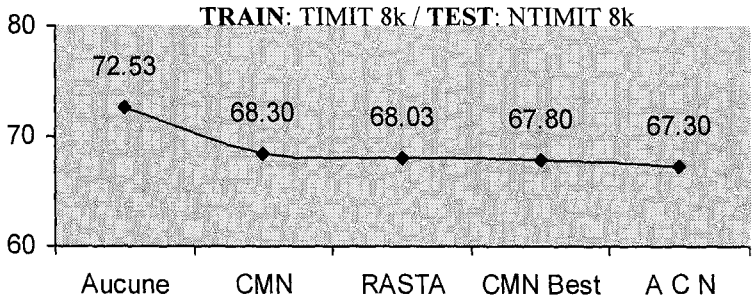

Figure 48 Résultats des techniques paramétriques de compensation

Les techniques « simples » de compensation que sont les techniques *CMN* et *RASTA* amènent des résultats intéressants comparées aux deux autres qui elles, demandent un effort de calcul supérieur en raison notamment de l'utilisation d'un détecteur d'activité de voix. Certes, *CMN Best* et *ACN* performent mieux mais la plus-value engendrée n'est pas énorme (1% environ).

4.3.2.2 Méthode proposée : apprentissage multiréférences

Au lieu d'entraîner le système avec la base TIMIT qui n'est pas du tout adaptée aux applications de reconnaissance via le réseau téléphonique commuté, nous allons nous

servir des modélisations de canaux imaginées en début de chapitre pour bruiter TIMIT et ainsi rendre la base plus proche de l'environnement que le système va rencontrer au cours de l'application. Nos modèles de Markov cachés seront donc appris avec de la parole bruitée (Figure 49).

Figure 49 Processus d'obtention de la nouvelle base TIMIT *BMC*

Ainsi, la base TIMIT bruitée par ces filtres peut être vue comme une nouvelle base d'entraînement multiréférences puisqu'aucun des enregistrements n'est filtré par la même modélisation du canal. Nous appellerons cette nouvelle base TIMIT *BMC* (Bruitée par une Modélisation de Canal téléphonique, modélisations imaginées à la section 4.1).

Le Tableau VI montre que l'adaptation de la base d'entraînement par une base adaptée au canal téléphonique, et aux bruits qu'il apporte, permet de diminuer le taux d'erreurs de 8.27 %. Ainsi, on réduit le non appariement des données d'apprentissage (TIMIT *BMC*) avec celles de test (NTIMIT).

Tableau VI

Résultats de la méthode proposée

TRAIN	**TEST**	**WER (%)**
TIMIT 8k	NTIMIT 8k	72.53
TIMIT *BMC* 8k	NTIMIT 8k	64.26

4.3.2.3 Méthode proposée combinée à l'adaptation des paramètres

RASTA ne performe pas aussi bien que CMN et accroît le taux d'erreurs (64.72%). Avec ACN, le WER atteint 61.98%. Sans adaptation de modèles, nous avions obtenu 67.30%, soit une augmentation de 5.32%. Combiner l'adaptation des modèles d'entraînement (TIMIT Train filtré par nos modélisations de canaux PSTN) avec la compensation des paramètres (normalisation du canal) a permis d'améliorer la robustesse du système même si le WER est toujours élevé (61.98 %).

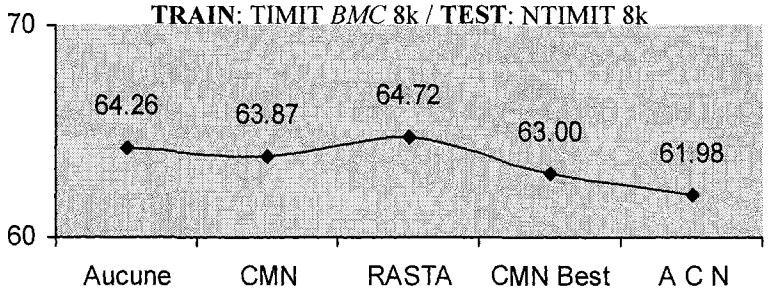

Figure 50 Résultats : Méthode proposée + Adaptations des paramètres

On pourrait sans doute encore améliorer les résultats en considérant les bruits additifs et les autres bruits convolutifs introduits par les éléments du réseau ainsi que par les dispositifs d'acquisition du signal vocal. En effet, les enregistrements NTIMIT, qui ont servi comme base de test, contiennent de la « vraie parole téléphonique » et donc renferment toutes sortes de bruits additifs et convolutifs (voir Chapitre 2). Il est donc logique que le taux d'erreurs, même après compensation du canal, soit élevé puisque notre base d'apprentissage TIMIT *BMC* ne modélise que la réduction de la bande passante par le canal et nul autre bruit.

CONCLUSION

La tâche de reconnaître la parole humaine est une tache difficile, complexe, et très sensible aux dégradations apportées par le milieu d'acquisition et la propagation du signal vocal.

Le signal enregistré, en entrée du système de reconnaissance, contient non seulement la parole du locuteur, mais également des bruits additifs dus à l'environnement sonore de la prise de son, et des distorsions liées au microphone et surtout la réduction de la bande passante par le canal de transmission. Ce dernier a fait l'objet d'une étude approfondie pour connaître les différents bruits présents sur le réseau commuté.

Dans le cadre de ce projet, nous avons étudié l'effet convolutif du canal qui entraîne une réduction du spectre du signal d'origine et donc modifie sensiblement les propriétés spectrales de la parole. Les bruits additifs et autres bruits liés aux microphones ont été ignorés dans la construction des filtres modélisant des canaux téléphoniques. À partir de 4 filtres prototypes représentants 4 grands types de canaux téléphoniques, une multitude de canaux téléphoniques différents a été dérivée.

Cela a débouché sur la construction d'une nouvelle base d'entraînement multiréférences. En apprenant le système de reconnaissance à l'aide de ce nouveau corpus, nous avons réduit le non-appariement des données de test (parole téléphonique) avec celles d'entraînement (parole propre) et ainsi pu réduire le taux d'erreurs de manière significative. Ces travaux ont donné lieu à une publication [3].

Pour améliorer encore la robustesse du système de reconnaissance, nous avons adjoint à la méthode proposée des méthodes de normalisation du canal (adaptation des paramètres) parmi lesquelles la technique *Augmented Cepstral Normalization* a donné les meilleurs résultats.

BIBLIOGRAPHIE

[1] Le Monde, *"Quand les puces ont des problèmes de syntaxe..."*. Journal Le Monde, 5 Octobre 1994.

[2] Mariani J., *La reconnaissance de la parole.* La recherche en intelligence artificielle, Points Science, Éditions du Seuil, pages 119-147, 1987.

[3] Preiss R., Gabrea M., *Modeling of the PSTN channel and multireferences training in robust speech recognition.* IEEE Int. Symp. on Industrial Electronics, ISIE'06, 2006.

[4] Jelinek F., Bahl L.R., Mercer L., *Design of a linguistic statistical decoder for the recognition of continuous speech.* IEEE Transactions on Information Theory, pages 250-256, 1975.

[5] Jelinek F., *Continuous speech recognition by statistical methods.* Proc. of the IEEE, pages 532-556, 1976.

[6] Lombard E., *Le signe de l'élévation de la voix.* Annales des maladies de l'oreille et du larynx, 1911.

[7] Junqua J.-C., Anglade Y., *Acoustic and perceptual studies of Lombard speech: application to isolated-words automatic speech recognition.* IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'90, pages 841-844, 1990.

[8] Rabiner L.R., Gold B., *Theory and Application of Digital Signal Processing.* Englewood Cliffs, New-Jersey: Prentice Hall, 1975.

[9] Deller J.R., Proakis J.G., Hansen J.H.L., *Discrete-Time Processing of Speech Signals.* 1993.

[10] Young S. et al., *The HTK Book*, Cambridge University Engineering Department, 2001.

[11] Vergin R., Farhat A., O'Shaughnessy D., *Generalized mel frequency cesptral coefficients for large vocabulary speaker independant continuous speech recognition*. IEEE Transactions on Speech and Audio Processing, 1999.

[12] Hermansky H., *Perceptual linear predictive analysis for speech*. Journal of the Acoustical Society of America, pages 1738-1752, 1990.

[13] Boite R., Kunt M., *Traitement de la parole*. Presses Polytechniques Romandes, 1987.

[14] Barras C., *Reconnaissance de la parole continue, Adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés*. Thèse de l'Université de Paris VI, 1996.

[15] Baker J.K., *The DRAGON system - An overview*. IEEE Transactions on Acoustics, Speech, and Signal Processing, pages 24-29, 1975.

[16] Baum L.E., *An inequity and associated maximization technique in statistical estimation for probabilistic functions of Markov processes*. Inequalities 3, pages 1-8, 1972.

[17] Brown P.F. et al, *Bayesian adaptation in speech recognition*. Proc. ICASP'83, pages 761-764, 1983.

[18] Rabiner L.R., *Speaker independant recognition of isolated words using clustering techniques*. Trans. IEEE ASSP-27, pages 336-349, 1979.

[19] Reynolds D.A., *A gaussian mixture modelling approach to text independent speaker identification*. Georgia Institute of Technology, 1992.

[20] Zikinf, http://www.zikinf.com/articles/home-studio/micros.php

[21] Gleiss N., *The effect of Bandwidth Restriction on Speech transmission Quality in Telephony*. Proc. of the fourth Int. Symp. on human factors in telephony, 1970.

[22] Moreno P.J., Stern R.M., *Sources of degradation of speech recognition in the telephone network*. Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, pages 109-112, 1994.

[23] Hunt M., *Further experiments in text-independent speaker recognition over communications channels*. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '83, pages 563-566, 1983.

[24] Gish H., *Investigation of text-independent speaker indentification over telephone channels*. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '85, pages 379-382, 1985.

[25] Furui S., *Recent advances in robust speech recognition*. ESCA-NATO Workshop on Robust speech recognition for unknown communication channels, pages 11-20, 1997.

[26] Furui S., *Cepstral analysis technique for automatic speaker verification*. IEEE Trans. Acoust. Speech Signal Processing, pages 254-272, 1981.

[27] Hermansky H., Morgan N., *RASTA Processing of Speech*. IEEE Trans. On Speech and Audio Processing, pages 578-589, 1994.

[28] Lippmann R., Martin E., Paul D., *Multi-style training for robust isolated-word speech recognition*. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pages 705-708, 1987.

[29] Varga A., Moore R., *Hidden Markov Model decomposition of speech and noise*. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pages 845-848, 1990.

[30] Liu F.H. et al., *Signal processing for robust speech recognition*. Proc. of the 1994 Int. Conf. on Spoken Language Processing, ICSLP '94, pages 330-335, 1994.

[31] Duvaut P., *Le point sur les méthodes de séparation des sources*. Traitement du signal, pages 407-418, 1990.

[32] Widrow B., Glover J., *Adaptative noise cancelling: principles and applications*. Proc. of the IEEE, pages 1692-1716, 1975.

[33] Klatt D., *A digital filter bank for spectral matching*. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '76, pages 573-576, 1976.

[34] Mockbel C. et al., *Compensation of Telephone Line Effects for Robust Speech Recognition*. ICSLP 94, 1994.

[35] Puel J.-B., *Reconnaissance automatique de la parole téléphonique: adaptation au G.S.M.* Université Paul Sabatier, Toulouse, 1997.

[36] Rabiner L.R., Sambur M.R., *An algorithm for determining the endpoints of isolated utterances*. The Bell System Technical Journal, 1975.

[37] Craciun A., *Implémentation d'une méthode robuste de détection d'activité vocale sur le processeur de signal TMS320C6711*. Maîtrise en génie électrique, École de Technologie Supérieure, Montréal, 2004.

[38] Prasad R.V., Sangwan A., Jamadagni H.S., *Comparison of Voice Activity Detection Algorithms for VoIP*. IEEE Proc. of the Seventh Int. Symp. on Computers and Communications, ISCC'02, 2002.

[39] Acero A., Huang X., *Augmented Cepstral Normalization for Robust Speech Recognition*. Proc. of the IEEE Workshop on Automatic Speech Recognition, 1995.

[40] Cygwin. *www.cygwin.com,* Version 1.5.18-1.

[41] Proakis J.G., *Digital Communications*. pages 520-528, 1989.

[42] Gerald C.F., Wheatly P.O., *Applied Numerical Analysis*. Addison-Wesley, 1999.

[43] T.I. & M.I.T., *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) (CD-ROM)*, 1990.

[44] Seneff S., Zue V., *Transcription and alignment of the TIMIT database, in Getting started with the DARPA TIMIT CD-ROM: an acoustic-phonetic continuous speech database*. NIST, 1988.

[45] Besacier L., Grassi S., Dufaux A., *GSM speech coding and speaker recognition*. IEEE Proc. of. ICASSP'00, 2000.

[46] Junqua J.-C., *Impact of the unknown communication channel on automatic speech recognition: A review*. Eurospeech'97, 1997.