Reconnaissance et transformation de locuteurs

(Quelle identité par la parole?)

Thèse

présentée à la section

Systèmes de Communication

de

l'Ecole Polytechnique Fédérale de Lausanne (EPFL)

par

Dominique Genoud

Le jury:
Prof. Martin Hasler EPFL
Prof. Gérard Chollet ENST-Paris
Prof. Christian Wellekens EPFL-EURECOM
Dr. Régine André-Obrecht IRIT

Table des matières

Résumé

Comment analyser, décomposer, modéliser et transformer l'identité vocale d'une personne lorsqu'elle est vue au travers d'une application de reconnaissance automatique du locuteur définit la motivation principale de cette thèse. Elle débute par une introduction qui explique les propriétés du signal de parole et les bases de la reconnaissance automatique du locuteur. Puis, elle analyse les erreurs d'un système de reconnaissance du locuteur en exploitation. Des carences et des erreurs relevées dans cette application, on tire des constatations permettant de réévaluer les paramètres qui caractérisent un locuteur et de reconsidérer plusieurs éléments de la chaîne de reconnaissance automatique du locuteur.

Un modèle d'analyse/synthèse harmoniques plus bruit (H+N) a servi à extraire du signal de parole les paramètres caractéristiques d'un locuteur. L'analyse et la re-synthèse des parties harmoniques et bruit ont permis de découvrir quelles sont les parties du signal de parole dépendant du locuteur plutôt que de la parole. Cette thèse montre que l'information discriminante d'un locuteur se trouve dans la partie résiduelle de la soustraction du signal par l'analyse/synthèse H+N. La thèse aborde ensuite l'étude des phénomènes d'imposture, essentiels dans le réglage d'un système de reconnaissance du locuteur. Les imposteurs ont été simulés automatiquement de deux manières: soit en transformant la voix d'un locuteur source (l'imposteur) en celle d'un locuteur cible (le client) et en agissant sur les paramètres extraits par le modèle H+N. Cette transformation est efficace puisque les taux d'erreur de fausses acceptations passent de 4% à 23%. Une méthode d'imposture automatique par concaténation de segments de parole a aussi été utilisée, méthode pour laquelle des taux de fausses acceptations de plus de 30% ont été atteints. De manière à se rendre robuste à ces tentatives d'impostures, tout au moins à celles qui utilisent la transformation spectrale, la partie harmonique du système H+N puis les parties harmoniques et bruit ont été supprimées du signal de parole original, permettant ainsi de ramener les fausses acceptations à environ 8% malgré l'utilisation d'imposteurs transformés.

Pour pallier au manque de données d'entraînement, une des causes importantes d'erreur de modélisation de locuteurs, une approche par décomposition de la tâche de reconnaissance en problèmes à 2 classes est proposée. Une matrice de classificateurs est construite et chacun de ses éléments doit classer, mot par mot, les données d'un locuteur client et d'un autre locuteur (nommé anti-client, choisi aléatoirement dans une base de données externe à l'application). Cette méthode permet de pondérer les résultats en fonction du vocabulaire prononcé par le locuteur ou de ses voisins dans l'espace des paramètres. La recombinaison des éléments de la matrice s'effectue ensuite par pondération des sorties de chacun des classificateurs. Cette pondération est estimée sur des données de réglage et, si elle est bien choisie, le système de reconnaissance par paires binaires donne des performances en fonctionnement souvent meilleures que le système de référence HMM à l'état de l'art.

De manière à régler un point de fonctionnement d'une application de reconnaissance du locuteur, il est nécessaire de déterminer un seuil de décision *a priori*. Si théoriquement le seuil est indépendant du locuteur lorsque les modèles utilisés sont stochastiques, il en va tout autrement dans la pratique. En effet, l'imperfection de la modélisation provoque une dépendance du seuil au locuteur et à la longueur d'une observation. Il est cependant possible de calculer cet ajustement du seuil pour chaque locuteur en se basant sur les rapports de vraisemblances locales.

Finalement, une dernière méthode de correction des erreurs de modélisation par la fusion de décisions de différents experts est proposée. Quelques cas pratiques montrent le profit et les limitations qu'une telle approche apporte aux systèmes de reconnaissance de locuteur.

Abstract

This PhD thesis tries to understand how to analyse, decompose, model and transform the vocal identity of a human when seen through an automatic speaker recognition application. It starts with an introduction explaining the properties of the speech signal and the basis of the automatic speaker recognition. Then, the errors of an operating speaker recognition application are analysed. From the deficiencies and mistakes noticed in the running application, some observations can be made which will imply a re-evaluation of the characteristic parameters of a speaker, and to reconsider some parts of the automatic speaker recognition chain.

In order to determine what are the characterising parameters of a speaker, these are extracted from the speech signal with an analysis and synthesis harmonic plus noise model (H+N). The analysis and re-synthesis of the harmonic and noise parts indicate those which are speech or speaker dependent. It is then shown that the speaker discriminating information can be found in the residual of the subtraction from the original signal of the H+N modeled signal. Then, a study of the impostors phenomenon, essential in the tuning of a speaker recognition system, is carried out. The impostors are simulated in two ways: first by a transformation of the speech of a source speaker (the impostor) to the speech of a target speaker (the client) using the parameters extracted from the H+N model. This way of transforming the parameters is efficient as the false acceptance rate grows from 4% to 23%. Second, an automatic imposture by speech segment concatenation is carried out. In this case the false acceptance rate grows to 30%. A way to become less sensitive to the spectral modification impostures is to remove the harmonic part or even the noise part modeled by the H+N from the original signal. Using such a subtraction decreases the false acceptance rate to 8% even if transformed impostors are used.

To overcome the lack of training data – one of the main cause of modeling errors in speaker recognition – a decomposition of the recognition task into a set of binary classifiers is proposed. A classifier matrix is built and each of its elements has to classify word by word the data coming from the client and another speaker (named here an anti-speaker, randomly chosen from an external database). With such an approach it is possible to weight the results according to the vocabulary or the neighbours of the client in the parameter (acoustic) space. The output of the matrix classifiers are then weighted and mixed in order to produce a single output score. The weights are estimated on validation data, and if the weighting is done properly, the binary pair speaker recognition system gives better results than a state of the art HMM based system.

In order to set a point of operation (i.e. a point on the COR curve) for the speaker recognition application, an *a priori* threshold has to be determined. Theoretically the threshold should be speaker independent when stochastic models are used. However, practical experiments show that this is not the case, as due to modeling mismatch the threshold becomes speaker and utterance length dependant. A theoretical framework showing how to adjust the threshold using the local likelihood ratio is then developed.

Finally, a last modeling error correction method using decision fusion is proposed. Some practical experiments show the advantages and drawbacks of the fusion approach in speaker recognition applications.

A Janick et Morgane

Remerciements

L'histoire d'une thèse, c'est aussi l'histoire d'un morceau de vie partagé avec de nombreuses personnes. Quelques-unes ont compté plus que d'autres, mais globalement toutes ont eu une importance dans ce morceau d'espace-temps, qu'elles en soient donc toutes remerciées. Je pense plus particulièrement à tous les gens de l'IDIAP qui m'ont supporté quotidiennement, ce qui ne fut pas nécessairement une tâche facile.

Je voudrais plus particulièrement remercier Gérard Chollet, sans qui cette thèse ne se serait pas faite. Ses corrections, suggestions et idées m'ont été précieuses. C'est grâce à la transmission de ses connaissances, à sa confiance et à sa rigueur scientifique que cette thèse a pu être écrite.

Mes remerciements vont aussi à Martin Hasler qui a bien voulu accepter la responsabilité de diriger cette thèse, ainsi qu'à tous les membres du jury qui ont accepté de lire ce document.

Je voudrais ensuite remercier Jean-Luc Cochard d'abord, Hervé Bourlard ensuite et puis finalement Chafic Mokbel pour avoir guidé le groupe parole de l'IDIAP durant ces années, me permettant de terminer cette thèse dans des conditions raisonnables.

Je tiens également à remercier, en plus de Gérard Chollet, Janick Genoud, Chafic Mokbel, Guillaume Gravier et Patrick Verlinde pour m'avoir aidé à corriger ce document.

Un grand merci encore à Georg Thimm, Eddy Mayoraz et, surtout, Miguel Moreira et Frédéric Gobry pour m'avoir aidé a dépasser ma latexophobie. Merci aussi à Olivier Bornet pour sa disponibilité et ses compétences aux commandes de l'informatique idiapienne.

Et finalement j'aimerais remercier L'OFES, le FNRS et Swisscom sans qui, je n'eusse point pu faire vivre ma famille durant cette période.

Chapitre 1

Introduction

Chaque être humain peut, dès son plus jeune âge, reconnaître les voix des personnes qui lui sont familières. Bien que le processus de reconnaissance de la parole soit fort développé chez l'homme, il ne lui est cependant pas immédiat de caractériser les indices qui permettent de distinguer un locuteur d'un autre. La figure 1.1 illustre ce problème en montrant que le signal de parole émis par un être humain transmet son identité en plus du message, ces deux entités étant intimement liées. D'évidence, l'interprétation du message transmis par la parole est fortement dépendante des interlocuteurs impliqués dans le processus de production et de reconnaissance de la parole. Le décodage des émotions ainsi que les variations physiologiques des intervenants peuvent en effet changer non seulement les caractéristiques physiques du signal de parole, mais également son sens.

La parole est certainement le moyen de communication directe entre humains qui est le plus sophistiqué. Les subtiles variations du langage sont capables de susciter chez l'auditeur non seulement une palette fort variée d'émotions et de sentiments, mais aussi une attention complète de son cerveau. Les ordinateurs et les logiciels qui se construisent actuellement, bien que capables de traiter énormément d'informations en un temps très court, n'ont pas encore la capacité de générer ou de comprendre les finesses de la parole humaine. Cependant, de nombreuses applications en reconnaissance de la parole sont déjà industrialisées, allant de la dictée vocale à la commande d'opérations diverses dans les navettes spatiales. De plus en plus, les entreprises de télécommunications et de services (banques, assurances), désireuses d'améliorer leur service à la clientèle, tentent d'introduire des applications basées sur les technologies de la parole. La palette de ces technologies est fort riche, partant de systèmes de reconnaissance de la parole entraînés pour un seul locuteur à des systèmes capables de reconnaître des centaines de milliers de mots. Dans un autre registre, un grand nombre de services demandent une reconnaissance de l'identité du locuteur (accès aux boîtes vocales, à des services par abonnements, consultation de comptes en banques, etc. . .). Finalement, pour que le dialogue homme-machine soit complet, le domaine de la synthèse de la parole essaie de produire de la voix humaine (ou y ressemblant fort) automatiquement.

Cette thèse est plutôt orientée sur la reconnaissance de l'identité d'un locuteur par sa voix. Cependant, comme toutes les technologies vocales font appel à différents aspects du même phénomène (la voix humaine), elles sont, par beaucoup d'aspects, indissociables, ce qui nous induira à traiter également des aspects de reconnaissance de la parole et d'analyse/synthèse de celle-ci.

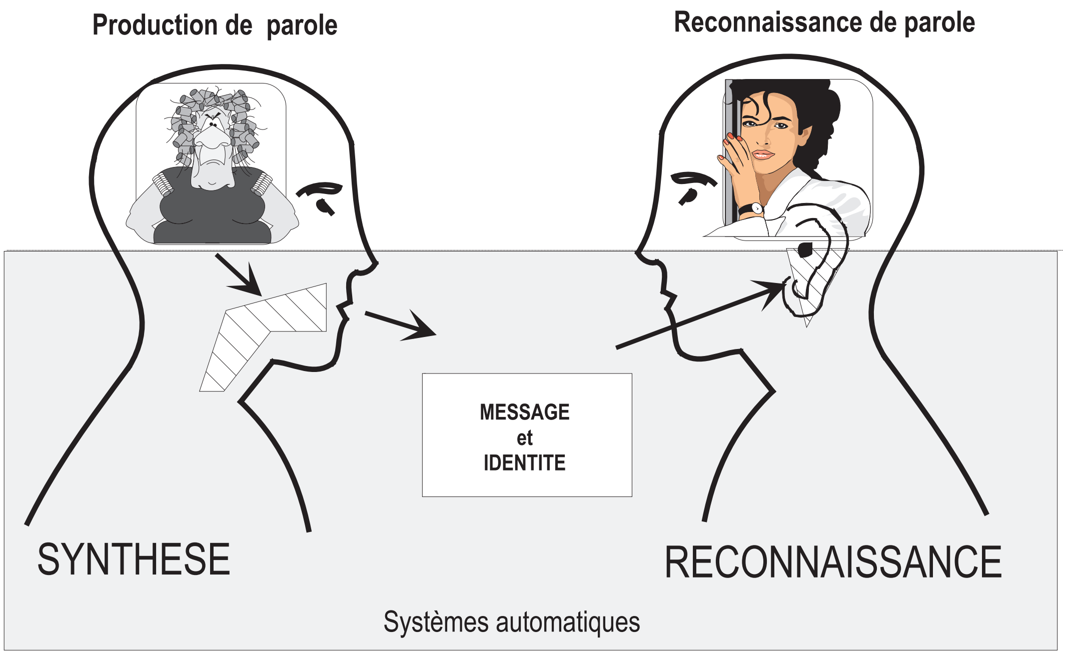

FIG. 1.1 – *Production et reconnaissance de la parole.*

Par mesure de simplification, nous supposerons que le message émis par un locuteur peut être considéré comme sans ambiguïtés pour un système de reconnaissance automatique de la *parole*.

Nous aborderons dans les sections suivantes de cette introduction l'analyse des caractéristiques du signal de parole ainsi que les caractéristiques principales des systèmes de reconnaissance du locuteur et d'analyse/synthèse de la voix. Et comme nous serons confrontés tout au long de cette thèse à certains aspects mathématiques de la théorie de la décision, nous en éclairerons quelques aspects utiles.

Enfin la section 1.4 exposera les idées apportées par cette thèse.

Précisons encore que cette thèse s'est effectuée dans le cadre de plusieurs projets européens et nationaux dont les références sont données dans l'annexe D.

1.1 Le signal de parole

La figure 1.2 (selon [Rabiner et Juang, 1993]) nous montre l'appareil phonatoire humain et les éléments qui le définissent. La commande de ces différents éléments physiologiques s'effectue à partir du cerveau lui-même soumis à des influences psychologiques pouvant modifier fortement le signal de parole lui-même (peur, colère, joie, etc...), voir par exemple [Homayounpour, 1995, Scherer *et al.*, 1998].

Lorsqu'on observe le signal de parole durant le temps (voir la figure 1.3) on peut constater de larges différences d'amplitude et de durée lors de la prononciation d'un même mot, d'un locuteur

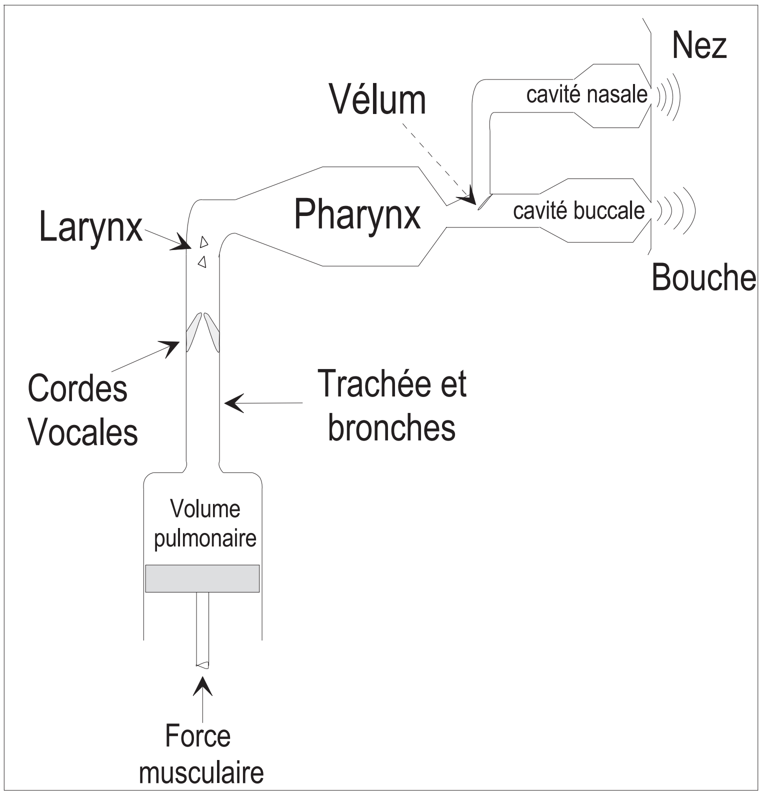

FIG. 1.2 – *Les différentes parties constituant le conduit vocal.*

à l'autre, mais aussi d'une prononciation à l'autre émanant du même locuteur. La transformation en temps/fréquences (figures 1.4 et 1.5) nous permet cependant de constater l'existence de lignes d'énergie à certaines fréquences. Si l'on utilise une analyse adaptée (spectrogamme large bande de la figure 1.5), on voit même apparaître des zones fréquencielles de forme similaire. Ce sont ces propriétés du signal que les phonéticiens et la **reconnaissance automatique de la parole** tentent d'exploiter. Comme ce signal fréquenciel transporte simultanément des informations sur l'identité du locuteur, l'analyse de ces mêmes motifs spectraux doit permettre d'effectuer une **reconnaissance du locuteur**.

1.1.1 Motifs spectro-temporels

L'analyse plus précise d'un spectrogramme permet de déterminer un espacement régulier entre les différentes lignes spectrales (voir la figure 1.6), représentant en fait les **harmoniques** d'une **fréquence fondamentale** (F_0). Cette fréquence fondamentale est directement liée à la source du signal de parole que sont les cordes vocales. La figure 1.8 (selon [Rabiner et Juang, 1993]) nous donne une idée de l'évolution de la vitesse du volume d'air à la sortie des cordes vocales. Nous le verrons par la suite, cette fonction est souvent modélisée par un train d'impulsion à la fréquence

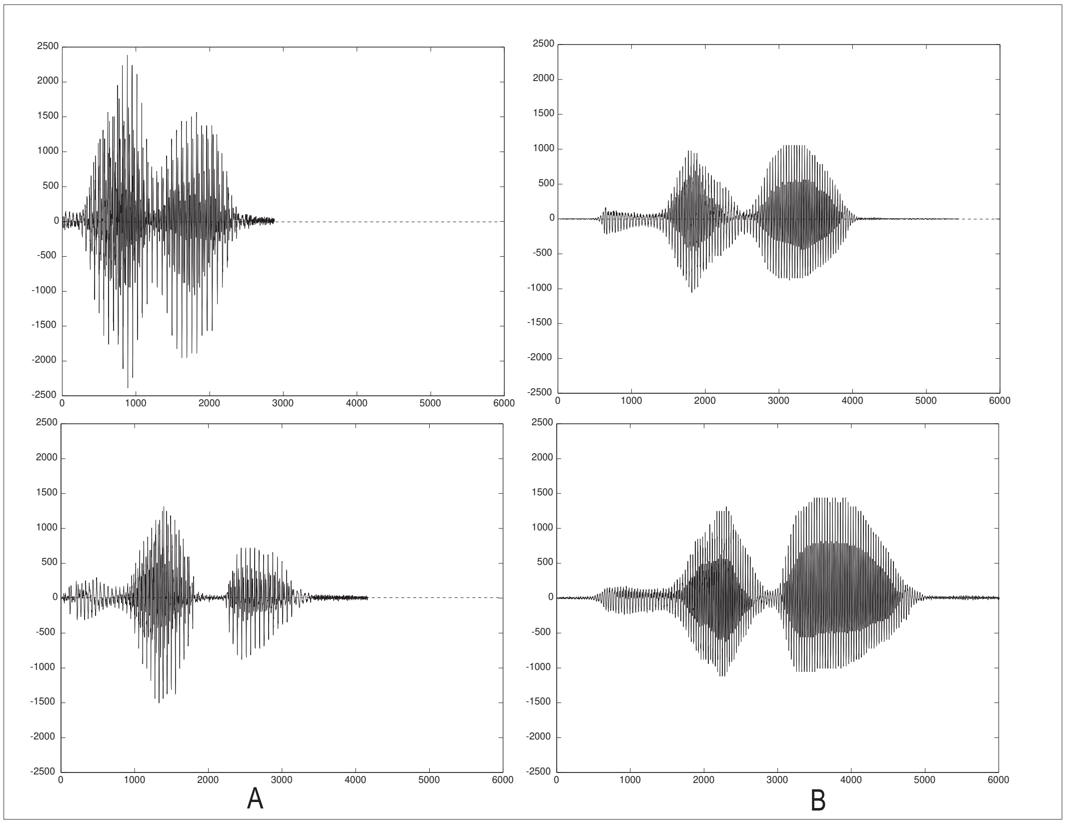

FIG. 1.3 – *Exemple de signaux parole prononciations du mot zéro: à gauche deux prononciations de la personne A à droite deux prononciations de la personne B.*

fondamentale F_0 .

L'inspection d'un spectrogramme généré à partir d'une analyse temps-fréquence en bande large permet de faire ressortir une accumulation d'énergie des harmoniques dans certaines zones. La figure 1.7 montre le déplacement du centre de ces zones, appelées **formants**. Les parties où la fréquence fondamentale existe sont appelées parties **voisées** et correspondent aux parties du signal où les cordes vocales sont en activité. On distingue aussi des parties du spectrogramme où il semble qu'aucun motif fréquenciel ne se dessine, elles correspondent aux régions où les cordes vocales ne vibrent pas.

Une des caractéristiques majeures du signal de parole est la variation de la valeur de la fréquence fondamentale selon l'état psychologique ou physiologique du locuteur et le sens que celui-ci veut donner à ce qu'il prononce. Cette variabilité pose un problème d'identification de motifs à des fréquences données. La figure 1.9 montre l'évolution de la F_0 et des formants tout au long d'une phrase (ici l'extrait de phrase "reconnaissance du locuteur?"). Remarquons les variations formantiques importantes dans la dernière partie du mot "locuteur" dues à la forme interrogative. Cette évolution de la fréquence fondamentale (et des formants) contribue à la **prosodie**, que l'on

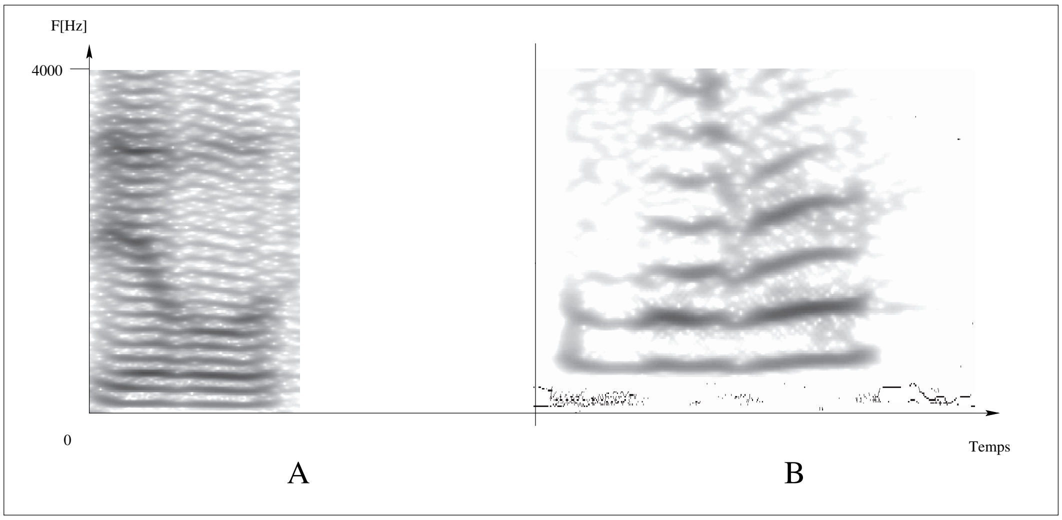

FIG. 1.4 – *Spectrogrammes en bande étroite pour le mot "zéro" des locuteurs A et B (extrait de la première prononciation du mot de la figure 1.3).*

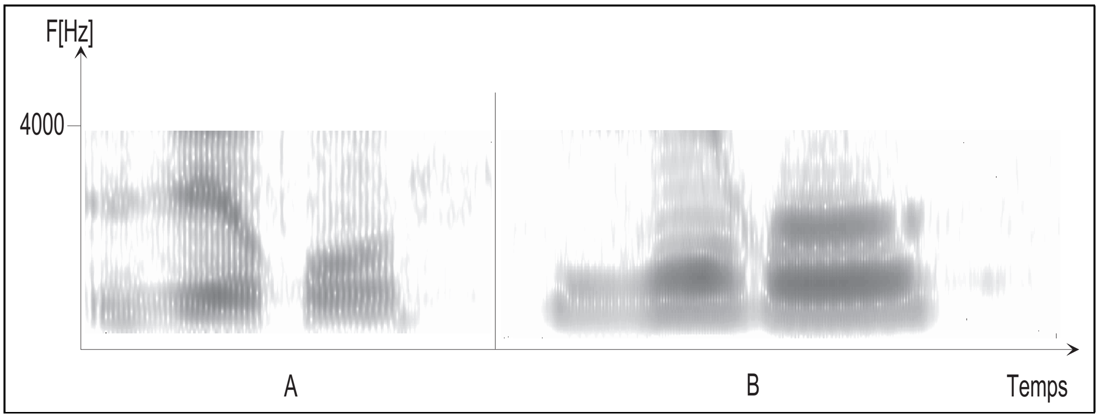

FIG. 1.5 – *Spectrogrammes en bande large pour le mot "zéro" des locuteurs A et B (extrait de la première répétition du mot de la figure 1.3).*

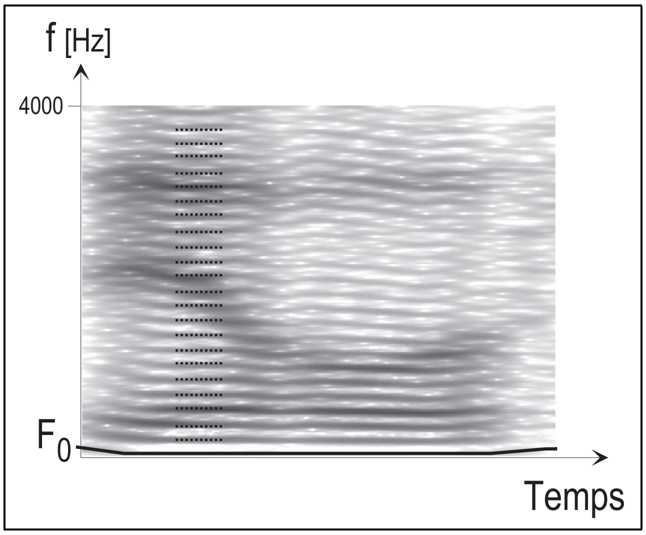

FIG. 1.6 – *Fréquence fondamentale (ligne pleine) et harmoniques (lignes pointillées) du mot "zé-ro".*

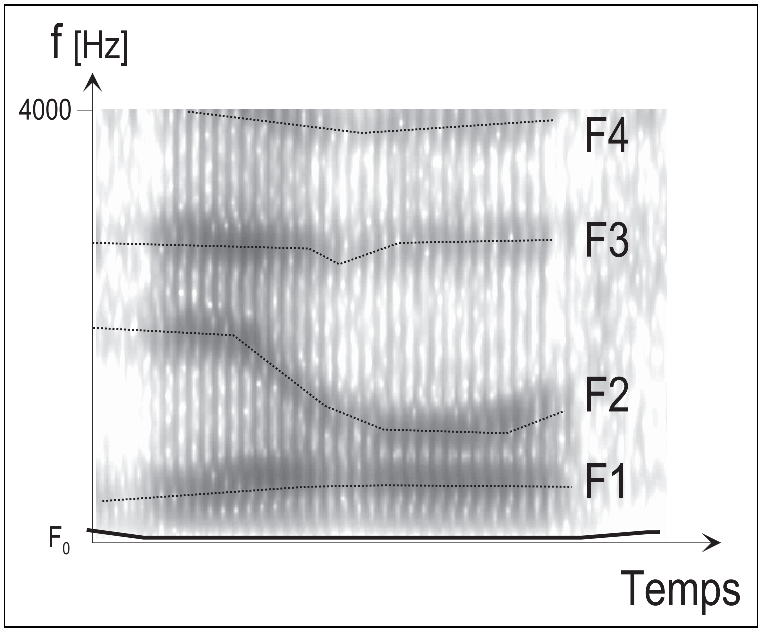

FIG. 1.7 – *Fréquence fondamentale (ligne pleine) et les 4 premiers formants (lignes pointillées).*

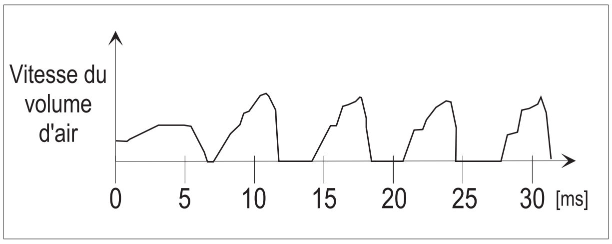

FIG. 1.8 – *Vitesse du volume d'air à la sortie des cordes vocales.*

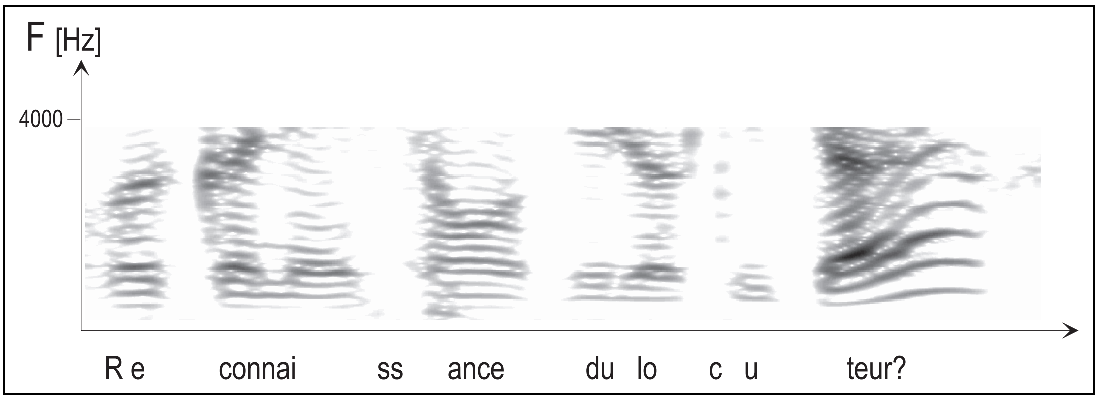

FIG. 1.9 – *Variation du spectre dû à la forme interrogative "Reconnaissance du locuteur?".*

pourrait aussi expliquer comme la *musique* de la parole.

1.1.2 Unités temporelles

Si l'on analyse la figure 1.9, on peut constater que certains motifs formantiques semblent se répéter. Ces motifs constituent les éléments d'une langue. Selon la durée à laquelle on les associe, on leur donne des noms différents. Les plus petites unités que les phonéticiens définissent sont des unités abstraites dont le corrélat acoustique est le **phonème**. Le nombre exact de phonèmes dépend des écoles, mais on en dénombre environ 35 dans la langue française, que l'on peut diviser en plusieurs types (ici aussi les classifications varient selon les phonéticiens) comme par exemple les voyelles (a, e, i, o, u), les fricatives (s,f), les plosives (p,t,k) ou les liquides (l). On peut définir des unités temporelles plus longues qui correspondent, par exemple, à des segments compris entre les maxima de stabilité spectrale de 2 phones consécutifs (diphones) ou des unités plus longues encore, tels les tri-phones, les poly-phones ou les syllabes.

1.2 Reconnaissance automatique du locuteur

Comme l'a montré la section précédente, la parole est un signal particulier, de par sa variabilité et sa richesse. C'est probablement pour cela que depuis plus de 30 ans, de nombreux chercheurs se sont penchés sur sa reconnaissance automatique sans vraiment parvenir à résoudre le problème complètement (voir par exemple [Doddington, 1976, Atal, 1976, Corsi, 1981, Doddington, 1985, Naik et Doddington, 1987, Naik, 1994]). Nous porterons notre contribution à l'éclaircissement de certains points en nous intéressant à reconnaître la voix de différents locuteurs. Dans le cadre d'une application de reconnaissance du locuteur, nous avons à disposition une base de données dans laquelle sont stockées les références des voix des locuteurs qui vont accéder à cette application. Ces locuteurs seront appelés les **clients** de l'application. La reconnaissance d'un client par sa voix se déroule en 2 phases: tout d'abord, nous devons identifier la voix de celui-ci parmi les voix stockées dans la base de données de l'application. Si cette opération est effectuée en analysant le signal de parole et en regardant à quel client la séquence appartient avec **la plus grande vraisemblance**, on parlera d'**identification du locuteur**. Lorsque l'identification du locuteur s'effectue par un autre moyen (code personnel, etc. . .) il reste à vérifier que le segment de parole testé appartient bien au locuteur, on parlera alors de **vérification du locuteur**. Selon qu'on tienne compte du texte prononcé par le locuteur ou que l'on ne s'intéresse qu'à des paramètres de celui-ci sans tenir compte de ce qu'il prononce, on parlera d'applications **dépendantes du texte** ou **indépendantes du texte**. Si le locuteur ne connaît pas à l'avance le texte qu'il doit prononcer et que l'application le lui impose, on parle alors de "**text prompted**" (voir pour plus de détails sur la classification en reconnaissance du locuteur [Chollet et Bimbot, 1995]). Quel que soit le type choisi, le processus de reconnaissance automatique du locuteur peut être décomposé en quatre parties principales ordonnées chronologiquement:

1. Le **prétraitement** du signal qui permet de compenser les déformations dues à la transmission du signal de parole, tels que le micro ou le canal téléphonique (section 1.2.1).
2. L'**analyse** du signal qui extrait les éléments caractéristiques du signal de parole (section 1.2.2).
3. La **modélisation** et mémorisation des paramètres caractéristiques du locuteur (section 1.2.3).
4. Le module de **décision** qui permet de tester si un échantillon de parole appartient bien au locuteur dont on vérifie l'identité (section 1.2.4).

1.2.1 Prétraitement du signal de parole

De manière à atténuer les déformations du signal dues à l'environnement (p.ex. échos, bruits de fond) et à tous les éléments intermédiaires nécessaires à le capter (p.ex. micros), à le transmettre (p.ex. lignes téléphoniques) ou à l'enregistrer (p.ex. convertisseurs analogique/numérique, déformations dues aux têtes d'enregistrement magnétique), un certain nombre de stratégies, de méthodes et d'algorithmes sont déployés. Pour la plupart, ce sont ceux utilisés dans le domaine du traitement du signal, avec cependant quelques particularités dues au signal de parole lui-même, citons-en quelques-unes ici:

- La plus grande partie de l'énergie du signal de parole se trouve entre 0 et 4000 [Hertz] .
- Le signal de parole est très redondant.

– On peut considérer que le signal varie de manière lente et qu'il est stationnaire sur une période d'environ 5 à 10 [ms] .

Ces considérations sur le signal de parole sont relativement grossières puisqu'on peut discerner en tous cas 5 formants, le 5ème formant étant pour les hommes au-dessus de 4000 [Hz] en général et que pour les femmes les 4ème et 5ème formants se situent au dessus de 4000 [Hz] . De plus, certaines plosives peuvent avoir une durée plus courte que 10 [ms] . Cependant ces caractéristiques sont celles qui ont permis de définir la largeur de bande téléphonique (300-3400 [Hz]) et qui sont utilisées encore de nos jours en codage GSM par exemple. Nous ne parlerons pas de ces aspects de traitement de signal dans cette thèse, mais ils restent cependant sous-jacents, puisque nous utilisons de la parole téléphonique, échantillonnée à 8 [kHz] (voir par exemple les travaux de [Mokbel, 1992] en milieu bruité pour plus de détails sur le prétraitement de la parole). Notons encore que certains traitements, visant à compenser les déformations de la bande téléphonique, s'appliquent sur les coefficients cepstraux directement (section 1.2.2).

1.2.2 Analyse de la parole

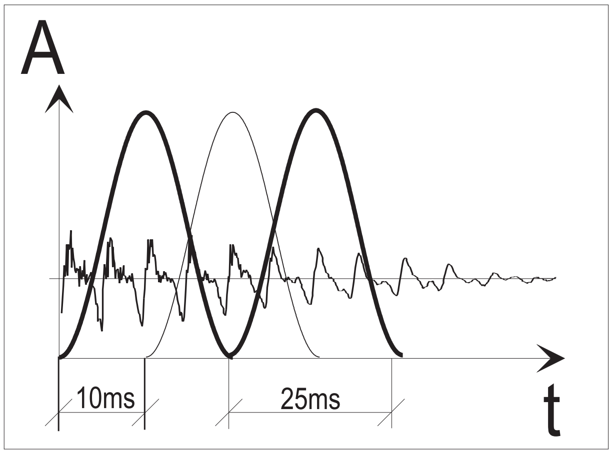

FIG. 1.10 – *Analyse du signal de parole par fenêtrage court terme. L'analyse se fait sur une durée de 25 [ms] , toutes les 10 [ms] .*

Comme nous l'avons vu dans les sections précédentes, il est possible d'identifier des motifs correspondant à des unités spectro-temporelles reconnaissables d'un individu à l'autre (les formants). Cependant, il est nécessaire, nous l'avons vu, que l'analyse du spectre se fasse sur des fenêtres temporelles de courte durée. La largeur des fenêtres d'analyse choisies sont de 25 [ms] réparties toutes les 10 [ms] (voir figure 1.10). De manière à compenser la distorsion crée par l'analyse fréquencielle sur des durées finies, on utilise une fenêtre de type Hamming, bien connue en traitement de signal. Une phase de pré-accentuation du signal est aussi utilisée pour compenser la pente du spectre (le premier formant contient plus d'énergie que les suivants). On utilise pour cela généralement un filtre RIF de premier ordre avec un coefficient ($0.9 < a < 1.0$). Voir par exemple [Rabiner et Juang, 1993] pour tous ces aspects de compensation du spectre de parole.

Analyse spectrale

Les systèmes d'analyse spectrale du signal de parole se basent principalement sur deux types de modèles:

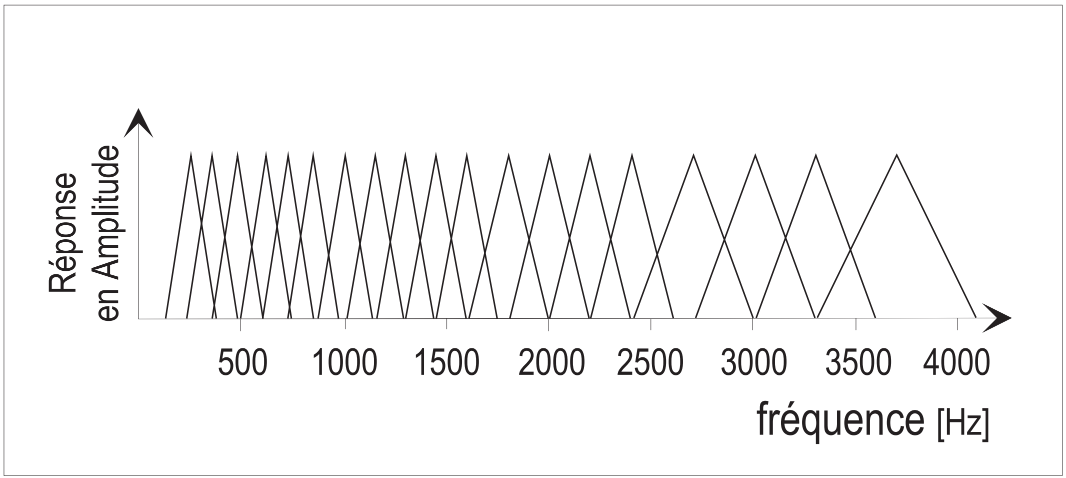

FIG. 1.11 – *Implémentation de bancs de filtres selon l'échelle MEL avec 19 canaux répartis entre 0 et 4000 [Hz]*

- Les modèles basés sur le système de perception humain (principalement la cochlée, voir par exemple [Pickles, 1988, Colombi *et al.*, 1993]). Cette approche est en fait une analyse en bancs de filtres, leur étagement en fréquences imitant la répartition et la forme des filtres de la cochlée. Cette répartition est non-linéaire et plusieurs implémentations sont possibles. Nous utiliserons dans cette thèse pour la reconnaissance de la parole l'**échelle MEL** (utilisée par le système de référence, voir l'annexe A). L'équation 1.1 donne la loi de transformation selon les fréquences (voir par exemple la figure 1.11 pour l'implémentation qu'en propose [Holmes,1980] avec 19 canaux répartis entre 0 et 4000 [Hz]). Il existe d'autres échelles non linéaires telle que par exemple l'échelle de Bark (voir [Zwiker et Terhardt, 1980]). Ce modèle d'analyse du signal de parole est appelé **non-paramétrique** puisqu'il ne suppose aucun modèle paramétrique de production du signal.

$$m = 1125 \cdot \log(0.0016 \cdot f + 1) \tag{1.1}$$

  Avec f la fréquence centrale du filtre avant transformation MEL.
- Les modèles basés sur l'appareil de production de la parole humaine (voir par exemple [Wakita, 1973, Wakita, 1979, Lin, 1995]). Ces modèles font l'hypothèse que l'appareil phonatoire de la figure 1.2 peut se modéliser par une série de tubes sans pertes (voir la fi-

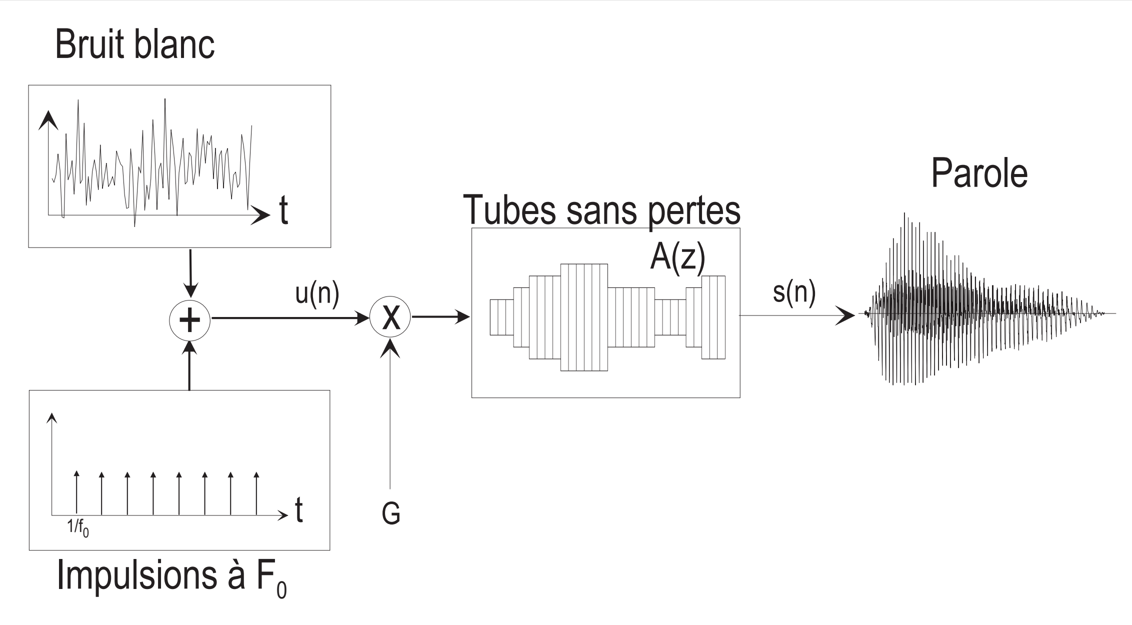

FIG. 1.12 – *Modèle de production de la parole faisant l'hypothèse que le conduit vocal est une série de tubes sans perte, auquel on fournit une source de bruits et/ou une source impulsionnelle.*

gure 1.12) et que la source du signal est soit un train d'impulsion de période $1/F_0$ (approximation de la fonction de vibration des cordes vocales de la figure 1.8) pour les parties voisées du signal, soit une source de bruit Gaussien pour les parties non-voisées [Mammone *et al.*, 1996, Rabiner et Juang, 1993], soit les deux simultanément [Stylianou, 1996]. Les tubes sans pertes (voir figure 1.12) sont équivalents à des filtres tous pôles appliqués aux sources du signal. De manière à estimer les coefficients de ces filtres, on suppose que le signal de parole $(s(n)\ ,\ n \in \{1, \cdots N\})$ se prédit, à chaque instant, comme une combinaison des échantillons aux instants précédents. La **LPC** (Linear Predicting Coding en anglais) pour un ordre p se définit de la manière suivante (équation 1.2) (voir par exemple [Oppenheim *et al.*, 1968]):

$$\tilde{s}(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p) = \sum_{i=1}^{p} a_i s(n-i) \qquad (1.2)$$

Si nous y incluons le terme d'excitation unitaire $u(n)$ et un gain G, la suite $s(n)$ devient:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \qquad (1.3)$$

Si nous exprimons l'équation 1.3 par sa transformée en z, nous obtenons:

$$S(z) = \sum_{i=1}^{p} a_i z^{-i} S(z) + GU(z) \tag{1.4}$$

Ce qui nous permet de calculer la fonction de transfert $H(z)$ du système:

$$H(z) = \frac{1}{G} \cdot \frac{S(z)}{U(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i \cdot z^{-i}} = \frac{1}{A(z)} \tag{1.5}$$

De manière à utiliser la prédiction linéaire pour reconstruire le signal $s(n)$, nous pouvons définir l'erreur de prédiction $e(n)$ qui est la différence entre le signal réel $s(n)$ et le signal approximé $\tilde{s}$:

$$e(n) = s(n) - \tilde{s}(n) = Gu(n) \tag{1.6}$$

Nous pouvons en déduire la fonction de transfert d'erreur:

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{i=1}^{p} a_i z^{-i} \tag{1.7}$$

En admettant que le modèle soit valable sur une fenêtre de temps donnée de longueur m, il reste à estimer les coefficients a_i. On utilise pour cela la méthode des moindres carrés avec $e(m) = s(m) - \tilde{s}(m)$, ce qui revient à minimiser l'erreur E_m:

$$E_m = \sum_m e^2(m) = \sum_m \left[s(m) - \sum_{i=1}^{p} a_i s(m-i) \right]^2 \tag{1.8}$$

Deux méthodes sont utilisées généralement pour calculer les coefficients a_i: la méthode d'autocorrélation ou la méthode de la covariance (pour plus de détails sur ces méthodes voir [Rabiner et Juang, 1993]).

La méthode LPC est une méthode dite **paramétrique** puisqu'on identifie (à court terme) des paramètres qui décrivent le signal de parole.
Notons encore que l'on définit les coefficients de réflexion k_i (PARCOR) à partir des coefficients a_i en utilisant un modèle auto-régressif (voir par exemple [Rabiner et Schafer, 1978, Fallside et Woods, 1985, Deller *et al.*, 1993, Makhoul, 1975]). On définit également le logarithme des rapports d'aires (log area ratio) lar comme étant

$$lar_i = \log\left(\frac{1-k_i}{1+k_i}\right)$$

Analyse cepstrale

Les coefficients produits à la sortie des bancs de filtres selon l'échelle MEL ou les coefficients LPC (les a_i) peuvent être utilisés pour mesurer des différences entre deux spectrogrammes (voir [Mammone *et al.*, 1996]). Ils présentent cependant de nombreux inconvénients comme par

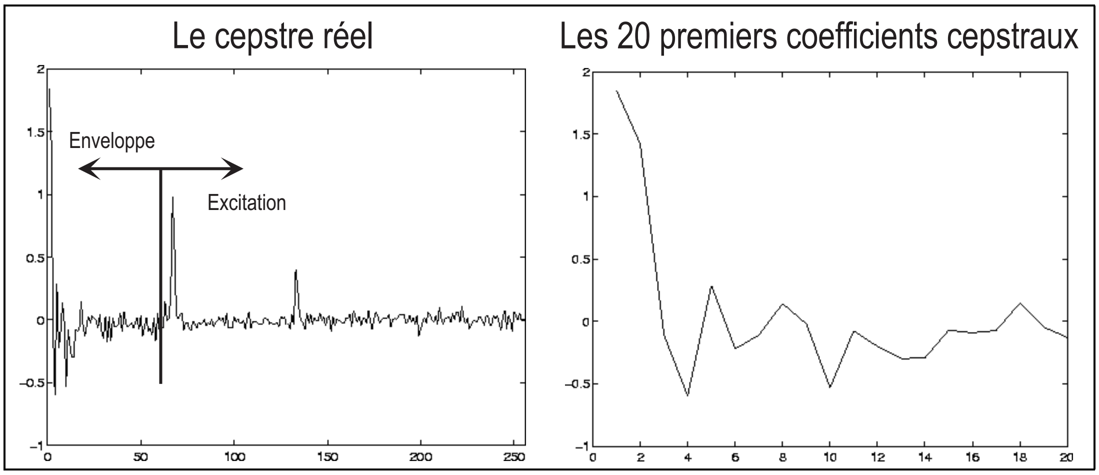

FIG. 1.13 – *Cepstre réel complet pour une fenêtre de 250 échantillons (à gauche), et les 20 premiers coefficients cepstraux (à droite).*

exemple de dépendre de l'énergie du signal et de l'excitation. De manière à pouvoir comparer différents spectres, plusieurs méthodes de normalisation et de mesure existent qui peuvent être exprimées dans un contexte plus général de la théorie de l'information (voir par exemple [Lee, 1991]). Nous présenterons ici la méthode la plus populaire pour extraire une information normalisée du spectre de parole: **la transformation cepstrale** (voir [Schroeder, 1985]).

Si nous admettons la représentation source/filtre du signal de parole, ce signal résulte d'une convolution dans le domaine temporel de la source et du filtre (équation 1.9).

$$s(t) = e(t) * h(t) \tag{1.9}$$

Cependant, ce qui nous intéresse pour identifier et pour mesurer des différences (ou des similitudes) entre spectres est l'enveloppe spectrale. Cette opération est effectuée de la manière suivante:

Nous transformons dans le domaine spectral l'équation 1.9 pour avoir le produit de l'excitation $E(f)$ et de la fonction de transfert du filtre $H(f)$:

$$S(f) = E(f) \cdot H(f) \tag{1.10}$$

Comme nous voulons découpler la source du filtre de manière à n'avoir plus que l'enveloppe spectrale, nous utilisons la fonction $\log$, de manière à ce qu'en ne prenant que le module du spectre nous obtenions:

$$\log |S(f)| = \log |E(f)| + \log |H(f)| \tag{1.11}$$

Une manière naturelle de découpler les composants de $\log |S(f)|$ qui varient lentement de ceux qui représentent les variations de l'excitation, consiste à appliquer une transformée de Fourrier inverse (calculé par une FFT^{-1}) à l'équation 1.11. Les coefficients temporels ainsi obtenus sont appelés

coefficients cepstraux. Les premiers coefficients donnent les paramètres de l'enveloppe spectrale (ou la réponse impulsionnelle du conduit vocal), les coefficients plus élevés, les variations de l'excitation (voir la figure 1.13).

Si les coefficients cepstraux sont issus d'une analyse en banc de filtres sur une échelle MEL, on les dénommera **MFCC** (Mel Frequency Cepstrum Coefficients), s'ils sont issus d'une analyse LPC on les appellera coefficients **LPCC** (Linear Predicting Coding Cepstrum). Il existe une méthode directe pour passer des coefficients a_p de la LPC à des coefficients cepstraux c_m LPCC qui utilise les récursions suivantes:

$$
\begin{aligned}
c_0 &= \ln G \\
c_m &= a_m + \sum_{j=1}^{m-1} \left(\frac{j}{m}\right) c_j a_{m-j}\,, \quad 1 \leq m \leq p \\
c_m &= \sum_{j=1}^{m-1} \left(\frac{j}{m}\right) c_j a_{m-j}\,, \quad m > p
\end{aligned}
$$

Rabiner [1993] propose que le nombre de coefficients cepstraux c_m soit environ 1.5 fois plus élevé que le nombre de coefficients a_p.

Un des avantages importants de la comparaison de spectres en utilisant les coefficients cepstraux est le fait que nous pouvons utiliser une mesure de distance euclidienne simple à estimer (voir [Basseville, 1989]).

Paramètres dynamiques Les caractéristiques dynamiques d'une fenêtre d'analyse s'étant montrées importantes pour la reconnaissance du locuteur (voir par exemple [Furui, 1986]), on adjoint aux coefficients cepstraux leurs dérivées et accélérations.

Autres corrections des coefficients cepstraux De manière à minimiser l'influence de la pente du spectre sur les coefficients cepstraux d'ordre inférieur et de la sensibilité au bruit des coefficients d'ordre supérieur, on leur applique une pondération, appelée liftering cepstral (voir [Rabiner et Juang, 1993]).

Analyse synchrone avec F_0

L'analyse du signal effectuée jusqu'à présent se base sur une répartition régulière des fenêtres temporelles à partir desquelles on extrait les caractéristiques du signal qui nous intéressent. De manière à extraire directement les harmoniques de la fréquence fondamentale F_0, nous utiliserons aussi une analyse avec des fenêtres court terme mais synchrones avec F_0. Ce type d'analyse requiert une première phase de recherche de F_0. De la précision de cette extraction dépendra directement la précision de l'estimation des harmoniques. Ces méthodes d'extraction de paramètres synchrones avec la F_0 sont surtout utilisées dans les applications de synthèse de parole à partir du texte [Stylianou, 1996, Dutoit, 1997] mais nous les utiliserons dans cette thèse pour transformer la voix de locuteurs (voir la section 1.3).

1.2.3 Modélisation de locuteurs

Que ce soit pour reconnaître le message prononcé par un locuteur ou son identité, il nous est nécessaire de modéliser les entités que nous voulons reconnaître automatiquement. Notre connaissance du cerveau humain ne nous aide pas beaucoup ici, car si l'analyse du signal effectuée par l'oreille humaine semble plus ou moins connue, il en va tout autrement de ce que fait le cerveau avec les informations reçues par la cochlée, de leur stockage et de leur interprétation.

Les systèmes de reconnaissance automatique de la parole et du locuteur actuels utilisent pour la plupart des algorithmes de comparaison de motifs ("patterns matching" en anglais). Les motifs utilisés sont basés sur des parties de spectrogrammes qui ont été évoqués dans la section 1.1 et dont on utilise les représentations cepstrales de la section 1.2.2. Dans le cadre de la **reconnaissance de la parole** indépendante du locuteur, on va chercher à modéliser les prononciations que *tous les locuteurs* peuvent avoir faites d'*un même motif*. Dans le cadre de la **reconnaissance de locuteurs**, nous modéliserons les différentes prononciations qu'*un locuteur* peut avoir effectuées pour le *même motif*. En étudiant la parole de locuteurs sur plusieurs prononciations des mêmes motifs, nous pouvons distinguer des variabilités caractéristiques du signal de parole nous permettant de séparer les locuteurs les uns des autres (variabilités inter-locuteurs) et d'autres, intrinsèques au locuteur (variabilités intra-locuteur).

Variabilités intra-locuteur La voix humaine, à la différence des empreintes digitales, varie avec le temps ou les conditions physiologiques et psychologiques du locuteur (voir par exemple [Homayounpour, 1995] ou [Scherer *et al.*, 1998]). Cependant, ces variations intra-locuteur ne sont pas identiques pour tous les humains. En effet, hormis les variations lentes de la voix dues au vieillissement, certains phénomènes extérieurs tels que la fumée ou l'état de santé d'une personne ont une influence variable sur sa voix.

Variabilités inter-locuteurs Elles proviennent des différences physiologiques (différences dimensionnelles du conduit vocal, fréquence d'oscillation des cordes vocales) et de différences de style de prononciation (accent, niveau social, etc. . .). Certaines de ces différences, qui influencent le spectre associé à chaque locuteur, vont nous permettre de les séparer.

Le locuteur et le monde Reconnaître un locuteur revient à essayer de le distinguer des autres. Cependant, quelle que soit la modélisation choisie, il est nécessaire de définir ce qui n'est pas le locuteur, ou, en d'autres termes, de trouver une mesure qui permette d'estimer les dimensions de l'hypercube dans lequel varient les paramètres du locuteur. Plusieurs manières de faire existent comme les modèles de voisinage (voir [Gravier, 1995] et le chapitre 4) ou les modèles de cohortes ou de monde (voir [Rosenberg *et al.*, 1991, Rosenberg *et al.*, 1992, Reynolds, 1992]).

Dans les applications pratiques, on distingue aussi les **clients**, qui sont des locuteurs dont on a enregistré des références et qui sont autorisés à pénétrer dans le système et les **imposteurs** qui sont des locuteurs qui tentent de se faire passer pour un client donné.

Algorithmes de modélisation du locuteur

De manière à mémoriser des caractéristiques qui dépendent du locuteur, nous utilisons des algorithmes capables de capturer les points communs entre différentes représentations de motifs spectraux issus du même locuteur (constituant ainsi un **modèle du locuteur**), tout en ayant la possibilité de s'adapter aux variations d'échelles fréquencielles et temporelles liées au signal de parole. Ces motifs peuvent être soit des segments de parole déterminés (mots, phonèmes) si nous travaillons en mode dépendant du texte, soit des segments de parole dont on ne connaît pas le contenu phonétique si l'application fonctionne en mode indépendant du texte. Ces algorithmes doivent être couplés avec une mesure qui permettra de donner une valeur de distorsion (ou de similitude) entre le modèle du locuteur et un motif inconnu dont on cherche à déterminer la provenance (voir par exemple [Gray et Markel, 1976, Raudys et Jain, 1991, Griffin *et al.*, 1994, Ng *et al.*, 1995]. Nous proposons ici un aperçu des algorithmes qui fournissent les meilleurs résultats en reconnaissance du locuteur (voir par exemple [CAVE,1998]), et nous présentons également leurs avantages et inconvénients.

La quantification vectorielle (QV) part du principe que l'on peut modéliser certaines parties du spectre avec des motifs standards associés aux unités acoustiques que l'on désire reconnaître (phonèmes, syllabes, etc. . .) (voir [Li et Wrench, 1983, Soong *et al.*, 1985], [Matsui et Furui, 1992] ou [Matsui et Furui, 1993]. Ces motifs standards sont de la taille d'un vecteur cepstral, leur nombre dépend des unités acoustiques que l'on définit et de leurs variations (par exemple on pourrait vouloir définir des phonèmes en contexte et donc avoir plusieurs motifs pour le même phonème selon le contexte qui le précède ou le suit).

Durant la **phase d'entraînement**, on crée un dictionnaire de N vecteurs-types $V_i, i = [1, \cdots, N]$ à partir de plusieurs vecteurs v^e_{ij} représentant le même motif. On estime alors une distance $\delta(V_i, v_{ij})$ entre les vecteurs v^e_{ij} et le vecteur du dictionnaire V_i.

Durant la **phase d'exploitation**, on mesure la distance entre chacun des vecteurs v^t_k d'une séquence de test et chacun des vecteurs V_i du dictionnaire. On attribue ainsi les vecteurs v^t_k au motif représenté par le V_i le plus proche. De manière à éviter les problèmes de motifs inconnus (bruits par exemple) on vérifie que la distance $\delta(V_i, v^t_{ik})$ n'est pas supérieure à la plus grande distance calculée sur les données d'entraînement.

Avantages de la QV Si le dictionnaire n'est pas trop grand, la méthode est très rapide.

Inconvénients de la QV Pour diminuer l'erreur de quantification, il est nécessaire d'augmenter rapidement la taille du dictionnaire.

Méthodes de programmation dynamique (Dynamic Time Warping, DTW)

L'algorithme de DTW est utilisé intensivement en recherche opérationnelle pour résoudre les problèmes d'alignement séquentiel. Il a été utilisé avec succès par [Sakoe et Chiba, 1978] en reconnaissance de la parole puis en reconnaissance du locuteur par [Furui, 1981b]. Il consiste principalement à effectuer une comparaison dynamique entre une matrice de référence et une matrice

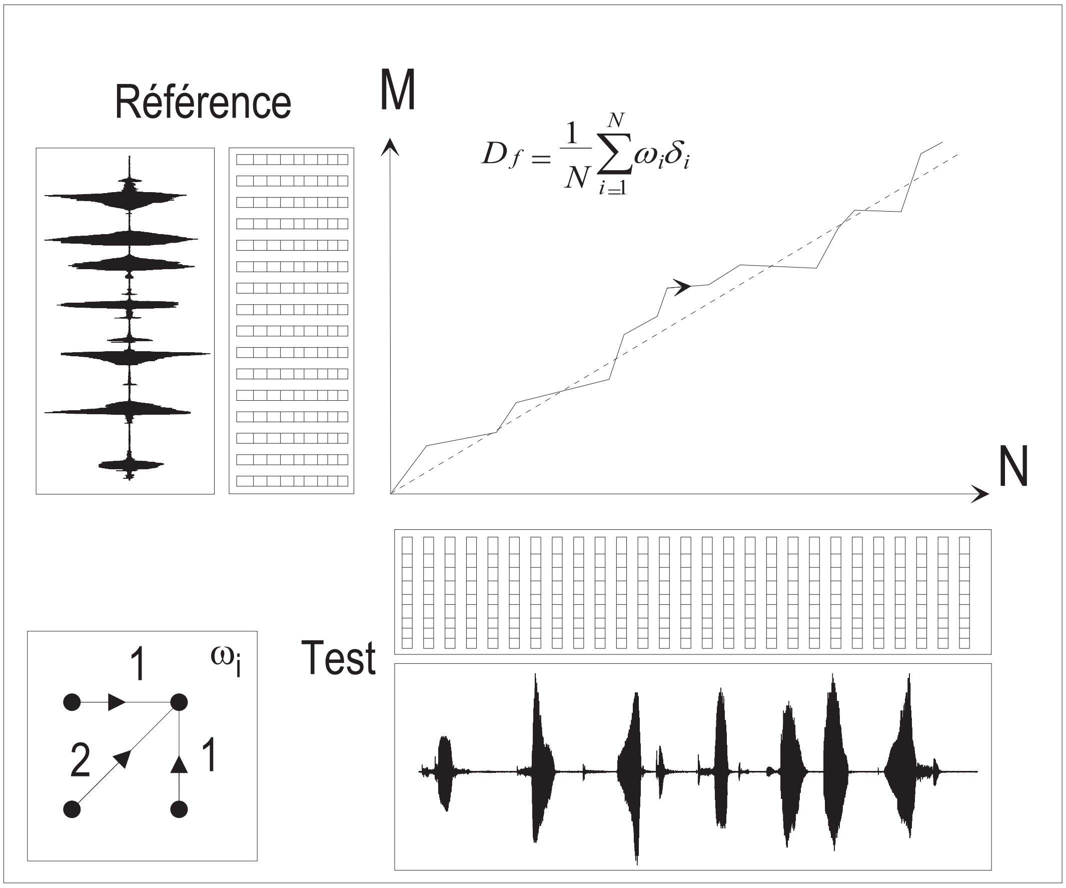

FIG. 1.14 – *Calcul de la distance dynamique par DTW. L'exemple ici montre le calcul effectué pour un des coefficients des vecteurs de paramètres extraits du signal de parole. L'encart en bas à gauche montre le principe des contraintes locales utilisées pour calculer le meilleur chemin.*

de test. Une mesure de distance est effectuée entre le test et la référence (voir la figure 1.14 qui montre le principe de la DTW).

Lors de la **phase d'entraînement**, les références du locuteurs sont simplement stockés.

Lors de la **phase d'exploitation**, la distance finale D_f entre une référence et la séquence de test est calculée comme la somme des distances partielles (δ_i) entre vecteurs le long du chemin optimal, pondérées par des contraintes locales (voir la figure 1.14 pour un exemple de contrainte locale). Le chemin optimal est le chemin dont la distance globale est minimale. Afin d'augmenter l'efficacité de la méthode, les chemins possibles sont limités et toute une gamme de contraintes locales peut être imposée. Lorsque l'on possède plusieurs exemplaires de référence R_j, le score de sortie final est la moyenne de toutes les distances calculées sur toutes les références.

Avantages de la DTW L'algorithme de DTW est rapide, bien adapté à la parole parce que capable de tenir compte des variations temporelles du signal. Il ne nécessite pas beaucoup de données pour fonctionner correctement.

Inconvénients de la DTW La DTW est très sensible à la segmentation du signal. En effet, si le point de départ du calcul dynamique n'est pas bon, l'algorithme peut rapidement diverger du chemin optimal. Il existe cependant des possibilités de corriger partiellement cette erreur (voir pour cela [Myers et Rabiner, 1981, Furui, 1981b, Rabiner et Juang, 1993]). De plus, ses capacités de modélisation des variabilités intrinsèques du locuteur sont relativement limitées puisqu'il n'est capable d'estimer que des points sur le meilleur chemin et non des distributions.

Méthodes statistiques (SSO+S, GMM, HMM)

L'idée de la quantification vectorielle de trouver une sorte de *"vecteur moyen"* représentant un motif caractéristique du spectre peut être étendue en supposant que l'on regarde aussi des statistiques d'ordre plus élevé pour capter des variations des motifs spectro-temporels. Plusieurs variations du même principe sont utilisées (voir par exemple [Papoulis, 1984, Pierrot, 1998] pour une description unifiée des modèles statistiques) comme les modèles statistiques du second ordre avec une mesure de sphéricité (SSO+S), les mélanges de Gaussiennes et les modèles de Markov cachés.

Statistique du second ordre avec mesure de sphéricité (SSO+S)

Cette méthode, auparavant utilisée en mécanique (voir par exemple [Drouiche, 1993]), a été proposée pour la première fois en reconnaissance du locuteur indépendante du texte par Bimbot et Mathan [1994] (voir également [Gish *et al.*, 1986, Hoffbeck et Landgrebe, 1996]). Elle suppose que la matrice de covariance des coefficients cepstraux pris sur une durée de parole suffisamment longue (de l'ordre de 30 secondes) est suffisamment caractéristique pour distinguer les locuteurs. On n'utilise donc ici plus qu'un seul motif spectro-temporel pour tout un échantillon de parole. Pour extraire ces caractéristiques on utilise une statistique du second ordre munie d'un test de sphéricité.

Lors de la **phase d'entraînement** nous calculons une matrice de covariance Σ_{X} (équation 1.12) de la séquence de référence X du locuteur. Si plusieurs séquences sont à disposition, la matrice est calculée sur toutes les séquences.

$$\Sigma_{\mathrm{X}} = \frac{1}{M} \sum_{t=1}^{M} X_t X_t^T \tag{1.12}$$

Lors de la **phase d'exploitation** nous estimons la matrice de covariance de la séquence à tester Y. Puis nous calculons la mesure de sphéricité symétrique $\mu_{AH}(\Sigma_{\mathrm{X}}, \Sigma_{\mathrm{Y}})$ qui est définie comme:

$$\mu_{AH}(\Sigma_{\mathrm{X}}, \Sigma_{\mathrm{Y}}) = \log\left[\frac{\mathbf{A}}{\mathbf{H}}\right] \tag{1.13}$$

1. avec $\mathbf{A}$ la somme des valeurs propres du produit $\Sigma_{\mathrm{X}} \cdot \Sigma_{\mathrm{Y}}^{-1}$:

$$\mathbf{A}(\lambda_1, \lambda_2, \ldots, \lambda_m) = \frac{1}{m} \sum_{i=1}^{m} \lambda_i = \frac{1}{m} \cdot tr(\Sigma_{\mathrm{Y}} \cdot \Sigma_{\mathrm{X}}^{-1}) \tag{1.14}$$

2. et $\mathbf{H}$:
$$\mathbf{H}(\lambda_1, \lambda_2, \ldots, \lambda_m) = m \left(\sum_{i=1}^{i=m} \frac{1}{\lambda_i} \right)^{-1} = m \cdot \left[tr \left(\Sigma_{\mathrm{X}} \cdot \Sigma_{\mathrm{Y}}^{-1} \right) \right]^{-1} \tag{1.15}$$
3. En utilisant la relation:
$$\mu_{AH}(\Sigma_{\mathrm{X}}, \Sigma_{\mathrm{Y}}) = 0 <==> A = H <==> X \propto Y \tag{1.16}$$
4. on peut déterminer une distance:
$$d_{XY} = \mu_{AH}(\Sigma_{\mathrm{X}}, \Sigma_{\mathrm{Y}}) \tag{1.17}$$

Notons que dans notre cas, les matrices de covariances $\Sigma_{\mathrm{X,Y}}$ se confondent avec les matrices de corrélation, car nous avons centré et réduit les distributions des coefficients (voir [Saporta, 1990], p84).

Avantages des SSO+S Les matrices de covariance munies d'une mesure symétrique sont intéressantes car leur calcul est très simple donc très rapide. En effet, aucune extraction explicite des valeurs propres n'est nécessaire, seules les traces des produits de $(\Sigma_{\mathrm{X}} \cdot \Sigma_{\mathrm{Y}}^{-1})$ et de $(\Sigma_{\mathrm{Y}} \cdot \Sigma_{\mathrm{X}}^{-1})$ sont calculées.

Inconvénients des SSO+S La mesure SSO+S ne permet que de distinguer des caractéristiques du locuteur qui sont stables tout au long du segment de parole. Ainsi toutes les variations spectro-temporelles locales sont moyennées.

Mélanges de Gaussiennes (GMM)

Si nous étendons l'idée de la matrice de covariances des coefficients cepstraux sur un morceau de parole du locuteur, nous pouvons définir un mélange de Gaussiennes comme étant la somme de N composantes Gaussiennes pondérées (voir principalement les travaux de Reynolds [1992, 1994, 1997] sur les GMM: Gaussian Mixture Models en anglais). Nous pouvons calculer la probabilité conditionnelle qu'un vecteur Y (cepstral à D composantes dans notre cas) ait pu être émis par le modèle du client M_C:
$$p(Y|M_C) = \sum_{i=1}^{N} p_i b_i(Y) \tag{1.18}$$
Avec $(p_i, i \in \{1, \cdots, N\})$ les probabilités de chaque composante et $(b_i, i \in \{1, \cdots, N\})$ les densités du mélange. Chaque composante a une densité de probabilité gaussienne multivariée de dimension D de moyenne μ_i et de matrice de covariances Σ_i ayant la forme suivante:
$$b_i(Y) = \frac{1}{(2\pi)^{D/2}} \cdot |\Sigma_i|^{1/2} \cdot e^{-\frac{1}{2}(Y-\mu_i)^T \cdot \Sigma_i^{-1}(Y-\mu_i)} \tag{1.19}$$
Les GMM étant une version dégénérée d'un modèle de Markov caché (modèle à un état), les phases d'entraînement et d'exploitation sont identiques aux HMM et seront détaillées dans la section suivante.

Avantages des GMM Avec un mélange contenant beaucoup de Gaussiennes, la modélisation GMM donne d'excellents résultats en reconnaissance du locuteur indépendante du texte (cet algorithme est à l'état de l'art dans ce domaine, voir par exemple le chapitre 6). Il est possible de fusionner des GMM de manière à tenir compte de l'environnement (voir par exemple [Reynolds, 1997]).

Inconvénients des GMM Bien qu'ils soient capables de capturer les informations à plus long terme d'un locuteur, ils ne contiennent pas d'aspects dynamiques. Pour une bonne modélisation (i.e. beaucoup de Gaussiennes) nécessitent beaucoup de données.

Les modèles de Markov cachés (HMM)

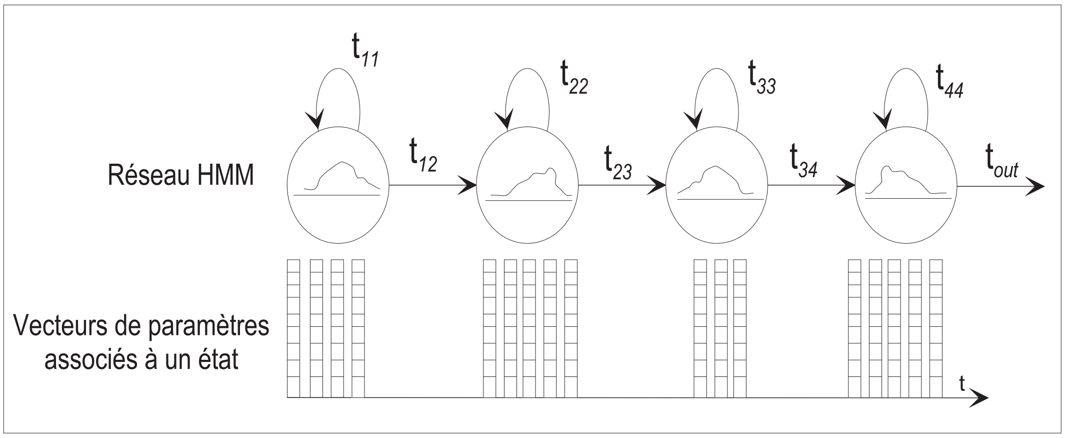

FIG. 1.15 – *Modèle HMM pour un mot.*

Les propriétés statistiques des modèles de Markov cachés (en anglais: Hidden Markov Models ou *HMM*) [Rosenberg *et al.*, 1991] en font une des modélisations les plus efficaces actuellement en reconnaissance du locuteur dépendante du texte. Les HMM permettent de modéliser des processus stochastiques variant dans le temps [Scharf, 1991, Rabiner et Juang, 1993]. Pour cela, ils combinent les propriétés à la fois des distributions de probabilités et d'une machine à états. Ils ont déjà été utilisés avec succès dans la modélisation de files d'attentes [Kleinrock, 1975].

Les modèles de Markov cachés combinent les propriétés des mélanges de Gaussiennes et ceux de la programmation dynamique. Ils sont constitués d'une machine à états, avec un mélange de Gaussiennes pour chaque état. Dans le cadre de ce modèle, nous considérons que le signal de parole est un processus stochastique par morceau (chacun des états). On peut donc introduire des contraintes temporelles empêchant la machine à états de revenir à un état précédent. Le modèle HMM ainsi créé est dit gauche-droite (voir la figure 1.15). Chaque état modélise un morceau de signal représenté dans l'espace des paramètres cepstraux par un certain nombre de vecteurs alignés temporellement. Il est à noter que pour un même état ($\equiv$ à une même distribution), l'alignement temporel des vecteurs de paramètres est perdu. Le passage d'un état à un autre s'effectue en tenant compte d'une probabilité de transition t_i d'un état à l'autre.

Lors de la **phase d'entraînement**, les paramètres (Gaussiennes et transitions) sont estimés à partir des données d'entraînement. Les modèles ont tous la même structure HMM gauche-droite constituée d'un état par phonème et un état par transition entre phonèmes [Gravier, 1995]. Deux sortes de modèles HMM sont créés pour chaque chiffre du code personnel du client:

– *Un modèle du monde*, créé à partir d'une base de données (p. ex. Polyphone, voir annexe B.4) comportant un grand nombre de locuteurs dont on a extrait plusieurs répétitions ($\simeq$ 300) de chaque mot du vocabulaire que l'on va utiliser pour l'application considérée. Ce modèle est indépendant du locuteur donc identique pour tous les clients (voir par exemple [Matsui et Furui, 1995a]). Les paramètres de ce modèle sont estimés par un entraînement standard en **reconnaissance de la parole**:

   1. Les vecteurs de paramètres d'un mot donné sont répartis dans tous les états de façon régulière.
   2. On applique l'algorithme de Viterbi [Rabiner et Juang, 1993] pour trouver la meilleure séquence d'état (celle qui maximise la vraisemblance des vecteurs).
   3. On réestime les Gaussiennes (moyennes et variances) de chaque état sur le chemin calculé en 2.
   4. On réestime les probabilités de transition par dénombrement des trames associées à un état.
   5. On revient à 2 jusqu'à ce que l'on converge, c'est-à-dire que le chemin optimal ne varie plus.
   6. On utilise ensuite un autre algorithme permettant de ré-estimer les probabilités de transition et de modifier les Gaussiennes de façon plus fine (algorithme de Baum-Welsh) [Baum *et al.*, 1970, Rabiner et Juang, 1993, Young et Bloothooft, 1997], par ré-estimation du chemin vers l'avant et vers l'arrière (forward-backward). On cesse de ré-estimer les paramètres lorsque la variation de la vraisemblance maximale totale de la séquence passe au-dessous d'un certain seuil.

– *Un modèle du client*, qui peut utiliser comme paramètres initiaux, ceux du modèle du monde, des Gaussiennes à moyennes nulles et à variances unitaires ("flat start"en anglais) ou une variance pré-calculée sur des données d'une base externe (voir [CAVE,1998]). La ré-estimation des paramètres pour chaque locuteur se fait en utilisant l'algorithme de Baum-Welsh décrit en 6 pour chaque locuteur sur ses données d'entraînement. Afin de garder une structure temporelle aux données (contrainte des applications), les données d'entraînement seront issues de la première session d'enregistrement effectuée par chaque client.

Lors de la **phase d'exploitation**, on estime le logarithme du rapport de vraisemblances entre le modèle du locuteur et le modèle du monde. Ce rapport (souvent appelé score) est comparé à un seuil. La section 1.2.4 explique en détail comment le rapport de vraisemblance peut se définir comme un test d'hypothèse. Nous verrons également au chapitre 5 comment l'établissement du seuil peut changer les performances d'un système HMM. L'annexe A propose un système HMM comme système de référence.

Méthodes neuronales et hybrides

Ces dernières années, de nouvelles méthodes alliant les qualités des algorithmes statistiques comme les HMM et les réseaux de neurones artificiels (RNA) sont apparus en reconnaissance de la parole et du locuteur sous le nom d'**architectures hybrides** (voir par exemple pour la reconnaissance de la parole [Morgan et Bourlard, 1995] ou [Naik et Lubensky, 1994] pour la reconnaissance du locuteur avec des architectures hybrides ou [Homayounpour, 1995], [Bennani et Gallinari, 1994] pour des réseaux de neurones artificiels seuls). Nous donnons un exemple d'une telle approche dans l'annexe C pour une application de vérification utilisée en mode indépendant du texte.

1.2.4 Mesures et décisions en reconnaissance du locuteur

Test d'hypothèses (calcul des scores)

Nous allons aborder ici la problématique de la prise de décision (voir par exemple [Green et Swets, 1988]). En effet, dans tous les systèmes de reconnaissance du locuteur il faut, à un moment ou à un autre, prendre la décision d'accepter ou de rejeter un segment de parole (noté ici O_t la suite des vecteurs issus de l'étage de paramétrisation) comme appartenant au client dont on cherche à vérifier l'identité. Ce genre de décisions peut se comprendre comme un test d'hypothèse statistique H_0 (le segment considéré appartient au locuteur) contre H_1 (ce segment n'appartient pas au locuteur). Ce qui revient à tester la probabilité conditionnelle d'un événement X sachant les hypothèses H_0 et H_1 (équations 1.20). Les quantités $P(H_0|X)$ et $P(H_1|X)$ sont appelées probabilités *a posteriori* de l'hypothèse H_0, respectivement H_1 sachant l'événement X.

$$H_0 : \ Locuteur, \ H_1 = \overline{H_0}$$

$$P(H_0) \underset{reject}{\overset{accept}{\gtrless}} P(H_1), \quad P(H_0|X) \underset{reject}{\overset{accept}{\gtrless}} P(H_1|X) \tag{1.20}$$

Comme pratiquement nous ne savons pas modéliser ce qui n'est pas le locuteur (virtuellement tous les autres locuteurs passés, présents et futurs de cette planète vivant ou ayant vécu en même temps que le locuteur considéré), nous traduisons l'hypothèse H_1 en : *"Ce segment appartient à un grand nombre de locuteurs qui ne sont ni des clients ni des imposteurs de l'application considérée"*. H_1 sera modélisée par un *modèle de monde* qui respectera cette approximation de l'hypothèse de départ. Appliquée au problème de la vérification, l'équation 1.20 devient donc le test de la probabilité *a posteriori* du modèle du client M_C sachant la séquence d'observation O_t, contre la probabilité *a posteriori* du modèle de monde M_W sachant la même séquence d'observation O_t (équation 1.21).

$$P(M_C|O_t) \underset{reject}{\overset{accept}{\gtrless}} P(M_W|O_t) \tag{1.21}$$

En utilisant la première règle de Bayes (équation 1.22) (voir p.ex. [Saporta, 1990]) et connaissant la probabilité *a priori* qu'une séquence de parole appartienne au locuteur $P(M_C)$ ou au modèle de monde $P(M_W)$, on peut en déduire les probabilités *a posteriori* $P(O_t|M_C)$ et $P(O_t|M_W)$ que la séquence O_t soit issue du modèle M_C ou du modèle M_W respectivement (équation 1.23).

$$P(B/A) = \frac{P(A/B)P(B)}{P(A)} \qquad \text{(Bayes)} \tag{1.22}$$

$$P(M_C|O_t) = \frac{P(O_t|M_C) \cdot P(M_C)}{P(O_t)} \underset{reject}{\overset{accept}{\gtrless}} \frac{P(O_t|M_W) \cdot P(M_W)}{P(O_t)} = P(M_W|O_t) \tag{1.23}$$

Que l'on peut transformer en:

$$\frac{P(O_t|M_C)}{P(O_t|M_W)} \underset{reject}{\overset{accept}{\gtrless}} \frac{P(M_W)}{P(M_C)} \tag{1.24}$$

L'estimation des deux probabilités *a posteriori* $P(O_t/M_C)$ et $P(O_t/M_W)$ se fait par estimation du maximum de vraisemblance (fonction $\mathscr{L}(\bullet)$) [Saporta, 1990]. On définit la quantité LR comme le rapport de vraisemblance de la séquence O_t sachant les deux modèles M_C et M_W. L'équation 1.24 devient:

$$LR = \frac{\mathscr{L}(O_t, M_C)}{\mathscr{L}(O_t, M_W)} \underset{reject}{\overset{accept}{\gtrless}} \frac{P(M_W)}{P(M_C)} \tag{1.25}$$

Pour optimiser le temps de calcul (transformation des multiplications en additions) et dû à ses propriétés de normalité, nous utiliserons le logarithme du rapport de vraisemblances. Le test d'hypothèse final (LLR: Log Likelihood ratio en anglais) devient donc:

$$LLR(M_C, M_W, O_t) = \log \mathscr{L}(O_t, M_C) - \log \mathscr{L}(O_t, M_W) \underset{reject}{\overset{accept}{\gtrless}} \log\left(\frac{P(M_W)}{P(M_C)}\right) \tag{1.26}$$

Avec $\log \mathscr{L}(O_t, M_C)$ et $\log \mathscr{L}(O_t, M_W)$ les log-vraisemblances du modèle du locuteur, respectivement du monde, calculées sur la séquence O_t.

Fonction d'erreur Supposons maintenant que pour un système donné nous cherchions à minimiser le coût total de ses erreurs. On peut pour cela définir une **fonction de coût** qui est la somme des erreurs faites en acceptant faussement une séquence qui n'est pas du locuteur (**fausse acceptation: FA**) ou en rejetant faussement une séquence qui appartient au locuteur (**faux rejet: FR**):

$$c_{tot} = c_{fr} \cdot P(C) \cdot E(FR|C) + c_{fa} \cdot P(\overline{C}) \cdot E(FA|\overline{C}) \tag{1.27}$$

Avec c_{fr} et c_{fa} les coûts d'un faux rejet, respectivement d'une fausse acceptation. Ces coûts seront déterminés en fonction de l'application et indiquent l'importance d'un type d'erreur par rapport à l'autre. $P(C)$ et $P(\overline{C})$ représentent respectivement les probabilités *a priori* que la séquence de test appartienne au client ou non. Finalement, $E(FR|C)$ et $E(FA|\overline{C})$ représentent les taux d'erreurs de faux rejet et de fausse acceptation effectuées par le système. On peut montrer [Scharf, 1991] que la minimisation de la fonction de coût c_{tot} revient à rajouter à l'équation 1.23 les coûts d'erreur c_{fr} et c_{fa}. De plus, si l'on admet que $P(O_t|M_{\overline{C}})$ puisse être approximée par $P(O_t|M_W)$ alors on peut réécrire l'équation 1.23 comme:

$$\frac{P(O_t|M_C) \cdot P(M_C)}{P(O_t)} \cdot c_{fr} \underset{reject}{\overset{accept}{\gtrless}} \frac{P(O_t|M_W) \cdot P(M_W)}{P(O_t)} \cdot c_{fa} \tag{1.28}$$

Ce qui revient, en utilisant les équations 1.24, 1.25 et 1.26, à estimer les inégalités suivantes:

$$LLR(M_C, M_W, O_t) = \log \mathscr{L}(O_t, M_C) - \log \mathscr{L}(O_t, M_W) \underset{reject}{\overset{accept}{\gtrless}} \log\left(\frac{P(M_W)}{P(M_C)} \cdot \frac{c_{fa}}{c_{fr}}\right) \tag{1.29}$$

Donc à comparer le logarithme du rapport de vraisemblances à un seuil de décision Θ.

$$\log\left(\frac{P(M_W)}{P(M_C)} \cdot \frac{c_{fa}}{c_{fr}}\right) = \log(R) = \Theta \tag{1.30}$$

La quantité R, qui sert de seuil de décision dans notre cas, est souvent appelé **rapport de risque**. Il est intéressant de constater que ce rapport est **indépendant du locuteur** et ne dépend que de valeurs *a priori* déterminées par les conditions d'une application, si l'on admet que les probabilités *a priori* de tous les clients sont identiques.

Les distributions statistiques de ces LLR calculées sur des séquences de locuteurs et d'imposteurs seront les entités que nous manipulerons pour calculer soit un point de fonctionnement, soit les performances intrinsèques du système.

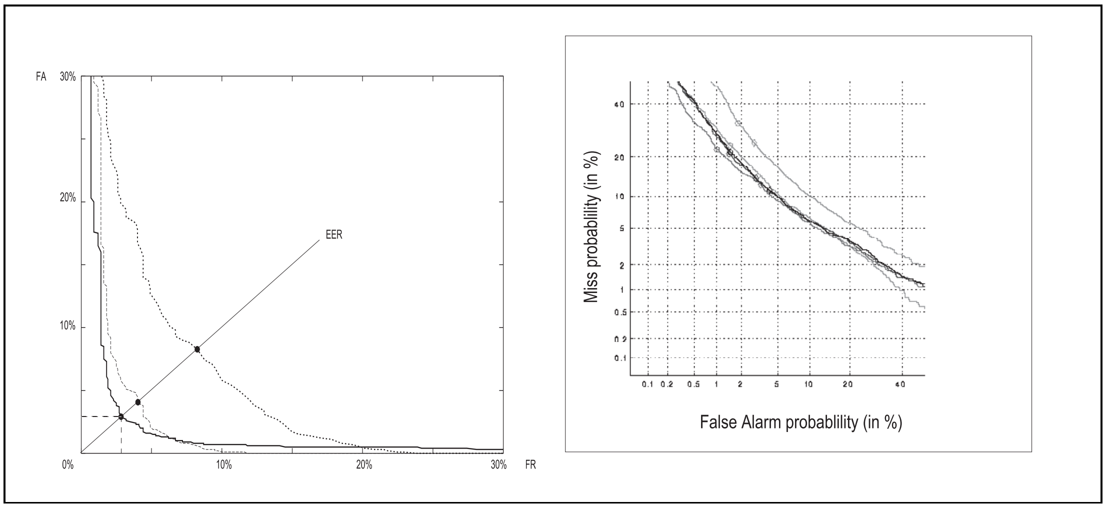

FIG. 1.16 – *Exemple de courbes de caractéristiques opérationnelles du récepteur et sa version modifiée pour les tests NIST.*

Taux d'erreur

De manière à représenter les performances d'un système de reconnaissance de locuteur, plusieurs mesures sont possibles (voir [Naik et Doddington, 1987, Oglesby, 1994]). En effet, selon que l'on veuille connaître ses performances intrinsèques ou en exploitation, nous n'utiliserons pas les mêmes mesures. On calcule les performances intrinsèques d'un système à partir des scores obtenus en utilisant des données de test de clients et d'imposteurs. On estime ensuite *a posteriori* les taux d'erreur lorsque l'on fait varier le seuil de décision sur toute la plage de réglage du système. Le résultat d'une telle mesure est une courbe **COR** (Caractéristique Opérationnelle du Récepteur) (voir par exemple [Oglesby, 1994, Chollet et Bimbot, 1995]) qui se situe dans le plan des Faux Rejets/Fausses Acceptations. Un des points de la courbe COR est très populaire car il correspond à un taux d'égale erreur de fausses acceptations et de faux rejets (EER: Equal Error Rate). La figure 1.16 nous donne un exemple de courbes COR avec la droite *EER* qui intercepte les courbes à l'EER, et un exemple de courbe COR-NIST avec une échelle "normal deviate" (voir [Martin *et al.*, 1997]). Notons que si l'on veut montrer avec une courbe COR les performances de tous les locuteurs pour une application donnée, il faut au préalable normaliser leurs scores si les seuils de décision sont dépendants du client. Pour chacun d'eux, la normalisation s'effectue simplement, pour chaque client, en soustrayant la valeur du seuil individuel aux scores obtenus par le modèle du client en question.

Pour connaître les performances d'un système en exploitation, on calcule d'abord un **point de fonctionnement** en utilisant des données de réglage. Ce point est estimé à partir du facteur de risque R (voir l'équation 1.30) qui définit un seuil de décision Θ. On évalue ensuite les performances du système avec le seuil Θ déterminé *a priori* ce qui nous permet de calculer un taux de faux rejets **FR** et un taux de fausses acceptations **FA** en exploitation. On calcule également souvent la moyenne de ces deux erreurs, le taux de $1/2$ erreur (**HTER**, Half Total Error Rate en anglais):

$$\text{HTER} = \frac{FA + FR}{2} \tag{1.31}$$

De manière à évaluer la significativité des réultats, ceux-ci seront présentés soit avec un taux en pourcent et le nombre de tests effectués, soit avec un taux d'erreur E en pourcent et une variance sur l'erreur var calculée de la manière suivante:

$$var = \pm 1.96 \cdot \sqrt{\frac{E \cdot (1 - E)}{N}}$$

Avec une limite de confiance de 0.95 (voir [Saporta, 1990]), cette formule n'est valable que pour un nombre d'échantillons N grand.

1.3 Analyse/synthèse de la parole

1.3.1 Introduction

La synthèse de la parole consiste à produire un signal ressemblant à de la parole humaine à partir de machines automatiques. La synthèse de la parole offre de grands débouchés comme élément du dialogue homme-machine. Les applications actuelles de la synthèse sont surtout dans les applications de génération de parole à partir du texte [Dutoit, 1997]. Si nous évoquons la synthèse de la parole dans une thèse de reconnaissance du locuteur, c'est parce que nous allons utiliser les méthodes mises au point dans ce domaine pour analyser la parole d'un locuteur afin de déterminer les paramètres qui constituent son identité. Le processus se déroule en 2 étapes: l'analyse du signal de parole d'un locuteur, de manière à en extraire des paramètres caractéristiques, puis la synthèse de la parole utilisant les paramètres extraits lors de l'analyse. Nous décrivons ci-après les méthodes de synthèse que nous avons utilisées. Si le synthétiseur harmonique plus bruit est à l'état de l'art, les synthétiseurs à formants ne le sont plus. Ils offrent cependant une approche intéressante du signal de parole.

1.3.2 Analyse/synthèse de formants

Le synthétiseur de Klatt [1980, 1990] est un synthétiseur à formants (voir section 1.1.1). Il est constitué de bancs de filtres séries et parallèles auxquels on donne comme paramètres les positions des formants, leur amplitudes et leurs largeurs de bande, on postule aussi 2 sources soit des impulsions à F_0 soit un bruit gaussien pour les parties non voisées. Dans notre approche analyse/synthèse, nous avons utilisé un extracteur de formants utilisant une programmation dynamique pour suivre les formants du signal de parole [Ent, 1996].

1.3.3 Analyse/synthèse harmoniques plus bruit (H+N)

Le système d'analyse/synthèse harmonique plus bruit (H+N) a été développé par Ioannis Stylianou en [1996] et a été utilisé avec succès dans des applications de génération de parole à partir du texte. Il part de l'hypothèse que la parole peut se modéliser en une partie déterministe et une partie stochastique. La partie déterministe est basée sur une somme de fonctions sinusoïdales multiples de la fréquence fondamentale F_0 (voir la section 1.1.1). Elle peut varier en phase et amplitude à chaque instant. La partie stochastique est modélisée par un système source/filtre à excitation Gaussienne. Bien que l'analyse/synthèse soit faite de façon synchrone avec F_0, les paramètres sont sauvegardés régulièrement toutes les 10 [ms]. Une détection de la fréquence de voisement maximum permet de modéliser le bruit aussi durant les parties voisées du signal.

Détection de la fondamentale, fréquence de voisement maximum

La fréquence fondamentale F_0, nous l'avons observé à la section 1.1.1, varie dans le temps. Une analyse de parole de qualité dépend fortement de l'extraction de la fréquence fondamentale puisque l'analyse se fait de façon synchrone avec F_0. La F_0 est calculée de manière très précise en essayant de minimiser le critère d'erreur ϵ qui indique la différence entre la F_0 originale et la F_0 resynthétisée (équation 1.32).

Le processus d'extraction de la F_0 et la détermination de la fréquence maximale de voisement se déroulent de la manière suivante [Stylianou, 1996]:

- On définit l'intervalle de fréquence de variation de la F_0 (entre 80 et 400 [Hz]).
- On utilise une fenêtre d'analyse de type Blackman au minimum 3 fois plus grande que la période maximale de la fréquence fondamentale.
- On minimise le critère d'erreur ϵ (équation 1.32) sur la gamme de fréquences décidée, ce qui nous donne, pour un signal à 8 [kHz], un calcul sur une période de 20 à 100 échantillons.

$$\epsilon = \frac{\int_{-1/2}^{1/2} \left[|S_\omega(f)| - |\widehat{S_\omega}(f)| \right] df}{\int_{-1/2}^{1/2} |S_\omega(f)|^2 df} \tag{1.32}$$

  Avec $S_\omega(f)$ le spectre issu de la transformée de Fourier sur une fenêtre prélevée sur le signal $s(t)$ et $\widehat{S_\omega}(f)$ le spectre de synthèse reconstruit à partir de F_0 (pour plus de détails sur cette minimisation, voir [Griffin et Lim, 1988]).
- Un fois détectée la fréquence fondamentale estimée $\widehat{F_0}$, on marque les fenêtres voisées en estimant l'erreur entre le spectre original $S_w(f)$ et sa reconstruction synthétique $\widehat{S_w}(f)$ effectuée à partir de F_0 et ses 4 premières harmoniques. Si l'erreur est en dessous d'un seuil donné (typiquement 15 [dB]) on déclare la fenêtre comme voisée.
- On calcule la fréquence de voisement maximum sur les fenêtres voisées en cherchant le pic maximum de fréquence qui correspond à une harmonique de $\widehat{F_0}$ en regardant que les pics des harmoniques soient supérieurs d'un facteur 2 aux pics voisins.
- On affine l'estimation de $\widehat{F_0}$ en utilisant les différentes harmoniques f_i déterminées à l'étape précédente pour minimiser une erreur $E(\widehat{F_0})$:

$$E(\widehat{F_0}) = \sum_{i=1}^{Ln} |f_i - i \cdot \widehat{F_0}|^2 \tag{1.33}$$

  Avec L_n le nombre maximum de fréquences voisées déterminées à l'étape précédente.
- On recommence le processus pour chaque fenêtre d'analyse. Celles-ci sont normalement séparées de 10 [ms].

Puis, sur les parties voisées du signal, on redéfinit de nouveaux instants d'analyse t_a^i synchrones avec F_0 de manière à ce que:

$$t_a^{i+1} = t_a^i + P(t_a^i) \tag{1.34}$$

Avec $P(t_a^i)$ la période de F_0 à l'instant t_a^i.

Analyse harmonique

Nous supposons que la période de F_0 et l'amplitude de ses harmoniques est constante autour d'un instant d'analyse t_a^i. Donc, pour de petits $|t - t_a^i|$, la phase instantanée de la $k^{\text{ème}}$ harmonique $\phi(t)$ peut être approximée dans le voisinage de t_a^i par:

$$\phi_k(t) = \phi_k(t_a^i) + 2\pi k F_0(t_a^i)(t - t_a^i) \tag{1.35}$$

On suppose que la fréquence de voisement maximale $F_{max}(t)$ est également constante dans le voisinage de t_a^i.
Les harmoniques $\widehat{h(t)}$ de la fréquence fondamentale F_0 peuvent être vues comme la somme de fonctions exponentielles complexes (équation 1.36).

$$\widehat{h(t)} = \sum_{k=-L}^{L} A_k(t_a^i) e^{j2\pi k F_0(t_a^i)(t - t_a^i)} \text{ avec } L = \frac{F_{max}(t_a^i)}{F_0(t_a^i)} \tag{1.36}$$

Avec L le nombre d'harmoniques choisies et F_0 la fréquence fondamentale à l'instant d'analyse t_a^i, $A_k(t_a^i)$ exprime l'amplitude complexe de la $k^{\text{ème}}$ harmonique et $A_{(-k)} = A_k^*$, le conjugué complexe de A_k.

Les fonctions exponentielles sont déterminées en cherchant à minimiser une erreur quadratique pondérée entre le signal d'origine et le signal de synthèse (équation 1.37).

$$\epsilon = \sum_{t=t_a^i - N}^{t_a^i + N} \omega^2(t)(s(t) - \hat{h}(t))^2 \tag{1.37}$$

Avec $s(t)$ le signal original, N l'entier le plus proche de la période de F_0, et $\omega(t)$ une fonction de pondération.

Comme nous nous intéressons à l'enveloppe de la partie harmonique on extrait ensuite des coefficients cepstraux discrets issus du cepstre discret du signal analysé (équation 1.38). Ces coefficients seront stockés pour chaque fenêtre d'analyse.

Le cepstre discret est obtenu en minimisant l'erreur quadratiqueϵ:

$$\epsilon = \sum_{k=1}^{L} || \log a_k - \log |S(f_k; \vec{c})| ||^2 \tag{1.38}$$

Avec $|S(f_k; \mathbf{c})|$ le spectre de puissance qui est reliée au cepstre réel par:

$$\log |S(f_k; \vec{c})| = c_0 + 2\sum_{i=1}^{p} c_i \cos(2\pi f i) \tag{1.39}$$

$\vec{c} = [c_0, \cdots, c_p]^T$ représente les coefficients cepstraux réels. Pour éviter les problèmes d'estimation de l'enveloppe avec p grand, on utilise une technique de régularisation du spectre (voir par exemple [Cappé et Moulines, 1994, Paliwal et Kleijn, 1995]).

Synthèse de la partie harmonique

Les paramètres extraits lors de l'analyse harmonique ($\vec{c}$, F_0) sont utilisés pour la re-synthèse du signal de parole. A chaque instant de synthèse (synchrone avec la F_0 de synthèse) on utilise une somme de fonctions sinusoïdales pour reconstituer les harmoniques (équation 1.40). Les amplitudes de celles-ci sont directement extraites des coefficients cepstraux discrets calculés lors de l'analyse et les phases sont extraites par re-échantillonnage de l'enveloppe spectrale de phase à chaque instant de synthèse. Ensuite, une opération de superposition et addition de chaque fenêtre temporelle synthétisée reconstitue le signal de parole toutes les 10 [ms].

La partie harmonique peut être reconstituée pour chaque fenêtre i:

$$\hat{h}(t) = \sum_{k=0}^{L(t_s^i)} a_k(t_s^i) cos(\phi_k(t_s^i) + 2k\pi F_0(t_s^i) t) \tag{1.40}$$

Avec:
- $t = [0, 1, \cdots, N]$, N la longueur d'une fenêtre de synthèse.
- $a_k(t_s^i)$ l'amplitude à l'instant t_s^i de synthèse, $\phi_k(t_s^i)$ la phase à l'instant t_s^i.
- $F_0(t_s^i)$ la valeur de la fondamentale à l'instant t_s^i.

L'amplitude instantanée $a_k(t)$ est estimée par interpolation linéaire entre les instants de synthèse i et $(i+1)$:

$$a_k(t) = a_k^i + \frac{a_k^{i+1} - a_k^i}{t_s^{i+1} - t_s^i} \cdot t \text{ , pour } t_s^i \leq t < t_s^{i+1}$$

La phase à l'instant t_s^{i+1} est prédite de la phase en t_s^i :

$$\phi_k^{i+1} = \phi_k^i + 2k\pi F_{0m}(t_s^{i+1} - t_s^i)$$

Avec F_{0m} la moyenne des fréquences fondamentales en t_s^{i+1} et t_s^i.
Ensuite on essaie de s'approcher de la valeur prédite en ajoutant à la phase le terme $2\pi M_k$ avec $(M_k \in I\!N)$ calculé de la manière suivante:

$$M_k = \left\langle \frac{1}{2\pi}(\phi_k^{i+1} - \phi_k^i) \right\rangle \text{ , avec } \langle \bullet \rangle \text{ la fonction d'arrondi à l'entier le plus proche}$$

Finalement on estime la phase instantanée $\phi_k(t)$:

$$\phi_k^i + \frac{\phi_k^{i+1} - 2\pi M_k - \phi_k^i}{t_s^{i+1} - t_s^i} \cdot t \text{ , pour } t_s^i \leq t < t_s^{i+1}$$

Analyse de la partie bruit

Toutes les parties non voisées de la parole peuvent être vues comme une source de bruit passée à travers des filtres (voir la section 1.2.2). Ici, la fonction de densité spectrale du bruit est estimée en utilisant un filtre tous pôles d'ordre 16. Les coefficients du filtre sont calculés sur une fenêtre de 40 [ms] autour de l'instant d'analyse t_a^i . La variance du signal original est estimée pour en extraire le gain du filtre. On ne mémorise que les coefficients de réflexion k_i (voir section 1.2.2). Comme une détection de la fréquence de voisement maximum (section 1.3.3) est effectuée pour chaque trame, le bruit peut être analysé aussi dans la partie voisée du signal de parole.

Synthèse de la partie bruit

Le bruit est estimé pour chaque instant de synthèse en injectant un bruit gaussien à variance unitaire à travers un filtre en treillis normalisé [Stylianou, 1996]. Les éléments du filtres sont les coefficients de réflexion k_i estimés au moment de l'analyse.

1.4 Entrons dans la thèse

Mises à part quelques applications pionnières (voir par exemple [Naik et Doddington, 1987, Gray et Kopp, 1994, Setlur et Jacobs, 1995]), pour la plupart, les systèmes vocaux de vérification d'identité sont sur le point de passer des laboratoires aux **applications industrielles**, ce qui soulève un certain nombre de problèmes que nous découvrirons dans le **chapitre 2**.

En plus des aspects de paramétrisation de la voix d'un locuteur permettant son analyse et sa reconnaissance, cette thèse aborde quelques points importants à résoudre pour franchir ce pas avec une fiabilité suffisante. En effet, bien que certains systèmes exhibent actuellement moins d'1% d'erreur lorsqu'on teste leurs qualités intrinsèques, quand un système automatique doit interagir directement avec un grand nombre de personnes, des difficultés supplémentaires surgissent, tels **l'imposture** (occasionnelle ou criminelle), **le réglage d'un point de fonctionnement** d'une application ou le **manque de données d'entraînement** pour la modélisation des locuteurs. Pour compenser les erreurs de modélisation, nous aborderons aussi les aspects de **fusion des décisions** de plusieurs systèmes permettant d'augmenter la fiabilité globale du système résultant.

Nous montrons dans cette thèse que l'étude des paramètres discriminant les locuteurs par décomposition du signal de parole permet d'augmenter la robustesse à l'imposture par transformation spectrale, cela sans pertes de performances (**chapitre 3**). La modélisation et la mémorisation des caractéristiques d'un locuteur nécessitent l'utilisation d'algorithmes capables de capturer des paramètres dont on ne connaît pas *a priori* les caractéristiques de linéarité. C'est pourquoi nous utilisons des modèles estimés à partir des données. Toutefois, les modèles performants nécessitent la détermination de nombreux paramètres et donc consomment beaucoup de données et de temps de calcul. Le **chapitre 4** montrera qu'il est possible, sans perte de performance, de trouver des systèmes avec peu de paramètres à estimer, lorsque l'on tire parti de toute l'information à disposition dans une application. La logique de décision actuellement utilisée pour la reconnaissance du locuteur est basée (pour les systèmes statistiques, section 1.2.4) sur une règle de décision Bayesienne. Nous montrons dans le **chapitre 5** que cette règle doit être complétée pour tenir compte des erreurs de modélisation. Nous montrons également que ces erreurs de modélisation peuvent être en partie compensées par la fusion des décisions de plusieurs algorithmes, offrant ainsi un accroissement de performances intéressant (**chapitre 6**).

Les résultats produits tout au long de ce document ont été obtenus en utilisant un **système de référence** à l'état de l'art dont la description exacte est donnée dans l'**annexe A**. Ce système servira de base à toutes les modifications que nous opérerons soit dans la paramétrisation soit dans la modélisation de la parole d'un locuteur.

Les données d'analyse et de test dont nous disposons ont aussi des conséquences importantes sur les résultats. Nous n'utiliserons ici que des **bases de données** de parole ayant été enregistrées sur des **lignes téléphoniques**. Ce choix peut être justifié par le fait que la plupart des applications en reconnaissance de locuteur, actuelles et futures, utilisent le téléphone comme support. Mais aussi parce que le signal téléphonique étant de relativement mauvaise qualité (bande passante réduite, distorsions dues au canal, codage GSM, etc...), les performances obtenues sur celui-ci seront validables sur un signal de meilleure qualité. Les bases de données utilisées dans cette thèse sont décrites dans **l'annexe B**. Notons encore qu'il est important d'obtenir des résultats sur des bases de données contenant beaucoup de locuteurs parce que les différences de variabilités intra-locuteur peuvent être grandes et pour avoir des résultats statistiquement valides. Cependant, comme nous ne disposons malheureusement pas de base de données pour tous les phénomènes à l'étude, nous avons recouru à des techniques différentes. Par exemple, il n'existe aucune base de données accessible permettant d'effectuer des tests durant lesquels un imposteur essaie de modifier ses caractéristiques vocales de manière à ce qu'elles se rapprochent de celles du locuteur qu'il cherche à imiter. Nous avons eu recours dans ce cas à un système d'analyse/synthèse de la parole (voir la section 1.3) qui nous a permis de transformer artificiellement et automatiquement de la voix (voir le chapitre 3).

Chapitre 2

Application générique

2.1 Introduction

Les systèmes de reconnaissance du locuteur doivent maintenant franchir la porte des laboratoires pour déboucher sur des applications en grandeur réelle. Ce chapitre est consacré aux différentes expériences que nous avons effectuées pour atteindre ce but. Il décrit le premier système que nous avons mis au point pour les Télécoms Suisse [van Kommer, 1995]. Partant des erreurs et problèmes de ce système (section 2.7), nous discuterons d'un certain nombre d'aspects théoriques et pratiques qui se sont posés durant sa mise au point et son utilisation. Cette application avait pour but de mettre au point un système de démonstration (section 2.2) et un système prototype, pour de nouveaux services de télécommunications [Moser, 1997]. Elle a également servi de système générique pour les projets européens CAVE et M2VTS (voir l'annexe D.1). Le premier système avait pour but d'évaluer les performances d'une application de vérification du locuteur pour des services téléphoniques. Le second système (nommé Voice Telecom Card), se basant sur la technologie du premier, avait pour but d'augmenter l'ergonomie des services, en supprimant l'usage des touches du téléphone lors de l'interaction avec le serveur vocal. Nous ne parlerons ici que du système qui nous a permis de mettre au point la technologie du système de vérification d'identité par la parole.

2.2 Description du système de démonstration

L'application consiste à sécuriser l'accès à un serveur d'information téléphonique personnalisé. Le système est basé sur une liaison téléphonique RNIS. Cette application comporte 2 phases : l'inscription et l'accès au service.

– Lors de l'**inscription** (voir la figure 2.1), le locuteur prononce ses nom, prénom, adresse, tous les chiffres de 0 à 9 en séquence et 5 fois son code client composé de 7 chiffres. Les modèles des différentes méthodes utilisées sont estimés à partir de ces données. On utilise une petite base de données acquise préalablement (Polycode-γ, voir l'annexe B.1) pour effectuer des accès client et d'imposture qui permettront ainsi de déterminer le point de fonctionnement du système (seuil *a priori*).

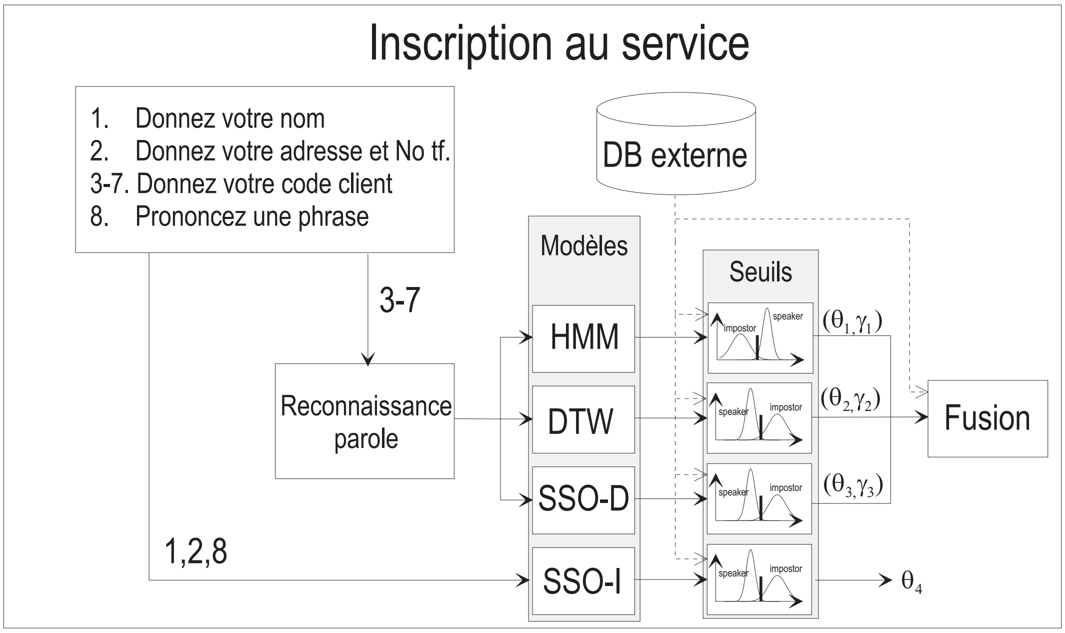

FIG. 2.1 – *Schéma de principe de l'inscription au service, calcul des modèles de locuteurs.*

– Lors de l'**accès** (voir la figure 2.2), le locuteur prononce une fois son code client, celui-ci étant ensuite vérifié par le système. La vérification se déroule en deux étapes:

1. Tout d'abord, la séquence de chiffres est identifiée en utilisant un système de reconnaissance de la parole indépendante du locuteur basé sur une technique HMM (voir l'annexe A.2). Un système de correction d'erreurs (ECC) a été mis en place, permettant de corriger une erreur de reconnaissance et d'en détecter 2.
2. La séquence de parole est ensuite comparée chiffre par chiffre aux références du locuteur correspondant au code client reconnu durant la première phase. Selon l'algorithme, une somme ou moyenne des vraisemblances (distances) de chaque chiffre est ensuite effectuée. Finalement, on compare le résultat obtenu à un seuil de décision qui permettra d'accepter ou de rejeter le locuteur qui accède au service. Si la comparaison ne peut être effectuée de façon assez fiable, le système peut décider de douter. En cas de doute, une question est posée au locuteur et une vérification indépendante du texte est effectuée, aboutissant au rejet ou à l'acceptation de l'appelant. La figure 2.2 explique les différentes étapes de l'accès au service.

Remarquons que, comme cette démonstration était destinée à des gens de passage, seule une session d'enregistrement était à notre disposition pour estimer les modèles de vérification.

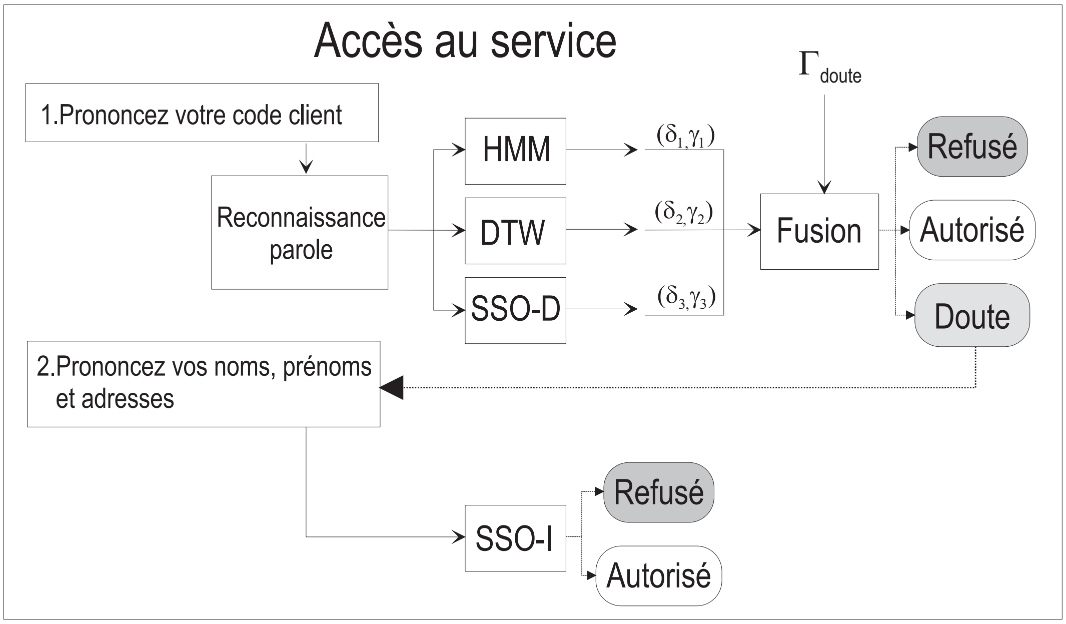

FIG. 2.2 – *Schéma de principe de l'accès, prise de décision à 2 niveaux.*

2.3 Les algorithmes utilisés

Des tests préliminaires pour déterminer l'algorithme le plus efficace à résoudre la tâche de vérification du locuteur nous ont permis de constater que les performances de chacun des systèmes était différentes selon les locuteurs (voir la section 2.6). De plus, ayant à disposition les capacités de parallélisation élevées des processus que l'informatique offre de nos jours, nous avons décidé d'utiliser trois algorithmes de vérification simultanément. Les algorithmes que nous utilisons ici ont été décrits dans l'introduction (section 1.2.3), nous nous contentons ici d'en rappeler quelques points importants.

2.3.1 Paramétrisation de la parole

Nous avons choisi d'utiliser 12 coefficients cepstraux issus de la prédiction linéaire (LPCC, voir section 1.2.2), l'énergie, ainsi que les dérivées et accélérations constituant un vecteur de 39 composantes. L'extraction s'effectue toutes les $10\,[\mathrm{ms}]$ sur une fenêtre de $25\,[\mathrm{ms}]$. Nous utilisons également une fenêtre de Hamming et une pré-accentuation de 0.94 (voir la section 1.2.2).

2.3.2 Statistiques du second ordre avec mesure de sphéricité (SSO+S)

– Lors de la **phase d'entraînement**, nous calculons une matrice de covariances Σ_{X} (équation 1.12, voir chapitre 1) de la séquence X de référence du locuteur. Si plusieurs séquences sont à disposition, la matrice est calculée sur toutes les séquences.

– Lors de la **phase de test**, nous estimons la matrice de covariances de la séquence de test Y et nous utilisons une mesure de sphéricité symétrique $\mu_{AH}(\Sigma_{\mathrm{X}}, \Sigma_{\mathrm{Y}})$ (voir la section 1.2.3).

Avantages des SSO+S

Les matrices de covariances munies d'une mesure symétrique sont intéressantes car leur calcul est très simple donc très rapide. En effet aucune extraction explicite des valeurs propres n'est nécessaire, seules les traces des produits de $(\Sigma_{\mathrm{X}} \cdot \Sigma_{\mathrm{Y}}^{-1})$ et $(\Sigma_{\mathrm{Y}} \cdot \Sigma_{\mathrm{X}}^{-1})$ sont calculées. On recalcule ici une matrice de covariances sur une longue période temporelle de manière à extraire les caractéristiques du conduit vocal à long terme.

Inconvénients des SSO+S

La mesure SSO+S ne permet que de distinguer des caractéristiques du locuteur qui sont stables tout au long de la phrase qu'il a prononcée. Il n'est pas possible de tenir compte de caractéristiques variables temporellement.

2.3.3 Dynamic Time Warping (DTW)

– Lors de la **phase d'entraînement**, les paramètres du locuteurs sont simplement stockés, par exemple pour chaque mot de chaque client.
– Lors de la **phase d'exploitation**, la distance finale D_f entre une référence et la séquence de test est calculée comme la somme des distances partielles (δ_i) entre vecteurs le long du chemin optimal, pondérée par des contraintes locales (voir la section 1.2.3 pour le fonctionnement de la DTW). Le chemin optimal est le chemin dont la distance globale est minimale. Afin d'augmenter l'efficacité de la méthode, les chemins possibles sont limités et toute une gamme de contraintes locales peuvent être imposées. Lorsque l'on possède plusieurs exemplaires de référence R_j, le score de sortie final est la moyenne de toutes les distances calculées sur toutes les références.

Avantages de la DTW

L'algorithme de DTW est rapide, bien adapté à la parole parce que capable de tenir compte des variations temporelles du signal. Il ne nécessite pas beaucoup de données pour fonctionner correctement.

Inconvénients de la DTW

La DTW est très sensible à la segmentation du signal. En effet, si le point de départ du calcul dynamique n'est pas bon, l'algorithme peut rapidement diverger du chemin optimum. Il existe cependant des possibilités de corriger partiellement cette erreur (voir pour cela [Furui, 1981b], [Myers et Rabiner, 1981, Rabiner et Juang, 1993]). De plus, ses capacités de modélisation des variabilités intrinsèques du locuteur sont relativement limitées, puisqu'il n'est capable d'estimer que des points sur le meilleur chemin et non des distributions.

2.3.4 Modèles de Markov cachés (HMM)

Nous utilisons deux types de modèles HMM pour chaque chiffre du code personnel du locuteur (voir la section 1.2.3 et l'annexe A pour plus de détails sur le système HMM):

1. *Un modèle du monde*, créé à partir d'une base de données (Polyphone, décrit dans l'annexe B.4) comportant un grand nombre de locuteurs dont on a extrait 300 répétitions de chaque chiffre. Les paramètres de ce modèle sont estimés par un entraînement standard (initialisation par l'algorithme de Viterbi, réestimation de paramètres par Baum-Welch [Young et Bloothooft, 1997]) ce modèle est identique pour tous les clients.
2. *Un modèle du client*, qui utilise comme paramètres initiaux ceux du modèle du monde, et dont on réestime les paramètres pour chaque locuteur avec les données de celui-ci. Ce modèle est donc dépendant du client.

   Les modèles ont tous la même structure HMM gauche-droite constituée d'un état par phonème et un état par transition entre phonèmes [Gravier, 1995].

   - Lors de la **phase d'entraînement**, les paramètres du modèle HMM dépendants du locuteur sont estimés, chiffre par chiffre, en utilisant 2 itérations de l'algorithme de Baum-Welch.
   - Lors de la **phase d'accès**, pour chaque chiffre du code que prononce le locuteur, on calcule le rapport de vraisemblances (voir la section 1.2.4, équation 1.26):

   $$LLR(M_C, M_W, O_t) = \log \mathscr{L}(O_t, M_C) - \log \mathscr{L}(O_t, M_W)$$

     avec $\log \mathscr{L}(O_t, M_C)$ et $\log \mathscr{L}(O_t, M_W)$ les vraisemblances de la séquence O_t obtenues pour le modèle du client et le modèle du monde respectivement. Les vraisemblances obtenues sur chaque chiffre sont ensuite sommées pour former le score de sortie final.

Avantages des HMM

Les HMM sont capables, comme la DTW, de modéliser des variabilités temporelles. Ils ont, de plus, la possibilité de modéliser des variations locales dans les caractéristiques des locuteurs. L'utilisation d'un modèle de monde offre un test statistique puissant (voir section 1.2.4) et une normalisation des scores de sortie qui permet de comparer les scores des différents locuteurs entre eux.

Inconvénients des HMM

Les HMM présentent cependant l'inconvénient de comporter beaucoup de paramètres, donc de nécessiter beaucoup de données d'entraînement pour les estimer. La procédure que nous utilisons ici, et qui consiste à utiliser la somme des scores sur toute une séquence, ne tient pas compte du pouvoir discriminant différent que pourrait avoir chaque mot pour différents locuteurs.

2.4 Détermination du seuil de décision *a priori*

Dans toute application de vérification du locuteur, il est nécessaire de prendre, à un moment ou à un autre, la décision d'accepter ou de rejeter la séquence de parole que l'on est en train de tester. Pour ce faire, il est nécessaire de déterminer un seuil de décision avant d'effectuer la vérification (seuil *a priori* voir la section 1.2.4). Nous avons choisi 2 méthodes de réglage du seuil. Chacune d'entre elles tire parti de manière différente des distributions des scores obtenus par les clients ou les imposteurs sur une base de réglage. La détermination des paramètres nécessaires au réglage de ces seuils s'est faite sur les données de Polycode-γ, alors que les tests se sont effectués sur Polycode (voir annexe B.1 pour les subdivisions de la base de données Polycode).

2.4.1 Seuil EER global

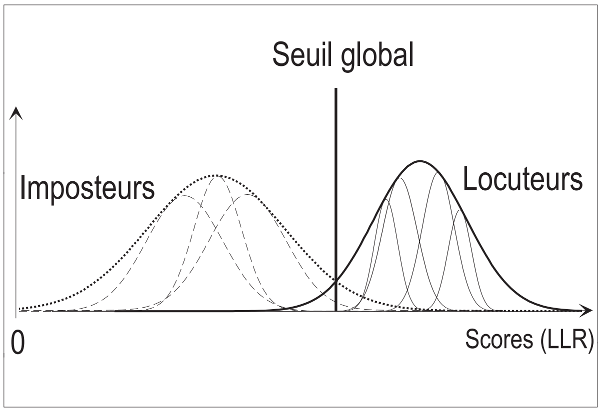

FIG. 2.3 – *Détermination du seuil global sur les distributions de tous les locuteurs et de leurs imposteurs.*

Ce seuil est déterminé en calculant la distribution globale des scores de tous les clients de la base d'évaluation et de toutes les tentatives d'imposture sur ceux-ci (voir la figure 2.3). Ce seuil global est sensé déterminer un point de fonctionnement (à l'EER dans notre cas) qui serait indépendant des clients de l'application, et que nous pourrions directement utiliser pour des clients nouvellement enregistrés. Cette approche est théoriquement valide, en tous cas lorsque l'on utilise des algorithmes basés sur une décision Baysienne (voir la section 1.2.4 pour les considérations théoriques d'une décision de type Baysien). De manière plus pratique, un seuil global peut être déterminé de façon plus aisée lorsque l'on a peu de données d'entraînement pour les clients à disposition, puisqu'on utilise les scores de tous les clients pour calculer une seule distribution.

2.4.2 Seuil individuel établi sur distribution d'imposture

Furui a montré [1981a, 1994] que lorsque très peu de données d'entraînement sont à disposition, il y a certains avantages à estimer le seuil de décision uniquement en utilisant les scores des

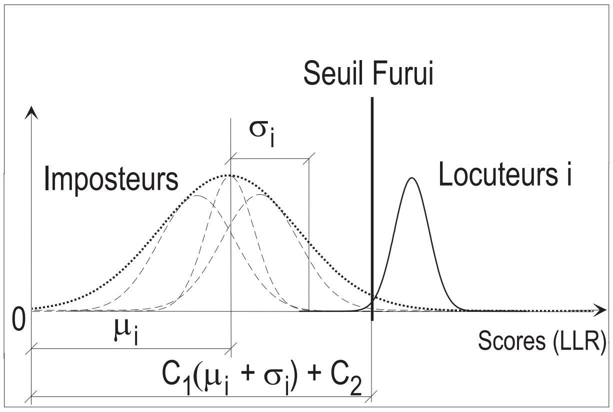

FIG. 2.4 – *Estimation des paramètres μ_i et σ_i pour chaque locuteur. Ici nous calculons ($\mu_i + \sigma_i$) car les scores de sorties sont des vraisemblances.*

tentatives d'imposture sur un client i. Cette détermination du seuil s'effectue en 2 étapes:

1. Nous calculons sur une base de données de développement (Polycode-γ) les distributions des scores de quelques clients et d'imposteurs sur lesquelles nous estimons un seuil à l'EER (S_{EER}). Nous faisons l'hypothèse que les accès d'imposture suivent une loi normale $\mathcal{N}(\mu_{\mathrm{N}}, \sigma_{\mathrm{N}})$. Puis nous faisons l'hypothèse que la distribution des valeurs des accès d'imposture sur tous les clients d'une application (voir la figure 2.4) peut être estimée en utilisant une régression linéaire (voir la figure 2.5) dans le plan ($\mu_{\mathrm{N}} + \sigma_{\mathrm{N}}, S_{\mathrm{EER}}$), ce qui nous permet d'estimer les deux paramètres de la droite de régression C_1 et C_2. Ces deux constantes sont valables pour tous les clients qui s'enregistreront dans l'application. Le choix de la distribution de ($\mu_{\mathrm{N}} + \sigma_{\mathrm{N}}$) est valable lorsque nous cherchons un seuil de décision entre des distributions de *vraisemblances*. Si nous cherchons un seuil pour séparer des distributions de *distances*, il nous faudra alors estimer la droite ($\mu_{\mathrm{N}} - \sigma_{\mathrm{N}}$).
2. Lors de l'inscription d'un nouveau client x, son seuil est estimé en utilisant les deux constantes C_1 et C_2 de la manière suivante:

$$Seuil_x = C_1(\mu_{ix} + \sigma_{ix}) + C_2 \tag{2.1}$$

   Avec μ_{ix}, σ_{ix} les paramètres de la gaussienne $\mathcal{N}(\mu_{ix}, \sigma_{ix})$ estimée sur les scores d'imposture pour le nouveau client. Cette manière de déterminer le seuil part du principe que les distributions des scores d'imposture sont suffisantes pour calculer le bon point de fonctionnement de l'application, ce qui est évidemment une faiblesse, car les distributions des scores des clients sont aussi importantes, comme nous le verrons par la suite.

2.5 Combinaison de méthodes

Dans le problème qui nous préoccupe ici, la combinaison de méthodes vise à obtenir, par l'utilisation de différentes mesures, une meilleure information globale sur le locuteur. En effet,

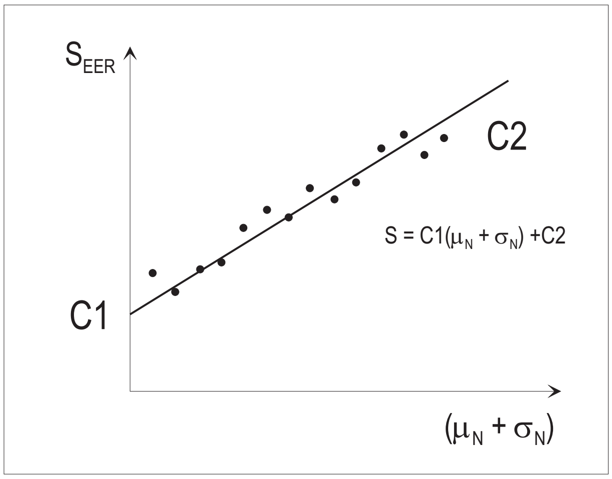

FIG. 2.5 – *Détermination des constantes de la méthode Furui par régression linéaire.*

si chacune de ces mesures ou méthodes donne une image impropre mais différente du locuteur, leur combinaison devrait améliorer le résultat final. Dans le cas de la vérification du locuteur, la réponse finale du système est simple: soit on accepte le segment de parole testé comme appartenant au locuteur proclamé, soit on le rejette. Cette prise de décision peut se faire à différents niveaux (voir [Dasarathy, 1994, Antony, 1995]). Soit, par exemple, chaque méthode prend une décision partielle et on les combine pour prendre une décision finale, soit on combine les mesures fournies par chaque méthode et on prend une décision sur le résultat de leur fusion. L'application que nous avons mise en oeuvre ici utilise la fusion des décisions, mais, nous le verrons au chapitre 6, la fusion des scores est aussi une approche valide.

2.5.1 Combinaison des décisions

La combinaison de décisions nécessite un algorithme capable de fusionner de manière intelligente les qualités de chaque méthode, tout en ne répétant pas leurs erreurs. La connaissance *a priori* de la qualité de chacune des méthodes peut, par exemple, nous aider à pondérer leurs décisions (voir [Thévenaz, 1993]). Cependant, dans le cas de notre application, nous n'avons pas de données pour estimer une telle pondération *a priori*. La décision de chacune des méthodes est considérée comme indépendante des autres. Dès lors, une solution simple consiste à opérer un vote majoritaire parmi les décisions de chaque méthode. Bien que n'ayant aucune information sur la qualité mutuelle des méthodes, nous pouvons connaître pour chacune d'elle la confiance avec laquelle elle prend sa propre décision. Cette mesure s'effectue en calculant à quelle distance du seuil la décision d'accepter ou de rejeter le segment de parole considéré a été prise.

Afin de combiner plus facilement les mesures de confiance, nous les avons normalisées de façon à ce que la confiance tende vers 0 lorsque l'on s'approche du seuil et tende vers 1 lorsque l'on s'en éloigne. Nous avons utilisé comme fonction d'accroissement de la confiance une $\frac{1}{2}$ sigmoïde dont la pente p est réglée expérimentalement. La valeur de la confiance est la distance entre le score

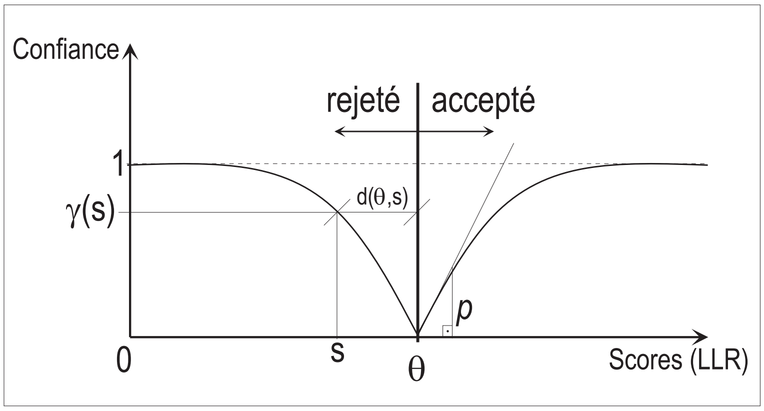

FIG. 2.6 – *Calcul de la confiance dans une décision. Chaque demi-sigmoïde a pour valeur 0 au seuil Θ, la pente p est un paramètre déterminé expérimentalement.*

s courant obtenu par la méthode et le seuil de décision Θ (équation 2.2). La figure 2.6 montre la fonction de confiance $c(s)$ par rapport au seuil.

$$c(s) = |\tanh \frac{(s - \Theta)}{p}| \tag{2.2}$$

Pour prendre sa décision, le système procède en 2 phases:

1. Il procède à un vote majoritaire.
2. Pour les méthodes qui sont dans la majorité, le système calcule la moyenne m de leur confiance qui est ensuite comparée à un seuil Γ.

   – Si ($m > \Gamma$), le système prend la décision de la majorité.
   – Si ($m \leq \Gamma$), le système **doute** et, dans ce cas, l'application redemande une phrase au client qui veut accéder (voir la figure 2.2).

   Ce seuil Γ, appelé ici seuil de doute, permet au système de ne pas décider d'accepter ou de rejeter la séquence. Le réglage de ce seuil de doute est lui aussi déterminé expérimentalement selon le niveau de sécurité désiré.

2.6 Résultats

Les résultats présentés ici sont ceux obtenus en utilisant la base de données Polycode (voir annexe B.1). Les modèles des différentes méthodes ont été estimés sur les données d'entraînement, les seuils ont été estimés sur Polycode-γ.

Le tableau 2.1 montre que le seuil global EER ne donne des résultats probants qu'avec les HMM, ce qui est somme toute conforme aux prédictions théoriques (voir la section 1.2.4), mais nous constatons cependant que le point de fonctionnement **n'est plus à l'EER**. Nous en verrons

Méthode	FR% (200 tests)	FA% (1800 tests)	HTER% (2000 tests)
DTW	31.5	21.8	26.6
SSO+S	22.2	30.3	26.2
HMM L/R	2.5	5.33	3.9
Décision combinée	8.0	3.2	5.6

TAB. 2.1 – *Performance des méthodes avec seuil EER global, HTER = (FA+FR)/2.*

l'explication au chapitre 5. Pour les 2 autres méthodes, le seuil global ne donne pas de résultats valables, probablement parce qu'aucune normalisation des scores de sortie n'est effectuée entre locuteurs.

Les performances du système de fusion de décision avec des seuils indépendants du locuteur (dernière ligne du tableau 2.1) sont elles aussi mauvaises, ce qui est compréhensible si deux méthodes sur trois font trop d'erreurs. Mais rien ne dit qu'avec une méthode de fusion plus performante on ne puisse pas être meilleur que la meilleure méthode.

Méthode	FR% (200 tests)	FA% (1800 tests)	HTER% (2000 tests)
DTW	23.5	7.67	15.8
SSO+S	14.0	5.3	9.9
HMM L/R	5.5	2.7	4.1
Décision Combinée	2.0	2.7	2.3

TAB. 2.2 – *Performance des méthodes,seuil FURUI, HTER = (FA+FR)/2.*

Part contre le tableau 2.2 indique que la détermination du seuil de décision avec la méthode de Furui permet de prendre des décisions plus sûres (i.e. avec moins d'erreurs) pour chacune des méthodes. On remarque que le point de fonctionnement a été déterminé sans tenir compte des distributions de scores clients, car le taux de faux rejets est 2 fois supérieur aux fausses acceptations. C'est là un des défauts principaux de cette méthode.

Cependant, la fusion des décisions par vote majoritaire apporte une amélioration des performances du système. Comme l'indique la dernière ligne du tableau 2.2, la fusion permet de supprimer presque totalement le biais dû au seuil de Furui et le point de fonctionnement réel se trouve proche de celui estimé *a priori* à l'EER.

Finalement, le tableau 2.3 montre comment il est possible de diminuer encore les erreurs de l'application lorsqu'on modifie le seuil de doute (voir section 2.5.1). Il est ainsi possible de régler la confiance dans les décisions des méthodes selon le niveau de sécurité désiré de l'application. Remarquons que la diminution des erreurs du système ne se fait pas de manière linéaire et qu'un taux de doute de 20% permet de diminuer les taux d'erreur de moitié.

Seuil de doute	FR% (200 tests)	FA% (1800 tests)	HTER% (2000 tests)	Doute% (2000 tests)
0.2	2.0	2.7	2.3	0.0
0.5	2.0	2.3	2.15	0.8
0.7	2.0	1.1	1.55	10.1
0.8	1.0	0.9	0.95	20.2

TAB. 2.3 – *Performances lors de changement de seuil de doute, HTER = (FA+FR)/2*

2.7 Constatations

Le système de démonstration, bien qu'à l'état de l'art pour un système en fonctionnement, au moment où il a été conçu, recèle un certain nombre de carences et faiblesses détaillées ci-après:

- La paramétrisation utilisée n'est pas optimale, car l'utilisation de l'énergie normalisée du signal a montré son efficacité sur du signal téléphonique en reconnaissance du locuteur (voir [CAVE,1998]).
- La paramétrisation utilisée est issue du domaine de la reconnaissance de la parole et a été optimisée pour atténuer les variabilités dues aux locuteurs, et n'est donc pas vraiment adaptée au problème de la reconnaissance du locuteur.
- Le manque de données d'apprentissage pose un problème important dans l'estimation de modèles statistiques un peu complexes.
- Le réglage du point de fonctionnement du système s'est montré fort complexe et non totalement maîtrisé. Une étude de ce problème semble nécessaire.
- La mesure d'un seul point de fonctionnement n'est pas suffisante et il vaudrait mieux donner des résultats sur toute la plage de fonctionnement du système.
- La fusion des décisions s'est montrée fort prometteuse et mérite qu'on s'y attarde, même si fondamentalement elle ne donne aucune information sur les systèmes fusionnés.
- Le nombre de paramètres de réglage d'un système de vérification du locuteur étant très élevé et chacun de ceux-ci ayant une influence plus ou moins grande sur le fonctionnement du système, il semble nécessaire de travailler avec un **système de référence** à partir duquel nous pourrons mesurer l'influence de la modification de certains de ces paramètres.
- De manière à évaluer notre système de référence par rapport à l'état de l'art il semble qu'il soit important de le mettre en concurrence avec d'autres sur des tests ayant un protocole commun.
- Il est important d'étudier les particularités du signal de parole comme les variabilités intra- et inter-locuteurs, si l'on veut mettre en place des systèmes de reconnaissance du locuteur à large échelle.
- Comme le point de fonctionnement dépend des distributions des imposteurs, il est nécessaire d'approfondir les phénomènes d'imposture.

Nous porterons notre contribution à la résolution de ces problèmes au cours des prochains chapitres. En effet, dans le **chapitre 3**, nous analyserons la paramétrisation du locuteur en essayant de découvrir quelles sont les spécificités de celle-ci. Puis nous tenterons d'agir sur elle de manière à la transformer, ce qui nous permettra d'éprouver sa robustesse et de traiter du problème d'imposture. Le **chapitre 4** montrera comment il est possible de décomposer le problème de la reconnaissance du locuteur dépendante du texte en problèmes plus simples et nécessitant moins de données d'entraînement. Le **chapitre 5** permettra d'expliquer de manière théorique quels sont les problèmes qui se posent lorsque l'on est confronté à l'estimation de seuils de décision *a priori*. Puis nous nous appliquerons à montrer par l'exemple, dans le **chapitre 6**, que la fusion de décisions est un apport intéressant dans les applications pratiques.

Chapitre 3

Paramétrisation et transformation de locuteurs

3.1 Introduction

Comme nous avons pu le constater au chapitre 1 (figure 1.1), l'information du signal de parole émis par un locuteur comporte une partie dépendante du message et une partie caractéristique du locuteur. La modélisation du locuteur se fait en extrayant un jeu de vecteurs de paramètres du signal de parole puis en créant un modèle dépendant du locuteur à partir de ces données (voir la section 1.2.3). Dans le chapitre 3, nous tenterons de déterminer s'il existe un jeu de paramètres qui puissent représenter au mieux les caractéristiques du locuteur.

Un certain nombre d'auteurs se sont penchés sur ce problème auparavant [Atal, 1974], Furui [1981a, 1991] ou [Colombi *et al.*, 1993], ou encore [Thévenaz, 1993, Algazi *et al.*, 1993], [Homayounpour et Chollet, 1994], [Assaleh et Mammone, 1994], [Homayounpour, 1995], [Baudoin et Stylianou, 1996]. Cependant actuellement la paramétrisation utilisée pour la reconnaissance de la parole, bien que développée dans le but d'atténuer la variabilité des prononciations (locuteurs, accents, etc. . .), reste aussi utilisée pour la reconnaissance du locuteur. Les paramètres les plus utilisés dans les systèmes de reconnaissance de locuteurs à l'état de l'art sont les LPCC (voir section 1.2.2). Bien que cette paramétrisation donne des résultats de bonne qualité, en tout cas en laboratoire, plusieurs raisons nous ont déterminés à étudier cette paramétrisation en y détectant des indices plus spécifiques à la reconnaissance du locuteur:

– Etudier les paramètres les plus discriminants pour améliorer les performances des systèmes de reconnaissance du locuteur en fonctionnement dans un milieu ouvert (robustesse à l'imposture, au bruit).
– Rechercher un jeu de paramètres caractérisant le locuteur plutôt que la parole.
– Etudier l'évolution intra- et inter-locuteurs des paramètres les plus discriminants afin de comprendre l'évolution de la voix d'un locuteur au cours du temps.

Afin de déterminer quels sont les paramètres distinctifs du locuteur et quels sont ceux qui dépendent du message, nous avons choisi de travailler dans un contexte où le message est connu (reconnaissance du locuteur dépendante du texte). Pour ce faire nous avons utilisé la base de données Polycode (annexe B.1) dans laquelle les locuteurs ont prononcé des séquences de chiffres.

Ces séquences ont été segmentées en mots ou en phonèmes. Pour comprendre et identifier les paramètres importants, nous avons choisi de décomposer la parole en analysant ses caractéristiques fréquencielles. Enfin, nous postulons que les paramètres les plus discriminants sont ceux qui permettent de transformer la voix d'un locuteur pour ressembler au mieux à celle d'un locuteur cible. Pour ce faire, il faut un système qui permette d'**analyser** la parole, d'en extraire des paramètres que l'on pourra **transformer** et finalement qui nous permette de **re-synthétiser** ces paramètres pour générer un signal de parole (voir figure 3.1). Plusieurs métriques différentes peuvent être prises en considération pour évaluer une distance entre 2 locuteurs. Nous avons choisi ici de prendre comme critère les taux d'erreur (FA, FR, HTER, voir section 1.2.4) du système de référence HMM (voir annexe A). Plusieurs arguments motivent ces choix:

1. Tout d'abord, la volonté de comparer nos résultats à des systèmes de reconnaissance du locuteur à l'état de l'art. Comme ce genre de systèmes comporte un nombre de paramètres de réglage assez élevé, seule la comparaison des erreurs à un système de référence fixé au préalable peut nous donner un critère de mesure du système global.
2. Ensuite, l'étude des phénomènes d'imposture volontaire (donc de l'augmentation du taux de fausses acceptations) auxquels risquent d'être soumises les applications de reconnaissance du locuteur nous a paru importante avant de déployer ces systèmes à large échelle.

Dans un premier temps, pour déterminer les paramètres sur lesquels les transformations auront le plus d'impact, nous mesurerons l'évolution des performances du système de référence lorsque notre algorithme d'analyse/transformation/synthèse est appliqué sans transformation des paramètres (section 3.3). Ensuite, nous nous intéresserons aux phénomènes d'imposture et évaluerons les performances de la concaténation de segments de parole ainsi que les transformations appliquées aux imposteurs qui joueront le rôle de locuteurs *sources* alors que les clients seront les locuteurs *cibles*. Le nombre de fausses acceptations (FA) que les imposteurs ainsi transformés vont générer indiquera dans quelle mesure les paramètres transformés sont caractéristiques des locuteurs (section 3.4).

La figure 3.1 indique quelques algorithmes possibles pour chacune des étapes du système **analyse/transformation/synthèse** de la parole. La partie analyse du signal est issue de 2 modèles principaux: le modèle de perception de la parole (FFT) ou le modèle de production (calcul de la fonction de transfert) (voir 1.2.2). Nous utiliserons ici une analyse issue de la FFT (extraction de formants) et une analyse issue du modèle de production de la parole (analyse harmonique plus bruit). Nous utiliserons comme algorithmes de transformation des paramètres des méthodes basées sur des modifications de distributions statistiques. Les algorithmes de synthèse, quant à eux, dépendent directement des algorithmes d'analyse choisis. Bien que seuls les résultats obtenus en utilisant les systèmes automatiques de reconnaissance du locuteur soient considérés pour l'identification des paramètres, quelques indications subjectives de la qualité de la parole transformée seront donnés.

3.2 Analyse/synthèse de la parole

Afin de mesurer la qualité du système d'analyse/synthèse avant d'y effectuer des transformations, nous allons l'appliquer aux locuteurs de Polycode. Pour que notre système d'analyse/-

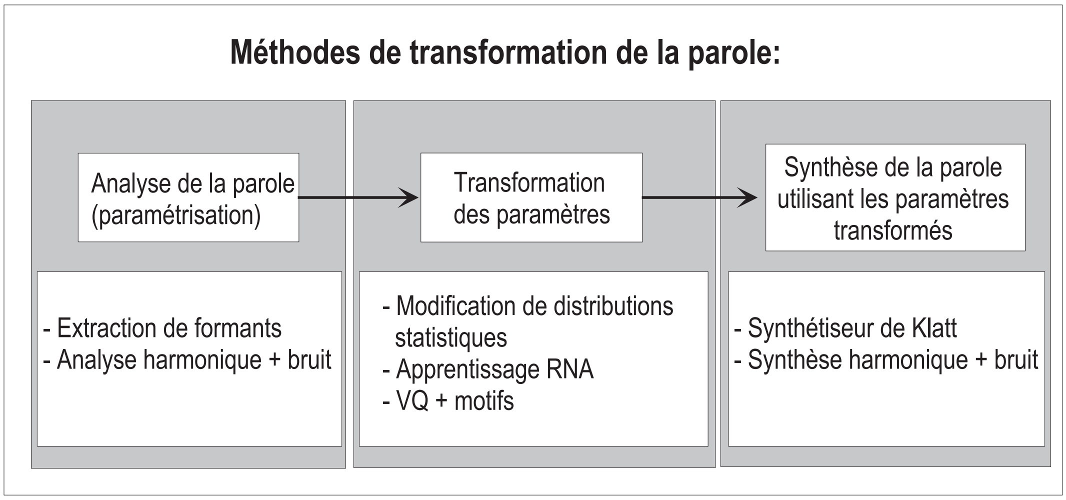

FIG. 3.1 – *Schéma bloc du processus de transformation de la parole.*

transformation/synthèse puisse fonctionner, nous désirons qu'il ait les propriétés suivantes:

- La partie du système qui effectue l'analyse de la parole doit extraire des paramètres que nous pourrons transformer et réutiliser lors de la synthèse.
- Le processus analyse/synthèse doit être de suffisamment bonne qualité pour que le système de référence décèle un minimum de déformations du signal analysé/synthétisé.

Pour simuler une analyse selon un modèle de perception, nous choisirons un système permettant de suivre les formants (voir la section 1.1) générés par la parole humaine et d'extraire à intervalles réguliers leur position en fréquence, leur nombre et leur largeur de bande, ainsi que la fréquence fondamentale. Ces paramètres seront directement injectés dans un synthétiseur de Klatt (voir section 1.3.2). Cependant ce système formants/Klatt (*FORM-KLATT*) a beaucoup d'inconvénients, le principal étant que l'extraction de formants ne fonctionne que sur les parties voisées.

Pour simuler une analyse selon un modèle de production, nous avons choisi un système Harmonique plus Bruit (H+N) (voir la section 1.3.3). Ce système représente l'état de l'art dans le domaine de l'analyse/synthèse de la parole [Stylianou, 1996]. Contrairement au *FORM-KLATT*, il est capable de modéliser aussi bien les parties voisées de la parole (représentées par des harmoniques de la fréquence fondamentale F_0) que les parties non-voisées (représentées par une source de bruit gaussienne passée dans un filtre dont les caractéristiques varient toutes les 40 [ms]). Comme il n'est pas possible de transformer directement les amplitudes lors de changement de F_0, les paramètres transmis seront des coefficients cepstraux régularisés (voir la section 1.3.3), qui peuvent être modifiés et qui nous permettrons de régénérer une enveloppe du spectre lors de la synthèse.

Le tableau 3.1 montre les résultats en terme de faux rejets (FR) lorsqu'on applique à la parole des clients de Polycode les systèmes *FORM-KLATT* et H+N et qu'on l'injecte dans le système de

Méthode	FR%
Système de référence	$2.3^{\pm 1.2}$
FORM-KLATT	$41.2^{\pm 3.2}$
H+N	$4.6^{\pm 1.8}$

TAB. 3.1 – *Taux de faux rejet (FR) lorsque l'on applique à la parole des clients de Polycode les systèmes d'analyse/synthèse FORM-KLATT et H+N et qu'on la ré-injecte dans le système de référence HMM.*

référence sans aucune adaptation de celui-ci. Sans surprise, l'approche H+N donne de meilleurs résultats que le système *FORM-KLATT*. En effet, les différences de performance sont faibles entre de la parole non modifiée et de la parole ayant traversé le modèle H+N. Ce résultat est confirmé par la figure 3.2 qui montre locuteur par locuteur l'évolution des moyennes et des écart-types des scores normalisés du système de référence avec ou sans l'application du modèle H+N.

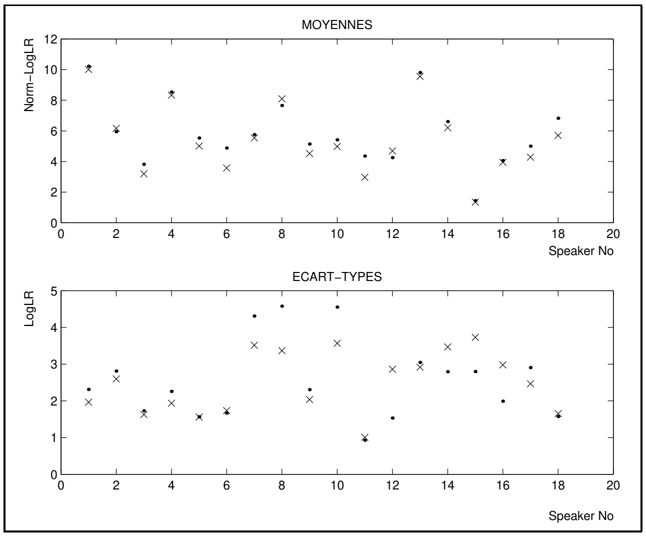

FIG. 3.2 – *Moyennes normalisées et écart-types des scores LLR locuteur par locuteur lors d'une synthèse H+N (×) comparés au système de référence (•).*

Puisque le modèle H+N permet de décomposer la parole en une partie harmonique déterministe et une partie bruit stochastique, il est possible d'évaluer comment ces deux parties contribuent à la discrimination des locuteurs. La section 3.3 va nous le faire découvrir.

3.3 Décomposition de la parole

Nous avons re-synthétisé la parole des clients et des imposteurs du système de référence en utilisant la partie harmonique seule du modèle H+N, puis la partie bruit seule et enfin les deux ensembles. Pour chacun de ces cas de figure, les modèles de monde et de client ainsi que le point de fonctionnement du système de référence ont été recalculés sur des données (entraînement et évaluation) ayant suivi le même processus d'analyse/synthèse. Une expérience complémentaire consistant à rajouter un bruit aléatoire à la partie harmonique seule nous a permis de diminuer notablement les fausses acceptations. Cela s'explique par la façon dont le modèle de référence calcule les scores. En effet, lorsque l'on ne re-synthétise que des harmoniques, tous les échantillons de parole qui ne contenaient que la partie bruit sont mis à zéro. Comme le zéro devient une valeur déterministe, sa modélisation par une distribution de probabilité gaussienne devient impossible et la vraisemblance de tels échantillons devient si faible qu'elle influe beaucoup sur la vraisemblance moyenne des observations avec un modèle HMM. Afin de ne pas trop perturber les mesures, tout en évitant ce problème de valeurs zéro, un bruit gaussien de faible intensité a donc été superposé aux harmoniques.

Le tableau 3.2 donne les résultats de ces expériences de décomposition de la parole. Ces résultats montrent que les parties bruit et harmonique analysées et re-synthétisées contiennent chacune de l'information sur le locuteur, mais la partie bruit permet une meilleure discrimination entre locuteurs. Cependant, seule la synthèse complète H+N conduit à des performances identiques au système de référence, confirmant que l'information sur l'identité des locuteurs se trouve bien dans les 2 parties harmonique et bruit. La figure 3.3 montre les courbes COR (voir section 1.2.4) des mêmes expériences. Celles-ci confirment qu'effectivement le système H+N complet obtient les mêmes performances qu'avec de la parole non modifiée.

Analysons maintenant plus en détails les conséquences d'une telle décomposition. Le système H+N peut être vu comme un filtre appliqué au signal d'entrée. Le premier étage de ce filtre, l'extraction de la fondamentale, consiste à trouver, dans les parties voisées, les endroits stables du signal, donc les endroits où l'on peut considérer que les coefficients du filtre sont constants. De même pour les parties non-voisées, on admet que celles-ci sont stables sur une période fixée. Entre l'étage d'analyse et de synthèse, seuls les coefficients de ces filtres (coefficients de réflexion k_i, voir 1.2.2) sont transmis pour la partie bruit. En ce qui concerne la partie voisée, la fréquence fondamentale F_0 et soit les amplitudes et phases, soit les coefficients cepstraux régularisés sont transmis.

Comme la modélisation H+N se base sur des hypothèses qui limitent la richesse du signal de parole, il serait intéressant de supprimer la contribution dans la synthèse H+N, soit de la partie bruit soit de la partie harmonique soit des deux réunies, afin de vérifier la quantité d'information caractéristique du locuteur contenue dans les parties résiduelles. Ce sont les expériences qui sont relatées dans le tableau 3.3 et la figure 3.4. Ces résultats amènent quelques commentaires:

- Le résidu de la suppression de toutes les parties modélisées par le système H+N est suffisamment discriminant pour que l'on puisse encore distinguer les locuteurs les uns des autres avec cependant une perte de qualité de la discrimination.
- Le modèle H+N et la parole non-filtrée par H+N ont des performances identiques.
- Comme il reste dans le signal auquel nous avons enlevé soit les harmoniques, soit le bruit, suffisamment d'informations pour que les performances du système de référence ne soient que peu dégradées, nous pouvons en déduire que le signal de parole est fortement redondant.
- Si l'écoute de la partie harmonique seule nous permet de reconnaître la voix d'un locuteur, l'écoute de la partie bruit ou du résidu H+N ne nous le permet plus, bien que l'on puisse

Méthode	FR%	FA%	HTER%
Système de référence	$2.3^{\pm 1.2}$	$4.2^{\pm 0.7}$	3.2
H+N complet	$3.2^{\pm 1.45}$	$3.1^{\pm 0.62}$	3.1
Bruit seul du H+N	$16.1^{\pm 3.0}$	$6.3^{\pm 0.8}$	11.2
Harmoniques seules du H+N	$31.4^{\pm 3.8}$	$18.4^{\pm 1.3}$	24.9
Harmoniques seules du H+N +(bruit aléatoire)	$31.7^{\pm 3.8}$	$5.7^{\pm 0.83}$	18.7

TAB. 3.2 – *Taux d'erreur lorsque l'on utilise comme pré-traitement de la parole les harmoniques seules, le bruit seul ou les deux parties du modèle H+N*

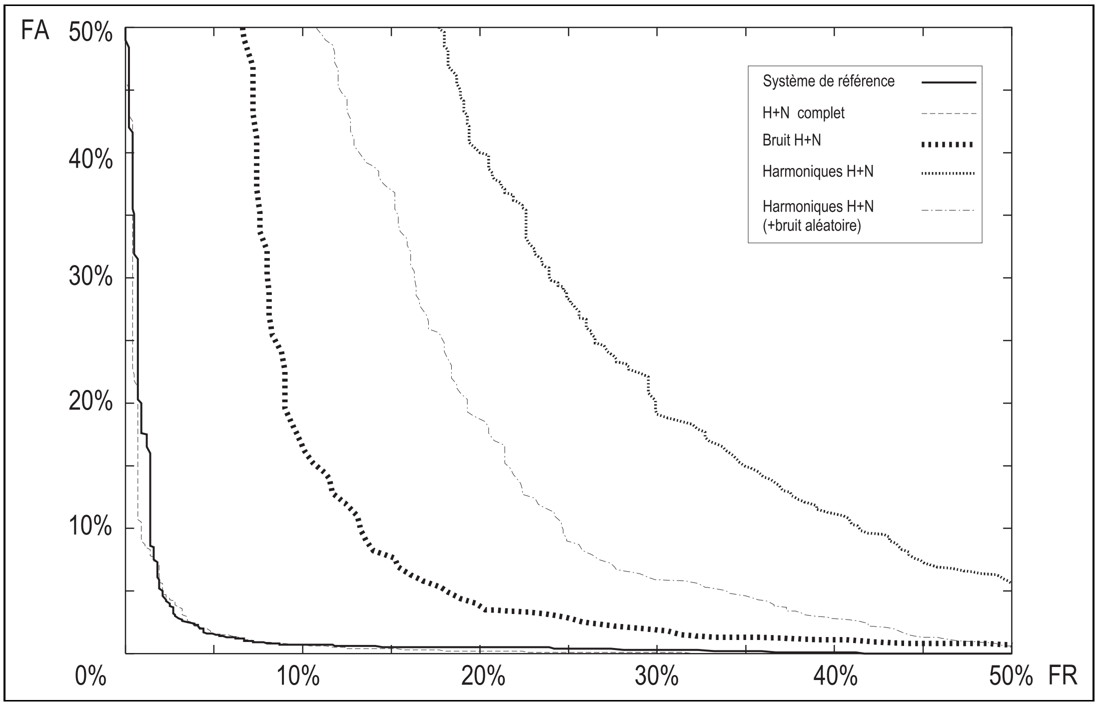

FIG. 3.3 – *Courbes COR des performances du système utilisant comme filtrage d'entrée le bruit ou les harmoniques du modèle H+N ou les deux.*

Méthode	FR%	FA%	HTER%
Système de référence	$2.3^{\pm 1.2}$	$4.2^{\pm 0.7}$	3.2
bruit (soustraction des harmoniques de synthèse)	$5.1^{\pm 1.8}$	$2.8^{\pm 0.6}$	3.9
harmoniques (soustraction du bruit de synthèse)	$3.6^{\pm 1.5}$	$3.1^{\pm 0.6}$	3.4
résidu (soustraction du bruit et des harmoniques de synthèse)	$4.2^{\pm 1.6}$	$3.6^{\pm 0.7}$	3.9

TAB. 3.3 – *Taux d'erreur lorsque l'on utilise comme pré-traitement de la parole les résidus de la soustraction des harmoniques ou du bruit du modèle H+N.*

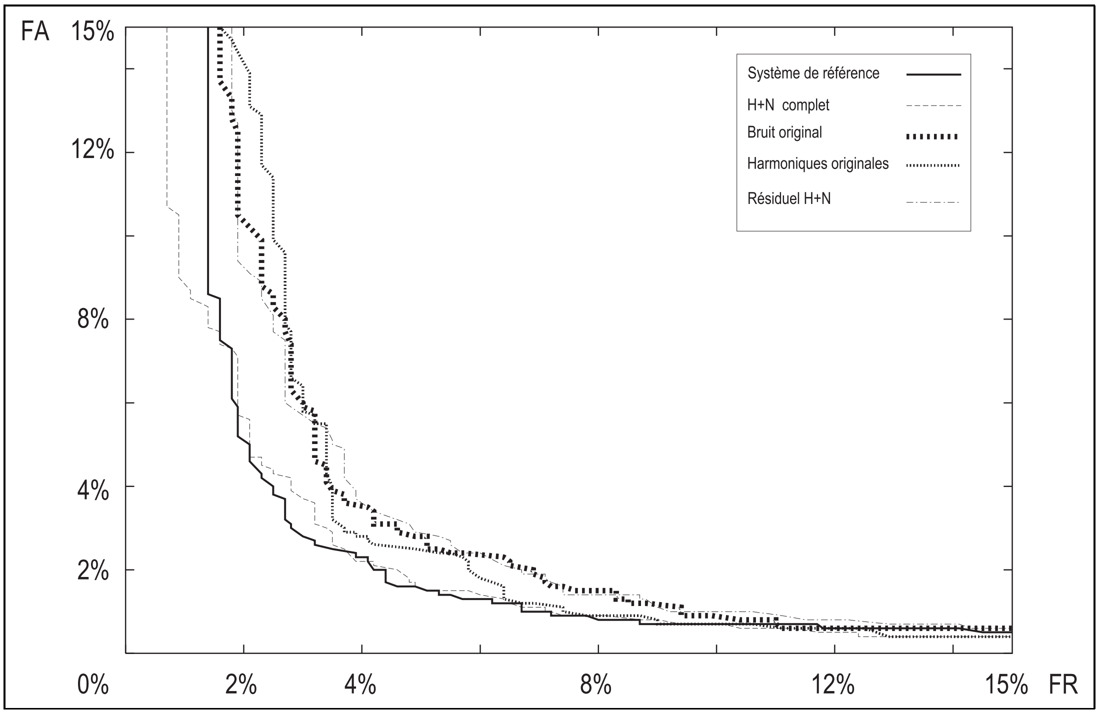

FIG. 3.4 – *Courbes COR des performances des systèmes utilisant comme filtrage d'entrée le bruit ou les harmoniques résiduels du modèle H+N.*

encore distinguer certains aspects de la parole.
- Ces résultats sont vraiment à rapprocher de ceux du résidu de la LPC [Thévenaz, 1993] (voir section 1.2.2), car finalement le modèle H+N n'est qu'une version plus sophistiquée de la LPC, le modèle sous-jacent restant fondamentalement le même. Cela ouvre des perspectives en codage de la parole (transmission des coefficients des filtres, de la F_0, des harmoniques et amplitudes), ainsi que dans le stockage des données de locuteur pour les systèmes de vérification du locuteur, car on ne doit plus stocker que les résidus.
- La résistance au bruit des différents composants (bruits, harmoniques, résidu) doit cependant encore être testée.
- L'évolution temporelle de ces paramètres doit être analysée. En effet, on sait que certains paramètres évoluent avec le temps et que ces variabilités sont différentes d'un locuteur à l'autre.

Maintenant que nous avons décomposé la parole d'un locuteur en partie bruit et en partie harmonique, que ces deux éléments sont exprimables en paramètres modifiables, nous allons tenter d'influer sur ceux-ci pour voir s'ils permettent de changer l'individualité de la voix d'un locuteur.

3.4 Transformation de locuteurs

Comme nous l'avons vu dans la section précédente, la parole humaine est fortement redondante et les informations caractéristiques du locuteur sont réparties sur de nombreux coefficients. Dans cette section, nous allons modifier les paramètres issus de l'analyse de la voix d'un locuteur pour mieux comprendre quels sont ceux qui influent sur l'identité de la voix d'une personne, mais aussi afin d'utiliser cette compréhension pour tenter de modifier ces caractéristiques et analyser les phénomènes d'imposture par imitation spectrale.

3.4.1 Phénomènes d'imposture

Les phénomènes d'imposture en vérification d'identité par la parole ont été relativement peu étudiés en profondeur (voir cependant [Furui, 1994, Schalkwyk *et al.*, 1994b]). Quelques résultats ont été obtenus en utilisant des imitateurs (voir [Homayounpour, 1995] ou [Homayounpour *et al.*, 1994]), mais, la plupart du temps, les imposteurs sont simplement des locuteurs qui, au mieux, prononcent les mêmes paroles que le client sans essayer de l'imiter (voir aussi [Schalkwyk *et al.*, 1994a]). Le manque de bases de données contenant ce genre d'enregistrement en est une des causes principales. Nous allons nous limiter ici à tenter des impostures utilisant les moyens informatiques. Dans ce cadre, si nous voulons vraiment imiter la voix d'un locuteur, plusieurs possibilités s'offrent à nous, dépendant d'un certain nombre de contraintes:

Contenu

La première contrainte pour imiter un locuteur est le contenu des données de parole à disposition. En effet, si par exemple, nous possédons un échantillon au moins de chaque diphone (section 1.1) du locuteur à imiter, nous pouvons simplement utiliser des algorithmes de synthèse à partir du texte comme par exemple MBROLA [Dutoit, 1997]. Si nous ne possédons que des phonèmes, la synthèse à partir du texte devient plus délicate, mais, nous le verrons par la suite, encore relativement efficace pour confondre les systèmes automatiques. Si enfin nous ne possédons que quelques mots, il faut compenser le manque de données par une transformation de la parole de l'imposteur en celle du client.

Quantité

La quantité de données du client disponible est également importante car elle décide du nombre d'éléments qui nous permettront par exemple de concaténer de la parole dans divers contextes (co-articulations, énergie et qualité du signal). Elle détermine aussi les algorithmes de transformation utilisables, la quantité de données étant directement reliée au nombre de paramètres du locuteur extractibles du signal de parole.

Répartition temporelle

L'évolution de la voix de certains locuteurs durant le temps pouvant être importante, une répartition des données sur une durée plus ou moins longue peut nous permettre d'en capter les variabilités.

Dans les expériences qui suivent, nous allons démontrer qu'il est possible de créer des impostures gênantes pour les systèmes de vérification actuels. Les premiers tests que nous présentons ici utilisent la concaténation de phonèmes, alors que les seconds se basent sur l'hypothèse que nous n'avons qu'une seule répétition du code personnel (voir annexe A.1) du client que nous utiliserons pour rapprocher la voix de l'imposteur de chaque mot du code personnel du client.

3.4.2 Imposture par concaténation

Pour notre expérience, nous utilisons un sous-ensemble de locuteurs de Polyvar (annexe B.3). Nous utilisons 17 mots de commande, que nous assimilons à des mots de passe que chaque client pourrait avoir choisi pour protéger l'entrée d'un service téléphonique sécurisé. Pour entraîner les modèles de chaque client, nous utilisons 5 répétitions de chaque mot de commande (les mots sont enregistrés lors de sessions différentes). Le modèle de monde est estimé sur des locuteurs différents que les clients ou les imposteurs.

Pour les tests, nous utilisons environ 70 répétitions de chaque mot de commande, pour les tests client, et $\simeq$ 100 tests d'imposture (voir l'annexe B.3 pour le détail de la répartition de ces données).

Le système de référence que nous utilisons est celui du Projet Européen Picasso (voir la section D.1). Il est basé sur une technique HMM identique à notre système de référence. Il utilise comme paramétrisation 16 coefficients LPCC, l'énergie, ainsi que leurs dérivées et accélérations. Une soustraction cepstrale est aussi effectuée. Les modèles sont constitués de 2 états par phonème et d'une gaussienne par état.

Nous calculons ensuite les performances du système avec un seuil EER *a posteriori*. Le tableau 3.4 indique les performances obtenues pour chacun des mots sur tous les clients.

Mots	EER%	Mots	EER%
annulation	$1.82{\pm}0.80$	Louis Moret	$3.15{\pm}1.04$
Casino	$3.58{\pm}1.11$	manifestation	$1.61{\pm}0.76$
cinéma	$2.81{\pm}0.99$	message	$4.62{\pm}1.25$
concert	$4.95{\pm}1.29$	mode d'emploi	$1.11{\pm}0.63$
Corso	$6.66{\pm}1.48$	musée	$6.40{\pm}1.47$
exposition	$1.82{\pm}0.80$	précédent	$3.64{\pm}1.14$
galerie du Manoir	$1.20{\pm}0.65$	quitter	$7.61{\pm}1.58$
Gianadda	$3.94{\pm}1.16$	suivant	$4.76{\pm}1.25$
guide	$5.04{\pm}1.31$	**TOTAL**	3.81

TAB. 3.4 – *Taux d'erreur à l'EER estimé a posteriori sur des données Polyvar en utilisant le système de référence Picasso.*

Imposture

Pour notre imposture par concaténation, nous avons utilisé entre 300 et 1300 phrases pour chaque client. La première étape consiste à effectuer une reconnaissance de la parole des ces phrases avec une segmentation temporelle au niveau phonétique. Les modèles de phonèmes utilisés ont été entraînés sur une base de données différente (Polyphone, annexe B.4).

Une fois en possession des phrases segmentées phonétiquement, nous recherchons dans celles-ci les séquences de phonèmes qui minimisent le nombre de segments nécessaires à reconstituer un des mots de commande prononcé par le client. Lorsque nous avons trouvé une de ces séquences, nous supprimons les phrases les ayant contenues et nous recommençons l'opération de manière à obtenir 5 répétitions de chaque mot de commande pour chaque client. Normalement, le nombre de segments phonétiques augmente à chaque itération (très dépendant du contenu phonétique des mots de commande), augmentant aussi le nombre de concaténations nécessaires pour former un mot et donc diminuant la qualité de l'imposture.

Le tableau 3.5 donne les taux de fausses acceptations lorsque l'on injecte dans le système les concaténations. Le seuil de décision (*a priori*) utilisé est celui déterminé à l'EER pour calculer les performances **a posteriori** proposées dans le tableau 3.4. Quoique très simples, ces impostures posent déjà un problème sérieux au système de référence et une inspection plus attentive des segments concaténés ainsi qu'une segmentation plus robuste de ceux-ci doit pouvoir augmenter encore les taux de fausses acceptations. Les locuteurs choisis dans la table 3.5 sont ceux pour lesquels aucun problème n'a été détecté durant la reconnaissance automatique des phonèmes.

3.4.3 Transformation des paramètres

Transformer la voix d'un locuteur en la voix d'un autre consiste à identifier les paramètres qui caractérisent le *locuteur source*, à les transformer puis à les utiliser pour restituer finalement un signal de parole ressemblant au *locuteur cible*. La figure 3.5 montre le processus de transformation de locuteurs tel qu'utilisé ici. Comme nous voulons effectuer ces opérations de manière automatique, il est nécessaire de trouver une loi de transformation des paramètres de la source en

Mots	Nombre de locuteurs	Nombre de séquences	FA%
annulation	13	69	50.72
Casino	13	70	24.29
cinéma	15	71	22.54
concert	15	72	18.06
Corso	13	70	32.86
exposition	14	69	43.48
galerie du Manoir	11	33	15.15
Gianadda	12	45	6.67
guide	8	22	40.91
Louis Moret	13	59	18.64
manifestation	14	69	59.42
message	13	70	15.71
mode d'emploi	14	62	20.97
musée	14	71	60.56
précédent	13	70	45.71
quitter	15	72	34.72
suivant	15	72	47.22
TOTAL		1066	33.86

TAB. 3.5 – *Taux de fausses acceptations lorsqu'on utilise des séquences de phonèmes concaténées.*

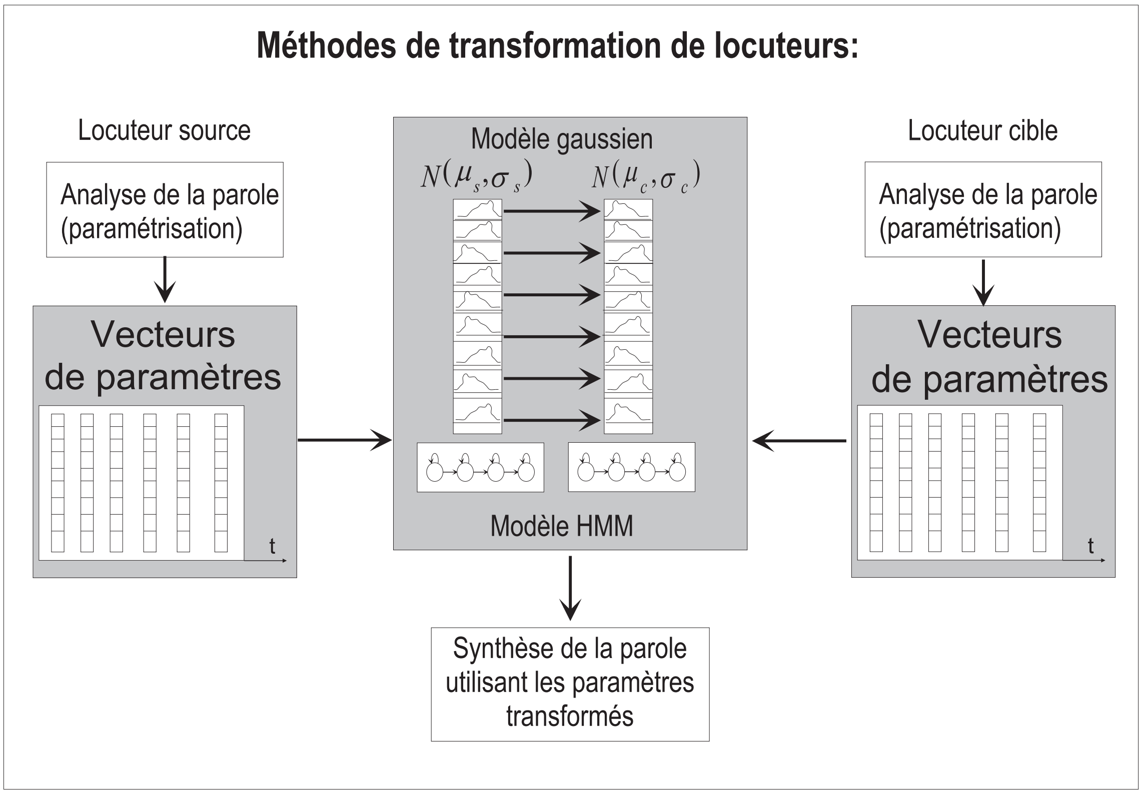

FIG. 3.5 – *Schéma bloc du processus de transformation de locuteurs.*

paramètres de la cible. Cette loi sera déduite des données, en tenant compte des particularités du signal de parole (variations temporelles, d'intensité, etc. . .).

Pour transformer un jeu de paramètres source en paramètres cible, tous les algorithmes de *"pattern matching"* [Duda et Hart, 1973, Ripley, 1996] tels les transformations linéaires, les réseaux de neurones ou la quantification vectorielle peuvent s'appliquer. Cependant les contraintes de quantité de données nous imposent de choisir des algorithmes nécessitant peu de paramètres à estimer. C'est pourquoi, nous supposons que sur un segment de parole donné, les paramètres utilisés se répartissent autour d'une moyenne en suivant une distribution gaussienne.

Transformation gaussienne

Tout d'abord, nous considérons que les coefficients cepstraux ainsi que le logarithme des rapports d'aire (LAR = log area ratio = $\frac{1-k_i}{1+k_i}$) suivent une distribution gaussienne sur la durée d'un mot ou d'un phonème. Cela signifie que chaque paramètre $c_{(i,source)}$ du locuteur source suit une distribution gaussienne $N(\mu_{(i,source)}, \sigma_{(i,source)})$ et chaque paramètre de la cible $c_{(i,cible)}$ suit une distribution gaussienne $N(\mu_{(i,cible)}, \sigma_{(i,cible)})$. Dans ce cas, la transformation des paramètres de la source vers la cible s'effectue selon l'équation 3.1, avec $\hat{c}_{(i,transfo)}$ le coefficient i transformé.

$$\hat{c}_{(i,transfo)} = (\sigma_{(i,cible)}/\sigma_{(i,source)})(c_{(i,source)} - \mu_{(i,source)}) + \mu_{(i,cible)} \tag{3.1}$$

Contrôle de la normalité des distributions

Afin de vérifier si les c_i, le k_i et les LAR suivent vraiment une distribution gaussienne, nous avons testé la normalité de la distribution des paramètres. Cette opération peut être effectuée en utilisant le test d'ajustement de Cramer/von Mises (voir [Saporta, 1990], p338). La statistique $N\omega_N^2$ (équation 3.2) qui est une mesure de l'écart existant entre une répartition théorique $\mathcal{F}(x)$ et une répartition empirique $\mathcal{F}_n^*(x)$ elle permet de tester l'hypothèse $H_0 : \mathcal{F}(x) = \mathcal{F}_0(x)$ contre $H_1 : \mathcal{F}(x) \neq \mathcal{F}_0(x)$. La distribution de $N\omega_N^2$ été tabulée (voir Saporta [1990]).

$$N\omega_N^2 = \int_{-\infty}^{+\infty} \left[\mathcal{F}_n^*(x) - \mathcal{F}(x)\right]^2 d\mathcal{F}(x) \tag{3.2}$$

On démontre que:

$$N\omega_N^2 = \frac{1}{12N} + \sum_{i=1}^{N} \left[\frac{2i-1}{2N} - \mathcal{F}(x_i)\right]^2$$

Les x_i sont les valeurs **ordonnées** des échantillons $(x_1 < x_2 < \cdots < x_N)$ et N le nombre d'échantillons.

Avec:

$$T_0 = \frac{1}{12N} + \sum_{i=1}^{N} \left[\frac{2i-1}{2N} - \mathcal{F}_0(x_i)\right]^2$$

On rejette H_0 si T_0 est supérieur à une valeur que la variable aléatoire $N\omega_N^2$ à une probabilité α de dépasser. (équation 3.3).
Au seuil $\alpha = 0.01$ on rejette l'hypothèse que la loi empirique soit la même que la loi théorique si $N\omega_N^2 > 0.178$ (valeur tabulée). Bien que la loi statistique de $N\omega_N^2$ ne soit pas connue, on peut utiliser une simulation empirique pour tester la normalité d'une distribution:
Si nous estimons la moyenne μ et l'écart-type σ:

$$\mu = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \sigma = \sqrt{\frac{1}{N-1}\sum (x_i - \overline{x})^2}$$

Avec:

$$T_g = (1 + 0.5/N)N\omega_N^2 > 0.178 \tag{3.3}$$

On rejette l'hypothèse H_0 ="la loi est normale" si $T_g > 0.178$ pour $\alpha = 0.01$.
Le tableau 3.6 nous donne les résultats du test de normalité pour les paramètres considérés.

Longueur du segment	k_i	$\frac{1-k_i}{1+k_i}$	c_i
Test sur un mot	$\sim 4\%$	$\sim 1\%$	$\sim 2\%$
Test sur les états HMM d'un mot	$\sim 99.5\%$	$\sim 99.5\%$	$\sim 99.5\%$

TAB. 3.6 – *Taux de gaussianité (en %) des paramètres selon la longueur des segments de parole.*

Transformation par HMM

L'hypothèse qui consiste à supposer que la distribution des paramètres est gaussienne sur toute la durée d'un mot s'avère fausse la plupart du temps. Cependant, pour une durée plus courte cette hypothèse devient vraie (voir le tableau 3.6). Une approche sûre consiste à utiliser une segmentation effectuée par l'occupation des états d'un modèle de Markov caché (HMM) (voir l'annexe A et les travaux de [Tokuda *et al.*, 1995, Masuko *et al.*, 1997]). Un système de reconnaissance de la parole basé sur un modèle HMM est utilisé pour aligner le signal de parole de la source et de la cible. On considère alors que chaque état du modèle HMM suit une distribution gaussienne et une transformation des paramètres est effectuée état par état en utilisant l'équation 3.1.

3.4.4 Re-synthèse des paramètres transformés

Les paramètres $\hat{c}_{(i,transfo)}$ sont ensuite injectés dans le système de synthèse H+N (voir la section 1.3.3). La re-synthèse utilisant le bruit ne donne pas des résultats satisfaisants. En effet, lorsque l'on transforme la partie bruit d'un locuteur source en celle d'un locuteur cible par transformation des coefficients de réflexion k_i ou les rapports d'aire LAR, les résultats montrent que la parole transformée éloigne la source de la cible au lieu de l'en rapprocher. Par contre, l'ajout d'un bruit tel qu'un bruit aléatoire rapproche la source de la cible ou tout au moins ne l'éloigne pas.

Ajout du bruit de la source

Dans cette expérience, la parole du locuteur source est analysée par H+N et seule la partie harmonique est ensuite re-synthétisée. Dans une seconde étape, on soustrait la partie harmonique analysée re-synthétisée du signal original de la source et on ne garde que la partie bruit. Cette partie bruit est ensuite ajoutée au signal harmonique transformé.

Ajout d'un bruit de fond

Dans ce cas, on ne re-synthétise que la partie harmonique à partir des paramètres transformés, puis on rajoute un bruit de fond aléatoire. Ce bruit de fond est important (comme nous l'avions déjà constaté dans la section 3.3), pour conserver le caractère de variable aléatoire des vecteurs de paramètres lorsqu'on considère le calcul des vraisemblances dans le modèle HMM du système de référence.

3.4.5 Expériences de transformation

Le système de référence HMM (annexe A) est entraîné avec les données d'entraînement de Polycode (annexe B.1). Un seuil *a priori* est ensuite déterminé avec les données d'évaluation de Polycode. Les données de test de Polycode sont ensuite utilisées de la manière suivante:

1. Les données d'imposture sont fournies au système de référence et une décision est prise en comparant le logarithme du rapport de vraisemblance de chaque phrase au seuil *a priori* dépendant du locuteur. Le taux de fausse acceptation (FA) est ensuite calculé comme étant le pourcentage de phrases d'imposteurs faussement acceptées comme codes personnels du client.

2. Les données d'imposture sont transformées en utilisant les systèmes de transformation de locuteur de la section 3.4. Ces phrases transformées sont ensuite fournies au système de référence et un nouveau taux de fausses acceptations est calculé avec le même seuil de décision qu'au point 1.

Méthode	FA% source	FA% aléatoire
Système de référence parole non transformée	$4.2^{\pm 0.7}$	$4.2^{\pm 0.7}$
Transformation gaussienne sur la durée d'un mot	$3.6^{\pm 0.6}$	$14.5^{\pm 1.3}$
Transformation par état HMM	$4.7^{\pm 0.8}$	$23.1^{\pm 1.5}$

TAB. 3.7 – *Taux de fausses acceptations avec un seuil de décision a priori fixé à l'EER. On utilise des données d'imposture transformées par le système gaussien et le système HMM, avec le bruit de la source ou un bruit aléatoire.*

Le tableau 3.7 donne les résultats des transformations gaussiennes et HMM sur un mot en ajoutant les deux types de bruits décrits dans la section 3.4.4. La figure 3.6 montre les courbes COR obtenues pour les transformations gaussiennes et HMM avec un bruit aléatoire.

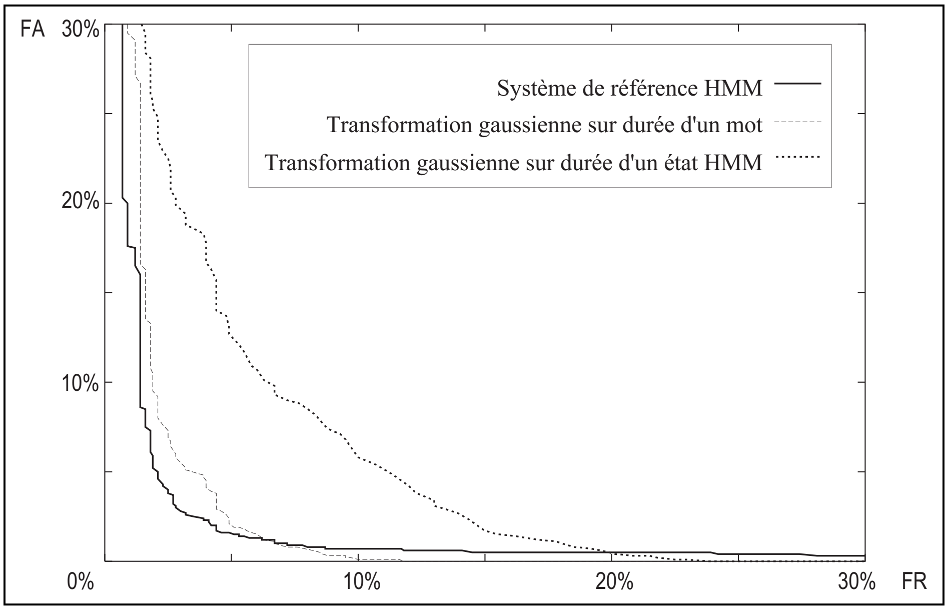

FIG. 3.6 – *Courbes COR indiquant les différentes tentatives de transformation de locuteurs.*

Les résultats de ces différentes transformations de la voix nous permettent de faire un certain nombre de constatations:

– Les harmoniques seules contiennent une information dépendante du locuteur qu'il est possible de modifier en jouant sur les coefficients cepstraux.

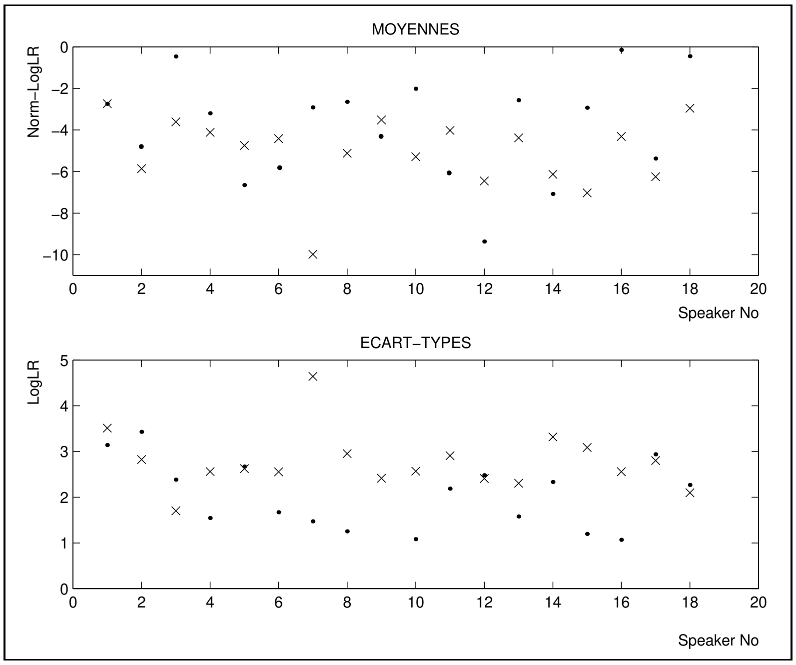

FIG. 3.7 – *Moyennes normalisées et variances des scores LLR pour les tests d'imposture sur chaque locuteur après transformation des harmoniques par HMM (×) comparés au système de référence (•).*

- La partie bruit contiennent une information dépendante du locuteur mais nous ne sommes pas capables de la transformer de façon valable.
- La figure 3.7 indique les différences de qualité de la transformation selon les locuteurs. On peut constater que pour la plupart des locuteurs, c'est la variance des scores des séquences transformées qui s'est le plus accrue. Cela indique que nos transformations augmentent la variabilité des séquences d'imposteurs.

Synthèse	Mixte (H+F) (18)	Hommes (15)	Femmes (5)
Transfo. harmoniques seules du H+N +(bruit aléatoire)	23.09 ± 1.5	24.22 ± 2.2	35.00 ± 6.9

TAB. 3.8 – *Taux de fausses acceptations selon le sexe pour une transformation HMM.*

- Comme les transformations que l'on effectue sont basées sur la fréquence fondamentale, il y a un risque, lorsque l'on transforme un homme en femme ou vice versa, de décalage des harmoniques. Nous avons donc conçu une expérience où l'on effectue uniquement des transformations hommes/hommes et femmes/femmes (tableau 3.8) pour une transformation HMM; l'amélioration de la transformation est assez nette pour les femmes.

3.5 Résistance à l'imposture

Comme les deux sections précédentes l'ont montré, il est possible de transformer la voix d'un locuteur en celle d'un autre. Il serait cependant utile de connaître les performances d'une application de vérification de locuteurs, lorsqu'on supprime les parties du signal de parole qui sont transformables. En effet, comme la section 3.3 l'a montré, il est possible de supprimer de l'information du signal de parole sans modification des performances du système de référence. Comme les transformations que nous avons effectuées dans la section 3.4 portent essentiellement sur la modification des harmoniques, si nous les éliminons du signal original, la résistance aux imposteurs qui tenteraient d'utiliser de telles méthodes devrait augmenter.

Méthode	FAR%
Référence sans transformations	4.2 ± 0.7
Référence avec imposteurs transformés	23.1 ± 1.5
Soustraction des harmoniques	8.17 ± 1.02
Résidu H+N	8.42 ± 1.04
Résidu LPC	15.65 ± 1.36

TAB. 3.9 – *Taux de fausses acceptations avec des imposteurs transformés.*

Le tableau 3.9 et la figure 3.8 indiquent quelles sont les fausses acceptations obtenues avec des imposteurs transformés, lorsqu'on garde du signal soit la partie bruit originale (soustraction des harmoniques du modèle H+N), soit le résidu de la soustraction du modèle H+N complet, soit, à titre de comparaison, le résidu de la LPC (voir [Thévenaz, 1993]). Le fait de ne garder que

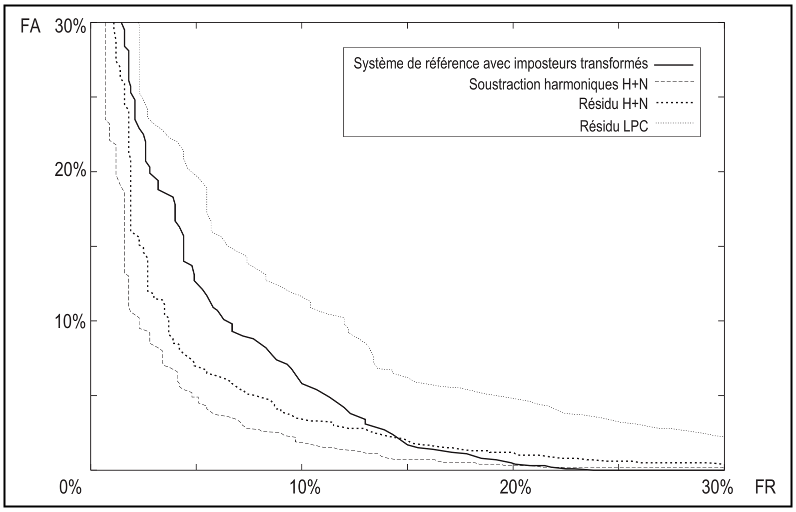

FIG. 3.8 – *Courbe COR des performances du système de référence avec le signal original, avec soustraction des harmoniques H+N, avec résidu H+N ou avec résidu LPC.*

le bruit du signal d'origine ou le résidu de la soustraction de modèle H+N permet effectivement de résister aux transformations spectrales telles que nous les avons conçues à la section 3.4. De plus, nous avons la confirmation que la partie bruit contient effectivement une partie importante de l'information discriminante entre locuteurs.

3.6 Conclusion sur les transformations de locuteur

Nous avons pu le constater dans ce chapitre, l'identité d'un locuteur vue par un système automatique peut être modifiée assez fortement par une transformation de l'enveloppe spectrale. Nous nous y sommes partiellement rendus résistants en soustrayant les harmoniques de la F_0. Poussée à son extrême, la soustraction de parties du signal nous a permis de constater une fois encore la redondance du signal de parole. Il faut cependant remarquer qu'à l'écoute, le résiduel de la LPC est beaucoup plus compréhensible pour un humain que le résidu de la soustraction H+N. L'information caractéristique est contenue dans les parties harmonique et bruit du signal de parole. Cependant, lorsque on soustrait du signal les harmoniques resynthétisées par le modèle H+N, le résiduel est encore suffisamment caractéristique du client pour que les performances de la reconnaissance du locuteur ne se dégradent pas. Le résiduel est constitué de la partie bruit, qui contient de l'information dépendante du locuteur, mais aussi quelques résidus harmoniques non modélisés par le H+N.

Chapitre 4

Reconnaissance de locuteurs

4.1 Introduction

Le chapitre 3 a montré quels sont les paramètres discriminant les locuteurs. Dans ce chapitre, nous allons nous intéresser à modéliser ces paramètres de manière à mémoriser et modéliser les caractéristiques du locuteur considéré.

La reconnaissance automatique du locuteur peut être divisée en *identification* ou *vérification* du locuteur selon que l'on suppose connue ou non la provenance de l'échantillon de parole que l'on désire reconnaître. Si l'on tient compte du texte prononcé, nous parlerons de *mode dépendant du texte*, si l'on ne tient pas compte du contenu phonétique, nous parlerons de *mode indépendant du texte* (voir la section 1.2 pour plus de détails sur ces différents modes de reconnaissance). Nous nous intéresserons ici au système de reconnaissance en mode **vérification du locuteur dépendante du texte**. Plusieurs raisons motivent ce choix:

– Le mode de vérification est plus pratique à utiliser lors d'applications à grande échelle car il ne nécessite pas de connaître les autres clients de l'application, ni d'opérer des tests sur tous les clients enregistrés. Il peut donc être considéré comme indépendant du nombre de clients inscrits dans une application.
– Ce mode est plus à même de modéliser les phénomènes d'imposture que le mode d'identification. En effet, en identification dans un ensemble fermé, on cherche à déterminer le client de l'application qui a la probabilité la plus grande d'avoir prononcé la phrase observée. Cependant, dès l'instant où les imposteurs ne se trouvent pas dans la base des clients, il faut passer en identification dans un ensemble ouvert, combinant ainsi les phénomènes d'imposture et d'identification.
– Les résultats de modélisation obtenus en mode vérification sont utilisables en identification.
– La dépendance au texte permet de mieux contrôler le message prononcé, vu le mélange complexe entre les informations linguistiques et celles dépendantes du locuteur contenues dans le signal de parole.
– La dépendance au texte permet une modélisation plus précise du contenu linguistique et donc de meilleures performances du système de reconnaissance du locuteur.

Quelques contraintes supplémentaires issues des problèmes pratiques rencontrés dans les différents systèmes dont nous avons participé à la mise au point (voir le chapitre 2 et la section D.1) sont à relever:

- Peu de données d'entraînement sont à disposition (généralement une ou 2 sessions de quelques secondes de parole).
- Le vocabulaire utilisé est limité et la grammaire sous-jacente également.

Le manque de données d'entraînement limite fortement la possibilité d'accroître la précision des modèles en augmentant le nombre de paramètres. En effet, la modélisation HMM (que nous utilisons comme système de référence, voir l'annexe A) donne de bons résultats lorsque suffisamment de données sont à disposition pour estimer les paramètres, mais les performances se dégradent rapidement avec la diminution de la quantité de données. Nous avons donc recherché d'autres algorithmes que les HMM nécessitant moins de données d'entraînement, mais tirant parti de toute l'information disponible sur le locuteur en exploitant les connaissances *a priori* que nous pouvons avoir sur le texte prononcé.

L'approche statistique du système de référence, qui nous permet de calculer la vraisemblance qu'une observation donnée appartient à un client plutôt qu'au *"monde"* (voir annexe 1.2.4) peut être exprimé d'une autre manière. En effet, plutôt que d'établir un modèle pour le client et un modèle pour le *monde*, c'est-à-dire tous les locuteurs sauf lui, il est possible de décomposer cette approche en problèmes à deux classes, dite aussi *décomposition binaire*. Cette méthode tente de séparer les données du client de celles d'un autre locuteur seulement (voir également les modèles de cohorte dans la section 1.2). Les locuteurs choisis comme anti-classe de la classe du client (appelés ici **anti-clients**) seront déterminés aléatoirement (voir la section 4.4). Rappelons qu'un client est un locuteur dont on possède un modèle, qui est autorisé à utiliser le système de vérification et dont on cherche à vérifier l'identité étant donnée une observation (une séquence de parole). La décomposition binaire divise la tâche de classification en opérations plus simples mais plus nombreuses et partage ainsi la complexité de la classification entre les classificateurs et la recombinaison des différentes décisions binaires. Les classificateurs binaires ont déjà été utilisés en vérification indépendante du texte par Castellano [1997].

Dans une application de reconnaissance de locuteur en mode dépendant du texte, le contrôle du texte est effectué normalement par un système de reconnaissance de la parole qui segmente temporellement la phrase prononcée et permet ainsi la comparaison du mot reconnu avec un modèle du client préalablement entraîné (voir l'annexe A.2). Cependant, dans la plupart des systèmes actuels, les scores obtenus pour chaque mot ou sous-mot de la phrase sont simplement combinés, sans tenir compte des connaissances *a priori* que l'on pourrait dégager pour chaque client, comme par exemple le fait que la prononciation de certains mots caractérise mieux un locuteur que d'autres. Nous allons décrire dans ce chapitre comment extraire et exploiter cette information. Chacun des classificateurs binaires séparera les données d'un seul mot pour un couple *client/anti-client* (section 4.2). Les sorties de chaque classificateur sont fusionnées en un seul score (voir la section 4.4) qui est ensuite comparé à un seuil de décision (*a priori* ou *a posteriori*) (section 4.5) le système utilise la même logique de décision que le système de référence.

4.2 Modélisation par matrice binaire

Pour chaque client C de la base de données Polycost (voir l'annexe B.2), une matrice D^C de classificateurs, de taille $M \times N$, est construite (voir figure 4.1). Chaque ligne de la matrice est associée à un chiffre du code personnel du client ($M = 10$), et chaque colonne à un anti-client ($N = 12$ dans la configuration choisie ici). Pour chaque client, un nombre total de 120 classificateurs est donc construit.

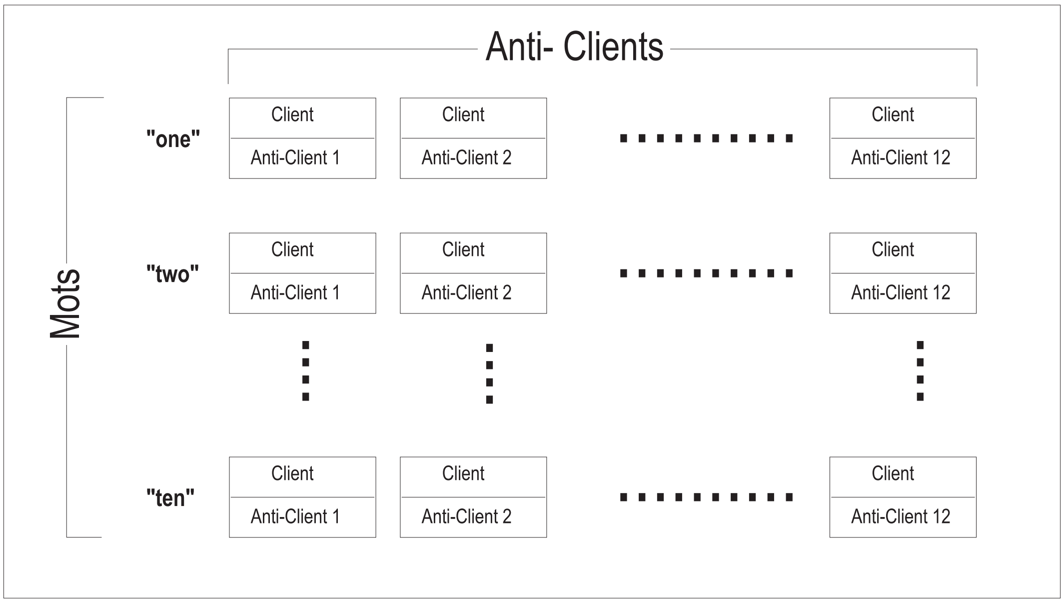

FIG. 4.1 – *La matrice D^C des classificateurs pour un client C.*

Chaque classificateur binaire est entraîné séparément en utilisant les répétitions de chaque mot M_i du client et d'un anti-client N_j. Nous avons utilisé ici comme classificateurs binaires des **arbres de décision** (voir la section 4.3).

Lors du test d'une séquence inconnue X, chaque classificateur de D^C retourne une valeur dans l'intervalle $[0, 1]$ exprimant avec quelle confiance il estime que X a pu être prononcée par le client c. Les sorties des $M \times N$ classificateurs sont ensuite fusionnées par combinaison linéaire et comparées à un seuil t selon l'équation 4.1.

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \omega_{ij} D^c_{ij}(X) \ \geq \ t. \tag{4.1}$$

Le choix des poids ω_{ij} et du seuil t sont discutés dans la section 4.4.

4.3 Arbres de décision

Parmi les algorithmes classiques utilisés en *apprentissage automatique* pour la résolution de problèmes de classification binaire, citons les séparateurs linéaires [Duda et Hart, 1973], les perceptrons multi-couches (MLP) [Ripley, 1996], les algorithmes génétiques [Mitchell, 1997], les Support Vector Machine (SVM) [Vapnik, 1995, Burges, 1997] et les arbres de décision.

Chacun de ces algorithmes a des propriétés intéressantes que nous ne détaillerons pas ici car les contraintes qui nous sont imposées en reconnaissance du locuteur (quantité de données, variabilité des données non présentes dans les données d'entraînement, etc..., voir la section 4.1) nous ont amenés à sélectionner les arbres de décision, plus précisément le C4.5 [Mitchell, 1997], développé par Quinlan [1993]. Le C4.5 est une version des arbres de décision qui permet d'utiliser des entrées avec des attributs à valeur continue. En dépit de sa relative simplicité, cet algorithme présente quelques qualités intéressantes en plus de celles déjà exigées: il demande très peu de temps d'entraînement et la taille d'un modèle entraîné est relativement petite (500 à 1000 bytes), deux propriétés importantes pour les applications à grande échelle où de la taille mémoire et la consommation en temps de calcul sont cruciales.

Pour notre application, **chaque** classificateur de la matrice D^c est entraîné en utilisant comme entrée les vecteurs de paramètres du client (13 LPCC, l'énergie et leurs dérivées) et un des N_j anti-clients pour chacun des M_i chiffres. L'algorithme C4.5 sépare les données d'entraînement en deux classes: les données du client (classe 1) et les données de l'anti-client (classe 0). Chacun des 28 éléments du vecteur d'entrée est considéré comme un attribut continu. Nous considérons également que les éléments du vecteur d'entrée sont indépendants, ce qui est normalement le cas pour des vecteurs cepstraux (voir la section 1.2.2).

Le choix des anti-clients est un paramètre essentiel du bon fonctionnement du système en paires binaires. En l'absence d'une approche établie sur le choix des anti-clients et vu sa complexité, nous proposons ici une méthode en deux étapes: tout d'abord, les anti-clients sont choisis d'une manière aléatoire, ce qui permet de construire des classificateurs binaires; ensuite, la sélection d'un sous-ensemble d'anti-clients choisis *a posteriori* permet de conserver ceux qui définissent au mieux l'hypercube du client dans l'espace des paramètres considérés.

4.3.1 Construction d'un arbre de décision

Les arbres de décision sont issus de la théorie de l'information décrite par le théorème de Shannon. On peut le reformuler de la manière suivante: supposons T l'ensemble total de notre problème. Soit S un sous-ensemble quelconque d'éléments de T. Appelons $card(C_i, S)$ le nombre d'éléments de S qui appartiennent à la classe C_i et $|S|$ le nombre d'éléments de S. La probabilité de la classe C_i, sachant un élément de S, peut être estimée par:

$$p(C_i|S) = \frac{card(C_i, S)}{|S|} \tag{4.2}$$

La quantité d'information qui lui est associée est de:

$$q\{(C_i|S)\} = -\log_2(p(C_i|S)) \tag{4.3}$$

La quantité d'information totale de S associée aux classes est de:

$$Q(S) = -\sum_{i=1}^{k} p(C_i|S) \times \log_2(p(C_i|S)) = E(C_i|S) \tag{4.4}$$

Dans notre cas, $Q(T)$ exprime la quantité moyenne d'information nécessaire pour identifier une classe de T, que l'on peut aussi estimer comme la quantité d'information ou **l'entropie de l'ensemble** $S = E(C_i|S)$.

Supposons maintenant que nous avons un ensemble de données d'entraînement E (échantillons) qui peut être divisé en sous-ensembles $E_1 \cdots E_n$ par un test X à n classes. Si nous voulons évaluer ce test sans connaître le détail des T_i partitions, la seule information que nous ayons à disposition est la distribution des données de E. Si nous considérons maintenant la mesure de l'approximation faite lorsque nous avons partitionné T par les n sorties de X, la quantité d'information $Q_{\mathrm{X}}(T)$ peut être estimée par:

$$Q_{\mathrm{X}}(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times Q(T_i). \tag{4.5}$$

Ce qui nous permet d'estimer le gain $G(X)$ obtenu par le test X:

$$G(X) = Q(T) - Q_{\mathrm{X}}(T) \tag{4.6}$$

On peut alors établir un **critère de gain** qui nous permettra de sélectionner le test qui maximise le gain d'information. Celui-ci peut être compris comme *l'information mutuelle* entre le test X et la classification idéale.

Afin de se rendre indépendant du nombre d'éléments par test, on normalise le gain par une quantité qui peut être considérée comme le coût d'une division de l'ensemble T en n sous-ensembles (équation 4.7). Ce gain normalisé est appelé *rapport de gain* $RG(X)$ (équation 4.8).

$$D(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times log_2 \frac{|T_i|}{|T|}. \tag{4.7}$$

$$RG(X) = \frac{G(X)}{D(X)} \tag{4.8}$$

Si nous considérons les données du client et d'un anti-client munies d'un ensemble de classes T, un arbre de décision se crée de la manière suivante: tout d'abord, la racine est associée à un espace contenant toutes les données d'entraînement E. Cet espace est ensuite partitionné en sous-régions récursivement, chaque subdivision de l'espace étant associée au test X d'un attribut (un coefficient du vecteur d'entrée $LPCC$, l'énergie ou leurs dérivées). A chaque noeud de l'arbre C4.5, on sélectionne l'attribut qui donne le meilleur rapport de gain $RG(X)$ et on détermine un seuil de décision. Le processus de sélection continue jusqu'à ce que tous les exemples de l'ensemble d'entraînement soient classés correctement ou que tous les attributs à disposition aient été utilisés. Lorsque l'arbre est complet, chaque feuille représente une classe.

Pour augmenter les performances du classificateur, une étape d'élagage de l'arbre est ensuite effectuée. Cet élagage est basé sur le remplacement de parties de l'arbre par une feuille lorsque le taux d'erreur généré par ce remplacement est inférieur au taux d'erreur de la partie de l'arbre que l'on a remplacée. Pour calculer ces taux d'erreur, on utilise une validation croisée de l'arbre, les données ayant servi à sa construction étant ensuite employées pour l'élagage et inversement. L'algorithme C4.5 est expliqué en détail dans [Quinlan, 1993, Mitchell, 1997].

Lorsqu'on applique un ensemble de données inconnues à un arbre de décision C4.5 construit pour séparer deux classes (0 et 1), la valeur du score de sortie peut être comprise comme une probabilité pour ces données d'appartenir à la classe 1 (celle du locuteur).

4.3.2 Utilisation des arbres de décision

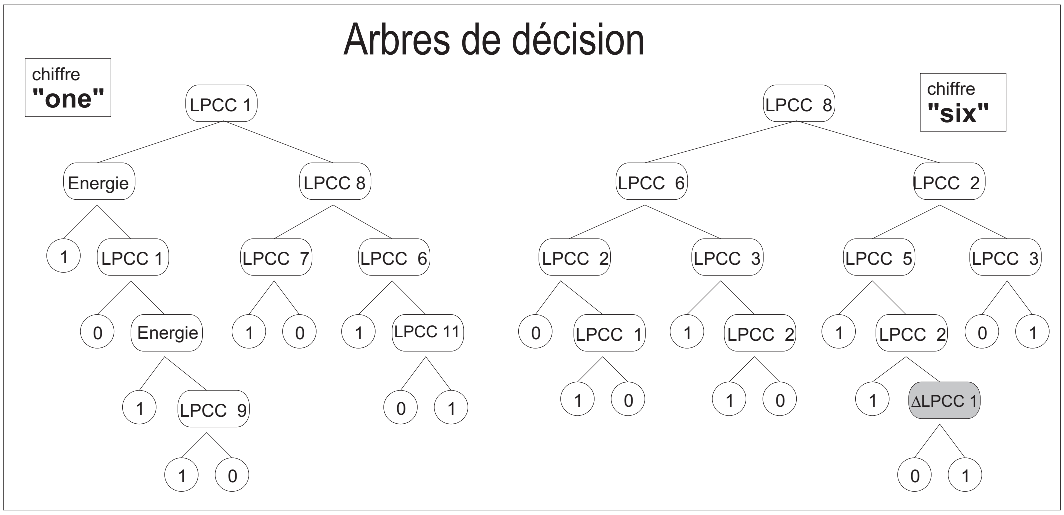

FIG. 4.2 – *Arbres de décision pour les mots "one" et "six" d'un couple client/anti-client.*

La figure 4.2 donne un exemple d'arbres de décision pour les mots "one" et six" d'un client et d'un anti-client de Polycost. De l'observation de ces arbres on peut tirer quelques conclusions:

- Les arbres constitués sont relativement compacts, ce qui signifie que l'information discriminante entre deux locuteurs est répartie sur peu de coefficients du vecteur d'entrée.
- Les coefficients sélectionnés diffèrent d'un mot à l'autre et d'un couple client/anti-client à l'autre (voir la figure 4.2), ce qui permet de valider l'approche de décomposition en classificateurs binaires. En effet, l'information discriminante ne se trouve pas répartie sur tous les coefficients.
- Dans le cas où les données d'entraînement sont suffisamment nombreuses, cette sélection des paramètres pourrait servir de pré-traitement à des modèles HMM, RNA ou hybrides.
- Les transformations de voix effectuées au chapitre 3 pourraient être optimisées en ne transformant que les paramètres qui discriminent au mieux le locuteur cible du locuteur source. Le tableau 4.1 propose les résultats de transformations avec et sans sélection des paramètres.

Méthode	FAR%
Système de référence	$4.2^{\pm 0.7}$
Transformation HMM	$23.1^{\pm 1.5}$
Transformation HMM avec sélection des paramètres	$11.54^{\pm 1.2}$

TAB. 4.1 – *Performances des transformations de locuteur utilisant seulement les paramètres sélectionnés par les arbres de décision comparées à une transformation utilisant tous les coefficients cepstraux.*

Cette solution n'augmente pas les fausses acceptations autant que les transformations de locuteurs utilisant tous les coefficients, ce qui tendrait à démontrer que même si certains coefficients portent moins d'information que d'autres, ils contribuent quand même à la caractérisation de la voix d'un locuteur.

– Si l'on se penche sur le taux d'utilisation des classificateurs pour tous les clients et qu'on observe la répartition des paramètres utilisés, on s'aperçoit que leur distribution est divisée en trois groupes: ceux qui sont fréquemment utilisés, tels les LPCC-1-2-3-4 et l'énergie (plus de $2/3$ des cas), ceux qui ont une fréquence moyenne, soit les coefficients LPCC-5-6-7-8-9-10-11 ($< 2/3, > 1/3$ des cas) et enfin tous les autres coefficients, soit les coefficients LPCC-12-13 et toutes les dérivées. La sélection des coefficients s'effectuant au moyen d'un critère d'entropie, on peut en déduire que, en moyenne, l'information la plus discriminante est répartie dans les premiers coefficients LPCC et l'énergie. La figure 4.3 donne, en plus des valeurs réelles, l'utilisation relative de chaque paramètre en dégradé de gris (en noir, les plus fréquemment utilisés et en blanc les plus rarement utilisés).

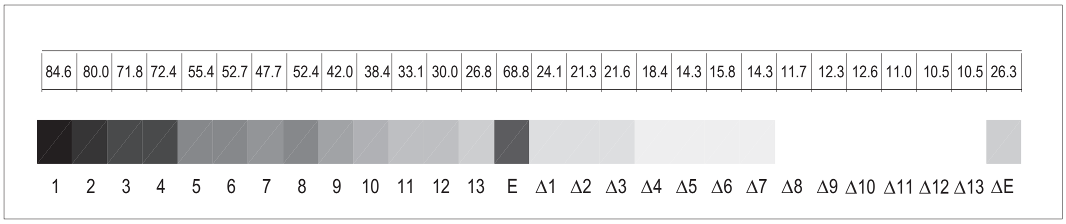

FIG. 4.3 – *Taux d'utilisation (en %) de chaque coefficient par rapport à une utilisation de tous les coefficients lors de tous les tests. La partie du bas représente les valeurs du tableau par des niveaux de gris, en noir les coefficients les plus fréquemment utilisés et en blanc les plus rarement utilisés.*

– Il est possible d'observer la répartition des coefficients selon les anti-clients. La figure 4.4 montre que si la répartition pour chaque locuteur ressemble à la répartition globale de la figure 4.3, il semble que pour les anti-clients femmes ($fnnn$ dans le tableau) la discrimination se fasse sur plus de coefficients.

– Si l'on analyse maintenant la répartition des coefficients par mot (tableau 4.5), l'allure générale de la répartition globale est respectée. On peut constater cependant des variations selon

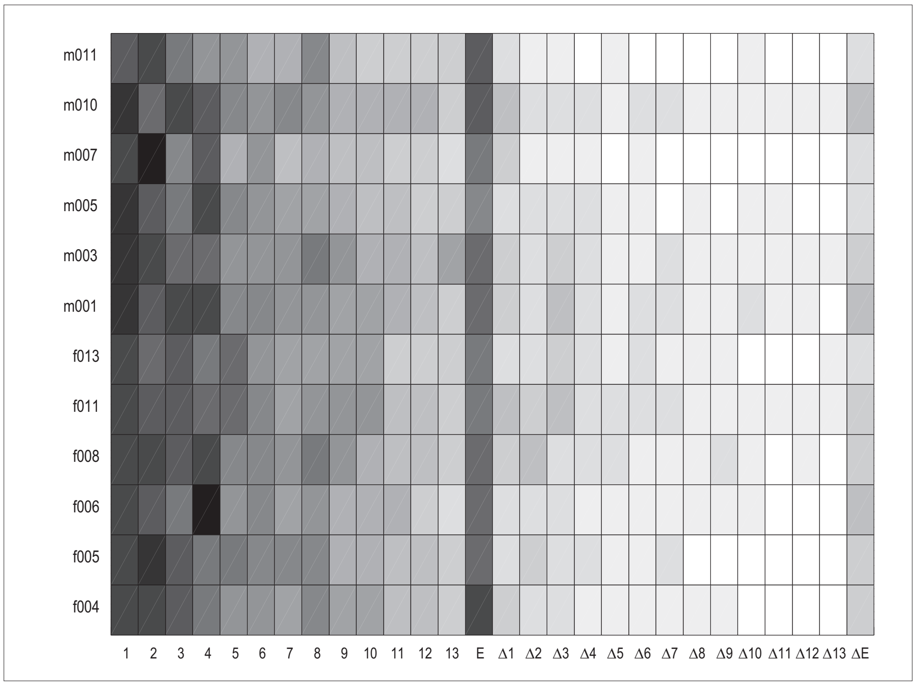

FIG. 4.4 – *Taux d'utilisation des coefficients, répartition selon les anti-locuteurs, même échelle que la figure 4.3, mnnn indique les anti-clients hommes, fnnn les anti-clients femmes.*

les mots, dépendant de leur contenu phonétique.

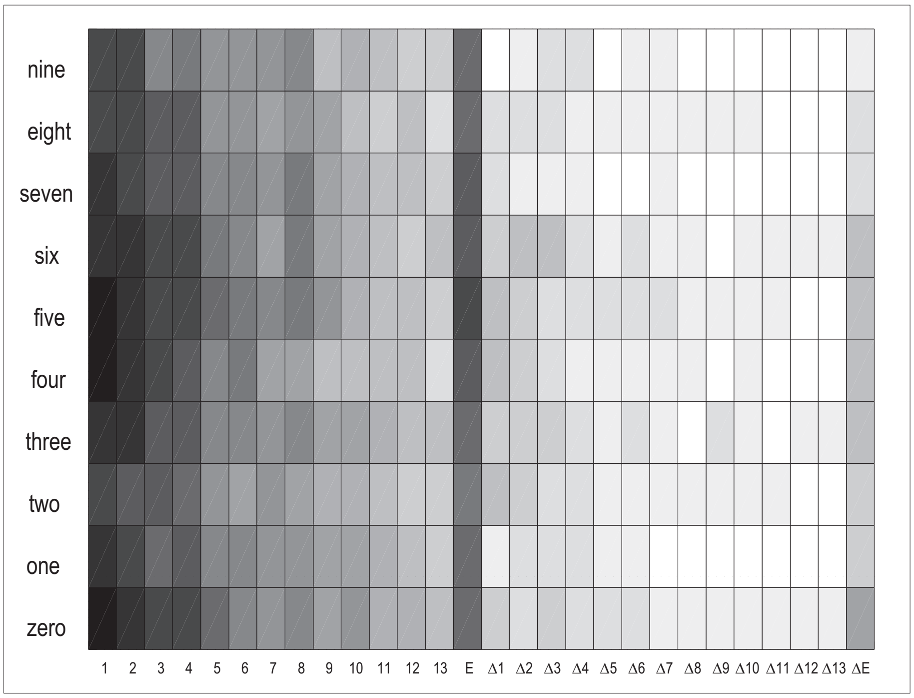

FIG. 4.5 – *Taux d'utilisation des coefficients, répartition selon les mots utilisés dans l'application, même échelle que la figure 4.3.*

- Puisque le C4.5 a la propriété de donner un ordre d'importance aux paramètres choisis (leur position dans l'arbre), nous avons extrait les 5 paramètres les plus sélectionnés pour tous les clients/anti-client/mots (figure 4.6). On peut constater que l'énergie, quoique rarement choisie en premier comme coefficient discriminant, joue un rôle non négligeable dans la séparation des données puisqu'elle se trouve souvent dans les 5 premiers coefficients sélectionnés.
- Pour terminer cette analyse, nous avons représenté sur la figure 4.7 la répartition des coefficients selon chacun des clients de Polycost. Remarquons cependant que les premiers coefficients sont les plus utilisés pour discriminer les paires binaires, mais regardés individuellement, certains locuteurs se comportent de manière différente.

Nous pouvons maintenant valider l'approche de la décomposition en tâches simples qui nous permet de capter les variations d'éléments discriminants selon les mots ou les couples client/anti-client. Les figures précédentes confirment également le fait que les premiers coefficients LPCC, lorsqu'ils sont utilisés, portent l'information la plus discriminante. Le fait que les coefficients d'ordre supérieur et les dérivées ne soient pas sélectionnés aussi souvent que les autres n'enlève cependant rien à leur importance. En effet, ils sont quand même utilisés et servent à la discrimination "fine" entre locuteurs, ce qui est essentiel pour atteindre des taux d'erreurs inférieurs au pourcent (avec un seuil *a posteriori*).

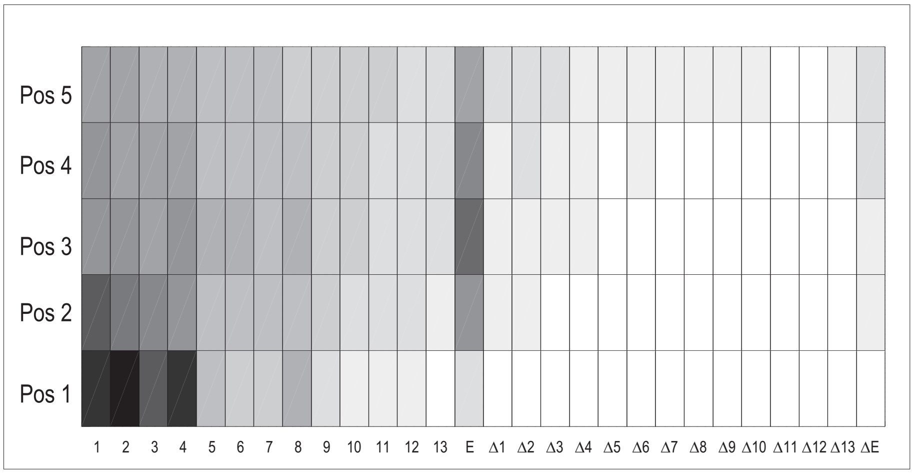

FIG. 4.6 – *Taux de sélection de chaque coefficient selon l'ordre dans lequel ils ont été choisis par le C4.5 (pos1 = sélectionné en premier), même échelle que la figure 4.3.*

4.4 Sélection et combinaison des classificateurs

La qualité de l'entraînement des différents classificateurs va dépendre des positions relatives d'un couple client/anti-client dans l'espace des paramètres. Cela implique que certains classificateurs seront moins discriminants que d'autres. Pour tenir compte de cet effet, nous allons pondérer l'influence de chacun des classificateurs binaires. Cette opération s'effectue en ajustant le coefficient de pondération (ω_{ij}) de chacun des classificateurs. Une fois ceux-ci entraînés, nous utilisons un **ensemble de réglage**, composé d'accès du locuteur et d'accès d'imposture (dont les locuteurs sont différents des clients et des anti-clients, voir l'annexe B.2), pour calculer des distributions de scores de sortie de chacun des classificateurs. Ces distributions permettent par la suite de définir un critère de calcul des (ω_{ij}). La figure 4.8 montre ces deux distributions ainsi que les paramètres utiles à la sélection des classificateurs.

La sortie de chaque classificateur fournit une décision client/imposteur. Il existe plusieurs possibilités de recombiner ces décisions partielles: la plus simple consiste à prendre la moyenne (ou la somme) des scores de tous les classificateurs de D^c (voir la section 4.2). On peut comparer cette approche à la somme des scores LLR du système de référence (voir l'annexe A). Cependant, les deux approches diffèrent dans le fait que chaque classificateur de D^c se concentre seulement sur la tâche de séparer le client d'un anti-client, alors que dans le cas d'un système HMM utilisant un modèle de client et un modèle de monde, la séparation est effectuée entre le client et tous les autres. Plusieurs techniques de recombinaison des sorties des classificateurs ont été expérimentées et sont décrites ci-dessous. Elles s'appuient sur les erreurs de classification sans faire d'hypothèse sur les distributions des scores (voir la section 4.4.1), soit de manière plus générale sur les paramètres de la distribution des scores du client, soit des deux distributions (voir la section 4.4.2). L'élagage des classificateurs peut se faire de manière binaire et dans ce cas, soit on garde le classificateur ($\omega_{ij} = 1$), soit on l'élimine ($\omega_{ij} = 0$). Il est aussi possible d'estimer une pondération

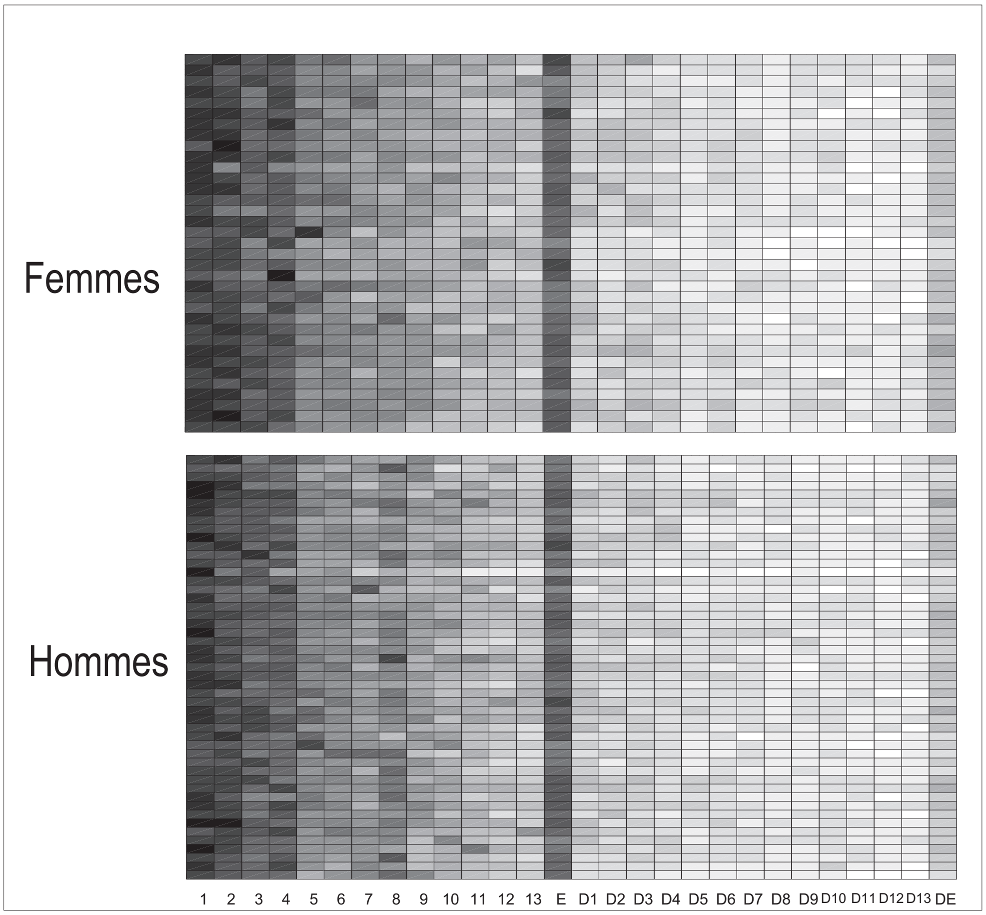

FIG. 4.7 – *Taux d'utilisation de chaque coefficient par rapport à une utilisation de tous les coefficients lors de tous les tests répartis selon les clients de Polycost, même échelle que la figure 4.3.*

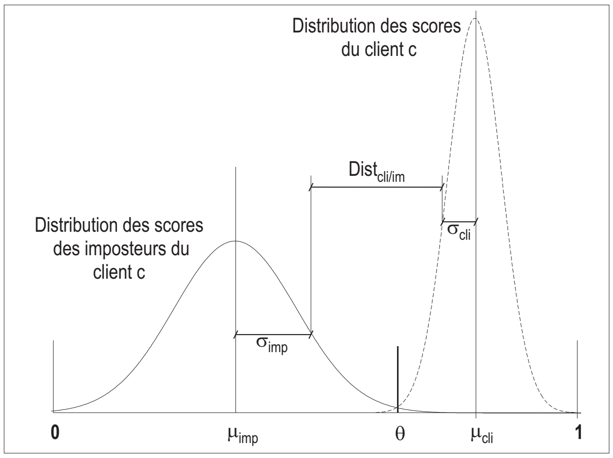

FIG. 4.8 – *Distributions des scores clients et imposteurs pour un classificateur de D^c et les mesures utilisées.*

de ceux-ci par des valeurs de ($\omega_{ij} \in I\!R$) avec comme condition ($\sum_i \sum_j \omega_{ij} = 1$) pour avoir des scores normalisés.

4.4.1 Elagage par dénombrement d'erreurs

L'élagage le plus simple consiste à ne faire aucune hypothèse sur les distributions des scores. On balaie simplement l'axe des scores avec un seuil de décision Θ pour trouver le point d'égale erreur de classification (Equal Error Rate, EER) (voir la figure 4.9). Si ($EER > 0$), le classificateur correspondant est alors désactivé ($\omega_{ij} = 0$).

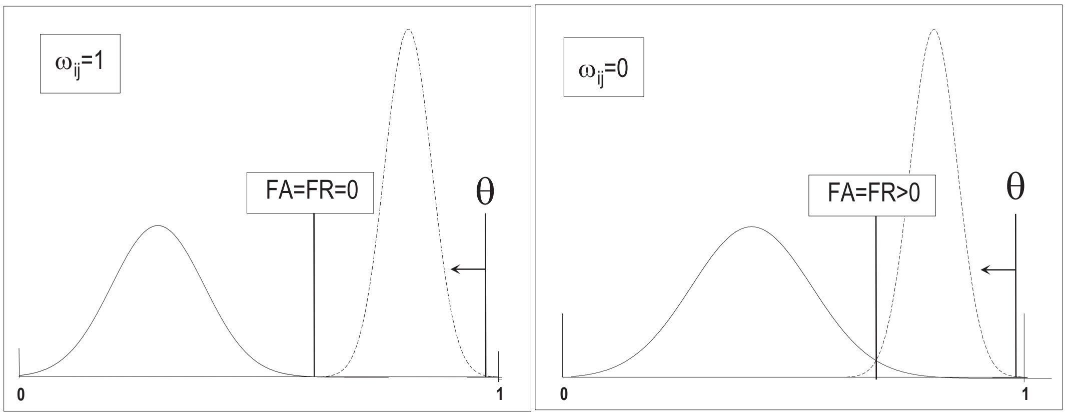

FIG. 4.9 – *Processus d'élagage à erreur 0*

4.4.2 Elagage par rapport aux paramètres des distributions

Si l'on tient compte de la forme des distributions des scores d'accès clients et imposteurs, il est possible de définir des critères de sélection des classificateurs selon les paramètres de ces distributions. Nous approximerons les distributions des scores par des distributions gaussiennes (hypothèse vérifiée à $\simeq 98\%$ avec le test de Cramer-Von Mises, voir [Saporta, 1990] et la section 3.4.3).

Elagage selon la distribution du client

La moyenne et l'écart-type de la distribution des scores du client $\mathcal{N}(\mu_{\text{cli}}, \sigma_{\text{cli}})$ (voir figure 4.8), sont estimés sur les données de réglage. Les classificateurs pour lesquels les accès imposteurs ont des scores supérieurs à un seuil ($\Theta = \mu_{\text{cli}} - \sigma_{\text{cli}}$) et les classificateurs pour lesquels plus d'un score des accès client est inférieur à Θ sont supprimés ($\omega_{ij} = 0$). Tous les autres sont conservés ($\omega_{ij} = 1$).

Elagage selon la distance inter-distribution

On utilise ici la distance entre les distributions des scores du client et des imposteurs (voir la figure 4.8):

$$Dist_{\text{cli/im}} = (\mu_{\text{cli}} - \mu_{\text{im}}) - (\sigma_{\text{cli}} + \sigma_{\text{im}}) \tag{4.9}$$

Avec $\mathcal{N}(\mu_{\text{cli}}, \sigma_{\text{cli}})$ et $\mathcal{N}(\mu_{\text{im}}, \sigma_{\text{im}})$ les distributions des scores de sortie pour un classificateur binaire $D^c(i, j)$ du client et d'un imposteur respectivement.

La sélection d'un classificateur binaire s'opère en calculant la distance $Dist_{\text{cli/im}}$. Si elle est négative ou nulle, le classificateur est éliminé ($\omega_{ij} = 0$). Lorsque la distance $Dist_{\text{cli/im}}$ est positive, nous avons soit sélectionné tous les classificateurs de manière binaire ($\omega_{ij} = 1$), soit déterminé les (ω_{ij}) de la manière suivante:

1. En utilisant un *masque continu normalisé*, tel que tous les poids des classificateurs avec $Dist_{\text{cli/im}} > 0$ soient proportionnels à $Dist_{\text{cli/im}}$ et que leur somme soit égale à 1 ($\sum\limits_i \sum\limits_j \omega_{ij} = 1$).
2. *Elagage avec suppression des résidus*. Dans ce cas les poids utilisés au point précédent sont triés par ordre décroissant, puis sommés jusqu'à ce que le résultat soit supérieur à un seuil Ω. Les poids inférieurs à Ω sont forcés à zéro (si $\omega_{ij} < \Omega, \quad \omega_{ij} = 0$), les poids dont la valeur est supérieure au seuil $\omega_{ij} > \Omega$) sont re-normalisés de façon à ce que ($\sum\limits_i \sum\limits_j \omega_{ij} = 1$).

   Cette méthode d'élagage amplifie la contribution des bons classificateurs et supprime celle des plus mauvais. Elle présente cependant l'inconvénient de rajouter un paramètre de réglage supplémentaire.

4.4.3 Elagage avec contraintes *a priori*

Le critère de suppression des classificateurs utilisé jusqu'ici tient compte du pouvoir discriminant de chacun des classificateurs par rapport aux données qu'on lui a fourni. Nous allons cependant introduire des contraintes *a priori* supplémentaires pour assurer qu'au moins un classificateur de chaque couple client/anti-client soit actif. Cette contrainte permet d'assurer que la distribution des anti-clients reste aléatoire. De même, on peut désirer conserver au moins un classificateur par mot du vocabulaire de la matrice D^c afin de garantir une certaine représentation phonétique. Pour cela, nous avons introduit deux élagages supplémentaires:

Elagage avec contrainte sur les anti-locuteurs

Dans ce cas, soit nous gardons *un et un seul classificateur* de chaque couple client/anti-client, celui qui a la plus grande distance $Dist_{\text{cli/im}}$ (voir équation 4.9), ce qui permet d'assurer la présence de chacun des anti-clients, même si $Dist_{\text{cli/im}}$ est négative. Ou alors nous gardons tous les classificateurs dont ($Dist_{\text{cli/im}} > 0$) et *au moins un classificateur* pour chaque couple client/anti-client.

Elagage avec contrainte sur le vocabulaire

Nous utilisons ici la même procédure que pour la contrainte client/anti-client mais, cette fois, nous assurons la présence d'au moins un classificateur par mot. Nous aurons donc une configuration où nous aurons *au moins un classificateur* par mot et une configuration où nous aurons *un et un seul classificateur* par mot.

4.5 Résultats

Les résultats obtenus ont été calculés sur les données de **test** de la base de données Polycost (voir l'annexe B.2). Pour chacune des configurations décrites dans la section 4.4, nous allons calculer les performances obtenues en comparaison avec le système de référence HMM (voir annexe A) utilisé sur Polycost. Les classificateurs sont estimés sur les données **d'entraînement**. Nous estimons les performances intrinsèques du système en calculant les taux d'erreurs (FA et FR) obtenus en utilisant un seuil de décision *EER (Equal Error Rate a posteriori)*. Ensuite, nous mesurons les erreurs lorsque l'on fixe un point de fonctionnement pour le système en utilisant un seuil *EER a priori* calculé sur les données de **réglage**.

Les abréviations suivantes sont utilisées dans les différents tableaux et figures de résultats:

1. **HMM**: Le système de référence HMM.
2. **MB**: La matrice de classificateurs binaires est sélectionnée avec ($\omega_{ij} = 1$) pour tous les classificateurs.
3. **MB-clidist-d**: Les ω_{ij} sont déterminés en regardant le taux d'erreur obtenu au point ($\mu_{\text{cli}} - \sigma_{\text{cli}}$),avec les contraintes: nombre de fausses acceptations $= 0$, nombre de faux rejets ≤ 1.
4. **MB-dist-b**: Les ω_{ij} sont binaires et estimés selon la distance $D_{\text{cli/im}}$.

5. **MB-dist-c**: Les ω_{ij} sont réels et estimés selon la distance $Dist_{\text{cli/im}}$.
6. **MB-dist-d-minAnti**: Les ω_{ij} sont déterminés comme en 4 mais au minimum un classificateur par anti-locuteur doit être sélectionné afin de garantir la répartition aléatoire des anti-locuteurs. Cela implique que si ($Dist_{\text{cli/im}} < 0$) on prend le classificateur avec la plus petite distance, proche de 0.
7. **MB-dist-d-maxAnti**: Les ω_{ij} sont déterminés comme en 4 et on sélectionne seulement le meilleur classificateur pour chaque anti-client (i.e. un et un seul classificateur).
Si ($Dist_{\text{cli/im}} < 0$), on procède comme en 6.
8. **MB-dist-d-minChif**: Les ω_{ij} sont déterminés comme en 4 mais au minimum un classificateur par mot de vocabulaire doit être sélectionné afin de garantir la plus grande couverture phonétique possible. Cela implique que si $Dist_{\text{cli/im}} < 0$ on prend le classificateur avec la plus petite distance proche de 0.
9. **MB-dist-d-maxChif**: Les ω_{ij} sont déterminés comme en 4 et on sélectionne seulement le meilleur classificateur pour chaque mot de vocabulaire (i.e un et un seul classificateur), si $Dist_{\text{cli/im}} < 0$, on procède comme en 8.

Référence **HMM**		$\mathbf{0.25}^{\pm 0.11}$(ERR%)	
Méthodes	EER%	Méthodes	EER%
MB	$1.03^{\pm 0.24}$	MB-dist-d-maxAnti	$1.56^{\pm 0.27}$
MB-clidist-b	$1.28^{\pm 0.25}$	MB-dist-d-minAnti	$\mathbf{0.78}^{\pm 0.19}$
MB-dist-b	$\mathbf{0.85}^{\pm 0.2}$	MB-dist-d-maxChif	$1.58^{\pm 0.27}$
MB-dist-c	$1.06^{\pm 0.24}$	MB-dist-d-minChif	$\mathbf{0.75}^{\pm 0.19}$

TAB. 4.2 – *Performances des différents systèmes avec un seuil de décision a posteriori.*

Le tableau 4.2 montre les résultats obtenus avec le seuil *EER a posteriori* sur les données de **test**. Dans ce cas, le système de référence HMM donne les meilleurs résultats, bien que les performances des deux élagages, qui tendent à conserver au moins un classificateur par couple client/anti-client ou au moins un classificateur par mot, donnent des résultats très proches. Constatons encore que l'élagage détériore les performances intrinsèques du système dans bien des cas.

Le tableau 4.3 montre les performances obtenues par les différents systèmes avec un point de fonctionnement (seuil *a priori* calculé à l'*EER* sur les données de **réglage**). Dans ce cas, on retrouve la dégradation des performances classique des HMM avec un taux d'erreur 20 fois plus grand ici (voir par exemple [Bimbot *et al.*, 1997], ou le chapitre 2). A l'opposé, quelques-uns des systèmes élagués sont beaucoup plus robustes, même si la dégradation des résultats reste non négligeable par rapport aux performances intrinsèques du tableau 4.2. Le tableau 4.2 nous indique cependant que si une différence significative peut être véritablement observée entre les différentes méthodes et le système de référence, il est évidemment nécessaire de confirmer ces résultats sur d'autres bases de données.

Méthodes	FR%	FA%	HTER%
HMM	10.22$\pm$ 1.54	0.16$\pm$0.21	**5.19**
MB	11.10$\pm$1.63	0.71$\pm$0.45	5.9
MB-clidist-b	9.39$\pm$1.51	0.62$\pm$0.42	**5.01**
MB-dist-b	8.23$\pm$1.43	0.78$\pm$0.47	**4.50**
MB-dist-c	13.01$\pm$1.75	0.93$\pm$0.51	6.98
MB-dist-d-maxAnti	11.49$\pm$1.66	1.25$\pm$0.59	6.36
MB-dist-d-minAnti	10.2$\pm$1.57	0.50$\pm$0.37	5.36
MB-dist-d-maxChif	11.41$\pm$1.66	1.37$\pm$0.62	6.39
MB-dist-d-minChif	10.16$\pm$1.57	0.47$\pm$0.37	5.31

TAB. 4.3 – *Performances des différents systèmes avec un seuil de décision a priori. HTER est l'erreur totale : (FA+FR)/2.*

Méthodes	FR%	FA%	HTER%
MB-0err	15.60$\pm$1.88	1.22$\pm$0.59	8.40
MB-clidist-b	26.60$\pm$2.34	2.40$\pm$0.82	14.49
MB-dist-d-minAnti	14.00$\pm$1.84	0.31$\pm$0.30	7.15
MB-dist-d-minChif	14.08$\pm$1.84	0.33$\pm$0.30	7.21

TAB. 4.4 – *Performances de quelques-uns des systèmes binaires lorsque le signal d'entrée est le résiduel H+N et qu'un seuil déterminé a priori est utilisé. HTER est l'erreur totale : (FA+FR)/2.*

4.6 Reconnaissance avec pré-traitement H+N

De manière à vérifier quels sont les paramètres sélectionnés par les arbres de décision lorsqu'on utilise le résiduel H+N du chapitre 3 comme signal d'entrée, nous avons injecté celui-ci dans notre système. Nous pouvons observer dans le tableau 4.4 que le résiduel contient moins d'information caractéristique du locuteur puisque les faux rejets augmentent significativement. La figure 4.10 montre que l'ordre des coefficients les plus sélectionnés change peu, la seule différence remarquable étant que le coefficient d'énergie est utilisé plus rapidement.

4.7 Améliorations du système

Bien que le système binaire que nous avons choisi ici atteigne déjà des performances intéressantes, il est cependant imaginable de l'améliorer à plusieurs points de vue. Tout d'abord, il est évidemment possible d'ajouter des anti-clients à la matrice D^c afin de mieux décrire la position d'un client dans l'espace des paramètres. Le nombre d'anti-clients, mais surtout leur répartition autour du client, est importante. La stratégie qui consiste à garder au moins un classificateur par anti-client semble efficace, et somme toute cohérente avec notre tâche, puisque nous assurons ainsi la répartition statistique des anti-clients. Il est également possible d'effectuer un test qui soit comparable à celui d'un rapport de vraisemblances où l'on essaie de discriminer un client de tous les autres en utilisant un modèle de monde (voir l'annexe A). On pourrait ajouter une co-

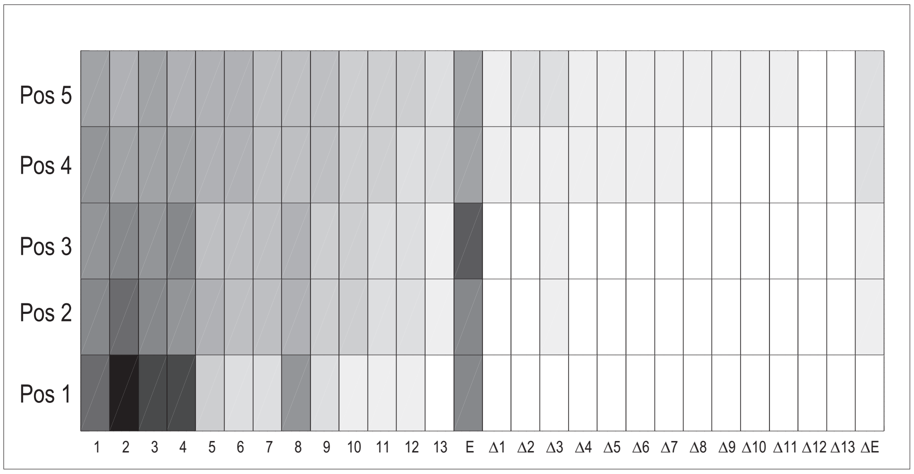

FIG. 4.10 – *Taux de sélection de chaque coefficient selon l'ordre dans lequel ils ont été choisis par le C4.5 (pos1 = sélectionné en premier) avec le résiduel H+N comme signal d'entrée, même échelle que la figure 4.3.*

lonne de la matrice où chaque classificateur serait entraîné avec les données du client et de tous les anti-clients. En ce qui concerne les arbres de décision, plusieurs améliorations sont possibles, comme par exemple l'utilisation de classificateurs correcteurs [Moreira et Mayoraz, 1998], où, pour chaque paire binaire, on rajoute un deuxième classificateur qui sépare les données du client et d'un anti-client de tous les autres anti-clients. Des expériences doivent également être effectuées avec plus de coefficients LPCC et leur accélération.

4.8 Conclusion sur la décomposition en éléments binaires

Nous avons pu montrer dans ce chapitre qu'une bonne décomposition d'un problème complexe (la reconnaissance du locuteur) pour lequel nous possédons peu de données, permet d'atteindre des performances aussi bonnes (voir meilleures) que les systèmes à l'état de l'art en exploitation. On obtient ces résultats en utilisant une stratégie qui consiste à employer des classificateurs simples et à supprimer ceux qui n'ont pas été correctement entraînés. De plus, les arbres de décision, utilisés ici comme classificateurs binaires, nous ont permis d'analyser plus en détails quels sont les coefficients LPCC qui portent l'information la plus discriminante (au sens d'un critère d'entropie), selon le vocabulaire ou les anti-clients utilisés. L'utilisation du résiduel H+N discuté au chapitre 3 comme pré-traitement du signal utilisé avec notre système à paires binaires confirme qu'il reste bien de l'information discriminant les locuteurs.

Chapitre 5

Ajustement du point de fonctionnement

5.1 Introduction

Comme nous l'avons vu au chapitre 2, les systèmes de vérification du locuteur sont constitués de deux modules principaux, l'entraînement des modèles, mémorisant les caractéristiques de chaque locuteur et un module de test. Celui-ci est composé d'une mesure de distorsion entre le modèle de référence et un échantillon de test et d'un étage de décision qui permet de décider d'accepter ou de rejeter une séquence de parole présentée au système. Lorsque nous utilisons des modèles probabilistes (typiquement des HMM), nous nous basons sur l'estimation du logarithme (*log*) du rapport de vraisemblances (voir la section 1.2.4) qui nécessite un *modèle de locuteur* pour chaque *client* de l'application et un modèle de non-locuteur, appelé modèle de *monde*, identique pour tous les clients. Ce *log* du rapport de vraisemblances est comparé à un seuil de décision défini *a priori*. Ce dernier est le point de fonctionnement du système et est défini par les contraintes de l'application et ne devrait pas, en théorie, dépendre des clients de l'application (voir l'équation 1.30).

En pratique cependant, une différence entre le modèle et les données est souvent observée, ce qui invalide l'idée d'un seuil de décision indépendant du locuteur (voir par exemple le chapitre 2). Les sources de divergence entre modèle et données sont multiples, citons par exemple le mauvais dimensionnement du modèle, la non-représentativité des données d'entraînement ou la non- prise en compte des variabilités intra- et inter-locuteurs.

Nous proposons dans ce chapitre un ajustement du test du rapport de vraisemblances, de manière à corriger quelques-unes des différences modèle/données en faisant quelques hypothèses supplémentaires sur la distribution des données. Nous allons montrer comment on peut estimer un seuil ajusté en estimant la moyenne et l'écart-type des distributions des vraisemblances locales issues des modèles du client et du monde.

5.2 Rappels

Si nous reprenons la théorie des modèles probabilistes de la section 1.2.4, nous pouvons définir M_C comme étant le modèle d'un client C et M_W le modèle de tous les locuteurs qui ne sont pas le client (i.e. le reste de la population terrestre), que nous appelons ici *modèle de monde*. Supposons maintenant que nous disposons d'une séquence de test Y dont nous voulons savoir si elle a été prononcée par le client C ou non.

Si nous appelons $P_{FR/C}$ et $P_{FA/W}$ les probabilités de faux rejets, respectivement les probabilités de fausses acceptations du système, P_C et P_W les probabilités *a priori* de la séquence de test d'appartenir au locuteur ou non, la fonction de coût d'erreur totale du système devient (équation 1.27):

$$c_{tot} = c_{fr} \cdot P_C \cdot P_{FR/C} + c_{fa} \cdot P_W \cdot P_{FA/W} \tag{1.27}$$

Avec c_{fr} et c_{fa} les coûts d'une erreur de rejeter un vrai locuteur et respectivement d'accepter un imposteur.
Notons maintenant $\mathscr{L}_C$ et $\mathscr{L}_W$ les fonctions de vraisemblance des modèles de locuteur et de monde. On peut montrer que la minimisation de c_{tot} est obtenue en implémentant le test du rapport de vraisemblances [Scharf, 1991] selon l'équation 1.29 que nous rappelons ici:

$$LR = \frac{\mathscr{L}(O_t, M_C)}{\mathscr{L}(O_t, M_W)} = \frac{\mathscr{L}_C}{\mathscr{L}_W} \mathop{\gtrless}_{reject}^{accept} \frac{P_W}{P_C} \cdot \frac{c_{fa}}{c_{fr}} \; = \; \Theta\,(R) \; = \; R \tag{1.29}$$

Avec R le rapport de risque:

$$R \;=\; \frac{P_W}{P_C}\,\frac{c_{fa}}{c_{fr}} \tag{5.1}$$

Comme on peut le constater en analysant l'équation 5.1, le seuil optimal ne dépend, en théorie, que du rapport de coût de (fausse acceptation / faux rejet) et du rapport des probabilités *a priori* des imposteurs et du client. Dans le cas particulier où c_{fa} et c_{fr} sont égaux et quand les accès d'un locuteur et d'un imposteur sont équiprobables *a priori* (ce qui est du reste impossible à déterminer), le système est ajusté dans une condition d'**équi-risque** et le choix de ($\Theta = 1$) comme seuil de décision devrait conduire à une **erreur totale** minimum du système en fonctionnement:

$$TER = P_{FR/C} + P_{FA/W} \tag{5.2}$$

5.3 Ajustement du test LR

Cependant, la pratique a montré que le test *LR* avec ($\Theta \;=\; R$) comme seuil de décision ne détermine pas le minimum de la fonction de coût c_{tot}. En fait, le *LR* de l'équation 1.29 est calculé sur la base d'estimations de fonctions de vraisemblance qui ne correspondent pas exactement aux distributions des paramètres acoustiques du client et des imposteurs. La conséquence immédiate est qu'il est souvent avantageux de corriger le seuil du test *LR* pour chaque client, de manière à corriger l'erreur d'ajustement entre le modèle et les données.

Si nous notons maintenant $\widehat{\mathscr{L}}_C$ et $\widehat{\mathscr{L}}_W$ les fonctions de vraisemblance des **modèles** du locuteur et des imposteurs, le test *LR* peut être réécrit dans un cadre plus général comme:

$$\widehat{LR}\,(Y) \;=\; \frac{\widehat{\mathscr{L}}_C\,(Y)}{\widehat{\mathscr{L}}_W\,(Y)} \quad \underset{reject}{\overset{accept}{\gtrless}} \quad \Theta_C\,(R,n) \tag{5.3}$$

Avec n le nombre de vecteurs d'entrée de la séquence de test Y. La fonction Θ_C peut être vue comme un seuil ajusté, dépendant du client, qui tient compte de l'imprécision créant une différence entre LR et $\widehat{LR}$ et de l'influence de la longueur de la séquence de test sur la distribution de $\widehat{LR}$.

Dans le cas général, il n'y a pas de moyen direct de modéliser ou estimer $\Theta_C\,(R,n)$. Cependant, si nous observons la *log*-vraisemblance obtenue en effectuant la moyenne des valeurs de *log*-vraisemblance de vecteurs indépendants issus d'une même variable aléatoire, le seuil ajusté Θ_C peut être directement déduit des moyennes et variances des distributions de vraisemblance locale (i.e. de chaque fenêtre d'analyse) à la sortie des modèles du client et du non-client.

5.4 Distribution de $\widehat{LR}$

Pour la plupart des modèles probabilistes, le logarithme du numérateur de $\widehat{LR}$ peut être réécrit comme la moyenne de *log*-vraisemblances locales [1]:

$$\log\ \widehat{\mathscr{L}}_C\,(Y) \;=\; \frac{1}{n}\sum_{i=1}^{n} \log\ \widehat{\mathscr{L}}_C(y_i) \tag{5.4}$$

Avec y_i le $i^{\text{ème}}$ vecteur de paramètres de la séquence Y de longueur totale n. Si n est suffisamment grand, les valeurs des *log*-vraisemblances locales $\left(\log\ \widehat{\mathscr{L}}_C(y_i)\right)$ sont indépendantes. $\widehat{\mathscr{L}}_C(Y)$ suit une distribution gaussienne $\mathcal{N}\,(\mu_{\mathscr{L}_C};\,\sigma_{\mathscr{L}_C}/\sqrt{n})$, où $\mu_{\mathscr{L}_C}$ et $\sigma_{\mathscr{L}_C}$ sont la moyenne et écart-type de la distribution des *log*-vraisemblances locales, quelle que soit cette distribution (théorème central limite, voir [Saporta, 1990]). La même propriété est aussi applicable au dénominateur, ce qui donne:

$$\begin{aligned}
\log\ \widehat{\mathscr{L}}_C(Y) &\longrightarrow \mathcal{N}\,(\mu_{\mathscr{L}_C}, \sigma_{\mathscr{L}_C}/\sqrt{n})\\
\log\ \widehat{\mathscr{L}}_W(Y) &\longrightarrow \mathcal{N}\,(\mu_{\mathscr{L}_W}, \sigma_{\mathscr{L}_W}/\sqrt{n})
\end{aligned}$$

Si nous distinguons maintenant les séquences réellement prononcées par le client S_C de celles prononcées par les imposteurs de ce client S_I, les numérateur et dénominateur de $\widehat{LR}$ suivent deux distributions conditionnelles différentes:

1. Une approche similaire peut être suivie pour les classificateurs qui utilisent la *somme* plutôt que la moyenne des log-vraisemblances.

– Pour la séquence prononcée par le client S_C on a:

$$\log \widehat{\mathscr{L}}_C(Y|S_C) \longrightarrow \mathcal{N}(\mu_{\mathscr{L}_C}(S_C);\ \sigma_{\mathscr{L}_C}(S_C)/\sqrt{n}) = \mathcal{N}_C^C$$
$$\log \widehat{\mathscr{L}}_W(Y|S_C) \longrightarrow \mathcal{N}(\mu_{\mathscr{L}_W}(S_C);\ \sigma_{\mathscr{L}_W}(S_C)/\sqrt{n}) = \mathcal{N}_W^C$$

– Et pour la séquence prononcée par un imposteur S_I on a:

$$\log \widehat{\mathscr{L}}_C(Y|S_I) \longrightarrow \mathcal{N}(\mu_{\mathscr{L}_C}(S_I);\ \sigma_{\mathscr{L}_C}(S_I)/\sqrt{n}) = \mathcal{N}_C^I$$
$$\log \widehat{\mathscr{L}}_W(Y|S_I) \longrightarrow \mathcal{N}(\mu_{\mathscr{L}_W}(S_I);\ \sigma_{\mathscr{L}_W}(S_I)/\sqrt{n}) = \mathcal{N}_W^I$$

Par exemple, la notation $\mu_{\mathscr{L}_C}(S_I)$ représente la valeur de la moyenne de la *log*-vraisemblance d'une séquence d'imposture lorsqu'on la calcule avec un modèle du client. Finalement, on a les distributions suivantes:

$$\log\ \widehat{LR}\,(Y|S_C) \longrightarrow \mathcal{N}\ (m_{SC}, s_{SC}/\sqrt{n}) = \mathcal{N}_{SC}^{(n)}$$

$$\log\ \widehat{LR}\,(Y|S_I) \longrightarrow \mathcal{N}\ (m_{SW}, s_{SW}/\sqrt{n}) = \mathcal{N}_{SW}^{(n)}$$

Avec :

$$\begin{array}{r|l} m_{SC} = \mu_{\mathscr{L}_C}(S_C) - \mu_{\mathscr{L}_W}(S_C) & s_{SC} = \sqrt{(\sigma_{\mathscr{L}_C}(S_C))^2 + (\sigma_{\mathscr{L}_W}(S_C))^2} \\ m_{SW} = \mu_{\mathscr{L}_C}(S_I) - \mu_{\mathscr{L}_W}(S_I) & s_{SW} = \sqrt{(\sigma_{\mathscr{L}_C}(S_I))^2 + (\sigma_{\mathscr{L}_W}(S_I))^2} \end{array}$$

5.5 Expression du seuil ajusté

Si maintenant nous notons:

$$\mathcal{F}_{\text{SC}}^{(n)}\,(\tau) = \int_{-\infty}^{\tau} \mathcal{N}_{\text{SC}}^{(n)}\,(v)\,dv \quad \text{et} \quad \mathcal{F}_{\text{SW}}^{(n)}\,(\tau) = \int_{-\infty}^{\tau} \mathcal{N}_{\text{SW}}^{(n)}\,(v)\,dv$$

Les fonctions $\left[1 - \mathcal{F}_{\text{SW}}^{(n)}\,(\tau))\right]$ et $\left[\mathcal{F}_{\text{SC}}^{(n)}\,(\tau)\right]$ peuvent être utilisées pour modéliser les probabilités de fausses acceptations et de faux rejets comme fonctions du seuil τ et peuvent être utilisées pour la minimisation de l'équation de coût. Dans ce cas, $\Theta_C\,(R,n)$ devient:

$$\log\ \Theta_C\,(R,n)\ = Argmin_{\,\tau}\ \left\{R\ \left[1 - \mathcal{F}_{SW}^{(n)}\,(\tau)\ +\ \mathcal{F}_{SC}^{(n)}\,(\tau)\right]\right\} \tag{5.5}$$

Le seuil ajusté $\Theta_C\,(R,n)$ est donc estimé à partir des modélisations gaussiennes des distributions de *log*-vraisemblances fournies par les modèles de client et de monde en utilisant des données du client et d'imposteurs.

Approximation du seuil ajusté

Une solution numérique de l'équation (5.5) peut être calculée, sachant $\mathcal{N}(\mu, \sigma)$, en utilisant une approximation polynomiale:

$$\widehat{\mathcal{F}(\tau)} = \int_{-\infty}^{\tau} \mathcal{N}(v)dv$$

de la manière suivante (selon Saporta [1990]), avec une précision de l'ordre de 10^{-7}:

$$u = \frac{\tau - \mu}{\sigma}, \; s = sgn\,(u), \; g = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}u}, \; t = \frac{1}{1 + \, s\,a\,u}$$

$$f \simeq 1 - g(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5)$$

Et finalement:

$$\widehat{\mathcal{F}(\tau)} = sf + \frac{1-s}{2}$$

En utilisant les constantes numériques suivantes:

$a\ = 0.231641900$	$b_1 = \ \ 0.319381530$	$b_2 = -0.356563782$
$b_3 = 1.781477937$	$b_4 = -1.821255978$	$b_5 = \ \ 1.330274429$

5.6 Estimation du seuil

Pour chaque client C, le modèle statistique qui lui est associé $\mathscr{L}_C$ est estimé à partir des données d'entraînement (sous-ensemble $\mathcal{E}$), afin d'analyser l'effet des variabilités intra-locuteur et de la quantité de données sur l'estimation du seuil. Pour ce faire nous utilisons 2 ensembles d'entraînement, le premier, noté $\mathcal{E}_1$, ne contient qu'une seule session d'enregistrement (4 séquences du code personnel, voir la base de données Polycode annexe B.1) et le second utilise 2 sessions d'entraînement ($\mathcal{E}_2$).

Le modèle de monde est estimé sur une base de données différente (Polyphone, annexe B.4) dont les locuteurs sont différents de Polycode.

Nous utilisons également des locuteurs (notés *pseudo-imposteurs*) issus des données de réglage de Polycode (sous-ensemble $\mathcal{I}$) et qui sont différents des deux autres ensembles de locuteurs. Les pseudo-imposteurs de $\mathcal{I}$ nous permettent d'estimer les paramètres des distributions $\mathcal{N}_C^I$ et $\mathcal{N}_W^I$ en utilisant les modèles du client $\mathscr{L}_C$ et de monde $\mathscr{L}_W$.

Nous calculons les distributions $\mathcal{N}_C^C$ et $\mathcal{N}_W^C$ des modèles du client $\mathscr{L}_C$ et de monde $\mathscr{L}_W$ en utilisant des sous-ensembles de données différents:

1. Une seule session d'entraînement $\mathcal{E}_1$, qui représente 4 répétitions du code personnel.
2. Deux sessions d'entraînement $\mathcal{E}_2$, séparées au moins d'un jour entre les sessions.
3. Une, deux ou trois sessions de réglage différentes des données utilisées pour les tests et pour les entraînements $\mathcal{E}_{1,2}$. Ces sessions sont nommées respectivement $\mathcal{T}_1$,$\mathcal{T}_2$ et $\mathcal{T}_3$.
4. Cinq sessions de tests, composant le sous-ensemble $\mathcal{A}$. $\overline{\mathcal{A}}$ représente tous les tests d'imposture effectués sur les locuteurs.

5.6.1 Répartition temporelle des données

Les données d'entraînement $\mathcal{E}_1$, $\mathcal{E}_2$, les données d'évaluation $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ et les données de test $\mathcal{A}$ et $\overline{\mathcal{A}}$ sont réparties temporellement de la façon suivante:

$$\mathcal{E}_1 < \mathcal{E}_2 < \mathcal{T}_1 < \mathcal{T}_2 < \mathcal{T}_3 < (\mathcal{A}, \overline{\mathcal{A}})$$

Où le signe "$<$" désigne une précédence chronologique.

5.6.2 Résultats

Estim. de Θ		FA (%)	FR (%)	TER (%)
1.	$\log \Theta = 0$	0.40	28.11	28.51
2.	$\mathcal{E}_1$ et $\bar{\mathcal{I}}$	0.47	8.94	9.41
3.	$\mathcal{T}_1$ et $\bar{\mathcal{I}}$	5.37	3.18	8.55
4.	$\mathcal{T}_2$ et $\bar{\mathcal{I}}$	5.11	2.59	7.70
5.	$\mathcal{T}_3$ et $\bar{\mathcal{I}}$	4.81	1.58	6.39
6.	$\mathcal{A}$ et $\bar{\mathcal{A}}$	2.83	1.00	3.83
7.	min TER	0.11	1.29	1.40

TAB. 5.1 – *Taux de fausses acceptations, de faux rejets et taux d'erreur total pour divers ajustements du seuil Θ. Résultats obtenus avec* ***une session d'entraînement*** *des modèles clients.*

Estim. de Θ		FA (%)	FR (%)	TER (%)
1.	$\log \Theta = 0$	1.21	6.47	7.68
2.	$\mathcal{E}_2$ et $\bar{\mathcal{I}}$	0.37	6.68	7.05
3.	$\mathcal{T}_1$ et $\bar{\mathcal{I}}$	4.68	2.25	6.93
4.	$\mathcal{T}_2$ et $\bar{\mathcal{I}}$	3.84	2.25	6.09
5.	$\mathcal{T}_3$ et $\bar{\mathcal{I}}$	3.84	1.25	5.09
6.	$\mathcal{A}$ et $\bar{\mathcal{A}}$	2.44	1.00	3.44
7.	min TER	0.03	1.29	1.32

TAB. 5.2 – *Taux de fausses acceptations, de faux rejets et taux d'erreur total pour divers ajustements du seuil Θ. Résultats obtenus avec* ***deux sessions d'entraînement*** *des modèles clients.*

Les tableaux 5.1 et 5.2 résument les résultats obtenus en utilisant les différents sous-ensembles expliqués à la section 5.6. Le tableau 5.1 correspond à l'utilisation d'une seule session d'apprentissage $\mathcal{E}_1$ et le tableau 5.2 à l'emploi de 2 sessions d'apprentissage $\mathcal{E}_2$ pour estimer les modèles de client. On compare notre ajustement de seuil au cas théorique où $\log \Theta = 0$ (point *1.* des tableaux de résultats) à l'estimation du seuil sur les données d'entraînement (point *2.*), puis lorsqu'on utilise une, deux ou trois sessions de réglage (points *3.*, *4.* et *5.*). Les points *6.* et *7.* donnent les résultats avec des seuils déterminés *a posteriori*.

Les différences entre les résultats *1.* et *2.* illustrent l'importance d'utiliser un seuil d'entraînement ajusté et dépendant du locuteur. Les résultats *3.*, *4.* et *5.* montrent que l'estimation du seuil devient de plus en plus précise lorsque l'on capte de mieux en mieux la variabilité des locuteurs.

Un autre point intéressant est la différence entre les résultats *5.* et *6.* qui montre l'erreur d'estimation que l'on fait parce que la distribution gaussienne estimée sur des données de réglage ne s'ajuste par exactement sur les distributions réelles.

Distributions parole/non-parole

Estim. de Θ		FA (%)	FRR (%)	TER (%)
1.	$\log\Theta = 0$	1.08	16.54	17.62
2.	$\mathcal{E}_1$ et $\bar{\mathcal{I}}$	0.95	4.94	5.89
3.	$\mathcal{T}_1$ et $\bar{\mathcal{I}}$	4.26	2.00	6.26
4.	$\mathcal{T}_2$ et $\bar{\mathcal{I}}$	3.55	1.50	5.05
5.	$\mathcal{T}_3$ et $\bar{\mathcal{I}}$	2.89	1.00	3.89
6.	$\mathcal{A}$ et $\bar{\mathcal{A}}$	2.96	1.00	3.96
7.	min TER	0.05	1.54	1.59

TAB. 5.3 – *Taux de fausses acceptations, de faux rejets et taux d'erreur total pour divers ajustements du seuil Θ. Résultats obtenus avec* ***une session d'entraînement*** *des modèles clients, en ayant conservé* ***95% des trames*** *les plus vraisemblables.*

Estim. de Θ		FA (%)	FRR (%)	TER (%)
1)	$\log\Theta = 0$	2.21	5.34	7.55
2)	$\mathcal{E}_2$ et $\bar{\mathcal{I}}$	0.71	4.09	4.80
3)	$\mathcal{T}_1$ et $\bar{\mathcal{I}}$	3.66	1.50	5.16
4)	$\mathcal{T}_2$ et $\bar{\mathcal{I}}$	2.95	1.25	4.20
5)	$\mathcal{T}_3$ et $\bar{\mathcal{I}}$	2.87	1.00	3.87
6)	$\mathcal{A}$ et $\bar{\mathcal{A}}$	2.65	1.00	3.65
7)	min TER	0.00	1.00	1.00

TAB. 5.4 – *Taux de fausses acceptations, de faux rejets et taux d'erreur total pour divers ajustements du seuil Θ. Résultats obtenus avec* **deux sessions d'entraînement** *des modèles clients, en ayant conservé* **95% des trames** *les plus vraisemblables.*

L'observation pratique des distributions des *log*-vraisemblances locales a permis de constater que souvent deux distributions distinctes apparaissaient. Nous avons donc émis l'hypothèse que certains vecteurs, dont la vraisemblance d'être issus d'un modèle est très faible, provient de parties du signal ne contenant pas de la parole. Un moyen simple de s'affranchir de ce problème consiste à supprimer les trames de trop faible vraisemblance. C'est la solution pour laquelle nous avons opté, en ne gardant que 95% des trames les plus vraisemblables. Les tableaux 5.3 et 5.4 donnent les résultats ainsi obtenus. On peut constater une nette amélioration de tous les résultats obtenus avec un seuil *a priori*. Remarquons encore qu'une meilleure estimation peut être effectuée en calculant le seuil de réjection sur la distribution des *log*-vraisemblances très faibles. Cependant, cette méthode a l'inconvénient de rajouter un paramètre de réglage supplémentaire à un système qui en comporte déjà beaucoup.

5.7 Conclusion sur les ajustements de seuils

L'approche suivie dans ce chapitre montre que, lorsque les algorithmes utilisés sont basés sur une modélisation stochastique, les distributions des *log* des rapports de vraisemblances (LLR) des clients et des imposteurs peuvent être approximées par des modèles gaussiens, si l'on tient compte du fait que les tests LLR deviennent dépendants du locuteur et du nombre de vecteurs utilisés pour le calculer. Nous montrons également qu'il y a intérêt, lors de la segmentation du signal par un système de reconnaissance automatique de la parole, à supprimer les valeurs de vraisemblances locales trop faibles.

Chapitre 6

Fusion de décisions

La fusion de données ou le mélange d'experts est une partie du domaine de l'apprentissage automatique ("machine learning") qui prend de l'importance dans le traitement de problèmes complexes, plus particulièrement ceux qui nécessitent une prise de décision concertée entre plusieurs sources (voir [Dasarathy, 1994] et [Fusion,1998, Xu *et al.*, 1992]). En effet, au lieu de confier la solution d'un problème à un seul algorithme (expert), on tente de diminuer l'erreur globale d'un système en demandant à plusieurs experts simultanément de prendre une décision selon leurs compétences, celles-ci étant ensuite fusionnées pour aboutir à une décision finale. Les premières expériences de combinaison que nous avons effectuées dans le chapitre 2 ont permis de constater les bénéfices d'une telle approche.

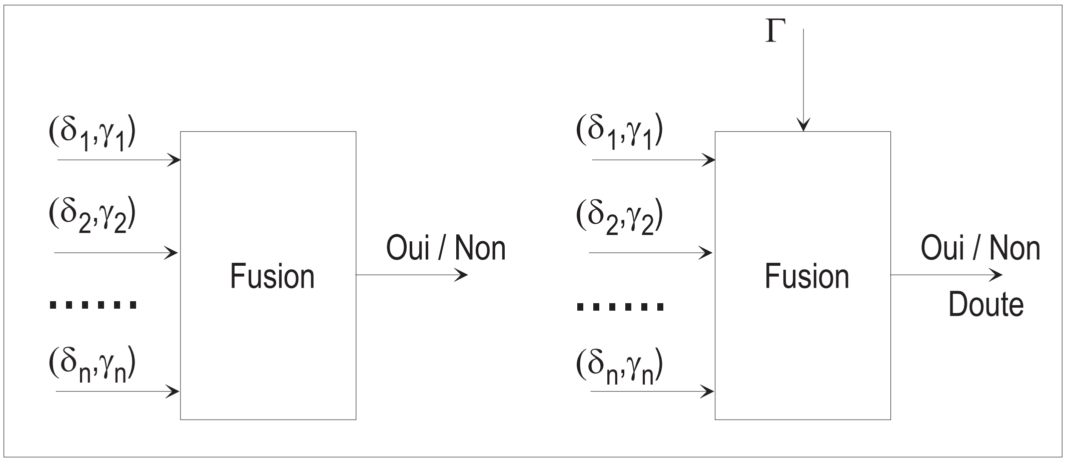

FIG. 6.1 – *Fusion de décision, avec sortie binaire (oui/non) ou ternaire (oui/non/doute)*

Nous définissons un système de **fusion de décisions** comme une boîte noire qui accepte en entrée des sources de décision partielles δ_i, avec $(i \in 1 \cdots n)$ et $\delta_i \in \{0, 1\}$, éventuellement munies d'une confiance γ_i et qui fournit en sortie une décision finale basée sur la combinaison des entrées. La figure 6.1 donne une idée d'un tel système. La décision finale peut être soit binaire (oui/non), soit ternaire (oui/non/doute) comme nous l'avons déjà vu au chapitre 2. Dans ce dernier cas, il faut rajouter comme paramètre d'entrée supplémentaire au système de fusion, un seuil Γ qui permettra de régler le niveau de doute du système.

Le domaine de la fusion étant extrêmement vaste, nous ne prétendons pas ici le traiter de manière exhaustive, mais nous allons plutôt l'aborder par des cas pratiques qui ont permis d'améliorer les résultats de nos systèmes. Nous présenterons donc ici, sous un autre angle, certains des travaux expliqués lors des chapitres précédents (chapitres 2 et 4), mais aussi nos expériences effectuées dans le cadre du consortium ELISA [1].

6.1 Niveau de confiance dans une décision

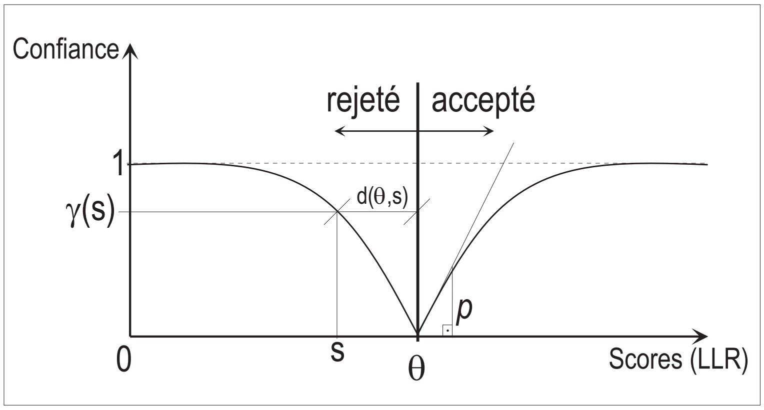

FIG. 6.2 – *Calcul de la confiance $\gamma(s)$ dans une décision pour un score s et un seuil Θ. Chaque demi-sigmoïde a pour valeur 0 en Θ, la pente p est un paramètre fixé expérimentalement.*

De manière à pondérer les décisions partielles δ_i (voir figure 6.1), il est possible de leur adjoindre une valeur réelle donnant **la confiance** avec laquelle l'expert a donné sa décision. Si les confiances de chacun des experts sont *normées*, chacune d'elle va varier dans le même intervalle, par exemple $[0, 1]$. Si elles ne sont pas normées, il faut alors connaître leur plage de variation et un paramètre d'échelle supplémentaire doit être transmis en entrée du fusionneur. Notons la différence entre fusion de décisions, qui ne nécessite aucune hypothèse *a priori* sur les *interactions* entre experts et la fusion d'informations quelconques issues de différentes sources qui demande une phase de calcul *a priori* pour déterminer les poids relatifs de chaque expert. La confiance dans une décision peut typiquement être calculée comme une fonction de la distance au seuil de décision $d(\Theta, S)$ (voir la figure 6.2). La confiance est nulle sur le seuil, et maximale en $(\pm\infty)$. La figure 6.2 rappelle les différents paramètres utilisés pour calculer une confiance.

1. ELISA est constituée de plusieurs laboratoires Européens (ENST, IRISA, EPFL, IDIAP, LIA) qui ont mis en commun leur savoir-faire pour créer un système de référence évolutif, en vérification du locuteur. La première version de cette plate-forme a été présentée avec succès aux évaluations NIST-1998.

6.2 Niveaux de fusion

Selon les sources des décisions partielles qui forment l'entrée du système de fusion, on peut distinguer plusieurs niveaux de combinaison: la **fusion d'éléments**, la **fusion de méthodes** ou la **fusion de modes**. Nous allons détailler ces différentes approches dans ce paragraphe au travers d'exemples pratiques.

6.2.1 Fusion d'éléments

La fusion d'éléments consiste à combiner des décisions partielles prises sur plusieurs éléments d'une même entité. Par exemple, au lieu de combiner les scores de chacun des classificateurs de la matrice D^C du chapitre 4 pour en extraire un score moyen comparé à un seuil, on peut demander à chaque classificateur de donner une décision pondérée par une confiance et ensuite fusionner ces confiances partielles.

6.2.2 Fusion de méthodes

La fusion de méthodes consiste à combiner des décisions provenant d'algorithmes différents mais qui s'appliquent à des données identiques. Comme exemple, nous pouvons prendre la combinaison des trois méthodes de vérification du locuteur du chapitre 2, soit les *SSO+S*, *DTW* et *HMM*, où les algorithmes travaillent sur les mêmes vecteurs d'entrée (LPCC, voir section 1.2.2) issus d'un mot prononcé par un locuteur. Le tableau 6.1 montre le bénéfice que l'on retire en combinant les trois décisions. L'algorithme de fusion utilisé ici est un système à vote majoritaire (voir la section 2.5). Si de plus on adjoint une confiance aux décisions de chaque algorithme, on peut choisir parfois de ne pas prendre de décision binaire (accepter/refuser) lorsque la confiance des experts majoritaires devient trop faible (voir le tableau 6.2). Cette approche nous oblige cependant à définir un nouveau paramètre de réglage (le seuil Γ de la figure 6.1) et à traiter les cas de doute de manière différente (par exemple en demandant plus de données), mais nous permet d'augmenter la fiabilité du système en évitant de générer des erreurs.

Méthode	FR% (200 tests)	FA% (1800 tests)	HTER% (2000 tests)
DTW	23.5	7.67	15.8
SSO+S	14.0	5.3	9.9
HMM L/R	5.5	2.7	4.1
Décision combinée	2.0	2.7	2.3

TAB. 6.1 – *Apport de la combinaison de méthodes. La décision combinée est effectuée par vote majoritaire des trois méthodes SSO+S, DTW et HMM.*

6.2.3 Fusion de modes

On parle de **fusion de modes** lorsque les entrées du système de fusion proviennent de décisions prises sur des données d'entrée de types différents, comme, par exemple, la fusion de diverses

Seuil de doute	FR% (200 tests)	FA% (1800 tests)	HTER% (2000 tests)	Doute% (2000 tests)
0.2	2.0	2.7	2.3	0.0
0.5	2.0	2.3	2.15	0.8
0.7	2.0	1.1	1.55	10.1
0.8	1.0	0.9	0.95	20.2

TAB. 6.2 – *Diminution des erreurs de classification lorsque l'on adjoint un calcul de confiance au vote majoritaire et qu'on le compare à un seuil de doute.*

modalités biométriques (voir [Multim-1,1997, Multim-2,1997, Multim-3,1997]), où l'on combine la fusion d'informations de vérification du locuteur et des informations de reconnaissance labiale en estimant une somme pondérée des confiances.

On peut également considérer la fusion des modes issus par exemple d'une paramétrisation différente du signal de parole (MFCC, LPCC, etc. . .). Telle est la solution que nous avons choisie pour les évaluations NIST 1998 [2] [Martin, 1998a, Martin, 1998b].

Dans notre cas, les algorithmes et la paramétrisation sont différents:

– Le premier système, **ENST**, est basé sur les GMM (Gaussian Mixture Models) équivalents à un HMM à un état. Cette modélisation se prête bien à la vérification du locuteur indépendante du texte, comme l'a montré Reynolds en [1992, 1997]. Les vecteurs de paramètres d'entrée sont constitués de 16 coefficients cepstraux issus d'une analyse LPC ainsi que de leurs dérivées et le *log* de la dérivée de l'énergie. Ces paramètres sont extraits sur une fenêtre de 32 [ms] chaque 10 [ms]. Le modèle GMM est constitué d'un mélange de 256 Gaussiennes avec une matrice de covariances diagonale. Les scores fournis à l'étage de décision sont les logarithmes des rapports de vraisemblances (voir la section 1.2.4) du modèle de client et du modèle de monde.
– Le second système, **LIA** (voir [Besacier et Bonastre, 1997a, Besacier et Bonastre, 1997b]), est constitué du système de référence [3] ELISA-1.0, basé sur les statistiques du second ordre (voir la section 1.2.3) mais avec une mesure de vraisemblance plutôt que de sphéricité. Ce système est basé sur la fusion des résultats de plusieurs classificateurs *SSO+S*. Les vecteurs fournis en entrée de chacun d'eux sont constitués des coefficients issus de bancs de filtres divisés en sous-bandes après transformation temps-fréquence par une FFT. Le système est composé de 24 sous-bandes de fréquence. Chaque sous-bande est constituée de 20 canaux issus de la bande de fréquence totale, mais avec une large superposition de celles-ci. Chaque classificateur associé à une sous-bande prend une décision partielle qui est ensuite fusionnée dans un système à vote majoritaire. Les scores de sortie sont des moyennes des logarithmes du rapport de vraisemblances issues des classificateurs de la majorité. Le principe est ici similaire à la fusion de décisions multi-méthodes de la section 6.2.2, munie d'un système de fusion de décisions avec une moyenne des confiances sur les décisions.

2. NIST: National Institute for Standardization and Technologies, cet institut organise chaque année une évaluation d'algorithmes de vérification du locuteur indépendante du texte.

3. Système conçu par les partenaires d'ELISA.

Choix d'un algorithme de fusion

N'ayant *a priori* aucune information sur les performances respectives des deux systèmes à fusionner, nous avions le choix de combiner ces deux méthodes soit par vote majoritaire, ce qui pose un problème d'indécision lorsque les méthodes sont d'un avis opposé (une voix pour, une voix contre), soit par l'algorithme de Dempster Shafer (voir [Shafer, 1976] et plus loin). Pour éviter le problème de non décision du vote majoritaire mais aussi pour l'algorithme de Dempster, il nous a été nécessaire de calculer une confiance dans la décision prise par chacune des méthodes. Cette confiance est calculée à partir des intégrales des deux distributions (i.e. fonction de répartition) d'imposteurs et de locuteurs issus de données NIST97. Nous désirons que la confiance en chaque décision ait les propriétés suivantes:

- La confiance doit être normée, ce qui implique qu'elle doit varier dans l'intervalle $[0, 1]$, cela afin de les combiner facilement.
- La confiance dans un rejet ($\equiv$ rejeter un échantillon x qui n'appartient pas au locuteur) ne doit pas varier de la même manière que la confiance dans une acceptation ($\equiv$ accepter un échantillon x qui appartient au locuteur), puisque la fonction de coût des évaluations NIST pénalise plus fortement les faux rejets que les fausses acceptations [Martin, 1998a, Martin, 1998b].
- La confiance doit pouvoir être estimée à partir des décisions et des scores de sortie de chacune des méthodes, mais sans connaître la valeur du seuil qui a permis de prendre la décision.

Calcul de la confiance

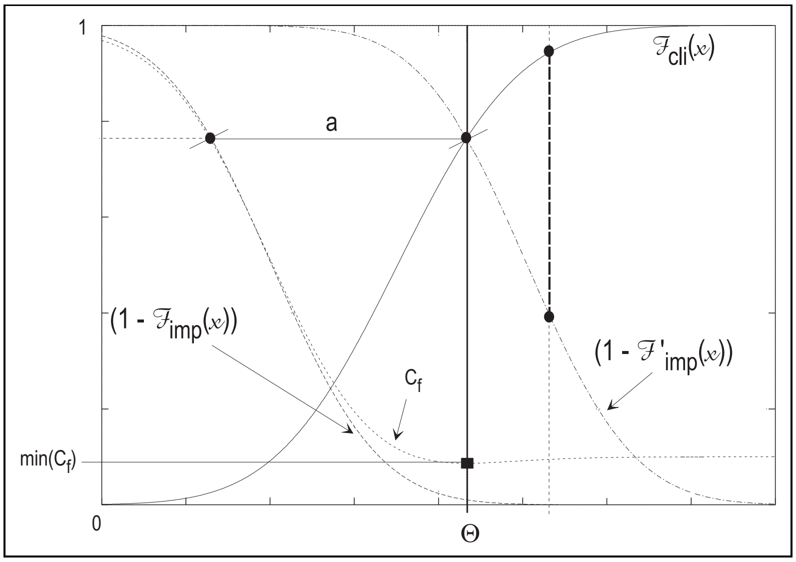

FIG. 6.3 – *Etablissement de la valeur de confiance selon la distance au seuil Θ et les deux fonctions ($\mathcal{F}_{cli}$) et $(1 - \mathcal{F}_{imp})$*

La figure 6.3 montre le principe de calcul de confiance qui a été choisi. La détermination du seuil s'effectue selon les étapes suivantes:

1. On estime les deux distributions
$$\mathcal{N}(\mu_{\text{cli}}, \sigma_{\text{cli}}) = f_{\text{cli}}(.) \text{ et } \mathcal{N}(\mu_{\text{imp}}, \sigma_{\text{imp}}) = f_{\text{imp}}(.) \tag{6.1}$$
En calculant leurs moyennes et écart-types à partir des scores de sortie calculés sur un ensemble de données provenant de NIST97.
2. On estime les intégrales de ces distributions:
$$\int_{-\infty}^{x} f_{\text{cli}}(\xi)d\xi = \mathcal{F}_{\text{cli}}(x) \text{ et } \int_{-\infty}^{x} f_{\text{imp}}(\xi)d\xi = \mathcal{F}_{\text{imp}}(x) \tag{6.2}$$
3. On décide de placer le seuil Θ de référence au minimum de la fonction de coût (voir l'équation 1.27 et [Martin, 1997]), qui peut être exprimée comme:
$$c_f = c_{fr} \cdot P(C) \cdot [1 - \mathcal{F}_{\text{imp}}(x)] + c_{fa} \cdot P(\overline{C}) \cdot \mathcal{F}_{\text{cli}}(x) \tag{6.3}$$
Et son minimum
$$\min(c_f) = \frac{d(c_f(x))}{dx} = 0$$
4. Comme on veut que la confiance soit nulle en Θ, on décale la courbe $[(1 - \mathcal{F}_{\text{imp}}(x))]$ pour que
$$\mathcal{F}_{\text{cli}}(x) = 1 - \mathcal{F}_{\text{imp}}(x - a)$$
5. Au signe près, la confiance pour un score x est
$$(1 - (\mathcal{F}_{\text{imp}}(x) + a)) - \mathcal{F}_{\text{cli}}(x) \text{ avec } \gamma = \mathcal{F}_{\text{cli}}(x) - \left[1 - (\mathcal{F}_{\text{imp}}(x - a))\right]$$
La nouvelle courbe $[(\mathcal{F}_{\text{imp}}(x - a))]$ est notée $[(\mathcal{F}'_{\text{imp}}(x)]$.

L'algorithme de Dempster en clair

Les notions et notations introduites par Shafer [1976] gardant un coté hermétique, nous les retraduisons ici, adaptées au problème de décision binaire:

- On considère la valeur de confiance γ_i donnée par la décision d'une méthode i et son complément ($\overline{\gamma_i} = 1 - \gamma_i$).
- On fait varier les valeurs de confiance dans l'intervalle $[0, 1]$ pour la décision d'acceptation et pour celle de rejet.
- Si l'on suppose que $\gamma_i^O, i = \{1, \cdots, O\}$ sont toutes les méthodes qui décident d'accepter et $\gamma_j^N, j = \{1, \cdots, N\}$ sont toutes les méthodes qui décident de refuser, on peut estimer les 2 valeurs suivantes:

$$M_0 = \prod_{i=1}^{O} \gamma_i^O \cdot \prod_{j=1}^{N} \overline{\gamma_j}^N \tag{6.4}$$

$$M_1 = \prod_{i=1}^{O} \overline{\gamma_i}^O \cdot \left[\sum_{j=1}^{N} \left(\gamma_j{}^N \cdot \prod_{k=0}^{k=j-1} \overline{\gamma_k}^N \right) \right] \quad , \quad \overline{\gamma_0}^N = 1 \tag{6.5}$$

– Si $M_0 > M_1$ on accepte, sinon on refuse la séquence.
– Dans le cas où l'on a 2 méthodes seulement, et qu'elles sont en désaccord, nous avons à estimer l'inégalité suivante:

$$\gamma^0 \cdot \overline{\gamma^N} \underset{reject}{\overset{accept}{\gtrless}} \gamma^N \cdot \overline{\gamma^O} \tag{6.6}$$

Résultats

Les résultats que chaque participant doit présenter aux évaluations NIST se divisent en plusieurs catégories, selon les conditions d'entraînement des modèles et la durée des échantillons de test [Martin, 1997, Martin, 1998a]. La fusion que nous avons opérée ici ne porte que sur une des 9 conditions de test possibles, à savoir la condition 10 secondes. Nous montrons dans la figure 6.4 une courbe COR-NIST [Martin *et al.*, 1997] qui indique que la fusion des décisions des 2 systèmes est meilleure que le meilleur des 2 systèmes.

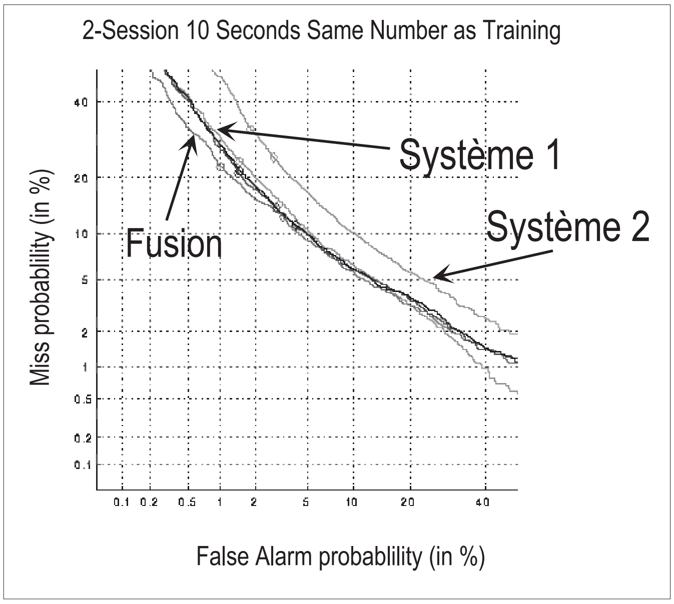

FIG. 6.4 – *Résultat de la fusion dans les évaluations NIST98. On constate que la fusion est meilleure que la meilleure des 2 méthodes.*

Fusion multi-niveaux

Il est évidemment possible de fusionner des décisions à plusieurs niveaux différents simultanément. C'est ce qui se passe par exemple dans le cas du système LIA où une fusion des décisions est opérée au niveau des classificateurs (fusion d'éléments), puis au niveau des algorithmes.

6.3 Conclusion sur la fusion

La fusion de méthodes apporte une amélioration certaine des performances d'un système. En effet, nous arrivons dans la plupart des cas à produire des résultats meilleurs que la meilleure des entités fusionnées. Cependant, une fois ces résultats atteints, il n'est plus possible de continuer à améliorer le résultat global sans passer par une amélioration des performances de chacune des entités, que ce soit au niveau de sa prise de décision (estimation des seuils *a priori*) ou de ses qualités intrinsèques (pouvoir séparateur, calculé avec un seuil *a posteriori*).

Conclusion

Nous avons débuté cette thèse par un chapitre introductif expliquant les caractéristiques du signal de parole ainsi que les outils utilisés pour effectuer la reconnaissance automatique du locuteur. Puis, nous avons décrit au chapitre 2 la conception, la mise au point et le fonctionnement d'un démonstrateur effectuant la reconnaissance de locuteurs. Des carences et des erreurs constatées dans cette application, nous tirons des constatations permettant de réévaluer les paramètres qui caractérisent un locuteur et de reconsidérer plusieurs éléments de la chaîne de reconnaissance automatique du locuteur. Nous proposons des solutions originales pour étudier quels sont les paramètres caractéristiques d'un locuteur qui définissent son identité, ce qui nous a conduit à étudier les phénomènes d'imposture. Puis, nous nous sommes intéressés à la modélisation des paramètres en proposant trois solutions: une correction du calcul du point de fonctionnement des modèles stochastiques existants, une modélisation robuste même avec peu de données d'entraînement ou une correction par fusion de décisions.

Pour déterminer quels sont les paramètres caractéristiques d'un locuteur, nous les avons extraits du signal de parole en utilisant un modèle harmoniques plus bruit (H+N). L'analyse et la resynthèse des parties harmoniques et bruit nous ont permis de découvrir dans le chapitre 3 quelles sont les parties du signal dépendantes du locuteur plutôt que de la parole. Nous avons ainsi découvert que l'information discriminante d'un locuteur se trouve dans la partie résiduelle après soustraction des harmoniques et que le résiduel de la soustraction du signal par l'analyse/synthèse H+N restait encore caractéristique du locuteur. Puis nous avons montré que la décomposition du signal de parole en harmoniques et bruit permet de reconnaître les parties du signal sur lesquelles intervenir pour modifier l'identité de la voix d'un locuteur ou tout au moins l'identité telle qu'elle est perçue par un système automatique. Nous avons également présenté une possibilité de se rendre partiellement résistant à ces transformations en supprimant du signal de parole les harmoniques re-synthétisées. Cette démarche a permis de découvrir que, malgré que l'on ait retiré du signal de parole presque toute l'information qui permet à un être humain de reconnaître le signal prononcé et d'en identifier le propriétaire, le système automatique est encore capable de le déterminer avec une bonne fiabilité. Ce prétraitement est intéressant car il permet au système de reconnaissance de se rendre plus robuste à des tentatives d'imposture par transformations du spectre. Nous avons également montré les limites de la protection contre l'imposture lorsqu'on possède des enregistrements de la parole du locuteur client, sans même qu'ils contiennent le mot de passe du client.

La modélisation et la mémorisation des caractéristiques d'un locuteur nécessitent l'utilisation d'algorithmes capables de capturer des paramètres dont on ne connaît pas *a priori* les caractéristiques. C'est pourquoi nous utilisons des modèles estimés à partir des données. Toutefois, les modèles performants nécessitent la détermination de nombreux paramètres et donc consomment beaucoup de données et de temps de calcul. Le chapitre 4 a montré qu'une bonne décomposition du problème de la reconnaissance du locuteur lorsqu'on ne possède que peu de données permet d'obtenir de meilleures performances, en exploitation, que les systèmes à l'état de l'art. La décomposition du problème en tâches simples (classificateurs binaires) mais nombreuses, permet de sélectionner les tâches qui ont le meilleur pouvoir discriminant. De plus, les arbres de décision, utilisés ici comme classificateurs binaires, ont permis d'analyser plus en détails quels sont les coefficients LPCC qui portent le plus d'information discriminante (au sens d'un critère d'entropie) selon le vocabulaire ou les couples clients/anti-clients utilisés.

Un paramètre très souvent négligé lorsqu'on met au point une application de reconnaissance du locuteur est la détermination *a priori* du point de fonctionnement du système. Celui-ci est influencé par de nombreux paramètres, dont les défauts de modélisation et les variabilités intra- et inter-locuteurs. Nous montrons au chapitre 5 que, lorsque les algorithmes utilisés sont basés sur une modélisation stochastique, les distributions des logarithmes du rapport de vraisemblances (LLR) des clients et des imposteurs peuvent être approximées par des modèles gaussiens. Il suffit alors de prendre en compte le fait que les tests LLR deviennent dépendant du locuteur et du nombre de vecteurs utilisés pour les calculer. Nous montrons également qu'il y a un intérêt, lors de la segmentation du signal par un système de reconnaissance automatique de la parole, à supprimer les valeurs de vraisemblances locales trop faibles, qui indiquent des erreurs de segmentation du signal.

La compensation des erreurs de modélisation peut être effectuée par la fusion des décisions de plusieurs algorithmes. C'est ce qu'a montré le chapitre 2, prouvant que l'on peut améliorer les performances des modèles de Markov cachés par fusion de décisions, bien que ceux-ci regroupent les qualités des systèmes statistiques du 2ème ordre et celles de la programmation dynamique. Dans le chapitre 6, nous développons cette idée en montrant qu'il existe plusieurs niveaux de fusion et quel profit on peut espérer en retirer.

Bien que plusieurs éléments de la chaîne de reconnaissance automatique du locuteur aient été améliorés au cours de cette thèse, on peut y distinguer deux grandes articulations. La première est liée à l'imposture générée par transformation du locuteur et comment rendre le système robuste à celle-ci. La seconde est la décomposition du problème de modélisation et sa recombinaison en fusionnant les résultats d'experts.

Perspectives

Identité d'un locuteur

– Constatons tout d'abord qu'aucune des expériences d'imposture effectuées jusqu'à maintenant ne résiste à l'analyse humaine. Il reste donc des progrès à effectuer pour que celles-ci deviennent crédibles à l'oreille humaine.
– L'analyse/synthèse de la parole nous a permis de constater que l'information permettant de discriminer des locuteurs est contenue principalement dans le résidu du modèle H+N, ce

qui ouvre une perspective intéressante dans le stockage des données des locuteurs, puisque seul ce résiduel devrait être conservé.

– En ce qui concerne la transformation de l'identité d'un locuteur vue par un système automatique, les expériences de décomposition de la parole par analyse/synthèse peuvent être étendues en essayant d'extraire de la partie bruit des paramètres plus aisément transformables. Il est évident que la modélisation du signal de parole comme une partie bruit et une partie harmonique séparée est insuffisante pour les transformations que nous désirons effectuer; il faudrait pour cela étendre le modèle. La transformation mot à mot telle qu'effectuée ici est un premier pas vers une transformation plus généralisée qui pourrait s'effectuer phonème par phonème. Les essais que nous avons effectués dans ce sens ne donnent, pour l'instant, pas de résultats probants, parce que nos modèles HMM de phonèmes ne sont pas de bonne qualité et que, très probablement, les erreurs dues au contexte deviennent importantes surtout pour les phonèmes courts. Cependant, ce principe de transformation est établi et va évoluer avec la qualité des systèmes d'analyse/synthèse de la parole.
– Nous avons vu qu'il est possible de se rendre moins sensible aux transformations spectrales en supprimant la partie harmonique aisément modifiable. Cette perspective est intéressante car elle démontre bien que l'identité d'un locuteur se trouve dans la partie non modélisée et qu'une étude plus approfondie du résiduel du spectre s'avère nécessaire et doit nous permettre de trouver un nouveau jeu de coefficients mieux adaptés à la tâche de reconnaissance de locuteurs.
– Afin de compenser les risques d'imposture criminelle dans les systèmes de reconnaissance de la parole, il est certainement plus judicieux d'utiliser des systèmes multi-modaux se basant sur plusieurs paramètres biométriques simultanés.

Modélisation

– La décomposition en éléments simples d'un système de reconnaissance du locuteur en est seulement à ses débuts, la taille des unités à choisir (mots, phonèmes, classes phonétiques) est encore à analyser. Les anti-classes d'un locuteur doivent être aussi analysées encore plus en détails, par exemple en y ajoutant une anti-classe représentant le monde. Cette décomposition est une alternative aux algorithmes statistiques qui modélisent tous les coefficients y compris ceux qui ne contribuent en rien à la classification.
– L'utilisation de classificateurs correcteurs devrait, en outre, augmenter encore les performances des arbres de décision. Quoi qu'il en soit, d'autres classificateurs peuvent être utilisés si le nombre de données est suffisant, et il serait tout à fait possible d'utiliser d'autres algorithmes classificateurs (HMM, RNA, etc. . .) comme classificateurs binaires.
– Les corrections que nous avons effectuées pour tenir compte des erreurs de modélisation lorsqu'on utilise des modèles stochastiques, invitent à étudier plus en détails quelles sont les variabilités intra- et inter-locuteurs qui influencent ces modèles. De même, peut-être que l'étude de l'évolution de ces erreurs en utilisant plus ou moins de données sur plus ou moins de temps pour un locuteur nous permettrait de mieux en fixer les bornes.

Annexe A

Le système de référence HMM

A.1 Introduction

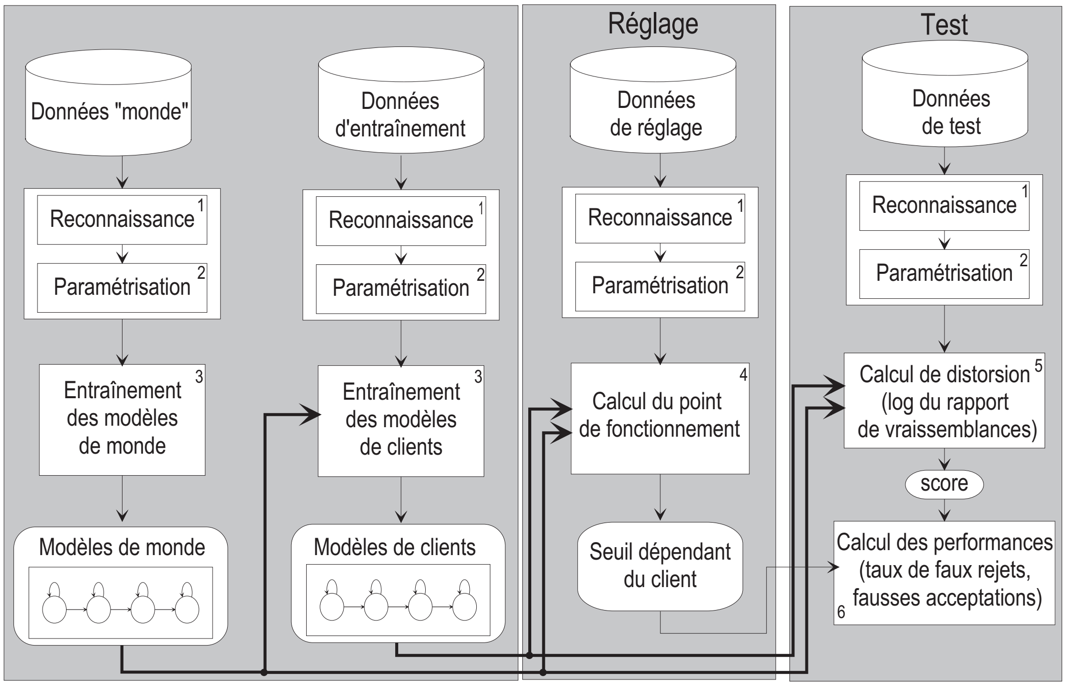

FIG. A.1 – *Schéma bloc du système de référence.*

La complexité de la parole humaine nécessite des systèmes de reconnaissance automatique de la parole et du locuteur comportant un grand nombre de paramètres de réglage. Chacun de ces paramètres a une influence plus ou moins grande sur les résultats du système global. Comme il n'est évidemment pas envisageable, et d'ailleurs peu intéressant, de donner des résultats pour tous les paramètres de réglage possibles, nos résultats sont produits par un système à l'état de l'art et décrits ci-après. Les données utilisées ont aussi une influence sur les résultats. Elles sont normalement divisées en *données d'entraînement*, *données de réglage* et *données de test*. Le détail des bases de données utilisées lors des différentes étapes de l'utilisation du système de référence est exposé dans l'annexe B. Le système de référence travaille en mode **dépendant du texte**, ce qui implique une reconnaissance préalable de la parole. Ce système de reconnaissance automatique de la parole (SRAP) sera décrit ici dans la section A.2.

Comme entrée de notre système de référence, nous ne considérerons que les phrases parfaitement reconnues par le SRAP. Notre système de référence fonctionne en mode de **vérification du locuteur**. Cela implique que nous ayons identifié le locuteur au préalable. Dans notre cas, le locuteur prononce un code personnel (PIN: Personnal Identification Number) de 7 ou 10 chiffres, ce code est ensuite reconnu automatiquement par le SRAP puis utilisé mot à mot pour effectuer la vérification de l'identité du locuteur.

Le système de référence se compose de 6 modules (voir figure A.1):

1. Reconnaissance/segmentation de la parole.
2. Paramétrisation.
3. Entraînement des modèles de vérification.
4. Détermination du point de fonctionnement.
5. Test d'hypothèses (calcul des scores).
6. Calcul des performances du système.

A.2 Reconnaissance de la parole

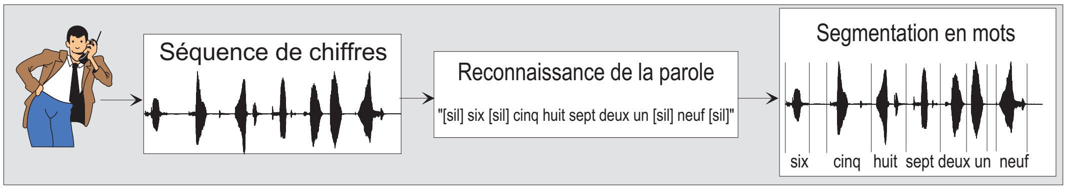

FIG. A.2 – *Etage de reconnaissance de la parole.*

Afin de contrôler ce que dit le client de l'application, une étape de reconnaissance automatique du message prononcé est effectuée. Comme il ne s'agit ici que de séquences de chiffres, une grammaire contrainte et des modèles de mots peuvent être utilisés. Le système de reconnaissance utilise une paramétrisation qui lui est propre (12 MFCC, énergie, Δ et $\Delta\Delta$). Lorsque nous utilisons les séquences du code personnel de chaque locuteur, un code correcteur (RS code: Reed Solomon code) et la limitation du nombre de codes autorisés permettent de contraindre fortement la grammaire et d'éliminer les codes mal reconnus. Ainsi, toutes les phrases passées à l'étage de paramétrisation sont sensées être correctes, ce qui permet de découpler les problèmes de reconnaissance de la parole et de vérification du locuteur. Dans le cas des séquences de 10 chiffres, nous utilisons une grammaire contraignant le nombre de chiffres possibles et contrôlons qu'elle correspond bien à ce qui devait être dit. La figure A.2 indique les différentes étapes du processus de reconnaissance. La parole ainsi segmentée sera fournie mot par mot à l'étage de paramétrisation.

A.3 Paramétrisation de la parole

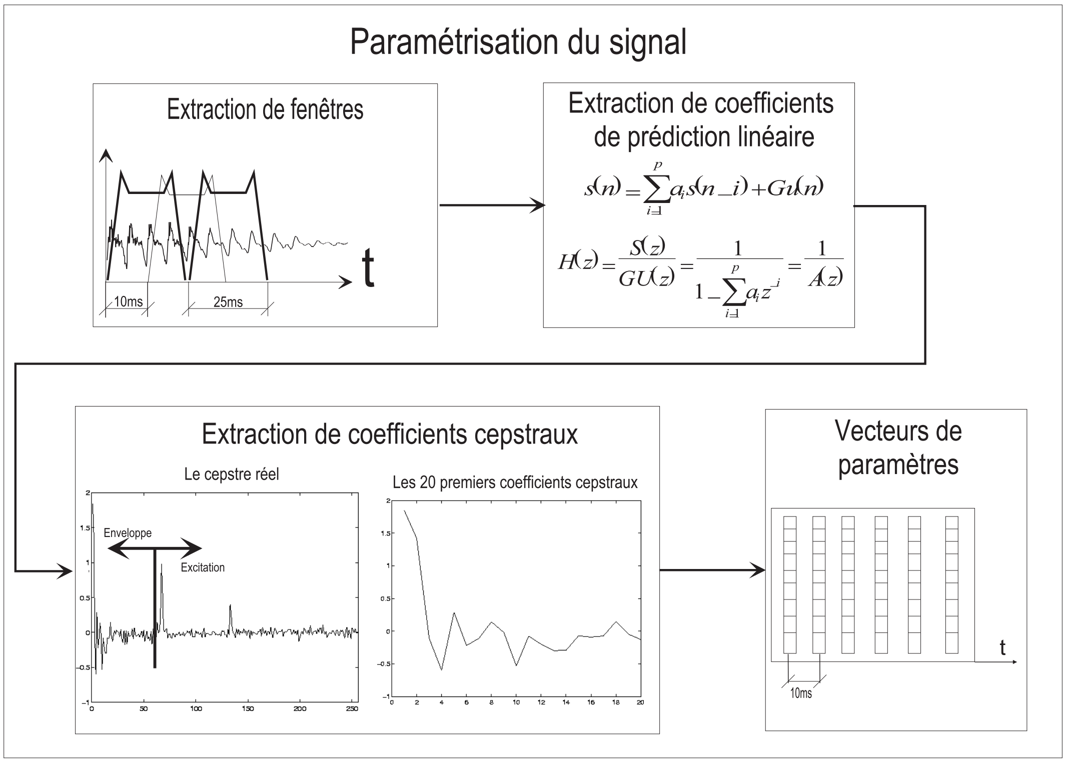

FIG. A.3 – *Etage de paramétrisation de la parole*

Cette étape consiste à extraire du signal de parole les caractéristiques du locuteur qui vont permettre de le reconnaître. Le système de référence utilise ici une paramétrisation LPCC (Linear Predicting Cepstral Coefficients) connue pour donner de bons résultats en vérification du locuteur (voir par exemple [Furui, 1981a, Furui, 1994]). La figure A.3 indique le processus d'extraction des paramètres: chaque 10 [ms] , une fenêtre de 25 [ms] est extraite du signal. A cette fenêtre, on applique ensuite une fenêtre de Hamming, puis une pré-accentuation de 0.94 est appliquée à chaque échantillon de la fenêtre, enfin on en extrait les 13 premiers coefficients cepstraux (voir la section 1.2.2). Afin de récupérer la dynamique des événements acoustiques, les dérivées et les accélérations des coefficients cepstraux sont aussi calculés. Cet étage de paramétrisation se retrouve à chaque étape du système (entraînement des modèles, réglage des seuils, test). Pour certaines expériences, seuls 12 coefficients cepstraux et leurs dérivées sont utilisés afin de réduire les coûts de calcul de l'étage de modélisation (comme par exemple dans le chapitre 4).

A.4 Calcul des scores

Comme nous l'avons vu dans l'introduction (voir la section 1.2.4), il est possible, dans le cas des modèles statistiques, de prendre une décision de type Bayesien en calculant le logarithme du rapport de vraisemblances d'un modèle de locuteur et d'un modèle de non-locuteur (approximé par un modèle de *monde*). Rappelons que la vérification du locuteur implique de prendre une décision d'accepter ou de rejeter un segment de parole (noté ici O_t la suite des vecteurs issus de l'étage de paramétrisation) comme appartenant au client que l'on cherche à tester. Reprenons l'équation de décision (1.29):

$$LLR(M_C, M_W, O_t) = \log \mathscr{L}(O_t, M_C) - \log \mathscr{L}(O_t, M_W) \underset{reject}{\overset{accept}{\gtrless}} \log \left(\frac{P(M_W)}{P(M_C)} \cdot \frac{c_{fa}}{c_{fr}} \right) = \Theta$$

Les distributions statistiques de ces LLR calculées sur des séquences de locuteurs et d'imposteurs seront les entités que nous manipulerons pour calculer soit un point de fonctionnement (voir la section A.6), soit les performances intrinsèques du système (voir la section A.7).

A.5 Entraînement des modèles de vérification

A.5.1 Le modèle de Markov caché

Les propriétés statistiques des modèles de Markov cachés (en anglais: Hidden Markov Models ou *HMM*) [Rosenberg *et al.*, 1991] en font une des modélisation les plus efficaces actuellement en reconnaissance du locuteur. Les HMM permettent de modéliser des processus stochastiques variant dans le temps [Scharf, 1991, Rabiner et Juang, 1993]. Pour cela, ils combinent les propriétés à la fois des distributions de probabilités et une machine à états (voir la section 1.2.3).

A.5.2 Entraînement d'un HMM

Les paramètres (gaussiennes et transitions) sont estimés à partir des **données d'entraînement**. Les modèles ont tous la même structure HMM gauche-droite constituée d'un état par phonème et un état par transition entre phonèmes [Gravier, 1995]. Deux sortes de modèles HMM sont créés pour chaque chiffre du code personnel du client:

– *Un modèle du monde*, créé à partir d'une base de données (p. ex. Polyphone, annexe B.4) comportant un grand nombre de locuteurs dont on a extrait plusieurs ($\approx$ 300 répétitions) de chaque mots du vocabulaire que l'on va utiliser pour l'application considérée. Ce modèle est indépendant du locuteur donc identique pour tous les clients. Les paramètres de ce modèle sont estimés par un entraînement standard en **reconnaissance de la parole**.
– *Un modèle du client*, qui utilise comme paramètres initiaux ceux du modèle du monde ou alors des gaussiennes à moyennes nulles et à variances unitaires ("flat start" en anglais). La ré-estimation des paramètres pour chaque locuteur se fait en utilisant l'algorithme de Baum-Welsh dont on réestime les paramètres pour chaque locuteur sur ses données d'entraînement.

Afin de garder une structure temporelle aux données (contrainte issue des applications), les données d'entraînement seront issues de la première session d'enregistrement effectuée par chaque client.

A.6 Calcul du point de fonctionnement du système

Le point de fonctionnement du système est déterminé par le second terme de l'équation (1.29):

$$\log\left(\frac{P(M_W)}{P(M_C)} \cdot \frac{c_{fa}}{c_{fr}}\right) = \Theta$$

Si nous supposons maintenant que, pour une application donnée, la probabilité que le segment de parole que l'on veut tester provient d'un client ou d'un imposteur de manière identique, les probabilités *a priori* du modèle du monde $P(M_W)$ et du modèle du client $P(M_C)$ sont égales. Si de plus, on admet que les coûts d'une erreur de faux rejet c_{fr} et de fausse acceptation c_{fa} sont identiques, le terme précédent devient:

$$\log(1) = 0 = \Theta$$

Et qui devrait être totalement indépendant du locuteur. Mais comme l'a montré le chapitre 5, Θ dépend du locuteur et du nombre de vecteurs de paramètres utilisés pour estimer le LLR. Pour ce faire, nous utilisons des données du locuteur qui sont différentes des données d'entraînement et de test. Typiquement, nous utilisons la deuxième session d'enregistrement du client pour le faire. En ce qui concerne les données d'imposture, des imposteurs différents de ceux de test sont utilisés. Le point de fonctionnement ainsi déterminé est aussi appelé **seuil de décision *a priori***.

A.6.1 Normalisation

Nous mesurons aussi le système sur toute sa plage de fonctionnement lorsque cela s'avère nécessaire, en calculant une courbe COR [Oglesby, 1994] (voir la section 1.2.4).

Cependant, comme le point de fonctionnement du système est fixé indépendamment pour chaque locuteur, une correction des scores de sortie est nécessaire pour calculer des courbes COR sur tout l'ensemble des locuteurs. Cette correction consiste simplement à soustraire la valeur du point de fonctionnement des scores de sortie du système. Ces valeurs normalisées sont indiquées dans les graphiques comme Norm-logLLR.

A.7 Test des performances du système

Pour tester les performances du système 2 mesures sont utilisées:

- Les taux d'erreur avec seuil *a priori* qui montre les performances du système **en fonctionnement**. Ce seuil est déterminé comme à la section A.6.
- Les taux d'erreur avec seuil *a posteriori*. Dans ce cas nous effectuons tous les tests, puis le point d'égale erreur (EER) est déterminé sur les distributions des scores des clients et des

imposteurs. Nous pouvons ainsi déterminer les performances **intrinsèques** du système ou le pouvoir discriminant de celui-ci.
- On complète, si possible, les mesures par une courbe COR estimée sur les données de tests.

Voir la section 1.2.4 pour plus de détails sur le calcul des scores.

Annexe B

Bases de données utilisées

Nous décrivons ici les différentes bases de données utilisées pour les expériences décrites dans ce document.

B.1 Polycode

Conditions d'enregistrement

Polycode est une base de données téléphonique qui a été enregistrée sur une plate-forme SUN-ISDN en format a-law et converti en PCM 16 bits. Les enregistrements ont été effectués, pour la majorité, en communication locale à partir du même téléphone. Tous les téléphones sont de même type. Les locuteurs parlent français et sont de langue maternelle française pour la plupart.

Contenu d'une session

Une session est constituée des prompts suivants:

1. Donnez vos nom, prénom, adresse, date et lieu de naissance.
2. Epelez lettre par lettre votre nom et prénom.
3. Prononcez votre numéro de client <Code client 7 chiffres> chiffre par chiffre.
4. Prononcez 0 1 2 3 4 5 6 7 8 9 chiffre par chiffre.
5. Prononcez votre numéro de client <Code client 7 chiffres> chiffre par chiffre.
6. Prononcez 8 3 9 4 6 1 7 2 0 5 chiffre par chiffre.
7. Prononcez votre numéro de client <Code client 7 chiffres> chiffre par chiffre.
8. Prononcez 5 0 6 9 2 8 1 3 7 4 chiffre par chiffre.
9. Prononcez votre numéro de client <Code client 7 chiffres> chiffre par chiffre.
10. Prononcez 9 8 7 6 5 4 3 2 1 0 chiffre par chiffre.
11. Prononcez votre numéro de client <Code client 7 chiffres> chiffre par chiffre.
12. Veuillez répétez deux fois "Sésame, ouvre-toi".

Les sessions sont identiques pour chacun des clients, mais différentes de client en client.

Vérification et annotation de la base

La base de données a été vérifiée par un système de reconnaissance automatique de la parole. Les prompts (1, 2 et 12) n'ont fait l'objet d'aucun contrôle. Les prompts 3, 5, 7, 9, 11 ont été contrôlés grâce à la composition du code client constitué d'un RS-code (Reed Solomon) 4-3, permettant de détecter 2 erreurs. La grammaire utilisée contraint le nombre de mots possibles dans la séquence. Pour les séquences de 10 chiffres (prompts 4, 6, 8), seule une contrainte sur le nombre de chiffres possibles a été utilisée. Dans tous les cas, seules les séquences parfaitement reconnues sont utilisées. La reconnaissance de la parole fournit également une segmentation temporelle en mots.

Contenu (sessions, locuteurs)

La base de données Polycode est constituée de 42 locuteurs ayant enregistré entre une et 63 sessions sur une période de six mois.

Sous-ensembles de la base

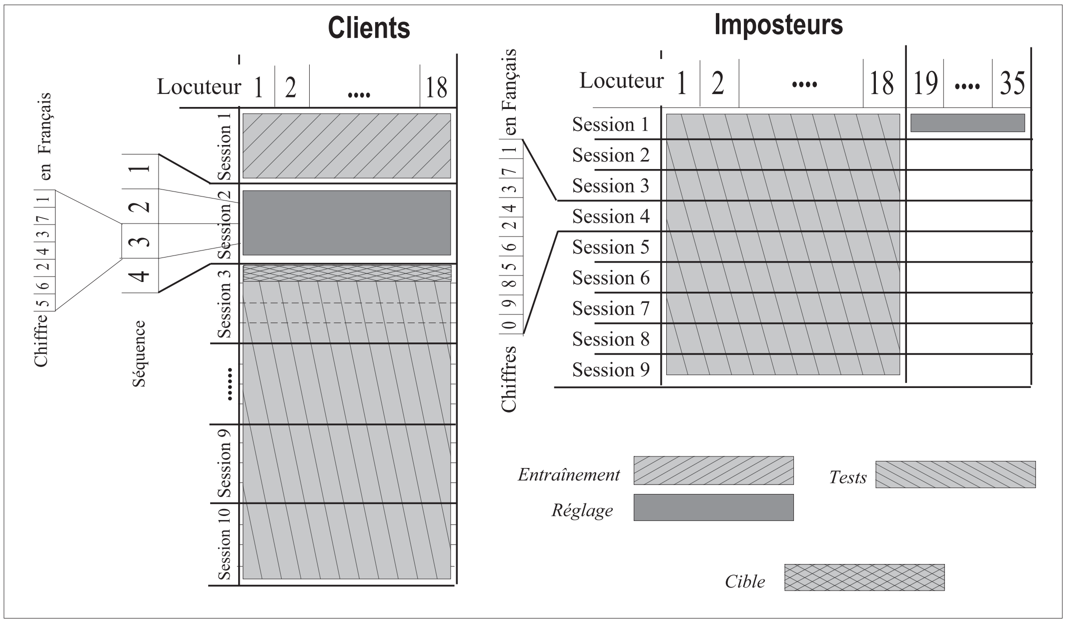

FIG. B.1 – *Sous-ensembles de données utilisés dans Polycode.*

Ensemble des clients

L'ensemble des clients (voir la figure B.1) est composé de 18 locuteurs (13 hommes, 5 femmes) qui ont au moins 10 sessions d'enregistrement où 4 codes clients ont été correctement reconnus.

Pour des raisons d'uniformité des résultats, nous utilisons seulement 4 codes clients sur 5 pour toutes les sessions client.

Pour chaque client trois types de données sont utilisées:

1. Les données d'**entraînement**, composées d'une session de 4 séquences de code client.
2. Les données de **réglage**, composées d'une session de 4 séquences de code client. Cette session a été enregistrée après la session d'entraînement.
3. Les données de **test** qui comportent 8 sessions de 4 séquences de code client pour chaque client. Les données de test ont été enregistrées après la session de réglage.

Ensemble des imposteurs

L'ensemble des imposteurs est constitué des mêmes 18 locuteurs que les clients (13 hommes, 5 femmes, voir la figure B.1). Comme chaque locuteur de la base a prononcé des codes clients différents mais des séquences de 10 chiffres identiques, nous utilisons les séquences de 10 chiffres comme base pour les tests d'imposture. Pour chaque test d'imposture, on reconstruit le code client par extraction des chiffres de la séquence de 10 chiffres. Le nombre de séquences utilisées par imposteur est d'environ 10, ce qui représente environ 170 tests d'impostures par client.

Ensemble de réglage *(Polycode-γ)*

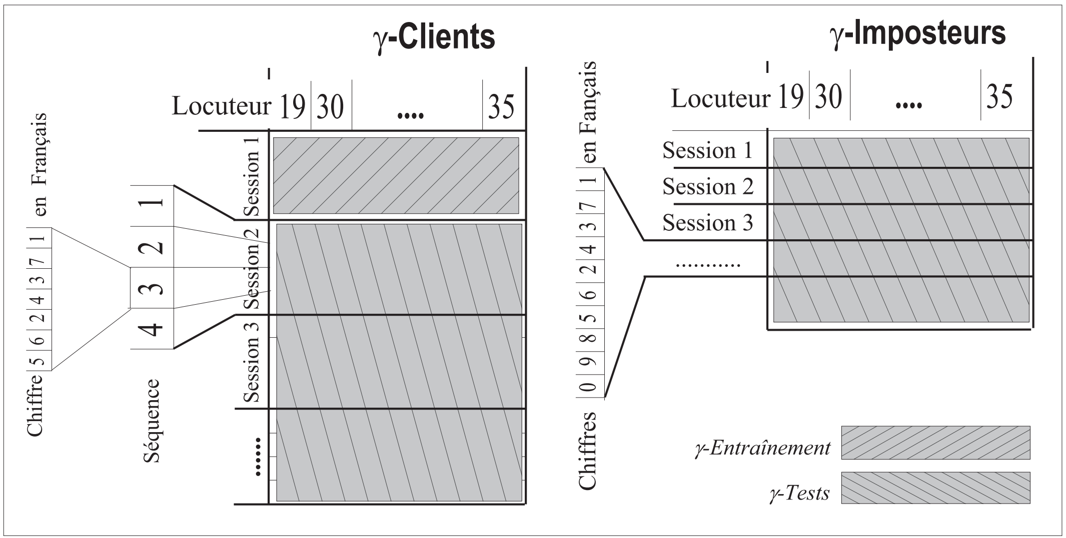

FIG. B.2 – *Sous-ensembles de données de réglage de Polycode (= Polycode-γ).*

Polycode-γ est un morceau de Polycode qui a servi à calculer certaines constantes (par exemple les constantes de Furui (voir la section 2.4.2) ou des seuils indépendant des locuteurs utilisés pour les expériences, comme le seuil de décision global (voir la section 2.4.1). Certains locuteurs ont été utilisé comme imposteurs pour déterminer des seuils dépendant du client. Les clients et les imposteurs de Polycode-γ (voir la figure B.2) sont au nombre de 17 (12 hommes, 5 femmes). Le nombre de sessions de tests pour chaque client est variable, de 1 à 7, et le nombre de tests d'imposture varie lui aussi entre 1 et 10.

Variations

Certaines expériences particulières nous ont obligés à modifier parfois ce protocole de base. C'est le cas par exemple du chapitre 4 où nous avons supprimé la première session de test pour utiliser la première séquence de la session comme séquence cible des transformations. C'est le cas également pour le chapitre 5 où nous avons utilisé 2 sessions pour l'entraînement et de une à trois sessions de réglage (voir la section 5.6).

Remarques sur Polycode

Polycode est une base qui contient peu de biais dû au canal de transmission. Les appels sont numériques, locaux et tous les téléphones utilisés de la même qualité.

B.2 Polycost

Conditions d'enregistrement

Polycost a été enregistrée sur une plate-forme Sun X-Tel ISDN en a-law. La plupart des enregistrements sont issus de lignes téléphoniques internationales provenant de 13 pays différents. 80% des locuteurs ont utilisé le même appareil téléphonique. Les enregistrements sont en anglais parlé par des Européens de langues maternelles diverses. La table B.1 résume la provenance des appels.

Pays	Nombre d'appels
France-FR	205
Switzerland-CH	169
Netherlands-NL	107
Ireland-IR	103
Spain-SP	102
United-Kingdom-UK	99
Italy-IT	99
Sweden-SE	98
Denmark-DK	97
Turkey-TR	94
Belgium-BE	59
Portugal-PT	44
Lituany-LI	9
Total	1285

TAB. B.1 – *Répartition géographique de la provenance des appels dans Polycost.*

Contenu d'une session

Une session est constituée des prompts suivants:

1. Please type your 7 digits code on your keypad (DTMF detection).
2. Say your client code 0, 1, 2, 3, 4, 5, 6, digit by digit.
3. Say your name, christian name, sex (female/male), town, country and mother tongue. (speak in your MOTHER TONGUE)
4. Say 0 1 2 3 4 5 6 7 8 9, digit by digit.
5. Say your client code 0, 1, 2, 3, 4, 5, 6, digit by digit.
6. Say 8 3 9 4 6 1 7 2 0 5, digit by digit.
7. Say the sentence "Joe took father's green shoe bench out".
8. Say 5 0 6 9 2 8 1 3 7 4, digit by digit.
9. Say your client code 0, 1, 2, 3, 4, 5, 6, digit by digit.
10. Say the sentence "He eats several light tacos".

11. Say 9 8 7 6 5 4 3 2 1 0, digit by digit.
12. Say what you have done today or/and describe your environment
    (free speech in MOTHER TONGUE with different text at each call).
13. Say 1 0 2 9 3 8 4 7 5 6, digit by digit.
14. Say your international phone number (for example: 00 41 27 721 77 26).
15. Say your client code 0, 1, 2, 3, 4, 5, 6, digit by digit.

Les sessions sont identiques pour chacun des clients, mais différentes de client en client.

Vérification et annotation de la base

Une première sélection automatique des enregistrements s'est faite en effectuant la reconnaissance du code personnel de chaque session par code DTMF (premier prompt). Par la suite, une reconnaissance automatique de la parole a été effectuée sur les séquences de 10 chiffres et sur les codes clients (voir pour les détails [Polycost-1,1996]). Finalement,des annotateurs ont vérifié tous les enregistrements (voir [Polycost-2,1996]).

Contenu (sessions, locuteurs)

Le sous-ensemble de la base de données Polycost utilisée pour les expériences décrites dans ce document est composée de 134 locuteurs (74 hommes et 60 femmes) issus de 13 pays différents totalisant 1285 sessions d'enregistrement.

Pour des raisons pratiques (disponibilité des enregistrements au moment des tests), nous avons extrait 104 locuteurs enregistrés sur 6 sessions. Chaque session est composée de 4 séquences de 10 chiffres en anglais (tous les chiffres de 0 à 9 prononcés dans un ordre différent pour chaque séquence). Les séquences de 10 chiffres sont identiques pour chaque locuteur et toutes les séquences ont été alignées temporellement chiffre par chiffre en utilisant un système automatique de reconnaissance de parole (voir [Polycost-1,1996]).

Sous-ensembles de la base

Ensemble des clients

L'ensemble des clients est composé de 82 locuteurs (48 hommes et 34 femmes). Pour chaque client, on utilise quatre types de données:

1. Des données d'**entraînement**, composées d'une session de 4 séquences de 10 chiffres.
2. Des données de **réglage**, composées d'une session de 4 séquences de 10 chiffres. Cette session a été enregistrée après la session d'entraînement.
3. Des données de **test** qui comportent 4 sessions de 4 séquences de 10 chiffres pour chaque client. Les sessions de test ont été enregistrées après la session de réglage.

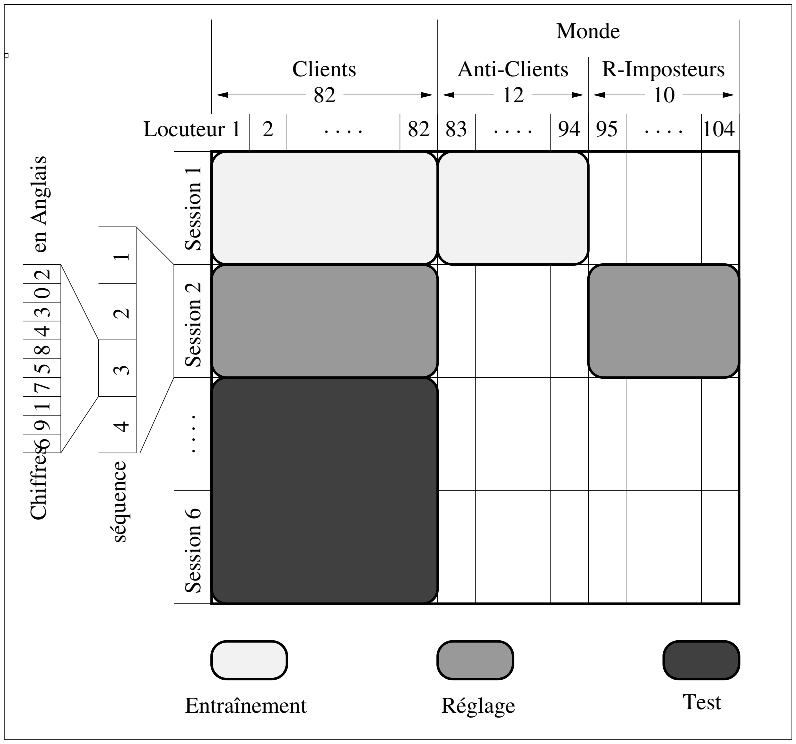

FIG. B.3 – *Répartition des locuteurs de la base de données Polycost utilisée dans ce document.*

Ensemble des imposteurs

L'ensemble des imposteurs est constitué des mêmes 82 locuteurs (voir la figure B.3) que les clients. 4 séquences tirées des sessions de test de chaque client sont utilisées comme imposture sur les autres clients, générant ainsi 324 séquences d'accès d'imposteurs pour chaque client.

Ensemble d'imposteurs complémentaires

Cet ensemble est constitué de 22 locuteurs. Il est utilisé de différents manières (voir la figure B.3):

- Soit il est divisé en 2 sous-populations, les *anti-clients* et les *imposteurs de réglage* (notés **R-Imposteurs** sur la figure B.3). Les anti-clients servent à entraîner les classificateurs du chapitre 4. Les imposteurs de réglage servent à estimer des distributions de scores d'imposture pour calculer des seuils, par exemple pour élaguer les matrices binaires (voir le chapitre 4).
- Soit comme population servant à créer un modèle de *monde*. (**Anti-clients** + **R-Imposteurs**).

Remarques sur Polycost

Les appels enregistrés dans Polycost sont issus, pour la plupart, de communications internationales. L'influence du canal de transmission est donc importante. Nous ne sommes pas en mesure pour l'instant de déterminer dans quelle mesure les variabilités du canal sont impliquées dans l'identification des locuteurs.

B.3 Polyvar

Conditions d'enregistrement

Polyvar a été enregistrée sur une plate-forme analogique et numérique en a-law. Les enregistrements sont composés de locuteurs parlant français (et de langue maternelle française pour la plupart).

Contenu d'une session

Une session contient:

1. 4 séquences de chiffres (1 nombre de 4 chiffres, 1 numéro de carte de crédit, 1 séquence de 6 chiffres et un numéro de téléphone).
2. 24 mots d'application (17 mots touristiques de la région de Martigny, 6 mots provenant d'une liste et un nom de ville).
3. 10 phrases lues.
4. 4 nombres (2 nombres entiers et 2 montants).
5. 2 demandes contenant des dates (1 lue, 1 spontanée).
6. 2 demandes d'heure (1 lue, 1 spontanée).
7. 3 mots épelés.
8. 3 réponses spontanées (questions à propos de l'adresse, de la langue maternelle et du temps).
9. 1 commentaire.
10. 1 demande téléphonique.

Vérification et annotation de la base

La base a été vérifiée par des annotateurs humains. L'annotation est faite au niveau de la phrase, sans segmentation temporelle.

Une des parties de Polyvar que nous utilisons dans cette thèse (les 10 phrases) a ensuite été segmentée phonétiquement par un système de reconnaissance automatique de la parole en mode de reconnaissance forcée (i.e. le contenu phonétique de la phrase est donné au préalable).

Contenu (sessions, locuteurs)

La base de données Polyvar est constituée de 143 locuteurs (85 hommes et 58 femmes). Le sous-ensemble que nous utilisons ici est constitué de 18 locuteurs (12 hommes et 6 femmes). Chaque locuteur a dû prononcer les 17 mots de commande suivants entre 30 et 130 fois:

annulation, Casino, cinéma, concert, Corso, exposition, galerie du Manoir, Gianadda, guide, Louis Moret, manifestation, message, mode d'emploi, musée, précédent, quitter, suivant.

De plus, nous extrayons pour chacun des locuteurs les 10 phrases qu'ils ont prononcées (ce qui donne entre 300 et 1300 phrases) et qu'on segmente phonétiquement (voir le paragraphe précédent).

Sous-ensembles de la base

Ensemble des clients

L'ensemble des clients est constitué de 18 locuteurs pour lesquels on utilise 3 types de données:

1. Les données d'**entraînement**, composées de 5 sessions d'un mot de commande.
2. Les données de **test** composées de 20 à 70 répétitions de chacun des mots de commande.
3. Les données de **concaténation** constituées d'extraits de phrases et recomposant des mots de commande.

Ensemble des imposteurs

L'ensemble des imposteurs est le même que celui des clients, 5 tests d'imposture sont effectués par imposteur sur chaque locuteur, la sélection des sessions d'imposture s'effectuant au hasard sur toutes les sessions de test de chaque locuteur.

B.4 Polyphone

Conditions d'enregistrement

La base de données Polyphone a été enregistrée sur une plate-forme ISDN. Environ 4500 personnes ont dû prononcer **une fois** en français 38 éléments différents composés de chiffres, de nombres, de noms, de mots de commande, de phrases et de phrases spontanées). Pour plus de détails, voir [Chollet *et al.*, 1995].

Vérification et annotation de la base

La base de données Polyphone a été totalement vérifiée et annotée au niveau de la phrase par des annotateurs humains.

Sous-ensembles de la base

Nous avons extrait de cette base de données environ 300 répétitions de chaque chiffre de zéro à neuf. Ces répétitions ont servi à créer les modèles de monde utilisés pour les HMM (voir par exemple le système de référence à l'annexe A).

Annexe C

Vérification du locuteur par réseaux de neurones

C.1 Introduction

De manière à montrer qu'il existe des approches différentes permettant d'obtenir des résultats intéressants en vérification du locuteur indépendante du texte, nous exposons ici un exemple d'architecture à base de réseaux de neurones artificiels. Une partie des travaux présentés dans cette annexe ont été menés conjointement entre l'IDIAP et l'ENST Paris. Le système a été conçu pour participer aux évaluations NIST 1997 (voir [Martin, 1997, Martin, 1998b]).

C.2 Tâche de classification NIST 1997

L'évaluation 1997 proposait une détection (vérification) du locuteur indépendante du texte. La base de données utilisée pour les évaluations est un extrait de Switchboard [Ldc, 1994] (base de données téléphonique). Les données d'entraînement se divisent en trois catégories:

- Une session.
- Un combiné téléphonique (handset en anglais).
- Deux combinés téléphoniques.

La durée de parole disponible pour chaque type d'entraînement est de 1 minute.

L'**ensemble de test** est composé de séquences de durées de 3 [s] , 10 [s] et 30 [s] .

Les segments de test doivent être utilisés avec les trois conditions d'entraînement. Deux types de résultats doivent être donnés:

- Une décision binaire accepté/refusé basée sur un seuil déterminé par la fonction de coût (voir l'équation 1.27) que nous rappelons ici:

$$c_{tot} = c_{fr} \cdot P_C \cdot P_{FR/C} + c_{fa} \cdot P_W \cdot P_{FA/W} \tag{1.27}$$

Les différents paramètres sont fixés aux valeurs suivantes:
$c_{fr} = 10$; $c_{fa} = 1$; $P_C = 0.01$; $P_W = 1 - P_C = 0.99$

– Un score doit être fourni pour chaque segment de test de manière à calculer une courbe COR (voir la section 1.2.4).

C.3 Le système IDIAP

C.3.1 Paramétrisation du système

Nous utilisons une paramétrisation standard de 16 coefficients LPCC avec leurs dérivées et accélérations plus la dérivée et l'accélération de l'énergie. Cependant, en suivant les travaux de D. Charlet [1996] qui montrent que les derniers coefficients LPCC et les coefficients dynamiques suffisent à caractériser un locuteur dans une tâche de vérification indépendante du texte, nous avons utilisé les 8 derniers coefficients LPCC (c9-c16) et tous les coefficients dynamiques, constituant ainsi un vecteur de 42 coefficients.

On extrait des fenêtres de 32 [ms] toutes les 10 [ms] du signal de parole, avec une pré-accentuation de 0.95 et un liftering de 16 (voir la section 1.2.2). On utilise également une CMS (Cepstral Mean Substraction) pour compenser les déviations dues au canal. Cependant, aucune normalisation ne fut effectuée pour compenser les différents types de combinés téléphoniques.

C.3.2 Modélisation utilisée

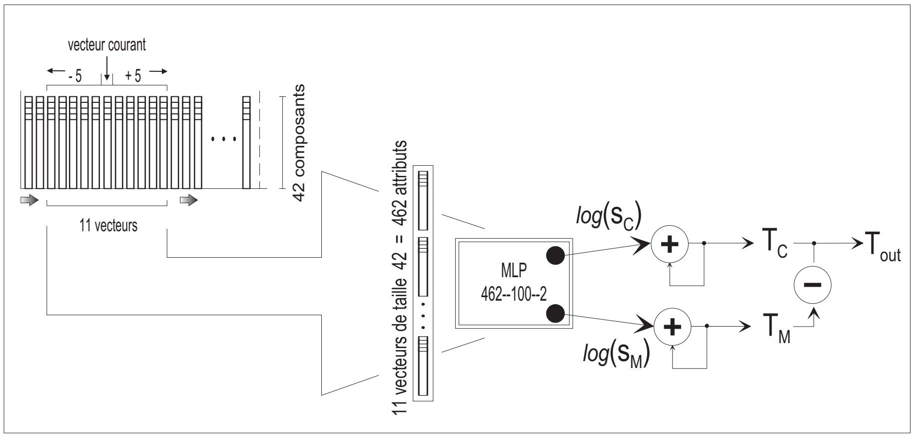

FIG. C.1 – *Le classificateur MLP utilisé pour les évaluations NIST97.*

Nous avons utilisé des réseaux de neurones artificiels inspirés de ce qui s'est fait en reconnaissance de la parole (voir par exemple [Morgan et Bourlard, 1995]). Le réseau, un MLP (Multi Layer Perceptron), est constitué de 462 neurones d'entrée, de 100 neurones sur la couche cachée et de 2 neurones sur la couche de sortie. Les neurones d'entrée correspondent à 11 vecteurs de paramètres consécutifs, de manière à capturer des événements acoustiques de l'ordre de 100 [ms]. Les 2 neurones de la couche de sortie (voir la figure C.1) sont entraînés de manière à ce que le neurone du client donne une valeur de $S_C = 1$ si les vecteurs d'entrée appartiennent au *client* ("target speaker" dans le vocable NIST) ($S_M = 0$) et $S_C = 0$ si les vecteurs appartiennent au *monde* ("non-target speaker") ($S_M = 1$), voir la section 1.2.4 pour la notion de monde.

Durant la phase d'**entraînement**, on crée un MLP par client. On utilise pour la classe *monde* des segments de parole provenant de locuteurs de l'évaluation NIST96 (40 hommes et 40 femmes). La durée totale des segments utilisés pour le modèle de monde est équivalente à celle du client (1 minute).

Durant la phase de **test**, les valeurs de sortie S_C et S_M de chacun des neurones client et monde sont sommées sur les N vecteurs issus du segment de test:

$$T_C = \sum_{i=1}^{N} \log([s_C]_i)\,, \quad T_M = \sum_{i=1}^{N} \log([s_M]_i)$$

On normalise ensuite la valeur de sortie du client par celle du monde:

$$T_{out} = T_C - T_M$$

Sur le même principe que le score LLR (log du rapport de vraisemblances, voir section 1.2.4), voir [Morgan et Bourlard, 1995] pour les justifications théoriques.

C.3.3 Détermination des seuils de décision

Nous avons choisi d'estimer les seuils de décision selon la méthode proposée par Furui [1981a] (voir la section 2.4.2) puisque, lors des tests, nous n'avons pas de données de réglage:

$$th_C = C1 \cdot (\mu_M + \sigma_M) + C2$$

Nous utilisons cependant une version étendue de la détermination de ces seuils (voir [Pierrot, 1998]:

$$th_C = a \cdot \mu_M \sigma_M + b \cdot \mu_M + c \cdot \sigma_M$$

De manière à rendre le seuil indépendant du locuteur, nous lui appliquons une correction:

$$Th'_C = T_{out} - (A \cdot \mu_M \cdot \sigma_M + B \cdot \mu_M + C \cdot \sigma_M)$$

Qui permet de s'adapter à la fonction de coût (voir C.2). Pour calculer les constantes A,B,C, on utilise des données de locuteurs de NIST96 qui servent ici d'imposteurs. Les données des imposteurs sont utilisées pour déterminer les paramètres μ_M et σ_M estimés sur la distribution des scores de sortie $\mathcal{N}(\mu_M, \sigma_M)$ des modèles de client de l'évaluation 1997.

C.4 Résultats

Comme il y a en tout 9 conditions de tests différentes et que de plus NIST calcule aussi des scores sur les données qui ont servi à l'entraînement, il est difficile de donner un classement sans montrer toutes les courbes. Nous avons choisi ici 2 cas qui nous semblaient intéressants, le premier où notre système fonctionne aussi bien que les approches à l'état de l'art avec compensation des distorsions dues aux combinés (figure C.2 à gauche) et un résultat où le système fonctionne moins bien (figure C.2 à droite).

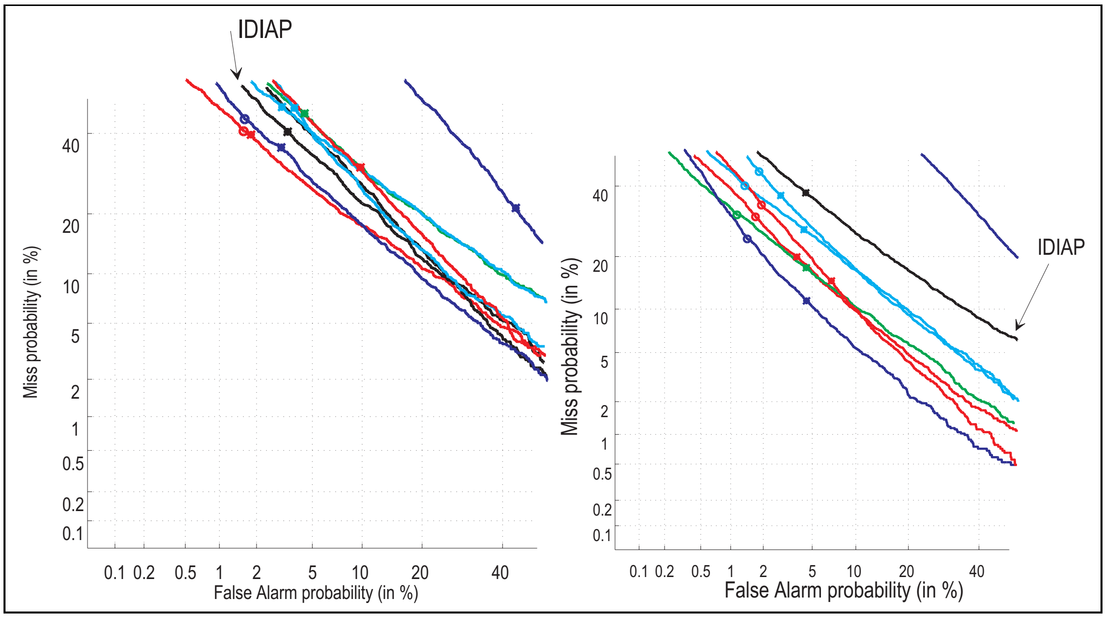

FIG. C.2 – *Les courbes ROC-NIST. A gauche pour les tests sur l'ensemble d'entraînement de 3 secondes de la même session. A droite pour des segments de tests de 30 secondes sur 2 combinés différents.*

Annexe D

Connexions nationales et internationales

Cette thèse a été réalisée dans le cadre de plusieurs projets, soit purement scientifiques (projets Fond National, projet COST250), soit en collaboration avec des industriels et des utilisateurs de technologies (CAVE/PICASSO, M2VTS, ATTACKS). Les démonstrateurs utilisés dans cette thèse ont servi comme systèmes de référence à plusieurs de ces projets.

D.1 Projets européens

Nous avons collaboré aux projets européens suivants:

- Le projet **COST250** intitulé *"Speaker recognition over the telephone line"* (Janvier 1996 - Décembre 1998). Ce projet est subventionné, pour la partie Suisse, par l'Office Fédéral pour l'Education et la Science (OFES). Il a pour but de réunir des experts européens dans le domaine de la reconnaissance du locuteur. Au niveau national, il a permis d'acquérir une expertise dans le domaine de la reconnaissance du locuteur.
- Projet européen Telematics No **LE 1930** nommé **CAVE** dont le titre est: *"Speaker verification to provide secure transaction in banking and telecommunications"* (Janvier 1996 - Décembre 1997). Un consortium s'est constitué, formé laboratoires de recherche européens (IDIAP et Ubilab Suisse, ENST France, KUN Pays-Bas, KTH Suède) d'industriels des télécommunications (Telecoms hollandais et suisses), de banques (UBS) et de fournisseurs de technologie (VOCALIS, Grande-Bretagne). Le but de ce projet était d'effectuer de la recherche et d'évaluer les technologies nécessaires pour introduire la vérification du locuteur dans des services grand public.
- Projet européen Telematics No **LEX 8369** nommé **PICASSO** (Janvier 1998 - Décembre 1999). Ce projet est la suite du projet CAVE.
- Projet européen ACTS project No **474** nommé **M2VTS** dont le titre est: "Multi-modal verification for tele-services and security applications". C'est également un consortium européen constitué de laboratoires de recherche et d'industriels. Le but du projet est d'effectuer des vérifications biométriques multimodales telles que la voix, le visage et les lèvres.

D.2 Projets FNRS

Le projet Fonds National Suisse pour la Recherche Scientifique No **20-45624.95**, intitulé “Enhanced automatic speaker recognition in telephony” nous a permis, pour une part, de poursuivre nos recherches dans la reconnaissance du locuteur.

D.3 Projets pré-industriels

Nous avons collaboré au projet ATTACKS [van Kommer, 1995] commandé par les Télécoms suisses, c’est ce projet qui nous a permis de mettre en place la première plate-forme qui a servi de système de référence. Les Télécoms nous ont également fourni l’accès à la base de données Polyphone (voir section B.4).

Bibliographie personnelle

[CAVE,1998] F. Bimbot, M. Blomberg, L. Boves, D. Genoud, H.P. Hutter, C. Jaboulet, J. Koolwaaij, J. Lindberg and J.B. Pierrot, *An overview of the CAVE project research activities in speaker verification*, submitted to SpeechCom journal, 1998

[RobuImp,1998] Dominique Genoud and Gérard Chollet, *Speech pre-processing against intentional imposture in speaker recognition*, in proc. of ICSLP-98, Sidney, 1998.

[Appli,1998] G. Caloz, C. Jaboulet, J. Mariéthoz, A. Glaeser et D. Genoud, *Voice-B project*, in proc. of IVTTA-98, Torino, 1998

[BinClassF,1998] Dominique Genoud, Miguel Moreira and Eddy Mayoraz, *Système de vérification du locuteur dépendante du texte utilisant des classificateurs binaires*, in proc. of Journées d'Etudes sur la Parole, Martigny, 1998.

[Fusion,1998] Patrick Verlinde, Dominique Genoud, Guillaume Gravier and Gérard Chollet, *Proposition d'une stratégie de fusion de données à trois niveaux pour la vérification d'identité*, in proc. of Journées d'Etudes sur la Parole, Martigny, 1998.

[TransfoImp,1998] Dominique Genoud and Gérard Chollet, *Voice transformation, a tool for imposture of speaker verification*, in proc. of International Phonetic Science conference IPS98, Washington, 1998.

[BinClass,1998] Dominique Genoud, Miguel Moreira and Eddy Mayoraz, *Text dependent speaker verification using binary classifiers*, in proc. of ICASSP-98, Seatle,1998.

[Thresh-2,1998] J.B. Pierrot, J. Lindberg, J. Koolwaaij, H.P. Hutter, M. Blomberg, D. Genoud et F. Bimbot, *A comparison of a priori threshold setting procedures for speaker verification in the CAVE project*, in proc. of ICASSP98, Seatle, 1998.

[Thresh-1,1998] J. Lindberg, J. Koolwaaij, H.-P. Hutter, D. Genoud, J.-B. Pierrot, M. Blomberg and F. Bimbot, *Techniques for a priori decision threshold estimation in speaker verification*, in proc. of RLA2C, Avignon, 1998.

[Polycost-3,1998] D. Petrovska, J. Hennebert, H. Melin et D. Genoud, *POLYCOST: A telephone-speech database for speaker recognition*, in proc. of RLA2C, Avignon, 1998.

[Th-adjust,1997] Frédéric Bimbot and Dominique Genoud, *Likelihood ratio adjustment for the compensation of model mismatch in speaker verification*, in proc. of EUROSPEECH97, Rhodos, 1997.

[Multim-3,1997] Pierre Jourlin, Juergen Luettin, Dominique Genoud and Hubert Wassner, *Acoustic-labial speaker verification*, in Pattern Recogntion Letters, Number 9, Volume 18, pp 853-858, 1997.

[Multim-2,1997] Pierre Jourlin, Juergen Luettin, Dominique Genoud and Hubert Wassner, *Integrating acoustic and labial information for speaker identification and verification*, in proc. of EUROSPEECH97, Rhodos, 1997.

[Multim-1,1997] Pierre Jourlin, Juergen Luettin, Dominique Genoud and Hubert Wassner, *Acoustic-labial speaker verification*, in proc. of the First International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'97)", 1997

[Combi-1,1996] Dominique Genoud, Frédéric Bimbot, Guillaume Gravier and Gérard Chollet, *Combining methods to improve speaker verification decision*, in proc. of the 4th International Conference on Spoken Language Processing, ICSLP 96, Philadelphia, USA, 1996.

[Combi-2,1996] Dominique Genoud, Guillaume Gravier, Frédéric Bimbot and Gérard Chollet, *Amélioration des performances de vérification du locuteur par combinaison de méthodes*, in proc. of "Journés d'études sur la parole", 1996.

[VoiceServ,1996] Olivier Bornet, Gérard Chollet, Jean-Luc Cochard, Andrei Constantinescu and Dominique Genoud, *Secured vocal access to telephone servers*, in proc. of IVTTA 1996 IEEE Third Workshop Interactive Voice Technology for Telecommunications Applications, 1996.

[Polycost-2,1996] Dijana Petrovska, Jean Hennebert, Dominique Genoud and Gérard Chollet, *Semi-Automatic HMM-based Annotation of the Polycost Database*, in proc. of COST workshop in Vigo, Spain, November 1996.

[Polycost-1,1996] Dominique Genoud, Jean Hennebert and Hakan Melin, *Polycost Database* in "COST250 minutes", Stockholm, 1996, *URL:http://circwww.epfl.ch/polycost/*.

[Polycode,1995] Dominique Genoud and Gérard Chollet, *Polycode: a speaker verification database*, Unpublished IDIAP technical rapport, 1995

Bibliographie

[Algazi *et al.*, 1993] V. R. Algazi, K. L. Brown, M. J. Ready, D. H. Irvine, C. L. Cadwell, et S. Chung. Transform representation of acoustic speech segments with applications-I: general approach and application to speech recognition. *IEEE trans. on speech and audio proc.*, 1(2):180–195, 1993.

[Antony, 1995] Richard T. Antony. *Principles of Data Fusion Automation*. Artech House, 685 Canton Street Norwood,MA 02062, 1995.

[Assaleh et Mammone, 1994] Khaled T. Assaleh et Richard J. Mammone. New lp-derived features for speaker identification. *IEEE Transactions on speech and audio processing*, 3(4):630–638, october 1994.

[Atal, 1974] Bishnu S. Atal. Effectivness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *JASA*, 55(6):1304–1312, 1974.

[Atal, 1976] Bishnu Atal. Automatic recognition of speakers from their voice. *Proceedings of IEEE*, 64(4):460–475, 1976.

[Basseville, 1989] Michèle Basseville. Distance measures for signal processing and pattern recognition. *Elsevier, Signal processing*, 18:349–369, 1989.

[Baudoin et Stylianou, 1996] G. Baudoin et Y. Stylianou. On the transformation of the speech spectrum for voice conversion. In *ICSLP96*, Philadelphia USA, October 1996. IEEE.

[Baum *et al.*, 1970] L.E. Baum, T.D. Petrie G. Soules, et N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Stat.*, 41:164–171, 1970.

[Baum et Petrie, 1966] Leonard E. Baum et Ted Petrie. Statitical inference for probabilistic functions of finite state markov chains. *Ann. math. Stat.*, 37:1554–1563, 1966.

[Baum, 1972] Leonard E. Baum. An inequality and associated maximisation technique in statistical estiomation for probabilistic functions of markov process. *Inequalities*, 3:1–8, 1972. Accademic press, NY.

[Bennani et Gallinari, 1994] Younès Bennani et Patrick Gallinari. Connexionist approaches for automatic speaker recognition. In ESCA [1994], pages 95–102.

[Besacier et Bonastre, 1997a] L. Besacier et J.F. Bonastre. Independent processing and recombination of partial frequency bands for automatic speaker recognition. In IEEE Korea Signal Processing Society, editor, *Fourteenth International Conference on Speech Processing. IEEE Korea Council*, Seoul, Korea, August 1997.

[Besacier et Bonastre, 1997b] L. Besacier et J.F. Bonastre. Subband approach for automatic-speaker recognition : optimal division of the frequency domain. *Computer Science*, I(1206):195–202, 1997.

[Bimbot *et al.*, 1997] Frédéric Bimbot, M. Blomberg, et al. Sv algorithms improvements and evaluation, deliverable 4.2. Technical report, Telematics European Project LE-1930: Caller Verification in Banking and Telecommunications (CAVE), 1997.

[Bimbot et Mathan, 1994] Frédéric Bimbot et Luc Mathan. Second-order statistical measures for text-independent speaker identification. In ESCA [1994], pages 51–54.

[Burges, 1997] Christopher J.C. Burges. A tutorial on support vector machine for pattern recognition. *Data Mining and Knowledge Discovery*, 1997.

[Cappé et Moulines, 1994] O. Cappé et E. Moulines. Regularization techniques for discrete cepstrum estimation. *IEEE Signal Processing Letters*, December 1994.

[Castelano *et al.*, 1997] P. Castelano, S. Slomka, et S. Sridharan. Telephone based speaker recognition using multiple binary classifier and gaussian mixture models. In *Proceedings ICASSP 97*, volume 2, pages 1075–1078, 1997.

[Charlet et Jouvet, 1996] D. Charlet et D. Jouvet. Optimisation du paramétrage acoustique pour la vérification du locuteur. In *Acte JEP 96*, pages 399–402, 1996.

[Chollet *et al.*, 1995] G. Chollet, J.L. Cochard, A. Constantinescu, et Ph. Langlais. Swiss french polyphone and polyvar: telephone speech databases to study intra and inter speker variability. Technical report, IDIAP, 1995.

[Chollet et Bimbot, 1995] Gérard Chollet et Frédéric Bimbot. *Spoken Language Ressources and Assessment*, volume 1 of *Handbook of Standards and Resources for Spoken Language Systems*, chapter Assessment of speaker verification systems. De Gruyter, Berlin, 1995.

[Colombi *et al.*, 1993] J. Colombi, T. Anderson, S. Rogers, D. Ruck, et G. Warhola. Auditory model representation for speaker recognition. In *ICASSP 93*, pages 700–703, 1993.

[Corsi, 1981] P. Corsi. Speaker recognition: A survey. Technical report, Nato Advanced Summer Institute, 1981.

[Dasarathy, 1994] Belur V. Dasarathy. *Decision Fusion*. IEEE Computer Society Press, Los Alamitos, California, 1994.

[Deller *et al.*, 1993] John R. Deller, John G. Proakis, et John H. L. Hansen. *Discrete-Time Processing of Speech Signals*. Maxwell Macmillan International, 1993.

[Digalakis *et al.*, 1995] V. V. Digalakis, D. Rtischev, et L. G. Neumeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE transactions on speech and audio proc.*, 3(5):357–366, September 1995.

[Doddington, 1976] G. Doddington. Personal identity verification using voice. *Proc. Electro.*, pages 22–24, 1976.

[Doddington, 1985] G. Doddington. Speaker recognition-identifying people from their voices. *IEEE*, 73(11):1651–1664, 1985.

[Drouiche, 1993] Karim Drouiche. *Chapitre IV: Test de sphéricité*. PhD thesis, ENST, Jan. 1993.

[Duda et Hart, 1973] R.O. Duda et P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[Dutoit, 1997] Thierry Dutoit. *An introduction to text-to-speech synthesis*, volume 3 of *Text, speech and language technology*. Kluwer Academic, Drodrecht, April 1997.

[Ent, 1996] Entropic Research Laboratory,Inc., Cambridge England. *Waves + manual*, ver 5.1 edition, March 1996.

[ESCA, 1994] ESCA, editor. *Workshop on Automatic Speaker Recognition Identification Verification*. Frédéric Bimbot and Gérard Chollet and Andrea Paoloni, Martigny April 1994.

[Fallside et Woods, 1985] Frank Fallside et William A. Woods. *Computer Speech Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[Forsyth, 1995] Mark Forsyth. Discriminating observation probability (dop) hmm for speaker verification. *Speech communication*, 17:117–129, 1995.

[Furui, 1981a] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. on ASSP*, ASSP-29(2):254–272, 1981.

[Furui, 1981b] Sadaoki Furui. Comparison of speaker recognition methods using statistical features and dynamic features. *IEEE Trans. on ASSP*, 29(3):342–350, 1981.

[Furui, 1986] Sadoki Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans, Acoustics, Speech Signal Proc., ASSP*, 34(1):52–59, February 1986.

[Furui, 1991] Sadaoki Furui. Speaker-dependent-feature extraction, recognition and processing techniques. *Speech Communication, Elsevier*, 10:505–520, 1991.

[Furui, 1994] Sadaoki Furui. An overview of speaker recognition technology. In ESCA [1994], pages 1–9.

[Gish *et al.*, 1986] H. Gish, M. Kraner, W. Russel, et J. Wolf. Methods and experiments for text-independent speaker recognition over the telephone line. In *ICASSP 86*, page 865, 1986.

[Gravier, 1995] Guillaume Gravier. Vérification du locuteur par modèles de markov cachés gauche-droite. Rapport de stage dea, IDIAP, CH-1920 Martigny, 1995.

[Gray et Kopp, 1994] C.H. Gray et G.A. Kopp. Voiceprint identification. Bell telephone report, Bell Laboratories, 1994.

[Gray et Markel, 1976] Augustine H. Gray et John D. Markel. Distance measure for speech processing. *IEEE Trans. on acc. speech. and sig. proc.,ASSP*, 24(5):380–391, October 1976.

[Green et Swets, 1988] David M. Green et John A. Swets. *Signal detection theory and pschophysics*. John Wiley & sons, 1988. reprint of the 1966 edition.

[Griffin *et al.*, 1994] Ch. Griffin, T. Matsui, et S. Furui. Distance measure for text-independent speaker recognition based on mar model. In *ICASSP94*, volume I, pages 309–312, Adelaïde, Australia, April 1994. IEEE.

[Griffin et Lim, 1988] D.W. Griffin et J.S. Lim. Multiband-excitation vocoder. *IEEE Trans. Acoust. Speech and Signal Proc., ASSP*, 36(2):236–243, February 1988.

[Hoffbeck et Landgrebe, 1996] Joseph P. Hoffbeck et David A. Landgrebe. Covariance matrix estimation and classification with limited training data. *IEEE trans. on pattern analysis and machine intelligence*, 18(7):763–767, July 1996.

[Homayounpour *et al.*, 1994] Mehdi Homayounpour, Gérard Chollet, et Jacqueline Vaissière. Etude des variabilités intra- et inter-locuteurs provoquées par l'émotion et l'imitation pour les systèmes de vérification du locuteur. In *JEP94*, pages 269–274. GFCP, 1994.

[Homayounpour et Chollet, 1994] Mehdi Homayounpour et Gérard Chollet. A comparison of some relevant parametric representations for speaker verification. In ESCA [1994], pages 185–188.

[Homayounpour, 1995] Mehdi Homayounpour. *Vérification vocale d'identité Dépendante et indépendante du texte*. PhD thesis, Université PARIS-SUD, centre d'Orsay, May 1995.

[JEP98, 1998] GFCP. *Actes des XXIIèmes Journées d'Etudes sur la parole*. IDIAP, CH-1920 Martigny, juin 1998.

[Klatt et Klatt, 1990] D.H. Klatt et L.C. Klatt. Analysis, synthesis and perception of voice quality variations among female and male talkers. *Journal of the Acoustical Society of America*, 87(2):820–857, February 1990.

[Klatt, 1980] D.H. Klatt. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustic Society of America*, 67(3):971–995, March 1980.

[Kleinrock, 1975] Leonard Kleinrock. *Queueing systems*, volume Volume I: Theory. John Wiley and Sons, 1975.

[Kokkinakis *et al.*, 1997] G. Kokkinakis, N. Fakotakis, et E. Dermatas, editors. *Eurospeech97, 5th conference on Speech Communication and Technology*, Rhodos Greece, September 1997. ESCA.

[Ldc, 1994] Linguistic data consortium. HTML link: http://www.ldc.upenn.edu/, 1994. University of Pennsilvania.

[Lee, 1991] Yi-Teh Lee. Information-theoretic distortion measures for speech recognition. *IEEE Trans, Acoustics on Signal Proc.*, 39(2):52–59, February 1991.

[Li *et al.*, 1995] H. Li, J.P. Haton, J. Su, et Y. Gong. Speaker recognition with temporal transition models. In *EUROSPEECH95*, pages 617–620. ESCA, 1995.

[Li et Wrench, 1983] K. P. Li et Jr. Wrench. An approach to text-independent speaker recognition with short utterances. In *ICASSP-83*, pages 55–58, 1983.

[Lin, 1995] Qiguang Lin. A fast algorithm for computing the vocal tract impulse response from the transfer function. *IEEE transactions on speech and audio proc.*, 3(6):449–457, November 1995.

[Makhoul, 1975] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, April 1975.

[Mammone *et al.*, 1996] Richard J. Mammone, Xiaoyu Zhang, et Ravi P. Ramachandran. Robust speaker recognition. *IEEE signal processing magazine*, pages 58–71, september 1996.

[Martin *et al.*, 1997] A. Martin, G. Doddington, T. Kamm, M. Ordowski, et M. Przybocki. The det curve in assessment of detection task performance. In Kokkinakis et al. [1997].

[Martin, 1997] A. Martin. Nist 1997 speaker recognition evaluation plan. Technical report, NIST, 1997. web http://www.cis.ohio-state.edu/%7ekchen/.

[Martin, 1998a] A. Martin. Nist 1998 speaker recognition evaluation plan. Technical report, NIST, 1998. web http://www.nist.gov/speech/msrec98.html.

[Martin, 1998b] A. Martin. Nist speaker recognition evaluations, review of the 1997 & 1998 evaluations. Technical report, NIST, 1998. http://www.nist.gov/speech/rla2c_pres/index.htm.

[Masuko *et al.*, 1997] T. Masuko, K. Tokuda, T. Kobayashi, et S. Imai. Voice characteristics conversion for hmm-based speech synthesis system. In *ICASSP 97*, pages 1611–1614, Münich, 1997. IEEE.

[Matsui et Furui, 1992] T. Matsui et S. Furui. Speaker recognition using concatenated phoneme models. In *ICSLP*, page 603, 1992.

[Matsui et Furui, 1993] T. Matsui et S. Furui. Concatenated phoneme models for text-variable speaker recognition. In *ICASSP-93*, pages 391–394, 1993.

[Matsui et Furui, 1994] T. Matsui et S. Furui. Speaker adaptation of tied-mixture-based phoneme models for text-prompted speaker recognition. In *ICASSP-94*, page 13.1, 1994.

[Matsui et Furui, 1995a] T. Matsui et S. Furui. Likelihood normalization for speaker verification using a phoneme and speaker-independent model. *Speech Communication, Elsevier*, 17:109–116, 1995.

[Matsui et Furui, 1995b] T. Matsui et S. Furui. A study of speaker adaptation based on minimum classification error training. In ESCA, editor, *EUROSPEECH95*, pages 81–84, Madrid, September 1995.

[Mitchell, 1997] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Mokbel, 1992] Chafic Mokbel. *Reconnaissance de la parole dans le bruit: bruitage/débruitage*. PhD thesis, ENST, Paris, 1992.

[Moreira et Mayoraz, 1998] Miguel Moreira et Eddy Mayoraz. Improved pairwise coupling classification with correcting classifiers. In Claire Nédellec et Céline Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 160–171. Springer, April 1998.

[Morgan et Bourlard, 1995] Nelson Morgan et Hervé Bourlard. An introduction to the hybrid hmm/connectionist approach. *IEEE sig. proc. magazine*, pages 24–42, May 1995.

[Moser, 1997] Thomas Moser. Bericht vorprojekt: Einsatz der spracherkennung und sprecherverifikation. Technical report, Swisscom, 1997.

[Myers et Rabiner, 1981] C.S. Myers et L.R. Rabiner. Connected digit recognition using level building dtw algorithm. *IEEE trans. on ASSP*, 29(3):351–363, June 1981.

[Naik *et al.*, 1994] D. Naik, K. Assaleh, et R. Mammone. Robust speaker identification using pole filtering. In *Esca Workshop on Speaker Recognition, Identification, and Verification*, pages 225–228, 1994.

[Naik et Doddington, 1987] J. Naik et G.R. Doddington. Evaluation of a high performance speaker verification system for acces control. In *ICASSP 87*, pages 2392–2395, 1987.

[Naik et Lubensky, 1994] Jayant M. Naik et David M. Lubensky. A hybrid hmm-mlp speaker verification algorithm for telephone speech. In *ICASSP-94*, pages 153–156, 1994.

[Naik, 1994] Jay Naik. Speaker verification over the telephone network: databases, algorithms and performances assessment. In ESCA [1994].

[Ng *et al.*, 1995] K.T. Ng, H. Li, et J.P. Haton. Some non-parametric distance measures in speaker verification. In *EUROSPEECH 95*, pages 317–320, 1995.

[Oglesby, 1994] John Oglesby. What's in a number? moving beyond the equal error rate. In ESCA [1994], pages 87–90.

[Oppenheim *et al.*, 1968] A. V. Oppenheim, R.W. Shafer, et T.G. Stockham. Nonlinear filtering of multiplied and convolved signals. *Proc IEEE*, 56:1264–1291, August 1968.

[Paliwal et Kleijn, 1995] Kuldip K. Paliwal et W. Bastiaan Kleijn. *Quantization of LPC parameters*, chapter 12, pages 433–466. Elsevier Science, 1995.

[Papoulis, 1984] Anasthasios Papoulis. *Random variables and stochastic processes*, chapter 12. McGraw-Hill, 3rd edition, 1984.

[Pickles, 1988] James O. Pickles. *An introduction to the physiology of hearing*, volume 1. Academic Press Limited, San Diego,CA, 2 edition, 1988.

[Picone, 1993] Josef W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247, September 1993.

[Pierrot, 1998] Jean-Benoît Pierrot. *Elaboration et validation d'approches en vérification du locuteur*. PhD thesis, ENST, Paris, Septembre 1998.

[Quinlan, 1993] J. Ross Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Rabiner et Juang, 1993] Lauwrence Rabiner et Bing-Hwang Juang. *Fundamentals of speech recognition*. signal processing. Prentice Hall, Englewood Cliffs, NJ, 1993.

[Rabiner et Schafer, 1978] L. R. Rabiner et R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, New Jersey, 1978.

[Raudys et Jain, 1991] Sarunas J. Raudys et Anil K. Jain. Small sample size effects in statistical pattern recognition: recommendations for practitionners. *IEEE Trans. on pattern and machine intelligence*, 13(3):252–264, March 1991.

[Reynolds, 1992] Douglas A. Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification*. PhD thesis, Georgia Institute of Technology, 1992.

[Reynolds, 1997] Douglas A. Reynolds. Comparison of background normalization methods for text independent speaker verification. In Kokkinakis et al. [1997].

[Ripley, 1996] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge university press, 1996.

[Rose *et al.*, 1994] R.C. Rose, E.M. Hofstetter, et D.A. Reynolds. Integrated models of signal and background with application to speaker identification in noise. *IEEE transactions on speech and audio proc.*, 2(2):245–257, April 1994.

[Rosenberg *et al.*, 1991] A.E. Rosenberg, C.H. Lee, et S. Gokoen. Connected word talker verification using whole word hidden markov model. In *ICASSP-91*, pages 381–384, 1991.

[Rosenberg *et al.*, 1992] A.E. Rosenberg, J. Delong, C.H. Lee, et B.H. Juang andF.K. Soong. The use of cohort normalized scores for speaker verification. In *ICSLP-92*, pages 599–602, 1992.

[Sakoe et Chiba, 1978] H. Sakoe et Chiba. Dynamic programing algorithm optimization for spoken word recognition. *IEEE Trans. on ASSP*, 26(1):43–49, 1978.

[Saporta, 1990] Gilbert Saporta. *Probabilités, analyse des données et statistique*, volume I. Editions Technip, 1990.

[Schalkwyk *et al.*, 1994a] J. Schalkwyk, E. Baranard, et J.R. Sachs. Detecting an imposter in telephone speech. In *ICASSP94*, volume I, pages 169–172, Adelaïde, Australia, April 1994. IEEE.

[Schalkwyk *et al.*, 1994b] Johan Schalkwyk, Etienne Barnard, et Jeffrey R. Sachs. Detecting an imposter in telephone speech. In *ICASSP-94*, volume I, pages 169–172, 1994.

[Scharf, 1991] L.L. Scharf. *Statistical Signal Processing. Detection, Estimation and Time Series Analysis.* Addison-Wesley Publishing Company, 1991.

[Scherer *et al.*, 1998] K.R. Scherer, T. Johnstone, et J. Sangsue. Lt'état émotionnel du locuteur: facteur négligé mais non négligeable pour la technologie de la parole. In JEP98 [1998].

[Schroeder, 1985] M.R. Schroeder. *Speech and Speaker Recognition*. Karger, 1985.

[Setlur et Jacobs, 1995] Arnand Setlur et Thomas Jacobs. Results of a speaker verification service trial using hmm models. In *EUROSPEECH95*, pages 639–642. ESCA, 1995.

[Shafer, 1976] G. Shafer. *A mathematical Theory of Evidence*. MIT Press, Cambridge, 1976.

[Soong *et al.*, 1985] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, et B.H. Juang. A vector quantization approach to speaker recognition. In *ICASSP 85*, pages 387–390, 1985.

[Stylianou, 1996] I. Stylianou. *Modèles harmonique plus bruit combinés avec méthodes statistiques, pour la modification de la parole et du locuteur*. PhD thesis, ENST Paris, 1996.

[Thévenaz, 1993] Ph. Thévenaz. *Résidu de prédiction linéaire et reconnaissance de locuteurs indépendante du texte*. PhD thesis, Université de Neuchâtel, 1993.

[Tokuda *et al.*, 1995] K. Tokuda, T. Masuko, Yamada, T. Kobayashi, et S. Imai. An algorithm for speech parameter generation from continuous mixture hmm with dynamic features. In *EUROSPEECH95*, pages 757–760, Madrid, 1995. ESCA.

[van Kommer, 1995] Robert van Kommer. Attacks project. Technical Report FE321.061, Swiss Telecom PTT, 1995.

[Vapnik, 1995] Valdimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Wakita, 1973] Hirashi Wakita. Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms. *IEEE trans. on audio and electroacoustics*, AU-21(5):417–427, October 1973.

[Wakita, 1979] Hirashi Wakita. Esitmation of vocal-tract shapes from acoustical analysis of the speech wave: the state of the art. *IEEE transactions on acoustics, speech and sig. proc., ASSP*, 27(3):281–285, 1979.

[Xu *et al.*, 1992] L. Xu, A. Krzyzak, et C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on systems, man and cybernetics*, 22(3), may/June 1992.

[Young et Bloothooft, 1997] Steve Young et Gerrit Bloothooft. *Corpus based methods in language and speech processing*, volume 2 of *Text speech and language technology*. Kluwer Academic, Dordrecht, 1997.

[Zwiker et Terhardt, 1980] E. Zwiker et E. Terhardt. Analytical expressions for crtitical-band rate and critical bandwith as a function of frequency. *JASA*, 68(5):1523–1525, 1980.

Curriculum vitæ

Name :	Dominique Alain Genoud
Birth date :	June 26, 1962
Origin :	Bagnes, VS, Switzerland
Languages :	French (mother tongue), English (fluent), Spanish, German
Family status :	married, one child.
Private address :	Rue de Latigny 16, CH-1955 Chamoson VS, Switzerland.
Professional address :	IDIAP
	CP 592, CH-1920 Martigny, VS, Switzerland
voice	+ 41 - 27 - 721 77 26
fax	+ 41 - 27 - 721 77 13
e-mail	Dominique.Genoud@idiap.ch

Education

1989 Eng. Dipl. of the Swiss Federal Institute of Technology Lausanne. (Microtechnic).

Additional education

1991–1992 Postgraduate course at the Computer Science Department, EPFL : Biological and Artificial Neural Networks.

Experiences

1995-1998 Researcher at IDIAP in speaker verification domain, involved in different European projects (Cost, Telematics, ACTS).

1994-1995 Research engineer at IDIAP, Martigny, Switzerland, research and development of speaker recognition and verification systems for Swisscom and other companies.

1993-1994 Round the world trip.

1992-1993 Team manager of the research and development department of CIMTEC (swiss competence center for quality insurance, Sion, Switzerland), teaching product development methods at Ecole d'ingénieurs du Valais (EIV), Sion, Switzerland, project leader in technical support to Swiss industry.

1991-1992 Project leader at EIV, Sion, Switzerland, responsibility of industrial projects in high precision measurement systems (mechanical and optical systems).

1989-1991 Member of the research and development team in NAGRA-Kudelski, Cheseaux, Switzerland, development of a new data format, of software and hardware for professional digital audio tape recorders.