# Table des matières

# Chapitre 1

# Introduction

Aujourd'hui, la reconnaissance automatique de la parole est un domaine de recherche bien établi, à partir duquel émergent des technologies permettant le développement d'applications réelles. Or dans une application réelle, les systèmes de reconnaissance sont soumis à de nombreuses sources de bruit engendrant une dégradation significative des performances. Des taux de reconnaissance raisonnables peuvent être atteints à condition d'imposer suffisamment de contraintes sur l'application ; parmi ces contraintes on peut citer la limitation de la taille du vocabulaire, l'utilisation du même microphone en phase de test et d'apprentissage, l'utilisation du système dans une ambiance relativement calme, *etc.* Ces conditions sont très contraignantes et ne peuvent pas être remplies dans bon nombre d'applications. L'objectif principal de cette thèse est de proposer des algorithmes permettant de changer de microphone ou d'environnement acoustique sans que cela n'affecte les performances du système de reconnaissance : il s'agit de l'adaptation des systèmes de reconnaissance aux conditions acoustiques d'utilisation.

Le travail effectué dans cette thèse se situe dans le cadre de la reconnaissance de la parole par des modèles statistiques. Le principe de la reconnaissance de la parole s'inscrit dans le domaine plus général de la reconnaissance des formes. L'idée de base est d'apprendre des formes (des statistiques sur ces formes) pour pouvoir les reconnaître par la suite, la forme reconnue étant celle parmi toutes les formes apprises qui ressemble le plus à la forme inconnue. La mesure de ressemblance utilisée est généralement matérialisée par une mesure de distance de similarité. Si les formes de test ne subissent pas de distorsions majeures par rapport aux formes apprises, avant que ces mesures de similarité ne soient appliquées, le système de reconnaissance des formes donne les meilleures performances. En revanche, si les formes de test sont modifiées par des événements inconnus *a priori*, les performances du système de reconnaissance des formes chutent. On dit dans ce cas qu'il y a des différences entre les conditions d'apprentissage et les conditions de test. Dans le cas d'un système de reconnaissance de la parole, ces différences peuvent être causées, par exemple, par la présence de bruit en phase de test, par le changement de microphone, par le changement de locuteur, par l'effet Lombard *etc.* Ces différences sont rassemblées sous le vocable "variabilités acoustiques". L'objectif de ce travail est de réduire les effets des différences entre les conditions d'apprentissage et de test ; c'est-à-dire d'atteindre les mêmes performances que dans le cas où l'apprentissage et le test se déroulent dans les mêmes conditions acoustiques.

Il est supposé que le système de reconnaissance est fondé sur une approche statistique et que le signal à modéliser est une représentation discrète du signal temporel correspondant, obtenue par une analyse centiseconde de type spectrale ou cepstrale. Le problème de modélisation du signal de parole non bruité $\mathbf{s}$ consiste à estimer $f_\lambda(\mathbf{s}|\mathbf{w})$, la densité de probabilité du signal $\mathbf{s}$ étant donné le message $\mathbf{w}$ (la phrase prononcée), $\lambda$ désigne l'ensemble des paramètres définissant $f$. Une solution couramment retenue pour $f$ est une fonction probabiliste d'une chaîne de Markov (appelée modèle de Markov caché : MMC), où le vecteur de paramètres $\lambda$ est alors estimé à partir d'observations par la technique du maximum de vraisemblance en faisant usage de l'algorithme E.M. ("Expectation-Maximization") [15] ou d'une de ses variantes. Le but principal de cette thèse est de modéliser le signal de parole en présence de bruits afin d'augmenter la robustesse des systèmes de reconnaissance automatique de la parole. La majeure partie du travail présenté concerne la conception, la réalisation et l'évaluation de traitements visant à minimiser les différences entre les conditions acoustiques d'apprentissage et de test. Nous abordons le problème essentiellement du point de vue du système de reconnaissance évitant ainsi la nécessité d'estimer le signal propre (sans bruit). Le gain est alors mesuré en terme de taux de reconnaissance et non en terme d'amélioration du Rapport Signal sur Bruit (RSB). La problématique de l'estimation du signal propre en vue d'améliorer son RSB a néanmoins été abordée.

## 1.1 Variabilités acoustiques

On peut considérer comme source de bruit tout facteur susceptible d'introduire des variabilités acoustiques dans les formes spectrales d'apprentissage ou de test. Nous distinguons dans ce qui suit différentes sources de bruits, que nous regroupons suivant des classes en vue d'un traitement automatique :

• **Le locuteur** : le même message prononcé deux fois par un même locuteur dans les mêmes conditions produit deux formes spectrales différentes. Cette variabilité s'appelle variabilité *intra-locuteur*. Ce phénomène est amplifié lorsqu'une même unité acoustique est prononcée par deux locuteurs différents. On parle alors de variabilité *inter-locuteur*. Pour modéliser les variabilités *intra-locuteur* et *inter-locuteur* on utilise des modèles statistiques, notamment les modèles de Markov cachés [67]. Cependant, avec l'augmentation de la perplexité du langage de l'application, la modélisation de ces variabilités[1] par des modèles statistiques atteint ses limites et les performances des systèmes de reconnaissance baissent quel que soit le nombre de locuteurs utilisés pour l'apprentissage.

• **L'environnement** : l'environnement dans lequel s'exprime le locuteur peut générer des bruits corrélés avec le signal de parole, comme la réverbération ou non corrélés, comme le bruit ambiant.

[1] Surtout la variabilité *inter-locuteur*.

• **L'effet Lombard** : l'environnement agit aussi sur la manière de parler de la personne qui s'adresse au système de reconnaissance. En effet, si une personne parle dans un environnement où le rapport signal sur bruit est faible alors celle-ci va généralement augmenter la puissance de sa voix, ce qui entraîne certaines modifications du spectre du signal de parole résultant. Ces modifications portent le nom d'effet Lombard. Elles se traduisent généralement par une augmentation de la fréquence du premier formant et de la fréquence fondamentale. Le second formant a tendance à se déplacer vers les basses fréquences, la pente spectrale se trouve modifiée et la vitesse d'élocution peut varier considérablement. Plusieurs expériences ont montré que l'effet indirect du bruit (effet Lombard) peut être plus nuisible à la reconnaissance que l'effet additif du bruit lui-même, en particulier lorsque le bruit ambiant est très stationnaire [69, 48].

• **Le canal d'enregistrement** : le canal d'enregistrement génère des variabilités dépendant du type de microphone utilisé pour l'enregistrement, de la distance et de l'orientation du locuteur par rapport au microphone. Dans le cas d'un système de transmission comme la ligne téléphonique, on observe des distorsions et des échos dépendant du support utilisé. Ce type de variabilité a un effet convolutif dans le domaine temporel, multiplicatif dans le domaine des densités spectrales et additif dans le domaine cepstral, le bruit qui en résulte est appelé *bruit convolutif*.

• **Le gain du signal** : le niveau moyen en terme d'énergie varie d'une phrase à l'autre et même au sein d'une même phrase. Cette variation peut être due au fait que le niveau d'élocution varie naturellement d'un locuteur à l'autre. D'autres causes pouvant influencer le gain du signal enregistré sont la distance et l'orientation du locuteur par rapport au microphone. Pour rendre le système insensible au gain du signal enregistré, il est commode de procéder à une normalisation. Cette normalisation ne peut pas être réalisée trame par trame, car le niveau d'élocution constitue une caractéristique à long terme du signal de parole. Il faut donc attendre un certain temps (typiquement la fin de la phrase) pour pouvoir procéder à une telle normalisation. Cependant, dans les petites applications, l'utilisation de l'énergie dans les vecteurs de paramètres ne semble pas indispensable. Dans ce cas le gain du signal en terme d'énergie n'intervient pas et donc aucune normalisation n'est nécessaire.

La plupart des attributs de l'environnement acoustique introduisant des variabilités, restent relativement constants durant l'élocution d'une phrase. Parmi ces attributs on peut citer l'équipement d'enregistrement, la quantité d'écho dans la salle d'enregistrement, et les caractéristiques acoustiques du locuteur utilisant le système. D'autres facteurs comme le bruit ambiant, le niveau ou le volume du signal, peuvent être considérés comme variant lentement par rapport au signal de parole. Les systèmes de reconnaissance de la parole actuels sont fondés sur les Modèles de Markov Cachés [67]. Ces derniers modélisent principalement les caractéristiques à court terme du signal (message linguistique), tandis que les propriétés à long terme sont très mal modélisées, voire ignorées.

Il est bien évidemment impossible de modéliser toutes les variabilités acoustiques par les MMC, car d'une part la taille du corpus d'apprentissage augmente très rapidement avec le nombre de facteurs de variabilité et d'autre part même avec des corpus infiniment grands pour modéliser toutes les variabilités acoustiques, les modèles obtenus deviendraient peu précis (avec de grandes variances) entraînant une augmentation d'erreur[2]. Dans la mesure du possible certaines variabilités peuvent être modélisées par les MMC en augmentant la taille du corpus d'apprentissage. Parmi ces variabilités nous avons déjà cité les variabilités *intra-locuteur* et *inter-locuteur*. D'autres variabilités comme le bruit ambiant sont difficilement modélisables dans les MMC à partir des données d'apprentissage. Ceci est dû au fait que le bruit ambiant affectant les données de test est en général inconnu *a priori* et au fait que les types et les niveaux des bruits pouvant affecter les données de test sont très variables et inconnus *a priori*.

En vue d'un traitement automatique, nous répartissons les variabilités acoustiques en trois classes : la première est celle des bruits additifs, la deuxième est celle des bruits convolutifs et la troisième est celle rassemblant toutes les autres variabilités acoustiques[3].

### 1.1.1 Le bruit additif

Le bruit additif affecte directement le spectre du signal, ce qui se traduit par l'apparition de certains pics qui n'existent pas dans le signal d'origine, par la disparition de certains pics du signal d'origine, par l'aplatissement de l'enveloppe spectrale (perte d'information) et par la modification de la largeur de bande et de la pente spectrale. Le bruit ambiant est toujours considéré comme étant additif. Les sources de ce genre de bruit sont nombreuses, on peut citer l'exemple d'une ambiance de gare : le bruit des trains, des conversations à proximité du système, les annonces, *etc.* La plupart des bruits sont additifs, ils peuvent être corrélés (réverbération et écho) ou non corrélés avec le signal de parole, stationnaires ou non stationnaires. Si le bruit ambiant est non corrélé avec le signal de parole, il devient additif dans le domaine de l'autocorrélation et par suite dans le domaine des densités spectrales. La difficulté de traiter le bruit dépend de ses caractéristiques. Le bruit le plus étudié est celui qui est plus ou moins stationnaire et non corrélé avec le signal de parole.

[2] On peut envisager de modéliser de manière implicite les différentes sources de variabilité à l'intérieur des modèles acoustiques. Ceci est fait actuellement dans les systèmes de reconnaissance pour les variabilités *intra-locuteur* et *inter-locuteur* par les MMC, qui appris sur un grand corpus (quelques dizaines de milliers de phrases) *multi-locuteur* (quelque centaines de locuteurs) donnent de bon résultats [43]. Pour ne pas aboutir à des modèles avec des variances trop élevées où toutes les formes à discriminer se chevauchent de manière importante, il devient nécessaire de construire différents types de modèles acoustiques en fonction d'un facteur de variabilité donné. Il faut alors chercher le compromis entre modélisation précise (faibles variances) et les contraintes (temps, mémoire, corpus) du système de reconnaissance. Un exemple typique d'un tel compromis est l'estimation de modèles dépendant du sexe du locuteur, aboutissant à deux ensembles de modèles à variances plus réduites qu'un seul ensemble de modèles indépendant du locuteur. Ceci nécessite également l'étiquetage du corpus de parole en terme du facteur de variabilité modélisée (ici homme/femme). Cet étiquetage est plus ou moins simple suivant la source de variabilité considérée. Par exemple l'étiquetage homme/femme est trivial, notamment si cette information a été retenue lors de la création de la base de données. Une telle solution est moins évidente dans le cas de la variabilité due au bruit ambiant.

[3] Sauf la variabilité liée au message linguistique.

### 1.1.2 Le bruit convolutif

Le bruit convolutif est celui qui affecte le spectre quand le signal de parole subit un filtrage linéaire invariant dans le temps. Dans le domaine *log-spectral* ou *cepstral* l'effet de ce type de bruit est strictement additif. Il existe plusieurs sources de ce type de bruit :

• **L'équipement d'enregistrement :** si on utilise différents microphones, on peut constater des modifications tout au long du spectre. Le changement de microphone peut être raisonnablement modélisé par un filtre linéaire.

• **Réverbération :** l'effet de la réverbération peut être modélisé par un filtre linéaire qui dépend de l'architecture, du contenu de la salle, et de la position du locuteur par rapport au microphone.

• **Les caractéristiques spectrales spécifiques à un locuteur :** la moyenne à long-terme de la réponse fréquentielle du conduit vocal varie d'un locuteur à l'autre. Ceci est dû aux différences physiologiques des conduits vocaux. Des systèmes de vérification et d'identification ont été construits en utilisant ce principe. La variabilité causée par ce genre de différences peut être vue comme un filtrage d'un signal produit par un locuteur de référence. Le filtre représente les différences physiologiques entre le vrai locuteur et ce locuteur de référence (c'est un locuteur fictif)[4] [86].

Le bruit convolutif est une cause majeure de la dégradation des performances. Les mesures de distorsions spectrales utilisées en reconnaissance de la parole se trouvent sérieusement affectées par ce type de bruit.

### 1.1.3 Les autres variabilités

Dans cette classe sont rassemblées toutes les variabilités qu'on ne sait pas explicitement modéliser[5] comme la variabilité *inter-locuteur*[6], l'effet Lombard, les effets d'une ligne téléphonique, la position et l'orientation par rapport au microphone *etc.* La justification de cette classification vient du fait que les seules variabilités que nous savons modéliser explicitement et simplement sont celles correspondant aux bruits additifs (non corrélés avec le signal de parole) et aux bruits convolutifs.

Dans ce travail les variabilités liées aux bruits additifs et convolutifs seront explicitement compensées. Les autres types de variabilités "résiduelles" seront compensées globalement d'une manière aveugle.

## 1.2 Travaux réalisés

Les solutions apportées au problème de la robustesse des systèmes de reconnaissance au bruit peuvent intervenir à tous les niveaux : au niveau de l'augmentation de qualité du signal (débruitage), au niveau d'une représentation spectrale robuste

[4] La variabilité inter-locuteur n'est que partiellement prise en compte par filtrage linéaire.

[5] Nous connaissons leurs effets mais nous ne disposons pas de modèle mathématique.

[6] La variabilité *inter-locuteur* est prise en compte partiellement par les MMC et le modèle du bruit convolutif.

au bruit (paramétrisation), au niveau des modèles acoustiques représentant les références (adaptation), *etc.* Nous pensons qu'il n'existe pas de solution unique au problème ; notre seul recours est de combiner les meilleures techniques de robustesse intervenant à tous les niveaux des traitements, espérant atteindre des performances comparables à celles obtenues en conditions acoustiques favorables. Dans ce travail, après une étude théorique et expérimentale approfondie, nous avons abouti à la réalisation de techniques de robustesse intervenant à plusieurs niveaux. Dans la suite de cette section nous décrivons notre contribution à chaque niveau.

### 1.2.1 Niveau de la paramétrisation acoustique

Le but du travail au niveau de la paramétrisation acoustique est d'extraire du signal acoustique des vecteurs de paramètres dépendant le moins possible du bruit.

Une étude expérimentale sur la robustesse de certaines techniques de paramétrisation (MFCC[7], LPCC[8], PLPCC[9]), nous permet d'affirmer qu'une paramétrisation de types MFCC ou PLPCC donne de meilleurs résultats qu'une paramétrisation de type LPCC. Cependant, quel que soit le bruit affectant les données de test ou d'apprentissage, et quel que soit le type de paramètres acoustiques choisis, le système de reconnaissance de la parole se trouve sérieusement perturbé, surtout pour des rapports signal sur bruit faibles. L'étude et la mise en oeuvre de techniques permettant de compenser les effets du bruit sont donc indispensables pour garantir une bonne robustesse d'un système de reconnaissance lors d'une application dans des conditions acoustiques d'utilisation difficiles.

Dans ces travaux nous avons choisi d'utiliser une paramétrisation de type MFCC. Tous les algorithmes de compensation développés dans cette thèse peuvent être appliqués quelle que soit la technique de paramétrisation utilisée (MFCC, LPCC, PLPCC, *etc*).

### 1.2.2 Niveau des modèles acoustiques

L'essentiel du travail effectué dans cette thèse porte sur l'adaptation des modèles acoustiques pour compenser les diverses sources de variabilités décrites dans la section 1.1. Nous distinguons deux approches pour réaliser cette adaptation ; la première consiste en l'adaptation explicite des bruits additifs et convolutifs ; la seconde consiste en l'adaptation aveugle de toutes les autres variabilités résiduelles.

#### Compensation des bruits additifs et convolutifs

Dans ce travail, la compensation des bruits additifs et convolutifs est fondée sur le modèle du canal de transmission : si on suppose que le canal d'enregistrement peut être représenté par un filtre linéaire alors les signaux enregistrés peuvent être raisonnablement modélisés par (voir Figure 1.1) :

$$y(t) = (s(t) + n(t)) * h(t), \tag{1.1}$$

[7] Mel scaled Frequency Cepstrum Coefficients.
[8] Linear Predictive Cepstrum Coefficients.
[9] Perceptually-based Linear Prediction Cepstrum Coefficients.

où $s(t)$ représente le signal de parole propre, $n(t)$, le bruit additif, $h(t)$, le bruit convolutif et $y(t)$, le signal de parole bruité. Le bruit $n(t)$ que nous allons traiter est supposé non corrélé avec le signal de parole $s(t)$.

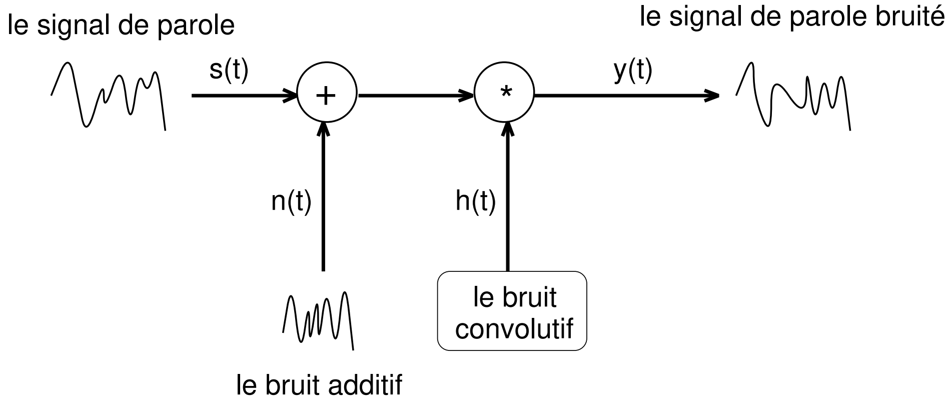



**Figure 1.1 :** *Modèle du canal de transmission utilisé :* $y(t) = (s(t) + n(t)) * h(t)$*, avec* $y(t)$ *le signal de parole bruité,* $s(t)$ *le signal de parole propre,* $n(t)$ *le bruit additif et* $h(t)$ *le bruit convolutif.*

Supposons dans un premier temps que les données d'apprentissage sont propres et les données de test sont bruitées. L'adaptation au bruit ambiant des modèles acoustiques représentant les références consiste en une compensation explicite des bruits (additif et convolutif) en utilisant le modèle du canal de transmission (équation 1.1). La technique d'adaptation proposée dans cette thèse opère en mode non supervisé, en utilisant seulement les données de test[10].

La compensation des bruits additif et convolutif se fait en deux étapes : la première consiste à estimer d'une manière itérative $h$ et $n$, et la deuxième consiste à compenser leurs effets en bruitant les modèles avec les bruits estimés. Cette technique permet d'adapter les modèles acoustiques à la parole bruitée en réduisant le décalage entre les conditions acoustiques d'apprentissage et d'utilisation. Cette approche constitue l'une des originalités de ce travail de thèse : elle permet à la fois d'adapter le système aux bruits ambiants et aux bruits convolutifs, et elle permet d'adapter les coefficients cepstraux dynamiques[11] sans aucune approximation quelles que soient les formules utilisées pour leurs estimations[12]. Nous appelons cette technique *Combinaison Parallèle de Données* (CPD), car elle consiste à combiner les trames de la parole propre avec celles du bruit, les trames résultantes sont utilisées pour estimer les modèles de la parole bruitée.

Nous avons réalisé une étude expérimentale afin d'évaluer l'apport de notre approche. Cette étude a été effectuée avec différents types de bruits additifs, stationnaires comme le bruit d'hélicoptère (Lynx) [81], et non stationnaires comme le bruit de brouhaha (Voice babble noise) [81] et un bruit enregistré à la gare Saint-Lazare.

[10] Cela veut dire que tout le processus de compensation des bruits (estimation des bruits et leurs compensations) se fait automatiquement, sans introduction d'information *a priori*.

[11] $\Delta$*cepstre* et $\Delta\Delta$*cepstre*.

[12] Contrairement à d'autres techniques comme la combinaison parallèle de modèles [22, 23].

En plus de ces bruits additifs, différents bruits convolutifs ont été compensés par cette technique. Quel que soit le type de bruit affectant les données de test et quelle que soit son intensité, nous avons toujours observé des gains considérables en terme de taux de reconnaissance, surtout dans le cas des bruits stationnaires et dans le cas de rapports signal sur bruit faibles. Une des caractéristiques principales de cette technique de compensation est qu'elle ne dégrade pas les performances du système même en cas d'absence de bruits additif ou convolutif. Le succès de cette technique n'est pas étonnant dans la mesure où on estime la densité de probabilité de la parole bruitée avec très peu d'approximations[13].

Les nombreuses publications concernant le problème de la compensation des différences entre les conditions d'apprentissage et celles du test, supposent généralement que les données d'apprentissage sont propres et les données de test bruitées. Dans la pratique cette hypothèse est rarement vérifiée. Dans cette thèse nous proposons une extension à la combinaison parallèle de données permettant de compenser conjointement les bruits additifs et convolutifs de test et d'apprentissage. L'approche consiste à adapter les paramètres des modèles acoustiques (dans notre cas, ce sont des MMC) en utilisant le bruit présent en phase de test et à compenser les trames représentant les données de test en utilisant le bruit présent en phase d'apprentissage. Cette approche constitue une autre originalité de ce travail.

Une étude expérimentale a été réalisée pour évaluer l'apport de cette nouvelle technique de compensation des bruit additifs et convolutifs présents en phases de test et (ou) d'apprentissage. A la suite de cette étude nous concluons que la compensation du bruit d'apprentissage et de test fondée sur le modèle du canal de transmission $y = (s+n)*h$ apporte un gain significatif par rapport aux résultats de base (sans compensation) et aux résultats avec compensation du bruit de test seulement. Ceci est valable pour les deux types de bruits d'apprentissage stationnaire et non stationnaire. Le gain obtenu par compensation du bruit d'apprentissage est particulièrement important quand les données de test sont propres. Une caractéristique importante de cette technique est qu'elle améliore considérablement les performances du système de reconnaissance en présence de bruits et ne dégrade pas les performances en cas d'absence de bruits dans l'apprentissage ou dans le test.

#### Compensation des autres variabilités

La compensation des variabilités résiduelles non représentées dans le modèle du canal de transmission peut se faire d'une manière aveugle. Pour réaliser cette adaptation aveugle nous nous sommes inspirés de l'approche développée par C.J. Leggetter et P.C. Woodland [45] qui consiste à appliquer des transformations linéaires aux moyennes et aux variances des gaussiennes des MMC afin de les adapter aux données de test. Nous avons étendu cette technique à des transformations plus complexes prenant en compte les moments d'ordre 1 et 2 des gaussiennes.

[13] Soit $f_{\lambda_s}(\mathbf{s}|\mathbf{w})$ la densité de probabilité du signal propre $s$, étant donné le message $\mathbf{w}$ (la phrase prononcée), $\lambda_s$ désigne l'ensemble des paramètres définissant $f$ ; $f$ est une fonction probabiliste d'une chaîne de Markov. La densité de probabilité du signal bruité $f_{\lambda_y}(\mathbf{y}|\mathbf{w})$ est estimée en utilisant la densité de probabilité du signal propre $f_{\lambda_s}(\mathbf{s}|\mathbf{w})$, la densité de probabilité du bruit $f_{\lambda_{n*h}}(\mathbf{n}*\mathbf{h})$ ($n$ est le bruit additif, $h$ est le bruit convolutif) et la relation existant entre le signal propre et le signal bruité $G(s) = y = (s+n)*h$.

L'adaptation aveugle peut être vue comme une technique permettant de corriger (partiellement) les erreurs dues aux éventuelles approximations commises par la combinaison parallèle de données. Ces approximations peuvent être liées soit à l'insuffisance du modèle du canal de transmission utilisé, soit à la technique de combinaison elle-même. Ce point de vue est présenté dans la Figure 1.2 : l'apprentissage sur les signaux propres $\mathbf{s}$ résulte en des modèles acoustiques notés $\mathbf{M}$. En utilisant la CPD, on estime et on compense les bruits additif et convolutif affectant le signal de test, $\mathbf{y} = (\mathbf{s} + \mathbf{n}) * \mathbf{h}$. Ceci résulte en des modèles acoustiques déterminés en fonction de $\mathbf{n}$, $\mathbf{h}$ et $\mathbf{M}$ que nous notons $\mathbf{f}(\mathbf{M}, \mathbf{n}, \mathbf{h})$. Pour compenser les variabilités résiduelles nous estimons des transformations $\mathbf{W}$ qui maximisent la probabilité que les données de test soient générées par les modèles acoustiques adaptés. Les modèles adaptés sont obtenus par l'application des transformations $\mathbf{W}$ aux modèles $\mathbf{f}(\mathbf{M}, \mathbf{n}, \mathbf{h})$, ce qui résulte en des modèles plus représentatifs des données de test que nous notons $\mathbf{g}(\mathbf{f}(\mathbf{M}, \mathbf{n}, \mathbf{h}), \mathbf{W})$. Enfin, le message linguistique caché dans le signal bruité $\mathbf{y}$ est décodé en utilisant les modèles $\mathbf{g}(\mathbf{f}(\mathbf{M}, \mathbf{n}, \mathbf{h}), \mathbf{W})$.

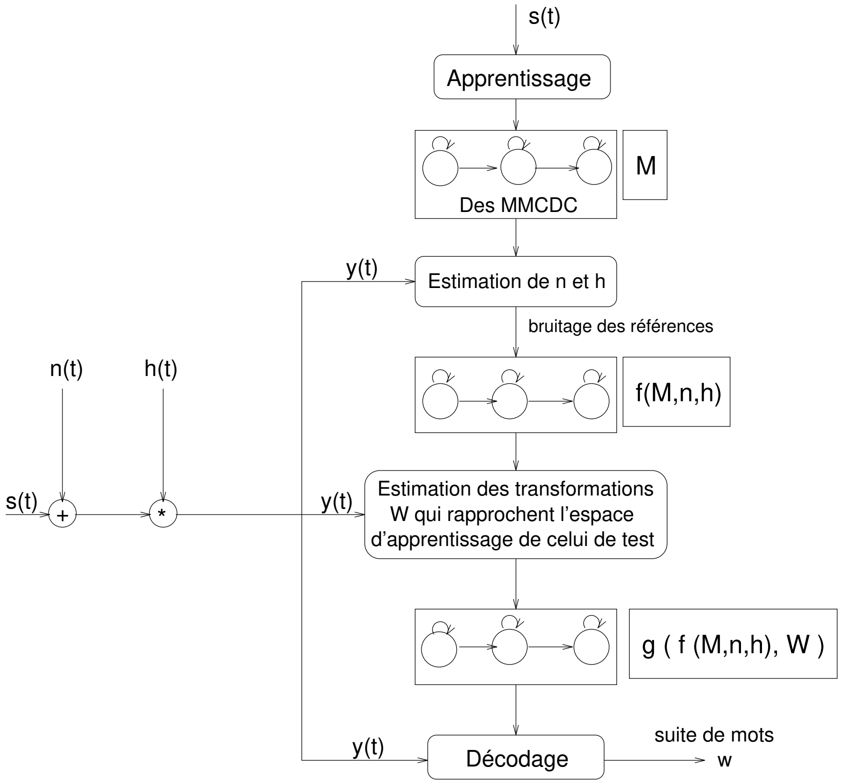



**Figure 1.2 :** *Stratégie d'adaptation : elle est composée de deux étapes principales, la première est la compensation explicite des bruits additif et convolutif n et h. La deuxième consiste en une adaptation aveugle utilisant des transformations W appliquées sur les modèles afin de rapprocher les deux espaces acoustiques d'apprentissage et de test. Ici nous supposons que les données d'apprentissage sont propres.*

### 1.2.3 Niveau de la qualité du signal

Le débruitage peut être utilisé soit pour augmenter la robustesse des systèmes de reconnaissance au bruit ambiant, soit pour améliorer la qualité d'un signal de parole dégradé par la présence d'un bruit ambiant.

Les techniques évoquées jusqu'à présent visent la robustesse des systèmes de reconnaissance aux différentes variabilités acoustiques, et ne nécessitent pas l'estimation du signal propre correspondant au signal bruité enregistré. Cependant dans certaines applications on s'intéresse à la qualité du signal acoustique et non pas au message linguistique qu'il porte. Dans ce cadre nous avons développé une technique de débruitage fondée sur un filtrage dépendant de la gaussienne. C'est-à-dire que le filtre appliqué à une trame donnée du signal bruité est calculé en fonction de la gaussienne qui a pu générer cette trame (la gaussienne la plus probable *a posteriori*). Ces filtres sont estimés en utilisant les modèles appris sur la parole propre (des MMCAR) et le modèle correspondant au bruit (gaussienne AR) [18]. Afin de rendre l'estimation des modèles correspondant à la parole propre plus correcte nous combinons cette technique avec celle de la combinaison parallèle de données. Autrement dit nous utilisons deux systèmes, le premier est un système de reconnaissance classique permettant de faire l'appariement entre les trames du signal bruité et les gaussiennes des MMC du système, à chaque gaussienne dans le système classique correspond une gaussienne auto-régressive dans le second système[14] permettant d'estimer le filtre qu'il faut appliquer à toutes les trames alignées avec cette gaussienne.

Cette technique de débruitage donne des résultats satisfaisants en terme d'amélioration de la qualité du signal. Sa principale caractéristique est l'absence du bruit musical qui constitue le principal défaut de la soustraction spectrale [9]. Nous montrons que le débruitage par filtrage dépendant de la gaussienne permet d'améliorer légèrement les résultats de reconnaissance par rapport à la combinaison parallèle de données. Il faut cependant garder à l'esprit que ce gain n'est pas obtenu par débruitage seul. L'estimation des filtres s'appuie sur les résultats obtenus par reconnaissance avec des modèles adaptés en utilisant la CPD. Cette vision des choses est sûrement à exploiter plus profondément dans des travaux futurs afin de déterminer la combinaison optimale entre méthodes de bruitage et débruitage en fonction du signal à traiter.

## 1.3 Plan de la thèse

Dans le chapitre 2, nous décrivons les modèles utilisés pour la reconnaissance de la parole. Ainsi, nous présentons les techniques de paramétrisation du signal les plus utilisées en reconnaissance de la parole (LPCC, PLPCC, MFCC) ; ensuite nous introduisons les Modèles de Markov Cachés (MMC). Nous décrivons la théorie d'estimation des paramètres des MMCDC (MMC à Distributions Continues) vus

[14] Le second système est fondé sur des MMC Auto-Régressifs, il est estimé sur le même corpus que le système classique (fondé sur le cepstre), l'apprentissage est réalisé une seule fois (celui du système classique), seules les formules d'estimation des paramètres sont différentes.

comme des générateurs de données incomplètes, d'où l'utilisation de l'algorithme EM (Expectation and Maximisation) [15].

Le chapitre 3 montre l'influence, en terme de taux d'erreur sur les mots, de la présence de bruits en phase d'apprentissage et de test, sur les différentes techniques de paramétrisation présentées dans le chapitre 2. Nous montrons aussi, grâce aux spectrogrammes, les caractéristiques spectrales de différents bruits (additif). Ces bruits proviennent généralement de la base de données NOISEX-92 [81]. Nous avons effectué différentes expériences de reconnaissance afin de comparer leurs influences sur les performances du système de reconnaissance dans le cadre d'une analyse spectrale de type MFCC.

L'objectif principal de ce chapitre expérimental est de montrer que quel que soit le bruit affectant les données de test, le système de reconnaissance de la parole se trouve sérieusement perturbé. Il est donc indispensable de mettre en oeuvre des techniques permettant de compenser les effets du bruit pour garantir une bonne robustesse d'un système de reconnaissance lors d'une application dans des conditions acoustiques d'utilisation difficiles.

Le chapitre 4 présente un état de l'art des techniques qui ont été envisagées pour rendre les systèmes de reconnaissance de la parole plus robustes aux différentes variabilités acoustiques. Nous proposons une classification de ces techniques et nous montrons leurs avantages et leurs inconvénients. Cette étude nous permettra de faire le choix des axes de recherche à suivre.

Dans le chapitre 5, nous décrivons une technique de compensation des bruits additif et convolutif fondée sur la Combinaison Parallèle de Modèles (CPM) [22, 23]; nous présentons les avantages et les inconvénients liés à cette technique. Dans le cadre de la CPM, nous proposons une technique pour adapter les paramètres statistiques dynamiques et une technique permettant de compenser conjointement les bruits additifs et convolutifs. Comme alternative à la combinaison parallèle de modèles, nous proposons une nouvelle approche de compensation fondée sur la Combinaison Parallèle de Données (CPD). Cette technique permet de compenser, d'une manière itérative le bruit additif et convolutif affectant les données de test [56]. La combinaison parallèle de données procède à l'adaptation des paramètres statistiques statiques et dynamiques avec peu d'approximations par rapport aux techniques de compensation proposées dans le cadre de la PMC.

Jusque là nous avons supposé les données d'apprentissage propres et les données de test bruitées. L'hypothèse que les données d'apprentissage sont propres est rarement vérifiée dans la pratique. Nous proposons ainsi dans le chapitre 5 une technique itérative pour compenser les bruits additif et convolutif présents en phase de test et les bruits additif et convolutif présents en phase d'apprentissage. Des résultats expérimentaux montrent la capacité de notre approche à compenser les bruits additifs et convolutifs présents en phases de test et d'apprentissage, tout en gardant les mêmes performances en cas d'absence de bruits.

Dans le chapitre 6, nous présentons une technique de débruitage fondée sur un filtrage dépendant de la gaussienne. C'est-à-dire que le filtre appliqué à une trame donnée du signal bruité est calculé en fonction de la gaussienne qui a pu générer cette trame (la gaussienne la plus probable *a posteriori*). Ces filtres sont estimés en utilisant le critère du maximum *a posteriori*. Les équations des filtres estimées font intervenir les modèles appris sur la parole propre (des MMC Auto-Régressifs : AR) et le modèle correspondant au bruit (gaussienne AR) [18]. Afin de rendre l'estimation des modèles AR correspondant à la parole propre plus correcte nous combinons cette technique avec celle de la combinaison parallèle de données décrite dans le chapitre 5.

Nous montrons dans le chapitre 6 que les résultats en terme de taux de reconnaissance du système utilisant le signal débruité avec les modèles appris sur les signaux propres sont meilleurs que ceux du système utilisant le signal bruité avec les modèles obtenus par la CPD. Théoriquement les deux approches bruitage et débruitage sont équivalentes si elle sont toutes les deux optimales[15]. Dans le cas où les modèles obtenus par la CPD ne sont pas optimaux au sens du maximum de vraisemblance, l'approche par débruitage peut être meilleure en terme de taux de reconnaissance.

Les techniques de compensation que nous avons présentées jusqu'à présent sont fondées sur le modèle du canal de transmission[16]. Ce modèle n'inclut pas toutes les variabilités acoustiques (effet Lombard, changement de locuteur, *etc*). Dans le chapitre 7 nous présentons une technique d'adaptation ne supposant aucun modèle *a priori*, mais utilisant un critère d'optimisation fondé sur le maximum de vraisemblance pour estimer des transformations à appliquer sur les modèles acoustiques (MMCDC) du système de reconnaissance. Les transformations utilisées sont généralement des régressions linéaires estimées en utilisant le maximum de vraisemblance d'où le nom en anglais *Maximum Likelihood Linear Regression* [45]. Les transformations concernant les moyennes utilisées dans [45] sont de type : $f(x) = \mathbf{A}x + \mathbf{B}$. Dans ce chapitre nous proposons une extension de cette technique permettant d'utiliser des transformations plus complexes de type : $f(x) = \mathrm{A}xx' + \mathrm{B}x + \mathrm{C}$. Nous présentons aussi une technique d'adaptation permettant d'adapter conjointement les moyennes et les variances. Enfin des résultats expérimentaux sont présentés.

Finalement, le chapitre 8, contient un résumé des résultats, des conclusions et des suggestions pour les travaux de recherche futurs.

[15] Bruitage au sens du maximum de vraisemblance et débruitage au sens du maximum *a posteriori*.

[16] $y = (s + n) * h$, $y$ est le signal observé, $s$ est le signal propre correspondant, $n$ est le bruit additif et $h$ est le bruit convolutif.

# Chapitre 2

# Modèles pour la reconnaissance de la parole

## 2.1 Représentations spectrales

Le signal numérisé est trop redondant et variable pour servir directement dans une application de reconnaissance de la parole. Il doit être traité de manière à extraire au mieux l'information nécessaire et suffisante à la caractérisation de son contenu linguistique. Une représentation traditionnelle pour le traitement et l'interprétation du signal de parole est la représentation temps-fréquence. Elle montre l'évolution temporelle de l'information fréquentielle du signal de parole. L'adéquation de cette représentation provient du fait que les propriétés spectrales du signal de parole restent stables (stationnaires) durant des intervalles de temps relativement courts (quelques dizaines de ms). On peut donc voir le signal de parole comme une suite de formes spectrales, chaque forme spectrale représente un intervalle de temps de 10 à 20ms. Ces formes spectrales contiennent toujours de l'information redondante, inutile et nuisible dans une application de reconnaissance de la parole. En effet, dans le signal de parole, on peut distinguer plusieurs types d'information ; l'information de nature verbale[1] et l'information de type prosodique. Dans une application de reconnaissance de la parole, l'information de type verbale doit être conservée par l'étage de l'analyse, tandis que l'information de nature prosodique doit être éliminée. Une fois isolée, l'information linguistique peut être codée avec un débit très faible par rapport au signal d'origine. Il est donc nécessaire de mettre en aval du module d'acquisition du signal un module de paramétrisation dont le rôle est de transformer le signal temporel en une suite de vecteurs de paramètres. Ce sont ces vecteurs de paramètres qui vont être utilisés pour l'apprentissage et le test du système de reconnaissance de la parole.

Dans ce chapitre nous décrivons les trois principales techniques de paramétrisation qui ont montré un grand intérêt dans les applications de reconnaissance automatique de la parole. Ces techniques de paramétrisation sont : l'analyse MFCC (*Mel scaled Frequency Cepstrum Coefficients*), l'analyse LPCC (*Linear Predictive*

[1] Dans un signal de parole correspondant à l'élocution d'une phrase donnée, l'information verbale est la suite de mots qui forment la phrase.

*Cepstrum Coefficients*), et l'analyse PLPCC (*Perceptually-based Linear Prediction Cepstrum Coefficients*).

Avant toute analyse du signal de parole, la procédure communément utilisée est la suivante :

- filtrage passe-bas anti-repliement à 8 kHz
- échantillonnage à 16kHz et quantification sur 16bits
- préaccentuation par un filtre d'équation $1 - 0.95z^{-1}$ pour aplatir le spectre moyen du signal qui décroît de 6 dB par octave.
- élimination de la composante continue.
- fenêtrage

La taille $L$ de la fenêtre d'analyse doit être supérieure ou égale au double du pas d'analyse. Dans nos expériences, nous allons utiliser un pas d'analyse de 10ms : à une fréquence d'échantillonnage de 16kHz, cela correspond à un pas d'analyse de 160 points. Nous avons choisi une fenêtre d'analyse de taille $L$ égale à 480 points, c'est-à-dire trois fois le pas d'analyse (30ms). Le choix d'une telle taille est justifié par le fait que la fenêtre d'analyse doit pouvoir contenir au moins deux périodes de la fréquence fondamentale des voix les plus basses, on a ainsi $L \geq 28.5ms$ pour une fréquence fondamentale à 70 Hz. L'application d'un filtre de préaccentuation améliore l'estimation spectrale en présence de bruit.

Le gabarit de la fenêtre appliquée au signal est choisi de manière à minimiser la discontinuité aux extrémités de chaque trame. La fenêtre la plus utilisée est celle de Hamming :

$$w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1})\right), \quad 0 \leq n \leq N-1. \tag{2.1}$$

### 2.1.1 L'analyse LPCC (*Linear Predictive Cepstrum Coefficients*)

La théorie du codage par prédiction linéaire appliquée à la parole figure parmi les techniques les plus utilisées en traitement automatique de la parole [54]. Dans ce paragraphe nous allons décrire le principe de cette technique.

**principe :**

La prédiction linéaire du signal de parole s'appuie sur la corrélation existante entre les échantillons adjacents du signal de parole. Ainsi l'échantillon $s(n)$ à l'instant $n$ peut être prédit approximativement ($\hat{s}(n)$) par la connaissance des $p$ échantillons le précédant :

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i), \tag{2.2}$$

où les coefficients $a_i$ sont constants sur une fenêtre d'analyse, cette approximation devient une égalité en incluant un terme d'excitation, $Gu(n)$ :

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n), \tag{2.3}$$

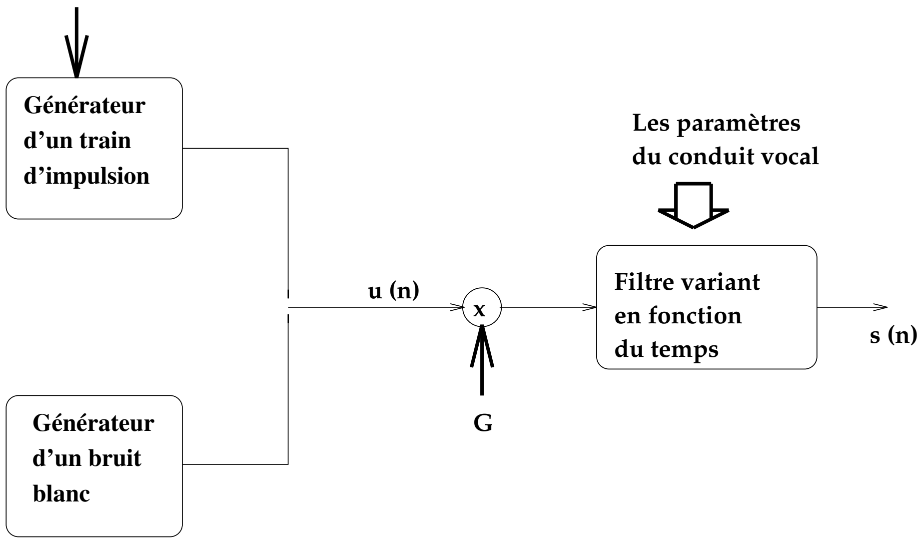



FIG. 2.1 - Modèle de synthèse de la parole fondé sur l'analyse LPC

où $u(n)$ est le signal d'excitation normalisé, G est le gain de cette excitation. La transformée en z de cette égalité donne :

$$GU(z) = (1 - \sum_{i=1}^{p} a_i)S(z), \tag{2.4}$$

ce qui implique :

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{G}{A(z)}, \tag{2.5}$$

L'equation 2.5 peut être interprétée comme suit : le signal $s(n)$ est le résultat de l'excitation du filtre tout-pôles $H(z) = \frac{G}{A(z)}$ par le signal d'excitation $u(n)$. En prédiction linéaire, la source d'excitation peut simplement être modélisée par un générateur d'impulsion dans le cas des sons voisés et un générateur de bruit blanc dans le cas des sons non voisés. La théorie acoustique prédit que le filtre associé au conduit vocal ne contient que des pôles pour les sons non nasalisés, et des pôles et des zéros pour les sons nasalisés. Il est cependant possible d'approximer une fonction de transfert pôles/zéros par un filtre tout-pôles en augmentant le nombre de pôles. La Figure 2.1 montre un modèle de synthèse de la parole fondé sur le modèle LPC.

Le problème est donc d'estimer les coefficients du filtre prédicteur à partir des échantillons de parole. Puisque le signal évolue dans le temps, cette estimation doit être répétée pour chaque fenêtre d'analyse. Définissons le signal d'erreur de prédiction $e(n)$ comme suit :

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{2.6}$$

Pour mettre en évidence l'évolution temporelle du signal de parole, on note $s_n(m) = s(n+m)$ et $e_n(m) = e(n+m)$, le signal vocal et le signal d'erreur à court terme. Les coefficients $a_i$ sont déterminés en minimisant l'erreur quadratique moyenne :

$$E_n = \sum_m e_n^2 = \sum_m \left[ s_n(m) - \sum_{i=1}^{p} a_i s_n(m-i) \right]^2 \tag{2.7}$$

Cette minimisation de l'erreur peut être résolue en dérivant $E_n$ par rapport à chaque coefficient $a_k$ et en égalisant avec zéro :

$$\frac{\partial E_n}{\partial a_k} = 0, \quad k = 1, 2, \ldots, p \tag{2.8}$$

ce qui est équivalent à :

$$\sum_m s_n(m-k)s_n(m) = \sum_{i=1}^{p} \hat{a}_i \sum_m s_n(m-k)s_n(m-i). \tag{2.9}$$

En remarquant que les termes $\sum_m s_n(m-k)s_n(m-i)$ correspondent à la covariance à court terme du signal $s_n(m)$ :

$$\phi_n(k,i) = \sum_m s_n(m-k)s_n(m-i) \tag{2.10}$$

l'équation 2.9 peut s'écrire dans ce cas :

$$\phi_n(k,0) = \sum_{i=1}^{p} \hat{a}_k \phi_n(k,i) \tag{2.11}$$

Le calcul des coefficients est donc en principe simple, puisqu'il suffit de choisir les limites $[m_0, m_1]$ de la fenêtre d'analyse, de calculer les coefficients $\phi_n(k,i)$ à partir du signal, et de résoudre le système d'équations correspondant. Différentes méthodes pour le calcul des coefficients de prédiction ont été proposées. Les deux méthodes de base sont la méthode dite d'*autocorrélation* et la méthode dite de *covariance*. Ces deux méthodes diffèrent par le choix de la fenêtre $[m_0, m_1]$ pour calculer l'erreur quadratique moyenne. Dans nos expériences, c'est la méthode d'autocorrélation qui est utilisée.

J.D. Markel et A.H. Gray [54] définissent les coefficients cepstraux $c_i$ comme étant les coefficients de développement de Taylor du logarithme du filtre tout-pôles :

$$\ln \left[ \frac{G^2}{|A(z)|^2} \right] = \sum_{-\infty}^{+\infty} c_i z^{-i} \tag{2.12}$$

ce qui donne après dérivation par rapport à $z^{-1}$, multiplication par $z^{-1}$ et ensuite par identification :

$$\begin{aligned} c_0 &= \ln(G^2) \\ c_m &= a_m + \textstyle\sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \le m \le p \\ c_{-m} &= c(m). \end{aligned} \tag{2.13}$$

Les coefficients cepstraux sont beaucoup plus intéressants à utiliser dans une application de reconnaissance que les coefficients LPC eux-mêmes, car il constituent

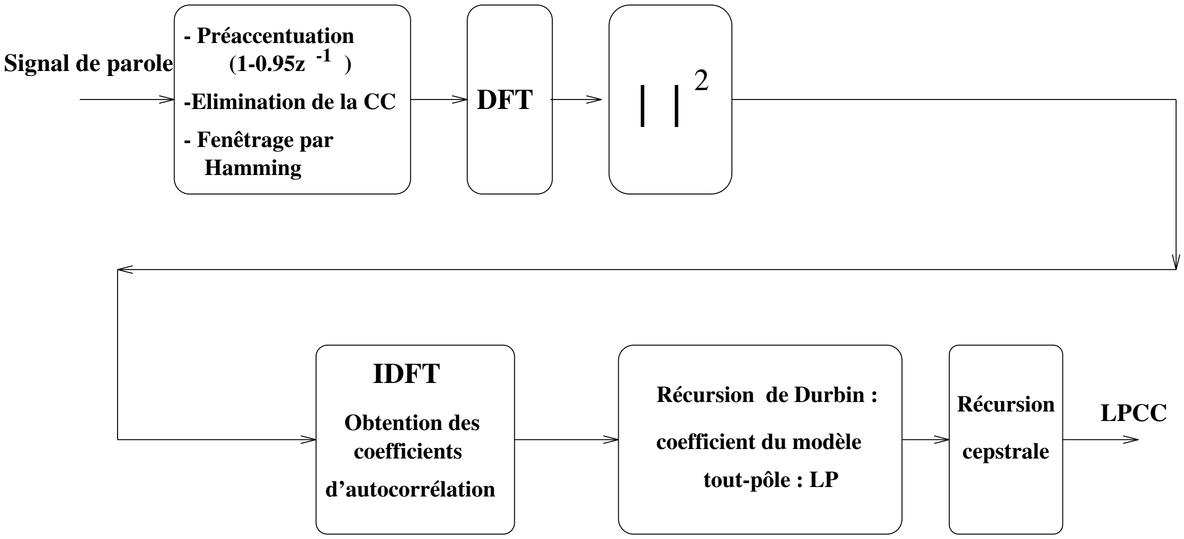



FIG. 2.2 - Calcul des coefficients LPCC (*Linear Predictive Cepstrum Coefficients*)

une suite de variables aléatoires peu corrélées entre elles. Ce qui permet par exemple d'utiliser des matrices de covariances diagonales au lieu de matrices pleines pour leurs moments de second ordre. Une conséquence immédiate de ceci, est que la quantité de données nécessaires à l'estimation de ces matrices est moins importante ; une autre conséquence est que le temps de calcul et la place mémoire nécessaires au décodage en phase d'utilisation sont moins importants avec des matrices de covariance diagonales. Les étapes d'une analyse LPCC sont présentées dans la Figure 2.2.

### 2.1.2 L'analyse PLP (*Perceptually-based Linear Prediction*)

La technique conventionnelle de la prédiction linéaire (LPC) est fondée sur un modèle tout-pôles autorégressif qu'on estime en utilisant une analyse par prédiction linéaire. Ce modèle constitue une approximation de l'enveloppe de la densité spectrale à court terme. Les positions des pôles correspondent aux zones de concentration de l'énergie qui souvent correspondent aux fréquences de résonances du conduit vocal. Cette analyse est plus en accord avec les aspects de la production de la parole qu'avec les aspects de la perception de la parole.

L'analyse PLP [34] est une alternative à l'analyse LPC qui tient compte de trois aspects de la perception :

• **Intégration des bandes critiques :** l'analyse LPC donne la même approximation de l'enveloppe de la densité spectrale pour toute la zone de fréquences utiles, ceci n'est pas en accord avec le fonctionnement de l'oreille où chaque fréquence porte une force sonore résultant de l'intégration de l'information sur une zone de fréquences appelée bande critique. Les bandes critiques sont réparties suivant l'échelle fréquentielle de Bark[2] sur toute la zone de fréquence utilisée. Un Bark correspond à une augmentation de fréquence d'une quantité égale à une bande critique. Pour simuler ce fonctionnement dans le cadre de l'analyse PLP, la densité spectrale est reportée dans l'échelle de Bark, puis convoluée avec une fonction qui simule une bande critique. Cette fonction est la même pour toutes les fréquences dans l'échelle de Bark (à une translation près). Ce qui réduit largement la résolution, surtout dans les hautes fréquences dans l'échelle Hertz. Généralement la nouvelle densité spectrale est échantillonnée à chaque intervalle de 1 Bark.

• **Préaccentuation par courbe d'isosonie :** des expériences de psychoacoustique ont montré que l'oreille possède des caractéristiques non linéaires. En effet, des expériences réalisées par Fletcher et Munson [19] ont mis en évidence que l'intensité perçue, lorsqu'on écoute un son pur d'intensité acoustique constante, varie avec la fréquence de ce son pur. Pour simuler ce phénomène dans le cadre de l'analyse PLP, on multiplie la densité spectrale résultante de l'étape précédente par une fonction de pondération. Il est possible d'estimer cette fonction en reportant sur une abaque des lignes isosoniques le long desquelles un son pur donne à l'oreille une égale sensation d'intensité. Ceci est à l'origine de ce qu'on appelle la sonie considérée comme l'intensité subjective des sons. Dans nos expériences, nous remplaçons cette opération par l'opération de préaccentuation habituelle, c'est-à-dire l'application dans le domaine temporel du filtre passe-haut $(1 - 0.95z^{-1})$.

• **Loi de Stevens :** les deux traitements précédents ne sont pas suffisants pour avoir une correspondance entre l'intensité mesurée et l'intensité subjective (la sonie). La loi de Stevens [76] affirme qu'après avoir réalisé l'intégration des bandes critiques et la préaccentuation, la relation entre l'intensité et la sonie devient :

$$\text{sonie} = (\text{intensité})^{0,33}$$

Dans la figure 2.3, on simule la loi de Stevens par une puissance $\gamma$. Dans son module de pré-traitement, Hermansky [34] a choisi conformément à la loi de Stevens une valeur de $\gamma$ égale à 0.33. Dans le chapitre 3, nous testons différentes valeurs de $\gamma$ afin de choisir la plus optimale en terme de taux de reconnaissance.

### 2.1.3 Analyse par banc de filtres (MFCC)

L'analyse par banc de filtres est une technique qui a été initialement utilisée pour le codage du signal de parole (vocodeur à canaux). Dans ce cadre elle fait partie des techniques de codage par analyse-synthèse à partir du modèle source/filtre. Dans un tel dispositif, le système d'analyse délivre, pour chaque trame :

- un signal binaire voisé/non voisé,
- la fréquence fondamentale (en cas de voisement),

[2] Le passage de Bark en Hertz est obtenu en appliquant la transformation suivante : $f = 600 sinh(z/6)$, où $f$ est la fréquence en Hertz et $z$ est la fréquence en Bark. Inversement, nous avons $z = 6 * \log(f + \sqrt{(1 + f^2)})$.

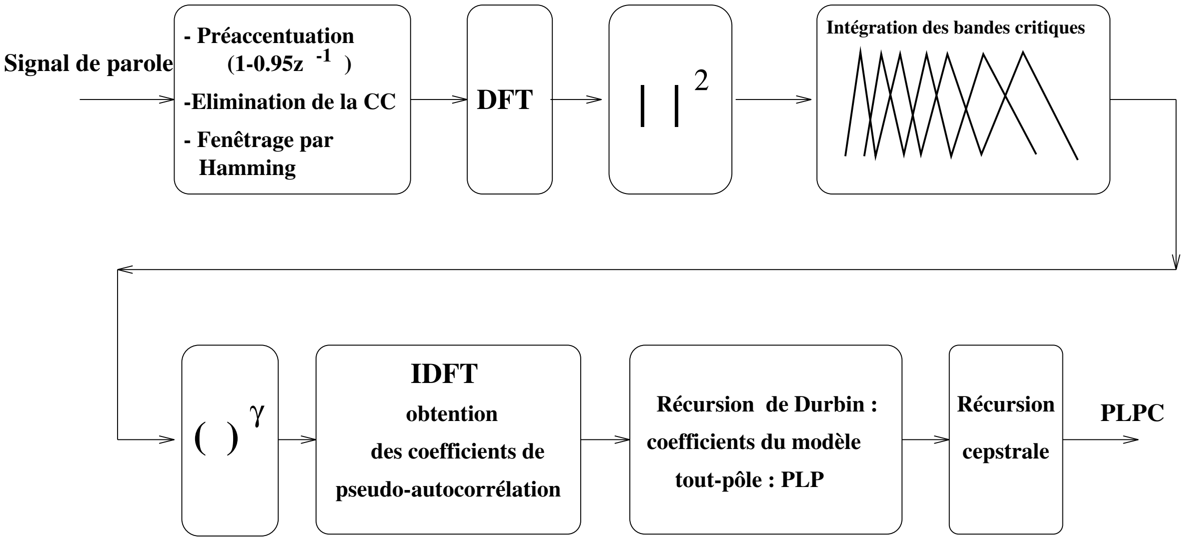



FIG. 2.3 - Calcul des coefficients PLPCC (*Perceptually-based Linear Prediction Cepstrum Coefficients*)

– une estimation de l'enveloppe spectrale par analyse de l'énergie présente dans diverses bandes de fréquence.

Dans le cadre d'une application de reconnaissance de la parole, seule l'estimation de l'enveloppe spectrale est nécessaire. L'analyse par banc de filtres consiste à filtrer le signal par un ensemble de filtres passe-bande. L'énergie en sortie de chaque filtre est attribuée à sa fréquence centrale. Pour simuler le fonctionnement du système auditif humain, les fréquences centrales sont réparties uniformément sur une échelle perceptive. Plus la fréquence centrale d'un filtre est élevée, plus sa bande passante est large. Cela permet d'augmenter la résolution dans les basses fréquences, zone qui contient le plus d'information utile dans le signal de parole. Les échelles perceptives les plus utilisées sont le Mel ou le Bark. Du point de vue des performances des systèmes de reconnaissance de la parole, ces deux échelles sont quasiment identiques. Dans nos expériences, nous avons fait le choix d'utiliser l'échelle MEL.

Le nombre de filtres utilisés dans une telle analyse est choisi de manière empirique ; Zwicker propose 24 filtres [87]. De la même manière que pour le nombre de filtres, le choix de la nature des filtres optimaux du point de vue de la reconnaissance de la parole reste empirique. L'analyse par banc de filtres est réalisée de la manière suivante : le spectre du signal pondéré par une fenêtre (Hamming) est déterminé par une simple FFT (*Fast Fourier Transform*) ; ensuite le filtrage est effectué en multipliant le spectre obtenu après la FFT par les gabarits des filtres. Ces filtres sont en général, soit triangulaires soit sinusoïdaux. Dans nos expériences, nous avons choisi d'utiliser des filtres en cosinus répartis sur une échelle MEL.

Le traitement décrit dans le paragraphe précédent permet d'obtenir une estimation de l'enveloppe spectrale (densité spectrale lissée). Il est possible d'utiliser les sorties du banc de filtres comme entrée dans le système de reconnaissance. Cependant, d'autres coefficients dérivés des sorties d'un banc de filtres, sont plus discriminants, plus robustes au bruit ambiant et moins corrélés entre eux. Il s'agit des coefficients cepstraux dérivés des sorties du banc de filtres répartis linéairement sur l'échelle MEL, on les appelle "**MFCC**" comme "MEL Frequency Cepstrum Coefficients". Le cepstre est défini comme la transformée de Fourier inverse du logarithme de la densité spectrale. Ceci a une interprétation du point de vue de la déconvolution homomorphique : alors que le filtrage linéaire permet de séparer des composantes combinées linéairement, dans le cas de composantes combinées de façon non linéaire (multiplication ou convolution), les méthodes homomorphiques permettent de se ramener au cas linéaire. Pour le signal de parole, la source d'excitation glottique est convoluée avec la réponse impulsionnelle du conduit vocal considéré comme un filtre linéaire :

$$s(t) = e(t) * h(t) \tag{2.14}$$

où $s(t)$ est le signal de parole, $e(t)$ est la source d'excitation glottique et $h(t)$ est la réponse impulsionnelle du conduit vocal. L'application à l'équation 2.14 du logarithme du module de la transformée de Fourier donne :

$$Log|S(f)| = Log|E(f)| + Log|H(f)| \tag{2.15}$$

Par une transformée de Fourier inverse on obtient :

$$s'(cef) = e'(cef) + h'(cef) \tag{2.16}$$

La dimension du nouveau domaine est compatible avec celle du temps et s'appelle *la quéfrence* (*cef*), le nouveau domaine s'appelle le domaine *quéfrentiel*. Un filtrage dans ce domaine s'appelle *liftrage*. Ce domaine est intéressant pour faire la séparation du conduit vocal et de la source d'excitation. En effet, si les contributions relevant du conduit vocal et les contributions de la source d'excitation évoluent avec des rapidités différentes dans le temps, alors il est possible de les séparer par application d'une simple fenêtre dans le domaine *quéfrentiel* (*liftrage* passe-bas pour le conduit vocal). Le conduit vocal possède une contribution fréquentielle assez lisse, qui aboutit à un cepstre basse-quéfrence. Réciproquement, la source possède une contribution qui varie très rapidement dans le domaine fréquentiel, son cepstre sera donc dans les hautes quéfrences. Le domaine quéfrentiel est le domaine idéal pour séparer les deux composantes, car non seulement leur contributions sont séparées dans ce domaine mais aussi elles sont additives. Les étapes d'une analyse MFCC sont présentées dans la Figure 2.4.

Dans le chapitre 3 nous présentons une étude expérimentale permettant de comparer les performances en terme de taux de reconnaissance de chacune des techniques de paramétrisation présentée dans ce paragraphe (MFCC, LPCC, PLPCC). La comparaison est réalisée dans des conditions propres pour l'apprentissage et le test, bruitées pour l'apprentissage et propres pour le test et inversement.

## 2.2 Approche statistique de la RAP : les MMC

Dans le cadre d'une application de la Reconnaissance Automatique de la Parole (RAP), trois principaux facteurs interviennent (voir Figure 2.5) : 1) le locuteur qui

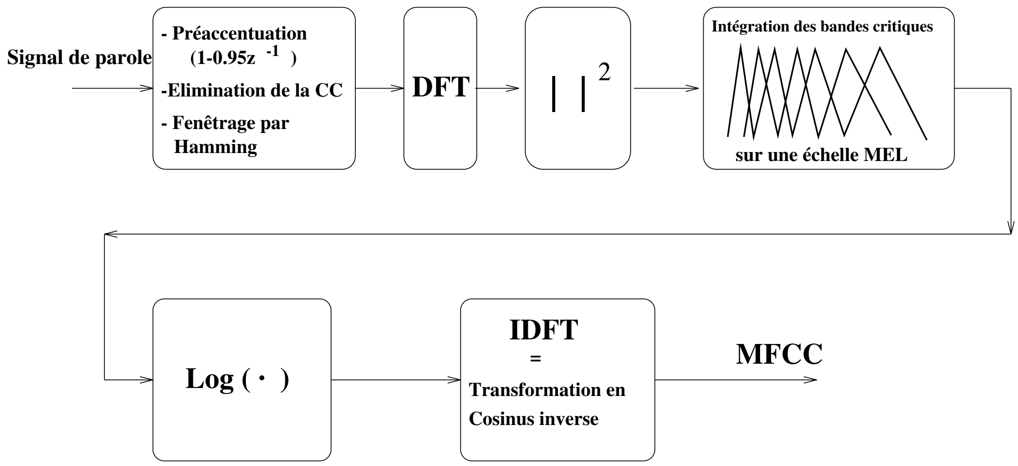



FIG. 2.4 - Calcul des coefficients MFCC (*Mel-scaled Frequency Cepstrum Coefficients*)

à partir d'un message $w$ (suite de mots) produit un signal acoustique $y(t)$, 2) l'analyseur acoustique qui à partir du signal $y(t)$ produit une paramétrisation $\mathbf{y}$, faisant correspondre au signal $y(t)$ une suite de vecteurs contenant l'information pertinente pour la reconnaissance et 3) le décodeur dont le rôle consiste à déterminer à partir de l'observation $\mathbf{y}$ la séquence $\hat{w}$ qui correspond au mieux à $w$. Notons $Y$ le processus stochastique dont les réalisations sont les observations $\mathbf{y}$. Si $\mathbf{y}$ est une observation formée de $T$ vecteurs $(y_1, \ldots, y_T)$ alors $Y$ est une suite de $T$ variables aléatoires $Y_{t,1\leq t\leq T}$, telle que $y_t$ est la réalisation de $Y_t$.

Le décodeur procède à une recherche, parmi toutes les séquences $w$ possibles, celle qui est la plus probable connaissant l'observation $\mathbf{y}$. En d'autres termes, il cherche la séquence $\hat{w}$ telle que :

$$\hat{w} = \operatorname*{argmax}_{w} P(w|Y = \mathbf{y})$$

la formule de Bayes donne :

$$P(w|\mathbf{y}) = \frac{P(Y = \mathbf{y}|w)P(w)}{P(Y = \mathbf{y})}$$

D'où la séquence $\hat{w}$ qui fournit la meilleure interprétation à la réalisation $\mathbf{y}$ est :

$$\hat{w} = \operatorname*{argmax}_{w} P(Y = \mathbf{y}|w)P(w) \tag{2.17}$$

Cette formulation du problème de la reconnaissance de la parole, met en évidence deux termes : le premier, $P(Y = \mathbf{y}|w)$ est le terme lié au canal acoustique, c'est la probabilité d'observer $\mathbf{y}$ sachant que la séquence prononcée est $w$. Le deuxième, $P(w)$ est la probabilité *a priori* que la séquence $w$ soit émise. Ce terme est lié au modèle de langage de l'application en question.

Avec cette modélisation, le décodeur du système de reconnaissance a donc besoin de deux modèles, un pour le canal acoustique permettant de calculer $P(Y = \mathbf{y}|w)$ , et un pour le langage de l'application permettant le calcul de $P(w)$. La qualité d'un

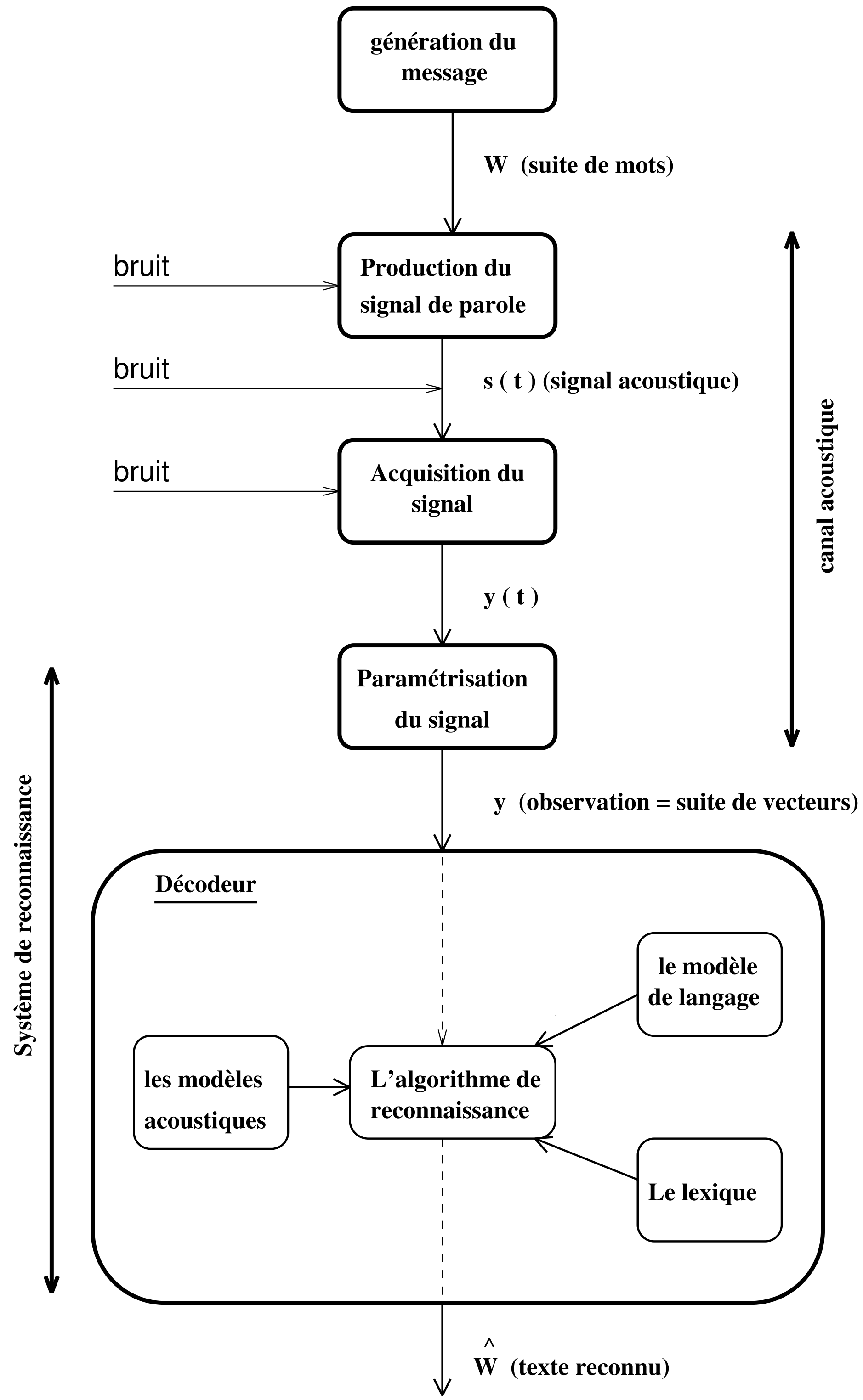



FIG. 2.5 - Les composantes principales du processus de la reconnaissance de la parole, de la génération du message (suite de mots) et production du signal de parole correspondant par le locuteur, jusqu'au décodage de ce signal par le système de reconnaissance, en passant par les modules d'acquisition et de paramétrisation du signal.

système de reconnaissance de la parole peut être caractérisée par la précision et la robustesse de ces deux modèles.

L'espace de toutes les séquences de mots du vocabulaire $\{w\}$ augmente très rapidement avec la taille du vocabulaire. Il convient donc de restreindre la recherche à l'espace des séquences de mots les plus plausibles[3]. Les applications récentes en reconnaissance de la parole utilisent souvent des grammaires stochastiques. Le modèle de langage est un automate d'états fini avec les états représentant les mots du vocabulaire et les arcs des probabilités conditionnelles de transition. Ces probabilités sont apprises sur des corpus de texte servant pour l'apprentissage du modèle de langage de l'application en question.

Considérons le cas d'une séquence $w$ constituée de la suite de mots $w_i$, avec i allant de 1 à L (longueur de la phrase).

$$P(w) = P(w_1 w_2 \dots w_L) = P(w1)\Pi_{i=2}^{L} P(w_i|w_1 \dots w_{i-1})$$

Dans la pratique on approxime $P(w_i|w_1 \dots w_{i-1})$ par $P(w_i|w_{i-1})$, on parle dans ce cas de modèle de langage bigramme, ou par $P(w_i|w_{i-2}w_{i-1})$, on parle dans ce cas de modèle de langage trigramme. Les modèles bigramme et trigramme sont les options les plus courantes; elles impliquent en général peu de restrictions grammaticales, puisque celles-ci portent seulement sur des séquences de 2 ou 3 mots.

### 2.2.1 Modélisation du canal acoustique

L'unité acoustique élémentaire fréquemment utilisée est le phonème : un mot est formé par la concaténation de phonèmes, une phrase est formée par la concaténation de mots. Cependant beaucoup d'autres unités acoustiques sont envisageables, parmi lesquelles on peut citer le phonème en contexte, la syllabe, la dissyllabe qui correspond à un segment de signal partant de la partie stable d'une voyelle à la partie stable de la voyelle suivante. Plus la taille de l'unité augmente, plus son caractère discriminant s'affirme, mais plus il faut d'unités élémentaires pour couvrir la langue. Dans le cadre d'un système de reconnaissance de la parole continue avec un grand vocabulaire, l'unité acoustique la plus utilisée est le phonème dépendant (ou non dépendant) du contexte. Lorsque le contexte phonémique comprend le phonème précédent et le phonème suivant, on parle de modèle de "triphone". On ne modélise pas systématiquement tous les contextes phonémiques, seules les contextes phonémiques dont les comptes dépassent un certain seuil sont modélisés.

Une fois le choix de l'unité acoustique fait, le problème de la modélisation de cette unité se pose : il s'agit de trouver un modèle représentant une séquence de vecteurs acoustiques dont les caractéristiques spectrales et temporelles peuvent varier considérablement entre deux élocutions d'un même locuteur (variabilité *intra-locuteur*), ou d'un locuteur à l'autre (variabilité *inter-locuteur*).

Le signal de parole se présente toujours sous forme d'une succession de zones stables séparées par des transitions relativement brusques. Il est donc tout-à-fait

[3] Cela permet d'éviter au mieux des erreurs de reconnaissance sur des séquences quasi-homophones.

possible de le modéliser par une chaîne de Markov. Pour tenir compte de l'évolution temporelle du signal de parole (causalité du signal de parole), il suffit d'utiliser une chaîne n'autorisant que les transitions de la gauche vers la droite (ce type de chaîne s'appelle gauche-droite). Pour relier ce modèle aux vecteurs de paramètres acoustiques du signal de parole, on associe à chaque état une distribution de probabilité modélisant la génération des vecteurs acoustiques par cet état. La chaîne de Markov n'est plus accessible directement (les états sont cachés par les distributions de probabilité), on parle alors de modèles de Markov cachés (MMC).

Les distributions de probabilité utilisées au niveau de chaque état peuvent être soit discrètes, on parle dans ce cas de Modèle de Markov Caché à Distributions Discrètes (MMCDD), soit continues, on parle dans ce cas de Modèle de Markov Caché à Distributions Continues (MMCDC). Dans ce travail, on s'intéresse principalement au MMCDC.

Un MMCDC est caractérisé par les paramètres suivants :

- $N$ : le nombre d'états dans le modèle.
- $\omega = (\omega_i)_{(1\leq i\leq N)}$ : l'ensemble des états du modèle.
  La chaîne de Markov d'ordre 1 sous-jacente est un processus stochastique $\Omega = (\Omega_1, \ldots, \Omega_T)$, avec $\Omega_i$ une variable aléatoire dont les réalisations portent sur l'ensemble des états $\omega$ et dont la relation de dépendance entre les variables aléatoires $\Omega_i$ est définie par la relation suivante :
  $$P(\Omega_t = \omega_{[t]} | \Omega_1^{t-1} = \omega_{[1]}^{[t-1]}) = P(\Omega_t = \omega_{[t]} | \Omega_{t-1} = \omega_{[t-1]})$$
  où $\Omega_\alpha^\beta = (\Omega_\alpha, \Omega_{\alpha+1} \ldots, \Omega_\beta)$. $\omega_{[t]}$ désigne une réalisation de la variable aléatoire $\Omega_t$. $[t]$ en indice désigne le numéro de l'état occupé à l'instant $t$.
- $A = (a_{ij})_{1\leq i,j\leq N}$ : une matrice $N \times N$, où $a_{ij}$ représente la probabilité de passer de l'état $\omega_i$ à l'état $\omega_j$ ($a_{ij} = P(\Omega_t = \omega_j | \Omega_{t-1} = \omega_i)$)[4]. La matrice $A$ doit satisfaire la contrainte suivante : $\forall j$, $\sum_{j=1}^{N} a_{ij} = 1$.
- $(\pi_i)_{1\leq i\leq N}$ : $\pi_i = P(\Omega_1 = \omega_i)$ est la probabilité d'occupation de l'état $\omega_i$ à l'instant initial (instant 1). La somme des probabilités initiales doit être égale à 1 : $\sum_{i=1}^{N} \pi_i = 1$. La probabilité de générer une séquences d'états $\omega_{[1]}^{[T]}$ s'écrit comme suit :
  $$P(\Omega_1^T = \omega_{[1]}^{[T]}) = \pi_{[1]} \prod_{t=1}^{T-1} a_{[t][t+1]}.$$
- $b_i$ : la densité de probabilité d'observation associée à l'état $\omega_i$.
  La forme de densité de probabilité la plus utilisée est une somme finie de gaussiennes. A l'état $\omega_i$ on associe une densité de probabilité sous forme d'une somme de gaussiennes comportant $M_i$ termes. $b_i$ peut être décrite comme suit :
  - $c_{ik}$ : le poids de la $k^{ième}$ gaussienne de l'état $\omega_i$.

[4] $a_{ij}$ est indépendant de $t$, l'indice $t$ est une variable muette.

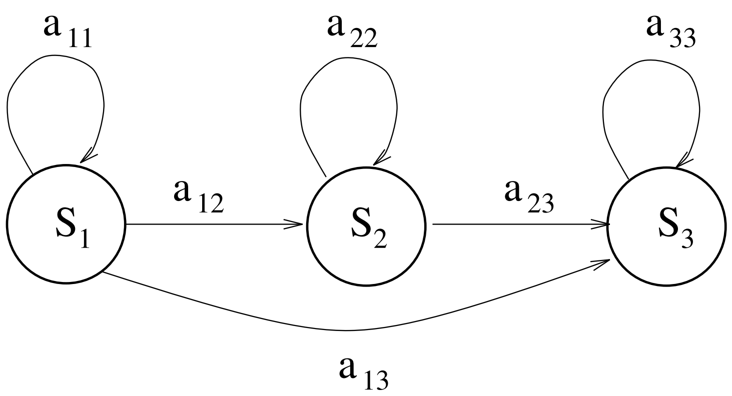



FIG. 2.6 - Un exemple d'un MMC gauche-droite à trois états

- $\mu_{ik}$ : la moyenne de la $k^{ième}$ gaussienne de l'état $\omega_i$.
- $\Sigma_{ik}$ : la matrice de covariance $(n \times n)$ de la $k^{ième}$ gaussienne. où $1 \leq k \leq M_i$, et $n$ est la dimension du vecteur représentant une trame.

La matrice de covariance $\Sigma_{ik}$ peut être soit pleine, pour modéliser la corrélation entre les composantes des vecteurs de paramètres soit diagonale, si on suppose que les composantes sont indépendantes entre elles. Etant donnée la trame de parole à l'instant $t$ (vecteur $y_t$), la probabilité que celle-ci soit générée par la $k^{ième}$ gaussienne du $i^{ième}$ état est :

$$b_{ik}(y_t) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{ik}|^{\frac{1}{2}}} e^{-\frac{1}{2}(y_t-\mu_{ik})'\Sigma_{ik}^{-1}(y_t-\mu_{ik})} \tag{2.18}$$

La probabilité pour que $y_t$ soit généré par le $i^{ième}$ état est :

$$b_i(y_t) = \sum_{k=1}^{M_i} c_{ik} b_{ik}(y_t) \tag{2.19}$$

Un MMCDC est complètement défini par le nombre d'états $N$, la matrice de transition $A$, les probabilités initiales $\{\pi_i\}$ et les densités de probabilité $\{b_i\}$. Cet ensemble de paramètres va être désigné le long de ce document par $\theta : \theta = (N, A, \{\pi_i\}, \{b_i\})$.

Soit $\mathbf{y} = (y_1, \ldots, y_T)$ une suite de vecteurs acoustiques, on suppose que $\mathbf{y}$ est une réalisation d'un MMCDC dont l'ensemble des paramètres est $\theta = (N, A, \{\pi_i\}, \{b_i\})$. La probabilité de générer $\mathbf{y}$ connaissant la séquence d'états générateurs $\omega = (\omega_{[1]}, \ldots \omega_{[T]})$ est :

$$P(Y = \mathbf{y} | \Omega_1^T = \omega_{[1]}^{[T]}, \theta) = \prod_{t=1}^{T} P(Y_t = y_t | \Omega_t = \omega_{[t]}, \theta) = \prod_{t=1}^{T} b_{[t]}(y_t)$$

L'objectif en reconnaissance de la parole est de déterminer la séquence d'unités (mots, phonèmes, etc ...) la plus probable connaissant l'observation $\mathbf{y} = (y_1 \ldots y_T)$, c'est-à-dire trouver la séquence $\hat{w}$ telle que :

$$\hat{w} = \arg\max_{w} P(Y = \mathbf{y}|w) P(w)$$

Pour procéder à cette maximisation, il faut trouver un moyen permettant le calcul de $P(Y = \mathbf{y}|w)$. Le terme $P(w)$ donné par le modèle de langage est ignoré dans ce qui suit. Pour cela supposons l'existence d'un modèle acoustique représentant la séquence $w$. Dans notre cas ce modèle est un MMCDC défini comme dans la section précédente. Notons $\theta$ l'ensemble des paramètres de ce modèle représentant $w$.

$$P(Y = \mathbf{y}|w) = P(Y = \mathbf{y}|\theta) = \sum_{\omega^T} P(Y = \mathbf{y}|\theta) P(\omega^T|\theta) \tag{2.20}$$

$$= \sum_{\omega^T} \pi_{\omega_{[1]}} b_{\omega_{[1]}}(y_1) \prod_{t=2}^{T} a_{\omega_{[t-1]}\omega_{[t]}} b_{\omega_{[t]}}(y_t) \tag{2.21}$$

où $\omega^T = \{\omega_{[1]}, \omega_{[2]} \ldots \omega_{[T]}\}$ est l'ensemble de toutes les séquences composés de $T$ états dans le modèle. $[t]$ en indice désigne le numéro de l'état occupé á l'instant $t$. Pour calculer cette probabilité, on peut utiliser l'algorithme bien connu "avant-arrière" (forward-backward).

### 2.2.2 Les paramètres optimaux d'un MMCDC

Pour pouvoir utiliser les MMCDC en reconnaissance de la parole, il faut estimer leurs paramètres : $\theta = (N, A, \{\pi_i\}, \{b_i\})$. Cette phase s'appelle "phase d'apprentissage". L'objectif de l'apprentissage est la détermination des paramètres optimaux d'un modèle (MMCDC), en fixant *a priori* sa topologie ; c'est-à-dire le nombre d'états, les transitions entre les états, le type des distributions acoustiques associées à chaque état de la chaîne *etc* ... [40].

Soit $m$ un MMCDC, soit $\theta$ l'ensemble des paramètres associés au modèle $m$. Supposons que $m$ modélise le message linguistique $w$. Soit $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ une observation correspondant à l'élocution du message $w$. Le critère généralement utilisé pour estimer les paramètres $\theta$ est celui du maximum de vraisemblance qui consiste à déterminer les paramètres $\hat{\theta}$ maximisant la probabilité que l'observation $\mathbf{y}$ (données d'apprentissage) soit générée par le modèle $m$. Ce critère suppose qu'on dispose d'une quantité suffisante de données d'apprentissage pour estimer correctement les paramètres du modèle. Plus le nombre de paramètres du modèle est important, plus la quantité de données d'apprentissage nécessaire à son estimation augmente.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(Y = \mathbf{y}|m) \tag{2.22}$$

$$= \underset{\theta}{\operatorname{argmax}}\, P(Y = \mathbf{y}|\theta). \tag{2.23}$$

Un MMCDC est un cas typique des modèles générant des données incomplètes. Soit $\mathbf{y} = (y_1, \ldots, y_T)$ une observation générée par un MMCDC, $\mathbf{y}$ constitue les données incomplètes, les données manquantes correspondantes sont[5] :

$$\mathbf{z} = ((\omega_{[1]}, k_{[1]}), \ldots, (\omega_{[T]}, k_{[T]})),$$

[5] C'est la raison pour laquelle ce modèle est qualifié de "caché". En effet, étant donnée une observation $\mathbf{y} = (y_1, \ldots, y_T)$ générée par un tel modèle, on est incapable de retrouver (avec certitude) l'état ou la gaussienne qui a généré la trame $y_t$.

où $\omega_{[t]}$ désigne l'état qui a généré $y_t$ et $k_{[t]}$ indique que c'est précisément la $k^{ième}$ gaussienne associée à l'état $[t]$ qui a généré $y_t$. Ainsi les données complètes sont $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ ou $\mathbf{x} = (y_t, (\omega_{[t]}, k_{[t]}))_{(1 \leq t \leq T)}$. A cause de l'incomplétude des données observées, le problème de la maximisation associé à l'équation 2.23 devient très compliqué et il est pratiquement impossible de le résoudre directement. C'est pour cette raison, qu'on utilise l'approche **EM** [15], qui partant d'une estimation initiale $\theta^0$ des paramètres, utilise les étapes **E** (*Expectation*) et **M** (*Maximization*)

- **E** : attribuer des valeurs $\mathbf{z}^l$ aux données manquantes connaissant les données incomplètes $\mathbf{y}$ et $\theta = \theta^l$.
- **M** : trouver une nouvelle valeur $\theta^{l+1}$ des paramètres qui maximise la vraisemblance des données complètes $P(\mathbf{x}^l|\theta)$ avec $\mathbf{x}^l = (\mathbf{y}, \mathbf{z}^l)$.

L'approche **EM** devient un véritable algorithme lorsqu'on spécifie la stratégie du choix des données manquantes au niveau de l'étape **E**. C'est cette étape qui différencie les algorithmes utilisés pour l'estimation des paramètres des modèles générant des données incomplètes. Dans le cadre des MMCDC les données manquantes sont remplacées par :

- leur espérance conditionnelle connaissant $\mathbf{y}$ et $\theta^l$. Dans ce cas l'étape **M** à l'itération $l$ consiste à déterminer la valeur $\theta^{l+1}$ qui maximise la quantité suivante :

  $$E[log(P(\mathbf{x}|\theta)|\mathbf{y}, \theta^l)].$$

  L'algorithme **EM** résultant a été proposé par Baum dans les années 70 [7]. L'algorithme de Baum utilise la procédure "avant-arrière" pour réaliser l'étape **E**, et c'est pour cette raison qu'il est aussi connu sous le nom de *Forward-Backward*.

- leur valeur la plus probable connaissant $\mathbf{y}$ et $\theta^l$. L'algorithme **EM** résultant est connu sous le nom de l'algorithme de **Viterbi**. D'autres stratégies relatives à l'étape **E** fondées sur des tirages aléatoires des données manquantes ont montré leurs supériorités par rapport à l'algorithme de Baum et celui de Viterbi. Cependant elles n'ont pas encore été testées dans le domaine de la reconnaissance de la parole.

Il est important de noter que l'algorithme EM converge vers un maximum local. Sa convergence est donc très dépendante de la valeur initiale $\theta^0$ des paramètres. Pour un mélange de densités, il est généralement initialisé par le *codebook* résultant de l'application de l'algorithme des plus proches voisins (k-means).

Dans la pratique, pour construire les modèles acoustiques correspondants à une nouvelle application, on utilise un système de reconnaissance de la parole existant pour faire la segmentation par rapport aux états (alignement forcé). Dans cette étape on peut utiliser l'algorithme de Viterbi. A partir de cette segmentation des données d'apprentissage par rapport aux états des MMCDC, on applique l'algorithme EM au niveau de chaque état pour déterminer les paramètres du mélange de gaussiennes correspondant. Ce processus est répété jusqu'à la convergence.

Si L'étape $E$ de l'algorithme $EM$, consiste à remplacer les données manquantes par leur espérance conditionnelle connaissant les données observées et les paramètres $\theta^l$ du modèle à l'itération $l$, alors celle-ci peut être vue comme la phase de calcul de la fonction auxiliaire $Q(\theta^l, \theta) = E[log(P(\mathbf{x}|\theta)|\mathbf{y}, \theta^l)]$. L'espérance est calculée sur les données manquantes ; c'est-à-dire les états et les gaussiennes à l'intérieur des états. Notons $K_{[t]}$ la variable aléatoire associée à l'état $\omega_{[t]}$, dont les réalisations portent sur un numéro de gaussienne dans $\omega_{[t]}$. Si $k_{[t]}$ est une réalisation de $K_{[t]}$ alors $k_{[t]} \in 1, \ldots, M_{[t]}$, où $M_{[t]}$ est le nombre de gaussiennes dans l'état $\omega_{[t]}$. Notons $K^T = (K_{[1]}, \ldots, K_{[T]})$ et $k^T = (k_{[1]}, \ldots, k_{[T]})$ une réalisation de $K^T$. $Q(\theta^l, \theta)$ peut donc s'écrire comme suit :

$$Q(\theta^l|\theta) = \sum_{\omega^T} \sum_{k^T} P(\mathbf{y}, \omega^T, k^T|\theta^l) log[P(\mathbf{y}, \omega^T, k^T|\theta)] \tag{2.24}$$

la première sommation est faite sur l'ensemble de toutes les séquences d'états de longueur $T$ possibles dans la chaîne de Markov. La deuxième sommation est faite sur l'ensemble de toutes les séquences de gaussiennes possibles au sein d'une séquence d'états.

L'étape $M$ consiste à déterminer l'ensemble des paramètres $\theta$ qui maximise cette fonction auxiliaire. Cet algorithme fournit de très bons résultats, sa convergence vers la solution optimale est assurée, pour peu que la position initiale des paramètres du modèle soit assez proche de cette solution optimale.

La fonction auxiliaire peut s'écrire comme somme de termes, où chaque terme ne dépend que d'une seule famille de paramètres :

$$\begin{gathered} Q(\theta^l, \theta) = \textstyle\sum_{\omega^T} \sum_{k^T} P(\mathbf{y}, \omega^T, k^T|\theta^l) \times \\ \left\{ log(\pi_{\omega_{[1]}}) + \textstyle\sum_{t=2}^T log(a_{\omega_{[t-1]}\omega_{[t]}}) + \sum_{t=1}^T log(c_{\omega_{[t]}}) + \sum_{t=1}^T log(b_{k_{[t]}}) \right\} \end{gathered} \tag{2.25}$$

La maximisation de la fonction auxiliaire peut donc se faire individuellement, par rapport à chaque famille de paramètres du modèle. Sans entrer dans les détails, voici les formules de réestimation des paramètres :

**1. Les probabilités initiales :**

$$\pi_i^{l+1} = \frac{P(\mathbf{y}, \Omega_1 = i|\theta^l)}{P(\mathbf{y}|\theta^l)}, \quad 1 \leq i \leq N. \tag{2.26}$$

**2. Les probabilités des transitions :**

$$a_{ij}^{l+1} = \frac{\sum_{t=2}^T P(\mathbf{y}, \Omega_{t-1} = i, \Omega_t = j|\theta^l)}{\sum_{t=1}^T P(\mathbf{y}, \Omega_{t-1} = i|\theta^l)}, \quad 1 \leq i \leq j \leq N. \tag{2.27}$$

**3. Les poids des gaussiennes à l'intérieur des états :**

$$c_{ik}^{l+1} = \frac{\sum_{t=1}^T P(\mathbf{y}, \Omega_{[t]} = i, K_{[t]} = k|\theta^l)}{\sum_{t=1}^T P(\mathbf{y}, \Omega_t = i|\theta^l)}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq M_i. \tag{2.28}$$

avec $M_i$ le nombre de gaussiennes dans l'état $i$. $c_{ik}$ est le poids de la $k^{ième}$ gaussienne dans le $i^{ième}$ état.

**3. Les moyennes et les matrices de covariances :**

$$\mu_{ik}^{l+1} = \frac{\sum_{t=1}^{T} P(\mathbf{y}, \Omega_{[t]} = i, K_{[t]} = k|\theta^l) y_t}{\sum_{t=1}^{T} P(\mathbf{y}, \Omega_{[t]} = i, K_{[t]} = k|\theta^l)}, \tag{2.29}$$

$$\Sigma_{ik}^{l+1} = \frac{\sum_{t=1}^{T} P(\mathbf{y}, \Omega_{[t]} = i, K_{[t]} = k|\theta^l)(y_t - \mu_{ik}^l)(y_t - \mu_{ik}^l)'}{\sum_{t=1}^{T} P(\mathbf{y}, \Omega_{[t]} = i, K_{[t]} = k|\theta^l)}, \tag{2.30}$$

$$1 \leq i \leq N, \quad 1 \leq k \leq M_i$$

**<u>Procédures "avant-arrière" :</u>** les formules de réestimation peuvent être calculées en utilisant les procédures "avant-arrière" (*Forward-Backward*). Définissons pour l'état $i$ ($1 \leq i \leq N$) de la chaîne de Markov la probabilité "avant" comme suit :

$$\begin{aligned} \alpha_t(i) &= P(y_1, y_2, \ldots, y_t, \Omega_t = i|\theta) \\ &= \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} b_i(y_t), \\ et & \\ \alpha_0(i) &= \pi_i, \end{aligned} \tag{2.31}$$

de la même manière, définissons la probabilité "arrière" pour l'état $i$ ($0 \leq i \leq N$) de la chaîne de Markov comme suit :

$$\begin{aligned} \beta_t(i) &= P(y_{t+1}, y_{t+2}, \ldots, y_T|\Omega_t = i, \theta) \\ &= \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(y_{t+1}), \\ et & \\ \beta_T(i) &= 1, \end{aligned} \tag{2.32}$$

Définissons la probabilité "avant" pour un état donné $i$ ($1 \leq i \leq N$) et une gaussienne dans cet état $k$ ($1 \leq k \leq M_i$) :

$$\begin{aligned} \gamma_t(i,k) &= P(y_1, y_2, \ldots, y_t, \Omega_t = i, K_t = k|\theta) \\ &= \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} c_{ik} b_{ik}(y_t). \end{aligned} \tag{2.33}$$

La probabilité que le modèle $\theta$ génère $\mathbf{y}$ en passant par l'état $i$ à l'instant $t$ (c'est-à-dire que $y_t$ est générée par l'état $i$) est :

$$P(\mathbf{y}, \Omega_t = i|\theta) = \alpha_t(i)\beta_t(i), \quad 1 \leq t \leq T \tag{2.34}$$

ce qui implique que la probabilité d'observer $\mathbf{y}$ est :

$$P(\mathbf{y}|\theta) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i), \quad 1 \leq t \leq T \tag{2.35}$$

Pour $t = T$, cette équation devient :

$$P(\mathbf{y}|\theta) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.36}$$

La probabilité que le modèle $\theta$ génère $\mathbf{y}$ en faisant la transition de l'état $i$ à l'état $j$ à l'instant $t$ est :

$$P(\mathbf{y}, \Omega_{t-1} = i, \Omega_t = j|\theta) = \alpha_{t-1}(i)a_{ij}b_i(y_t)\beta_t(j). \tag{2.37}$$

La probabilité que le modèle $\theta$ génère $\mathbf{y}$ et que la trame $y_t$ soit générée par la $k^{i\grave{e}me}$ gaussienne du $i^{i\grave{e}me}$ état est :

$$P(\mathbf{y}, \Omega_t = i, K_t = k|\theta) \quad = \quad \gamma_t(i,k)\beta_t(i)$$

Toutes les probabilités utilisées dans les formules de réestimation peuvent être remplacées par leurs valeurs en fonction des probabilités “avant” et “arrière”. Ces dernières sont calculées d'une manière récurrente en utilisant les équations 2.31 et 2.32.

### 2.2.3 Quelques lacunes des MMC

Le moyen le plus commode permettant d'améliorer la modélisation acoustique passe souvent par une augmentation du nombre de paramètres du modèle. Cependant, cette amélioration doit respecter un compromis entre l'augmentation du nombre des paramètres, le gain apporté en performance et le coût de calcul supplémentaire. Par exemple, au sein d'un état d'un MMCDC, pour approximer au mieux la densité de probabilité des trames générées par cet état, il faut modéliser la corrélation entre les composantes de ces trames. Cette modélisation ne peut se faire qu'en utilisant des matrices de covariances pleines ou en augmentant suffisamment le nombre de gaussiennes associées à cet état. Cette augmentation du nombre de paramètres dans le système doit prendre en compte la taille du corpus d'apprentissage : la possibilité d'augmenter le nombre de paramètres dans le modèle augmente avec la taille du corpus d'apprentissage disponible.

Afin d'améliorer la modélisation acoustique tout en gardant un nombre raisonnable[6] de paramètres, on utilise une technique permettant le partage des paramètres. Par exemple, deux états appartenant à deux MMCDC représentant deux unités acoustiques différentes peuvent partager la même distribution. Le partage des paramètres peut être appliqué à différents niveaux, au niveau des modèles (MMCDC), au niveau

[6] Par rapport aux moyens dont on dispose (la taille mémoire vive et secondaire, rapidité du processeur *etc*), et par rapport aux contraintes de l'application (temps réel par exemple).

des états ou encore au niveau de la gaussienne [6, 84, 85]. Il est évident que les modèles obtenus avec cette technique (partage des paramètres) sont moins corrects que ceux qu'on aurait estimés sans le partage. Cependant, l'intérêt d'une telle approche est que la perte d'information causée par la réduction du nombre de paramètres peut être compensée en modélisant d'importantes corrélations comme celles liées au phénomène de la co-articulation.

L'hypothèse que la parole est un processus markovien d'ordre 1 et donc qu'elle peut être modélisée par des MMC d'ordre 1 est incorrecte. Cependant, c'est cette hypothèse simplificatrice qui est à l'origine du succès des MMC pour modéliser la parole[7]. Afin de réduire les effets de cette approximation sans changer les algorithmes d'apprentissage et de décodage, on modélise la dépendance entre les trames successives en introduisant les paramètres dynamiques[8] [21].

Un autre défaut des MMC est celui de la modélisation des durées. En effet, un segment court a une probabilité plus grande d'être généré par un MMC qu'un segment plus long. Quelques tentatives de modélisation de durée ont été proposé [77, 46]. Un modèle de trajectoires stochastiques, prenant en compte les durées a été proposé par Y.Gong et J.P Haton [29]. A l'heure actuelle il n'existe pas de bonne modélisation de durée permettant une technique de décodage efficace.

De manière générale le bruit affecte principalement les densités de probabilité d'observation $\{b_i\}$. Lorsque le bruit devient vraiment gênant pour le locuteur (Effet Lombard), sa vitesse d'élocution peut également changer, ce qui peut modifier les probabilités de transition $\{a_{ij}\}$. Nous n'abordons pas ce dernier cas dans cette thèse. Par la suite, nous nous intéressons surtout aux différentes manières de transformer un MMCDC de paramètres $\theta = (N, A, \{\pi_i\}, \{b_i\})$ en un MMCDC de paramètres $\theta' = (N, A, \{\pi_i\}, \{b_i\}')$ mieux adapté aux conditions acoustiques d'utilisation, en agissant exclusivement sur la famille des densités de probabilités d'émissions $\{b_i\}$.

[7] Car on dispose d'algorithmes de décodage et d'apprentissage très efficaces pour ce type de modèles : l'algorithme de Viterbi pour le décodage et l'algorithme EM ou de Baum pour l'apprentissage.

[8] Des approximations des dérivées premières et secondaires : $\Delta cepstre$ et $\Delta\Delta cepstre$.

# Chapitre 3

# L'influence du bruit sur les systèmes de reconnaissance

Ce chapitre se situe dans le cadre d'une étude expérimentale permettant de choisir parmi les paramétrisations utilisées actuellement dans les systèmes de reconnaissance celle qui donne les meilleures performances dans le cas de signaux propres et qui est en même temps robuste aux dégradations causées par la présence de bruit ambiant. Au-delà de cet aspect, nous montrons dans ce chapitre, la dégradation des performances de reconnaissance en fonction de différents types de bruits (bruits blancs, colorés, stationnaires, non stationnaires).

## 3.1 Robustesse des techniques de paramétrisation au bruit

Afin de comparer la robustesse au bruit[1] des techniques de paramétrisation acoustique du signal présentées dans le chapitre 2 (LPCC, MFCC, PLPCC), nous avons utilisé le système de reconnaissance du LIMSI [?] avec différentes paramétrisations et différents bruits. L'étude de l'effet du bruit sur ces techniques de paramétrisations a été réalisée dans différentes situations : 1- données d'apprentissage et de test bruitées, 2- données d'apprentissage propres et données de test bruitées, 3- données d'apprentissage bruitées et données de test propres, 4- données d'apprentissage et de test propres. Le système utilisé est entraîné sur 22148 phrases prononcées par 460 locuteurs, provenant du corpus MASK [42]. Les phrases prononcées par 450 locuteurs ont été utilisées pour l'apprentissage du système et les phrases prononcées par les 10 locuteurs restants ont été utilisées pour le test. Les données ont été enregistrées par deux microphones simultanément, un microphone de tête ($S_a$ avec un Rapport Signal sur Bruit (RSB) $\simeq$ 35dB) et un microphone de table ($S_b$ avec un RSB $\simeq$ 20dB). Le signal de parole a subi un filtrage passe-bas avec une fréquence de coupure égale à 8kHz et a été échantillonné à 16kHz.

MASK est une application en grandeur réelle de la technologie de la langue, dont l'objectif est de développer un système de réservation de trains et d'information

[1] En terme de taux de reconnaissance.

automatique. Une personne utilisant le système peut demander des informations concernant les voyages en train à travers 500 villes de France. Le système est multi-locuteur, capable de reconnaître la parole continue et spontanée. Le système utilise un vocabulaire de 1500 mots et un modèle de langage *bigramme*.

Trois techniques de paramétrisation ont été comparées, la LPCC, la MFCC et la PLPCC. Les vecteurs paramétriques caractérisant une trame sont composés de 13 coefficients cepstraux auxquels sont ajoutées leurs dérivées premières et secondes (vecteur à 39 paramètres) qui sont estimées tous les 10ms sur une fenêtre de 30ms. La soustraction du cepstre moyen est réalisée sur chaque phrase. 608 phonèmes dépendant du contexte sont modélisés. Chaque phonème dépendant du contexte est un Modèle de Markov Caché à Densité Continue (MMCDC) avec un mélange de gaussiennes comme densité d'observation (typiquement 20 gaussiennes).

Pour toutes les expériences, les modèles ont été appris sur les signaux $S_a$. Deux valeurs du rapport signal sur bruit dans l'apprentissage ont été testées. Un RSB de 35dB qui correspond aux données enregistrées avec $S_a$ (signal propre), et un RSB de 20dB résultant de l'addition du bruit de Lynx aux données enregistrées avec $S_a$. Le bruit du Lynx provient de la base de données des bruits NOISEX-92 [81]. Un bruit enregistré dans la gare Saint-Lazare à Paris a été additionné à $S_b$ avec différents niveaux pour former le signal de test avec trois différents RSB : 20dB, 12dB, 9dB. La première ligne dans chaque table utilise le signal $S_a$ pour le test.

Dans le cadre de la PLPCC, Hermansky [34] ainsi que d'autres auteurs ont choisi conformément à la loi de Stevens [76] une valeur de $\gamma$ égale à 0.33. Dans une application de reconnaissance de la parole, il n'est pas prouvé que 0.33 soit la valeur optimale de $\gamma$ en terme de taux d'erreur sur les mots. Pour montrer l'effet de $\gamma$ du point de vue de la reconnaissance de la parole, nous avons testé quatre valeurs autour de 0.33 : $\frac{1}{7}$, $\frac{1}{5}$, $\frac{1}{3}$, $\frac{1}{2}$.

**Table 3.1 :** LPCC

| Conditions de test | Conditions d'apprentissage | |
|---|---|---|
| | $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
| $S_a$ avec RSB=35dB | 7.7% | 17.4% |
| $S_b$ avec RSB=20dB | 13.67% | 17.6% |
| $S_b$ avec RSB=12dB | 32.3% | 26.3% |
| $S_b$ avec RSB=9dB | 57.4% | 46.3% |

**Table 3.2 :** MFCC

| Conditions de test | Conditions d'apprentissage | |
|---|---|---|
| | $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
| $S_a$ avec RSB=35dB | 5.9% | 11.1% |
| $S_b$ avec RSB=20dB | 11.1% | 14.8% |
| $S_b$ avec RSB=12dB | 27.8% | 25.7% |
| $S_b$ avec RSB=9dB | 51.5% | 49.1% |

**Table 3.3 :** PLPCC avec $\gamma = \frac{1}{7}$

| Conditions de test | Conditions d'apprentissage $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
|---|---|---|
| $S_a$ avec RSB=35dB | 5.9% | 6.4% |
| $S_b$ avec RSB=20dB | 11.6% | 14.0% |
| $S_b$ avec RSB=12dB | 28.48% | 28.1% |
| $S_b$ avec RSB=9dB | 49.6% | 46.5% |

**Table 3.4 :** PLPCC avec $\gamma = \frac{1}{5}$

| Conditions de test | Conditions d'apprentissage $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
|---|---|---|
| $S_a$ avec RSB=35dB | 5.7% | 6.9% |
| $S_b$ avec RSB=20dB | 12.0% | 13.9% |
| $S_b$ avec RSB=12dB | 27.7% | 28.5% |
| $S_b$ avec RSB=9dB | 48.1% | 47.5% |

**Table 3.5 :** PLPCC avec $\gamma = \frac{1}{3}$

| Conditions de test | Conditions d'apprentissage $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
|---|---|---|
| $S_a$ avec RSB=35dB | 5.8% | 7.7% |
| $S_b$ avec RSB=20dB | 12.1% | 16.6% |
| $S_b$ avec RSB=12dB | 26.7% | 27.1% |
| $S_b$ avec RSB=9dB | 50.6% | 46.8% |

**Table 3.6 :** PLPCC avec $\gamma = \frac{1}{2}$

| Conditions de test | Conditions d'apprentissage $S_a$ avec RSB=35dB | $S_a$ avec RSB=20dB |
|---|---|---|
| $S_a$ avec RSB=35dB | 6.2% | 7.61% |
| $S_b$ avec RSB=20dB | 11.9% | 15.42% |
| $S_b$ avec RSB=12dB | 29.6% | 27.9% |
| $S_b$ avec RSB=9dB | 52.0% | 47.7% |

Une comparaison des résultats obtenus en utilisant la LPCC avec ceux obtenus en utilisant la MFCC ou la PLPCC (voir tables 3.1 - 3.6), montre clairement la supériorité de la MFCC et la PLPCC par rapport à la LPCC. Ceci est certainement dû au fait que la LPCC ne tient pas compte des caractéristiques non linéaires de l'oreille, notamment de la résolution fréquentielle qui doit être une fonction logarithmique de la fréquence linéaire (Hertz). En effet, l'information contenue dans les basses fréquences est plus discriminante que celle qui se trouve dans les hautes fréquences. Nous remarquons aussi que la différence, en terme de taux d'erreur, entre la LPCC et les autres techniques de paramétrisation, est plus importante quand les données d'apprentissage sont bruitées et les données de test sont propres : quand le RSB dans l'apprentissage est de 20dB et lorsqu'on teste avec un signal propre, la LPCC résulte en un taux d'erreur sur les mots de 17.4% alors que la MFCC et la PLPCC ($\gamma = \frac{1}{7}$) donnent des taux d'erreur sur les mots de 11.1% et 6.4% respectivement.

Lorsque les données d'apprentissage sont propres, les résultats obtenus par la MFCC et la PLPCC sont pratiquement identiques. En revanche, avec des données d'apprentissage bruitées et des données de test propres, la PLPCC donne de meilleurs résultats : quand le RSB dans l'apprentissage est de 20dB et lorsqu'on teste avec un signal propre, la MFCC résulte en un taux d'erreur sur les mots de 11.1% et la PLPCC ($\gamma = \frac{1}{7}$) donne un taux d'erreur sur les mots de 6.4%, soit une différence relative de 42%.

Les expériences réalisées avec différentes valeurs de $\gamma$ montrent que les valeurs $\frac{1}{7}$ et $\frac{1}{5}$ donnent de meilleurs résultats dans la majorité des configurations. Cependant, nous pensons que la valeur optimale de $\gamma$ dépend des types de bruits présents en phases d'apprentissage et de test.

La compression spectrale en racine a pour effet de lisser la densité spectrale. S.M. Kay [41] montre que plus la densité spectrale d'un processus ARMA est lisse, plus court est le processus AR l'approchant. Ainsi, la valeur optimale de $\gamma$ est obtenue en faisant un compromis entre la perte d'information au niveau des pics spectraux (formants) et le gain d'information au niveau des vallées spectrales. Le gain d'information au niveau des vallées spectrales est indésirable car ces parties du spectre sont les plus sensibles aux distorsions causées par le bruit ambiant. Dans ce sens, la compression spectrale en racine peut être équivalente à l'ajout d'un bruit blanc aux signaux de test et d'apprentissage. Dans ce cas le choix optimal du niveau du bruit blanc qu'il faut ajouter dépend, comme le choix de $\gamma$, du corpus et des types de bruits présents en phases d'apprentissage et de test.

Dans la suite de ce chapitre nous comparons l'effet de différents types de bruits sur le système de reconnaissance. La technique de paramétrisation choisie pour réaliser cette étude comparative est la MFCC. La comparaison est réalisée dans le cas où seules les données de test sont bruitées.

## 3.2 L'effet de différents types de bruits sur la RAP

La plupart des bruits utilisés pour tester les approches de compensation présentées dans cette thèse sont additionnés artificiellement aux signaux de la parole propre. Ceci est dû au fait qu'il est difficile d'avoir les mêmes phrases prononcées par des locuteurs avec différents rapports signal sur bruit. Mis à part l'effet Lombard, additionner artificiellement du bruit sur les signaux simule très bien le cas réel de présence de bruit.

La majorité des bruits utilisés proviennent de la base de données NOISEX-92 [81]. Cette base de données contient, en effet, une variété de bruits avec différentes caractéristiques. Dans ce chapitre, nous décrivons, grâce aux spectrogrammes, les caractéristiques spectrales de 12 bruits différents. 10 de ces bruits proviennent de la base de données NOISEX-92, un bruit enregistré à la gare Saint-Lazare et un bruit obtenu par filtrage passe-bas d'un bruit blanc. Afin de comparer les effets de ces bruits sur le système de reconnaissance de la parole, nous allons utiliser un système de reconnaissance appris sur des signaux propres et testé avec le même corpus de test affecté par les différents bruits avec le même RSB pour tous les bruits.

Trois bruits parmi ceux étudiés dans ce chapitre seront utilisés pour montrer l'apport de nos techniques de compensation présentées dans les chapitres suivants : le bruit d'hélicoptère (Lynx) et le bruit Hautes Fréquences qui peuvent être considérés comme étant stationnaires, le bruit de brouhaha (Voice babble noise) et le bruit enregistré à la gare Saint-Lazare à Paris qui sont des bruits non stationnaires.

### 3.2.1 Résultats expérimentaux

Pour pouvoir comparer les effets des différents bruits sur le système de reconnaissance de la parole, nous avons utilisé le système de reconnaissance du LIMSI décrit dans la section 3.1. Les expériences de cette section utilisent le canal $S_a$ pour le test et l'apprentissage (pas de bruit convolutif). Dans la Figure 3.1 on représente le signal ainsi que le spectrogramme de la phrase *"Je voudrais aller de Paris à Bordeaux"*. Le signal correspondant à cette phrase est considéré comme propre (RSB=35dB). Cette phrase va être utilisée comme "phrase de référence" pour illustrer l'effet des différents bruits sur le signal propre. Lorsque le signal de test est propre (RSB=35dB), le taux d'erreur sur les mots est de **5.9%**.

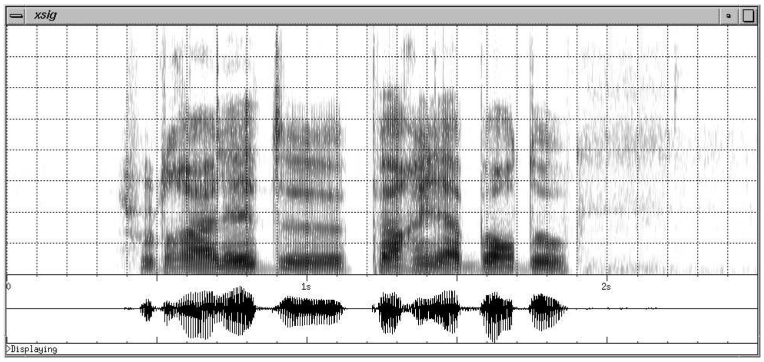


**Figure 3.1 :** dans cette figure on représente le signal temporel et le spectrogramme de la phrase *"Je voudrais aller de Paris à Bordeaux"*. le RSB est de 35dB.

Dans ce qui suit nous présentons les spectrogrammes d'une série de bruits issus généralement de la base de bruits NOISEX-92 [81], ainsi que les spectrogrammes de la phrase de la Figure 3.1 bruitée avec les bruits en question (le rapport signal sur bruit est de 5 dB). Dans toutes les figures l'axe des y représente les fréquences allant de 0 à 8000 Hz :

**Figure 3.2 : Bruit blanc**

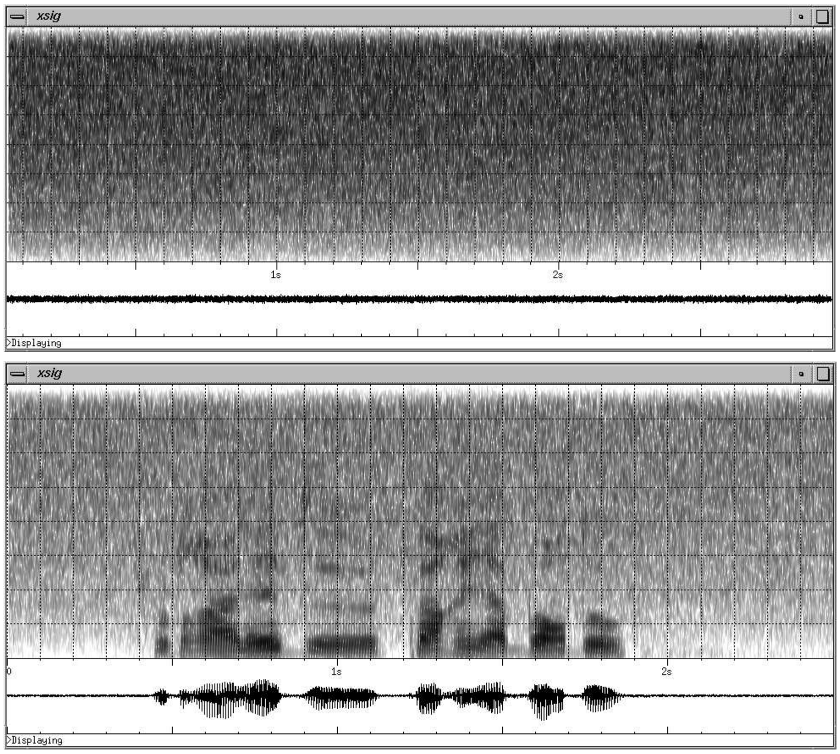



**Figure 3.3 : Bruit de Char (M-109 30km/h)**

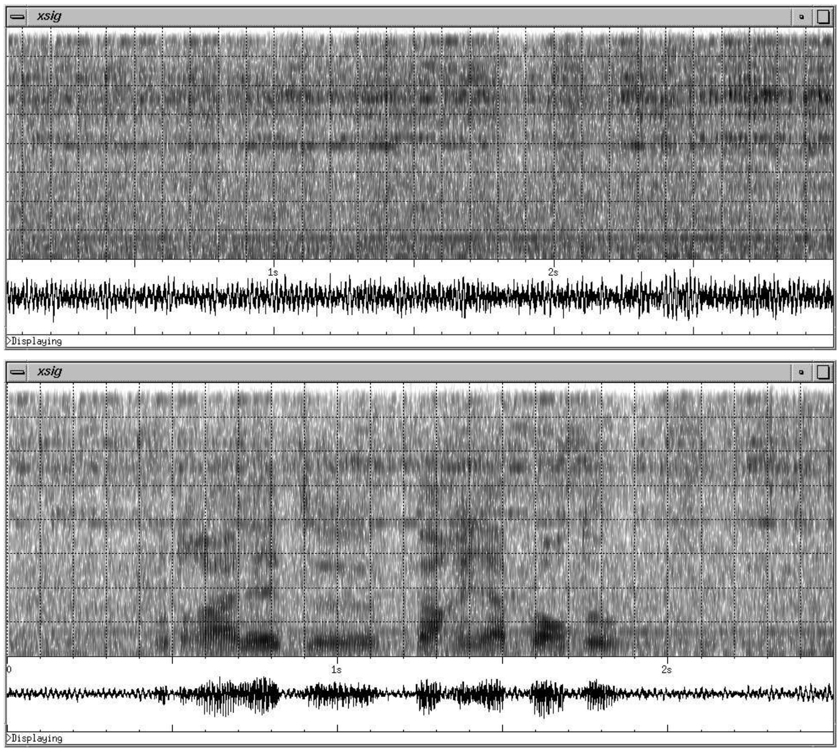


**Figure 3.4 : Bruit d'avion de chasse (190 noeuds 1000 pieds)**

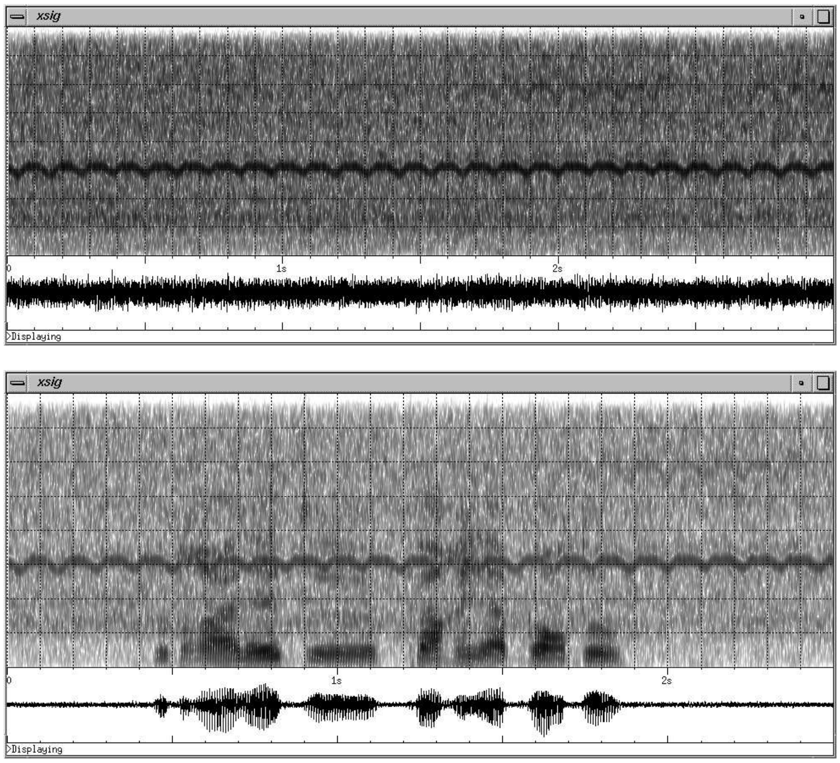



**Figure 3.5 : Bruit de char (Leopard, 70km/h)**

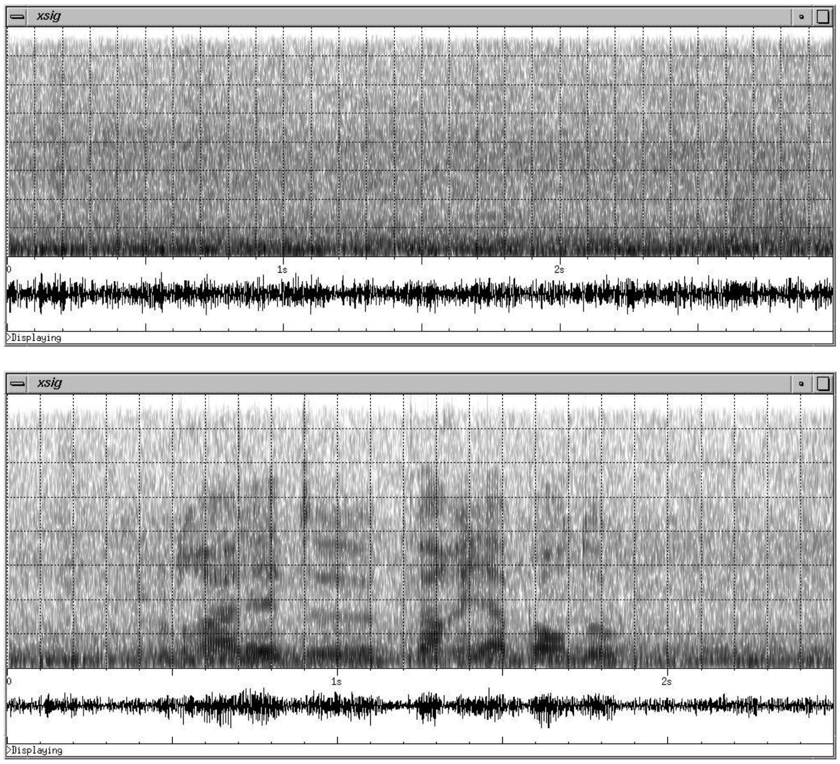


**Figure 3.6 : Bruit d'hélicoptère (Lynx)**

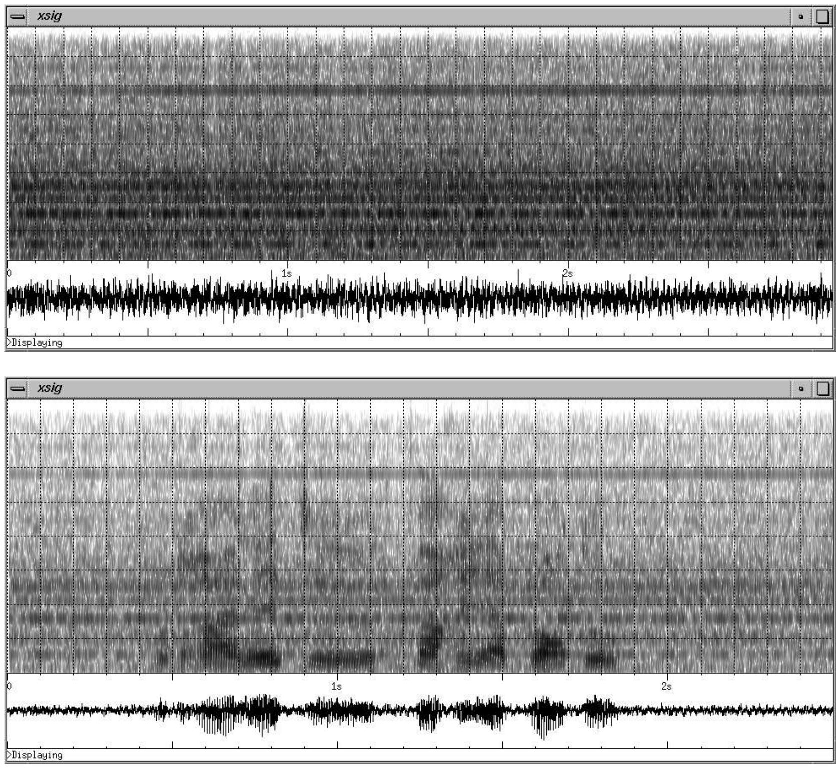



**Figure 3.7 : Bruit Hautes Fréquences**

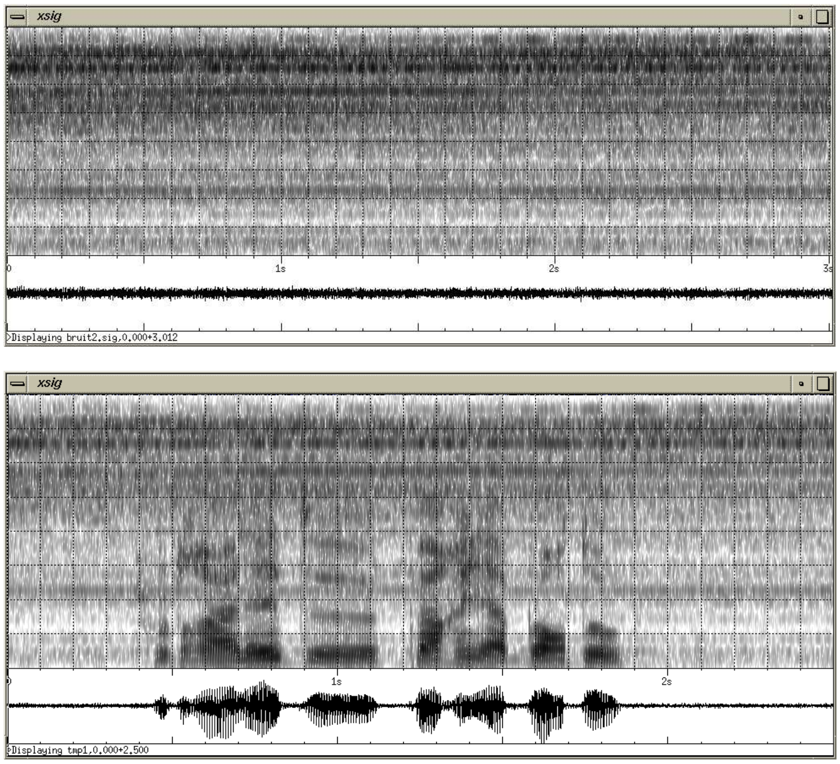


**Figure 3.8 : Bruit de bateau (salle de commande)**

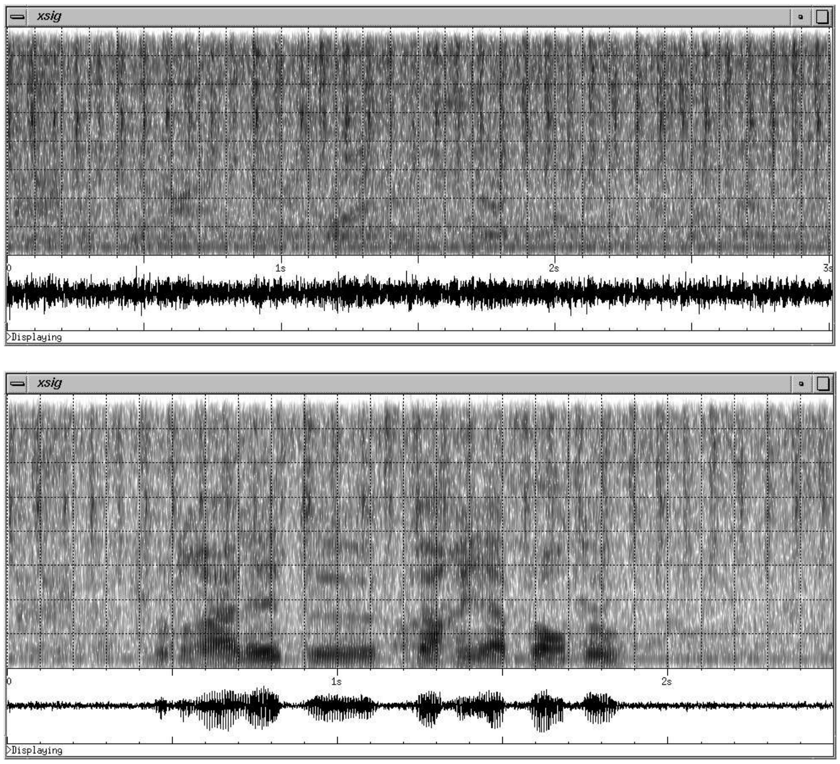



**Figure 3.9 : Bruit de bateau (salle des machines)**

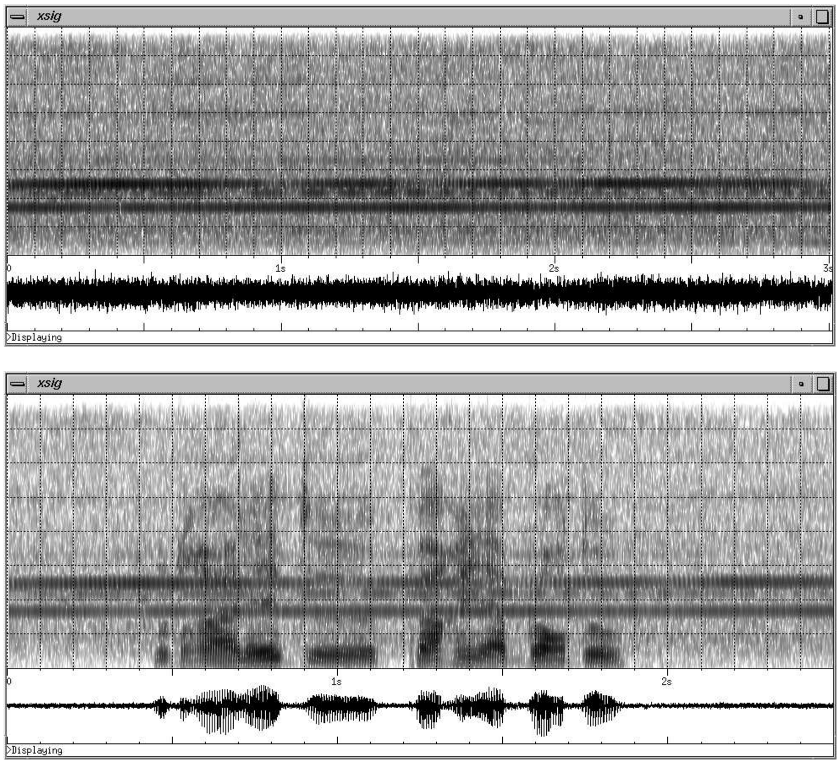


**Figure 3.10 : Bruit de foule (brouhaha)**

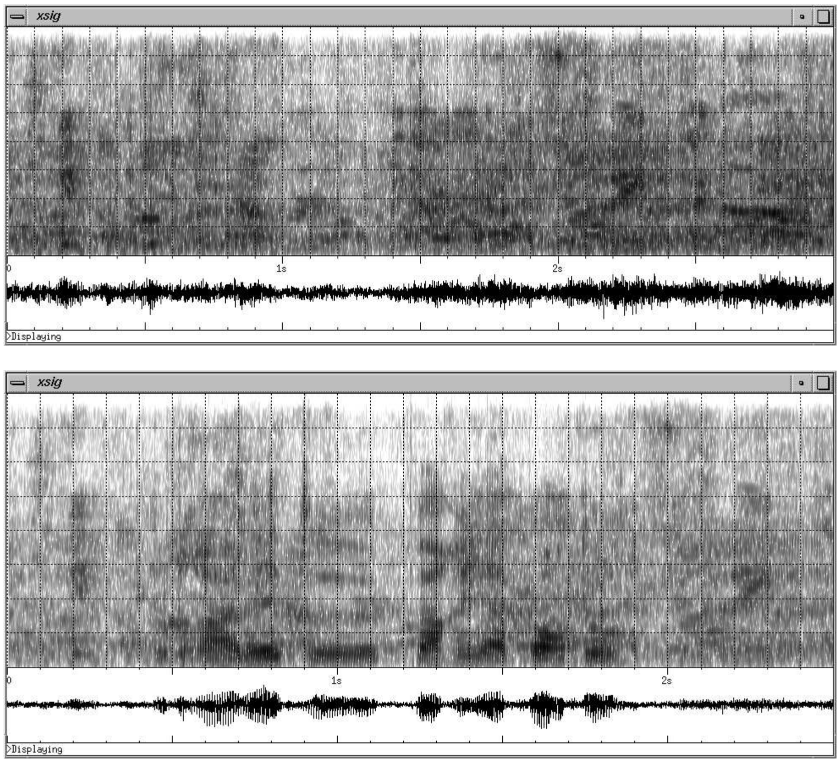



**Figure 3.11 : Bruit d'avion à réaction (F-16, 300-600 pieds)**

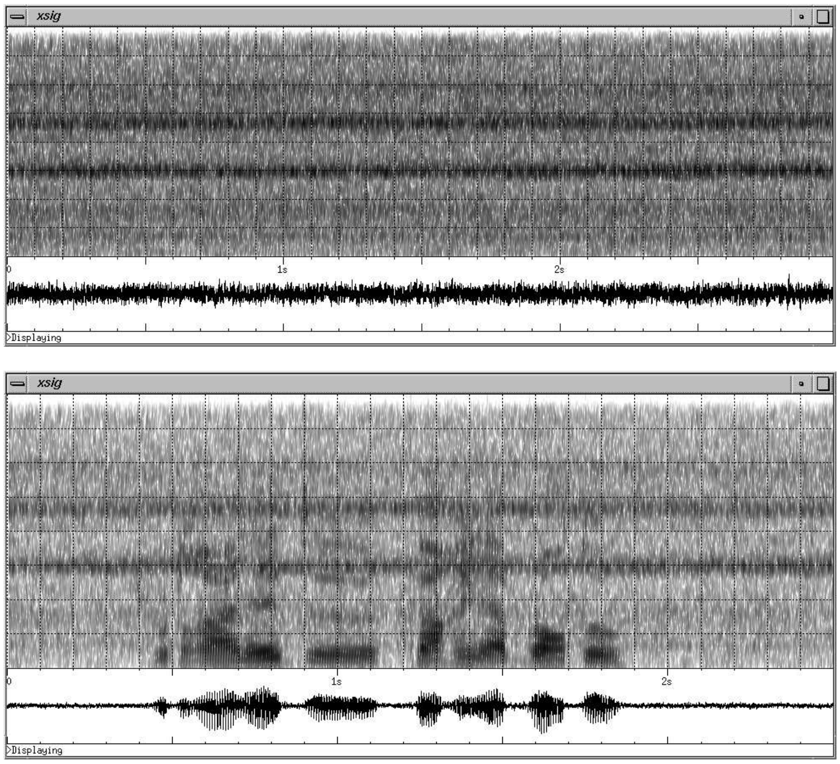


**Figure 3.12 : Atelier de fabrication Automobile**

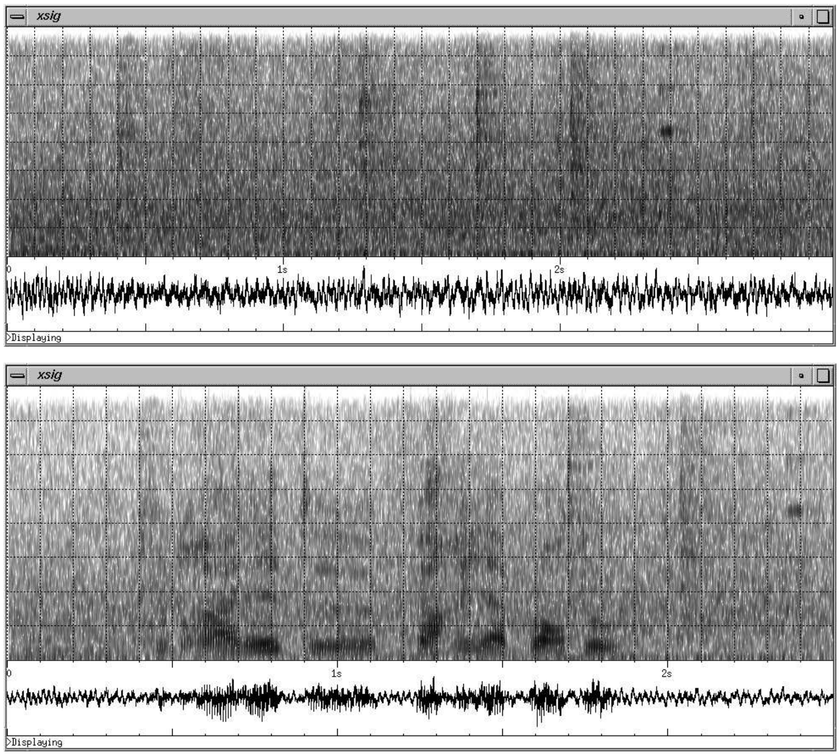
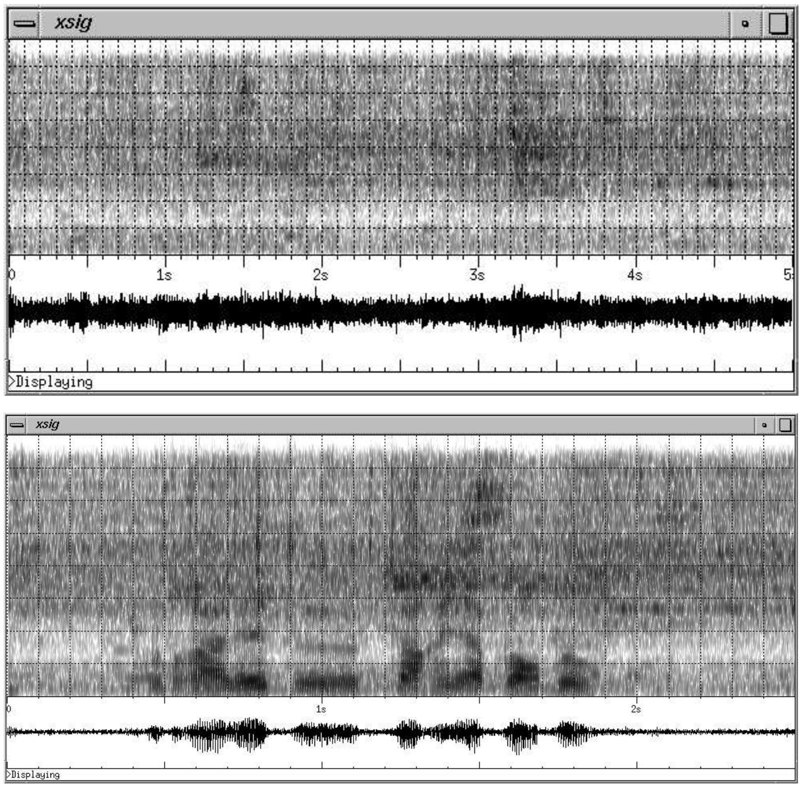



**Figure 3.13 : Bruit enregistré à la gare Saint-Lazare**



Dans la Table 3.7, mise sous forme de courbes dans la Figure 3.14, nous donnons les taux d'erreur sur les mots produits par le système de reconnaissance dans le cas de données de test affectées par les différents bruits présentés dans ce chapitre. Plusieurs rapport signal sur bruit sont testés afin d'étudier le comportement du système de reconnaissance en fonction des types de bruits et des RSB.

### 3.2.2 Interprétation des résultats

Quel que soit le type de bruit affectant les données de test, le rapport signal sur bruit constitue un important facteur caractérisant les performances du système de reconnaissance. Nous constatons d'ailleurs, que pour des rapports signal sur bruit assez élevés (20 dB), les taux d'erreur sur les mots pour tous les bruits varient légèrement autour de 7.5%. Cependant, la nature du bruit affectant les données de test joue un rôle non négligeable et parfois dominant, surtout pour des RSBs assez faibles. Par exemple, avec un RSB de 5dB, le taux d'erreur sur les mots, quand les données de test sont affectées par le bruit blanc, est de 44.1% ; en revanche le taux d'erreur quand les données de test sont affectées par le bruit de "Bateau (salle de commande)" (avec un RSB=5dB) est de 78.9% ; soit une augmentation relative de 80%. Nous remarquons que cet écart ne diminue pas forcément avec l'augmentation du rapport signal sur bruit : avec un RSB de 8dB, le taux d'erreur sur les mots, quand les données de test sont affectées par le bruit blanc, est de 24%, et le taux d'erreur quand les données de test sont affectées par le bruit de "Bateau (salle de commande)" est de 58.9%. Soit une augmentation relative de 145%.

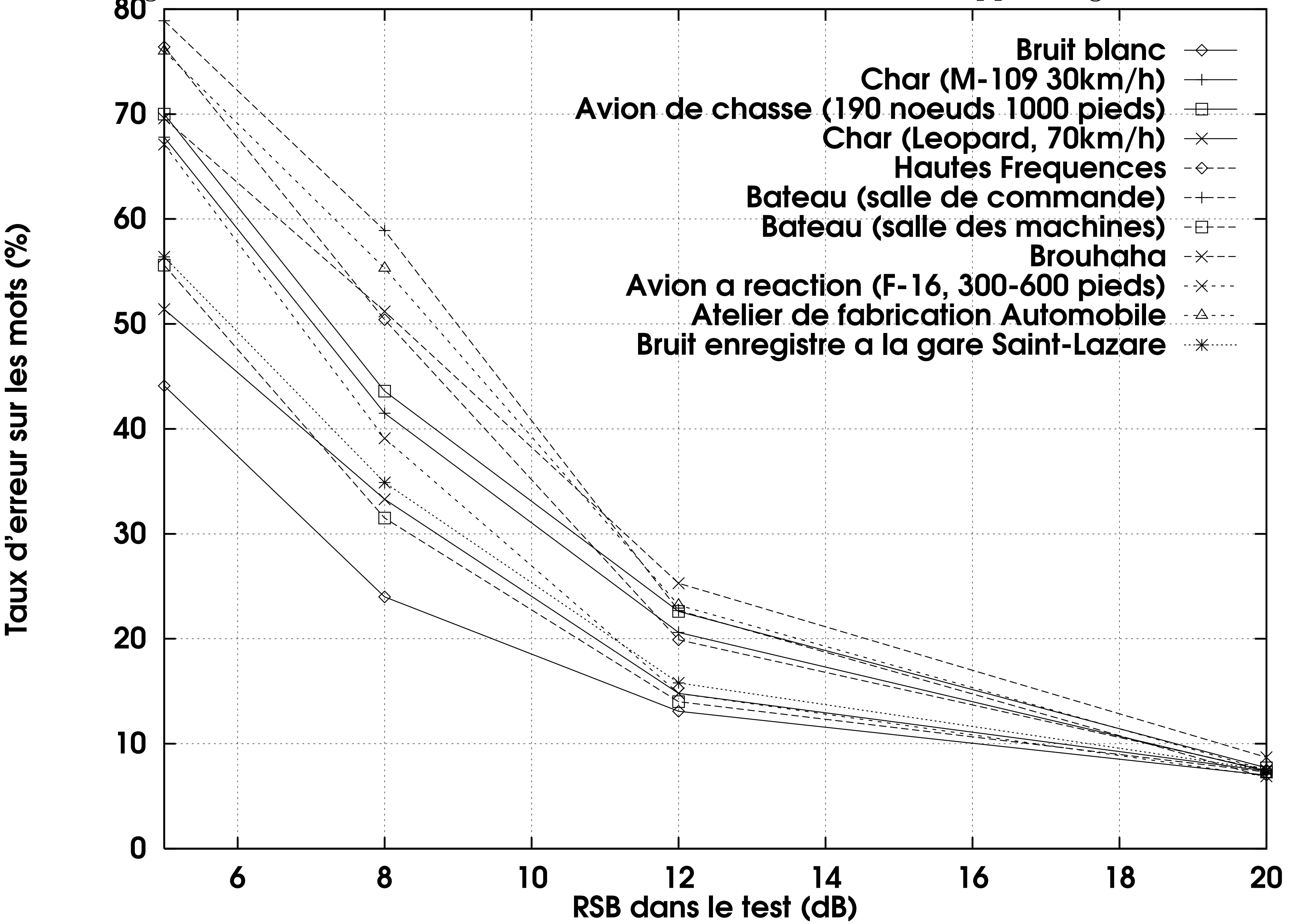
Comme le montre la Figure 3.14, le système de reconnaissance est plus robuste au bruit blanc en comparaison avec les autres bruits, ceci est valable pour tous les rapports signal sur bruit. Nous remarquons aussi que la stationnarité du bruit qui affecte les données de test n'est pas toujours un avantage pour le système de reconnaissance. En effet, certains bruits stationnaires peuvent être plus gênants que d'autres bruits qu'on peut qualifier de non stationnaires. Par exemple, avec un RSB de 5dB, quand les données de test sont affectées par le bruit Hautes Fréquences (Figure 3.7), le taux d'erreur sur les mots est de 76.4%, et quand les données de test sont affectées par le bruit de brouhaha, le taux d'erreur sur les mots est de 69.6%. Ce constat n'est pas valable pour tous les rapports signal sur bruit. Nous verrons dans les chapitres concernant la compensation du bruit, qu'il est beaucoup plus commode de compenser un bruit stationnaire que de compenser un bruit non stationnaire. Une première vue sur la Figure 3.14, montre que la robustesse du système de reconnaissance de la parole à un bruit par rapport à un autre n'est pas constante en fonction du rapport signal sur bruit. Par exemple, avec un RSB de 5dB, lorsque les données de test sont affectées par le bruit Hautes Fréquences (Figure 3.7), le taux d'erreur sur les mots est de 76.4%, et le taux d'erreur sur les mots lorsque les données de test sont affectées par le bruit de brouhaha est de 69.6% ; en revanche avec un RSB de 12dB, lorsque les données de test sont affectées par le bruit Hautes Fréquences, le taux d'erreur sur les mots est de 19.9%, et le taux d'erreur sur les mots lorsque les données de test sont affectées par le bruit de brouhaha est de 25.3%.

La Figure 3.14 montre que pour des faibles RSBs (5dB , 8dB), le bruit le plus gênant (parmi tous les bruits étudiés) est celui de “Bateau (salle de commande)” ; et pour des RSBs relativement élevés (12dB, 20dB), le bruit qui perturbe le plus le système de reconnaissance est celui du brouhaha.

| Type de bruit | RSB dans les données de test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Bruit blanc | 5.9% | 7.0% | 13.1% | 24.0% | 44.1% |
| Char (M-109 30km/h) | 5.9% | 7.4% | 20.6% | 41.5% | 67.8% |
| Avion de chasse (190 noeuds 1000 pieds) | 5.9% | 7.7% | 22.6% | 43.6% | 70.0% |
| Char (Leopard, 70km/h) | 5.9% | 7.4% | 14.8% | 33.3% | 51.4% |
| Bruit Hautes Fréquences | 5.9% | 7.5% | 19.9% | 50.4% | 76.4% |
| Bateau (salle de commande) | 5.9% | 6.8% | 22.7% | 58.9% | 78.9% |
| Bateau (salle des machines) | 5.9% | 7.3% | 14.0% | 31.5% | 55.6% |
| Brouhaha | 5.9% | 8.7% | 25.3% | 51.2% | 69.6% |
| Avion à réaction (F-16, 300-600 pieds) | 5.9% | 6.9% | 14.8% | 39.1% | 67.1% |
| Atelier de fabrication Automobile | 5.9% | 7.5% | 23.2% | 55.3% | 76.0% |
| Bruit enregistré à la gare Saint-Lazare | 5.9% | 7.5% | 15.8% | 34.9% | 56.4% |

**Table 3.7** *Dans ces expériences, les données d'apprentissage sont propres et elles de test sont bruitées avec différents bruits et avec différents RSB. Cette table présente les taux d'erreur sur les mots. Tous les bruits utilisés dans ces expériences sont issus de la base de données NOISEX-92 [81], sauf celui de la cinquième ligne (Hautes Fréquences) et celui de la dernière ligne qui est enregistré à la gare Saint-Lazare à Paris.*

**Figure 3.14 :** Les taux d'erreur sur les mots en fonction du rapport signal sur bruit.



## 3.3 Conclusion

Dans ce chapitre nous avons réalisé deux études : la première consiste à examiner l'effet du bruit ambiant sur le système de reconnaissance en fonction de la technique de paramétrisation du signal utilisée ; la deuxième consiste à examiner l'effet de différents types de bruits sur le système de reconnaissance en fixant la technique de paramétrisation du signal.

La première étude nous a permis de constater qu'en général une technique de paramétrisation de type MFCC ou PLPCC est meilleure que la technique de paramétrisation LPCC. La principale cause est que la LPCC ne tient pas compte du fait que l'importance de l'information résidante dans le signal est une fonction logarithmique de la fréquence linéaire (Hertz). Lorsque les données d'apprentissage sont propres, les résultats obtenus par la MFCC et la PLPCC sont similaires. En revanche, lorsque les données d'apprentissage sont bruitées et les données de test sont propres, la PLPCC donne de meilleurs résultats. Nous avons constaté, en outre, qu'il n'y a pas de valeur optimale de $\gamma$ dans le cadre de la PLPCC. La valeur optimale de ce paramètre dépend du corpus d'apprentissage, du corpus de test et de la nature et du niveau du bruit ambiant.

La deuxième étude nous a permis de constater que quel que soit le type de bruit affectant les données, le RSB constitue un important facteur caractérisant les performances du système. Nous avons aussi constaté que certains bruits stationnaires peuvent être plus gênants (avec le même RSB) que certains bruits non stationnaires. Il est aussi à noter que les bruits possédant une forte énergie dans les hautes fréquences sont aussi gênants que les autres bruits. Cependant, comme nous allons le voir dans le chapitre 5, ce type de bruit est plus facile à compenser que celui possédant une forte énergie dans les basses fréquences. Cela vient du fait que le bruit avec une forte énergie dans les hautes fréquences affecte le spectre de chaque trame mais cause peu de perte d'information par rapport aux bruits avec une forte énergie dans les basses fréquences ; il cause donc une dégradation qui peut être considérable si on ne le compense pas, mais il est facile à compenser.

Enfin, quel que soit le bruit affectant les données de test, le système de reconnaissance de la parole se trouve sérieusement perturbé, surtout pour des rapports signal sur bruit faibles. L'étude et la mise en oeuvre de techniques permettant de compenser les effets du bruit sont donc indispensables pour garantir une bonne robustesse d'un système de reconnaissance lors d'une application dans des conditions acoustiques d'utilisation difficiles.

Dans le chapitre 4, nous présentons un état de l'art des efforts portés pour compenser les effets du bruit, puis dans les chapitres 5, 6 et 7 notre contribution dans ce domaine.

# Chapitre 4

# Robustesse aux variabilités acoustiques : état de l'art

## 4.1 Introduction

La reconnaissance de la parole dans des conditions favorables a atteint des performances acceptables. Par exemple, dans le cadre des évaluations ARPA, avec le système de reconnaissance de la parole continue du LIMSI [26], un taux d'erreur sur les mots proche de 10% est obtenu sur le corpus de test ARPA NAB CSR[1] de novembre 1995 dont le rapport signal sur bruit est de 30dB environ. En revanche, en utilisant le même système sur le même corpus de test mais avec un rapport signal sur bruit variant entre 7 et 18dB, le taux d'erreur sur les mots est supérieur à 50% [26]. Ceci montre bien à quel point les performances sont dépendantes des conditions acoustiques d'utilisation.

L'objectif des techniques utilisées pour rendre les systèmes de reconnaissance plus robustes est de compenser les différences entre les conditions d'apprentissage et les conditions d'utilisation du système. Ces différences sont en général réduites en utilisant une des trois approches suivantes :

- 1. La première consiste à considérer que le système de reconnaissance peut être utilisé de la même manière pour reconnaître la parole propre et la parole bruitée. Dans ce cas la solution consiste à extraire du signal de parole des paramètres acoustiques moins sensibles au bruit tout en gardant l'information pertinente pour la reconnaissance[2]. Ce type d'approche ne nécessite pas d'hypothèses *a priori* sur la nature du bruit.

- 2. Dans la deuxième approche, on suppose que l'apprentissage s'est déroulé dans une ambiance calme et on procède dans ce cas au débruitage des signaux

[1] Il s'agit d'une tâche de reconnaissance de la parole avec un vocabulaire illimité sur le corpus *"North American Business News"* [65].

[2] Les paramètres cepstraux classiques sont très sensibles au bruit, on montre d'ailleurs que dans le cas d'un bruit blanc, la norme des vecteurs cepstraux ($c_0$ le premier coefficient cepstral exclu) décroit avec le rapport signal sur bruit [52].

de test[3]. Ceci permet de se ramener à des conditions acoustiques similaires entre l'apprentissage et le test du système de reconnaissance : conditions d'apprentissage. Cette deuxième approche, contrairement à la première, nécessite généralement une connaissance *a priori* sur la nature du bruit.

- 3. La troisième approche, consiste à ne pas modifier les données de test, mais à adapter les modèles acoustiques, appris sur des signaux propres, pour reconnaître la parole bruitée. Ceci permet, comme la seconde approche, de se ramener à des conditions acoustiques similaires entre l'apprentissage et l'utilisation du système : conditions de test.

Ces trois approches peuvent être utilisées conjointement pour appréhender le problème du bruit dans le cadre de la reconnaissance de la parole. Par exemple, J. A. Nolazco et S. J. Young [63] utilisent la soustraction spectrale [9] ($2^{ème}$ approche) dans le cadre de la combinaison parallèle de modèles [23] ($3^{ème}$ approche).

Dans ce qui suit, nous allons décrire les techniques qui visent à augmenter la robustesse des systèmes de reconnaissance suivant les trois approches que nous venons de présenter.

## 4.2 Paramètres acoustiques robustes au bruit

Les paramètres spectraux utilisés actuellement (MFCC, LPCC) dans les systèmes de reconnaissance donnent des résultats satisfaisants, à condition que l'apprentissage et le test du système se déroulent dans les mêmes conditions acoustiques et dans un milieu non bruité. Ces paramètres sont en général assez sensibles au bruit, ce qui implique une chute des performances lorsque les conditions de bruits sont différentes entre l'apprentissage et le test du système. Ce problème subsiste à un degré moindre si l'apprentissage et le test se déroulent dans les mêmes conditions bruitées.

Les techniques de paramétrisations acoustiques robustes au bruit ont pour objectif d'extraire des paramètres acoustiques qui donneraient des performances aussi bonnes que les paramètres usuels dans une ambiance calme et qui, en même temps, soient plus robustes au bruit. Un avantage de ce type de technique est qu'en général aucune contrainte sur la nature du bruit n'est imposée. Ceci est profitable dans la mesure où il est impossible de traiter tous les types de bruits au cas par cas.

### 4.2.1 L'analyse PLPCC et l'analyse par prédiction linéaire généralisée

L'analyse PLPCC (*Perceptually-based Linear Prediction Cepstrum Coefficients*) [34] est une technique de paramétrisation qui tient compte de trois aspects de la perception : l'intégration des bandes critiques, la préaccentuation par courbe d'isosonie et la loi de Stevens (voir chapitre 2). Une différence principale entre l'analyse PLPCC et l'analyse LPCC standard (*Linear Predictive Cepstrum Coefficients*) [54] est que

[3] Débruitage du signal temporel lui même ou des paramètres acoustiques correspondants

l'analyse PLPCC prend en compte le fait que l'aspect discriminatoire du système auditif humain diminue d'une manière logarithmique avec l'augmentation de la fréquence, contrairement à l'analyse LPCC qui fournit une représentation spectrale approximant la densité spectrale du signal analysé avec la même précision pour toutes les fréquences. L'effet de cette différence augmente avec l'augmentation de la largeur de bande du signal de parole analysé.

Une autre différence entre l'analyse LPCC et l'analyse PLPCC est la compression spectrale en racine qui consiste à appliquer la fonction $(\cdot)^\gamma$ où $\gamma = 0.33$, à la densité spectrale avant de retourner au domaine de l'autocorrélation. L'objectif de cette compression spectrale en racine est de lisser la densité spectrale. S.M. Kay [41] montre que plus la densité spectrale d'un processus ARMA est lisse, plus le processus AR l'approchant est court. Ainsi, la valeur de $\gamma$ doit être choisie en faisant un compromis entre la perte d'information au niveau des pics spectraux (formants) et la préservation de l'information au niveau des vallées spectrales. La préservation de l'information au niveaux des vallées spectrales est indésirable car ces parties du spectre sont les plus sensibles aux distorsions causées par le bruit ambiant.

L'analyse par prédiction linéaire généralisée constitue une généralisation de l'analyse LPCC. En effet, elle consiste à remplacer la fonction $(\cdot)^2$ après la transformation de Fourier par la fonction $(\cdot)^\gamma$ où $\gamma \in ]0, 2]$. Le signal issu de la transformation de Fourier inverse qui en découle est la fonction d'autocorrélation du signal compressé spectralement en racine. De ce fait, elle est appelée fonction d'autocorrélation généralisée. L'analyse cepstrale en racine consiste à remplacer la fonction logarithmique dans l'analyse LFCC standard (*Linear Frequency Cepstral Coefficients*) par la fonction $(\cdot)^\gamma$. Ceci permet d'avoir une vue unifiée [3] des deux analyses LPCC et LFCC. En effet, quand $\gamma$ tend vers zéro la fonction $(\cdot)^\gamma$ tend vers la fonction Logarithme et donc on s'approche d'une analyse LFCC. Quand $\gamma$ tend vers 2 on s'approche d'une analyse LPCC. La valeur optimale choisie pour $\gamma$ est celle donnée par la loi de Stevens à savoir $0.33 \times 2$ [76].

K. Tokuda [78] propose plutôt la fonction $\left((\cdot)^\gamma - 1\right)/\gamma$ qui est plus adaptée que $(\cdot)^\gamma$ dans le sens où elle approxime mieux la fonction logarithmique quand $\gamma$ tend vers zéro. Il faut remarquer que dans le cadre de l'analyse cepstrale en racine, l'absence d'un opérateur de type logarithmique rend la normalisation par rapport au bruit convolutif, impossible dans le domaine cepstral. Par contre au sein de l'analyse par prédiction linéaire généralisée, il est possible de faire cette normalisation grâce au logarithme introduit au moment du passage des coefficients de prédiction aux coefficients cepstraux.

La seule différence qui reste entre l'analyse cepstrale en racine et l'analyse PLPCC est l'intégration en bande de filtres suivant l'échelle Bark. Comme nous l'avons constaté dans les expériences présentées dans le chapitre 3, cette différence a toute son importance, surtout si l'analyse est effectuée sur une large bande de fréquence (de l'ordre de 8kHz).

Dans des expériences réalisées dans le chapitre 3, nous montrons que lorsque les données d'apprentissage sont propres et les données de test sont propres ou bruitées, les résultats obtenus par l'analyse MFCC et l'analyse PLPCC sont pratiquement identiques. En revanche, avec des données d'apprentissage bruitées et des données de test propres, l'analyse PLPCC donne de meilleurs résultats que les analyses MFCC et LPCC. Cependant, quelle que soit la technique de paramétrisation utilisée, les résultats en terme de reconnaissance deviennent inacceptables pour des RSB faibles.

### 4.2.2 La SMC (*Short-time Modified Coherence*)

D. Mansour et B.H. Juang [53] montrent que l'autocorrélation d'un signal obtenu par le passage d'un bruit blanc ou d'un train d'impulsions dans un filtre tout-pôle $\frac{1}{A(z)}$, est elle même un signal auto-régressif et que le filtre tout-pôle correspondant est $\frac{1}{A(z)^2}$. De ce fait il est possible d'utiliser le signal d'autocorrélation afin d'estimer $A(z)$. L'intérêt de passer par l'autocorrélation pour estimer $A(z)$ est que dans le domaine de l'autocorrélation il est plus facile (pour certains bruits) d'isoler une partie de la contribution du bruit. En effet, pour des bruits à densités spectrales relativement plates (bruit blanc), seuls les quelques premiers coefficients de l'autocorrélation sont affectés par le bruit, il suffit donc de les éliminer de l'analyse pour aboutir aux coefficients de prédiction du signal non bruité.

Dans la pratique l'analyse LPC de la fonction d'autocorrélation n'aboutit pas forcément à $p$ pôles doubles ($p$ est l'ordre de la LPC du signal d'origine). En outre, la réponse impulsionnelle du filtre (dans le domaine de l'autocorrélation) est deux fois plus longue que celle du signal d'origine, ce qui génère une interaction plus importante entre les formants et la fréquence fondamentale. Afin de résoudre ces problèmes on applique un opérateur de type "racine carré" à la densité spectrale du signal d'autocorrélation avant de lui appliquer l'analyse LPC [53]. Cette analyse est connue sous le nom *"Short-time Modified Coherence" (SMC)*.

Des expériences faites par les auteurs [53] utilisant un très petit corpus de test (39 mots : lettres+chiffres) montrent que la SMC donne des résultats comparables à l'analyse LPC standard dans le cas de signaux propres et une amélioration significative dans le cas de signaux bruités avec un bruit blanc. Cependant, il serait plus instructif de comparer les performances de cette technique avec d'autres techniques plus performantes que l'analyse LPC (comme l'analyse MFCC ou l'analyse PLPCC)[4] et avec des corpus et vocabulaires plus importants.

### 4.2.3 Soustraction du *cepstre* moyen

La technique de la soustraction du *cepstre* moyen à long terme est couramment utilisée pour éliminer les distorsions liées au changement du canal d'enregistrement [5, 20]. Après soustraction du cepstre moyen à long terme on aboutit à des paramètres cepstraux moins dépendants du matériel d'enregistrement utilisé. Il faut noter que cette technique suppose qu'on dispose d'une bonne estimation de la moyenne du *cepstre* et donc d'une séquence assez longue de vecteurs cepstraux. Cette contrainte constitue une entrave à la réalisation des systèmes de reconnaissance de la parole en temps réel, puisqu'il est nécessaire d'avoir suffisamment de données pour estimer convenablement la moyenne *cepstrale*.

[4] Une idée qui peut être intéressante est d'appliquer le même principe dans le cadre de l'analyse PLP. C'est-à-dire, qu'on réalise une analyse PLP sur le signal d'autocorrélation.

Il est clair que la moyenne estimée dépend de la proportion du silence (bruit) dans la phrase. Pour résoudre ce problème, on estime la moyenne du silence (bruit) qui est soustraite aux trames correspondant au silence (bruit) et la moyenne de la parole qui est soustraite aux trames correspondant à la parole. On utilise pour cela un mélange de deux gaussiennes : une pour les trames de parole et une autre pour les trames du silence (bruit). Cette classification se fait en utilisant l'algorithme EM avec l'énergie comme paramètre de classification [12]. Ainsi la soustraction du *cepstre* moyen se fait comme suit :

$$\hat{y}^c(t) = y^c(t) - [\gamma \bar{y}_p^c + (1-\gamma)\bar{y}_b^c] \tag{4.1}$$

où $\gamma$ désigne la probabilité que $y^c(t)$ soit de la parole. $\bar{y}_p^c$ et $\bar{y}_b^c$ désignent respectivement la moyenne des vecteurs de parole et la moyenne des vecteurs de silence [38]. Cependant, du point de vue des taux de reconnaissance, cette manière de soustraire le cepstre moyen donne des résultats similaires à ceux donnés par la soustraction du cepstre moyen standard.

### 4.2.4 RASTA-PLP et *cepstre* dynamique

Dans de nombreux cas, les bruits additifs et les bruits convolutifs varient lentement par rapport au signal de parole. Le filtrage de ces variations lentes apporte souvent une amélioration significative des performances [33]. Ce filtrage peut être implanté de différentes manières et dans différents domaines de paramétrisation du signal de parole. Par exemple, dans le cadre d'une analyse PLPCC, on applique un filtrage passe-haut dans le domaine cepstral [33], qui est réalisé par la soustraction de la moyenne mobile. Le même principe peut être réalisé dans le domaine log-spectral (*RASTA-PLP : RelAtive SpecTrAl - Perceptually-based Linear Prediction*) [35]. Les résultats (du point de vue de la reconnaissance) de ce type de filtrage sont similaires à ceux de la soustraction du *cepstre* moyen. Par contre ce principe a le mérite de fonctionner en temps réel, dans la mesure où on n'est pas contraint d'attendre la fin de la phrase pour réaliser cette opération (contrairement à la soustraction du cepstre moyen).

Il est important de noter que ce type de filtrage permet aussi de normaliser par rapport au locuteur dans le cadre d'un système de reconnaissance de la parole multi-locuteur et de réduire les effets Lombard dans une ambiance bruitée [33].

L'utilisation des coefficients cepstraux dynamiques ($\Delta$*cepstre*, $\Delta\Delta$*cepstre*) permet d'introduire des coefficients relativement indépendants des variations lentes additives dans le domaine *cepstral*. Ces coefficients supplémentaires ont montré des gains en performances dans les deux cas, bruité et non bruité [32, 33, 4].

### 4.2.5 L'analyse linéaire discriminante

L'analyse discriminante linéaire (*LDA Linear Discriminant Analysis*) est une technique statistique utilisée dans le cadre de la classification des formes [44]. Supposons que l'espace acoustique soit réparti en un ensemble de classes, chaque classe étant représentée par une gaussienne de matrice de covariance $\Sigma_1$. Supposons que toutes les classes aient la même matrice de covariance. La LDA consiste à appliquer une transformation (rotation + dilatation) à tout l'espace acoustique de manière à rendre la variance intra-classe égale à l'identité. En supposant que les nouveaux centres des classes soient distribués selon une loi gaussienne de matrice de covariance inter-classes $\Sigma_2$, une deuxième rotation est appliquée à l'espace acoustique de manière à dégager les axes principaux. Ces axes sont les vecteurs propres de la matrice $\Sigma_2$. Ainsi un sous-ensemble de ces axes correspondant aux directions des grandes variances (les grandes valeurs propres) est utilisé pour former les nouveaux vecteurs de paramétrisation du signal. Il faut noter que cette technique n'effectue aucun traitement spécifique au bruit ; son but est d'augmenter la discrimination entre les formes à classer, quel que soit le type des données bruitées ou non bruitées. Nous l'avons évoquée ici car l'augmentation de la discrimination est spécialement requise dans le cas de données bruitées. Cette technique est plus connue sous le nom IMELDA (*Integrated Mel-scale representation with LDA*) car elle a été appliquée en premier lieu dans le cadre d'une paramétrisation MFCC.

## 4.3 Débruitage

L'amélioration de la qualité du signal de parole constitue une étape de pré-traitement du signal, dont le but est de retrouver les vecteurs de paramètres du signal propre correspondant au signal bruité observé. Ce pré-traitement peut s'effectuer soit dans l'espace des paramètres (spectral ou *cepstral*), soit directement sur le signal observé.

### 4.3.1 Soustraction spectrale

La soustraction spectrale introduite par Boll [9] dans les années 70 traite les bruits additifs, stationnaires et non corrélés avec le signal de parole. Elle consiste à effectuer une décomposition spectrale de chaque trame du signal bruité. Puis chaque canal du spectre est atténué selon que le niveau mesuré localement dans ce canal dépasse plus ou mois l'estimation du bruit. La fonction qui permet de déterminer l'atténuation, constitue ce qu'on appelle **la règle de suppression**. Plusieurs règles de suppression ont été suggérées [10, 12, 49], dans ce qui suit nous décrivons la règle de suppression la plus utilisée. Soit $\hat{N}$ une estimation du spectre du bruit, $Y$ le spectre du signal bruité, l'estimation du spectre du signal propre $\hat{X}$ est donnée par [8] :

$$\begin{array}{ll} \hat{X} = Y - \alpha\hat{N} \quad si & Y - \alpha\hat{N} > \beta Y \\ \hat{X} = \beta Y \quad sinon & \end{array}$$

où $\alpha$ est un facteur de surestimation du bruit. L'intérêt de cette surestimation est de réduire le niveau du bruit résiduel, ce qui implique une augmentation du rapport signal sur bruit[5]. $\beta$ est le facteur de seuil, son intérêt est de ne pas avoir des valeurs négatives causées par la variance du bruit. Il faut noter que seule l'amplitude spectrale du signal est modifiée, le signal estimé conserve la phase du signal bruité. D.L. Wang montre l'insignifiance de la phase dans l'amélioration de la qualité du signal de parole. Nous avons vu dans le chapitre précédent que dans le cadre de la reconnaissance de la parole, seule l'amplitude spectrale est considérée.

En utilisant cette technique, on aboutit à un signal avec un meilleur rapport signal sur bruit. Cependant, l'utilisation des facteurs $\alpha$ et $\beta$ introduit des transformations non linéaires et par suite des distorsions sur le signal de parole. Il faut donc aboutir à un compromis entre l'augmentation du rapport signal sur bruit et les distorsions introduites dans le signal. L'inconvénient principal de cette technique est le fait de supposer que le bruit est parfaitement stationnaire et donc de ne pas prendre en compte l'effet du bruit sur les variances des modèles. C'est pour ces raisons que la réduction du taux d'erreur est marginale, il peut même y avoir une augmentation du taux d'erreur pour des rapports signal sur bruit supérieurs à 15dB.

Plusieurs études ont été réalisées afin de combler les défauts liés à cette technique. R.J. McAulay et M.L Malpass [51], ont formulé le problème de la soustraction spectrale en utilisant l'approche du maximum de vraisemblance. Ce travail, avait pour objectif d'éliminer ou de réduire le bruit musical qui subsiste après l'application d'une soustraction spectrale classique. Le mérite principal de l'approche de R.J. McAulay et M.L Malpass est d'être parmi les premières à prendre en compte l'information linguistique existante dans le signal de parole à débruiter.

### 4.3.2 L'inhibition latérale

La robustesse au bruit de l'oreille humaine est en partie attribuée au processus de l'inhibition latérale dans le système auditif qui se traduit par la modification de la réponse d'un neurone à une stimulation. Les signaux inhibiteurs sont issus d'autres neurones voisins. Un modèle d'inhibition latérale spectrale provoque une amplification des caractéristiques spectrales les plus importantes. L'inhibition latérale peut se comprendre de la manière suivante : considérons un sujet qui écoute un son constitué de deux tons simultanés. Le premier ton a une intensité et une fréquence ($F_s$) fixes, le deuxième a une intensité $I_c$ fixe et une fréquence variable ($F$). L'intensité subjective ($I$) du second ton perçue par le système auditif humain varie en fonction de la fréquence. La diminution de l'intensité perçue dans certaines zones de fréquences avoisinant la fréquence $F_s$ (voir figure 4.1) constitue l'inhibition latérale [11].

[5] Il faut être prudent vis-à-vis de la notion d'augmentation du RSB. Dans le cas de la soustraction spectrale, seul le RSB calculé sur le signal résultant augmente (signal très différent du signal d'origine et contenant d'autres composantes que la parole). Ceci n'implique pas forcément une augmentation du vrai rapport signal sur bruit. En poussant ce raisonnement à l'extrême, il est possible d'augmenter le RSB tout simplement en faisant une classification des trames en deux classes bruit/parole (en utilisant l'algorithme EM sur les énergies des trames), puis en affectant à toutes les trames de la classe bruit une valeur minimale et à toutes les trames de la classe parole une valeur maximale. Le RSB dans ce cas sera très élevé, mais le signal résultant de cette opération sera quelconque. Ici le terme RSB est utilisé avec un abus de langage.

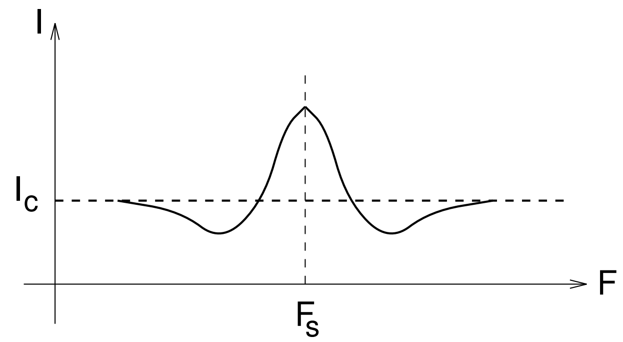



**Figure 4.1 :** Illustration du phénomène de l'inhibition latérale.

Afin de modéliser ce phénomène, notons $Z(\omega,\Omega)$ la Fonction de l'Inhibition Latérale Spectrale (FILS) associée à la courbe de la figure 4.1, où $\Omega$ est la fréquence pour laquelle on veut estimer le spectre de puissance et $\omega$ est une variable désignant toutes les fréquences. Supposons que cette fonction satisfasse la condition suivante :

$$\int_{-\infty}^{+\infty} Z(\omega,\Omega)\, d\omega = 0 \quad \forall\Omega$$

On suppose que le signal de parole est affecté par un bruit blanc, le signal observé peut être exprimé sous la forme suivante :

$$x(k) = s(k) + n(k)$$

Dans le cadre d'une analyse à court terme, on peut supposer que les signaux de parole et du bruit sont stationnaires[6]. Si on suppose que le bruit est non corrélé avec le signal de parole alors la densité spectrale du signal observé correspondant à la $i^{ème}$ trame peut s'écrire comme suit :

$$\Gamma_x^i(\omega) = \Gamma_s^i(\omega) + \Gamma_n^i(\omega) = \Gamma_s^i(\omega) + \mu^i$$

où $\mu^i$ est une constante (indépendante de la fréquence). En convoluant cette densité spectrale avec $Z(\omega,\Omega)$, on obtient une nouvelle densité spectrale :

$$\hat{\Gamma}_x^i(\Omega) = \int_{-\infty}^{+\infty} \Gamma_x^i(\omega) Z(\omega,\Omega)\, d\omega = \hat{\Gamma}_s^i(\Omega) + \mu^i \int_{-\infty}^{+\infty} Z(\omega,\Omega)\, d\omega = \hat{\Gamma}_s^i(\Omega)$$

Les équations ci-dessus montrent qu'un tel modèle, non seulement provoque un renforcement des pics spectraux mais qu'en plus il élimine le bruit blanc additif. En terme de RSB[7], cette technique donne de meilleurs résultats que la soustraction spectrale (voir paragraphe 4.3.1) pour des signaux très bruités et des résultats similaires pour des signaux moyennement bruités [11]. Cependant son grand avantage est qu'elle ne nécessite pas d'estimation *a priori* du bruit et que les contraintes imposées sur le bruit traité sont moins fortes.

---

[6] Le bruit affectant le signal de parole est supposé être blanc et stationnaire durant une trame (de l'ordre de 20ms), mais dans l'ensemble ce bruit peut ne pas être stationnaire, c'est-à-dire que son niveau peut varier d'une trame à l'autre.

[7] voir note de bas de page dans le paragraphe 4.3.1

Il est important de noter que cette technique comme la soustraction spectrale utilise un facteur de seuil pour éliminer les valeurs négatives. En outre le spectre de puissance à court terme peut posséder des pics causés par les extrémités de la fenêtre d'analyse. L'inhibition latérale est très sensible aux pics spectraux, même ceux de faible amplitude, ce qui donne naissance à un bruit musical (comme celui de la soustraction spectrale), notamment dans le cas de signaux très bruités. Cette approche, intéressante de par son analogie avec l'oreille humaine, n'est cependant pas utilisée dans les systèmes de reconnaissance actuels !

### 4.3.3 La SDCN (*SNR-Dependent Cepstral Normalization*)

La plupart des techniques dont le principe est la détermination d'une fonction paramétrique mettant en correspondance les deux espaces acoustiques propre et bruité, opèrent dans le domaine spectral. Les fonctions utilisées sont en général des transformations linéaires estimées en utilisant le principe des moindres carrés ou du maximum de vraisemblance. Les plus simples parmi ces techniques nécessitent l'utilisation des signaux stéréophoniques. La SDCN [1] est une technique qui opère dans le domaine cepstral, et la transformation utilisée entre les deux espaces acoustiques propre et bruité est un simple *offset* dépendant exclusivement du rapport signal sur bruit mesuré localement dans la trame en question.

$$\hat{s}_t = y_t + w(RSB)$$

où $\hat{s}$ une estimation de la trame sans bruit dans le domaine cepstral, $y_t$ la trame bruitée observée à l'instant $t$, et $w(RSB)$ un *offset* dépendant du rapport signal sur bruit mesuré à l'instant $t$ ; les vecteurs $w(RSB)$ de correction sont estimés en utilisant le critère des moindres carrés :

$$w[j,k] = \frac{\Sigma_{t=1}^{T}(s_t[j] - y_t[j])\delta[RSB_t - k\Delta_{RSB}]}{\Sigma_{t=1}^{T}\delta[RSB_t - k\Delta_{RSB}]}$$

où $w[j,k]$ désigne la $j^{ème}$ composante du vecteur de correction correspondant au rapport signal sur bruit $k \times \Delta_{RSB}$, $s_t$ et $y_t$ correspondent aux vecteurs cepstraux de la trame à l'instant $t$ du signal stéréophonique. $\Delta_{RSB}$ est le pas de discritisation de l'espace continu des rapports signal sur bruit. Toutes les trames de la base stéréophonique sont utilisées pour réaliser cette estimation.

Le grand inconvénient de ce genre de techniques est qu'elles supposent la connaissance *a priori* de l'environnement et du type de microphone en phase d'utilisation du système. La nécessité de données stéréophoniques est une contrainte très forte. Il est cependant possible d'utiliser le même principe mais avec beaucoup moins de contraintes. L'algorithme CDCN du paragraphe suivant constitue une solution à ce problème.

### 4.3.4 La CDCN (*Codeword-Dependent Cepstral Normalization*)

L'algorithme CDCN (*Codeword-Dependent Cepstral Normalization*) est dû à Acero [1]. Il consiste à représenter l'espace acoustique d'apprentissage supposé propre par un répertoire de vecteurs[8] dans le domaine cepstral (une somme de gaussiennes), puis à estimer le bruit additif $n$ et convolutif $h$ qui rapprochent le plus possible l'espace de test de l'espace d'apprentissage. L'espace de test est représenté par les trames de la phrase de test dans le domaine cepstral.

La CDCN repose sur un modèle du canal de transmission avec deux types de bruits : additif et convolutif, i.e. le signal observé est $y = s*h+n$, où $n$ et $h$ désignent respectivement les bruits additif et convolutif. Dans le domaine cepstral on obtient :

$$y_{cp} = s_{cp} + h_{cp} + r(s_{cp}, n_{cp}, h_{cp})$$

$$r(s_{cp}, n_{cp}, h_{cp}) = IDFT\left[\log(1 + exp(n_{cp} - h_{cp} - s_{cp}))\right]$$

Les valeurs de $n$ et $h$ sont celles qui maximisent $\log(p(Y|h_{cp}, n_{cp}))$, où $Y = y_0, \ldots, y_{n-1}$ est la séquence de trames constituant la phrase dans le domaine cepstral. Cette maximisation de la vraisemblance étant difficile à réaliser directement par la méthode du gradient, on a recours à l'algorithme EM pour trouver les valeurs de $n$ et $h$ maximisant $\log(p(Y|h_{cp}, n_{cp}))$. Une fois les deux paramètres d'environnement $n$ et $h$ estimés, on applique le critère des moindres carrés pour estimer la séquence $S = s_0, \ldots, s_{n-1}$ connaissant la séquence $Y = y_0, \ldots, y_{n-1}$ et les paramètres $n$ et $h$. En appliquant le critère des moindres carrés, on obtient :

$$\hat{s}_{MC} = E\left[s|y, n, h\right]$$

où $E[X]$ désigne l'espérance de la variable aléatoire $X$. La détermination de $E[s|y, n, h]$ fait intervenir le code-book représentant l'espace d'apprentissage. Certaines approximations dans le calcul de cette espérance nécessitent l'utilisation d'un code-book assez fin (un nombre de gaussiennes assez grand). Acero utilise un code-book d'apprentissage de 128 gaussiennes. Cette technique semble donner de bons résultats pour des rapports signal sur bruit moyens, mais ne semble pas être efficace dans le cas de rapport signal sur bruit faible. Ceci est certainement dû à l'utilisation d'un modèle très simpliste pour représenter la parole (mélange de gaussiennes). Il serait donc judicieux de penser qu'un modèle plus précis (MMC) donnerait des résultats plus satisfaisants.

### 4.3.5 Débruitage dépendant de l'état dans un MMC

Une meilleure estimation du signal propre peut être obtenue si on dispose de statistiques assez précises concernant le signal propre et le bruit qui a affecté le signal de parole. Supposons que le modèle statistique du bruit puisse être obtenu en enregistrant, par exemple, le bruit durant les périodes de silence. Il reste à estimer les statistiques correspondant au signal propre. Une façon de faire ceci, est d'exécuter une première passe de reconnaissance, qui permet d'avoir un alignement trames/gaussiennes. Ainsi, nous disposons de statistiques relatives au signal propre, puisque le système est supposé être appris sur des signaux propres. Cependant, pour obtenir un alignement trames/gaussiennes le plus correct possible, il est préférable

[8] Le terme couramment utilisé est *"code-book"*, c'est ce terme que nous allons utiliser dans le reste de ce document.

d'utiliser pour cette première passe de reconnaissance un système appris sur des signaux bruités avec le même bruit que celui qui a affecté le signal correspondant à la phrase de test bruitée. Pour obtenir un tel système de reconnaissance, il est possible d'utiliser une approche de type "combinaison parallèle de modèles" [22, 23] ou une approche de type "combinaison parallèle de données" [56] (voir chapitre 5). Bien sûr, il faut garder une correspondance entre les modèles du système appris sur les signaux propres et les modèles du système adapté au bruit ambiant. Une fois les statistiques correspondant à la parole propre obtenues, une estimation de la trame propre $x(t)$ dans le domaine spectral, connaissant la trame bruitée $y(t)$ et des statistiques relatives à $x(t)$ et au bruit, est donnée comme suit [73] :

$$\hat{x}_i(t) = \mu_i^p + \frac{\Sigma_{ii}^p}{\Sigma_{ii}^p + \Sigma_{ii}^b}(y_i(t) - \mu_i^p - \mu_i^b) \tag{4.2}$$

où $\mu^p$ et $\Sigma^p$ désignent, respectivement, la moyenne et la matrice de covariance de la gaussienne, dans le domaine spectral, qui correspond à la trame $y(t)$ dans l'alignement par le système de reconnaissance. $\mu^b$ et $\Sigma^b$ désignent, respectivement, la moyenne et la matrice de covariance de la gaussienne du bruit dans le domaine spectral. L'indice $i$ désigne la $i^{ème}$ composante.

Une approche utilisant le même principe, mais avec des Modèles de Markov dont les densités de probabilités associées aux états sont des sommes de gaussiennes Auto-Régressives (AR), donne des résultats satisfaisants [18].

### 4.3.6 Filtrage Stochastique Optimal

L'idée principale du filtrage stochastique optimal est l'utilisation d'une transformation linéaire faisant correspondre deux espaces acoustiques, $\mathbf{x}$ supposé propre et $\mathbf{y}$ supposé bruité. Cette transformation est appelé POF (*Probabilistic Optimum Filtering*) [62]. En général, l'espace acoustique propre $\mathbf{x}$ est utilisé pour l'apprentissage du système de reconnaissance ; et l'espace bruité $\mathbf{y}$ est utilisé pour le test. L'application de cette technique se fait en deux étapes, la première, est une phase d'apprentissage, qui consiste à estimer une transformation faisant correspondre les deux espaces $\mathbf{x}$ et $\mathbf{y}$, la deuxième, est une phase de test qui consiste à appliquer cette transformation estimée à un signal bruité appartenant à l'espace $\mathbf{y}$ afin d'obtenir une estimation de son correspondant dans l'espace $\mathbf{x}$. Soit $y_n$ une trame bruitée de l'espace $\mathbf{y}$, l'objectif est d'estimer la trame propre $\hat{x}_n$ qui lui correspond dans l'espace $\mathbf{x}$ ($n$ indique l'indice de la trame dans une phrase). Pour cela l'espace $\mathbf{y}$ est réparti en plusieurs sous-espaces, et à chaque sous-espace correspond une transformation. Supposons qu'il y ait $I$ sous-espaces, soient $g_{i,(1 \leq i \leq I)}$. La transformation associée au sous-espace $g_i$ est donnée comme suit :

$$\hat{x}_n = \mathbf{W_i^T Y_n} \tag{4.3}$$

avec

$$\mathbf{W_i^T} = [A_{i,-p}, \ldots, A_{i,-1}, A_{i,0}, A_{i,1}, \ldots, A_{i,p}, b_i] \tag{4.4}$$

et

$$\mathbf{Y_n^T} = \left[y_{n-p}^T, \ldots, y_{n-1}^T, y_n^T, y_{n+1}^T, \ldots, y_{n+p}^T, 1\right] \tag{4.5}$$

L'erreur commise par cette estimation est :

$$e_{ni} = x_n - \hat{x}_n \tag{4.6}$$

L'estimation de la transformation $\mathbf{W_i^T}$ associée au sous-espace $g_i$ est réalisée en minimisant l'erreur quadratique conditionnelle :

$$E_i = \sum_{n=p}^{N-1-p} \|e_{ni}\|^2 p(g_i|y_n) \tag{4.7}$$

La transformation qui minimise l'erreur quadratique conditionnelle est donnée comme suit :

$$\mathbf{W_i^T} = \mathbf{R_i^{-1} r_i} \tag{4.8}$$

avec

$$\mathbf{R_i} = \sum_{n=p}^{N-1-p} \mathbf{Y_n Y_n^T} p(g_i|y_n) \tag{4.9}$$

$$\mathbf{r_i} = \sum_{n=p}^{N-1-p} \mathbf{Y_n} x_n^T p(g_i|y_n). \tag{4.10}$$

Ainsi, l'estimation de la trame propre $\hat{x}_n$ correspondant à la trame bruitée $y_n$ est donnée comme suit :

$$\hat{x}_n = \left\{ \sum_{i=1}^{i=I} \mathbf{W_i^T} p(g_i|y_n) \right\} \mathbf{Y_n} \tag{4.11}$$

Notons que l'inconvénient majeur de cette technique est la nécessité de disposer de données stéréophoniques pour tous les environnements de test où on veut utiliser le système. Ceci constitue une contrainte très forte. En plus de cette contrainte, l'algorithme nécessite la connaissance *a priori* de l'environnement de test. Ceci constitue une contrainte supplémentaire, difficile à satisfaire.

## 4.4 Adaptation des modèles

### 4.4.1 Masquage et la décomposition parole/bruit

L'idée du masquage du bruit [80, 61] peut se résumer comme suit : soit $S_t^{'}(\nu)$ et $N_t^{'}(\nu)$ les niveaux d'énergie de la parole et du bruit (respectivement), dans un canal d'un banc de filtres. Le niveau du log-énergie du signal bruité $Y_t(\nu)$ dans le même canal peut s'écrire :

$$\begin{aligned} Y_t(\nu) &= \log(S_t^{'}(\nu) + N_t^{'}(\nu)) && (4.12) \\ &\simeq max\left[\log(S_t^{'}(\nu)), log(N_t^{'}(\nu))\right] && (4.13) \\ &\simeq max\left[S_t(\nu), N_t(\nu)\right], && (4.14) \end{aligned}$$

avec $S_t(\nu) = \log(S_t^{'}(\nu))$ et $N_t(\nu) = \log(N_t^{'}(\nu))$. Cette idée était à l'origine de plusieurs méthodes de compensation actuellement utilisées dans les systèmes de reconnaissance. En effet, ce même principe a été étendu pour donner lieu à une méthode optimale pour reconnaître la parole et le bruit simultanément [80, 61]. La technique fondée sur cette approximation développée par Varga [80] est appelée "Décomposition de la Parole et du Bruit" (DPB). Dans ce cadre l'évaluation de la probabilité d'observer $Y_t$ utilisant les modèles de la parole (MMC) et le(s) modèle(s) du bruit (MMC) se fait comme suit :

$$\begin{aligned} P(Y_t(\nu)|M_S, M_N) &\simeq P(max(S_t(\nu), N_t(\nu))|M_S, M_N) \quad (4.15)\\ &\simeq C(S_t(\nu), \mu_S, \sigma_S^2) P(N_t(\nu)|\mu_N, \sigma_N^2)\\ &+ C(N_t(\nu), \mu_N, \sigma_N^2) P(S_t(\nu)|\mu_S, \sigma_S^2) \quad (4.16) \end{aligned}$$

$M_S$ et $M_N$ désignent respectivement les modèles (des gaussiennes) de la parole propre et du bruit. $P(X|\mu, \sigma^2)$ désigne la valeur en $X$ de la densité de probabilité de la loi normale de moyenne $\mu$ et de variance $\sigma^2$. $C(X, \mu, \sigma^2)$ est la probabilité cumulative de toutes les observations dont le niveau est supérieur à $X$ et qui proviennent de la loi normale de moyenne $\mu$ et de variance $\sigma^2$. La validité de cette opération de combinaison repose sur l'hypothèse que l'effet du bruit sur un canal donné est indépendant de celui qui affecte le reste des canaux, ou du moins, a une influence moindre sur le reste des canaux.

L'approximation faite par Varga [80] donne une bonne estimation de la densité de probabilité de la parole bruitée. Cependant, elle suppose que le bruit agit indépendamment d'un canal à l'autre dans le domaine log-spectral. En plus cette technique impose l'utilisation de matrices de covariance pleines pour modéliser la corrélation entre les composantes. La Combinaison Parallèle de Modèles (CPM) est une alternative pour pallier ce problème ; elle consiste à opérer dans le domaine cepstral où les composantes cepstrales sont relativement décorrélées.

### 4.4.2 Combinaison Parallèle de Modèles

La Combinaison Parallèle de Modèles (CPM) [22, 23] a été introduite pour surmonter les limitations de la DPB (voir paragraphe précédent). Elle est cependant directement inspirée de cette dernière. Elle procède à toutes les opérations inverses pour revenir du domaine cepstral au domaine spectral (changement des densités de probabilités suivant l'opération) et inversement.

L'approximation log-normale [22, 23] est souvent utilisée pour revenir du domaine spectral au domaine cepstral. Elle suppose que la somme de deux variables aléatoires log-normalement distribuées est elle-même log-normalement distribuée. Il est important de noter que l'approximation log-normale devient inacceptable dans le cas de grandes variances.

La différence principale entre la DPB et la CPM est le type d'approximation : la CPM utilise soit l'approximation log-normale, soit procède à une intégration numérique, soit par génération des données (*DDPMC: Data Driven Parallel Model Combination*) [25], par contre la DPB utilise l'approximation du MAX définie précédemment. Une autre différence est que la DPB opère dans le domaine log-spectral alors que la CPM opère dans le domaine cepstral. Il est important de noter que la DPB augmente considérablement le nombre de paramètres dans le système. La difficulté des calculs et le nombre d'approximations dans le cadre de la CPM augmentent rapidement avec la complexité des formules utilisées pour les paramètres cepstraux différentiels, ce qui représente un grand inconvénient de l'approche. Cette technique sera décrite d'une manière plus détaillée dans le chapitre 5.

### 4.4.3 L'appariement stochastique

L'appariement stochastique est une technique proposée par Ananth Sankar et Chin-Hui Lee [72], elle consiste à utiliser le critère du maximum de vraisemblance pour adapter les modèles à l'environnement où le signal de test a été généré (ou inversement). Cette adaptation se fait par le biais d'une transformation appliquée ou bien sur les modèles ou bien sur le signal de test. Le choix de la forme de la transformation peut tirer profit d'une connaissance *a priori* du bruit affectant le signal de parole.

Plus formellement, soit $\Lambda_X = \{\lambda_{x_i}\}$ un ensemble de modèles, représentant des classes de formes, appris sur un corpus d'apprentissage $X$, où $\lambda_{x_i}$ représente le modèle associé à la $i^{ème}$ classe. Soit $Y = \{y_1, y_2, \ldots, y_T\}$ une observation représentant le corpus de test. On a choisi deux symboles différents $X$ et $Y$ pour désigner le corpus de test et le corpus d'apprentissage afin de mettre en évidence la différence entre les deux espaces. Cette différence peut être la conséquence de différentes variabilités acoustiques entre le test et l'apprentissage (bruit ambiant, changement de microphone, échos, ...). L'objectif est de reconnaître la suite d'événements (les indices des classes) $W = w_1, w_2, \ldots, w_L$ cachés dans $Y$. En reconnaissance de la parole, $\lambda_{x_i}$ peut être un MMC correspondant à la $i^{ème}$ unité phonétique et $Y$ la phrase à reconnaître. Pour décoder le message contenu dans $Y$ (suite d'unités phonétiques), le décodeur du Maximum *a posteriori* est utilisé comme suit :

$$\hat{W} = \arg\max_{W} P(Y|W, \Lambda_X)P(W) \tag{4.17}$$

La différence entre les espaces $X$ et $Y$ peut être vue comme une fonction paramétrique $G_\nu$ qui fait correspondre à l'ensemble des modèles $\Lambda_X$ un autre ensemble de modèle $\Lambda_Y$ qui donne une meilleure interprétation à $Y$ (qui génère $Y$ avec une plus grande vraisemblance).

$$\Lambda_Y = G_\nu(\Lambda_X) \tag{4.18}$$

Il faut donc trouver l'ensemble des paramètres $\nu$ et la succession d'unités acoustiques $W$ tels que :

$$\begin{aligned}(\hat{\nu}, \hat{W}) &= \arg\max_{(\nu,W)} P(Y|W, G_\nu(\Lambda_X))P(W) && (4.19)\\ &= \arg\max_{(\nu,W)} P(Y|W, \nu, \Lambda_X)P(W) && (4.20)\end{aligned}$$

Cette maximisation peut être réalisée itérativement par rapport à $\nu$ en supposant $W$ fixe, puis par rapport à $W$ en supposant $\nu$ fixe. La maximisation par rapport à $W$ en supposant $\nu$ fixe se fait en utilisant les algorithmes classiques de décodage.

La maximisation par rapport à $\nu$ en supposant $W$ fixe, utilise l'algorithme EM (Expectation and Maximisation) [15]. L'algorithme EM est une procédure itérative à deux étapes, la première (étape E) consiste à calculer la fonction auxiliaire :

$$Q(\nu', \nu) = E\left\{log(Y, S, C|\nu', \Lambda_X)|Y, \nu, \Lambda_X\right\} \tag{4.21}$$

où $S = \{s_1, s_2, \ldots, s_T\}$ est l'ensemble de toutes les séquences d'états possibles et $C = \{c_1, c_2, \ldots, c_T\}$ l'ensemble de toutes les séquences de gaussiennes possibles.

La seconde étape, appelée étape de maximisation (étape M), consiste à trouver $\nu'$ qui maximise la fonction auxiliaire $Q(\nu', \nu)$.

La difficulté de cette approche augmente avec la complexité des transformations utilisées. Les auteurs ont dérivé les formules d'adaptation pour des transformations simplistes de type :

$$\mu_y = \mu_x + \mu_b \tag{4.22}$$

$$\sigma_y^2 = \sigma_x^2 + \sigma_b^2 \tag{4.23}$$

Cette technique peut être vue de deux manières différentes. Soit comme une technique qui permet un décodage optimal de l'observation $Y$. Dans ce cas il faut réestimer pour chaque phrase une transformation qui lui est spécifique, ce qui peut être très coûteux. Soit comme une technique d'adaptation, celle-ci ne serait pas optimale lorsqu'on dispose d'une quantité importante de données d'adaptation, car elle utilise la même transformation pour adapter tous les paramètres du système.

La MLLR (Maximum Likelihood Linear Regression) est une technique d'adaptation plus générale que l'alignement stochastique, elle permet des transformations d'adaptation des modèles plus complexes et plus générales, les transformations correspondant à cette approche tiennent compte de la quantité de données d'adaptation à notre disposition, ce qui la rend plus optimale. Dans le paragraphe suivant nous présentons le principe de la MLLR.

### 4.4.4 L'adaptation par régression linéaire (MLLR)

L'adaptation par régression linéaire est une technique ne supposant aucun modèle *a priori*, elle utilise un critère d'optimisation généralement fondé sur le maximum de vraisemblance pour estimer des transformations à appliquer sur les modèles acoustiques (MMCDC) du système de reconnaissance afin de les adapter à l'environnement d'utilisation (locuteur inclu). Les transformations utilisées sont généralement des régressions linéaires estimées en utilisant le maximum de vraisemblance d'où le nom en anglais MLLR (Maximum Likelihood Linear Regression) [45]. Dans ce cadre, l'estimation des paramètres des transformations est optimale dans le sens où elle tient compte de la quantité de données d'adaptation disponibles. En effet, suivant la quantité de données d'adaptation disponible, on utilise des transformations plus ou moins globales.

Vu l'intérêt de cette technique nous avons fait le choix de l'étudier plus en détail, nous avons étudié la possibilité d'étendre cette technique à des transformations plus complexes prenant en compte la variance des données. Le chapitre 7 est consacré à cette étude ; des résultats expérimentaux sont présentés.

# Chapitre 5

# Adaptation des Systèmes de Reconnaissance par Bruitage

## 5.1 Introduction

Plusieurs auteurs ont montré que les meilleurs résultats, en terme de taux de reconnaissance, sont obtenus lorsque l'apprentissage et le test du système de reconnaissance se déroulent dans les mêmes conditions [39] [59]. Supposons que les modèles acoustiques sont entraînés sur des signaux enregistrés dans une ambiance calme, et que les signaux de test sont bruités. La question est la suivante : que faire pour compenser au mieux les différences existantes entre l'apprentissage et le test ? deux réponses sont possibles :

- 1) débruiter le signal de test pour atteindre les mêmes performances que celles qu'on obtiendrait en faisant le test dans les mêmes conditions que celles de l'apprentissage ; c'est-à-dire dans des conditions calmes.

- 2) bruiter les références avec le bruit de test pour atteindre les mêmes performances que celles qu'on obtiendrait en faisant l'apprentissage dans les mêmes conditions de bruit que celles du test.

D'une manière plus formelle, notons $\lambda_x$ l'ensemble des modèles acoustiques entraînés sur des signaux propres $x$. La densité de probabilité correspondant à $x$ est une fonction probabiliste d'une chaîne de Markov notée $f_{\lambda_x}(x)$. $\lambda_x$ est estimé en utilisant le critère du maximum de vraisemblance, ce qui implique que le classifieur bayèsien sur les signaux $x$ utilisant les modèles $\lambda_x$ est optimal en terme de taux de reconnaissance. Soit $\lambda_n$ le modèle de bruit qui affecte les signaux de test $x$ résultant en un signal de test bruité[1] $y = x + n$. Soit $\hat{\lambda}_y$ l'ensemble des modèles acoustiques obtenus en combinant (bruitage) les modèles $\lambda_x$ avec le modèle de bruit $\lambda_n$ :

$$\hat{\lambda}_y = \lambda_x \oplus \lambda_n \tag{5.1}$$

[1] Pour des raisons de clarté, le bruit convolutif $h$ est ignoré.

où $\oplus$ désigne l'opération de combinaison (bruitage). Puisqu'on ne dispose que du signal $y$ bruité et des statistiques concernant $x$ et $n$, nous avons deux possibilités pour décoder le message linguistique caché dans le signal $y$:

- 1) la première possibilité est d'estimer le signal propre $\hat{x}$ correspondant au signal bruité $y$ et d'utiliser $\lambda_x$ pour décodage ; la suite de mots cachée dans $y$ est alors :
$$\hat{w} = \operatorname*{argmax}_{w} f_{\lambda_x}(\hat{x}|w)p(w), \tag{5.2}$$
où $p(w)$ désigne la probabilité *a priori* de la séquence de mots $w$ (issue du modèle de langage).

- 2) la deuxième possibilité est d'utiliser directement le signal bruité $y$, mais avec les modèles acoustiques $\lambda_y$. Or, nous ne disposons que d'une estimation $\hat{\lambda}_y$ de ce modèle (obtenue en faisant l'opération de combinaison $\oplus$) ; le classifieur bayèsien (qui est optimal si les modèles acoustiques sont corrects) donne, dans ce cas, la séquence de mots $\hat{w}$ :
$$\hat{w} = \operatorname*{argmax}_{w} f_{\hat{\lambda}_y}(y|w)p(w) \tag{5.3}$$

La question devient : quel est le système de reconnaissance, parmi les deux systèmes fondés sur les équations 5.2 et 5.3, qui donne les meilleures performances en terme de taux de reconnaissance ? Dans le cas où $\hat{\lambda}_y$ est très proche de $\lambda_y$ (estimé en utilisant le maximum de vraisemblance) et où l'estimation de $\hat{x}$ est optimale au sens du Maximum *A Posteriori* (MAP)[2], les performances en terme de taux de reconnaissance des deux systèmes doivent être similaires.

Il est évident que si l'estimation de $\hat{x}$ n'est pas optimale au sens du maximum *a posteriori*, alors il y a de fortes chances que la solution 2 soit largement plus performante que la solution 1. Ce cas se produit avec beaucoup de techniques de débruitage, comme les techniques de type soustraction spectrale (filtrage du bruit) [9, 13, 49] ou l'algorithme CDCN (*Codeword-Dependent Cepstral Normalization*) dû à Acero [1] dont l'efficacité diminue rapidement avec le RSB. Pendant une vingtaine d'années la recherche s'est poursuivie dans ce sens, dans le cadre de la reconnaissance de la parole, les résultats obtenus en suivant cet axe de recherche n'ont pas donné le succès escompté. L'erreur commise par ce type de technique est due à l'espoir de reconstituer correctement l'information dégradée ou perdue à cause du bruit sans introduire toute l'information dont on dispose, c'est-à-dire ($\lambda_x$, $y$, $\lambda_n$).

La solution 1 est plus difficile à réaliser correctement que la solution 2, c'est-à-dire qu'il est plus facile d'estimer correctement $\hat{\lambda}_y$ que d'estimer correctement $\hat{x}$. En effet, comme nous allons le voir dans le chapitre 6, la réalisation de la solution 1 passe forcément par la réalisation de la solution 2. Cette discussion nous permet de conclure que, pour la reconnaissance de la parole, la solution 2 est plus appropriée que la solution 1. Dans ce chapitre nous nous intéressons à la réalisation de la solution 2, c'est-à-dire l'estimation de $\hat{\lambda}_y = \lambda_x \oplus \lambda_n$ puis le décodage du message linguistique caché dans le signal bruité $y$. La solution 1 sera abordée dans le chapitre 6 dans le

[2] $\hat{x} = \operatorname{argmax}_x f(x|y, \lambda_x, \lambda_n)$

but principal d'améliorer la qualité des signaux de parole dégradés par la présence d'un bruit ambiant.

Plusieurs chercheurs ont montré que la solution 2 est plus appropriée que la solution 1 : pour une reconnaissance en milieu bruité par un système entraîné en milieu propre, il est préférable de bruiter les références avec le bruit présent en phase de test [59]. Cette deuxième approche fonctionne mieux que la première, dans une application de reconnaissance, même si elle est moins intuitive. C. Mokbel [59] donne une justification théorique et expérimentale de la supériorité des techniques de bruitage par rapport aux techniques de débruitage. Cette justification ne peut être correcte que pour des techniques de débruitage utilisant des estimateurs non optimaux au sens du Maximum *a posteriori*[3].

Plusieurs travaux ont été réalisés pour adapter un système de reconnaissance à l'environnement du test en transformant les références [37, 71, 61, 60, 59, 22, 23]. Partout où cette approche a été appliquée, des résultats encourageants ont été observés. Dans ce cadre, la technique qui a montré le plus grand intérêt et qui est la plus utilisée actuellement est la Combinaison Parallèle de Modèles (CPM), qui est appliquée dans le cadre de l'adaptation des systèmes de reconnaissance fondés sur les Modèles de Markov Cachés à Distributions Continues (MMCDC), le terme anglophone introduit par M.J.F. Gales et S.J. Young correspondant à cette technique est *"Parallel Model Combination"* (PMC) [22, 23].

Dans la première section de ce chapitre, nous décrivons la CPM que nous avons implémentée au LIMSI et nous présentons ses avantages et ses inconvénients. Nous montrons comment adapter les paramètres statistiques relatifs aux coefficients *cepstraux* dynamiques ($\Delta cepstre$, $\Delta^2 cepstre$) et comment compenser le bruit convolutif dans le cadre de la CPM.

Pour résoudre les problèmes liés à la CPM, nous proposons dans dans section 5.3 une nouvelle technique de bruitage des références[4]. Cette nouvelle approche compense explicitement les deux types de bruits additif et convolutif et n'utilise aucune connaissance *a priori* sur les bruits affectant les données de test. Cela veut dire que la compensation se fait en deux étapes, la première est l'estimation des bruits à partir des données de test, la deuxième est la compensation proprement dite. Pour montrer l'apport de notre approche, des résultats expérimentaux seront présentés dans différentes configurations.

Dans la section 5.6 de ce chapitre nous abordons le problème de présence de bruits en phases de test et d'apprentissage. La plupart des techniques proposées pour appréhender le problème de bruit ambiant supposent que les données d'apprentissage sont propres et que celles de test sont bruitées. Cette hypothèse peut être vérifiée si l'on impose suffisamment de contraintes durant l'enregistrement des données d'apprentissage : utilisation d'un même microphone pour toutes les données d'apprentissage, enregistrement dans une salle insonorisée, *etc.* Il est évident que ce

[3] Nous montrons dans le chapitre 6 consacré au débruitage qu'il est possible d'obtenir de meilleures performances en utilisant la solution 1, à condition d'introduire de l'information obtenue en utilisant la solution 2. Cela est certainement dû au fait que l'estimation (par bruitage) des modèles représentant la parole bruitée n'est pas optimale au sens du maximum de vraisemblance.

[4] les références sont des MMCDC.

type de contraintes ne peut pas être vérifié dans bon nombre d'applications. Pour surmonter ce problème nous proposons une technique itérative permettant d'estimer et de compenser les bruits de test et les bruits d'apprentissage. Des résultats expérimentaux seront présentés dans différentes configurations, afin de montrer l'apport de notre approche.

## 5.2 La Combinaison Parallèle de Modèles (CPM)

La parole bruitée est un phénomène complexe qui résulte de l'interaction entre la source $S$ de la parole propre et la source $N$ de bruit. La source de la parole bruitée résultante va être désignée par le symbole $R$, les indices $cp$, $lg$, et $sp$ vont désigner respectivement les domaines *cepstral*, *log-spectral* et spectral. Une densité gaussienne va être représentée par $N(\mu, \Sigma)$, où $\mu$ désigne le vecteur moyen et où $\Sigma = \{\sigma_{ij} | 0 \leq i \leq p, 0 \leq j \leq p\}$ désigne la matrice de covariance. $X_b$ désigne la variable aléatoire $X$ dans le domaine $b$ où $b \in \{cp, lg, sp\}$. Par exemple, la parole bruitée dans le domaine *cepstral* sera notée $R_{cp}$. Si cette variable aléatoire a une distribution gaussienne, alors celle-ci sera notée $R_{cp} = N(\mu^{R_{cp}}, \Sigma^{R_{cp}})$.

Supposons que le système de reconnaissance qui fait l'objet de l'adaptation soit appris sur des signaux de la parole propre et utilise des MMCDC. Pour modéliser la parole bruitée, on combine les modèles représentant la parole propre et le(s) modèle(s) représentant le bruit. On obtient d'autres modèles que nous appelons MMCDC-PCB (MMCDC de la Parole Combinée avec le Bruit : MMCDC-PCB). Le terme “combiner” veut dire déduire les paramètres des modèles statistiques représentant la parole bruitée à partir des paramètres des modèles statistiques représentant la parole propre et des paramètres des modèles statistiques représentant le bruit, au lieu de les estimer directement sur le signal bruité. L'avantage de cette approche est d'éviter de refaire l'apprentissage à chaque fois que les caractéristiques du bruit changent significativement. L'adaptation au nouvel environnement peut se faire de deux manières différentes :

- Soit avoir une base de données des bruits typiques, et à chaque fois qu'on constate un changement des caractéristiques du bruit, on extrait le modèle de bruit le plus approprié de la base des bruits pour le combiner avec les modèles de la parole propre.

- Soit construire des modèles de bruit directement pendant les périodes de silence en phase d'utilisation. Cela est possible grâce à la taille réduite des modèles de bruit généralement utilisés.

### 5.2.1 Les paramètres du MMCDC-PCB

Etant données les références de la parole propre et du bruit (modélisées par des MMCDC), les paramètres du modèle composé s'obtiennent à partir de ceux des modèles de la parole propre et de ceux du bruit comme suit (voir Figure 5.1) :

- le nombre d'états est : $n^R = n^S n^N$, où $n^S$ et $n^N$ sont respectivement les nombres d'états dans le MMCDC de la parole propre et dans le MMCDC du bruit.

- Soient $Q^N$, $Q^S$ et $Q^R$ (respectivement) les ensembles d'états dans les MMCDC du bruit, de la parole propre et de la parole bruitée. l'application suivante permet de déterminer les états du MMCDC-PCB :

$$\begin{array}{rccc} \aleph : & Q^N \cdot Q^S & \rightarrow & Q^R \\ & (i_N, i_S) & \rightarrow & i_R = i_N n^S + i_S \end{array}$$

$$0 \leq i_N < n^N, \quad 0 \leq i_S < n^S, \quad 0 \leq i_R < n^R, \tag{5.4}$$

- la probabilité initiale de l'état $i_R = \aleph(i_N, i_S)$ est : ${\pi_{i_R}}^R = {\pi_{i_N}}^N {\pi_{i_S}}^S$.

- la probabilité de transition de l'état $i_R = \aleph(i_N, i_S)$ à l'état $j_R = \aleph(j_N, j_S)$ est : $a^R_{i_R j_R} = a^N_{i_N j_N} a^S_{i_S j_S}$.

- le nombre de gaussiennes associées à l'état $i_R = \aleph(i_N, i_S)$ est : ${M_{i_S}}^S {M_{i_N}}^N$, où $M^S_{i_S}$ et ${M_{i_N}}^N$ désignent respectivement le nombre de gaussiennes dans l'état $i_S$ du MMCDC de la parole propre et le nombre de gaussiennes dans l'état $i_N$ du MMCDC du bruit.

Tous les éléments du modèle composé (MMCDC-PCB) ont été décrits sauf les probabilités d'émission à partir d'un état donné. Pour déterminer cette dernière composante, il faut trouver un domaine où la relation entre les deux sources parole propre et bruit soit la plus simple possible. On choisit pour cela le domaine spectral où la relation entre ces deux sources est une simple addition[5] :

$$R_{sp} = N_{sp} + S_{sp} \tag{5.5}$$

$X_{sp}$ désigne la densité spectrale (ou le spectre de puissance) correspondant à une trame donnée. Partant du principe qu'on ne possède des statistiques que dans le domaine *cepstral*, il faut appliquer une série de transformations permettant d'avoir des statistiques dans le domaine spectral ; puis d'autres transformations permettant d'avoir des statistiques dans le domaine *cepstral* après combinaison. Bien sûr, ces transformations dépendent fortement du type des paramètres acoustiques utilisés. Dans cette étude, nous supposons que les paramètres acoustiques utilisés par le système de reconnaissance sont des vecteurs composés des coefficients MFCC (Mel scaled Frequency Cepstrum Coefficients) (voir section 2.1 concernant les paramétrisations acoustiques). Dans ce cas, les transformations pour revenir dans le domaine spectral sont fondées sur le fait que le *cepstre* (MFCC) est obtenu en appliquant une transformée de Fourier sur le logarithme du spectre. Une transformation simple sur

[5] Cette relation est valable dans le domaine des densités spectrales, lorsque le bruit additif dans le domaine temporel n'est pas corrélé avec le signal de parole.

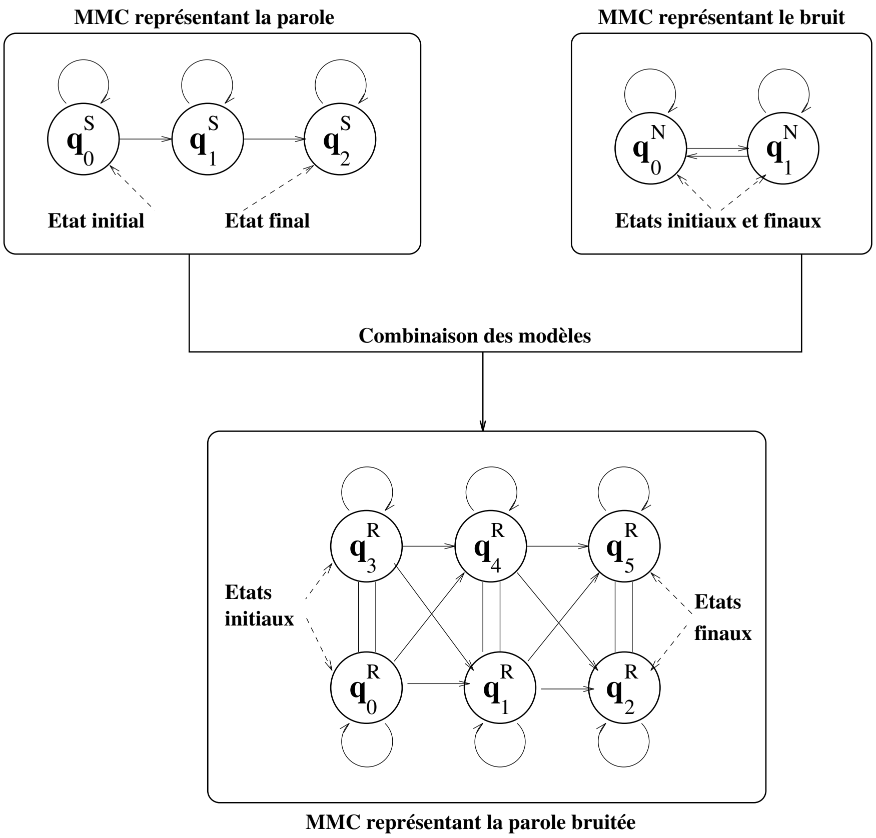



FIG. 5.1 - *Un exemple de la combinaison de modèles. Le nombre d'états du modèle résultant est le produit des nombres d'états dans les modèles représentant la parole propre et le bruit. De même, le nombre de gaussiennes dans un état donné du modèle résultant de la combinaison est le produit des nombres de gaussiennes dans les états des modèles de la parole propre et du bruit qui lui ont donné naissance.*

les données peut induire des transformations très complexes sur les densités de probabilités correspondantes. La complexité d'une telle transformation (sur les modèles) dépend fortement de la complexité de la densité de probabilité des données en question. Par exemple, au sein d'un état d'un MMCDC, la variable aléatoire générant les vecteurs *cepstraux* est un mélange de gaussiennes. Il est très difficile, voire impossible d'en déduire directement la densité de probabilité pour les mêmes données dans le domaine spectral. Pour résoudre ce problème, une première approximation s'impose, elle consiste à transformer chaque élément (gaussienne) du mélange de gaussiennes indépendamment des autres éléments. Après chaque transformation, la densité de probabilité dans le nouveau domaine est le mélange des transformées des densités élémentaires d'origine pondérées avec les mêmes poids que dans le domaine d'origine.

### 5.2.2 Estimation des probabilités d'émission

Un MMCDC modélise une densité de probabilité trop complexe pour être manipulée directement. Pour cela, on fait une approximation qui consiste à supposer que les états d'un MMCDC sont affectés par le bruit indépendamment les uns des autres. Cette supposition implique que ce type de compensation ne permet pas, par exemple, de prendre en compte la variabilité de la vitesse d'élocution due à l'effet Lombard. En dépit de cette approximation, la densité de probabilité au niveau de l'état, qui est un mélange de gaussiennes, reste complexe à manipuler. On fait donc une deuxième approximation qui consiste à supposer que chaque gaussienne du mélange de gaussiennes au niveau d'un état donné, est affectée par le bruit indépendamment des autres gaussiennes. Dans ce qui suit nous allons exposer les transformations concernant une seule gaussienne pour la parole propre et une seule gaussienne pour le bruit (voir Figure 5.2). Ces transformations ont pour but de passer du domaine *cepstral* où la relation entre les deux sources de parole propre et de bruit est complexe au domaine spectral où la relation entre les deux sources est une simple addition, puis de revenir au domaine *cepstral* après combinaison. Les transformations permettant de passer d'un domaine à l'autre dépendent du type de paramétrisation acoustique utilisée par le système de reconnaissance. Nous allons décrire celles qui correspondent à une paramétrisation de type MFCC. Les mêmes transformations sont valables pour une paramétrisation de type LPCC (Linear Predictive Cepstrum Coefficients) [55].

**La transformation en Cosinus**

Le passage du domaine *cepstral* au domaine *log-spectral* se fait par une transformation de Fourier. Dans notre cas, cette transformation se traduit par la multiplication par une matrice qu'on appelle la matrice de la transformée en Cosinus[6]. Notons $C$ la matrice de la transformation en Cosinus et $C_{ij}, (1 \leq i, j \leq p)$ ses éléments où $p$ est le nombre de coefficients *cepstraux* en comptant le $c_0$.

Pour déterminer les densités de probabilité des deux sources de bruit et parole propre dans le domaine *log-spectral*, on utilise le résultat suivant, bien connu en sta-

[6] Il s'agit de la transformée de Fourier d'une fonction réelle et paire.

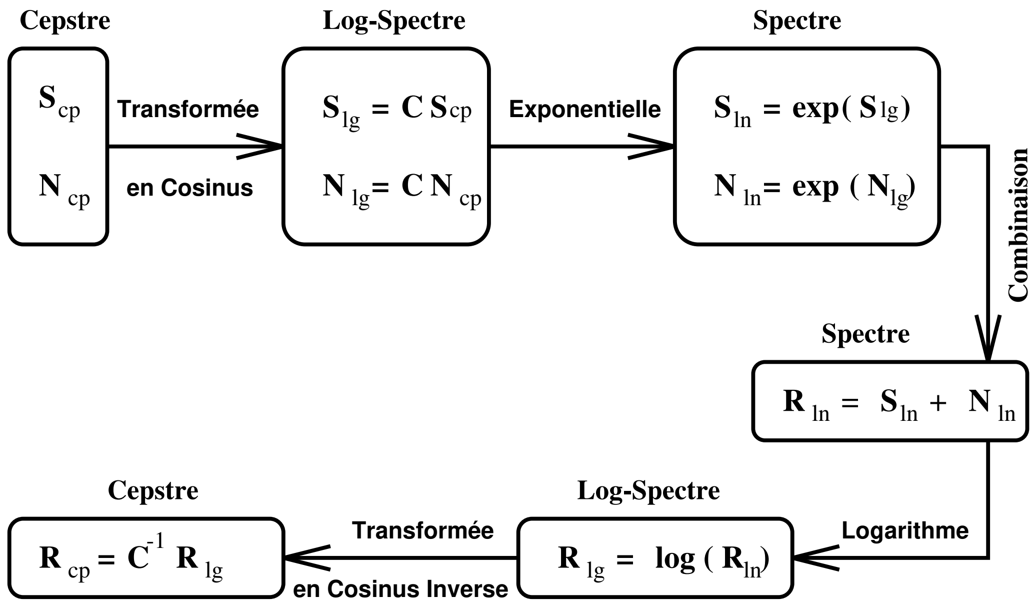



FIG. 5.2 - *Diagramme des transformations pour obtenir les MMCDC-PCB d'après F. Martin [55].*

tistique :

**Propriété 1 :** *Etant donné $X_1, ..., X_q$, $q$ variables aléatoires normalement distribuées dans $\mathbf{R}^n$, telles que:*

$$E[X_i] = \mu_i, \quad Covar[X_i, X_j] = \Sigma_{ij}, \quad (1 \leq i, j \leq q) \tag{5.6}$$

*Une combinaison linéaire $Y = \sum_{i=1}^{q} a_i X_i$ est aussi normalement distribuée avec :*

$$Y = N(\sum_{i=1}^{q} a_i \mu_i, \sum_{i=1}^{q} \sum_{j=1}^{q} a_i a_j \Sigma_{ij}) \tag{5.7}$$

*où $E[X]$ désigne l'espérance de la variable aléatoire $X$, $Covar[U, V]$ est la matrice de covariance des variables $U$ et $V$.*

Supposons que la source des trames de la parole (respectivement du bruit) dans le domaine *cepstral* est une variable aléatoire $S_{cp}$ (respectivement $N_{cp}$), dont la densité de probabilité est une loi normale : $f(s_{cp}) = N(\mu_{S_{cp}}, \Sigma_{S_{cp}})$ (respectivement $f(n_{cp}) = N(\mu_{N_{cp}}, \Sigma_{N_{cp}})$), $s_{cp}$ (respectivement $n_{cp}$) est une réalisation de $S_{cp}$ (respectivement $N_{cp}$).

On sait que la relation qui lie la variable $S_{cp}$ (respectivement $N_{cp}$) avec $S_{lg}$ (respectivement $N_{lg}$) est comme suit :

$$S_{lg} = CS_{cp} \quad respectivement \quad N_{lg} = CN_{cp}, \tag{5.8}$$

En utilisant la propriété 1, on peut affirmer que les deux variables aléatoires $N_{lg}$ et $S_{lg}$ déduites par une transformation linéaire des deux variables aléatoires normalement distribuées $N_{cp}$ et $S_{cp}$, sont elles-mêmes normalement distribuées ; leurs densités de probabilités sont respectivement $f(s_{lg}) = N(\mu_{S_{lg}}, \Sigma_{S_{lg}})$ et $f(n_{lg}) = N(\mu_{N_{lg}}, \Sigma_{N_{lg}})$. Les paramètres de ces deux distributions s'obtiennent comme suit :

$$\mu_u^{N_{lg}} = \sum_{j=1}^{p} C_{uj} \mu_j^{N_{cp}}, \quad \forall u \in \{1, \ldots, p\}, \tag{5.9}$$

$$\sigma_{uv}^{N_{lg}} = \sum_{i=1}^{p} \sum_{j=1}^{p} C_{ui} C_{vj} \sigma_{ij}^{N_{cp}}, \quad \forall u, v \in \{1, \ldots, p\}, \tag{5.10}$$

$$\mu_u^{S_{lg}} = \sum_{j=1}^{p} C_{uj} \mu_j^{S_{cp}}, \quad \forall u \in \{1, \ldots, p\}, \tag{5.11}$$

$$\sigma_{uv}^{S_{lg}} = \sum_{i=1}^{p} \sum_{j=1}^{p} C_{ui} C_{vj} \sigma_{ij}^{S_{cp}}, \quad \forall u, v \in \{1, \ldots, p\}. \tag{5.12}$$

Ces équations peuvent être exprimées sous formes matricielles :

$$\mu_{N_{lg}} = C \mu_{N_{cp}}, \tag{5.13}$$

$$\Sigma_{N_{lg}} = C \Sigma_{N_{cp}} C^t. \tag{5.14}$$

D'une manière similaire, on obtient pour le signal propre,

$$\mu_{S_{lg}} = C \mu_{S_{cp}}, \tag{5.15}$$

$$\Sigma_{S_{lg}} = C \Sigma_{S_{cp}} C^t. \tag{5.16}$$

### Application de l'exponentielle

Pour passer du domaine *log-spectral* au domaine spectral, il faut appliquer l'exponentielle. L'objectif est de déduire les densités de probabilité des variables aléatoires $S_{sp}$ et de $N_{sp}$ de celles de $S_{lg}$ et de $N_{lg}$. Ce problème se traduit par la détermination de la densité de probabilité de la variable aléatoire $Y = exp(X)$ lorsque $X$ est normalement distribuée. Cela est un résultat classique en statistique mathématique (voir annexe A) : la densité de probabilité de $Y$ est la loi *log-normale* avec les paramètres suivants :

$$\mu_Y = exp\left[\mu_X + \frac{\sigma_X}{2}\right], \tag{5.17}$$

$$\sigma_Y = \mu_X^2 \left[exp(\sigma_X) - 1\right]. \tag{5.18}$$

En appliquant ce résultat aux variables aléatoires $S_{lg}$ et $N_{lg}$, on aboutit à deux variables $S_{sp}$ et $N_{sp}$ *log-normalement* distribuées, avec les paramètres suivants :

$$\mu_u^{N_{sp}} = exp\left[\mu_u^{N_{lg}} + \frac{\sigma_{uu}^{N_{lg}}}{2}\right], \tag{5.19}$$

$$\sigma_{uv}^{N_{sp}} = \mu_u^{N_{sp}} \mu_v^{N_{sp}} \left[ exp(\sigma_{uv}^{N_{lg}}) - 1 \right], \tag{5.20}$$

avec $1 \leq u, v \leq p$ où $p$ est le nombre de coefficients *cepstraux*. D'une manière similaire on trouve pour le signal propre :

$$\mu_u^{S_{sp}} = exp \left[ \mu_u^{S_{lg}} + \frac{\sigma_{uu}^{S_{lg}}}{2} \right]$$
$$\sigma_{uv}^{S_{sp}} = \mu_u^{S_{sp}} \mu_v^{S_{sp}} \left[ exp(\sigma_{uv}^{S_{lg}}) - 1 \right], \quad 1 \leq u, v \leq p. \tag{5.21}$$

**Addition dans le domaine spectral**

En supposant que le signal de parole et le signal de bruit sont additifs et non corrélés, leurs densités spectrales sont alors additives :

$$R_{sp} = S_{sp} + N_{sp}. \tag{5.22}$$

En effet, nous avons :

$$r(t) = s(t) + n(t),$$

où $r$, $s$ et $n$ désignent respectivement, dans le domaine temporel, le signal de la parole bruitée, le signal de la parole propre et le signal de bruit. En calculant la fonction d'autocorrélation, on obtient :

$$E\left[r(t+\tau)r(t)\right] = E\left[s(t+\tau)s(t)\right] + E\left[n(t+\tau)n(t)\right] + E\left[s(t+\tau)n(t)\right] + E\left[n(t+\tau)s(t)\right]$$

puisque $s$ et $n$ sont non corrélés, on obtient :

$$E\left[r(t+\tau)r(t)\right] = E\left[s(t+\tau)s(t)\right] + E\left[n(t+\tau)n(t)\right],$$

puisque la densité spectrale et la fonction d'autocorrélation sont liées par une transformée de Fourier, on obtient :

$$R_{sp} = S_{sp} + N_{sp}.$$

où $R_{sp}$, $S_{sp}$ et $N_{sp}$ désignent respectivement les densités spectrales du signal de la parole bruitée, du signal de la parole propre et du signal de bruit. Il faut noter que toutes ces transformations sont réalisées sur des trames de longueurs assez petites pour qu'on puisse considérer les signaux étudiés comme étant stationnaires.

Pour continuer cette série de transformations afin de revenir dans le domaine *cepstral*, on fait une approximation supplémentaire qui consiste à considérer que la somme de deux variables aléatoires *log-normalement* distribuées est une variable aléatoire *log-normalement* distribuée. Dans notre cas, cela implique que $R_{sp}$ est *log-normalement* distribuée avec les paramètres suivants :

$$\mu_{R_{sp}} = \mu_{S_{sp}} + \mu_{N_{sp}}$$
$$\Sigma_{R_{sp}} = \Sigma_{S_{sp}} + \Sigma_{N_{sp}}. \tag{5.23}$$

Dans la suite de ce document, nous allons désigner cette approximation par le terme “approximation *log-normale*”.

**Application du logarithme**

L'approximation *log-normale* permet d'affirmer que la variable aléatoire $R_{sp}$ est *log-normalement* distribuée, ce qui est équivalent à dire que $R_{lg}$ est normalement distribuée. Avec cette approximation, les paramètres de $R_{lg}$ sont obtenus en inversant les équations (6.15) et (7.20) qui ont été utilisées pour la transformation exponentielle. La combinaison de la source de bruit avec celle de la parole est donc une variable aléatoire $R_{lg}$ normalement distribuée dans le domaine *log-spectral*; ses paramètres sont donnés comme suit :

$$\mu_u^{R_{lg}} = \log\left[\mu_u^{R_{sp}}\right] - \frac{1}{2}\log\left[\frac{\sigma_{uu}^{R_{sp}}}{\mu_u^{R_{lg}}\mu_u^{R_{lg}}} + 1\right] \quad \forall u \in \{1,\ldots,p\} \tag{5.24}$$

$$\sigma_{uv}^{R_{lg}} = \log\left[\frac{\sigma_{uv}^{R_{sp}}}{\mu_u^{R_{lg}}\mu_v^{R_{lg}}} + 1\right] \quad \forall u,v \in \{1,\ldots,p\} \tag{5.25}$$

**Transformation en Cosinus inverse**

Pour passer du domaine *log-spectral* au domaine *cepstral* il faut multiplier par l'inverse de la matrice en Cosinus citée précédemment:

$$R_{cp} = C^{-1}R_{lg}. \tag{5.26}$$

La variable aléatoire $R_{lg}$ est normalement distribuée, on peut donc appliquer à nouveau la propriété 1. On obtient dans le domaine *cepstral* une variable aléatoire $R_{cp}$ distribuée normalement avec les paramètres suivants :

$$\mu_{R_{cp}} = C^{-1}\mu^{R_{lg}} \tag{5.27}$$

$$\Sigma^{R_{cp}} = C^{-1}\Sigma_{R_{lg}}(C^{-1})^t. \tag{5.28}$$

Jusque là, seuls les paramètres cepstraux statiques sont compensés. Gales [25] propose pour la compensation des coefficients *cepstraux* dynamiques une méthode faisant appel à des techniques d'intégration numérique. Cette méthode est inutilisable vu le temps de calcul qu'elle nécessite. La Combinaison Parallèle de Modèles par Génération de Données (CPMGD) *(Data Driven Parallel Model Combination : DDPMC)* est une technique fondée sur la génération de données (vecteurs *cepstraux*) ; elle a été proposée par le même auteur [25, 24] afin de compenser les coefficients *cepstraux* statiques et dynamiques. Les vecteurs générés à partir des modèles de la parole et du modèle du bruit sont combinés suivant une fonction appelée, par son auteur, "fonction de distorsion" (Mismatch Fonction). Cette fonction ne pose aucun problème pour les paramètres *cepstraux* statiques, en revanche, elle fait intervenir, pour les paramètres *cepstraux* dynamiques à un instant donné, les paramètres *cepstraux* (statiques ou dynamiques) aux instants voisins, or ce contexte temporel ne peut être restitué qu'à partir des données d'apprentissage elles-mêmes ou en le modélisant explicitement.

Dans la suite de cette section, nous proposons une technique permettant de compenser les paramètres cepstraux dynamiques dans le cadre de la combinaison parallèle de modèles. Dans ce cadre, nous décrivons ensuite une méthode permettant de compenser conjointement le bruit additif et convolutif. La compensation des bruits additif et convolutif avec la combinaison parallèle de modèles comporte beaucoup d'approximations, notamment lors de la compensation des paramètres cepstraux dynamiques, ce qui nous incite à développer une nouvelle approche fondée sur la combinaison de données au lieu de la combinaison de modèles. Cette nouvelle approche est décrite dans le section 5.3.

### 5.2.3 Compensation des coefficients *cepstraux* dynamiques

Dans les systèmes de reconnaissance de la parole utilisant des grands vocabulaires, l'utilisation des coefficients *cepstraux* dynamiques est d'une très grande importance [21]. La combinaison parallèle de modèles devient d'autant plus profitable si on arrive à compenser également les coefficients *cepstraux* dynamiques. La compensation des coefficients *cepstraux* dynamiques est plus complexe que la compensation des coefficients *cepstraux* statiques décrite dans la section précédente. En effet, la compensation des coefficients *cepstraux* statiques repose sur la simplicité de la combinaison des sources de la parole propre et du bruit qui est une simple addition dans le domaine spectral ; en revanche, la combinaison des coefficients *cepstraux* dynamiques reste complexe quel que soit le domaine. Cette complexité augmente rapidement avec la complexité des formules utilisées pour les paramètres *cepstraux* dynamiques.

Dans le jargon de la reconnaissance de la parole, les coefficients *cepstraux* dynamiques d'ordre 1 sont appelés *delta-cepstre* et notés $\Delta$*cepstre* et ceux d'ordre 2 sont appelés *delta-delta-cepstre* et notés $\Delta\Delta$*cepstre*.

**Compensation du $\Delta$*cepstre***

Les coefficients $\Delta$*cepstre* représentent l'approximation de la dérivée première par rapport au temps des paramètres *cepstraux* ; ils peuvent être compris comme caractéristiques de la vitesse de l'évolution des paramètres *cepstraux*. Pour les applications de reconnaissance avec un petit vocabulaire et où l'indépendance par rapport au locuteur n'est pas requise, l'utilisation des coefficients $\Delta$*cepstre* n'est pas nécessaire, puisque les mots sont acoustiquement suffisamment distincts les uns des autres. Dans les applications où la taille du vocabulaire est grande et où l'indépendance par rapport au locuteur est requise, l'utilisation des coefficients $\Delta$*cepstre* devient indispensable. Dans la littérature, on trouve plusieurs formules du $\Delta$*cepstre*. La plus simple formule utilisée pour le calcul du $\Delta$*cepstre* est :

$$\Delta O_t^{cp} = -O_{t-2}^{cp} + O_{t+2}^{cp}, \tag{5.29}$$

où $\Delta O_t^{cp}$ désigne le $\Delta$*cepstre* à l'instant $t$ et $O_t^{cp}$ désigne le *cepstre* à l'instant $t$. Le $\Delta$*cepstre* est le résultat de la convolution du *cepstre* avec la fenêtre $[-1, 0, 0, 0, +1]$. Soit $\Delta O_t^{cp}$ le $\Delta$*cepstre* à un instant $t$, celui-ci peut s'écrire en fonction du *cepstre* statique comme suit :

$$\Delta O_t^{cp} = -O_{t-2}^{cp} + O_{t+2}^{cp} \quad (5.30)$$

$$= C \log \left( \frac{O_{t+2}^{sp}}{O_{t-2}^{sp}} \right). \quad (5.31)$$

Notons :

$$\Delta O_t{}^{cp} = C \Delta O_t^{lg} = C \log \left( \Delta O_t^{sp} \right) \quad (5.32)$$

Avec cette notation, on a :

$$\Delta O_t^{sp} = \left( \frac{O_{t+2}^{sp}}{O_{t-2}^{sp}} \right) \quad (5.33)$$

Partant d'une gaussienne dans le domaine *cepstral*, de moyenne $\mu^{\Delta S_{cp}}$ et de matrice de covariance $\Sigma^{\Delta S_{cp}}$ pour la parole propre, et d'une gaussienne dans le domaine *cepstral* de moyenne $\mu^{\Delta N_{cp}}$ et de matrice de covariance $\Sigma^{\Delta S_{cp}}$ pour le bruit, on transforme les deux distributions en appliquant la transformation en Cosinus inverse suivie de l'application du logarithme ; les transformations qui en résultent sont les mêmes que celles établies pour le *cepstre* statique. Une fois dans le domaine spectral, il faut trouver la manière de combiner les paramètres correspondant aux distributions de la parole propre et du bruit pour estimer les paramètres correspondant à la distribution de la parole bruitée dans le domaine. Après une série d'approximations (voir annexe B), on aboutit aux formules approximatives suivantes :

- les composantes de la moyenne de $\Delta R_{sp}$ s'écrivent :

$$\mu_{\Delta R_{sp}}^i \approx \alpha_i \mu_{\Delta S_{sp}}^i + (1 - \alpha_i) \mu_{\Delta N_{sp}}^i, \quad (5.34)$$

- les composantes de la matrice de covariance de $\Delta R_{sp}$ s'écrivent :

$$\sigma_{\Delta R_{sp}}^{ij} \approx \alpha_i \alpha_j \sigma_{\Delta S_{sp}}^{ij} + (1 - \alpha_i)(1 - \alpha_j) \sigma_{\Delta N_{sp}}^{ij} \quad (5.35)$$

avec

$$\alpha_i = \left( \frac{1}{1 + \frac{\mu_{N_{sp}}^i}{\mu_{S_{sp}}^i}} \right) \quad (5.36)$$

**Compensation du $\Delta\Delta$*cepstre***

Les coefficients $\Delta\Delta$*cepstre* représentent l'approximation de la dérivée seconde par rapport au temps des paramètres *cepstraux* ; ils peuvent être compris comme caractéristiques de l'accélération de l'évolution des paramètres *cepstraux* ou la vitesse de l'évolution des paramètres $\Delta$*cepstre*. Le besoin de leur utilisation se ressent dans les très grandes applications de reconnaissance où l'ambiguïté acoustique entre les mots augmente. Dans les formules nous utilisons $\Delta^2$ pour désigner $\Delta\Delta$. La formule définissant le $\Delta^2$*cepstre* que nous allons utiliser dans nos expériences est la suivante :

$$\Delta^2 O_t^{cp} = -\Delta O_{t-1}^{cp} + \Delta O_{t+1}^{cp} \quad (5.37)$$

$$= C \log \left( \frac{\Delta O_{t+1}^{sp}}{\Delta O_{t-1}^{sp}} \right). \quad (5.38)$$

Notons:

$$\Delta^2 {O_t}^{cp} = C\Delta^2 O_t^{lg} = C \log\left(\Delta^2 O_t^{sp}\right) \tag{5.39}$$

Avec cette notation, on a :

$$\Delta^2 O_t^{sp} = \left(\frac{\Delta O_{t+1}^{sp}}{\Delta O_{t-1}^{sp}}\right) \tag{5.40}$$

où $\Delta^2 O_t^{cp}$ représente $\Delta\Delta$*cepstre* à l'instant $t$. Les formules de combinaison correspondant au $\Delta\Delta$*cepstre* dépendent de celles utilisées pour le $\Delta$*cepstre*. Leurs complexités ainsi que les approximations nécessaires à leur établissement augmentent rapidement avec la complexité des formules utilisées pour le $\Delta$*cepstre*. Partant d'une gaussienne dans le domaine *cepstral*, de moyenne $\mu^{\Delta^2 S_{cp}}$ et de matrice de covariance $\Sigma^{\Delta^2 S_{cp}}$ pour la parole propre, et d'une gaussienne dans le domaine *cepstral* de moyenne $\mu^{\Delta^2 N_{cp}}$ et de matrice de covariance $\Sigma^{\Delta^2 S_{cp}}$ pour le bruit, les formules de combinaison du $\Delta\Delta$*cepstre* correspondant au $\Delta$*cepstre* de la formule 5.29, s'expriment dans le domaine spectral comme suit (voir annexe B)

$$\mu^i_{\Delta^2 R_{sp}} \approx \alpha_i \mu^i_{\Delta^2 S_{sp}} + (1-\alpha_i)\mu^i_{\Delta^2 N_{sp}} \tag{5.41}$$

Les composantes de la matrice de covariance de $\Delta^2 R_{sp}$ dans ce cas s'écrivent :

$$\sigma^{ij}_{\Delta^2 R_{sp}} \approx \alpha_i \alpha_j \sigma^{ij}_{\Delta^2 S_{sp}} + (1-\alpha_i)(1-\alpha_j)\sigma^{ij}_{\Delta^2 N_{sp}} \tag{5.42}$$

avec

$$\alpha_i = \frac{1}{1 + \frac{\mu^i_{\Delta N_{sp}} \mu^i_{N_{sp}}}{\mu^i_{\Delta N_{sp}} \mu^i_{N_{sp}}}} \tag{5.43}$$

De la même façon que pour le *cepstre* statique, après avoir effectué la combinaison dans le domaine spectral des distributions des paramètres *cepstraux* dynamiques, le logarithme est appliqué pour revenir au domaine *log-spectral* puis la transformation en Cosinus pour revenir au domaine *cepstral*. Des formules semblables à celles des paramètres statiques sont utilisées. Dans le système de reconnaissance de la parole du LIMSI, les gaussiennes utilisées ont des matrices de covariance diagonales. L'adéquation de tel type de gaussienne se justifie par l'utilisation d'un mélange de gaussiennes au sein de chaque état : une gaussienne avec une matrice de covariance non diagonale peut raisonnablement être approximée par un mélange de gaussiennes avec des matrices de covariance diagonales. Pour les *cepstres* statique et dynamique, quand on revient dans le domaine *cepstral* après combinaison avec le bruit, la matrice de covariance n'est plus diagonale. Pour garder la même structure que celle des modèles d'origine (les modèles de la parole propre), on fait une approximation consistant à mettre à zéro tous les éléments non diagonaux des matrices de covariance résultant de la combinaison avec le bruit.

Gales [25] démontre expérimentalement que la combinaison de deux variables aléatoires gaussiennes, représentant respectivement la parole propre et le bruit, n'est pas une variable aléatoire gaussienne. Il introduit la CPMGD (Combinaison Parallèle de Modèles par Génération de Données) [24], une alternative à la CPM, qui approche mieux la véritable distribution des données bruitées. Cette technique opère au niveau de l'état ; elle consiste à générer un nombre suffisant de trames à partir des densités de probabilités correspondant à un état du modèle de la parole propre et un état du modèle de bruit (on suppose que le modèle du bruit est un MMCDC avec un seul état et un mélange de gaussiennes). Une fois générées, les trames de la parole propre et celles du bruit sont combinées ; on obtient ainsi des trames correspondant à la parole bruitée. En utilisant ces trames, les paramètres de la vraie densité de probabilité de la parole bruitée dans l'état en question, peuvent être estimés en utilisant les procédures d'estimation habituelles (l'algorithme EM par exemple). Ce type de procédure est généralement itératif. Dans ce cas une bonne initialisation serait, par exemple, la densité de probabilité obtenue avec la CPM de base.

Le constat fait dans le paragraphe précédent est en contradiction avec la CPM fondée sur l'approximation *log-normale*, qui suppose que la combinaison de deux gaussiennes (une pour la parole propre et une pour le bruit) est une gaussienne pour la parole bruitée. Or, la seule approximation qu'on fait est l'approximation *log-normale*. Nous avons constaté que cette approximation est valable dans le cas de gaussiennes avec de petites variances ; en revanche, elle perd considérablement de sa validité au fur et mesure que les variances augmentent. Une solution, qui constitue d'ailleurs une alternative à la CPMGD, est d'utiliser des petites variances dans les modèles de la parole propre et dans le modèle de bruit. Pour pouvoir utiliser l'approximation *log-normale*, il faut utiliser un mélange de gaussiennes où chaque élément du mélange (gaussienne) possède une petite variance, au lieu d'utiliser une seule gaussienne avec une grande variance. En représentant le modèle de bruit par une seule gaussienne, nous avons présenté le cas le plus simple de la combinaison. Il est bien sûr possible d'étendre cette présentation au cas d'un bruit dont la distribution est un mélange de gaussiennes. Cependant, le nombre de paramètres augmente rapidement avec la complexité du modèle utilisé pour le bruit. Imaginons le cas relativement simple d'un mélange de deux gaussiennes pour le bruit. Un état contenant 20 gaussiennes avant la combinaison en contiendra 40 après. Il est possible dans ce cas de réduire le nombre de gaussiennes au sein d'un état en utilisant l'algorithme des k-moyennes pour faire une classification des gaussiennes. On remplace ainsi les gaussiennes d'une même classe par la gaussienne représentant la classe à laquelle elles appartiennent. Soit $N(\mu_{ij}, \Sigma_{ij})_{1 \leq j \leq M_i}$ les $M_i$ gaussiennes de la $i^{ème}$ classe. Supposons que le poids associé à la $j^{ème}$ gaussienne de cette classe est $\alpha_{ij}$. Les paramètres de la gaussienne représentant cette classe de gaussiennes sont donnés comme suit :
sa moyenne est

$$\hat{\mu}_i = \frac{\sum_{j=1}^{M_i} \alpha_{ij} \mu_{ij}}{\sum_{j=1}^{M_i} \alpha_{ij}} \tag{5.44}$$

et sa matrice de covariance est

$$\hat{\Sigma}_i = \frac{\sum_{j=1}^{M_i} \alpha_{ij} \left( \Sigma_{ij} + \mu_{ij} \mu_{ij}^t \right)}{\sum_{j=1}^{M_i} \alpha_{ij}} - \hat{\mu}_i \hat{\mu}_i^t \tag{5.45}$$

Le poids de cette gaussienne est

$$\alpha_i = \sum_{j=1}^{M_i} \alpha_{ij} \tag{5.46}$$

Pour faire cette classification fondée sur l'algorithme des k-moyennes, il faut trouver une distance entre deux gaussiennes. Une distance possible est la divergence. Soient deux densités de probabilité $f$ et $g$. La divergence $J(f,g)$ mesure une distance de dissimilarité entre les deux classes de données dont les distributions de probabilité sont respectivement $f$ et $g$ [79] :

$$J(f,g) = \int_x [f(x) - g(x)] \log\left(\frac{f(x)}{g(x)}\right) \tag{5.47}$$

Dans le cas de deux densités de probabilités gaussiennes $N_1(\mu_1, \Sigma_1)$ et $N_2(\mu_2, \Sigma_2)$ la divergence est donnée par :

$$\begin{aligned} J(N_1, N_2) &= \frac{1}{2} tr[(\Sigma_1 - \Sigma_2)(\Sigma_2^{-1} - \Sigma_2^{-1})] \\ &+ \frac{1}{2} tr[(\Sigma_1 - \Sigma_2)(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t] \end{aligned} \tag{5.48}$$

où $tr(A)$ désigne la trace[7] de la matrice $A$.

La figure 5.3 montre un exemple de réduction de nombre de gaussiennes au sein d'un état donné. La combinaison de quatre gaussiennes avec deux gaussiennes résulte en huit gaussiennes, la stratégie décrite ci-dessus permet de les réduire à cinq gaussiennes.

### 5.2.4 Compensation du bruit convolutif

Nous avons supposé, jusqu'à présent, que les données de test et les données d'apprentissage sont enregistrées avec le même microphone. En réalité, on est souvent amené à enregistrer les données en phase d'utilisation avec des microphones différents de celui utilisé lors de l'apprentissage. Les raisons de cela peuvent être d'ordre pratique ou économique. L'utilisation en phase de test de microphones différents de celui utilisé en phase d'apprentissage introduit un nouveau type de bruit, dont l'effet est convolutif dans le domaine temporel. Si on veut prendre en compte ce bruit convolutif, les signaux d'apprentissage et de test peuvent être raisonnablement modélisés par le modèle $M_1$ comme suit[8] :

$$\begin{array}{ll} y_1(t) = s(t) & \text{pour l'apprentissage} \\ y_2(t) = h(t) * (s(t) + n(t)) & \text{pour le test} \end{array} \tag{5.49}$$

[7] La trace d'une matrice est la somme des éléments diagonaux de la matrice.

[8] Remarquer que le modèle du signal acoustique bruité est $y(t) = h(t) * (s(t) + n(t))$ qui est différent de celui utilisé habituellement : $y(t) = h(t) * s(t) + n(t)$. La différence vient du fait que nous considérons que le bruit additif est $n(t)$ et non pas $h(t) * n(t)$ qui est le bruit additif convolué avec la réponse impulsionnelle du canal de transmission.

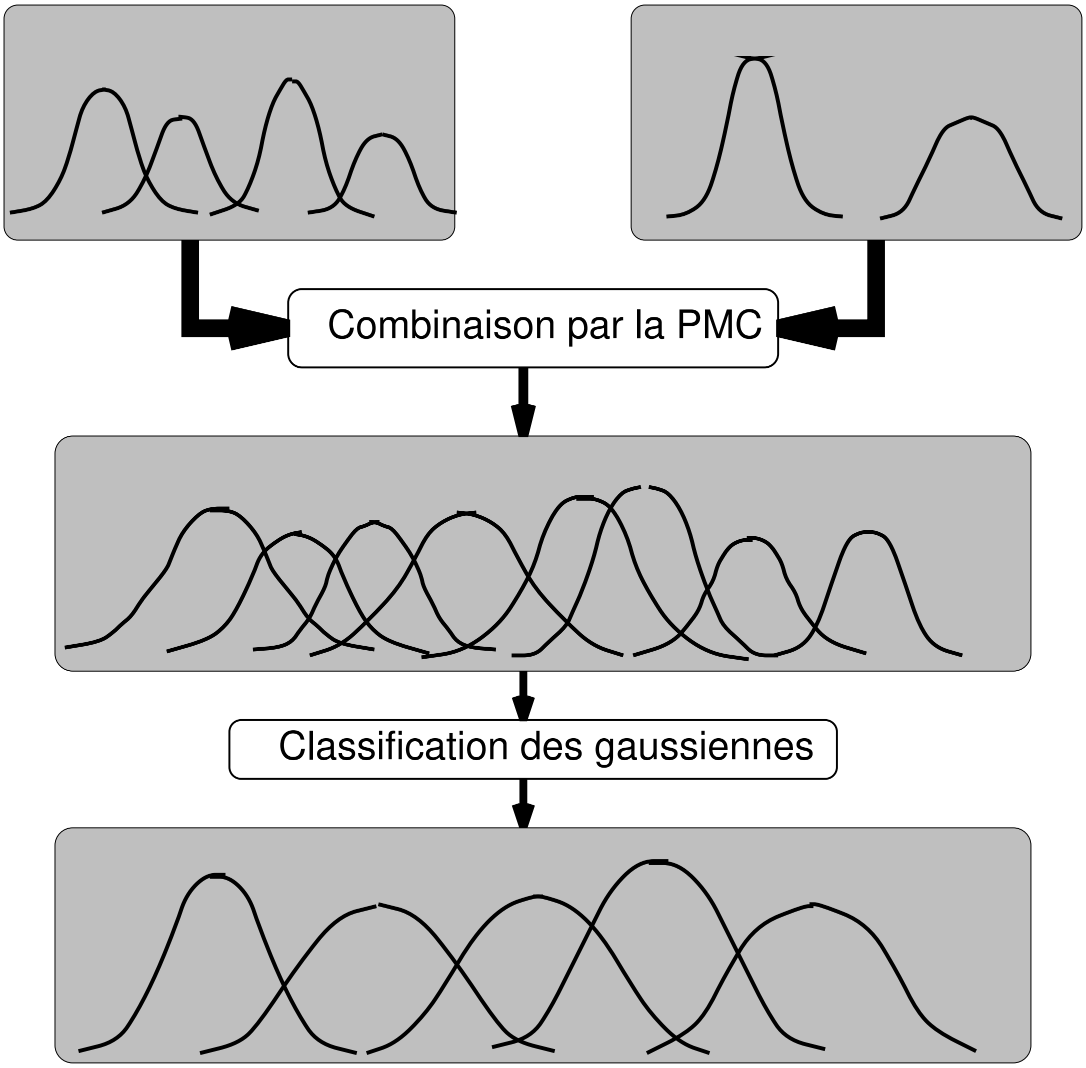



FIG. 5.3 - *La combinaison de modèles dans le cas d'un modèle de bruit multi-gaussien. Le modèle de la parole propre est constitué de 4 gaussiennes, celui du bruit de 2 gaussiennes. La première étape est la combinaison de toutes les paires de gaussiennes parole/bruit, elle résulte en un modèle pour la parole bruitée avec 8 gaussiennes. La deuxième étape est la classification des gaussiennes avec les k-moyennes résultant en un modèle avec 5 gaussiennes pour la parole bruitée.*

où $s_t$ désigne le signal propre (non bruité), $n(t)$ et $h(t)$ désignent respectivement les bruits additif et convolutif affectant les données de test. Le bruit convolutif dans le domaine temporel devient additif dans le domaine *log-spectral* ou *cepstral*, car la transformée de Fourier transforme un produit de convolution en un produit simple ; ensuite, le logarithme transforme ce produit simple en une addition. Notons $h^{cp}$ et $h^{lg}$ les composantes relatives au bruit convolutif dans le domaine *cepstral* et *log-spectral* respectivement. La compensation de ce bruit convolutif est constituée de deux étapes. La première est son estimation suivant un critère donné, la deuxième est sa compensation proprement dite. Le critère du maximum de vraisemblance pour estimer ce bruit est :

$$\hat{h}^{cp} = \arg\max_{h^{cp}} [p(Y_T^{cp}|h^{cp}, \lambda_s, \lambda_{n,h}, L)] \tag{5.50}$$

où $Y_T^{cp}$ désigne les données de test dont nous disposons pour faire cette estimation. $\lambda_s$ désigne les paramètres des modèles acoustiques relatifs à la parole propre ; $\lambda_{n,h}$ désigne les paramètres du modèle de bruit ; $L$ désigne le modèle de langage. Pour réaliser cette maximisation, il est possible d'utiliser l'algorithme EM. Cependant, le temps de calcul requis pour cela est très important. En effet, partant d'une valeur initiale de $h^{cp}$, à chaque itération de l'algorithme EM, il faut faire un décodage permettant d'aligner la séquence $Y_T^{cp}$ avec les gaussiennes représentant la parole bruitée à cette itération. Cet alignement permet de réestimer $h^{cp}$ et par là-même, de réadapter les modèles afin qu'ils représentent au mieux la parole bruitée. Les modèles de la parole propre sont compensés à chaque itération en utilisant un modèle pour le bruit qui est lui-même réadapté à chaque itération de l'algorithme EM. Cette réadaptation est nécessaire car c'est le modèle de bruit $\lambda_n$ qu'il faut utiliser dans la combinaison et non pas $\lambda_{n,h}$.

Si on considère que le bruit convolutif est relativement constant et indépendant du contenu linguistique de la séquence $Y_T^{cp}$, alors il est possible d'utiliser un modèle beaucoup plus simple au lieu d'utiliser le système de reconnaissance dans son intégralité. Pour représenter l'effet du bruit convolutif, seule la structure générale de la parole est nécessaire. Une seule gaussienne pour la parole propre et une seule gaussienne pour le bruit suffisent pour estimer $h^{cp}$ :

$$\hat{h}^{cp} = \arg\max_{h^{cp}} [p(Y_T^{cp}|h^{cp}, \mu_{S^{cp}}, \Sigma_{S^{cp}}, \mu_{N^{cp},h^{cp}}, \sigma_{N^{cp},h^{cp}})] \tag{5.51}$$

$h^{cp}$ peut être estimé itérativement partant d'une estimation initiale $h_0^{cp}$ . A chaque itération, on combine, en utilisant l'approximation *log-normale*, une estimation de la gaussienne du bruit $N(\mu_{N_{cp}}, \sigma_{N_{cp}})$ avec la gaussienne de la parole propre, ce qui permet d'avoir une estimation de la moyenne de la parole bruitée $\hat{\mu}_{R^{cp}}$. Notons $\mu_{R^{cp},h^{cp}}$ la moyenne dans le domaine *cepstral* des trames de la séquence de test $Y_T^{cp}$. A chaque itération $h^{cp}$ est estimée comme suit :

$$\hat{h}^{cp} = \mu_{R^{cp},h^{cp}} - \mu_{R^{cp}} \tag{5.52}$$

car dans le domaine *cepstral* on a :

$$\mu_{R^{cp},h^{cp}} = \mu_{R^{cp}} + h^{cp} \tag{5.53}$$

A cause de l'approximation *log-normale*, cette technique ne peut être appliquée telle qu'elle est. En effet, la variance de la gaussienne correspondant à la parole est trop grande pour pouvoir utiliser l'approximation *log-normale*. Pour cela, on utilise un mélange de gaussiennes pour modéliser la parole propre et un mélange de gaussiennes pour modéliser le bruit (voir Figure 5.4). Les deux mélanges de gaussiennes sont estimés en utilisant l'algorithme EM [15]. La densité de probabilité correspondant à la parole propre s'écrit ainsi :

$$f(x) = \sum_{i=1}^{M_s} \alpha_i N(x|\mu_i^{S^{cp}}, \Sigma_i^{S^{cp}}) \tag{5.54}$$

la densité de probabilité correspondant au bruit :

$$g(x) = \sum_{i=1}^{M_n} \beta_i N(x|\mu_i^{N^{cp},h^{cp}}, \Sigma_i^{N^{cp},h^{cp}}) \tag{5.55}$$

où $N(x|\mu_i^{S^{cp}}, \Sigma_i^{S^{cp}})$ (respectivement $N(x|\mu_i^{N^{cp}}, \Sigma_i^{N^{cp}})$) désigne la $i^{ème}$ gaussienne dans le mélange de gaussiennes correspondant à la parole propre (respectivement au bruit). Les poids des gaussiennes $\alpha_{i,(1\leq i\leq M_s)}$ et $\beta_{i,(1\leq i\leq M_n)}$ vérifient :

$$\textstyle\sum_{i=1}^{M_s} \alpha_i = 1, \qquad \sum_{i=1}^{M_n} \beta_i = 1$$

Le principe d'estimation de la composante convolutive dans le cas d'un mélange de gaussiennes reste le même que dans le cas de modèles mono-gaussiens, la seule différence réside dans le calcul d'une estimation de la moyenne de la parole bruitée $\hat{\mu}_{R^{cp}}$. Pour calculer une estimation de $\mu_{R^{cp}}$ à une itération donnée, on combine toutes les paires de gaussiennes parole/bruit (les gaussiennes du bruit sont celles correspondant à l'itération en cours). On obtient ainsi, un mélange de gaussiennes pour la parole bruitée avec $M_s \times M_n$ gaussiennes. Le poids de la gaussienne résultant de la combinaison de la $i^{ème}$ gaussienne du modèle de la parole propre avec la $j^{ème}$ gaussienne du modèle de bruit est $\alpha_i \times \beta_j$. La moyenne de la gaussienne représentant la parole bruitée $\hat{\mu}_{R^{cp}}$ est estimée comme suit:

$$\hat{\mu}_{R^{cp}} = \sum_{i=1}^{M_s} \sum_{j=1}^{M_n} \alpha_i \beta_j \mu_{ij}^{R^{cp}} \tag{5.56}$$

et la matrice de covariance $\hat{\Sigma}_{R^{cp}}$ :

$$\hat{\Sigma}_{R^{cp}} = \sum_{i=1}^{M_s} \sum_{j=1}^{M_n} \alpha_i \beta_j \left( \Sigma_{ij}^{R^{cp}} + \mu_{ij}^{R^{cp}} \mu_{ij}^{R^{cp}\,t} \right) - \hat{\mu}_{R^{cp}} \hat{\mu}_{R^{cp}}^t \tag{5.57}$$

où $\mu_{ij}^{R^{cp}}$ et $\Sigma_{ij}^{R^{cp}}$ sont respectivement la moyenne et la matrice de covariance de la gaussienne résultant de la combinaison de la $i^{ème}$ gaussienne du modèle de la parole propre avec la $j^{ème}$ gaussienne du modèle de bruit.

A une itération donnée, étant donnée une estimation de $h_{cp}$, il est possible d'estimer le modèle de bruit de deux manières. La première opère directement sur les

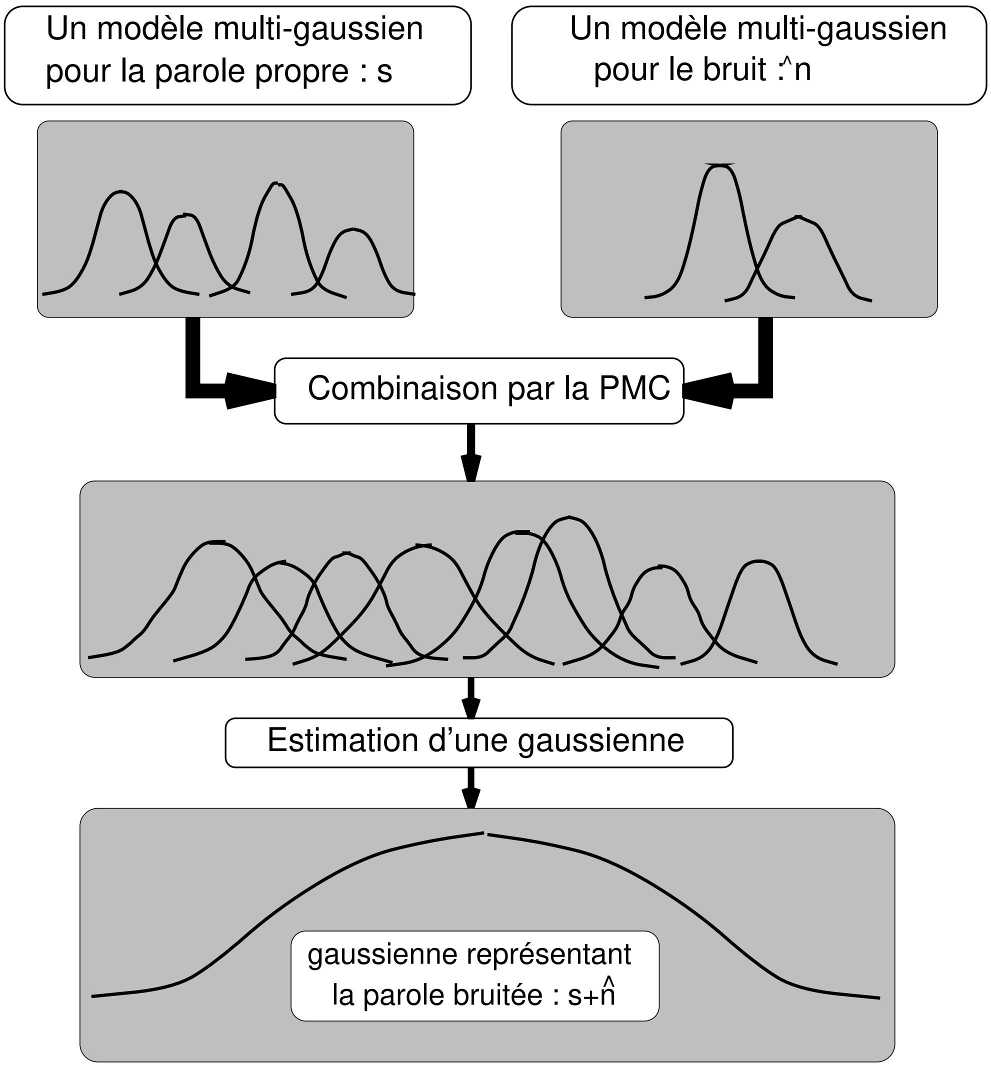



FIG. 5.4 - *Estimation de la gaussienne représentant la parole bruitée. Le modèle de la parole propre est constitué de 4 gaussiennes, celui du bruit de 2 gaussiennes. La première étape est la combinaison de toutes les paires de gaussiennes parole/bruit, elle résulte en un modèle pour la parole bruitée avec 8 gaussiennes. La deuxième étape est l'estimation de la gaussienne représentant la parole bruitée.*

données qui ont permis l'estimation du modèle de bruit. Elle consiste à soustraire $\hat{h}_{cp}$ à toutes les trames correspondant au bruit, puis à réestimer le modèle de bruit avec ces nouvelles trames. Cette soustraction, dans le domaine *cepstral*, représente un filtrage dont l'effet est l'inverse de celui causé par le microphone de test. Cette façon de faire est envisageable grâce à la taille réduite des modèles de bruit généralement utilisés. La deuxième manière consiste à adapter le modèle de bruit d'origine, c'est-à-dire celui appris sur $n * h$. On suppose que le filtrage n'altère pas la répartition des trames entre les gaussiennes. Dans ce cas, les poids et les variances des gaussiennes ne changent pas, les moyennes du modèle de bruit correspondant à $n$ sont réestimées comme suit :

$$\hat{\mu}_i^{N^{cp}} = \mu_i^{N^{cp},h^{cp}} - \hat{h}^{cp} \tag{5.58}$$

Avec la dernière estimation de $h^{cp}$, nous réestimons le modèle de bruit correspondant à $n$ comme décrit dans le paragraphe précédent. Nous combinons le modèle obtenu avec les modèles des unités acoustiques de la parole propre. Nous obtenons ainsi des modèles acoustiques pour la parole bruitée $y_1 = s + \hat{n}$. Le signal de test dont nous disposons est $y_2 = (s+n) * h$. Pour annuler l'effet de $h$ sur les données de test, on les filtre en utilisant l'inverse du filtre estimé. Ainsi, les données que nous utilisons pour le test deviennent $y_2^{cp} - \hat{h}^{cp}$, c'est-à-dire le signal de test observé filtré avec $\hat{h}^{-1}$.

## 5.3 Combinaison Parallèle de Données (CPD)

Dans le cadre de la CPM, l'approximation *log-normale* [23] est souvent utilisée pour passer du domaine spectral au domaine *cepstral* après combinaison. Elle suppose que la somme de deux variables aléatoires *log-normalement* distribuées est elle-même *log-normalement* distribuée. Cette approximation devient inacceptable dans le cas de grandes variances. Nous avons proposé, dans la section précédente, une solution consistant à répartir l'espace acoustique en plusieurs sous-espaces de petites variances. Cette solution conduit à un changement de la structure des modèles d'origine.

Comme nous l'avons constaté dans l'annexe B, la difficulté des calculs et le nombre d'approximations dans le cadre de la CPM augmentent rapidement avec la complexité des formules utilisées pour estimer les paramètres *cepstraux* dynamiques correspondant à la parole bruitée, ce qui représente un inconvénient majeur de l'approche. Pour chaque type de paramètres *cepstraux* dynamiques, il faut trouver des équations d'adaptation spécifiques aux formules les définissant. La qualité des approximations utilisées pour aboutir à ces équations d'adaptation, se mesure en terme de taux de reconnaissance et reste difficilement mesurable *a priori*. En effet, leur efficacité dépend des modèles utilisés pour la parole propre et du bruit qui fait l'objet de l'adaptation.

La technique utilisant la génération de données (CPMGD) a été introduite par Gales et Young pour surmonter les problèmes liés à la CPM [24]. Elle consiste à générer d'une manière aléatoire des vecteurs associés à chaque état. Ces vecteurs sont ensuite combinés avec les trames générées de façon aléatoire à partir du modèle de bruit. Le résultat de la combinaison est utilisé pour réestimer la nouvelle densité de probabilité, associée à l'état en question, en utilisant l'algorithme EM. Avec cette technique, seules les densités de probabilité relatives aux paramètres statiques sont correctement estimées. Cette technique est nécessairement coûteuse en temps de calcul si on veut générer un nombre suffisant de vecteurs, pour avoir une validité statistique. Pour résoudre les problèmes liés à la CPM et à la CPMGD, nous avons développé et nous présentons une nouvelle technique [56] utilisant la combinaison directement sur les données.

### 5.3.1 Modèle utilisé

Supposons que les données d'apprentissage sont propres et que les données de test sont bruitées. Supposons en outre, que le canal d'enregistrement peut être représenté par un filtre linéaire. Dans un premier temps, nous faisons l'hypothèse que les données de test et d'apprentissage sont enregistrées avec le même microphone. Dans ce cas, les signaux d'apprentissage et de test peuvent être raisonnablement modélisés par le modèle $M_2$ comme suit :

$$\begin{array}{ll} y_1(t) = s(t) & \text{pour l'apprentissage} \\ y_2(t) = s(t) + n(t) & \text{pour le test} \end{array}$$

où $y_1$ et $y_2$ désignent respectivement les signaux d'apprentissage et de test observés, $s$ désigne le signal propre (non bruité), et $n$ désigne le bruit additif dans les données de test.

Notre technique opère en mode non supervisé, en utilisant seulement les données de test. Du fait de la différence entre les conditions de test et d'apprentissage, les modèles dont on dispose ne représentent plus vraiment des statistiques correspondant aux données de test. Pour obtenir des modèles représentant le signal de test, il faut adapter les modèles appris sur $y_1$ en utilisant le signal de test $y_2$. Pour cela, on utilise une technique fondée sur la combinaison parallèle de données, inspirée de la technique de la combinaison parallèle de modèles [55, 24].

### 5.3.2 Principe de la Combinaison Parallèle de Données (CPD)

La Combinaison Parallèle de Données (CPD) pour l'adaptation consiste à stocker pour chaque gaussienne d'un état donné, et d'un modèle donné, les vecteurs *cepstraux* du corpus d'apprentissage qui ont permis son estimation. La combinaison consiste dans ce cas à faire la somme trame à trame dans le domaine spectral, des trames stockées pour la gaussienne en question et des trames représentant le bruit. Les trames représentant le bruit proviennent généralement d'une analyse à court terme du signal recueilli dans les périodes de silence au moment du test. On applique les opérations nécessaires pour revenir au domaine *cepstral*. On obtient ainsi les vecteurs *cepstraux* qui permettent de réestimer la moyenne et la matrice de covariance de la gaussienne en question. Comme pour la CPM, on suppose que le bruit n'affecte pas la distribution des trames entre les gaussiennes d'un état donné. Il est clair qu'il s'agit d'une supposition dont la validité dépend du type de bruit et de son niveau. Pour surmonter ce problème, il est possible d'adopter le même raisonnement au niveau de l'état et non au niveau de la gaussienne. Cela implique une reclassification des trames (en utilisant l'algorithme EM) dans chaque état. Il est également possible d'utiliser l'ancienne classification comme initialisation de l'algorithme EM. Ainsi seul le calcul nécessaire pour ajuster les modèles est effectué. En procédant de cette manière, la combinaison est réalisée sans aucune approximation.

Pour une trame donnée du corpus d'apprentissage appartenant[9] à un état donné, l'appartenance à une gaussienne dans cet état est déterminée en choisissant celle qui donne la plus grande probabilité *a posteriori*. Cette association entre les trames d'apprentissage et les gaussiennes des modèles, est déterminée avant d'entamer le processus de compensation. Il est éventuellement possible de prendre en compte la participation d'une trame à l'estimation des paramètres d'un ensemble de gaussiennes et non pas exclusivement à une gaussienne. Cela constitue d'ailleurs la différence principale entre la classification des données en utilisant l'algorithme EM et celui des k-moyennes. Pour faire cette association entre une trame du corpus d'apprentissage, appartenant à un état donné, et les gaussiennes pour lesquelles

[9] Dans ce contexte, le terme “appartenir” à un état (ou à une gaussienne) veut dire que la trame en question a participé à l'estimation de la densité de probabilité associée à cet état, ou que la trame en question est générée par la densité de probabilité associée à cet état.

elle a significativement participé à l'estimation des paramètres, nous choisissons les $N$ gaussiennes qui ont les $N$ plus grandes probabilités *a posteriori*. Ce nombre $N$ doit être choisi en faisant un compromis entre le coût en terme de temps de calcul nécessaire et le gain apporté. Plus $N$ est grand plus les modèles obtenus après compensation sont corrects, mais plus le temps de calcul requis pour la compensation est important. Il est évident que si on utilise un système de reconnaissance possédant un nombre important de gaussiennes par état, alors le nombre $N$ de gaussiennes à prendre en compte pour une trame donnée est petit. Le nombre $N$ optimal au sens du taux d'erreur sur les mots peut être obtenu en utilisant les données de développement[10]. Dans nos expériences, nous considérons qu'une trame du corpus d'apprentissage appartenant à un état donné est générée exclusivement par une seule gaussienne de cet état. Nous avons constaté qu'avec 20 gaussiennes par état, l'effet de cette approximation est négligeable.

Pour rendre le processus de compensation plus rapide il est bien sûr possible de stocker les densités spectrales relatives aux données d'apprentissage au lieu de stocker les paramètres *cepstraux* correspondants. De cette manière, on peut éviter la moitié des calculs.

Une compensation de type CPM, c'est-à-dire une compensation dont la combinaison est réalisée sur les modèles, est fortement dépendante de l'analyse effectuée sur le signal pour extraire l'information spectrale utile pour la reconnaissance. La complexité d'une telle approche augmente rapidement avec celle des opérateurs utilisés pour extraire cette information spectrale. Par exemple, les formules d'adaptation utilisées dans ce chapitre sont intrinsèques à une analyse de type MFCC ou LPCC. Elles ne peuvent pas, par exemple, être utilisées dans le cadre d'une analyse de type PLPCC (Perceptually-based Linear Prediction Cepstrum Coefficients). En revanche, l'approche fondée sur la combinaison parallèle de données peut être appliquée à n'importe quel type de paramétrisation du signal, pour peu que les opérateurs utilisés pour extraire ces paramètres soient inversibles.

### 5.3.3 Compensation des paramètres cepstraux dynamiques

La combinaison des données d'apprentissage $y_1$ avec l'estimation de $n$ est réalisée phrase par phrase et trame par trame, ce qui permet de calculer les dérivées première et seconde du *cepstre* de la manière habituelle. En effet, puisque la combinaison des trames d'apprentissage avec les trames représentant le bruit est réalisée phrase par phrase, le contexte temporel d'une trame donnée $x_t$ (paramètre statique) reste présent après la combinaison et peut donc être utilisé pour calculer les paramètres *cepstraux* dynamiques avec les formules désirées. Ce sont les paramètres *cepstraux* adaptés au bruit de test $(\ldots, x_{t-2}', x_{t-1}', x_{t+1}', x_{t+2}', \ldots)$ que nous utilisons pour le

[10] Pour réaliser une application de reconnaissance de la parole, les données sont généralement réparties en trois types : les données d'apprentissage, les données de développement et enfin les données de test. Les données d'apprentissage servent à estimer les paramètres des modèles acoustiques, les données de développement servent à déterminer expérimentalement les valeurs optimales de certains paramètres du système (elles sont utilisées plusieurs fois) et enfin les données de test qui sont utilisées une seule fois pour évaluer les performances du système.

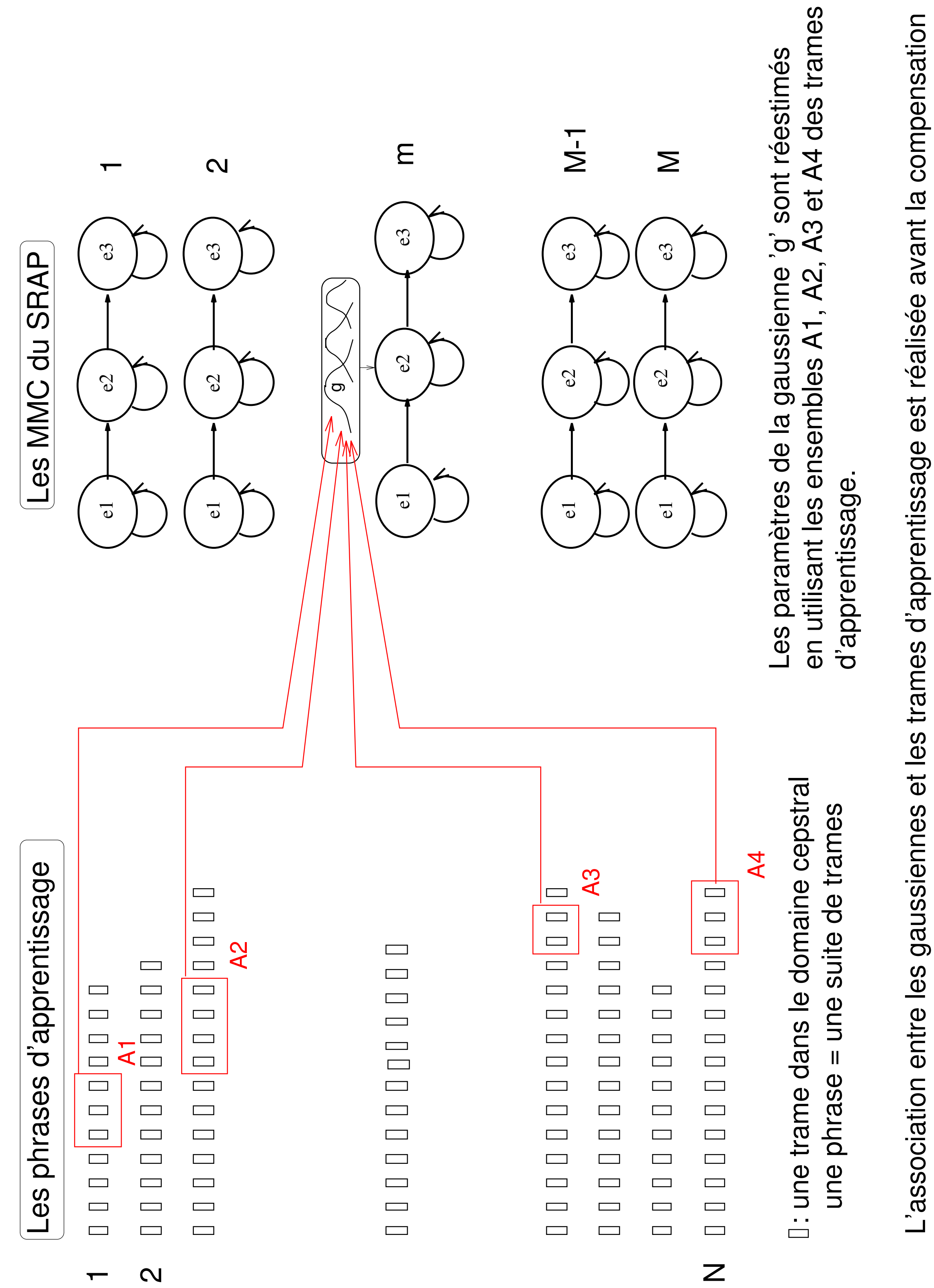



FIG. 5.5 - Cette figure illustre l'association entre les trames d'apprentissage et le gaussiennes des MMCDC. Après combinaison avec le bruit, les trames associées à une gaussienne permettent de réestimer ses paramètres (moyenne et matrice de covariance).

calcul des paramètres *cepstraux* dynamiques à l'instant $t$. Les paramètres *cepstraux* dynamiques sont ainsi calculés comme si l'apprentissage et le test se déroulaient dans les mêmes conditions de bruit, sans aucune approximation et quelle que soit la complexité des formules qui les définissent. Cette caractéristique constitue l'un des plus grands avantages de cette approche de compensation.

Quand toutes les phrases du corpus d'apprentissage ont été traitées, les vecteurs *cepstraux* (incluant les paramètres statiques et dynamiques) adaptés qui sont associés à une gaussienne sont utilisés pour réestimer ses paramètres (sa moyenne et sa matrice de covariance) (voir Figure 5.5). En revanche, les poids des gaussiennes au sein d'un état ne changent pas. Ceci est une approximation qui suppose que la répartition des trames bruitées entre les gaussiennes d'un état donné est la même que celle des trames propres[11].

### 5.3.4 Validité statistique de la CPD

Comme dit précédemment, la combinaison des trames d'apprentissage avec les trames représentant le bruit se fait phrase par phrase et d'une manière circulaire par rapport aux trames du bruit (voir Figure 5.6). Pour mettre en évidence la validité statistique de notre approche, nous allons nous situer dans un cas très simple. Soit $N(\mu, \Sigma)$ une gaussienne d'un état donné des modèles acoustiques représentant la parole propre. Supposons que $M$ trames parmi les données d'apprentissage participent à l'estimation des paramètres de la gaussienne $N(\mu, \Sigma)$ ; si on suppose que le bruit peut être représenté par une seule gaussienne de moyenne $\mu_n$ dans le domaine *cepstral*, notre stratégie de combinaison permet de combiner ces trames relatives à $N(\mu, \Sigma)$ avec des trames de bruit plus ou moins proches de la moyenne $\mu_n$. Ce qui permet de prendre en compte la variance du bruit dans les modèles adaptés.

Analysons le cas d'un bruit ayant plusieurs composantes principales. Supposons, par exemple que le bruit possède deux composantes principales, c'est-à-dire que le bruit peut être représenté par un mélange de deux gaussiennes dans le domaine *cepstral*. Soient $\alpha_1$ et $\alpha_2$ les poids respectifs de ces deux gaussiennes. Supposons que $M$ trames parmi les données d'apprentissage participent à l'estimation des paramètres de la gaussienne $N(\mu, \Sigma)$. Notre stratégie de compensation converge vers la situation suivante : parmi les $M$ trames, $\alpha_1 M$ trames vont être affectées par la première composante du bruit et $\alpha_2 M$ vont être affectées par la seconde. Il est donc clair, que pour tenir compte de la multi-modalité du bruit, le résultat de la combinaison doit être un mélange de deux gaussiennes au lieu d'une seule. Dans notre approche, nous approximons ce mélange par une seule gaussienne dont les paramètres sont estimés en utilisant la totalité des données adaptées (les $M$ trames). Pour corriger cette approximation, on peut faire appel une nouvelle fois à l'algorithme EM au niveau de l'état, en utilisant toutes les trames adaptées appartenant à cet état. Cela permet d'avoir une classification optimale, au sens du maximum de vraisemblance, des trames adaptées dans l'état en question. La meilleure initialisation de l'algorithme

[11] Le critère du maximum de vraisemblance n'est pas respecté : les paramètres des modèles acoustiques (après combinaison) ne maximisent pas la vraisemblance des données d'apprentissage bruitées.

EM dans ce cas est le *code-book*[12] obtenu avec la combinaison parallèle de données de base (décrite dans la section 5.3).

### 5.3.5 Compensation du bruit additif et convolutif

Dans les sections précédentes nous avons considéré que les données de test et les données d'apprentissage sont enregistrées avec le même microphone. Dans ce qui suit nous allons traiter le cas où les deux types de données sont enregistrées avec des microphones différents. Supposons que les données d'apprentissage sont propres et que celles du test sont bruitées. Supposons en outre, que le canal d'enregistrement peut être représenté par un filtre linéaire. Cette différence dans le canal d'enregistrement peut être considérée comme un bruit convolutif. Dans ce cas, les signaux d'apprentissage et de test peuvent raisonnablement être modélisés par le modèle $M_3$ comme suit :

$$\begin{array}{ll} y_1(t) = s(t) & \text{pour l'apprentissage} \\ y_2(t) = h(t) * (s(t) + n(t)) & \text{pour le test} \end{array}$$

où $y_1$ et $y_2$ désignent respectivement les signaux d'apprentissage et de test observés, $s$ désigne le signal propre (non bruité), $n$ et $h$ désignent respectivement les bruits additif et convolutif dans les données de test.

Pour le bruit présent en phase de test, nous disposons seulement des trames représentant $h * n$, mais pour faire la combinaison avec $y_1$ nous avons besoin d'une estimation de $n$. Pour cela, on estime le filtre $h^{-1}$ qu'on applique à $h*n$ pour obtenir une estimation de $n$. L'estimation du filtre $h^{-1}$ est obtenue itérativement en utilisant l'algorithme suivant fondé sur le modèle du canal de transmission $M_3$.

**Algorithme 1** :
La combinaison est faite dans le domaine spectral
Tous les autres calculs sont dans le domaine *cepstral*
**1-** $\tilde{n} := h * n$, $1^{ere}$ estimation de $n$
**2-** calculer $M_{test}$, la moyenne de $y_2 = h * (s + n)$
**3-** combiner $y_1$ avec $\tilde{n}$ pour obtenir une estimation de $(s + n)$ et calculer sa moyenne *cepstrale* $M_{appr}$
**4-** estimer $h^{-1}$ dans le domaine *cepstral* comme :
$$\tilde{H} := M_{appr} - M_{test}$$
**5-** utiliser le filtre $\tilde{H}$ pour obtenir une nouvelle estimation de $n$ :
$$\tilde{n} := \tilde{H} * h * n$$
**6** Reprendre en 3 jusqu'a la convergence

[12] *Code-book* (Dictionnaire de code), dans notre cas c'est un mélange (somme) de gaussiennes pondérées. Le poids d'une gaussienne mesure son importance dans le mélange (c'est-à-dire sa participation dans la génération des données). L'estimation d'un tel *code-book* à partir de données se fait en utilisant l'algorithme EM [70].

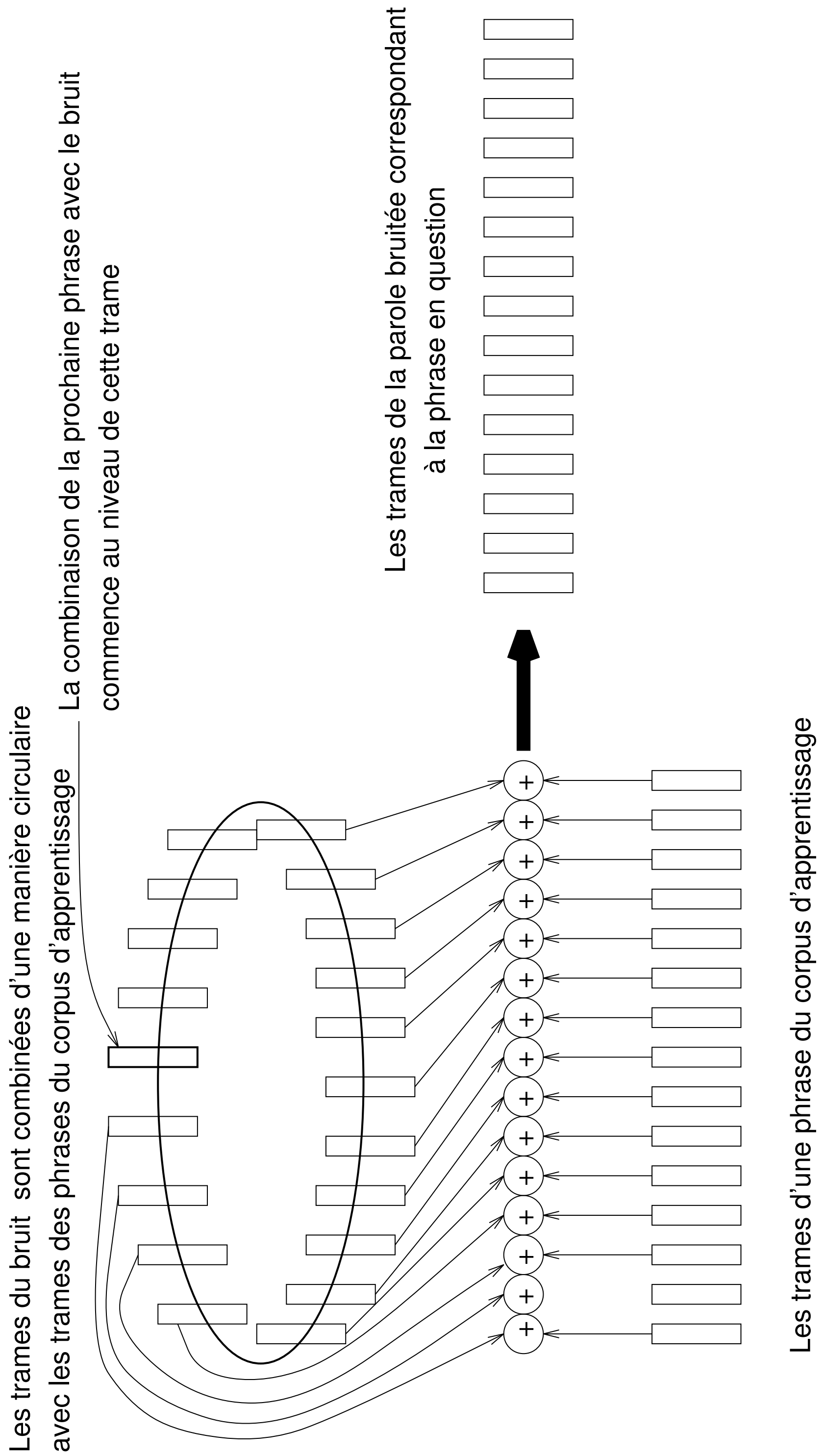



FIG. 5.6 - Les trames de bruit dont on dispose sont combinées d'une manière circulaire avec les trames correspondant aux phrases d'apprentissage les unes à la suite des autres.

Dans la pratique, cinq itérations suffisent pour estimer correctement $n$. Une fois estimé, $n$ est combiné avec $y_1$ pour obtenir les données d'apprentissage adaptées $s+\tilde{n}$. La normalisation par rapport à la composante convolutive $h$ est réalisée par soustraction du *cepstre* moyen des nouvelles données d'apprentissage et des données de test, le *cepstre* moyen utilisé est calculé sur la phrase. Avec les nouvelles données d'apprentissage, on estime de la même manière que précédemment les paramètres de chaque gaussienne dans le système. Dans cet algorithme itératif, puisqu'on s'intéresse principalement aux moyennes, seule une petite partie du corpus d'apprentissage peut être utilisée. Dans nos expériences, sur 450 locuteurs utilisés pour l'apprentissage, seulement 35 locuteurs ont été utilisés pour l'estimation de $n$ et $h$.

L'estimation de $n$ et $h$ peut se faire suivant l'algorithme 1 en utilisant l'approximation *log-normale* (voir section 5.2.4 ). Pour cela, il faut disposer d'un *code-book* représentant la parole propre (les données d'apprentissage). Ce *code-book* doit être assez fin pour que l'approximation *log-normale* reste valable. Il faut remarquer que l'estimation de $n$ et de $h$ est indépendante du processus de compensation, c'est-à-dire qu'on peut utiliser la technique qui convient, suivant le contexte, pour estimer $n$ et $h$ puis procéder à la compensation.

Pour montrer que l'algorithme 1 converge, il suffit de montrer, partant d'une estimation $n_0$ et $h_0$ de $n$ et de $h$, qu'à chaque itération les nouvelles estimations permettent d'augmenter la varaisemblance des données utilisées pour cette estimation. Soit $y_{cp} = y^t_{cp}$, $(t = 1, \ldots, T)$ une suite de trames représentant les données de test :

$$y^t_{cp} = (s^t_{cp} \oplus n^t_{cp}) + h_{cp}$$

où $s^t_{cp}$ désigne la trame à l'instant $t$ du signal propre, $n^t_{cp}$ désigne la trame à l'instant $t$ du bruit et $\oplus$ désigne l'opération de combinaison. Théoriquement $n$ et $h$ doivent être estimés en utilisant le critère du maximum de vraisemblance comme suit :

$$(\hat{h}^{cp}, \hat{\lambda}_{n_{cp}}) = \arg\max_{(h_{cp},\lambda_{n_{cp}})} \left[p(y_{cp}|h^{cp}, \lambda_{s_{cp}}, \lambda_{n_{cp}}, L)\right] \tag{5.59}$$

où $\lambda_{s_{cp}}$ désigne les paramètres, dans le domaine cepstral, des modèles acoustiques relatifs à la parole propre (MMCDC), $\lambda_{n_{cp}}$ désigne les paramètres, dans le domaine cepstral, du modèle de bruit et $L$ désigne le modèle de langage. Comme nous l'avons expliqué précédamment, cette maximisation est très coûteuse en temps de calcul, il convient donc d'utiliser des modèles beaucoup plus simples. Nous utilisons une gaussienne $N(s_{cp}|\mu_{s_{cp}}, \Sigma_{s_{cp}})$ pour modéliser la parole propre et une gaussienne $N(n_{cp}|\mu_{n_{cp}}, \Sigma_{n_{cp}})$ pour modéliser le bruit ; la maximisation de l'équation 5.59 devient :

$$(\hat{h}^{cp}, \hat{\mu}_{n_{cp}}, \hat{\Sigma}_{n_{cp}}) = \arg\max_{(h_{cp},\mu_{n_{cp}},\Sigma_{n_{cp}})} \left[p(y_{cp}|h^{cp}, N(s_{cp}|\mu_{s_{cp}}, \Sigma_{s_{cp}}), N(n_{cp}|\mu_{n_{cp}}, \Sigma_{n_{cp}}))\right] \tag{5.60}$$

Dans le domaine cepstral, la matrice de covariance du bruit $n_{cp}$ est indépendante de $h_{cp}$ et sa valeur est la même que celle de $(n*h)_{cp}$. L'equation 5.60 devient :

$$(\hat{h}^{cp}, \hat{\mu}_{n_{cp}}) = \arg\max_{(h_{cp},\mu_{n_{cp}})} \left[p(y_{cp}|h^{cp}, N(s_{cp}|\mu_{s_{cp}}, \Sigma_{n_{cp}}), N(s_{cp}|\mu_{n_{cp}}, \Sigma_{n_{cp}}))\right] \tag{5.61}$$

Pour résoudre le problème de maximisation de l'équation 5.61, nous procédons d'une manière itérative une fois nous maximisons par rapport à $h_{cp}$ en supposant l'existence d'une estimation de $\mu_{n_{cp}}$ (la dernière estimation de $\mu_{n_{cp}}$), et une fois par rapport à $\mu_{n_{cp}}$ en utilisant la nouvelle estimation de $h_{cp}$. Soit $h_{cp}^k$ et $\mu_{n_{cp}}^k$ les estimations de $h_{cp}$ et $\mu_{n_{cp}}$ à l'itération $k$, $h_{cp}^{k+1}$ est estimé comme suit :

$$\begin{aligned} h_{cp}^{k+1} &= \operatorname{argmax}_{h_{cp}} \left[ p(y_{cp}|h^{cp}, N(s_{cp}|\mu_{s_{cp}}, \Sigma_{s_{cp}}), N(n_{cp}|\mu_{n_{cp}}^k, \Sigma_{n_{cp}})) \right] \\ &= \operatorname{argmax}_{h_{cp}} \left[ p(y_{cp}|N((s_{cp} \oplus n_{cp}) + h_{cp}|\mu_{((s_{cp}\oplus n_{cp})+h_{cp})}, \Sigma_{((s_{cp}\oplus n_{cp}))})) \right] \end{aligned}$$

La maximisation de cette équation donne :

$$h_{cp}^{k+1} = \frac{1}{T}\sum_{t=1}^{T} y_{cp}^t - \mu_{(s_{cp}\oplus n_{cp}^k)}$$

$\mu_{n_{cp}}^{k+1}$ est estimé en utilisant la nouvelle estimation de $h_{cp}$, c'est-à-dire $h_{cp}^{k+1}$ :

$$\begin{aligned} \mu {n_{cp}}^{k+1} &= \operatorname{argmax}_{\mu_{n_{cp}}} \left[ p(y_{cp}|h^{cp}, N(s_{cp}|\mu_{s_{cp}}, \Sigma_{s_{cp}}), N(n_{cp}|\mu_{n_{cp}}^k, \Sigma_{n_{cp}})) \right] \\ &= \operatorname{argmax}_{\mu_{n_{cp}}} \left[ p(y_{cp} - h^{cp}|N((s_{cp} \oplus n_{cp})|\mu_{((s_{cp}\oplus n_{cp}))}, \Sigma_{((s_{cp}\oplus n_{cp}))})) \right] \end{aligned}$$

La maximisation de cette équation donne :

$$\mu_{n_{cp}}^{k+1} = \mu_{(n*h)_{cp}} - h_{cp}^{k+1}$$

nous construisons ainsi une suite croissante (à chaque itération) et majorée, donc convergente. Dans notre cas, nous initialisons ce processus itératif avec $\lambda_{n_{cp}}^0 = \lambda_{(n*h)_{cp}}$.

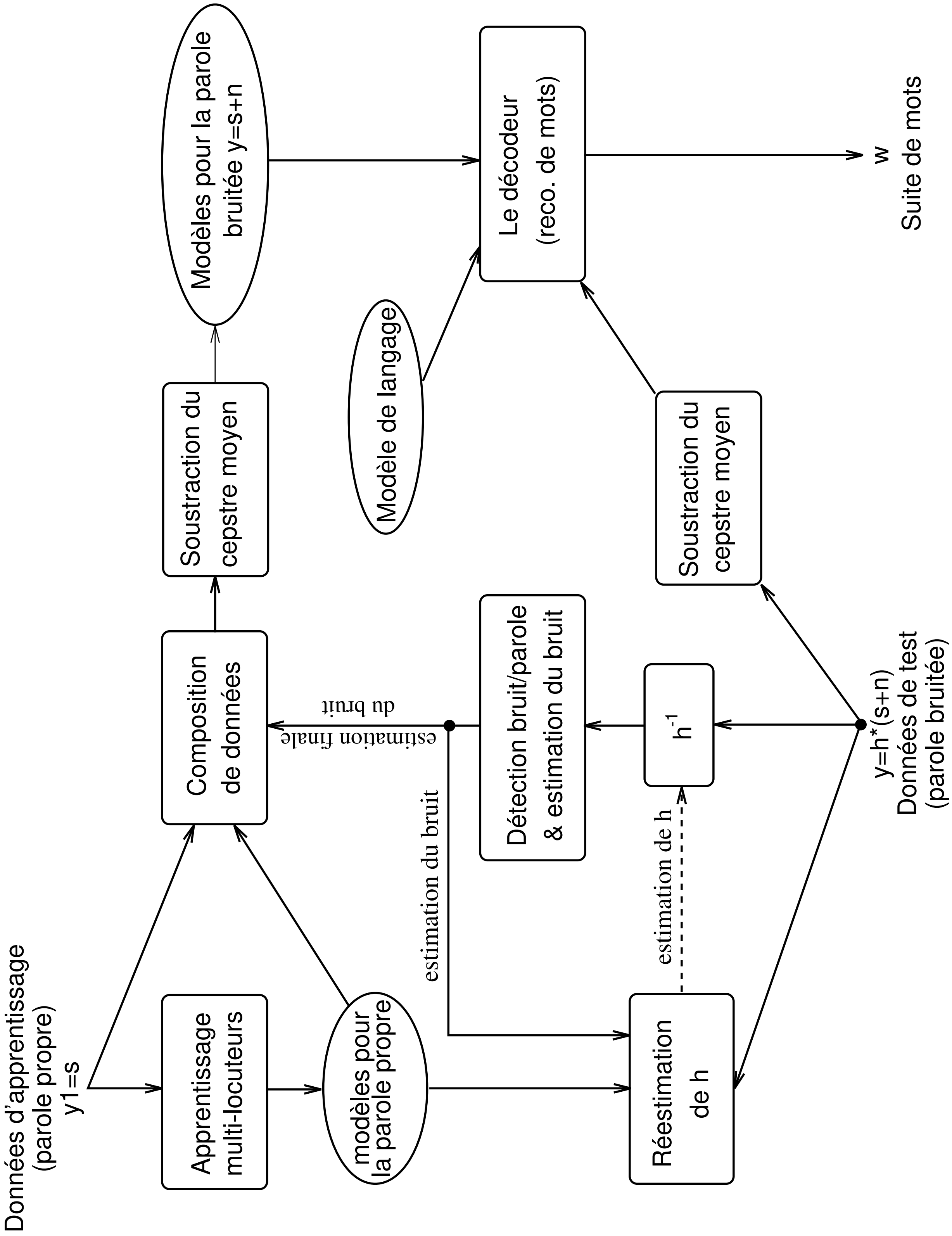



FIG. 5.7 - Cette figure montre la stratégie de la CPD pour compenser les bruits additifs et convolutifs. L'estimation du bruit qu'il faut utiliser dans la combinaison est réalisée itérativement. La compensation du bruit convolutif se fait après combinaison par soustraction du cepstre moyen des données d'apprentissage et de test.

## 5.4 Résultats expérimentaux

Pour évaluer les approches de compensation présentées dans les les sections précédentes, nous avons réalisé nos expériences essentiellement sur le corpus MASK [42] (voir paragraphe 3.1). Une seule expérience a été réalisée dans le cadre des évaluations ARPA [26].

### 5.4.1 Effet de l'approximation *log-normale*

La première expérience a pour objectif de montrer l'effet de l'approximation *log-normale* dans le cas d'un système de reconnaissance de la parole utilisant des MMCDC avec des gaussiennes dont les variances sont grandes. Pour cela, nous utilisons un système de reconnaissance utilisant des MMCDC avec, comme densité d'observation, une seule gaussienne associée à chaque état. Pour mettre en évidence l'effet de cette approximation sur le taux de reconnaissance, seuls les coefficients *cepstraux* statiques sont utilisés. Le bruit qui affecte les données de test est celui de brouhaha (Voice babble noise) extrait de la base de données NOISEX-92 avec un RSB de 12dB. Dans ce cas, le taux d'erreur sur les mots en utilisant l'approximation *log-normale* est de 39% ; en revanche, le taux d'erreur obtenu sans l'approximation *log-normale* est de 32%. Cela signifie que le taux d'erreur est augmenté de 22% à cause de l'approximation *log-normale*. Dans le cas du bruit d'hélicoptère (Lynx) avec un rapport signal sur bruit de 12dB, le taux d'erreur sur les mots en utilisant l'approximation *log-normale* est de 26.2% ; en revanche, le taux d'erreur obtenu sans l'approximation *log-normale* est de 23.2%. Dans ce cas, le taux d'erreur est augmenté de 13% à cause de l'approximation *log-normale*. L'approximation *log-normale* a plus d'effet négatif dans le cas du bruit de brouhaha que dans le cas du bruit d'hélicoptère. Cela est due au fait que le bruit d'hélicoptère utilisé varie très peu dans le temps et donc possède une très petite variance, contrairement au bruit de brouhaha.

### 5.4.2 La CPM avec l'approximation *log-normale*

Les expériences qui suivent ont pour objectif de montrer l'effet de la compensation des paramètres *cepstraux* statiques et dynamiques en utilisant l'approche de la combinaison parallèle de modèles, puis en utilisant l'approche de la combinaison parallèle de données. Les densités d'observation associées aux états sont des mélanges de gaussiennes (typiquement 20 gaussiennes). Trois types de bruits ont été utilisés : le bruit Hautes Fréquences (Figure 3.7), le bruit de brouhaha provenant de la base de données NOISEX-92 (Figure 3.10) et un bruit enregistré à la gare Saint-Lazare à Paris (Figure 3.13). Ces bruits ont été additionnés artificiellement aux données de test. Les données de test sont enregistrées avec le même microphone que celui utilisé pour enregistrer les données d'apprentissage (pas de bruit convolutif). Plusieurs niveaux de bruit ont été utilisés pour produire différents RSB dans les données de test. Le $\Delta$*cepstre* dans ces expériences est calculé en utilisant la fenêtre $[-1, 0, 0, 0, 1]$ et le $\Delta\Delta$*cepstre* est calculé en utilisant la fenêtre $[-1, 0, 1]$.

Les tables 5.1, 5.2 et 5.3 montrent que pour des RSB faibles dans les données de test, les performances du système de reconnaissance chutent considérablement quel que soit le type de bruit affectant les données de test. Par exemple, avec un RSB de 5dB, le taux d'erreur sur les mots est de 66.2% dans le cas du bruit enregistré à la gare Saint-Lazare, de 70.2% dans le cas du bruit Hautes Fréquences (Figure 3.7) et de 57% dans le cas du bruit de brouhaha.

Dans le cas du bruit Hautes Fréquences (Figure 3.7) nous remarquons que la compensation des coefficients *cepstraux* statiques apporte un gain très important par rapport à la compensation de tous les coefficients *cepstraux* (statiques & dynamiques). Par exemple, avec un RSB de 8dB, dans le cas du bruit Hautes Fréquences (table 5.2), la compensation des coefficients *cepstraux* statiques apporte un gain de 80% par rapport au résultat obtenu sans compensation (de 43.3% à 8.6%) ; en revanche la compensation du *cepstre* et du $\Delta$*cepstre* apporte un gain de 82% (de 43.3% à 7.9%) et la compensation du *cepstre* , du $\Delta$*cepstre* et du $\Delta^2$*cepstre* apporte un gain de 83% (de 43.3% à 7.5%) par rapport au résultat sans compensation. Par contre, pour les bruits non stationnaires (bruit enregistré à la gare Saint-Lazare : table 5.1, bruit de brouhaha : table 5.3), la compensation des coefficients *cepstraux* statiques et dynamiques apporte un gain significatif par rapport à la compensation des coefficients *cepstraux* statiques uniquement, surtout pour les RSB faibles dans les données de test. Par exemple, avec un RSB de 5dB, dans le cas du bruit de brouhaha (table 5.3), la compensation du cepstre statique apporte un gain de 38% par rapport au résultat sans compensation (de 57% à 35.3%) ; en revanche, la compensation des coefficients *cepstraux* statiques et dynamiques apporte un gain de 54% par rapport au résultat sans compensation (de 57% à 26.3%).

| Type de la compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Sans | 6.6% | 8.0% | 22.3% | 49.2% | 66.2% |
| *cepstre* | 6.7% | 7.7% | 10.3% | 17.2% | 32.7% |
| *cepstre* & $\Delta$*cepstre* | 7.0% | 7.3% | 10.0% | 12.7% | 26.7% |
| cepstre & $\Delta$*cepstre* & $\Delta^2$*cepstre* | 6.8% | 7.6% | 9.9% | 14.1% | 22.1% |

**Table 5.1 :** *Dans ces expériences, le bruit de test est enregistré à la gare Saint-Lazare, il est utilisé avec différents niveaux pour simuler différents RSBs dans les données de test. Les données d'apprentissage sont propres (35dB). Les taux d'erreur sont présentés dans différentes configurations : 1. sans compensation, 2. avec compensation du cepstre, 3. avec compensation du cepstre et $\Delta$cepstre, 4. avec compensation du cepstre, $\Delta$cepstre et $\Delta^2$cepstre.*

| Type de la compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Sans | 6.6% | 7.1% | 20.6% | 43.3% | 70.2% |
| *cepstre* | 6.7% | 6.1% | 7.4% | 8.6% | 12.7% |
| *cepstre* & $\Delta$*cepstre* | 7.0% | 5.7% | 6.8% | 7.9% | 11.3% |
| *cepstre* & $\Delta$*cepstre* & $\Delta^2$*cepstre* | 6.8% | 5.8% | 6.8% | 7.5% | 10.4% |

**Table 5.2 :** *Dans ces expériences, le bruit de test est celui de la Figure 3.7 (Hautes Fréquences), il est utilisé avec différents niveaux pour simuler différents RSBs dans les données de test. Les données d'apprentissage sont propres (35dB). Les taux d'erreur sont présentés dans différentes configurations : 1. sans compensation, 2. avec compensation du cepstre, 3. avec compensation du cepstre et $\Delta$cepstre, 4. avec compensation du cepstre, $\Delta$cepstre et $\Delta^2$cepstre.*

| Type de la compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Sans | 6.6% | 9.7% | 28.3% | 46.2% | 57.0% |
| *cepstre* | 6.7% | 8.4% | 17.3% | 25.9% | 35.3% |
| *cepstre* & $\Delta$*cepstre* | 7.0% | 8.8% | 15.9% | 22.8% | 29.3% |
| *cepstre* & $\Delta$*cepstre* & $\Delta^2$*cepstre* | 6.8% | 8.7% | 14.5% | 20.1% | 26.3% |

**Table 5.3 :** *Dans ces expériences, le bruit de test est celui de brouhaha (Voice babble), il est utilisé avec différents niveaux pour simuler différents RSBs dans les données de test. Les données d'apprentissage sont propres (35dB). Les taux d'erreur sont présentés dans différentes configurations : 1. sans compensation, 2. avec compensation du cepstre, 3. avec compensation du cepstre et $\Delta$cepstre, 4. avec compensation du cepstre, $\Delta$cepstre et $\Delta^2$cepstre.*

Les premières colonnes des tables 5.1, 5.2, 5.3, montrent que lorsque les données de test sont propres, la compensation des coefficients *cepstraux* statiques et dynamiques augmente le taux d'erreur de 3% par rapport au résultat obtenu sans compensation (de 6.6% à 6.8%). Nous remarquons aussi que la compensation des coefficients *cepstraux* dynamiques, n'est pas avantageuse dans le cas des grands RSB (20dB et 35dB). Cependant, mis à part le temps nécessaire aux calculs, il est généralement avantageux de compenser tous les types de coefficients cepstraux (statiques et dynamiques).

### 5.4.3 Combinaison Parallèle de Données : bruit additif

Dans cette section, nous présentons les résultats dans le cas d'une compensation des coefficients *cepstraux* (statiques et dynamiques) utilisant la combinaison parallèle de données. Afin de pouvoir comparer cette approche avec celle de la combinaison parallèle de modèles, les mêmes données de test que dans la section précédente sont utilisées. Nous ne présentons que les résultats obtenus en compensant conjointement tous les coefficients *cepstraux* (statiques et dynamiques), car une fois les coefficients *cepstraux* statiques compensés, la quantité de calculs supplémentaire pour compenser les coefficients *cepstraux* dynamiques n'est pas très importante. Nous présentons les résultats de la compensation dans deux cas ; le premier cas, lorsqu'on utilise la fenêtre $[-1, 0, 0, +1]$ pour le calcul du $\Delta$*cepstre*, et le deuxième cas, lorsqu'on utilise la fenêtre $[-2, -1, 0, +1, +2]$ pour le calcul du $\Delta$*cepstre*. Dans les deux cas nous utilisons la fenêtre $[-1, 0, +1]$ pour le calcul de $\Delta^2$*cepstre*. Puisque la soustraction du cepstre moyen ne coûte pratiquement rien en temps de calcul, nous présentons les résultats avec et sans soustraction du cepstre moyen. Le système utilisé dans les expériences de cette section est entraîné sur les données d'apprentissage propres (35dB). Le même canal d'enregistrement a été utilisé pour l'apprentissage et le test (pas de bruit convolutif). Les taux d'erreur sur les mots sont présentés dans différentes configurations : 1. sans compensation du bruit additif mais avec soustraction du cepstre moyen, 2. avec compensation du bruit additif mais sans soustraction du cepstre moyen, 3. avec compensation du bruit additif et soustraction du *cepstre* moyen. Dans les tables 5.4, 5.5 et 5.6 nous utilisons la notation “SCM” pour désigner le terme “Soustraction du Cepstre Moyen sur la phrase”.

La comparaison des résultats obtenus avec la fenêtre $[-1, 0, 0, +1]$ et ceux obtenus avec la fenêtre $[-2, -1, 0, +1, +2]$ pour le calcul du $\Delta$*cepstre* montre que la différence, en terme de taux d'erreur sur les mots, peut être très significative surtout si aucune compensation n'est réalisée. Lorsqu'on compense tous les coefficients cepstraux (statiques et dynamiques), l'utilisation de la fenêtre $[-2, -1, 0, +1, +2]$ pour le calcul du $\Delta$*cepstre* donne de meilleurs résultats dans la majorités des configurations et pour tous les types de bruits testés. Une comparaison entre les tables (5.1 / 5.4), (5.2 / 5.5) et (5.3 / 5.6) montre que dans le cas de bruits non stationnaires (le bruit de foule et le bruit enregistré à la gare Saint-Lazare), les résultats obtenus en utilisant la combinaison parallèle de données sont meilleurs que ceux obtenus avec la combinaison parallèle de modèles ; alors que dans le cas de bruit stationnaire les résultats sont similaires. Une comparaison des résultats obtenus avec et sans soustraction du cepstre moyen, montre qu'il est généralement avantageux de réaliser cette opération même si les données de test et d'apprentissage sont enregistrées avec le même microphone. Cela peut s'expliquer par le fait que la soustraction du cepstre moyen réalise une certaine normalisation (partielle) par rapport au locuteur.

Nous remarquons que pour les deux approches, la combinaison de modèles et la combinaison de données, la compensation du bruit Hautes Fréquences donne de très bons résultats, même pour des RSB faibles ; par exemple, pour un RSB de 5dB, la compensation de ce bruit avec l'approche de la combinaison de données fait passer le taux d'erreur sur les mots de 70.2% à 10.4%, soit un gain relatif de 85% ; avec l'approche de la combinaison de données, le taux d'erreur sur les mots passe de 70.2% à 8.9%, soit un gain relatif de 87%. En revanche, les taux d'erreur obtenus avec la compensation du bruit enregistré à la gare Saint-Lazare et la compensation du bruit de brouhaha demeurent non satisfaisants, surtout pour les faibles RSBs. Par exemple, avec un RSB de 5dB la compensation de bruit de brouhaha avec l'approche de la combinaison de données fait passer le taux d'erreur sur les mots de 57% à 26.9%. Ce dernier taux d'erreur reste 4.5 fois plus grand que celui obtenu lorsque le test se déroule comme l'apprentissage, dans une ambiance calme. Cela vient du fait que le bruit de brouhaha varie considérablement en fonction du temps et affecte d'une manière très variable les fréquences utiles du signal de parole. En revanche, le bruit Hautes Fréquences affecte le spectre du signal de parole mais cause peu de perte d'information dans la mesure où cette dernière est concentrée dans les basses fréquences. Ce type de bruit peut donc être très nuisible à la reconnaissance si on ne le compense, mais il est beaucoup plus facile à compenser que les bruits affectant les basses fréquences du signal de parole.

| Type de compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-1, 0, 0, 0, +1]$ | | | | | |
| Sans compensation + SCM | 6.6% | 8.0% | 22.3% | 49.2% | 66.2% |
| Compensation sans SCM | 7.1% | 7.2% | 9.4% | 13.0% | 19.9% |
| Compensation + SCM | 6.1% | 7.4% | 9.3% | 13.1% | 19.0% |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-2, -1, 0, +1, +2]$ | | | | | |
| Sans | 5.9% | 7.5% | 15.8% | 34.9% | 56.4% |
| Compensation sans SCM | 6.6% | 6.9% | 8.7% | 13.5% | 20.5% |
| Compensation + SCM | 5.9% | 7.4% | 8.5% | 12.9% | 18.2% |

**Table 5.4 :** *Dans ces expériences, le bruit de test est celui enregistré à la gare Saint-Lazare. SCM signifie "Soustraction du Cepstre Moyen".*

| Type de Compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-1, 0, 0, 0, +1]$ | | | | | |
| Sans compensation + SCM | 6.6% | 7.1% | 20.6% | 43.3% | 70.2% |
| Compensation sans SCM | 7.1% | 7.2% | 7.5% | 8.28% | 10.30% |
| Compensation + SCM | 6.1% | 7.2% | 7.7% | 7.8% | 9.7% |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-2, -1, 0, +1, +2]$ | | | | | |
| Sans compensation + SCM | 5.9% | 7.5% | 19.9% | 50.4% | 76.4% |
| Compensation sans SCM | 6.6% | 7.2% | 7.3% | 8.2% | 9.8% |
| Compensation + SCM | 5.9% | 6.5% | 7.4% | 8.4% | 8.9% |

**Table 5.5 :** *Dans ces expériences, le bruit Hautes Fréquences de la Figure 3.7 est utilisé pour bruiter les données de test. SCM signifie "Soustraction du Cepstre Moyen".*

| Type de compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-1, 0, 0, 0, +1]$ | | | | | |
| Sans compensation + SCM | 6.6% | 9.7% | 28.3% | 46.2% | 57.0% |
| Compensation sans SCM | 7.1% | 8.3% | 14.1% | 19.9% | 25.2% |
| Compensation + SCM | 6.1% | 8.1% | 14.1% | 19.7% | 27.8% |
| Le $\Delta cepstre$ est calculé en utilisant la fenêtre $[-2, -1, 0, +1, +2]$ | | | | | |
| Sans compensation + SCM | 5.9% | 8.7% | 25.3% | 51.2% | 69.6% |
| Compensation sans SCM | 6.6% | 7.9% | 14.8% | 20.1% | 28.3% |
| Compensation + SCM | 5.9% | 7.7% | 13.1% | 20.3% | 26.9% |

**Table 5.6 :** *Dans ces expériences, le bruit de brouhaha (Voice babble noise) extrait de la base de données NOISEX-92 est utilisé pour bruiter les données de test. SCM signifie "Soustraction du Cepstre Moyen".*

### 5.4.4 Combinaison Parallèle de Données : bruit additif et convolutif

**Résultats sur le corpus Mask:**

Pour évaluer l'algorithme de compensation du bruit additif et convolutif, nous avons enregistré les données par deux microphones simultanément, un microphone de tête ($S_a$ avec un RSB $\simeq$ 35dB) et un microphone de table ($S_b$ avec un RSB $\simeq$ 20dB). Comme dans les expériences précédentes, les modèles ont été entraînés sur les signaux $S_a$ (RSB = 35dB). La première colonne dans chaque table utilise le signal $S_a$ pour le test. La deuxième colonne dans chaque table utilise le canal $S_b$ pour le test. Pour les trois dernières colonnes de chaque table nous avons utilisé pour le test : 1- pour les expériences de la table 5.7, les signaux $S_b$ bruités avec le bruit enregistré à la gare Saint-Lazare (Figure 3.13), 2- pour les expériences de la table 5.8, les signaux $S_b$ bruités avec le bruit Hautes Fréquences (Figure 3.7) 3- pour les expériences de la table 5.9, les signaux $S_b$ bruités avec le bruit de foule (brouhaha : *Voice bable noise*)(Figure (Figure 3.7)(3.10). Cette configuration nous permet de tester l'algorithme dans les cas suivants : 1- absence du bruit additif et convolutif, 2- présence du bruit additif et absence du bruit convolutif, 3- présence du bruit additif et convolutif. Différents rapports signal sur bruit dans les données de test sont explorés.

La combinaison parallèle de données permet l'utilisation de n'importe quelle formule pour le calcul des coefficients cepstraux dynamiques. Dans ces expériences, nous avons fait le choix d'utiliser la fenêtre $[-2, -1, 0, +1, +2]$ pour le calcul du $\Delta cepstre$ et la fenêtre $[-1, 0, +1]$ pour le calcul du $\Delta^2 cepstre$. Tous les coefficients cepstraux (statiques et dynamiques) sont compensés. Dans les expériences de ce paragraphe (Tables 5.7, 5.8 et 5.9) la soustraction du cepstre moyen est appliquée à toutes les phrases d'apprentissage et de test.

| Combinaison parallèle de données | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| sans | 5.9% | 11.1% | 23.0% | 48.2% | 67.5% |
| avec | 5.9% | 10.2% | 10.5% | 12.8% | 19.3% |

**Table 5.7 :** *Dans ces expériences, le bruit enregistré à la gare Sain-Lazare est utilisé pour bruiter les données de test.*

| Compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Sans compensation | 5.9% | 11.1% | 16.3% | 24.9% | 42.4% |
| Avec compensation | 5.9% | 10.2% | 10.1% | 10.3% | 12.9% |

**Table 5.8 :** *Dans ces expériences, le bruit Hautes Fréquences (Figure 3.7) est utilisé pour bruiter les données de test.*

| Compensation | RSB dans le test | | | | |
|---|---|---|---|---|---|
| | 35dB | 20dB | 12dB | 8dB | 5dB |
| Sans compensation | 5.9% | 11.1% | 34.0% | 58.5% | 72.0% |
| Avec compensation | 5.9% | 10.2% | 15.3% | 23.6% | 35.5% |

**Table 5.9** *Dans ces expériences, le bruit de brouhaha (Voice babble noise) extrait de la base de données NOISEX-92 est utilisé pour bruiter les données de test.*

Les résultats de ces expériences montrent qu'il est toujours intéressant de compenser le bruit de test. Nous remarquons qu'en l'absence du bruit additif et convolutif, leur compensation ne trouble pas le bon fonctionnement du système. En effet, lorsque le système est appris et testé sur $S_a$, le taux d'erreur sur les mots ne change pas après la compensation (voir $1^{ère}$ colonne des tables 5.7, 5.8 et 5.9).

De même que dans les expériences de la section précédente, la compensation du bruit Hautes Fréquences (sur $S_b$) donne de très bons résultats même pour des RSB faibles, par exemple, pour un RSB de 5dB, la compensation des bruits additifs et convolutifs fait passer le taux d'erreur sur les mots de 42.4% à 12.9% (dernière colonne de la table 5.8), soit un gain relatif de 70%. En revanche, les taux d'erreur obtenus avec la compensation du bruit enregistré à la gare Saint-Lazare et la compensation du bruit de brouhaha restent non satisfaisants, surtout pour des faibles RSB. Par exemple, avec un RSB de 5dB, la compensation du bruit de brouhaha (sur $S_b$) fait passer le taux d'erreur sur les mots de 72.0% à 35.5%. Ce dernier taux d'erreur reste 6 fois plus grand que celui obtenu lorsque le test se déroule comme l'apprentissage dans une ambiance calme.

**Résultats sur le corpus ARPA :**

Dans le cadre des évaluations ARPA, nous avons utilisé le système de reconnaissance de parole continue du LIMSI [26] qui utilise un vocabulaire de 65.000 mots. Nous présentons ici les résultats obtenus sur les données de test ARPA NAB CSR de novembre 1995 qui ont été enregistrées dans un environnement bruité (47 à 61dBA). Ces données de test comprennent 300 phrases prononcées par 20 locuteurs (15 phrases par locuteur,$\cong$ 20 mots/phrase). Chaque énoncé a été enregistré simultanément avec deux microphones : un microphone Sennheiser HMD-410 (RSB moyen égal à 30dB), et un microphone inconnu du système (RSB de 7 à 18dB).

| *Combinaison parallèle des données.* | *Taux d'erreur sur les mots (%)* | |
|---|---|---|
| | *P0* | *C0* |
| non | >50.0 | 10.4 |
| oui | 20.5 | 10.4 |

**Table 5.10 :** *taux d'erreur (sur les mots) avec et sans technique de compensation pour les deux types de données (C0: propres, P0: bruitées).*

Le même système de reconnaissance, entraîné uniquement sur des données propres enregistrées avec un microphone Sennheiser HMD-410, a été utilisé pour décoder les deux ensembles de données (C0: Sennheiser microphone, P0: autres microphones). Les résultats en terme de taux d'erreur sur les mots sont donnés dans le table 5.10. On peut observer que l'utilisation des techniques de compensation améliore d'une manière significative les performances sur les données bruitées. La technique de combinaison parallèle de données compense le bruit additif lié à l'environnement et le bruit convolutif lié au microphone. Le grand avantage de cette technique est qu'elle améliore de manière très significative les performances dans le cas de présence de bruit additif ou convolutif ($>$ 50%/20.5% d'erreurs), tout en gardant les mêmes performances dans le cas d'absence de bruit (10.4%/10.4% d'erreurs).

## 5.5 Conclusion

Nous avons proposé et évalué une nouvelle approche de compensation permettant d'obtenir des modèles acoustiques correspondant aux signaux de parole bruitée avec très peu d'approximations. Elle permet d'estimer les paramètres statistiques correspondant aux coefficients *cepstraux* dynamiques quelles que soient les formules utilisées pour leurs calculs (une simple différence, une régression ou autre). Un autre avantage lié à cette approche est le fait qu'elle soit applicable quel que soit le type de paramétrisation acoustique utilisée[13] (MFCC, LPCC, PLPCC *etc*).

La compensation des bruits additifs et convolutifs affectant les données en phase

[13] Pour peu qu'il y ait possibilité de retour du domaine de cette paramétrisation à un domaine où la relation de combinaison entre le signal propre et le bruit pour former la signal bruité est la plus simple possible (par exemple une simple addition dans le domaine spectral).

d'utilisation est toujours fructueuse en présence de bruit, et ne dégrade pas les performances quand les données sont propres. Par exemple, dans le cadre de l'application MASK [42], pour des données de test enregistrées avec un microphone différent de celui utilisé en phase d'apprentissage, puis bruitées avec un bruit enregistré en gare Saint-Lazare (RSB=5dB), la compensation apporte un gain relatif (en terme de taux d'erreur sur les mots) de 71% (de 67.5% sans compensation à 19.3% avec compensation) ; dans le cas de bruit de brouhaha affectant les données de test, le gain relatif est de 50% ; dans le cas du bruit Hautes Fréquences le gain relatif est de 70%. Nous avons pu vérifier, d'après les expériences réalisées, qu'il est plus difficile de compenser les bruits affectant les basses fréquences (comme le bruit de brouhaha) que les bruits affectant les hautes fréquences (comme le bruit de la Figure 3.7) , surtout pour les faibles rapports signal sur bruit.

Dans le cadre des évaluations ARPA, avec le système de reconnaissance de la parole continue du LIMSI [26], on obtient un taux d'erreur sur les mots de 10.4% sur le corpus de test ARPA NAB CSR[14] de novembre 1995 dont le rapport signal sur bruit est de 30dB environ. En revanche, en utilisant le même système sur le même corpus de test mais avec un rapport signal sur bruit variant entre 7 et 18dB, le taux d'erreur sur les mots est supérieur à 50%. En compensant les bruits additifs et convolutifs qui affectent les données de test, le taux d'erreur sur les mots descend à 20.5%, soit une diminution du taux d'erreur sur les mots supérieure à 60%.

Nous avons montré dans ce chapitre le grand intérêt de l'approche CPD pour la reconnaissance de la parole bruitée. Nous montrons dans le chapitre 6 l'application de cette approche pour l'augmentation de la qualité (débruitage) des signaux de parole dégradés par la présence d'un bruit ambiant (additif).

[14] Il s'agit d'une tâche de reconnaissance de la parole avec un vocabulaire illimité sur le corpus *"North American Business News"* [65].

## 5.6 Compensation des bruits d'apprentissage et de test

Les nombreuses publications concernant le problème de la compensation des différences entre les conditions d'apprentissage et les condition de test, supposent généralement que les données d'apprentissage sont propres et les données de test sont bruitées. Dans la pratique cette supposition est rarement vérifiée. Il est important de noter que si le système de reconnaissance est appris dans un milieu bruité alors il devient moins performant surtout si les données de test sont propres [14, 58, 64, 28]. Dans cette section, nous proposons une technique itérative pour compenser le bruit de test et le bruit d'apprentissage. Nous avons évalué cette technique en utilisant le système de reconnaissance du LIMSI qui utilise des modèles de Markov cachés avec des densités continues pour modéliser les unités acoustiques (MMCDC) [26]. Trois configurations ont été comparées : sans compensation, avec compensation du bruit de test et enfin avec compensation des bruits d'apprentissage et de test. Plusieurs valeurs de rapport signal sur bruit en phase de test et d'apprentissage ont été explorées. Nous avons réalisé deux jeux d'expériences, le premier utilise un bruit stationnaire (bruit d'hélicoptère) dans l'apprentissage et le second utilise un bruit non stationnaire dans l'apprentissage (bruit de foule).

### 5.6.1 Compensation des bruits additifs

**Combinaison**

Dans ce chapitre nous utilisons constamment le terme "combinaison de trames". Il est donc important de préciser son sens. La combinaison de deux trames suppose qu'elles sont additives dans le domaine spectral. L'opération de combinaison dépend du type d'analyse du signal utilisée. Dans le cadre d'une analyse MFCC, il faut appliquer aux deux trames une transformée de Fourier inverse suivie d'une exponentielle. A ce niveau, les deux trames sont exprimées dans le domaine spectral, et la combinaison consiste en une simple addition. Ensuite, nous exprimons la trame résultante dans le domaine *cepstral* en appliquant successivement le logarithme et la transformée de Fourier.

Dans le cadre d'une paramétrisation PLPCC (voir chapitre 2), l'opération de combinaison consiste à faire l'inverse de la récursion *cepstrale* pour retrouver les coefficients de la prédiction linéaire ; à partir de ces coefficients nous appliquons l'inverse de la récursion de Durbin pour retrouver les coefficients de la pseudo-autocorrélation. Ensuite nous appliquons la transformée de Fourier pour revenir dans le domaine spectral (avec la puissance $\gamma$), nous appliquons $(.)^{\frac{1}{\gamma}}$ pour arriver finalement dans le domaine spectral (voir Figure 5.8). Dans ce domaine la combinaison est une simple addition. Ensuite, nous exprimons la trame résultante dans le domaine *cepstral* en appliquant l'opérateur $(.)^{\gamma}$, la récursion de Durbin suivie de la récursion *cepstrale* pour revenir dans le domaine PLPCC. Ainsi quelle que soit la paramétrisation du signal de parole utilisée, pour faire la combinaison il faut retourner dans un domaine où la relation entre les deux sources parole/bruit soit aussi simple qu'une addition. Il faut noter que pour éviter tous les calculs permettant de revenir dans le domaine spectral, il serait intéressant de stocker les données dans ce domaine. De cette façon seules les opérations permettant de passer du domaine spectral au domaine *cepstral*, après combinaison sont nécessaires.

**Modèles utilisés**

Supposons que non seulement les données de test sont bruitées mais aussi les données d'apprentissage. Supposons en outre, que le canal d'enregistrement peut être représenté par un filtre linéaire. Dans ce cas, les signaux d'apprentissage et de test peuvent être raisonnablement modélisés par le modèle $M_2$ comme suit :

$$\begin{aligned} y_1(t) &= h(t) * (s(t) + n_1(t)) \quad \text{pour l'apprentissage} \\ y_2(t) &= h(t) * (s(t) + n_2(t)) \quad \text{pour le test} \end{aligned}$$

où $s_t$ désigne le signal propre (non bruité), $n_1$ et $n_2$ désignent respectivement les bruits additifs dans les données d'apprentissage et dans les données de test, $h$ désigne la composante convolutive due au microphone utilisé. On suppose dans un premier temps que le même microphone est utilisé en phases de test et d'apprentissage.

**Stratégie de compensation**

L'approche consiste à adapter les paramètres des modèles phonétiques (dans notre cas, ce sont des Modèles de Markov Cachés MMC) en utilisant le bruit présent en phase de test et à compenser les trames représentant les données de test en utilisant le bruit présent en phase d'apprentissage. Aucune contrainte n'est imposée concernant les types de bruits présents en phase de test et d'apprentissage.

Pour adapter les paramètres des modèles phonétiques on utilise l'approche de la combinaison parallèle des données décrite dans le paragraphe précédent (5.3). Dans ce cas nous allons aboutir à des modèles acoustiques représentatifs de $y_1{}' = h * (s + n_1 + n_2)$, mais le signal de test est toujours $y_2 = h * (s + n_2)$. Il reste donc, après cette première adaptation, une différence entre les données de test et les données d'apprentissage qui est due au bruit d'apprentissage $n_1$. Pour rendre les conditions de bruit d'apprentissage identiques à celles de test, nous combinons le signal de test avec le bruit d'apprentissage $n_1$, ce qui résulte en un nouveau signal de test $y_2{}' = h * (s + n_2 + n_1)$. En fait, c'est la moyenne du bruit $n_1$ qui est utilisée dans la combinaison avec les données de test ; donc le signal de test adapté au bruit d'apprentissage devient : $y_2{}' = h * (s + n_2 + \overline{n}_1)$. Malgré le fait que la composante convolutive est la même pour les données adaptées d'apprentissage et de test, rien ne nous empêche de profiter des effets positifs de la soustraction du *cepstre* moyen. En effet, la soustraction du *cepstre* moyen, non seulement normalise par rapport au canal d'enregistrement utilisé, mais aussi par rapport à d'autres variabilités, comme la variabilité interlocuteurs (normalisation partielle).

Le fait de bruiter les données de test et les données d'apprentissage dans le but d'avoir les mêmes conditions d'apprentissage et de test ne constitue qu'un premier objectif. En effet, faire apprendre et tester le système de reconnaissance dans les mêmes "conditions" bruitées reste insatisfaisant par rapport à apprendre et tester le système dans les mêmes conditions propres, particulièrement pour des RSB

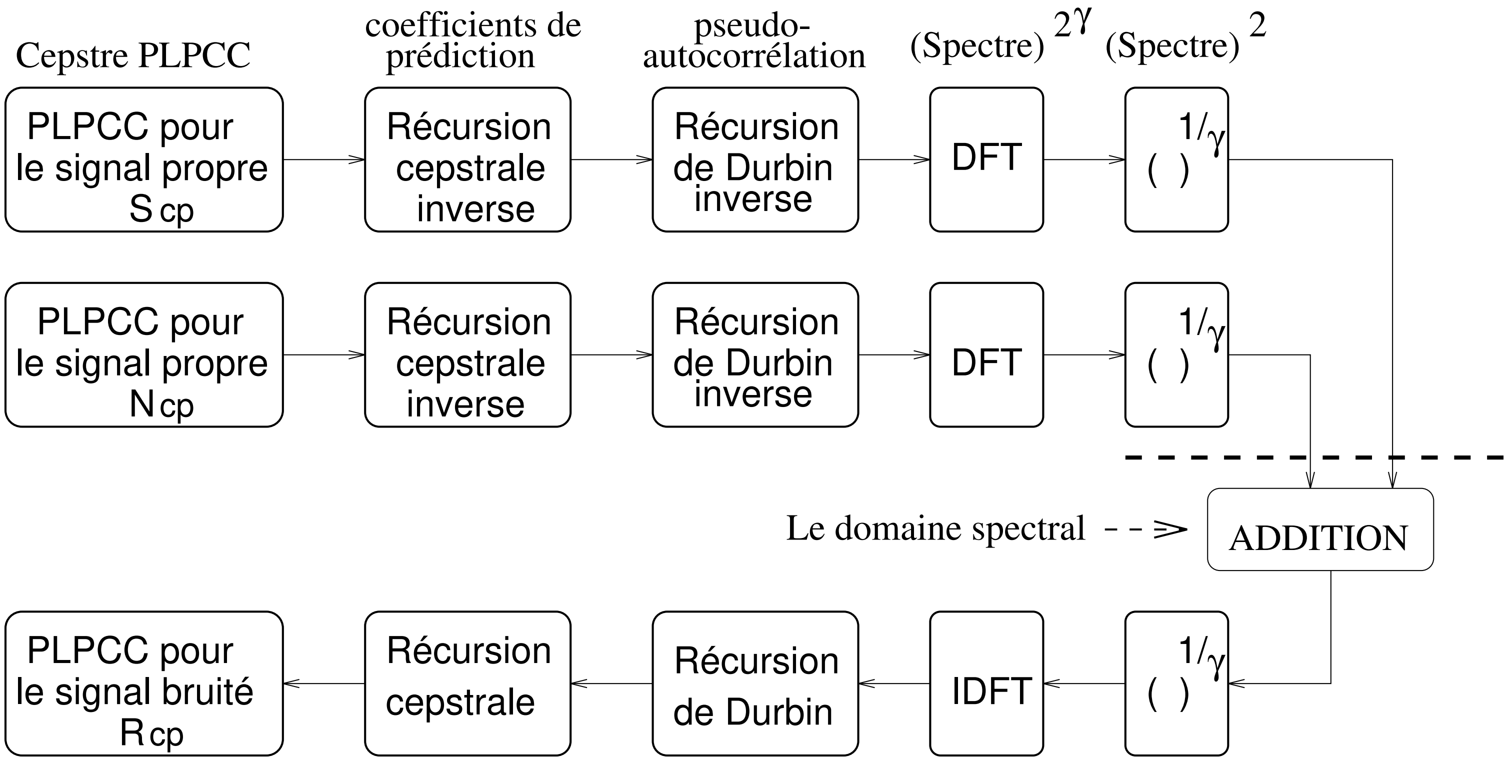



FIG. 5.8 - L'opération de combinaison de trames dans le cadre de la PLPCC.

faibles. Cela est dû au fait que le bruit (ou en général, les transformations affectant le spectre) n'est jamais parfaitement stationnaire, et donc sa présence donnerait forcément des modèles avec de grandes variances c'est-à-dire moins précis.

### 5.6.2 Compensation des bruits additifs et convolutifs

#### Modèles utilisés

Supposons que non seulement les données de test peuvent être bruitées mais aussi les données d'apprentissage. Supposons en outre que les données de test et d'apprentissage sont enregistrées avec des microphones différents. Dans ce cas le modèle du canal de transmission devient :

$$\begin{aligned} y_1(t) &= h_1(t) * (s(t) + n_1(t)) \quad \text{pour l'apprentissage} \\ y_2(t) &= h_2(t) * (s(t) + n_2(t)) \quad \text{pour le test} \end{aligned}$$

nous appelons ce modèle $M_3$, où $s_t$ désigne le signal propre (non bruité), $n_1$ et $h_1$ désignent respectivement les bruits additif et convolutif dans les données d'apprentissage, $n_2$ et $h_2$ désignent respectivement les bruits additif et convolutif dans les données de test.

#### Stratégie de compensation

Dans ce cas nous supposons conformément au modèle $M_3$ que non seulement les données de test sont bruitées, mais aussi les données d'apprentissage. Notre approche consiste à combiner le bruit de test avec les trames correspondant aux données d'apprentissage $y_1$ et la moyenne du bruit d'apprentissage avec les trames correspondant aux données de test. Pour réaliser ces combinaisons, nous avons besoin d'une estimation de $h_1 * n_2$ pour le bruit de test et d'une estimation de $h_2 * \overline{n}_1$ (la moyenne) pour le bruit d'apprentissage, alors que nous ne disposons que des trames représentant $h_2 * n_2$ et $h_2 * \overline{n}_1$ pour les bruits de test et d'apprentissage respectivement. Pour cela, on estime le filtre $h_2^{-1} * h_1$ qu'on applique ensuite à $h_2 * n_2$ pour obtenir une estimation de $h_1 * n_2$. De la même manière, on estime le filtre $h_1^{-1} * h_2$ qu'on applique ensuite aux trames représentant le bruit d'apprentissage $h_1 * \overline{n}_1$ afin d'obtenir une estimation de $h_2 * \overline{n}_1$. Les estimations des filtres $h_1^{-1} * h_2$ et $h_2^{-1} * h_1$ sont obtenues itérativement en utilisant l'algorithme suivant fondé sur le modèle $M_3$ :

**Algorithme 2** :
La combinaison est faite dans le domaine spectral
Tous les autres calculs se font dans le domaine *cepstral* ou *log-spectral*
**1.** Calculer $h_1 * \overline{n}_1$, la moyenne de $h_1 * n_1$
**2.** $\tilde{n}_1 := h_1 * \overline{n}_1$, $1^{ère}$ estimation de $h_2 * \overline{n}_1$
**3.** $\tilde{n}_2 := h_2 * n_2$, $1^{ère}$ estimation de $h_1 * n_2$
**4.** Combiner $y_1$ avec $\tilde{n}_2$ afin d'obtenir une estimation, $\tilde{M}_{appr}$, du *cepstre* moyen de $h_1 * (s + n_1 + n_2)$
**5.** Combiner $y_2$ avec $\tilde{n}_1$ afin d'obtenir une estimation, $\tilde{M}_{test}$, du *cepstre* moyen de $h_2 * (s + n_1 + n_2)$
**6.** Estimer $h_1^{-1} * h_2$ dans le domaine cepstral comme:
$\tilde{H} := \tilde{M}_{test} - \tilde{M}_{appr}$
**7.** Utiliser le filtre $\tilde{H}$ pour obtenir une estimation de $h_2 * \overline{n}_1$ et $h_1 * n_2$:
$\tilde{n}_1 := \tilde{h} * h_1 * \overline{n}_1$
$\tilde{n}_2 := \tilde{h}^{-1} * h_2 * n_2$
**8.** Reprendre en **4.** jusqu'a la convergence

Comme pour l'algorithme 1 du paragraphe 5.3.5, l'algorithme 2 converge au bout de cinq itérations et permet d'estimer correctement les trames correspondant à $h_1 * n_2$ et $h_2 * \overline{n}_1$. Les trames du bruit de test estimées $h_1 * n_2$ sont ensuite combinées dans le domaine spectral avec les trames d'apprentissage $y_1$ pour obtenir les nouvelles trames d'apprentissage $h_1 * (s + n_1 + n_2)$. De la même manière, la moyenne des trames du bruit d'apprentissage estimée $h_2 * \overline{n}_1$ est combinée dans le domaine spectral avec les trames de test $y_2$ pour obtenir les nouvelles trames de test $h_2 * (s + \overline{n}_1 + n_2)$. La normalisation par rapport à la composante convolutive est réalisée par soustraction du *cepstre* moyen des nouvelles trames d'apprentissage et de test, dans notre cas c'est la soustraction du *cepstre* moyen sur la phrase qui est réalisée.

L'extraction des trames de bruit correspondant à l'apprentissage ne pose aucun problème. En ce qui concerne les trames de bruit correspondant au test, nous avons utilisé les quatre premières secondes précédant chaque session de test. Dans la pratique des trames de bruit de fond peuvent être recueillies entre les sessions d'utilisation du système de reconnaissance. Il est ensuite possible de détecter les segments correspondant au bruit durant une session en utilisant le système de reconnaissance adapté avec le bruit initialement recueilli. Ces segments de bruit peuvent maintenant être utilisés pour une adaptation plus efficace.

Il est important de noter que les algorithmes d'estimation de la composante convolutive utilisent le fait que celle-ci est additive dans le domaine *cepstral*. Cela est évidemment valable dans une analyse de type MFCC. Cela est valable aussi dans une analyse de type LPCC et PLPCC. En effet, le *cepstre* dans ce cas n'est rien d'autre que la transformée de Fourier du logarithme du spectre de puissance calculé d'une manière paramétrique (modèle tout-pôles). La composante convolutive devient multiplicative dans le domaine spectral puis additive dans le domaine *log-spectral* et *cepstral*. Pour généraliser notre approche à n'importe quelle paramétrisation, il convient de faire la combinaison dans le domaine spectral et l'estimation de la composante convolutive dans le domaine *log-spectral* ou *cepstral*.

### 5.6.3 Résultats expérimentaux

Pour évaluer nos deux algorithmes de compensation, nous avons utilisé 22148 phrases prononcées par 460 locuteurs, provenant du corpus Mask [42]. Les phrases de 450 locuteurs ont été utilisées pour l'apprentissage du système et les 10 locuteurs restants ont été utilisés pour le test. Les données ont été enregistrées avec deux microphones simultanément : un microphone de tête ($S_a$ avec un RSB $\simeq$ 35dB) et un microphone de table ($S_b$ avec un RSB $\simeq$ 20dB). Le signal de parole a subi un filtrage passe-bas avec une fréquence de coupure égale à 8kHz et a été échantillonné à 16kHz. Mask est une application en grandeur réelle de la technologie de la langue, dont l'objectif est de développer un système de réservation de trains et d'information automatique. Une personne utilisant le système peut demander des informations concernant les voyages en train à travers 500 villes de France. Le système est multi-locuteur, capable de reconnaître la parole continue et spontanée en temps réel. Le système utilise un vocabulaire de 1500 mots et un modèle de langage *bigramme*. Ce système est conçu pour être placé dans différents endroits (des gares) dont les caractéristiques acoustiques d'environnement peuvent varier considérablement dans l'espace et dans le temps. Ces caractéristiques acoustiques d'environnement peuvent être très différentes de celles d'un laboratoire (bruit de fond faible). Pour pallier ce problème, une solution consiste à apprendre les paramètres du système sur des données recueillies au moment de l'utilisation, et donc à prendre en compte le bruit d'apprentissage.

Les vecteurs paramétriques caractérisant une trame sont composés de 13 coefficients *cepstraux* avec leurs dérivées première et seconde et sont estimés toutes les 10ms. La soustraction du *cepstre* moyen est réalisée sur chaque phrase. 608 phonèmes dépendants du contexte (silence inclu) sont modélisés. Chaque phonème dépendant du contexte est un Modèle de Markov Caché à Densité Continue (MMCDC) avec un mélange de gaussiennes comme densités d'observation (typiquement 20 gaussiennes).

Nous avons réalisé deux ensembles d'expériences : **l'expérience I** qui utilise un bruit d'apprentissage stationnaire, et **l'expérience II** qui utilise un bruit d'apprentissage non stationnaire. Le bruit ambiant de test est le même pour les deux ensembles d'expériences. Pour toutes les expériences, les modèles ont été appris sur les signaux $S_a$. Pour simuler les différents rapports signal sur bruit (RSB) dans les données d'apprentissage, le bruit du Lynx (pour l'expérience **I**) et le bruit de brouhaha (Voice babble noise pour l'expérience **II**) de la base de données de bruits NOISEX-92 [81] ont été additionnés avec différents niveaux à $S_a$. Trois valeurs de RSB ont été utilisées pour l'apprentissage : 35dB, 20dB, 12dB. Un bruit enregistré dans la gare Saint-Lazare à Paris a été additionné à $S_b$ avec différents niveaux pour former le signal de test avec trois différents RSB : 20dB, 12dB, 9dB. La première ligne dans chaque table utilise le canal $S_a$ pour le test. Trois configurations ont été expérimentées : *sans compensation*, *compensation du bruit de test* et *compensation* *du bruit de test & d'apprentissage.*

**Résultats de l'expérience I**

Dans ce premier ensemble d'expériences, on utilise un bruit stationnaire en phase d'apprentissage (bruit de Lynx) et un bruit de gare dans les données de test. Toutes les tables montrent les résultats en terme de taux d'erreur sur les mots. Les tables I.5.11, I.5.12 et I.5.13 montrent les taux d'erreur sur les mots pour les configurations suivantes : *sans compensation*, *compensation du bruit de test*, *compensation du bruit de test & d'apprentissage*. La comparaison entre les tables I.5.11 et I.5.12 montre que même pour des RSB relativement élevés, il est toujours intéressant de compenser le bruit de test. Quand les données de test sont propres, la compensation du bruit de test ne trouble pas le bon fonctionnement du système. Si le RSB dans les données d'apprentissage et de 12dB et si le RSB dans les données de test et de 20dB, alors le taux d'erreur décroît de 22.8% à 17.4%. Ce taux d'erreur reste trois fois supérieur au taux d'erreur obtenu en apprenant et en testant le système avec des signaux propres. La table I.5.11 montre que les performances décroissent rapidement quand le système est entraîné sur des données bruitées et testé sur des données propres, ou inversement, quand le système est entraîné sur des données propres et testé sur des données bruitées.

La comparaison entre les tables I.5.12 et I.5.13 montre qu'il y a toujours un gain apporté par la compensation du bruit d'apprentissage, spécialement quand les données de test sont relativement propres. Quand l'apprentissage se fait à un RSB de 12dB et le test se fait à un RSB de 20dB, la compensation du bruit de test permet une décroissance du taux d'erreur de 23% (de 22.8% à 17.4%), alors que la compensation du bruit de test et d'apprentissage permet une décroissance du taux d'erreur de plus de 50% (de 22.8% à 10.4%).

Lorsque les données de test sont propres, la compensation du bruit de test n'apporte aucun gain (comparaison des $1^{ères}$ lignes des tables I.5.11 et I.5.12). Par contre, la compensation du bruit de test et d'apprentissage (voir la $1^{ère}$ ligne de la table I.5.13) permet de réduire considérablement le taux d'erreur en comparaison avec le taux d'erreur obtenu sans compensation (table I.5.11) et avec le taux d'erreur obtenu en compensant le bruit de test (table I.5.12). On peut conclure que la compensation du bruit d'apprentissage est particulièrement fructueuse quand les données de test sont propres, et l'importance du gain apporté par cette compensation augmente avec le niveau de bruit présent dans les données d'apprentissage.

Les tables I.5.11 et I.5.12 montrent que même avec des RSB relativement élevés dans les données de test, la compensation du bruit de test apporte des gains significatifs par rapport aux résultats obtenus sans compensation. Par contre, quand les données de test et d'apprentissage sont légèrement bruitées, le gain apporté par la compensation du bruit d'apprentissage n'est pas très significatif (comparaison des tables I.5.12 et I.5.13). La table I.5.13 montre que le bruit d'apprentissage est mieux compensé quand les données de test sont propres (la première ligne), que le bruit de test quand les données d'apprentissage sont propres (la première colonne) : en faisant l'apprentissage du système sur $S_a$ avec un RSB=20dB ou 12dB, et en testant sur $S_b$ avec RSB=35dB, les taux d'erreur résultant sont 6.4% et 6.5% respectivement, ce qui reste comparable au taux d'erreur obtenu en faisant l'apprentissage et en testant dans les mêmes conditions propres (5.9%). Par contre, en faisant l'apprentissage sur $S_a$ avec un RSB=35dB et en testant sur $S_b$ avec un RSB=20dB ou 12dB, les taux d'erreur obtenus sont respectivement 9.9% et 10.9%. Enfin, il est important de noter que si les données d'apprentissage sont propres, la compensation du bruit d'apprentissage n'affecte pas les performances du système.

**Table I.5.11 :** Sans compensation

| Conditions de test | Conditions d'apprentissage | | |
|---|---|---|---|
| | RSB=35dB | RSB=20dB | RSB=12dB |
| $S_a$ : RSB=35dB | 5.9% | 11.1% | 35.2% |
| $S_b$ : RSB=20dB | 11.1% | 14.8% | 22.8% |
| $S_b$ : RSB=12dB | 27.8% | 25.7% | 24.9% |
| $S_b$ : RSB=9dB | 51.5% | 49.1% | 41.5% |

*Dans ces expériences, le bruit d'hélicoptère (Lynx) a été utilisé pour avoir différents RSB dans les données d'apprentissage. Dans le test, c'est le bruit enregistré à la gare SNCF (Saint-Lazare) qui a été utilisé. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test, dans le cas où aucune compensation n'est réalisée.*

**Table I.5.12 :** Compensation du bruit de test

| Conditions de test | Condition d'apprentissage | | |
|---|---|---|---|
| | RSB=35dB | RSB=20dB | RSB=12dB |
| $S_a$ : RSB=35dB | 5.9% | 11.1% | 35.0% |
| $S_b$ : RSB=20dB | 10.2% | 11.0% | 17.4% |
| $S_b$ : RSB=12dB | 10.9% | 11.9% | 12.7% |
| $S_b$ : RSB=9dB | 13.5% | 15.5% | 16.9% |

*Dans ces expériences, le bruit d'hélicoptère (Lynx) a été utilisé pour avoir différents RSB dans les données d'apprentissage. Pour simuler les différents RSB dans le test, nous utilisons le bruit enregistré à la gare Saint-Lazare. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test lorsqu'on compense seulement le bruit de test.*

**Table I.5.13 :** Compensation des bruits de test et d'apprentissage

| Conditions de test | Training Condition RSB=35dB | RSB=20dB | RSB=12dB |
|---|---|---|---|
| $S_a$ : RSB=35dB | 5.9% | 6.4% | 6.5% |
| $S_b$ : RSB=20dB | 9.9% | 10.4% | 10.4% |
| $S_b$ : RSB=12dB | 10.9% | 11.2% | 11.5% |
| $S_b$ : RSB=9dB | 13.8% | 14.9% | 15.0% |

*Dans ces expériences, le bruit d'hélicoptère (Lynx) a été utilisé pour avoir différents RSB dans les données d'apprentissage. Pour simuler les différents RSB dans le test, nous utilisons le bruit enregistré à la gare Saint-Lazare. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test lorsqu'on compense les deux bruits celui du test et celui d'apprentissage.*

### Résultats de l'expérience II

Dans ce deuxième ensemble d'expériences, on utilise un bruit non stationnaire en phase d'apprentissage (bruit de brouhaha) et un bruit de gare dans les données de test. Les tables II.5.11, II.5.12 et II.5.13 montrent les taux d'erreur sur les mots pour les configurations suivantes : sans compensation, compensation du bruit de test, et enfin compensation du bruit de test & d'apprentissage.

L'expérience **II** confirme que notre algorithme fonctionne, dans le cas d'un bruit non stationnaire, aussi bien que dans le cas d'un bruit stationnaire. On retrouve quasiment les mêmes constatations que dans l'expérience **I**. La seule différence est que la compensation du bruit d'apprentissage introduit une petite dégradation lorsqu'on apprend les modèles sur $S_a$ avec un RSB=12dB et que l'on teste sur $S_b$ avec un RSB=9dB. En effet, on passe de 17.5% dans le cas de la compensation du bruit de test à 18.9% dans le cas de la compensation du bruit de test & d'apprentissage (comparer les tables II.5.12 et II.5.13). Mais, dans la plupart des cas, l'apport de la compensation du bruit d'apprentissage reste toujours significatif.

**Table II.5.11 :** sans compensation

| Conditions de test | Conditions d'apprentissage RSB=35dB | RSB=20dB | RSB=12dB |
|---|---|---|---|
| $S_a$ : RSB=35dB | 5.9% | 10.7% | 32.0% |
| $S_b$ : RSB=20dB | 11.1% | 10.7% | 16.9% |
| $S_b$ : RSB=12dB | 27.8% | 20.0% | 21.8% |
| $S_b$ : RSB=9dB | 51.5% | 40.0% | 35.0% |

*Dans ces expériences, le bruit de brouhaha a été utilisé pour avoir différents RSB dans les données d'apprentissage. Dans le test, c'est le bruit enregistré à la gare SNCF (Saint-Lazare) qui a été utilisé. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test, dans le cas où aucune compensation n'est réalisée.*

**Table II.5.12 :** Compensation du bruit de test

| Conditions de test | Conditions d'apprentissage RSB=35dB | RSB=20dB | RSB=12dB |
|---|---|---|---|
| $S_a$ : RSB=35dB | 5.9% | 12.9% | 23.3% |
| $S_b$ : RSB=20dB | 10.2% | 10.3% | 16.8% |
| $S_b$ : RSB=12dB | 10.9% | 11.7% | 17.7% |
| $S_b$ : RSB=9dB | 13.5% | 14.2% | 17.5% |

*Dans ces expériences, le bruit de brouhaha a été utilisé pour avoir différents RSB dans les données d'apprentissage. Pour simuler les différents RSB dans le test, nous utilisons le bruit enregistré à la gare Saint-Lazare. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test, lorsqu'on compense seulement le bruit de test.*

**Table II.5.13 :** Compensation des bruits d'apprentissage et de test

| Conditions de test | Conditions d'apprentissage RSB=35dB | RSB=20dB | RSB=12dB |
|---|---|---|---|
| $S_a$ : RSB=35dB | 5.9% | 6.5% | 10.8% |
| $S_b$ : RSB=20dB | 9.9% | 9.5% | 12.1% |
| $S_b$ : RSB=12dB | 10.9% | 10.6% | 14.0% |
| $S_b$ : RSB=9dB | 13.8% | 13.4% | 18.9% |

*Dans ces expériences, le bruit de brouhaha a été utilisé pour avoir différents RSB dans les données d'apprentissage. Pour simuler les différents RSB dans le test, nous utilisons le bruit enregistré à la gare Saint-Lazare. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test lorsqu'on compense les deux bruits, celui du test et celui d'apprentissage.*

### 5.6.4 Conclusion

Dans ce chapitre, nous avons décrit une nouvelle technique de compensation qui prend en compte le bruit présent dans les données d'apprentissage et le bruit présent dans les données de test. Les données d'apprentissage et de test sont supposées enregistrées avec des microphones différents et dans des conditions acoustiques différentes. La compensation du bruit d'apprentissage et de test basée sur le modèle du canal de transmission $y = (s + n) * h$ apporte un gain considérable par rapport aux résultats de base (sans compensation) et aux résultats avec compensation du bruit de test seulement. Cela est valable pour les deux types de bruits d'apprentissage, stationnaire et non stationnaire. Le gain obtenu par compensation du bruit d'apprentissage est particulièrement important quand les données de test sont propres. Quand on teste sur le canal $S_a$ avec un RSB égal à 35dB et quand les modèles sont appris sur le canal $S_a$ avec un RSB égal 12dB (resp 20dB), la compensation des bruits d'apprentissage et de test apporte un gain de 80% (resp 40%) dans l'expérience **I** et de 54% (resp 50%) dans l'expérience **II**, par rapport aux résultats obtenus en compensant seulement le bruit de test.

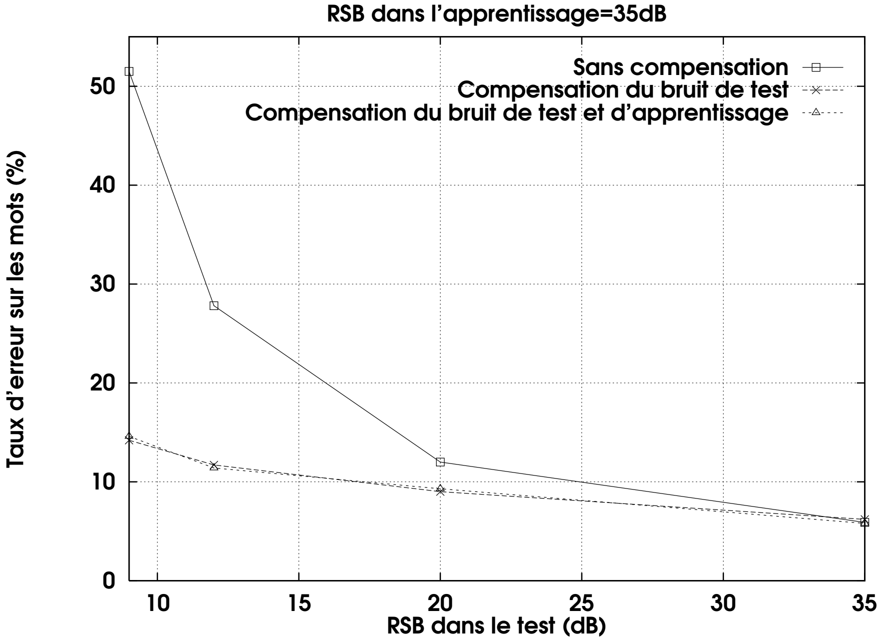



FIG. 5.9 - Dans ces expériences, les données d'apprentissage sont propres. Les trois courbes montrent les taux d'erreur sur les mots en fonction des rapports signal sur bruit dans les données de test.

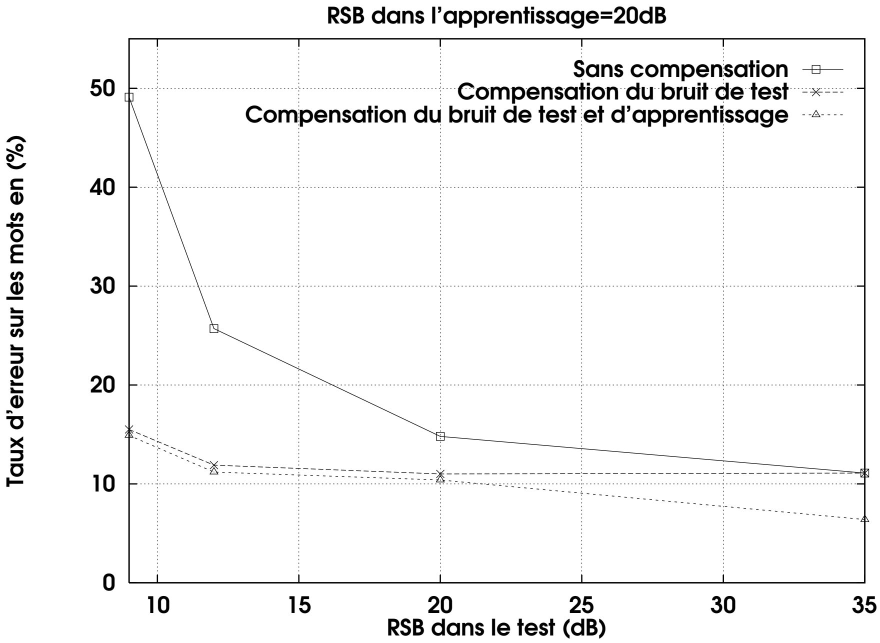



FIG. 5.10 - Dans ces 2 expériences, le données d'apprentissage sont bruitées avec un bruit d'hélicoptère (Lynx) et avec des rapports signal sur bruit de l'ordre de 20 et 12dB respectivement.

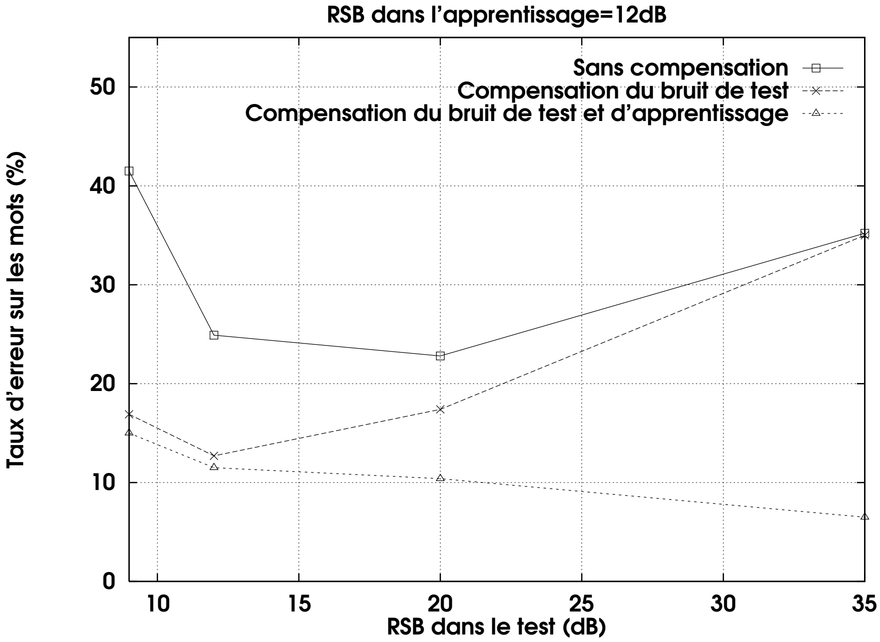



FIG. 5.11 - Dans ces 2 expériences, le données d'apprentissage sont bruitées avec un bruit d'hélicoptère (Lynx) et avec des rapports signal sur bruit de l'ordre de 20 et 12dB respectivement.

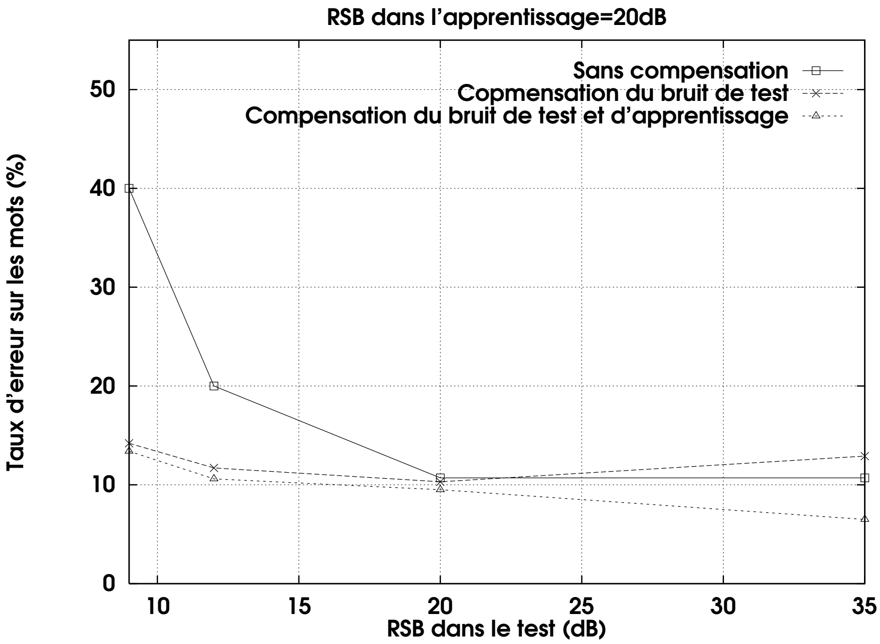



FIG. 5.12 - Dans ces 2 expériences, les données d'apprentissage sont bruitées avec un bruit de voix (brouhaha : non stationnaire) et avec des rapports signal sur bruit de l'ordre de 20 et 12dB respectivement.

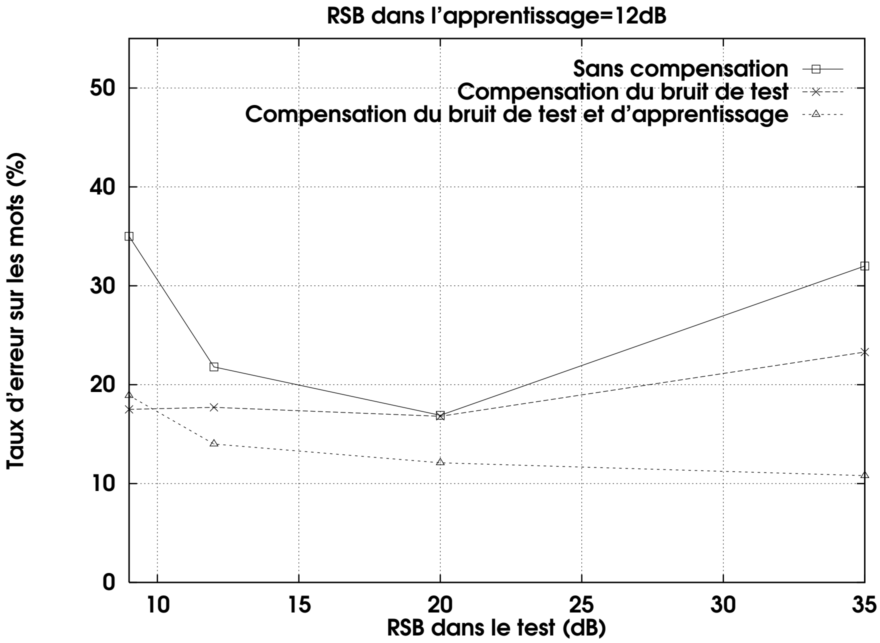



FIG. 5.13 - Dans ces 2 expériences, les données d'apprentissage sont bruitées avec un bruit de voix (brouhaha : non stationnaire) et avec des rapports signal sur bruit de l'ordre de 20 et 12dB respectivement.

Précisons qu'en général le but de la compensation est d'aboutir à des modèles qui maximisent la vraisemblance des données de test. Or l'approche de compensation que nous proposons est fondée sur le modèle du canal de transmission qui n'est pas rigoureusement correct. En outre, cette approche de compensation nécessite certaines approximations qui peuvent résulter en des modèles acoustiques plus ou moins éloignés de ceux maximisant la vraisemblance des données de test. L'effet de ces approximations sur les performances du système de reconnaissance se manifeste clairement dans le cas de bruits très variables dans le temps (brouhaha). Cependant, certaines parmi ces approximations peuvent être corrigées en maximisant la vraisemblance des données d'apprentissage bruitées (en utilisant le modèle du canal de transmission) au niveaux des états. Cette maximisation se fait trivialement en utilisant l'algorithme EM sur les données d'apprentissage bruitées au niveau de chaque état des modèles acoustiques.

Un autre moyen permettant de corriger les erreurs dues aux approximations de l'approche de compensation utilisée est l'emploi de techniques maximisant la vraisemblance de données de test d'une manière aveugle. La MLLR (*Maximum Likelihood Linear Regression*) en constitue un exemple typique[15]. Nous montrons (expérimentalement) dans le chapitre 7 que la MLLR permet de corriger (partiellement) les erreurs commises par la combinaison parallèle de données.

[15] Il faut noter que ce type de technique permet d'approcher le modèle optimal (celui maximisant la vraisemblance de toutes les données de test), mais ne permet d'atteindre le modèle optimal qu'en cas d'une très grande quantité de données d'adaptation (approchant la quantité de données d'apprentissage)

# Chapitre 6

# Application de la CPD : utilisation des MMC pour le débruitage

## 6.1 Introduction

Parmi les techniques de débruitage qui ont été les plus étudiées pendant les deux dernières décennies, on peut citer la technique de la soustraction spectrale [9, 8, 49, 83]. La soustraction spectrale, sous toutes ses formes, appliquée à un signal de parole bruité, afin de produire le signal propre correspondant, ne tire pas profit de l'information sur le contenu linguistique de ce signal. En fait, la plupart des implantations de la soustraction spectrale sont appliquées à un signal de parole exactement de la même manière qu'à n'importe quel autre signal bruité[1]. Cette insuffisance d'information *a priori* dans le processus de débruitage peut engendrer d'importantes distorsions dans les signaux restaurés.

Une première tentative pour prendre en compte l'information existante dans le signal de parole a été introduite par Drucker [16]. Son idée consiste à utiliser différents estimateurs (filtres) du signal propre pour différentes classes ou types d'informations existantes dans le signal. Dans son système de débruitage, il considère cinq catégories du signal de parole : les fricatives, les occlusives, les voyelles, les liquides et les nasales. Dans un premier temps, le système de débruitage de Drucker décompose le signal en zones, chaque zone correspond à l'une des cinq catégories, ensuite le filtre correspondant est appliqué pour déduire une estimation du signal propre. Les filtres utilisés par Drucker sont simplement passe-bas, passe-haut ou passe-bande. Ces filtres sont déterminés en utilisant les propriétés caractérisant chaque classe. Un travail similaire à celui de Drucker a été réalisé par R.J. McAulay et M.L Malpass [51], qui ont formulé le problème de la soustraction spectrale en utilisant l'approche du maximum de vraisemblance. Ce travail avait pour objectif d'éliminer ou de réduire le bruit musical qui subsiste après l'application d'une soustraction spec-

[1] Par exemple, dans le cadre de la réduction du bruit pour la restauration d'enregistrements musicaux, O. Cappé utilise plusieurs variantes de la soustraction spectrale [10].

trale classique. R.J. McAulay et M.L Malpass considèrent que l'appartenance d'une zone du signal de parole à une catégorie n'est pas exclusive. Pour prendre en compte cette propriété, le filtre résultant de la somme de tous les filtres pondérés avec leurs probabilités *a posteriori* est appliqué à la zone en question.

J.S. Lim et A.V. Oppenheim [47] ont proposé une technique qui consiste à modéliser le signal de parole par une succession de modèles AR[2]. Les modèles AR et le signal propre sont tous les deux estimés en utilisant le signal bruité et la densité spectrale du bruit, supposée connue *a priori*. Pour cela, l'approche du MAP (*Maximum a Posteriori*) a été utilisée. La maximisation d'une fonction de vraisemblance appropriée est réalisée d'une manière itérative : 1) par rapport aux paramètres des modèles AR en supposant que le signal dont on dispose est propre, 2) par rapport au signal propre en utilisant les modèles AR et l'estimation de la densité spectrale du bruit. Il est évident que cette technique possède beaucoup plus de variables inconnues (signal propre et modèles AR) que de variables connues (signal bruité). Ceci conduit à de grandes variances dans les estimateurs du signal et des modèles. Pour pallier ce problème, Emphraim et al. proposent d'estimer les modèles AR représentant le signal propre à partir de signaux d'apprentissage propres au lieu de les estimer directement sur le signal bruité.

Afin d'estimer les trames de parole propre maximisant la densité de probabilité *a posteriori*, Ephraim et al. [18] utilisent une procédure itérative fondée sur l'algorithme EM. Ce processus itératif est très dépendant de l'initialisation. Lorque l'initialisation est mauvaise, ce processus converge vers un maximum local qui peut être très loin de la solution optimale. Cela est toujours le cas avec des signaux très bruités. Logan et Robinson [50] proposent d'utiliser une technique de combinaison de modèles dans le cadre des MMC autorégressifs (MMCAR) permettant de mieux initialiser ce processus. L'initialisation est obtenue par décodage du signal bruité en utilisant le système de reconnaissance fondé sur des MMCAR adaptés au bruit de test. Cependant, il bien connu qu'un système de reconnaissance utilisant des MMC fondés sur le cepstre et ses dérivées est plus efficace qu'un système de reconnaissance utilisant des MMCAR, surtout dans le cas d'applications avec de grands vocabulaires. Dans ce chapitre, nous proposons une extension de cette dernière approche en utilisant un système de reconnaissance fondée sur le cepstre et ses dérivées pour l'initialisation.

Nous proposons d'utiliser deux ensembles de modèles acoustiques : le premier est un MMCDC fondé sur le cepstre utilisé pour l'initialisation du processus itératif ; le second est un MMCAR utilisé pour estimer la succession de filtres à appliquer aux trames constituant le signal bruité. Pour une trame du signal bruité, nous cherchons la gaussienne cepstrale qui a la plus grande probabilité *a posteriori*. Cette recherche est faite par décodage du signal bruité en utilisant les modèles cepstraux adaptés au bruit. Le décodage fournit l'alignement trame/gaussienne-cepstrale, où chaque gaussienne cepstrale correspond à une gaussienne autorégressive dans les MMCAR. Le filtre optimal est donc estimé en utilisant cette gaussienne autorégressive et la gaussienne autorégressive correspondant au bruit. Les MMC cepstraux et les MMCAR

[2] Le signal est découpé en une suite de zones stables (stationnaires). Chaque zone est modélisée par un modèle AR, autrement dit chaque zone est modélisée par sa densité spectrale lissée.

sont entraînés de sorte à avoir une bijection entre les deux ensembles de modèles au niveau de la gaussienne : à une gaussienne dans les MMC cepstraux correspond une et une seule gaussienne dans les MMCAR. Pour cela, nous estimons en premier les MMC cepstraux et nous utilisons les statistiques correspondant à la dernière itération pour estimer les paramètres des MMCAR.

Dans les sections suivantes nous décrivons la mise en oeuvre de cette approche dont l'objectif est l'amélioration de la qualité du signal de parole bruité. Des expériences avec différents types de bruits sont présentées afin de montrer l'apport de cette approche au niveau de l'amélioration de la qualité des signaux de parole dégradés par la présence de bruits ambiants. Les expériences de reconnaissance avec le signal débruité sont présentées dans le but de montrer qu'il n'y a pas de perte d'intelligibilité du point de vue du système de reconnaissance et non dans le but d'améliorer le taux de reconnaissance. En effet, la théorie prédit qu'aucun gain significatif en terme de taux de reconnaissance n'est à espérer par débruitage par rapport aux techniques de bruitage si les modèles mis en jeu sont de même qualité (voir introduction du chapitre 5). L'objectif est l'augmentation de la qualité du signal sans perte d'intelligibilité.

## 6.2 Processus de débruitage

### 6.2.1 Cas d'une gaussienne AR

Pour des raisons de clarté, nous commençons par le cas d'une observation générée par une seule gaussienne. Soit $y$ un signal aléatoire autorégressif (AR) d'ordre $p$. Il existe donc $p$ coefficients $a_{i,1\leq i\leq p}$ tels que :

$$y_i = -\sum_{k=1}^{p} a_k y_{i-k} + e_i. \tag{6.1}$$

où $e$ est un processus aléatoire de moyenne zéro et de variance $\sigma_y^2$ dont les réalisations sont indépendantes et identiquement distribuées. Supposons que $y$ soit de durée limitée $K$, $y$ peut donc être compris comme un vecteur de dimension $K$ : $y \in \mathbf{R}^K$, où $\mathbf{R}$ désigne l'ensemble des réels. L'équation précédente peut donc s'écrire sous forme matricielle comme suit :

$$A_y y = e \tag{6.2}$$

avec $y = (y_0, \ldots, y_{K-1})'$ et $e = (e_0, \ldots, e_{K-1})'$ ; la matrice $A_y$ est définie comme suit :

$$A_y = \begin{bmatrix} a_0 & 0 & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & 0 \\ \\ a_1 & a_0 & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ \\ a_2 & a_1 & a_0 & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ . & . & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ a_{p-1} & a_{p-2} & a_{p-3} & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ a_p & a_{p-1} & a_{p-2} & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ 0 & a_p & a_{p-1} & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & . \\ . & 0 & a_p & \ldots\ldots & \ldots\ldots & a_0 & \ldots\ldots & . \\ . & . & \ldots\ldots & \ldots\ldots & \ldots\ldots & a_1 & a_0 & 0 \\ 0 & 0 & \ldots\ldots & a_p & \ldots\ldots & a_2 & a_1 & a_0 \end{bmatrix} \tag{6.3}$$

avec $a_0 = 1$ et les $a_{i,1 \leq i \leq p}$ constituant les coefficients de la prédiction linéaire. Afin de calculer la matrice de covariance de $y$, calculons la matrice d'autocorrélation de $e$ :

$$R_e = E[ee'] = E[A_y y y'(A_y)'] = A_y R_y (A_y)', \tag{6.4}$$

où $E[\cdot]$ désigne l'espérance. Comme $R_e = \sigma_y^2 I$, $I$ désignant la matrice identité, l'équation précédente devient :

$$\sigma_y^2 I = A_y R_y (A_y)', \tag{6.5}$$

puisque $det(A_y)$ est non nul, $A_y$ est inversible et la matrice d'autocorrélation (ou de covariance) s'écrit donc :

$$R_y = \sigma_y^2((A_y)^{'} A_y)^{-1} \tag{6.6}$$

$y$ peut être compris comme une variable aléatoire de moyenne nulle et de matrice de covariance $S_y = R_y$. Si on suppose que les variables aléatoires $e_i$ sont gaussiennes dans $\mathbf{R}$ alors $e$ l'est aussi dans $\mathbf{R}^K$. Puisque $y$ est une transformation linéaire de $e$ et puisque $e$ est gaussienne dans $\mathbf{R}^K$ alors $y$ est aussi gaussienne dans $\mathbf{R}^K$. La densité de probabilité de $y$ est donc une gaussienne de moyenne zéro et de matrice de covariance $S_y = \sigma_y^2((A_y)^{'} A_y)^{-1}$. Cette gaussienne s'appelle *gaussienne autorégressive* et elle s'exprime comme suit :

$$f(y) = \frac{1}{(2\pi)^{\frac{K}{2}} \, |S_y|^{\frac{1}{2}}} exp\{-\frac{1}{2}(y^{'} S_y^{-1} y)\} \tag{6.7}$$

où $y^{'}$ désigne la matrice transposée de $y$, et $S_y$ est donnée comme suit:

$$S_y = \sigma_y^2 (A_y^{'} A_y)^{-1} \tag{6.8}$$

où $\sigma_y^2$ est la variance de la source. Si on imagine que le signal $y$ est le résultat de l'excitation d'un filtre tout-pôle (filtre AR) par un signal $e$, alors $\sigma_y^2$ représenterait l'énergie ou la variance du signal aléatoire $e(t)$ : $\sigma_y^2 = E[e_i^2]$.

Notons $\lambda_y$ les paramètres du modèle AR correspondant au signal propre $y$ : $\lambda_y = (\sigma_y^2, A_y)$. Soit $z$ un signal résultant de la génération d'un signal propre $y$ par $\lambda_y$, bruité avec un signal pouvant être modélisé par un modèle AR. Notons $\lambda_n$ les paramètres du modèle AR correspondant au bruit : $\lambda_n = (\sigma_n^2, A_n)$. Notons $q$ l'ordre du modèle AR correspondant au bruit. De la même manière que pour la source du signal propre, $\sigma_n^2$ désigne l'énergie de l'erreur résultant de la modélisation AR du bruit ; $A_n$ est une matrice triangulaire $K \times K$ de Toeplitz, dans laquelle les $q+1$ premiers éléments de la première colonne constituent les coefficients de la prédiction de la source du bruit :

$$A_n = \begin{bmatrix} b_0 & 0 & ..... & ..... & ..... & ..... & ..... & 0 \\ \\ b_1 & b_0 & ..... & ..... & ..... & ..... & ..... & . \\ \\ b_2 & b_1 & b_0 & ..... & ..... & ..... & ..... & . \\ . & . & . & ..... & ..... & ..... & ..... & . \\ b_{q-1} & b_{q-2} & b_{q-3} & ..... & ..... & ..... & ..... & . \\ b_q & b_{q-1} & b_{q-2} & ..... & ..... & ..... & ..... & . \\ 0 & b_q & b_{q-1} & ..... & ..... & ..... & ..... & . \\ . & . & b_q & ..... & ..... & b_0 & ..... & . \\ . & . & ..... & ..... & ..... & b_1 & b_0 & 0 \\ 0 & 0 & ..... & b_q & ..... & b_2 & b_1 & b_0 \end{bmatrix} \tag{6.9}$$

Dans le cas d'un bruit gaussien avec une densité de probabilité relativement plate, des petites valeurs de $q$ peuvent être utilisées. La densité de probabilité du signal de bruit est :

$$g(n) = \frac{1}{(2\pi)^{\frac{K}{2}} |S_n|^{\frac{1}{2}}} exp\{-\frac{1}{2}(n' S_n^{-1} n)\} \tag{6.10}$$

avec $n \in \mathbf{R}^K$ et $S_n = \sigma_n^2 (A_n' A_n)^{-1}$. Le signal bruité $z$ peut s'écrire : $z = y + n$. Le problème du débruitage consiste à estimer $y$ étant donné $z$ et des statistiques concernant $y$ et $n$. L'approche que l'on va utiliser consiste à estimer $y$ en utilisant l'approche du *Maximum A Posteriori* (MAP) :

$$\hat{y} = \underset{y}{\operatorname{argmax}} \, \mathbf{log}\{h(y,z)\} \tag{6.11}$$

où $h(y,z)$ désigne la densité de probabilité conjointe de $y$ et $z$. Soit $k(z|y)$ la densité de probabilité de $z$ connaissant $y$

$$h(y,z) = f(y)k(z|y) = f(y)g(z-y) \tag{6.12}$$

en remplaçant $f(y)$ et $g(z-y)$ par leurs valeurs, on obtient :

$$\mathbf{log}\{h(y,z)\} = y' S_y^{-1} y + (z-y)' S_n^{-1} (z-y) + cte \tag{6.13}$$

où $cte$ est une quantité indépendante de $y$. Donc sa dérivée par rapport à $y$ est nulle. La dérivée de $\mathbf{log}\{h(y,z)\}$ par rapport à $y$ est donnée comme suit:

$$2S_y^{-1} y + 2S_n^{-1} y - 2S_n^{-1} z \tag{6.14}$$

en égalisant cette quantité avec zéro, on obtient :

$$\hat{y} = (S_n S_y^{-1} + I)^{-1} z \tag{6.15}$$

La relation précédente peut s'écrire dans le domaine fréquentiel comme suit [31] :

$$\hat{y}_\theta = \frac{\Gamma_y(\theta)}{\Gamma_y(\theta) + \Gamma_n(\theta)} z_\theta \tag{6.16}$$

où $\Gamma_y(\theta)$ et $\Gamma_n(\theta)$ représentent les densités spectrales du signal propre et du bruit:

$$\Gamma_y(\theta) = \frac{\sigma_y^2}{|\Psi_y(\theta)|^2} \tag{6.17}$$

$$\Gamma_n(\theta) = \frac{\sigma_n^2}{|\Psi_n(\theta)|^2} \tag{6.18}$$

où $\Psi_y(\theta)$ et $\Psi_n(\theta)$ représentent respectivement les transformées de Fourier des coefficients de prédiction $(a_0, \ldots, a_p)$ et $(b_0, \ldots, b_q)$.

$\Gamma_y(\theta)$ et $\Gamma_n(\theta)$ sont des approximations des densités spectrales de $y$ et $n$, il n'est donc pas nécessaire de passer par les coefficients de prédiction pour les estimer. Il suffit de déterminer les fonctions d'autocorrélation correspondant à $y$ et $n$ puis de calculer leurs transformées de Fourier, ce qui permet d'aboutir aux densités spectrales de $y$ et $n$.

Une première remarque que l'on peut faire est que le filtre obtenu avec le critère du maximum de vraisemblance est réel et ne modifie pas la phase du signal. Cela est tout-à-fait logique dans la mesure où le bruit affectant le signal est non corrélé avec ce dernier. Il est important de noter que le filtre obtenu avec le critère du maximum de vraisemblance est tout simplement celui qui est connu sous le nom de filtre de Wiener. Dans la littérature, ce filtre est généralement obtenu en utilisant le critère des moindres carrés. Pour une gaussienne AR modélisant le bruit et une gaussienne AR modélisant la cible, le filtre défini par l'équation 6.16 est appelé filtre de Wiener associé aux deux gaussiennes AR (le bruit et la cible). Il faut remarquer que ce filtrage appliqué à un signal $z$ donné tente de le rapprocher le plus possible de la cible $y$ tout en respectant la contrainte $z = y + n$. Il est donc très important de viser la bonne cible.

Il est important de noter que le filtrage de Wiener optimise le critère des moindres carrés dans le domaine temporel. Cela ne veut pas dire que ce même critère est optimisé dans le domaine fréquentiel. L'optimisation de ce critère dans le domaine temporel est équivalente à la minimisation, par rapport à $H$ (H est le filtre optimal), dans le domaine fréquentiel de la fonction suivante :

$$e^2(\nu) = \Gamma_y(\nu) + \Gamma_y(\nu)H(\nu) + [\Gamma_y(\nu) + \Gamma n(\nu)]|H(\nu)|^2. \tag{6.19}$$

A titre de vérification, en supposant que le filtre recherché est réel, la minimisation de $e(\nu)$ donne exactement le filtre de Wiener classique ; l'annexe C donne une démonstration de ce résultat. Le filtre minimisant l'erreur des moindres carrés dans le domaine fréquentiel, entre la densité spectrale de la cible (signal propre) et la densité spectrale du signal bruité est donné par la formule suivante :

$$H(\nu) = \sqrt{\frac{\Gamma_y(\nu)}{\Gamma_y(\nu) + \Gamma_n(\nu)}}. \tag{6.20}$$

### 6.2.2 Cas d'un MMCAR

Considérons maintenant le cas plus général d'une observation générée par un MMC autorégressif (MMCAR). Un MMC autorégressif est un MMC (chapitre 2), dont les densités de probabilité associées aux états sont des mélanges de gaussiennes autorégressives (paragraphe 6.2).

Soit $f_\lambda$ la fonction de densité de probabilité associée à un MMC autorégressif, où $\lambda$ désigne les paramètres définissant le modèle . Supposons que le MMC AR soit formé de $N$ états ; à chaque état est associé un mélange de gaussiennes AR. Pour des raisons de clarté, nous avons choisi d'utiliser le même nombre de gaussiennes AR pour tous les états, soit $M$. Soit $y = y_{t,t=0,\dots,T}/y_t \in \mathbf{R}^k$ une séquence de vecteurs générés par le modèle. Soit $x = x_{t,t=0,\dots,T}$ la succession d'états qui a généré l'observation $y$ : $y_t$ a été généré par $x_t$. Soit $h = h_{t,t=0,\dots,T}$ la succession de gaussiennes AR, à l'intérieur de la succession d'états $x$, qui a généré $y$ : $y_t$ a été généré par $h_t$. La densité de probabilité de $y$ est donnée comme suit :

$$f_\lambda(y) = \sum_x \sum_h p_\lambda(x) p_\lambda(h|x) p_\lambda(y|h,x) \tag{6.21}$$

où $p_\lambda(x)$ désigne la probabilité d'observer la séquence d'états $x$, $p_\lambda(h|x)$ la probabilité d'observer la séquence de gaussiennes $h$ sachant que la séquence d'états observés est $x$ et $p_\lambda(y|h,x)$ est la probabilité d'observer $y$ connaissant $x$ et $h$ :

$$p_\lambda(x) = \prod_{t=0}^{T} a_{x_{t-1}x_t} \tag{6.22}$$

où $a_{x_{t-1}x_t}$ est la probabilité de transition de l'état $x_{t-1}$ à l'état $x_t$ avec $a_{x_{-1}x_0}$ désignant la probabilité que $x_0$ soit le premier état de la séquence d'états.

$$p_\lambda(h|x) = \prod_{t=0}^{T} p_\lambda(h_t|x_t) = \prod_{t=0}^{T} \alpha_{h_t|x_t} \tag{6.23}$$

où $\alpha_{h_t|x_t}$ désigne le poids de la gaussienne $h_t$ dans le mélange de gaussiennes associé à l'état $x_t$.

$$p_\lambda(y|h,x) = \prod_{t=0}^{T} p_\lambda(y_t|h_t,x_t) = \prod_{t=0}^{T} b_\lambda(y_t|h_t,x_t) \tag{6.24}$$

$$b(y_t|h_t=\gamma, x_t=\theta) = \frac{1}{(2\pi)^{\frac{K}{2}} \left|S_{\gamma|\theta}\right|^{\frac{1}{2}}} exp\{-\frac{1}{2}(y_t' S_{\gamma|\theta}^{-1} y_t)\} \tag{6.25}$$

Les définitions des paramètres sont les mêmes que celles du paragraphe 6.2 : $S_{\gamma|\theta} = \sigma^2_{\gamma|\theta}((A_{\gamma|\theta})' A_{\gamma|\theta})^{-1}$, où $\sigma^2_{\gamma|\theta}$ désigne l'énergie du résiduel au sens d'une analyse LPC comme définie dans le chapitre 2. $A_y$ est une matrice $K \times K$ de Toeplitz triangulaire, dans laquelle les $p+1$ premiers éléments de la première colonne constituent les coefficients du processus AR : $a_{\gamma|\theta} = (a^0_{\gamma|\theta}, a^1_{\gamma|\theta}, \ldots, a^p_{\gamma|\theta})$ et $a^0_{\gamma|\theta} = 1$.

La première étape dans ce processus de débruitage est d'estimer l'ensemble des paramètres $\lambda = (\pi, a, \alpha, A)$ avec $\pi = \{\pi_\theta\}$, $a = \{a_{\theta_1\theta_2}\}$, $\alpha = \{\alpha_{\gamma|\theta}\}$ et $A = \{S_{\gamma|\theta}\}$ avec $\theta_1, \theta_2 \in \{1, \ldots, N\}$ et $\gamma \in \{1, \ldots, M\}$, étant donnée une séquence d'apprentissage $y$ de signal de parole. Le critère d'estimation choisi est celui du maximum de vraisemblance. Les paramètres qui maximisent la vraisemblance sont obtenus en utilisant l'algorithme de réestimation de Baum [40].

Comme dans le paragraphe 6.2, le modèle de bruit est une simple gaussienne AR, notons $f_{\lambda_n}$ sa densité de probabilité. La densité de probabilité d'une succession $n = \{n_t, t = 0, \ldots, T\}$, $n_t \in \mathbf{R}^K$, de vecteurs de bruit générés par ce modèle est :

$$g_{\lambda_n}(n) = \prod_{t=0}^{T} \frac{1}{(2\pi)^{\frac{K}{2}} \, |S_n|^{\frac{1}{2}}} exp\{-\frac{1}{2}(n_t^{'} S_n^{-1} n_t)\} \tag{6.26}$$

Pour estimer le signal propre à partir du signal bruité et des connaissances *a priori* sur le bruit, Ephraim [18] utilise l'algorithme EM pour réaliser une estimation de type MAP. Soit $z$ une séquence de $T+1$ vecteurs de $\mathbf{R}^K$ représentant une séquence de segments de parole bruitée. L'approche MAP consiste à trouver le signal $y = y_t, t = 0, \ldots, T$, $y_t \in \mathbf{R}^K$ qui maximise $p_\lambda(y|z)$. Pour maximiser cette quantité, on utilise la procédure itérative connue sous le nom de l'algorithme EM : soit $y(k) = y_t(k), t = 0, \ldots, T$, $y_t(k) \in \mathbf{R}^K$, une estimation courante du signal de parole propre. Une nouvelle estimation $y(k+1) = y_t(k+1), t = 0, \ldots, T$, $y_t(k+1) \in \mathbf{R}^K$, telle que $p(y(k)|z) \leq p(y(k+1)|z)$ est celle qui maximise la fonction auxiliaire :

$$\Phi(y(k+1)) = \sum_{x,h} p_\lambda(x, h|y(k)) log\,(p_\lambda(x, h, y(k+1)|z)) \tag{6.27}$$

La maximisation de cette quantité s'obtient en dérivant par rapport à $y(k+1)$ et en égalisant avec zéro. Nous obtenons la relation suivante :

$$y_t(k+1) = \left[\sum_{\beta,\gamma} p_t(\beta, \gamma|y(k)) H^{-1}_{\gamma|\beta}\right]^{-1} z_t, \tag{6.28}$$

où $p_t(\beta, \gamma|y(k))$ est la probabilité que la trame à l'instant $t$ soit générée par la gaussienne $\gamma$ de l'état $\beta$. Ces probabilités sont généralement calculées en utilisant la procédure “avant-arrière” [40]. $H_{\gamma|\beta}$ est le filtre de Wiener associé à la gaussienne AR $\gamma$ de l'état $\beta$ et à la gaussienne AR modélisant le bruit. Cette formule est très intuitive ; en effet, le filtre appliqué à la trame $t$ n'est rien d'autre que la somme de tous les filtres associés à toutes les gaussiennes du MMC, pondérés avec les probabilités *a posteriori* que ces gaussiennes génèrent la trame à l'instant $t$. L'initialisation $y(0)$ de ce procédé d'estimation est le signal bruité $z$.

Une approximation généralement utilisée est celle qui consiste à remplacer le filtre de la formule 6.28 par le filtre associé à la gaussienne qui a la plus grande probabilité *a posteriori*. On considère dans ce cas qu'une trame est générée exclusivement par une seule gaussienne du MMC. L'association d'une trame à un état est déterminée en utilisant l'algorithme de Viterbi ; l'association de cette trame à une gaussienne dans cet état est déterminée en choisissant la gaussienne avec la plus grande probabilité *a posteriori*. Cette approximation est valable si le nombre de gaussiennes par état est suffisamment grand. Nous appelons ce filtrage : filtrage dépendant de la gaussienne.

Bien sûr, le filtrage peut être fait dans le domaine temporel ou fréquentiel. Cependant, grâce au coût moindre de la FFT, il est préférable de réaliser le filtrage dans le domaine fréquentiel (comme dans le paragraphe 6.2).

Le grand inconvénient de cette approche est le fait de prendre comme estimation initiale le signal bruité lui même. De cette manière, nous utilisons des modèles (MMC) estimés sur des signaux propres pour décoder des signaux bruités. Cela implique que pour des petites valeurs de RSB, les associations entre les trames du signal bruité et les gaussiennes sont en majorité fausses. Puisque le filtrage de Wiener dépendant de la gaussienne consiste tout simplement à modifier le signal bruité pour qu'il s'approche le plus possible de la cible, ce processus itératif converge vers une mauvaise solution. En effet, l'algorithme EM converge vers un maximum local qui dépend fortement de l'initialisation. Si l'initialisation est mauvaise, on risque de converger vers un maximum local qui est loin d'être la solution optimale du problème. Un autre inconvénient de cette approche est que le modèle de bruit est fixe et ne s'améliore pas d'une itération à l'autre.

### 6.2.3 Choix de l'initialisation de l'algorithme EM

Pour surmonter le problème lié à l'initialisation de l'algorithme EM, nous procédons de la manière suivante : nous utilisons un système de reconnaissance fondé sur le cepstre classique (MFCC par exemple) pour l'alignement trames/gaussiennes. Ce système de reconnaissance nous permet de faire l'association entre les gaussiennes des MMC et les trames du signal bruité. A chaque gaussienne dans le domaine cepstral correspond une gaussienne AR. Cette association entre gaussiennes cepstrales et gaussiennes AR est faite au moment de l'apprentissage. Pour bien initialiser le processus d'estimation, surtout pour des RSB faibles, le système permettant l'alignement trames/gaussiennes est celui qui résulte de la combinaison des MMC modélisant la parole propre avec le bruit affectant le signal bruité (voir le chapitre sur la combinaison). Posons $\lambda_{cp}$ l'ensemble des paramètres des MMCs qui résulte de la combinaison. Soit $z$ une séquence de $T+1$ vecteurs de $\mathbf{R}^K$ représentant une séquence de segments de parole bruitée. Le décodage de $z$ avec le système de reconnaissance $\lambda_{cp}$ permet de calculer $p_t(\beta_{cp}, \gamma_{cp}|z_{cp}, \lambda_{cp})$, la probabilité que la trame (dans le domaine cepstral) à l'instant $t$ soit générée par la gaussienne cepstrale $\gamma_{cp}$ de l'état $\beta_{cp}$. Soient $\beta_{AR}$ et $\gamma_{AR}$ l'état et la gaussienne correspondant respectivement à $\beta_{cp}$ et $\gamma_{cp}$. Le signal propre est estimé dans ce cas comme suit :

$$\hat{y}_t = \left[ \sum_{\beta_{AR}, \gamma_{AR}} p_t(\beta_{cp}, \gamma_{cp}|z_{cp}, \lambda_{cp}) H^{-1}_{\gamma_{AR}|\beta_{AR}} \right]^{-1} z_t \tag{6.29}$$

où $H_{\gamma_{AR}|\beta_{AR}}$ est le filtre de Wiener[3] associé à la gaussienne AR $\gamma_{AR}$ de l'état $\beta_{AR}$ et la gaussienne AR modélisant le bruit. $p_t(\beta_{cp}, \gamma_{cp}|z_{cp}, \lambda_{cp})$ permet de calculer le poids avec lequel le filtre $H_{\gamma_{AR}|\beta_{AR}}$ doit être appliqué à la trame $z_t$. Dans le cas où seul le filtre avec la plus grande probabilité *a posteriori* ($p_t(\beta_{cp}, \gamma_{cp}|z_{cp}, \lambda_{cp})$) est appliqué, cette probabilité est approximée par 1. Le meilleur système de débruitage (dans ce contexte) est celui qui offre le modèle le plus robuste pour le calcul de ces probabilités *a posteriori*. Pour le calcul de ces probabilités *a posteriori* nous avons choisi le modèle obtenu par la combinaison parallèle de données.

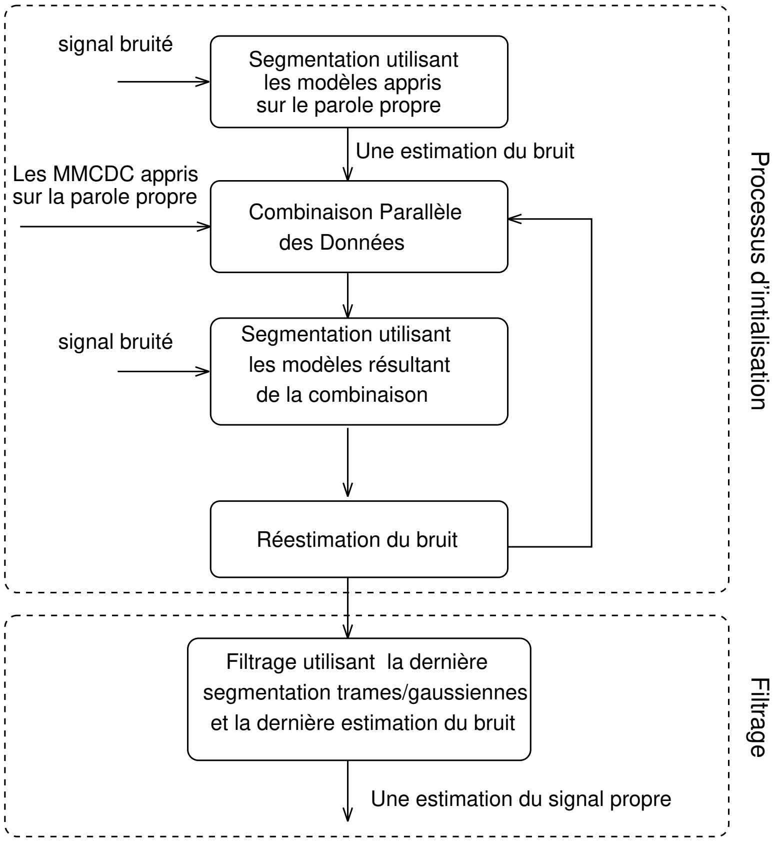



**Figure 6.1 :** *Débruitage du signal en utilisant la combinaison parallèle de données (CPD) et le filtrage dépendant de la gaussienne utilisant les MMCAR (Modèles de Markov Cachés à Distribution AutoRégressives).*

Dans l'approche présentée dans le paragraphe précédent le modèle du bruit est supposé correct et fixe durant tout le processus d'estimation. Il est cependant préférable de réestimer ce modèle en utilisant les segments correspondant aux silences (bruit) dans le signal bruité.

L'estimation du signal propre à partir du signal bruité se fait en deux étapes : la

[3] Dans la théorie, c'est le filtre de Wiener qui est estimé en utilisant les gaussiennes autorégressives du signal propre et du bruit. Dans la pratique n'importe quel filtre (choisi raisonnablement) peut être appliqué, par exemple le filtre de l'équation 6.20.

première consiste à utiliser la combinaison parallèle de données (CPD)[4] qui permet d'aboutir à des modèles acoustiques représentant la parole bruitée. Nous utilisons ce modèle combiné pour faire l'alignement trames/gaussiennes du signal bruité ; ensuite nous réestimons le modèle de bruit grâce à cet alignement. Nous reprenons ce processus à partir du début jusqu'à la convergence (jusqu'à ce que le modèle de bruit ne change plus). La deuxième étape consiste à utiliser le dernier alignement trames/gaussiennes pour estimer et appliquer les filtres dépendant de la gaussienne correspondant à chaque trame du signal bruité. Ce qui permet d'aboutir à une estimation du signal propre correspondant au signal bruité. Le filtrage dépendant de la gaussienne est appliqué une seule fois.

### 6.2.4 Estimation des paramètres d'un MMCAR

Supposons dans un premier temps que l'on dispose d'un certain nombre d'observations générées par un processus AR dont on ne connaît pas les paramètres $\lambda$. Comment est-t-il possible de déduire les paramètres du modèle AR générateur à partir des données générées ? Une façon de faire est d'utiliser le critère du Maximum de Vraisemblance. Soit $Y = (y_0, \ldots, y_T)$ une séquence de vecteurs appartenant à $\mathbf{R}^K$ générés par le processus AR de paramètres $\lambda$. Le critère du maximum de vraisemblance consiste à trouver les paramètres du modèle AR qui maximisent la probabilité que la séquence $Y$ soit générée par ce modèle.

$$\hat{\lambda} = \arg\max_{\lambda} \mathbf{log} f(Y|\lambda). \tag{6.30}$$

En supposant que les $y_{t,(0 \leq t \leq T)}$ sont indépendants et identiquement distribués, l'équation précédente s'écrirait :

$$\hat{\lambda} = \arg\max_{\lambda} \sum_{t=0}^{T} \mathbf{log} f(y_t|\lambda). \tag{6.31}$$

Une valeur approximative de $\lambda$ optimale est l'ensemble des paramètres du modèle $AR$ dont la fonction d'autocorrélation est [66] :

$$r(m) = \frac{1}{T+1} \sum_{t=0}^{T} \frac{1}{K} \sum_{k=0}^{K-|m|-1} y_t(k) y_t(k+|m|), \quad m = -p, \ldots, p \tag{6.32}$$

Considérons maintenant le cas plus général d'une observation générée par un MMCAR. Soit $Y = (y_0, \ldots, y_T)$ une séquence de vecteurs appartenant à $\mathbf{R}^K$ et censés être générés par le processus stochastique modélisé par un MMCAR de paramètres $\lambda_{AR}$. Les coefficients d'autocorrélation définissant la gaussienne AR $\gamma_{AR}$ de l'état $\beta_{AR}$ dans les MMC AutoRégressifs sont déterminés comme suit :

$$r'_{\gamma_{AR}|\beta_{AR}}(m) = \frac{1}{\sum_{t=0}^{T} p_t(\gamma_{cp}, \beta_{cp}|Y, \lambda_{cp})} \sum_{t=0}^{T} p_t(\gamma_{cp}, \beta_{cp}|Y, \lambda_{cp}) r_t(m) \tag{6.33}$$

[4] La CPD est appliquée sur les modèles acoustiques appris sur les signaux d'apprentissage propres et sur une estimation du bruit. La première estimation du bruit peut être tout simplement les trames correspondant au silence dans la segmentation du signal bruité par le système de reconnaissance utilisant les modèles appris sur les signaux d'apprentissage propres.

où

$$r_t(m) = \frac{1}{K} \sum_{k=0}^{K-|m|-1} y_t(k) y_t(k+|m|), \quad m = -p, \ldots, p \tag{6.34}$$

où $p$ désigne l'ordre du processus AR. $\beta_{cp}$, $\gamma_{cp}$ et $\lambda_{cp}$ sont définis comme dans 6.2.3.

Les autres paramètres du MMCAR (probabilités initiales, probabilités de transitions) sont estimés de la même manière que dans le cas des MMCDC. Dans ce travail seules les probabilités d'émission des MMCAR sont utilisées dans la mesure où ce sont les MMCDC leurs correspondant qui sont utilisés pour le décodage.

## 6.3 Résultats expérimentaux

Comme dans la section précédente, nous allons montrer dans cette section que notre approche de débruitage donne de très bons résultats du point de vue de la perception, à condition que le système de reconnaissance puisse reconnaître le contenu linguistique du signal de parole bruité. Nous utilisons un système de reconnaissance de la parole fondé sur des MMCDC modélisant l'évolution cepstrale des signaux de parole. Ce système a pour objectif de réaliser l'alignement trames/gaussiennes du signal de parole bruité. Dans ces expériences, le débruitage est réalisé dans le cadre de l'application Mask [42].

Les vecteurs paramétriques caractérisant une trame sont composés de 13 coefficients *cepstraux* avec leurs dérivées première et seconde et sont estimés toutes les 10ms. 608 phonèmes dépendant du contexte sont modélisés. Chaque phonème dépendant du contexte est un Modèle de Markov Caché à Densité Continue (MMCDC) avec un mélange de gaussiennes comme densités d'observation (typiquement 20 gaussiennes).

L'alignement trames/gaussiennes du signal de parole bruité est réalisé par le MMCDC obtenu en appliquant la combinaison parallèle de données (CPD : chapitre 5).

Le système fondé sur des MMC à Densités AutoRégressives (MMCAR), permettant d'estimer les filtres à appliquer sur les trames du signal bruité, est estimé sur le même corpus que le système cepstral permettant de réaliser l'alignement trames/gaussiennes. Les vecteurs paramétriques caractérisant une trame sont composés de 16 coefficients d'autocorrélation[5]. Les coefficients d'autocorrélation du signal du bruit affectant le signal de parole correspondant à une phrase donnée, sont estimés en utilisant les trames affectées au modèle de silence en phase d'alignement. Les coefficients d'autocorrélation associés à une gaussienne autorégressive dans les MMCAR sont obtenus en moyennant les coefficients d'autocorrélation de toutes les trames appartenant à cette gaussienne. Cette appartenance provient de l'alignement trames/gaussiennes des données d'apprentissage en utilisant les MMCDC cepstraux. Les autres paramètres du MMCAR (probabilités des transitions, probabilités initiales, les poids des gaussiennes à l'intérieur des états) sont les mêmes que ceux des

[5] Nous n'avons pas fait d'étude permettant de choisir le nombre de coefficient d'autocorrélation optimale.

MMCDC cepstraux. Il est important de noter que ce système est indépendant du sexe : les mêmes systèmes (cepstral et autorégressif) sont utilisés pour débruiter un signal de parole indépendamment du sexe du locuteur.

La stratégie de débruitage proposée dans ce chapitre est appliquée à des signaux de parole dégradés par un bruit additif. Nous présentons des résultats expérimentaux utilisant trois types de bruits provenant de la base de données NOISEX-92 : bruit blanc, bruit d'hélicoptère (Lynx) et bruit d'avion à réacteur (F16 jet). Après filtrage, chaque trame (30ms) est pondérée par la fenêtre de Hamming afin d'éliminer les effets de bord. Le signal restauré est constitué en utilisant la technique de *"Overlap-Add"* [68] sur les segments résultant du filtrage, suivi du fenêtrage par Hanning.

### 6.3.1 Amélioration de la qualité du signal de parole

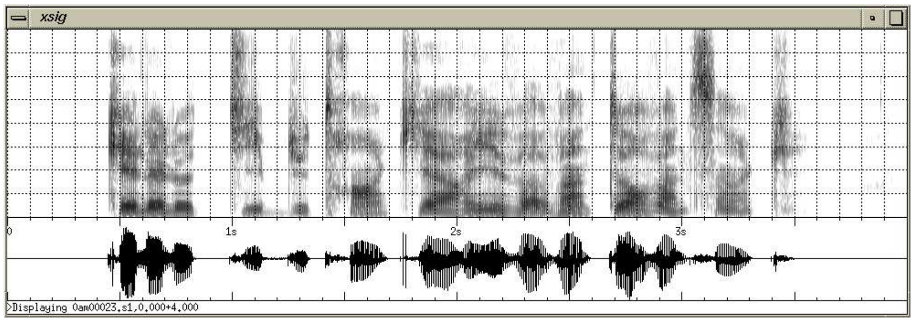


**Figure 6.2 :** *Dans cette figure nous représentons le signal ainsi que le spectrogramme de la phrase prononcé dans une ambiance calme (RSB=35dB) : "quel est le type du train qui arrive à 20 heures 25".*

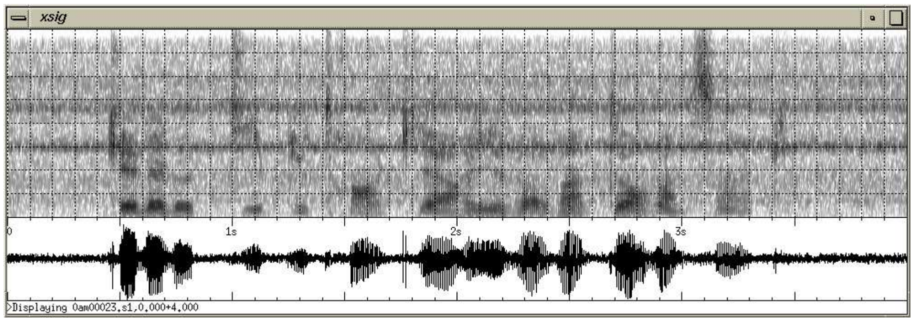


**Figure 6.3 :** *Signal de parole bruité (5.7dB) obtenu en additionnant le bruit d'avion à réaction (F16 Jet) au signal de la Figure 6.2.*

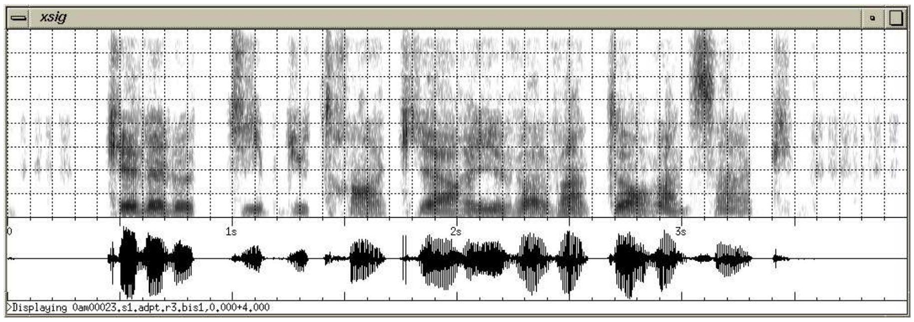



**Figure 6.4 :** *Signal de parole obtenu par débruitage du signal de la Figure 6.3. Le débruitage est réalisé par filtrage de Wiener dépendant de la gaussienne.*

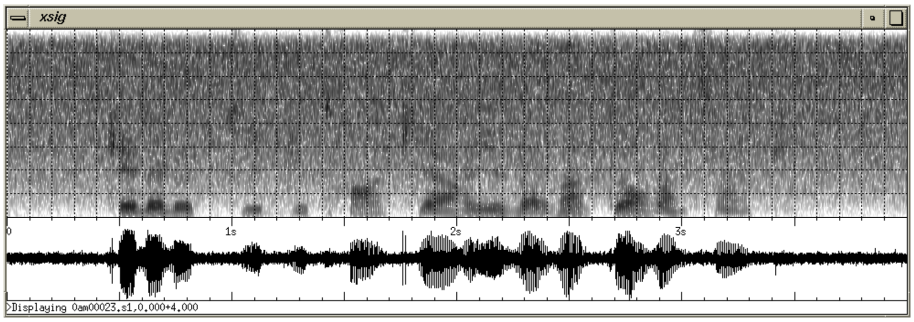



**Figure 6.5 :** *Signal de parole bruité (1dB) obtenu en additionnant le bruit blanc au signal de la Figure 6.4.*

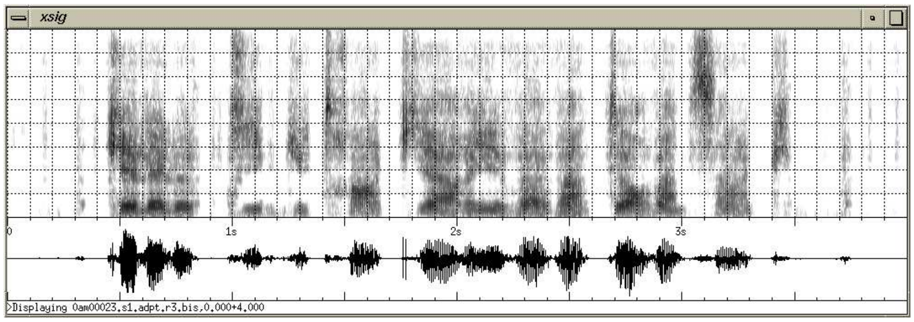



**Figure 6.6 :** *Signal de la Figure 6.5. Le débruitage est réalisé par filtrage de Wiener dépendant de la gaussienne.*

Le signal ainsi que le spectrogramme d'un signal propre sont présentés dans la Figure 6.2. Des versions bruitées de ce signal sont obtenues en additionnant le bruit d'avion à réaction (F16 Jet) et le bruit blanc, elles sont présentées dans les Figures 6.3 et 6.5 respectivement. Les versions du signal débruité sont présentées dans les Figures 6.4 et 6.6. Visuellement nous constatons une diminution importante du niveau du bruit par rapport au niveau de la parole. En écoutant les signaux obtenus par débruitage nous constatons l'absence du bruit musical qui a préoccupé pendant longtemps les chercheurs dans ce domaine. Cette technique de débruitage permet d'augmenter considérablement le rapport signal sur bruit en introduisant très peu de distorsions.

Puisque la technique de débruitage proposée dans ce chapitre repose principalement sur l'alignement trames/gaussiennes fourni par le système de reconnaissance, il serait intéressant de voir le résultat du débruitage lorqu'il y a des erreurs de reconnaissance. Pour cela prenons la phrase *"Toujours Marseille"* prononcée par un locuteur de sexe masculin, et bruitée avec un bruit blanc. Le rapport signal sur bruit résultant est d'environ 1dB.

Le système de reconnaissance produit un alignement trames/gaussiennes erroné. En fait, il a reconnu *"De Toulon Marseille"* au lieu de *"Toujours Marseille"*. Dans les Figures 6.7, 6.8 et 6.9 nous présentons respectivement le signal propre, le signal bruité et les signaux après débruitage.

En faisant écouter le signal débruité de la Figure 6.9 à un certain nombre de sujets, nous avons constaté qu'il y a une ambiguïté linguistique. En effet, certains sujets pensent avoir reconnu le message *"Toujours Marseille"* et d'autres pensent avoir reconnu *"De Toulon Marseille"*. Dans une application de débruitage, ce type d'ambiguïté peut être levé par le contexte. Ici, par exemple, la question posée par le système au locuteur est *"Quelle est votre destination"*, le locuteur répond, dans la mesure où il avait déjà donné cette information au système, *"Toujours Marseille"*.

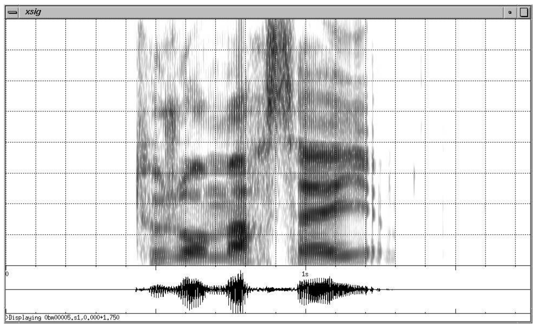


**Figure 6.7 :** *Signal et spectrogramme de la phrase "Toujours Marseille" prononcée dans une ambiance calme (RSB=35dB).*

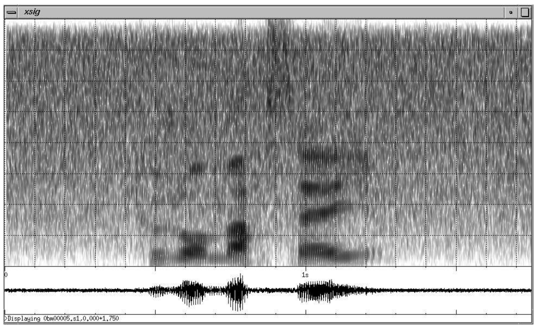


**Figure 6.8 :** *Signal et spectrogramme obtenus en bruitant le signal de la Figure 6.7 avec un bruit blanc. Le rapport signal sur bruit est de l'ordre de 1dB.*

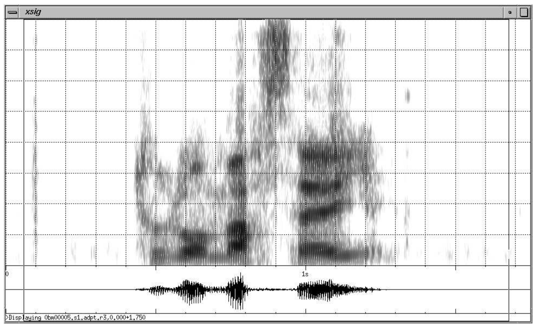


**Figure 6.9** : *Signal et spectrogramme obtenus par application du filtrage de Wiener dépendant de la gaussienne au signal de la Figure 6.8 lorsqu'il y a des erreurs dans l'estimation de la cible.*

### 6.3.2 Résultats de reconnaissance

Une question que l'on peut se poser après avoir débruité les signaux de test est : la reconnaissance sera-t-elle meilleure avec les signaux débruités en utilisant le système de reconnaissance appris sur les données d'apprentissage propres, ou avec les signaux bruités en utilisant les modèles acoustiques obtenus par la combinaison parallèle de données ? En théorie aucun gain significatif n'est à espérer par débruitage par rapport aux techniques de bruitage si les modèles mis en jeu sont de même qualité (voir introduction du chapitre 5), l'objectif premier de ce travail est l'augmentation de la qualité du signal sans perte d'intelligibilité.

Pour répondre expérimentalement à cette question, nous avons réalisé des expériences avec trois types de bruits extraits de la base de données NOISEX : bruit blanc, bruit d'avion à réaction et bruit d'hélicoptère (Lynx). Les résultats expérimentaux sont présentés dans la Table 6.1.

La Table 6.1 montre que si le système est entraîné dans une ambiance calme (35dB) et testé dans une ambiance très bruitée alors les performances sont désastreuses (79.9% d'erreur sur les mots dans notre cas du bruit blanc). Lorsqu'on adapte le système au bruit ambiant en utilisant la combinaison parallèle de données (CPD), le taux d'erreur sur les mots diminue d'une manière très significative : par exemple dans le cas du bruit blanc, la diminution relative est de 81% par rapport au cas où aucune compensation n'est réalisée (de 79.9% à 15.2%). En utilisant le système de reconnaissance dont les modèles acoustiques sont appris sur des signaux propre et en testant avec les signaux obtenus par débruitage en utilisant le filtrage de Wiener dépendant de la gaussienne, le taux d'erreur sur les mots reste relativement constant. En revanche, l'utilisation des signaux de test obtenus par débruitage avec le filtrage dépendant de la gaussienne de la formule 6.20 apporte un gain relatif de 4.6% dans le cas du bruit blanc, 10.3% dans le cas du bruit de F16 Jet et 5.1% dans le cas du bruit de Lynx, par rapport à l'utilisation des signaux bruités avec les modèles acoustiques obtenus par la CPD. Cela montre qu'il est possible, avec le débruitage, d'atteindre des performances légèrement meilleures qu'avec la combinaison parallèle de données à condition d'introduire dans le processus de débruitage de l'information. Dans notre cas, cette information est obtenue grâce à la combinaison parallèle de données. Le gain obtenu par débruitage par rapport au bruitage est dû au fait que les modèles obtenus avec combinaison parallèle de données ne sont pas parfaits à cause de diverses approximations.

Il faut noter que sans l'information obtenue en alignant le signal bruité avec les gaussiennes correspondantes, en utilisant la combinaison parallèle de données, cette technique de débruitage dépendant de la gaussienne, donnerait de très mauvais résultats.

| Type de compensation | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Bruit de F16 Jet (1dB) | 55.4% | 13.6% | 13.9% | 12.2% | 5.9% |
| Bruit de Lynx (5.5dB) | 60.7% | 21.4% | 21.7% | 20.3% | 5.9% |
| Bruit blanc (6.4dB) | 79.9% | 15.2% | 15.2% | 14.5% | 5.9% |

**Table 6.1** *Dans ces expériences les colonnes contiennent les taux d'erreur sur les mots dans les configurations suivantes :*

| *N° de colonne* | *Configuration de test* |
|---|---|
| $1^{ère}$ *colonne* | *Les modèles acoustiques sont appris sur des signaux propres, les données de test sont bruitées. Aucune compensation.* |
| $2^{ième}$ *colonne* | *Les données de test sont bruitées et les modèles acoustiques utilisés pour la reconnaissance sont issus de la CPD.* |
| $3^{ième}$ *colonne* | *Les modèles acoustiques sont appris sur des signaux propres, les signaux de test sont ceux obtenus après débruitage avec le filtrage de Wiener dépendant de la gaussienne.* |
| $4^{ième}$ *colonne,* | *Les modèles acoustiques sont appris sur des signaux propres, les signaux de test sont ceux obtenus après débruitage avec les filtres dépendant de la gaussienne de la formule 6.20.* |
| $5^{ième}$ *colonne,* | *Les données de test et d'apprentissage sont propres.* |

## 6.4 Conclusion

Dans ce chapitre nous avons étudié et expérimenté une approche de débruitage fondée sur le critère du maximum *a posteriori* utilisant des MMC autorégressifs. Cette technique introduite par Ephraim et al. utilise l'algorithme EM qui nécessite dans ce cas une bonne initialisation pour fonctionner correctement. Nous avons proposé une nouvelle stratégie d'initialisation combinant l'utilisation de MMCAR, l'utilisation de MMCDC fondés sur le cepstre et ses dérivées et l'utilisation de la technique de compensation de bruits (CPD) présentée dans le chapitre 5. Cette stratégie d'initialisation rend ce processus de débruitage très efficace même avec des rapports signal sur bruit faibles. Les expériences ont été réalisées en utilisant trois types de bruits provenant de la base de données NOISEX. Nous avons constaté une augmentation considérable de la qualité des signaux débruités, avec peu de distorsion et sans bruit musical. Des tests de reconnaissance avec les signaux débruités montrent qu'il n'y a pas de diminution de l'intelligibilité du signal débruité par rapport au signal bruité du point de vue du système de reconnaissance.

Il est évident que le système de débruitage que nous avons présenté dans ce chapitre dépend de la tâche. Dans notre cas, la tâche est celle de MASK [42]. Cette dépendance de la tâche vient du fait que l'alignement trames/gaussiennes est fondé sur la reconnaissance de suites de mots par le système en utilisant un modèle de langage spécifique à l'application. Cependant, il est tout-à-fait possible de concevoir des systèmes de débruitage indépendants de la tâche. Il suffit alors de faire la reconnaissance de suites phonétiques au lieu de la reconnaissance de suites de mots. L'utilisation d'un modèle jouant le rôle de modèle de langage dans le contexte de la reconnaissance phonétique est d'une grande importance afin d'augmenter la robustesse du système. Cette étude fera certainement l'objet de nos futurs travaux de recherche dans ce domaine. Une autre solution à ce problème est de concevoir des systèmes de débruitage en utilisant des système de reconnaissance avec de très grands vocabulaires comme ceux évalués dans le cadre d'ARPA [26].

# Chapitre 7

# Adaptation des systèmes par alignement stochastique

Les techniques d'adaptation que nous avons présentées jusqu'à présent sont fondées sur le modèle du canal de transmission : $y = (s + n) * h$, $y$ est le signal observé, $s$ est le signal propre correspondant, $n$ est le bruit additif et $h$ est le bruit convolutif. Ce modèle n'inclut pas toutes les variabilités acoustiques (effet Lombard, changement de locuteur, *etc*). Dans ce chapitre nous présentons une technique d'adaptation ne supposant aucun modèle *a priori*, utilisant un critère d'optimisation fondé sur le maximum de vraisemblance pour estimer des transformations à appliquer sur les modèles acoustiques (MMCDC[1]) du système de reconnaissance. Les transformations utilisées sont généralement des régressions linéaires estimées en utilisant le maximum de vraisemblance d'où le nom en anglais MLLR (*Maximum Likelihood Linear Regression*) [45]. Dans ce cadre, l'estimation des paramètres des transformations est optimale dans le sens où elle tient compte de la quantité de données d'adaptation disponible. En effet, suivant la quantité de données d'adaptation disponible, on utilise des transformations plus ou moins globales. Dans le cas d'une adaptation individuelle de chaque paramètre, cette technique est équivalente à l'algorithme d'estimation de Baum [7]. Les transformations utilisées dans [45] concernant les moyennes sont de type : $f(x) = \mathbf{A}x + \mathbf{B}$. Dans ce chapitre nous proposons une extension de cette technique permettant d'utiliser des transformations plus complexes de type : $f(x) = \mathbf{A}Diag(xx') + \mathbf{B}x + \mathbf{D}$, où $Diag(xx')$ désigne la diagonale de $xx'$. Nous présentons également une technique d'adaptation permettant d'adapter conjointement les moyennes et les matrices de covariance. Enfin des résultats expérimentaux sont présentés.

## 7.1 Adaptation des paramètres d'un MMCDC

On suppose qu'on dispose d'un système de reconnaissance de la parole indépendant du locuteur entraîné dans un environnement donné. Le système est adapté

[1] Modèle de Markov Caché à Distributions Continues (voir le chapitre 2).

à un nouvel environnement par des transformations linéaires[2] des paramètres. On suppose également qu'on ne dispose que d'une petite quantité de données d'adaptation dans le nouvel environnement (par rapport à la quantité de données nécessaire à l'apprentissage), ce qui rend impossible l'adaptation individuelle de chaque paramètre du modèle (les données ne couvrent pas d'une manière suffisante la totalité des modèles)[3]. Rappelons qu'un MMCDC est caractérisé par les paramètres suivants :

- $N$ : le nombre d'états dans le MMCDC.
- $A$ : une matrice carrée $N \times N$, où $a_{ij}$ représente la probabilité de passer de l'état $i$ à l'état $j$.
- $b_i$ : la densité de probabilité $b_i$ associée à l'état $i(1 \leq i \leq N)$.

La densité de probabilité associée à chaque état est une somme finie de gaussiennes, on suppose que chaque somme comporte $M$ termes. $b_i$ peut être décrite comme suit :

- $c_{ik}$ : le poids de la $k^{ième}$ gaussienne de l'état $i$.
- $\mu_{ik}$ : la moyenne de la $k^{ième}$ gaussienne de l'état $i$.
- $C_{ik}$ : la matrice de covariance $(n \times n)$ de la $k^{ième}$ gaussienne de l'état $i$, où $1 \leq k \leq M$, et $n$ est la dimension du vecteur de paramètre représentant une trame de parole, incluant les paramètres statiques et dynamiques.

Etant donnée une trame de parole (vecteur $o$), la probabilité que celle-ci soit générée par la $k^{ième}$ gaussienne du $i^{ième}$ état est :

$$b_{ik}(o) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|C_{ik}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(o-\mu_{ik})' C_{ik}^{-1} (o-\mu_{ik})} \tag{7.1}$$

La probabilité pour que $o$ soit généré par le $i^{ième}$ état est :

$$b_i(o) = \sum_{k=1}^{M} c_{ik} b_{ik}(o) \tag{7.2}$$

Théoriquement tous les éléments caractérisant un MMCDC peuvent être adaptés : la matrice de transition, les poids des gaussiennes, les moyennes des gaussiennes et les matrices de covariance des gaussiennes. Dans ce chapitre nous allons considérer seulement l'adaptation des moyennes et des matrices de covariance des gaussiennes, en supposant que les autres paramètres du MMCDC restent inchangés.

[2] Les transformations utilisées sont linéaires localement dans des parties de l'espace acoustique. Mais globalement elles sont non linéaires, du fait qu'elles sont différentes d'une partie de l'espace acoustique à l'autre.

[3] Nous allons montrer que l'adaptation individuelle de chaque paramètre du système est équivalente à l'estimation des paramètres du système en utilisant l'algorithme d'estimation habituel [40] sur les données d'adaptation (formule 7.21). Il est donc clair que si la quantité des données d'adaptation est inférieure à celle des données d'apprentissage, l'estimation individuelle des paramètres sera moins robuste.

## Adaptation de la moyenne

La moyenne d'une densité gaussienne d'un MMCDC est adaptée en utilisant une transformation linéaire $W$. $W$ est une matrice $n \times (n+1)$ estimée en utilisant le critère du maximum de la vraisemblance. Soit $\mu$ la moyenne de la gaussienne en question, on définit $\hat{\mu}$ comme suit[4] :

$$\hat{\mu} = \begin{bmatrix} 1 \\ \mu_1 \\ . \\ . \\ \mu_n \end{bmatrix}. \tag{7.3}$$

La moyenne adaptée est donnée par :

$$\mu_{ad} = W\hat{\mu}. \tag{7.4}$$

Dans un premier temps, on traite le cas d'une seule gaussienne par état ; à la fin de cette section on généralise au cas de plusieurs gaussiennes par état. On considère un état $s$ d'un MMCDC ; soient $\mu_s$ et $C_s$ la moyenne et la matrice de covariance de la gaussienne correspondant à cet état. On note $W_s$ la matrice de régression $(n \times (n+1))$ associée à l'état $s$ ; on définit $\hat{\mu}_s$ comme suit :

$$\hat{\mu}_s = \begin{bmatrix} 1 \\ \mu_{s,1} \\ . \\ . \\ \mu_{s,n} \end{bmatrix}, \tag{7.5}$$

où $\mu_{s,i}$ désigne la $i^{i\grave{e}me}$ composante de $\mu_s$. La densité de probabilité associée à l'état $s$ après l'adaptation est :

$$b_s(o) = \frac{1}{(2\pi)^{\frac{n}{2}} |C_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(o - W_s\hat{\mu}_s)' C_s^{-1} (o - W_s\hat{\mu}_s)}. \tag{7.6}$$

Le critère utilisé pour estimer $W_s$ est celui du maximum de vraisemblance. C'est-à-dire que la matrice $W_s$ est choisie de telle façon que les données d'adaptation soient générées par les modèles adaptés avec une probabilité maximale. La technique utilisée pour réaliser cette maximisation est l'algorithme EM [15], c'est-à-dire la détermination de la fonction auxiliaire puis sa maximisation.

## Détermination de la fonction auxiliaire

On suppose que les données d'adaptation forment une suite de $T$ observations $O = (o_1, \ldots, o_T)$. On note $\lambda$ l'ensemble des paramètres des modèles acoustiques courant,

[4] le premier élément de $\hat{\mu}$ qui est égal à 1 est utilisé afin d'introduire un terme *offset* dans la transformation d'adaptation.

et $\bar{\lambda}$ celui des nouveaux paramètres. On note $\theta = (\theta_1, \ldots, \theta_T)$ l'ensemble des états utilisés pour générer $O$. La probabilité de générer $O$ par la succession d'états $\theta$ est :

$$f(O, \theta|\lambda) = \pi_{\theta_1} \prod_{t=2}^{T} a_{\theta_{t-1}\theta_t} b_{\theta_t}(o_t) \tag{7.7}$$

La probabilité de générer $O$ est :

$$f(O|\lambda) = \sum_{\theta \in \Theta} f(O, \theta|\lambda) \tag{7.8}$$

où $\Theta$ désigne l'ensemble de toutes les séquences d'états de longueur $T$. La fonction auxiliaire de l'algorithme EM est donnée comme suit :

$$Q(\lambda, \bar{\lambda}) = \sum_{\theta \in \Theta} f(O, \theta|\lambda) \log(f(O, \theta|\bar{\lambda})). \tag{7.9}$$

L'objectif de l'algorithme EM est de trouver les paramètres du modèle qui maximisent (ou augmentent) la valeur de la fonction auxiliaire.

### Maximisation de la fonction auxiliaire

Dans ce qui suit, on traite la deuxième partie de l'algorithme EM, c'est-à-dire la maximisation de la fonction auxiliaire. Les nouveaux paramètres sont indiqués par une barre au dessus (par exemple $\bar{\mu}$). La fonction auxiliaire s'écrit comme une somme de trois termes, le premier correspond aux probabilités initiales, le deuxième correspond aux probabilités des transitions dans le MMCDC et le troisième correspond aux probabilités d'émissions. Puisqu'on s'intéresse seulement à l'adaptation des moyennes, seul le troisième terme sera développé :

$$Q(\lambda, \bar{\lambda}) \equiv \sum_{s=1}^{N} Q_b(\lambda, \bar{b}_s), \tag{7.10}$$

où

$$Q_b(\lambda, \bar{b}_s) = \sum_{\theta \in \Theta} \sum_{t=1}^{T} f(O, \theta_t = s|\lambda) \log \bar{b}_s(o_t). \tag{7.11}$$

On désigne par $\gamma_s(t)$ la probabilité d'être dans l'état $s$ à l'instant $t$ sachant que la séquence $O$ a été générée par le MMCDC :

$$\gamma_s(t) = \frac{1}{f(O|\lambda)} \sum_{\theta \in \Theta} f(O, \theta_t = s|\lambda) \tag{7.12}$$

L'équation 7.11 devient :

$$Q_b(\lambda, \bar{b}_s) = f(O|\lambda) \sum_{t=1}^{T} \gamma_s(t) \log \bar{b}_s(o_t). \tag{7.13}$$

Explicitons $\log \bar{b}_s(o_t)$ :

$$\log \bar{b}_s(o_t) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log |C_s| - \frac{1}{2}(o - W_s \hat{\mu}_s)' C_s^{-1} (o - W_s \hat{\mu}_s) \tag{7.14}$$

Dans ce cas la fonction auxiliaire devient :

$$Q_b(\lambda, \bar{b}_s) = -\frac{1}{2} f(O|\lambda) \sum_{t=1}^{T} \gamma_s(t)[n \log(2\pi) + \log |C_s| + (o_t - W_s\hat{\mu}_s)' C_s^{-1}(o_t - W_s\hat{\mu}_s)] \tag{7.15}$$

Pour maximiser $Q_b(\lambda, \bar{b}_s)$ par rapport à $W_s$, on dérive par rapport à $W_s$ et on détermine la valeur de $W_s$ pour laquelle cette dérivée s'annule.

$$\frac{d}{dW_s} Q_b(\lambda, \bar{b}_s) = -\frac{1}{2} f(O|\lambda) \frac{d}{dW_s} \sum_{t=1}^{T} \gamma_s(t)[n \log(2\pi) + \log |C_s| + h(o_t, s)] \tag{7.16}$$

avec

$$h(o_t, s) = (o_t - W_s\hat{\mu}_s)' C_s^{-1}(o_t - W_s\hat{\mu}_s) \tag{7.17}$$

La dérivée de $h(o_t, s)$ par rapport à $W_s$ est :

$$\frac{d}{dW_s} h(o_t, s) = -2C_s^{-1}[o_t - W_s\hat{\mu}_s]\hat{\mu}_s{}' \tag{7.18}$$

Donc $W_s$ qui maximise la fonction auxiliaire vérifie l'équation :

$$\sum_{t=1}^{T} \gamma_s(t) C_s^{-1}[o_t - W_s\hat{\mu}_s]\hat{\mu}_s{}' = 0 \tag{7.19}$$

ce qui implique :

$$\sum_{t=1}^{T} \gamma_s(t) C_s^{-1} o_t \hat{\mu}_s' = \sum_{t=1}^{T} \gamma_s(t) C_s^{-1} W_s \hat{\mu}_s \hat{\mu}_s' \tag{7.20}$$

Dans le cas où on associe à chaque état une transformation individuelle, la valeur de la moyenne adaptée se déduit facilement de l'équation précédente. Sachant que $\bar{\mu}_s = W_s\hat{\mu}_s$ et en utilisant l'équation (7.20), on obtient :

$$\bar{\mu}_s = W_s\hat{\mu}_s = \frac{\sum_{t=1}^{T} \gamma_s(t) o_t}{\sum_{t=1}^{T} \gamma_s(t)} \tag{7.21}$$

qui n'est rien d'autre que la formule habituelle de réestimation de la moyenne.

L'extension à un mélange de gaussiennes par état ne pose pas de problème particulier. En effet, le terme de la fonction auxiliaire correspondant à la $k^{ième}$ gaussienne du $j^{ième}$ état ($b_{jk}$) s'écrit :

$$Q_b(\lambda, \bar{b}_{jk}) = f(O|\lambda) \sum_{t=1}^{T} \gamma_{jk}(t) \log \bar{b}_{jk}(o_t) \tag{7.22}$$

où $\gamma_{jk}(t)$ est définie comme $\gamma_j(t)$ mais au niveau de la gaussienne, c'est-à-dire que $\gamma_{jk}(t)$ est la probabilité que la trame $o_t$ soit générée par la $k^{ième}$ gaussienne du $j^{ième}$ état :

$$\gamma_{jk}(t) = \frac{1}{f(O|\lambda)} \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} f(O, \theta_t = j, k_t = k|\lambda) \tag{7.23}$$

où $\Omega_b$ est l'ensemble de toutes les séquences de gaussiennes de longueur $T$ possibles, et $K$ un élément de $\Omega_b$, c'est-à-dire que $K = k_1, k_2, \ldots, k_T$.

## Matrice de régression associée à une famille d'états

L'équation (7.21) donne l'estimation d'une matrice de régression spécifique à un état donné. Dans le cas d'une matrice de régression associée[5] à une famille de $E$ états $(s_1, s_2, \ldots, s_E)$, il suffit de reprendre l'équation (7.20) en faisant la sommation sur tous les états dans cette famille, on obtient alors :

$$\sum_{t=1}^{T}\sum_{e=1}^{E}\gamma_{s_e}(t)C_{s_e}^{-1}o_t\hat{\mu}_{s_e}^{'} = \sum_{t=1}^{T}\sum_{e=1}^{E}\gamma_{s_e}(t)C_{s_e}^{-1}W_s\hat{\mu}_{s_e}\hat{\mu}_{s_e}^{'}. \tag{7.24}$$

Dans ce qui suit, on détermine la formule d'adaptation, c'est-à-dire la transformation $W_s$ associée à la famille d'états $(s_1, s_2, \ldots, s_E)$. Pour cela on écrit l'équation précédente sous la forme suivante :

$$\sum_{t=1}^{T}\sum_{e=1}^{E}\gamma_{s_e}(t)C_{s_e}^{-1}o_t\hat{\mu}_{s_e}^{'} = \sum_{e=1}^{E}V^{(e)}W_sD^{(e)} \tag{7.25}$$

avec

$$V^{(e)} = \sum_{t=1}^{T}\gamma_{s_e}(t)C_{s_e}^{-1}, \tag{7.26}$$

et

$$D^{(e)} = \hat{\mu}_{s_e}\hat{\mu}_{s_e}^{'}. \tag{7.27}$$

Si on note les éléments des matrices $V^{(e)}$, $W_s$ et $D^{(e)}$ respectivement par $v_{ij}^{(e)}$, $w_{ij}$ et $d_{ij}^{(e)}$, alors la partie droite de l'équation (7.25) est une matrice $Y$ d'ordre $(n \times (n+1))$ où $y_{ij}$ sont les éléments qui la composent, et on peut écrire :

$$y_{ij} = \sum_{p=1}^{n}\sum_{q=1}^{n+1} w_{pq}\left[\sum_{e=1}^{E} v_{ip}^{(e)}d_{qj}^{(e)}\right] \tag{7.28}$$

En utilisant le fait que toutes les matrices de covariance sont diagonales et que les matrices $D^e$ sont symétriques, on en déduit que :

$$\sum_{e=1}^{E} v_{ip}^{(e)}d_{qj}^{(e)} = \begin{cases} \sum_{e=1}^{E} v_{ii}^{(e)}d_{jq}^{(e)} & si \quad i = p \\ 0 & si \quad i \neq p \end{cases} \tag{7.29}$$

ce qui implique

$$y_{ij} = \sum_{q=1}^{n+1} w_{iq}g_{jq}^{(i)} \tag{7.30}$$

[5] Cela veut dire que les éléments de la famille d'états $(s_1, s_2, \ldots, s_E)$ partagent la même transformation. C'est-à-dire que toutes les données associées à tous les états de cette famille sont utilisées pour estimer une seule transformation, d'où la robustesse de l'estimation dans le cas de peu de données d'adaptation.

avec

$$g_{jq}^{(i)} = \sum_{e=1}^{E} v_{ii}^{(e)} d_{jq}^{(e)} \tag{7.31}$$

La partie gauche de l'équation (7.25) est une matrice $Z$ d'ordre $(n \times (n+1))$, notons $z_{ij}$ les éléments qui la composent, On peut donc écrire :

$$z_{ij} = y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \tag{7.32}$$

on peut en déduire que :

$$w_i' = G^{(i)-1} z_i' \tag{7.33}$$

où $w_i$ et $z_i$ sont les $i^{ièmes}$ lignes des matrices $W_s$ et $Z$ respectivement. Cette équation peut être résolue en utilisant par exemple la décomposition $LU$ pour calculer $W_s$ ligne par ligne.

## 7.2 Réduction du nombre de paramètres

La matrice de régression agit sur la moyenne de deux manières différentes :

$$\mu_{adp} = W\hat{\mu},$$

$$\mu_{adp_i} = w_{i,1} + \sum_{j=1}^{n} w_{i,j+1}\mu_j. \tag{7.34}$$

Cette dernière équation met en évidence deux termes, le premier $w_{i,1}$ réalise une compensation par un *offset*, le deuxième terme $\sum_{j=1}^{n} w_{i,j+1}\mu_j$ exprime le fait que la $i^{ième}$ composante de la moyenne est une combinaison linéaire de toutes les composantes de cette dernière. Dans le reste de cette section, nous allons discuter ces deux cas intéressants.

### Matrice de régression pleine mais sans *offset*

Les mêmes formules de réestimation sont appliquées, la seule différence est qu'on ne travaille qu'avec une matrice carrée $W$ $(n \times n)$.

### Utilisation d'une matrice de régression diagonale

Ceci diminue considérablement les calculs mais suppose l'indépendance des coefficients, contrainte qui n'est pas forcément respectée. Considérons la transformation suivante :

$$W_s = \begin{bmatrix} w_{1,1} & w_{1,2} & 0 & \dots & 0 \\ w_{2,1} & 0 & w_{2,3} & \dots & 0 \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ w_{n,1} & 0 & \dots & 0 & w_{n,n+1} \end{bmatrix} \tag{7.35}$$

On note $\hat{\mathrm{W}}_s$ la matrice colonne dont les éléments sont les coefficients de l'*offset* et de la diagonale de la matrice $n \times n$ restante.

$$\hat{\mathrm{W}}_s = \begin{bmatrix} w_{1,1} \\ . \\ . \\ w_{n,1} \\ w_{1,2} \\ . \\ . \\ w_{n,n+1} \end{bmatrix} \tag{7.36}$$

Calculons la forme quadratique dans ce cas :

$$\begin{aligned} h(o_t, s) &= (o_t - W_s\hat{\mu}_s)' C_s^{-1} (o_t - W_s\hat{\mu}_s) \\ &= (o_t - D_s\hat{\mathrm{W}}_s)' C_s^{-1} (o_t - D_s\hat{\mathrm{W}}_s) \end{aligned} \tag{7.37}$$

où $D_s$ est une matrice $n \times 2n$ définie comme suit :

$$D_s = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 & \mu_1 & 0 & \dots & & 0 \\ 0 & 1 & 0 & \dots & \dots & 0 & \mu_2 & 0 & \dots & 0 \\ . & & & & & & & & & . \\ . & & & & & & & & & . \\ . & & & & & & & & & . \\ 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 & \mu_{n-1} & . \\ 0 & \dots & \dots & 0 & 1 & 0 & \dots & \dots & \dots & \mu_n \end{bmatrix} \tag{7.38}$$

en dérivant par rapport à $\hat{\mathrm{W}}_s$ on obtient :

$$\frac{d}{d\hat{\mathrm{W}}_s} h(o_t, s) = -2D_s' C_s^{-1}(o_t - D_s\hat{\mathrm{W}}_s), \tag{7.39}$$

la dérivée de $Q_b(\lambda, \bar{b}_s)$ s'écrit alors (voir 7.15):

$$\frac{d}{d\hat{\mathrm{W}}_s} Q_b(\lambda, \bar{b}_s) = \sum_{t=1}^{T} \gamma_s(t) D_s' C_s^{-1}(o_t - D_s\hat{\mathrm{W}}_s) \tag{7.40}$$

ce qui implique que :

$$\hat{\mathrm{W}}_s = \left[\sum_{t=1}^{T} \gamma_s(t) D_s' C_s^{-1} D_s\right]^{-1} \left[\sum_{t=1}^{T} \gamma_s(t) D_s' C_s^{-1} o_t\right] \tag{7.41}$$

Pour le cas de matrice de régression partagée, il suffit de faire une sommation sur les distributions en question :

$$\hat{\mathrm{W}}_s = \left[\sum_{e=1}^{E}\sum_{t=1}^{T} \gamma_{s_e}(t) D_{s_e}' C_{s_e}^{-1} D_{s_e}\right]^{-1} \left[\sum_{e=1}^{E}\sum_{t=1}^{T} \gamma_{s_e}(t) D_{s_e}' C_{s_e}^{-1} o_t\right] \tag{7.42}$$

## 7.3 Transformations plus complexes

Dans cette section nous allons étendre l'approche d'adaptation, présentée dans les sections précédentes, à une transformation plus générale qu'une régression linéaire.

Notons $\mathbf{X}$ l'espace acoustique d'apprentissage, c'est-à-dire l'ensemble de toutes les trames (*cepstre* + $\Delta$ *cepstre* + ...) pouvant être générées durant la phase d'apprentissage, et $\mathbf{x}$ un élément de $\mathbf{X}$. Soit $\mathbf{Y}$ l'ensemble de toutes les trames pouvant être générées durant la phase de test, et $\mathbf{y}$ un élément de $\mathbf{Y}$. On suppose qu'à tout élément $\mathbf{y}$ de $\mathbf{Y}$ correspond un élément $\mathbf{x}$ de $\mathbf{X}$ et inversement. Cette correspondance peut être matérialisée par une fonction $f$ de $\mathbf{X}$ vers $\mathbf{Y}$ :

$$\begin{array}{rccc} f: & \mathbf{X} & \rightarrow & \mathbf{Y} \\ & \mathbf{x} & \rightarrow & \mathbf{y} = f(\mathbf{x}) \end{array} \tag{7.43}$$

Jusqu'à présent on a traité les fonctions suivantes $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$, $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}$ où $\mathbf{A}$ peut être diagonale ou non diagonale. Nous allons étendre cette étude au cas d'une fonction de type : $f(\mathbf{x}) = \mathbf{A}\mathbf{x}^2 + \mathbf{B}\mathbf{x} + \mathbf{D}$, où $\mathbf{x}^2$ est le vecteur dont les composantes sont les carrés des composantes de $\mathbf{x}$.

**La transformation $f(\mathbf{x}) = \mathbf{A}\mathbf{x}^2 + \mathbf{B}\mathbf{x} + \mathbf{D}$ où A et B sont pleines**

Dans ce cas l'espérance d'une variable aléatoire $Y$ dans $\mathbf{Y}$ peut s'exprimer en fonction des paramètres statistiques de la variable aléatoire $X$ qui lui correspond dans $\mathbf{X}$ comme suit :

$$E(Y) = \mathbf{A}E(X^2) + \mathbf{B}E(X) + \mathbf{D}, \tag{7.44}$$

où $X^2$ est la variable aléatoire dont les composantes des réalisations sont les carrés des composantes des réalisations de la variable aléatoire $X$. La formule d'adaptation de la moyenne de la variable aléatoire $X$ à l'espace $\mathbf{Y}$ (espace acoustique de test) est donnée comme suit :

$$\mu_Y = \mathbf{A}Diag(C_X) + \mathbf{B}\mu_X + \mathbf{D}, \tag{7.45}$$

où $Diag(C_X)$ désigne le vecteur composé des éléments diagonaux de la matrice de covariance $C_X$ de la variable aléatoire $X$. Soit $X$ une variable aléatoire dans $\mathbf{X}$ de densité de probabilité $N(\mu, C)$. La moyenne de la gaussienne adaptée à $\mathbf{Y}$ est :

$$\mu_{ad} = \mathbf{A}Diag(C) + \mathbf{B}\mu + \mathbf{D}, \tag{7.46}$$

Définissons $\hat{\mu}$ comme suit :

$$\hat{\mu} = \begin{bmatrix} 1 \\ \mu_1 \\ . \\ . \\ \mu_n \\ \sigma_1^2 \\ . \\ . \\ \sigma_n^2 \end{bmatrix} \tag{7.47}$$

où $(\sigma_i^2)$ est la variance de la $i^{ième}$ composante de la variable aléatoire dans l'espace $\mathbf{X}$. La formule d'adaptation devient alors :

$$\mu_{ad} = W\hat{\mu}, \tag{7.48}$$

où $W$ est une matrice $n\times(2n+1)$. La détermination de $W$ qui maximise la probabilité que les données d'adaptation (de l'espace $\mathbf{Y}$) soient générées par le modèle adapté se fait exactement de la même manière que dans la section 7.1.

**La transformation $f(\mathbf{x}) = \mathbf{A}\mathbf{x}^2 + \mathbf{B}\mathbf{x} + \mathbf{D}$ où A et B sont diagonales**

Considérons la transformations suivante :

$$W_s = \begin{bmatrix} w_{1,1} & w_{1,2} & 0 & \ldots & 0 & w_{1,n+2} & 0 & \ldots & 0 \\ w_{2,1} & 0 & w_{2,3} & \ldots & 0 & 0 & w_{2,n+3} & \ldots & 0 \\ . & & & & . & & & & . \\ . & & & & . & & & & . \\ . & & & & . & & & & . \\ w_{n,1} & 0 & \ldots & 0 & w_{n,n+1} & 0 & \ldots & 0 & w_{n,2n+1} \end{bmatrix} \tag{7.49}$$

On note $\hat{\mathrm{W}}_s$ la matrice colonne dont les éléments sont les coefficients du *offset* et les deux sous-diagonales de la matrice $n \times 2n$ restante.

$$\hat{\mathrm{W}}_s = \begin{bmatrix} w_{1,1} \\ . \\ . \\ w_{n,1} \\ w_{1,2} \\ . \\ . \\ w_{n,n+1} \\ w_{1,n+2} \\ w_{2,n+3} \\ . \\ . \\ w_{n,2n+1} \end{bmatrix} \tag{7.50}$$

Calculons la forme quadratique dans ce cas :

$$\begin{aligned} h(o_t, s) &= (o_t - W_s\hat{\mu}_s)' C_s^{-1} (o_t - W_s\hat{\mu}_s) \\ &= (o_t - D_s\hat{\mathrm{W}}_s)' C_s^{-1} (o_t - D_s\hat{\mathrm{W}}_s) \end{aligned} \tag{7.51}$$

où $D_s$ est une matrice $n \times 3n$ définie comme suit :

$$D_s = \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 & \mu_1 & 0 & \ldots & & 0 & \sigma_1^2 & 0 & \ldots & & 0 \\ 0 & 1 & 0 & \ldots & \ldots & 0 & \mu_2 & 0 & \ldots & 0 & 0 & \sigma_2^2 & 0 & \ldots & 0 \\ . & & & & & & & & & . & & & & & . \\ . & & & & & & & & & . & & & & & . \\ . & & & & & & & & & . & & & & & . \\ 0 & \ldots & 0 & 1 & 0 & \ldots & \ldots & 0 & \mu_{n-1} & . & \ldots & \ldots & 0 & \sigma_{n-1}^2 & . \\ 0 & \ldots & \ldots & 0 & 1 & 0 & \ldots & \ldots & \ldots & \mu_n & 0 & \ldots & \ldots & \ldots & \sigma_n^2 \end{bmatrix} \tag{7.52}$$

en dérivant par rapport à $\hat{\mathrm{W}}_s$ on obtient :

$$\frac{d}{d\hat{\mathrm{W}}_s} h(o_t, s) = -2D_s^{'} C_s^{-1}(o_t - D_s \hat{\mathrm{W}}_s) \tag{7.53}$$

la dérivée de $Q_b(\lambda, \bar{b}_s)$ s'écrit alors (voir 7.15):

$$\frac{d}{d\hat{\mathrm{W}}_s} Q_b(\lambda, \bar{b}_s) = \sum_{t=1}^{T} \gamma_s(t) D_s^{'} C_s^{-1}(o_t - D_s \hat{\mathrm{W}}_s) \tag{7.54}$$

ce qui implique que :

$$\hat{\mathrm{W}}_s = \left[\sum_{t=1}^{T} \gamma_s(t) D_s^{'} C_s^{-1} D_s\right]^{-1} \left[\sum_{t=1}^{T} \gamma_s(t) D_s^{'} C_s^{-1} o_t\right] \tag{7.55}$$

Pour le cas de matrices de régression partagées, il suffit de faire une sommation sur les distributions en question :

$$\hat{\mathrm{W}}_s = \left[\sum_{e=1}^{E}\sum_{t=1}^{T} \gamma_s(t) D_{s_e}^{'} C_{s_e}^{-1} D_{s_e}\right]^{-1} \left[\sum_{e=1}^{E}\sum_{t=1}^{T} \gamma_s(t) D_{s_e}^{'} C_{s_e}^{-1} o_t\right] \tag{7.56}$$

## 7.4 Adaptation des matrices de covariance

Dans les paragraphes précédents, nous nous sommes intéressés à l'adaptation des moyennes des gaussiennes des MMCDC en utilisant la régression linéaire. Cette régression linéaire a été estimée en utilisant le critère du maximum de vraisemblance. Dans ce paragraphe nous allons présenter les transformations permettant d'adapter à la fois les moyennes et les matrices de covariance des gaussiennes. Les transformations concernant l'adaptation des moyennes sont les mêmes que celles utilisées dans les paragraphes précédents. Si on suppose que les matrices de covariance sont diagonales, alors il est possible de les décomposer comme suit :

$$C = C^{\frac{1}{2}} C^{\frac{1}{2}} \tag{7.57}$$

avec :

$$C^{\frac{1}{2}} = \begin{bmatrix} \sigma_{11} & 0 & \dots & \dots & 0 \\ 0 & \sigma_{22} & 0 & \dots & \dots \\ . & & & & \\ . & & & & \\ . & & & & \\ 0 & \dots & 0 & & 0 \\ 0 & \dots & \dots & 0 & \sigma_{nn} \end{bmatrix} \tag{7.58}$$

où $\sigma_{ii}^2$, $(1 \leq i \leq n)$, sont les éléments diagonaux de la matrice de covariance $C$. Les transformations concernant l'adaptation des matrices de covariance sont comme suit :

$$C_{ad} = C^{\frac{1}{2}} S C^{\frac{1}{2}}, \tag{7.59}$$

où $C_{ad}$ est la nouvelle matrice de covariance. L'estimation de $S$ et $W$ (voir section 7.1 pour la définition de $W$) utilise le critère du maximum de vraisemblance. Afin de réaliser la maximisation de la fonction de vraisemblance nous employons l'algorithme EM. Soit $O = (o_1, \ldots, o_T)$ une suite de $T$ observations représentant les données d'adaptation. La fonction auxiliaire s'écrit comme une somme de trois termes, le premier correspond aux probabilités initiales, le deuxième correspond aux probabilités des transitions dans le MMCDC et le troisième correspond aux probabilités d'émissions associées aux états. Puisqu'on ne s'intéresse qu'à l'adaptation des moyennes et des matrices de covariance des gaussiennes, seul le troisième terme est concerné par la maximisation.

$$Q(\lambda, \bar{\lambda}) \equiv \sum_{s=1}^{N} Q_b(\lambda, \bar{b}_s) \tag{7.60}$$

où l'indice $s$ porte sur les numéros d'états, $N$ est le nombre d'états dans le MMCDC et $Q_b(\lambda, \bar{b}_s)$ vaut :

$$Q_b(\lambda, \bar{b}_s) = \sum_{\theta \in \Theta} \sum_{t=1}^{T} f(O, \lambda_t = s|\lambda) \log \bar{b}_s(o_t) \tag{7.61}$$

On désigne par $\gamma_s(t)$ la probabilité d'être dans l'état $s$ à l'instant $t$ sachant que la séquence $O$ a été générée par le MMCDC :

$$\gamma_s(t) = \frac{1}{f(O|\lambda)} \sum_{\theta \in \Theta} f(O, \theta_t|\lambda) \tag{7.62}$$

L'équation 7.61 devient :

$$Q_b(\lambda, \bar{b}_s) = f(O|\lambda) \sum_{t=1}^{T} \gamma_s(t) \log \bar{b}_s(o_t) \tag{7.63}$$

Explicitons $\log \bar{b}_s(o_t)$ :

$$\log \bar{b}_s(o_t) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\left|C_s^{\frac{1}{2}} S_s C_s^{\frac{1}{2}}\right| - \frac{1}{2}(o_t - W_s\hat{\mu}_{ik})'(C_s^{\frac{1}{2}} S_s C_s^{\frac{1}{2}})^{-1}(o_t - W_s\hat{\mu}_s) \tag{7.64}$$

Dans ce cas la fonction auxiliaire devient :

$$Q_b(\lambda, \bar{b}_s) = -\frac{1}{2} f(O|\lambda) \sum_{t=1}^{T} \gamma_s(t) \times$$

$$[n\log(2\pi) + \log|C_s| + \log|S_s| + C_s^{-\frac{1}{2}}(o_t - W_s\hat{\mu}_s)' S_s^{-1} C_s^{-\frac{1}{2}}(o_t - W_s\hat{\mu}_s)] \tag{7.65}$$

où $C_s^{-\frac{1}{2}}$ est la matrice inverse de $C_s^{\frac{1}{2}}$.

Il est évident que pour maximiser $Q_b(\lambda, \bar{b}_s)$ par rapport à $S_s$ et $W_s$, il suffit de maximiser itérativement par rapport à $W$ en supposant que les matrices de covariance sont fixes (S=I, I est la matrice identité), puis par rapport à $S$ en remplaçant les moyennes par leurs nouvelles estimations et en les supposant fixes. Ce processus itératif peut être exécuté jusqu'à la convergence (la vraisemblance des données d'adaptation atteint un maximum). L'augmentation de la vraisemblance des données d'adaptation est assurée après chaque itération :

$$Q_b(\lambda, \bar{b}_s) = Q_b(\mu_s, C_s; \bar{\mu}_s, \bar{C}_s) \leq Q_b(\bar{\mu}_s, C_s; \bar{\mu}_s, \bar{C}_s) \leq Q_b(\bar{\mu}_s, \bar{C}_s; \bar{\mu}_s, \bar{C}_s). \tag{7.66}$$

L'estimation de $W_s$ en fixant les matrices de covariance se fait exactement de la même manière que dans les paragraphes précédents. Il reste à estimer les matrices $S_s$ ; pour maximiser $Q_b(\lambda, \bar{b}_s)$ par rapport à $S$, on peut utiliser la méthode du gradient, ou tout simplement en remarquant que $Q_b(\lambda, \bar{b}_s)$ a la même forme que le cas standard d'estimation des paramètres d'un MMCDC à partir de données d'apprentissage, on obtient ainsi :

$$S_s = \frac{1}{\sum_{t=1}^{T} \gamma_s(t)} C_s^{-\frac{1}{2}} \left[ \sum_{t=1}^{T} \gamma_s(t)(o_t - W_s\hat{\mu}_s)(o_t - W_s\hat{\mu}_s)' \right] C_s^{-\frac{1}{2}} \tag{7.67}$$

Intuitivement, la transformation $S_s$ donnée par l'équation 7.67 constitue le ratio entre les variances des données d'apprentissage et les variances des données d'adaptation. Son application sur les variances des modèles a donc pour objectif de réduire au mieux la différence de variances entre les modèles (représentant les données d'apprentissage) et les données d'adaptation.

L'équation 7.67 donne l'estimation d'une matrice de régression spécifique à un état donné. Dans le cas d'une matrice de régression associée à une famille de $E$ états $(s_1, s_2, \ldots, s_E)$, il suffit de reprendre l'équation (7.67) en faisant la sommation sur tous les états dans cette famille, on obtient alors :

$$S_s = \frac{1}{\sum_{t=1}^{T} \sum_{e=1}^{E} \gamma_{s_e}(t)} \sum_{e=1}^{E} C_{s_e}^{-\frac{1}{2}} \left[ \sum_{t=1}^{T} \gamma_{s_e}(t)(o_t - W_s\hat{\mu}_{s_e})(o_t - W_s\hat{\mu}_{s_e})' \right] C_{s_e}^{-\frac{1}{2}} \tag{7.68}$$

L'extension à un mélange de gaussiennes par état ne pose pas de problème particulier, en effet, le terme de la fonction auxiliaire correspondant à la $k^{ième}$ gaussienne du $j^{ième}$ état $(b_{jk})$ s'écrit :

$$Q_b(\lambda, \bar{b}_{jk}) = f(O|\lambda) \sum_{t=1}^{T} \gamma_{jk}(t) \log \bar{b}_{jk}(o_t) \tag{7.69}$$

où $\gamma_{jk}(t)$ est définie comme $\gamma_j(t)$ mais au niveau de la gaussienne, c'est-à-dire que $\gamma_{jk}(t)$ est la probabilité que la trame $o_t$ soit générée par la $k^{ième}$ gaussienne du $j^{ième}$ état :

$$\gamma_{jk}(t) = \frac{1}{f(O|\lambda)} \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} f(O, \theta_t = j, k_t = k|\lambda) \tag{7.70}$$

où $\Omega_b$ est l'ensemble de toutes les séquences de gaussiennes de longueur $T$ possibles, et $K$ un élément de $\Omega_b$, c'est-à-dire que $K = k_1, k_2, \ldots, k_T$.

## 7.5 Mise en oeuvre

Dans cette section nous exposons seulement le cas des transformations d'adaptation $f(\mathbf{x}) = \mathbf{Ax} + \mathbf{B}$. L'extension aux transformations $f(\mathbf{x}) = \mathbf{Ax}^2 + \mathbf{Bx} + \mathbf{D}$ se déduit trivialement en examinant la section 7.3.

### 7.5.1 Le cas d'une matrice pleine

A ce niveau, on suppose que la classification des distributions qui vont partager la même transformation est déjà faite, on suppose qu'il y a $p$ classes qu'on note $(C_1, \ldots, C_p)$. Chaque classe $C_i$ $(1 \leq i \leq p)$ est un ensemble de distributions qui vont subir la même transformation $W_i$. A chaque classe $C_i = \{s_1^i, \ldots, s_{R_i}^i\}$ on associe $n+1$ accumulateurs initialisés à zéro, un pour la matrice $Z_i$ et les $n$ autres sont les matrices $G_i^j$ $(1 \leq j \leq n)$. En fait la matrice $G_i^j$ va être utilisée pour l'estimation de la $j^{ième}$ ligne de la matrice $W_i$. Ce qui suit concerne le cas général, c'est-à-dire celui d'une matrice $W_i$ pleine et d'une densité sous forme de mélange de gaussiennes dans chaque état. Les transformations $W_i$ peuvent être calculées en deux étapes, la première est l'accumulation des statistiques, la deuxième est l'utilisation de ces statistiques pour déterminer les transformations :

```
Algorithme 1:
Initialiser tous les accumulateurs de toutes les classes a zero
Pour chaque vecteur o_t dans les donnees d'adaptation
    pour chaque etat s utilise
        pour chaque gaussienne k dans s
            Determiner γ_{s_k}(t)
            i* = arg_{1≤j≤p}[s ∈ C_j]
            Z_{i*} += γ_{s_k}(t) C_{s_k}^{-1} o_t μ̂'_s
            pour(i = 1; i ≤ n; i++)
                pour(j = 1; j ≤ n+1; j++)
                    pour(q = 1; q ≤ n+1; q++)
                        G_{i*}^{(i)}[j][q] += γ_{s_k}(t) c_{ii}^{s_k} μ_j^{s_k} μ_q^{s_k}
                    fin-pour
                fin-pour
            fin-pour
        fin-pour
    fin-pour
fin-pour
```

Une fois toutes les statistiques correspondant à toutes les classes déterminées, on peut les utiliser pour déterminer les éléments des matrices $W_i (1 \leq i \leq p)$, où $p$ est le nombre de classes, avec la relation suivante :

$$\mathrm{W}_{\mathrm{i,j}}' = \mathrm{G}_{\mathrm{i}}^{(\mathrm{j})-1} \mathrm{z}_{\mathrm{i,j}}' \qquad (7.71)$$

Pour chaque classe $C_i (i \leq i \leq p)$
pour chaque ligne $j$ de la matrice $W_i$
$\mathrm{W}'_{\mathrm{i,j}} = \mathrm{G}_{\mathrm{i}}^{(\mathrm{j})-1} \mathrm{z}'_{\mathrm{i,j}}$
($\mathrm{W}_{\mathrm{i,j}}$ est la ligne numero $j$ de $W_i$)
fin-pour
fin-pour

où $\mathrm{W}_{\mathrm{ij}}$ désigne la $j^{ième}$ ligne de la transformation correspondant à la $i^{ième}$ classe et $z'_{ij}$ représente aussi la $j^{ième}$ ligne de la matrice $Z_i$ correspondant à la $i^{ième}$ classe. Cette équation peut être résolue en appliquant par exemple une décomposition LU.

### 7.5.2 Le cas d'une matrice diagonale avec *offset*

On utilise les mêmes notations que dans la section 7.1. Pour chaque classe $C_i$ $(i \leq i \leq p)$ nous avons besoin de deux accumulateurs que nous notons $AC_i^1$ et $AC_i^2$. La matrice de la transformation associée à l'état $s$ et à la $k^{ième}$ gaussienne dans cet état est donnée par :

$$\hat{\mathrm{W}}_{s_k} = \begin{bmatrix} w_{1,1}^{s_k} \\ . \\ . \\ w_{n,1}^{s_k} \\ w_{1,2}^{s_k} \\ . \\ . \\ w_{n,n+1}^{s_k} \end{bmatrix} \tag{7.72}$$

Pour la $k^{ième}$ gaussienne dans l'état $s$ on définit la matrice $D_{s_k}$ comme suit :

$$D_s = \begin{bmatrix} w & 0 & \dots & \dots & 0 & \mu_1 & 0 & \dots & & 0 \\ 0 & w & 0 & \dots & \dots & 0 & \mu_2 & 0 & \dots & 0 \\ . & & & & & & & & & . \\ . & & & & & & & & & . \\ . & & & & & & & & & . \\ 0 & \dots & 0 & w & 0 & \dots & \dots & 0 & \mu_{n-1} & . \\ 0 & \dots & \dots & 0 & w & 0 & \dots & \dots & \dots & \mu_n \end{bmatrix} \tag{7.73}$$

Pour plus de détails voir section 7.1.

**Algorithme 2**:
Initialiser tous les accumulateurs $AC_i^1$ et $AC_i^2$ a zero pour toutes les classes
Pour chaque vecteur $o_t$ dans les donnees d'adaptation
    pour chaque etat $s$ utilise
        pour chaque gaussienne k dans $s$
            Determiner $\gamma_{s_k}(t)$
            $i^* = arg_{1 \leq j \leq p}[s \in C_j]$
            $AC_{i^*}^1 += \gamma_{s_k}(t) D'_{s_k} C_{s_k}^{-1} D_{s_k}$
            $AC_{i^*}^2 += \gamma_{s_k}(t) D'_{s_k} C_{s_k}^{-1} o_t$
        fin-pour
    fin-pour
fin-pour

pour chaque classe $C_i$ faire
    $\hat{\mathrm{W}}_i = [AC_i^1]^{-1} [AC_i^2]$
fin-pour

### 7.5.3 Le cas d'une matrice pleine: Viterbi

Nous avons vu que la détermination de la transformation $W_i$ qu'il faut appliquer à l'ensemble de toutes les distributions appartenant à la classe $C_i (1 \leq i \leq p)$ nécessite le calcul de $\gamma_s(t)$ qui est la probabilité d'occupation de l'état s à l'instant $t$ sachant qu'on a observé la séquence $O$ (données d'adaptation). Ces quantités peuvent être calculées en utilisant l'algorithme "avant-arrière" de Baum. Mais une bonne approximation peut être faite en utilisant l'algorithme d'alignement de Viterbi [82]. Dans ce cas cette probabilité ne peut prendre que deux valeurs 0 ou 1. La formule utilisée dans ce cas pour déterminer la transformation $W_i$ est la suivante:

$$\sum_{t=1}^{T} C_{\theta_t}^{-1} o_t \hat{\mu}'_{\theta_t} \delta_{\theta_t} = \sum_{t=1}^{T} C_{\theta_t}^{-1} W_i \hat{\mu}_{\theta_t} \hat{\mu}'_{\theta_t} \delta_{\theta_t} \tag{7.74}$$

avec

$$\delta_{\theta_t} = \begin{cases} 1 & si \quad \theta_t \in \{s_1, \ldots, s_E\} \\ 0 & sinon \end{cases}$$

$\{s_1, \ldots, s_E\}$ désigne l'ensemble des états (ou des distributions) associés à la classe $C_i$. Notons $z_{ij}$ l'élément de la matrice $Z$ constituant la partie droite de l'équation précédente, de même notons $w_{ij}$ un élément de la matrice $W_s$.

$$z_{ij} = \sum_{q=1}^{n+1} w_{iq} \sum_{e=1}^{E} \frac{nbre}{\sigma_{ii}^{2e}} \mu_q^e \mu_j^e \tag{7.75}$$

cette équation peut s'écrire:

$$w'_i = G^{(i)-1} z'_i \tag{7.76}$$

où $w_i$ et $z_i$ sont les lignes numéro $i$ des matrices $W$ et $Z$ respectivement et $G^{(i)}$ est donnée par :

$$G^{(i)} = g_{pq}^{(i)} = \left( \sum_{e=1}^{E} \frac{nbre}{\sigma_{ii}^{2e}} \mu_p^e \mu_q^e \right) (1 \leq p, q \leq n+1) \tag{7.77}$$

où $nbre$ est le nombre de fois où l'état $e$ (ou la gaussienne $e$) a été rencontré au moment de l'alignement par Viterbi. On utilise les mêmes notations que dans la section 7.1 :

**Algorithme 3** :
Initialiser tous les accumulateurs de toutes les classes a zero
Pour chaque vecteur $o_t$ dans les donnees d'adaptation
    $i^* = arg_{1 \leq j \leq p}[\theta_t \in C_j]$
    $Z_{i^*} + = C_{\theta_t}^{-1} o_t \hat{\mu}'_{\theta_t}$
    pour($i = 1; i \leq n; i++$)
        pour($j = 1; j \leq n+1; j++$)
            pour($q = 1; q \leq n+1; q++$)
                $G_{i^*}^{(i)}[j][q] + = \frac{\mu_j^{\theta_t} \mu_q^{\theta_t}}{\sigma_{ii}^{2(\theta_t)}}$
            fin-pour
        fin-pour
    fin-pour
fin-pour

Une fois que toutes les statistiques correspondant à toutes les classes sont déterminées, on peut les utiliser pour déterminer les éléments des matrices $W_i$, $(1 \leq i \leq p)$ où $p$ est le nombre de classes avec la relation suivante :

$$\mathrm{W}'_{\mathrm{i,j}} = \mathrm{G}_{\mathrm{i}}^{(\mathrm{j})-1} \mathrm{z}'_{\mathrm{i,j}} \tag{7.78}$$

Pour chaque classe $C_i (1 \leq i \leq p)$
    pour chaque ligne j de la matrice $W_i$
        $\mathrm{W}'_{\mathrm{i,j}} = \mathrm{G}_{\mathrm{i}}^{(\mathrm{j})-1} \mathrm{z}'_{\mathrm{i,j}}$
        ($\mathrm{W}_{\mathrm{i,j}}$ est la ligne numero $j$ de $W_i$)
    fin-pour
fin-pour

**Remarque :** les transformations et les algorithmes décrits précédemment traitent le cas d'une seule observation. L'extension au cas d'observations multiples est triviale, elle se fait par sommation sur toutes les observations.

## 7.6 Résultats expérimentaux

Nous évaluons les approches de compensation présentées dans les paragraphes précédents sur l'application MASK [42]. Les signaux sont enregistrés avec deux microphones simultanément, un microphone de tête ($S_a$ avec un RSB $\simeq$ 35dB) et un microphone de table ($S_b$ avec un RSB $\simeq$ 20dB). Comme dans les expériences précédentes, les modèles ont été appris sur les signaux $S_a$ (RSB = 35dB). Les données de test et d'adaptation sont enregistrées avec $S_b$ et bruitées soit avec le bruit de brouhaha (Voice babble noise), soit avec le bruit d'hélicoptère (Lynx). Le rapport signal sur bruit dans les données de test et d'adaptation est de 12dB.

### 7.6.1 Adaptation des moyennes

| type d'adaptation | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| bruit de brouhaha dans le test : RSB=12dB | | | | | | | |
| test $\neq$ adaptation | 30.4% | 44.6% | 24.0% | 24.4% | 25.1% | 21.1% | 20.3% |
| test = adaptation | 30.4% | 42.3% | 23.9% | 24.4% | 21.2% | 17.7% | 14.2% |
| bruit d'hélicoptère (Lynx) dans le test : RSB=12dB | | | | | | | |
| test $\neq$ adaptation | 23.4% | 40.3% | 20.0% | 19.3% | 18.2% | 15.6% | 15.4% |
| test = adaptation | 23.4% | 38.7% | 19.6% | 18.7% | 15.3% | 13.6% | 11.9% |

**Table 7.1** *Dans ces expériences, le bruit de brouhaha de la base de données NOISEX-92 (Voice babble noise) ($3^{ième}$ et $4^{ième}$ lignes) et le bruit d'hélicoptère (Lynx) ($6^{ième}$ et $7^{ième}$ lignes) sont utilisés pour bruiter les données de test et d'adaptation avec un RSB=12dB, les données d'apprentissage sont propres (35dB). Les $3^{ième}$ et $6^{ième}$ lignes contiennent les résultats lorsque les données d'adaptation sont différentes des données de test, Les $3^{ième}$ et $6^{ième}$ lignes contiennent les résultats lorsqu'on utilise les données de test pour l'adaptation. L'adaptation est faite en mode supervisé. Les colonnes contiennent les taux d'erreur sur les mots dans le cas des transformations d'adaptation suivantes :*

| *N° de colonne* | *Type d'adaptation* |
|---|---|
| *$1^{ère}$ colonne* | *Sans aucune adaptation* |
| *$2^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu$, *où* $\mathbf{A}$ *est une matrice diagonale* |
| *$3^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ *où* $\mathbf{A}$ *une matrice diagonale et* $\mathbf{B}$ *le vecteur offset* |
| *$4^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}Diag(C) + \mathbf{D}$ *($C$ : la matrice de covariance)* <br> *où* $\mathbf{A}$ *et* $\mathbf{B}$ *des matrices diagonales et* $\mathbf{D}$ *un vecteur offset* |
| *$5^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu$ *où* $\mathbf{A}$ *une matrice pleine* |
| *$6^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ *où* $\mathbf{A}$ *une matrice pleine et* $\mathbf{B}$ *un vecteur offset* |
| *$7^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}Diag(C) + \mathbf{D}$ <br> *où* $\mathbf{A}$ *et* $\mathbf{B}$ *des matrices pleines et* $\mathbf{D}$ *le vecteur offset* |

Une première remarque est qu'en général plus le nombre de paramètres de la transformation d'adaptation est grand, plus les performances du système adapté sont meilleures, ceci est vrai dans tous les cas sauf un seul : dans la $3^{ième}$ ligne de la la table 7.1, lorsque la transformation d'adaptation est $\mu_{ad} = \mathbf{A}\mu$ avec $\mathbf{A}$ pleine, le taux d'erreur sur les mots est de 25.1% ($5^{ième}$ colonne), et lorsque la transformation d'adaptation est $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ avec $\mathbf{A}$ diagonale le taux d'erreur sur les mots est de 24% ($3^{ième}$ colonne). Ce qui implique que suivant la nature des variabilités acoustiques affectant les données de test, il est possible de choisir une transformation optimale. Dans cet exemple une transformation avec seulement $2n$ paramètres a donné un résultat meilleur qu'une transformation avec $n \times n$ éléments.

Une deuxième remarque importante est que l'adaptation du système en utilisant la transformation $\mu_{ad} = \mathbf{A}\mu$ avec $\mathbf{A}$ diagonale a considérablement dégradé les performances du système de reconnaissance. En effet, le taux d'erreur sur les mots, sans compensation, lorsque les données de test sont bruitées avec le bruit de brouhaha est de 30.4%, alors qu'il est de 44.6% ($2^{ième}$ colonne de la $3^{ière}$ ligne de la table 7.1) lorsqu'on compense les modèles avec la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu$ avec $\mathbf{A}$ diagonale, malgré l'augmentation considérable de la vraisemblance des données de test dans le deuxième cas. Ce comportement subsiste même lorsqu'on adapte les modèles du système en utilisant les données de test en mode supervisé : le taux d'erreur sur les mots, lorsqu'on compense les modèles avec la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu$ avec $\mathbf{A}$ diagonale, est de 42.3% ($2^{ième}$ colonne de la $4^{ième}$ ligne de la table 7.1). Les mêmes remarques sont valables lorsque les données de test sont bruitées avec le bruit d'hélicoptère (Lynx) (lignes 6 et 7).

On peut conclure des remarques faites dans le paragraphe précédent que quelle que soit la transformation utilisée pour l'adaptation, l'algorithme d'estimation des paramètres de cette transformation est conçu de façon à augmenter la vraisemblance des données d'adaptation et par suite celle des données de test, mais ceci ne garantit en aucun cas des performances meilleures en terme de taux de reconnaissance. Dans le cas des données de test utilisées dans ces expériences nous avons remarqué l'importance de *l'offset* correspondant aux transformations d'adaptation ; il est possible que pour d'autres données de test ce terme (*offset*) soit sans importance. Vu que nous ne disposons pas de cette connaissance *a priori*, il est préférable d'utiliser toujours des transformations d'adaptation prenant en compte ce terme (*l'offset*), d'autant plus que le coût en terme de temps de calcul supplémentaire nécessaire n'est pas très important.

En comparant la $3^{ième}$ (respectivement la $6^{ième}$) ligne de la table 7.1 avec la $4^{ième}$ (respectivement la $7^{ième}$) ligne de la table 7.1, nous constatons que la différence entre les taux d'erreur correspondant à l'adaptation avec les données de test et les taux d'erreur correspondant à l'adaptation avec des données différentes de celles du test augmente avec le nombre de paramètres de la transformation d'adaptation utilisée. Par exemple, lorsque les données de test et d'adaptation sont bruitées avec le bruit d'hélicoptère (Lynx), et lorsque la transformation d'adaptation est $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$, l'augmentation relative du taux d'erreur sur les mots entre les deux cas (test = adaptation et test $\neq$ adaptation) est de 2% (de 19.6% à 20.0%), en revanche, quand la transformation d'adaptation est $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}Diag(C) + \mathbf{D}$, l'augmentation relative entre les deux cas (test = adaptation et test $\neq$ adaptation) est de 29% (de 11.9% à 15.4%). Ceci nous permet de conclure que plus le nombre de paramètres de la transformation est grand plus l'adaptation qui en découle est spécifique aux données d'adaptation, elle donne donc de très bons résultats sur les données d'adaptation mais pas autant sur les données de test.

Sur les données de test utilisées dans ces expériences nous constatons que l'apport de la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}Diag(C) + \mathbf{D}$ est marginal par rapport à la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ où $\mathbf{A}$ est une matrice pleine (dans le cas du bruit de brouhaha le gain est de 3.8%). Par contre le gain est très important si les données de test sont utilisées pour l'adaptation (dans le cas du bruit de brouhaha le gain est de 19.8%).

### 7.6.2 Adaptation des moyennes et des matrices de covariance

| Le bruit affectant les données | Type d'adaptation 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| bruit de Lynx | 23.6% | 15.6% | 13.3% | 14.2% | 14.5% | 14.0% | 14.5% |
| bruit de brouhaha | 30.4% | 21.1% | 19.7% | 18.2% | 18.5% | 17.9% | 16.9% |

**Table 7.2** *Dans ces expériences l'adaptation est réalisée en mode supervisé, les données de test sont différentes des données d'adaptation. Dans la $1^{ère}$ ligne les données sont bruitées avec le bruit de Lynx et dans la $2^{ième}$ ligne les données sont bruitées avec le bruit de brouhaha. Les colonnes contiennent les taux d'erreur sur les mots dans le cas des transformations d'adaptation suivantes:*

| *N° de colonne* | *Type d'adaptation* |
|---|---|
| *$1^{ère}$ colonne* | *Sans aucune adaptation* |
| *$2^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$, *où* $\mathbf{A}$ *une matrice pleine et* $\mathbf{B}$ *vecteur offset* |
| *$3^{ième}$ colonne* | $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ *et* $C_{ad} = C^{\frac{1}{2}}SC^{\frac{1}{2}}$ *où* $C$ *la matrice de covariance,* $C^{\frac{1}{2}}$ *est définie dans 7.4,* $S$ *est une matrice diagonale. L'adaptation est réalisée avec une seule itération* |
| *$4^{ième}$ colonne, $5^{ième}$ colonne* | *même type d'adaptation que dans la $3^{ième}$ colonne, mais avec 2 itérations, 3 itérations* |
| *$6^{ième}$ colonne respectivement $7^{ième}$ colonne* | *1 itération (respectivement 2 itérations) compensant les moyennes et les matrices de covariance :* $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ *et* $C_{ad} = C^{\frac{1}{2}}SC^{\frac{1}{2}}$, *où* $\mathbf{A}$ *une matrice pleine,* $\mathbf{B}$ *le vecteur offset et* $S$ *une matrice diagonale ; ces itérations sont suivies d'une itération compensant les moyennes avec :* $\mu_{ad} = \mathbf{A}Diag(C) + \mathbf{B}\mu + \mathbf{D}$, *où* $\mathbf{A}$ *et* $\mathbf{B}$ *sont deux matrices pleines et* $\mathbf{D}$ *le vecteur offset* |

Nous remarquons que l'adaptation conjointe des moyennes et des matrices de covariance apporte toujours un gain (en terme de taux de reconnaissance) par rapport à l'adaptation des moyennes uniquement. Par exemple avec une seule itération, dans le cas où les données sont bruitées avec le bruit d'hélicoptère (Lynx), le gain apporté par l'adaptation des moyennes et des matrices de covariance par rapport à l'adaptation des moyennes uniquement est d'environ 15% (de 15.6% à 13.3%) ; dans les mêmes conditions, dans le cas du bruit de brouhaha, le gain apporté par l'adaptation des matrices de covariance est de 6.6% (de 21.1% à 19.7%). Comme on peut le remarquer dans la table 7.2 (colonne 3, ligne 1 / colonne 4, ligne 1) l'augmentation du nombre d'itérations n'implique pas forcément la diminution du taux d'erreur sur les mots, en dépit de l'augmentation de la vraisemblance des données de test et d'adaptation. Cependant, dans certains cas l'augmentation du nombre d'itérations peut considérablement diminuer le taux d'erreur sur les mots, par exemple, dans le cas du bruit de brouhaha ($2^{ième}$ ligne de la table 7.2) la $2^{ième}$ itération apporte un gain supplémentaire de 7.6% par rapport à la $1^{ère}$ itération, soit un gain d'environ 14% par rapport à l'adaptation des moyennes uniquement.

Dans la $7^{ième}$ colonne de la $2^{ième}$ ligne (dans le cas du bruit de brouhaha) nous remarquons qu'on obtient une chute considérable du taux d'erreur, de 21.1% à 16.9%, soit un gain de 20% par rapport à l'adaptation des moyennes uniquement. Cette colonne correspond à deux itérations adaptant les moyennes ($\mu$) et les matrice de covariance ($C$) avec les transformations $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ et $C_{ad} = C^{\frac{1}{2}} S C^{\frac{1}{2}}$, où $\mathbf{A}$ est une matrice pleine, $\mathbf{B}$ le vecteur *offset* et $S$ une matrice diagonale, (voir section 7.4 pour la définition de $C^{\frac{1}{2}}$ ) ; ces itérations sont suivies d'une seule itération compensant seulement les moyennes avec la transformation : $\mu_{ad} = \mathbf{A} Diag(C) + \mathbf{B}\mu + \mathbf{D}$, avec $\mathbf{A}$ et $\mathbf{B}$ deux matrices pleines et $\mathbf{D}$ le vecteur *offset*.

### 7.6.3 la CPD suivie de la MLLR

Reprenons l'expérience II de la section 5.6.3, Les données d'apprentissage sont affectées par le bruit de foule et le données de test sont affectées par un bruit enregistré à la gare Saint-Lazare. Les tables II.5.11, II.5.12 et II.5.13 présentent respectivement les taux d'erreur sur les mots lorsqu'aucune compensation n'est réalisée, avec compensation du bruit de test et enfin avec compensation des bruits d'apprentissage et de test. Dans la table 7.3 nous présentons les taux d'erreur sur les mots lorsqu'une compensation des bruits de test et d'apprentissage est suivie d'une adaptation aveugle de type MLLR.

L'adaptation MLLR compense les différences résiduelles entre les données d'apprentissage et les données de test correspondant, d'une part au bruit non représenté dans le modèle du canal de transmission, et d'autre part, aux erreurs commises lors de la combinaison parallèle de données. Une seule matrice de régression ($42 \times 43$) a été utilisée pour transformer les moyennes des modèles. Une comparaison entre les tables 7.3 et II.5.13 montre que la MLLR apporte toujours un gain supplémentaire surtout pour les faibles rapport signal sur bruit dans l'apprentissage et dans le test.

**Table 7.3 :** Compensation des bruits d'apprentissage et de test + MLLR non supervisée

| Conditions de test | Conditions d'apprentissage RSB=35dB | RSB=20dB | RSB=12dB |
|---|---|---|---|
| $S_a$ : RSB=35dB | 5.4% | 6.1% | 10.6% |
| $S_b$ : RSB=20dB | 8.8% | 9.1% | 10.8% |
| $S_b$ : RSB=12dB | 10.1% | 10.7% | 11.0% |
| $S_b$ : RSB=9dB | 13.2% | 12.9% | 13.5% |

*Dans ces expériences, le bruit de brouhaha a été utilisé pour avoir différents RSBs dans les données d'apprentissage. Le bruit de test est celui enregistré à la Gare Saint Lazare. Cette table montre les taux d'erreur moyens sur les mots pour une série de RSB dans l'apprentissage et dans le test, lorsqu'une compensation des bruits de test et d'apprentissage est suivie d'une adaptation aveugle de type MLLR.*

## 7.7 Conclusion

Dans ce chapitre nous avons montré que la technique d'adaptation aveugle (MLLR) [45, 74] constitue un outil intéressant pour l'adaptation des modèles acoustiques aux données de test, notamment pour les variabilités acoustiques pour lesquelles nous ne disposons pas de modèle. Nous avons souligné le fait que certaines transformations (matrice diagonale sans *offset*) utilisées pour l'adaptation des moyennes peuvent causer une dégradation importante des performances du système en dépit de l'amélioration de la vraisemblance des données de test.

Nous avons étendu cette technique d'adaptation aveugle à des transformations plus complexes prenant en compte les moment d'ordre 1 et d'odre 2 de chaque gaussienne. Le gain obtenu avec ces nouvelles transformations est marginal par rapport à l'augmentation du nombre de paramètres des transformations et au temps de calcul supplémentaire nécessaire à leur estimation. Nous avons montré que l'adaptation conjointe des moyennes et des matrices de covariance apporte un gain significatif par rapport à l'adaptation des moyennes uniquement.

Enfin, les techniques de compensation aveugle (n'utilisant aucun modèles *a priori*) peuvent être vues comme des techniques permettant de corriger les erreurs d'approximations commises par les techniques de compensation guidées par un modèle (dans notre cas c'est le modèle du canal de transmission : $y = (x + n) * h$, où $y$ est le signal bruité, $x$ est le signal propre correspondant, $n$ le bruit additif et $h$ le bruit convolutif) ). Ces approximations peuvent être liées soit à l'insuffisance du modèle utilisé soit à la technique de compensation elle même, dans notre cas c'est la combinaison parallèle de données.

Il est important de noter que ce type d'adaptation ne doit être appliqué qu'après avoir effectué toutes les adaptations (compensations) reposant sur un modèle. En effet, il est toujours préférable d'introduire de l'information *a priori* dans le processus d'adaptation, à condition que cette information *a priori* soit relativement correcte. Par exemple l'information : $y = (x + n) * h$, apporte beaucoup plus d'informations au processus de compensation que l'information $y = f(x)$, où $f$ est une fonction à estimer en utilisant des critères statistiques.

# Chapitre 8

# Conclusions et Perspectives

Dans cette thèse nous avons concentré nos efforts sur l'augmentation de la robustesse des systèmes de reconnaissance de la parole aux variabilités acoustiques et plus particulièrement celles causées par l'environnement. A des fins de traitements automatiques nous avons réparti les variabilités acoustiques selon trois classes : les bruits additifs, les bruits convolutifs et enfin toutes les autres variabilités acoustiques (effet Lombard, variabilité *inter-locuteur*, *etc*). Nous avons montré que le système de reconnaissance de la parole se trouve sérieusement perturbé lorsque les conditions acoustiques de test sont très différentes de celles d'apprentissage et ceci quelle que soit la technique de paramétrisation acoustique utilisée[1] (MFCC[2], LPCC[3], PLPCC[4]) (voir chapitre 3). Dans ce travail nous proposons une nouvelle approche compensant explicitement les variabilités liées aux bruits additifs et convolutifs en utilisant le modèle du canal de transmission (voir chapitre 5). Les variabilités résiduelles non représentées dans le modèle du canal de transmission sont compensées globalement en utilisant des transformations matricielles rapprochant l'espace acoustique d'apprentissage de celui de test (aucun modèle physique n'est appliqué).

Les nombreux travaux traitant de la compensation des différences entre les conditions d'apprentissage et de test, supposent généralement que les données d'apprentissage sont propres et les données de test sont bruitées. Dans les applications réelles cette supposition est rarement satisfaite. Dans cette thèse nous avons proposé une nouvelle technique itérative pour compenser conjointement les bruits additifs et convolutifs de test et d'apprentissage (voir chapitre 5).

Après avoir traité le problème de la robustesse des systèmes de reconnaissance, nous nous sommes intéressés à l'augmentation de la qualité d'un signal de parole dégradé par un bruit additif non corrélé avec le signal et relativement stationnaire (voir chapitre 6).

Dans la suite nous résumons les travaux réalisés dans cette thèse ainsi que les résultats obtenus.

---

[1] On ne dispose donc pas, à l'heure actuelle, d'une technique d'extraction de paramètres suffisamment robustes aux variabilités acoustiques

[2] Mel scaled Frequency Cepstrum Coefficients

[3] Linear Predictive Cepstrum Coefficients

[4] Perceptually-based Linear Prediction Cepstrum Coefficients

## 8.1 Travaux réalisés et résultats obtenus

**Bruits additifs et convolutifs (chapitre 5) :**

La compensation des bruits additif et convolutif présents en phase de test est fondée sur le modèle du canal de transmission :

$$\begin{aligned} y_1(t) &= s(t) && \text{pour l'apprentissage} \\ y_2(t) &= h(t) * (s(t) + n(t)) && \text{pour le test} \end{aligned}$$

où $y_1$ et $y_2$ désignent respectivement les signaux d'apprentissage et de test observés, $s$ désigne le signal propre (non bruité), $n$ et $h$ désignent respectivement les bruits additif et convolutif dans les données de test. Cette compensation est réalisée en deux étapes : la première étape consiste en l'estimation itérative des composantes convolutive $h$ et additive $n$ ; lors de la seconde étape la compensation des bruits additif et convolutif proprement dite se fait en bruitant les modèles acoustiques (MMCDC)[5] correspondants à la parole propre avec le bruit $n$ estimé dans la première étape. Enfin, la normalisation par rapport à la composante convolutive $h$ se fait par soustraction du cepstre moyen sur la phrase.

Notre approche de compensation permet d'obtenir des modèles acoustiques correspondant aux signaux de parole bruitée avec très peu d'approximations. Elle permet d'estimer les paramètres statistiques correspondant aux coefficients *cepstraux* dynamiques quelles que soient les formules utilisées pour leurs calculs (une simple différence, une régression ou autre). Un autre avantage lié à cette approche est le fait qu'elle soit applicable quel que soit le type de paramétrisation acoustique utilisée[6] (MFCC, LPCC, PLPCC *etc*).

La compensation des bruits additifs et convolutifs affectant les données en phase d'utilisation réduit considérablement les taux d'erreur en présence de bruit, et ne dégrade pas les performances quand les données sont propres. Nous avons pu vérifier, d'après les expériences réalisées, qu'il est plus difficile de compenser les bruits non stationnaires affectant les basses fréquences (comme le bruit de brouhaha) par rapport aux bruits stationnaires affectant les hautes fréquences, surtout pour les faibles Rapport Signal sur Bruit (RSB).

Dans le cadre des évaluations ARPA, avec le système de reconnaissance de la parole du LIMSI [26], on obtient un taux d'erreur sur les mots de 10.4% sur le corpus de test ARPA NAB CSR[7] de novembre 1995 dont le rapport signal sur bruit est de 30dB environ. En revanche, en utilisant le même système sur le même corpus de test mais avec un rapport signal sur bruit variant entre 7 et 18dB, le taux d'erreur sur les mots est supérieur à 50%. En compensant les bruits additifs et convolutifs qui affectent les données de test, le taux d'erreur sur les mots descend à 20.5%, soit une diminution du taux d'erreur sur les mots supérieure à 60%.

[5] Modèles de Markov Cachés à Distribution Continues

[6] Pour peu qu'il y ait possibilité de retour du domaine de cette paramétrisation à un domaine où la relation de combinaison entre le signal propre et le bruit pour former la signal bruité est la plus simple possible (par exemple une simple addition dans le domaine spectral).

[7] Il s'agit d'une tâche de reconnaissance de la parole avec un vocabulaire illimité sur le corpus *"North American Business News"* [65].

**Bruits en phases d'apprentissage et de test (chapitre 5) :**

Afin de compenser les bruits additifs et convolutifs présents en phase de test et dans le corpus d'apprentissage, nous avons proposé une technique itérative consistant à adapter les paramètres des modèles phonétiques (MMCDC) en utilisant le bruit présent en phase de test et à compenser les trames représentant les données de test en utilisant le bruit présent en phase d'apprentissage. Aucune contrainte n'est imposée concernant les types de bruits présents en phase de test et d'apprentissage. Les microphones utilisés en phase d'apprentissage et de test sont supposés différents et inconnus *a priori*. Cette technique a été évaluée en utilisant le système de reconnaissance du LIMSI modélisant les unités acoustiques par des MMCDC [26]. Trois configurations ont été comparées, sans compensation, avec compensation du bruit de test et avec compensation conjointe des bruits d'apprentissage et de test. Plusieurs valeurs de Rapport Signal sur Bruit (RSB) en phase de test et d'apprentissage ont été explorées.

La compensation du bruit d'apprentissage et de test fondée sur le modèle du canal de transmission $y = (s + n) * h$ apporte un gain considérable par rapport aux résultats de base (sans compensation) et aux résultats avec compensation du bruit de test seulement. Ceci est valable pour les deux types de bruits d'apprentissage, stationnaire et non stationnaire. Le gain obtenu par compensation du bruit d'apprentissage est particulièrement important quand les données de test sont propres. Quand on teste sur des données avec un RSB égal à 35dB et quand les modèles sont entraînés sur des données dont le RSB égal 12dB (resp. 20dB), la compensation des bruits d'apprentissage et de test apporte un gain de 80% (resp. 40%) dans le cas du bruit d'apprentissage stationnaire (Lynx) et de 54% (resp. 50%) dans le cas du bruit d'apprentissage non stationnaire (brouhaha), par rapport aux résultats obtenus en ayant compensé seulement le bruit de test. La compensation des bruits d'apprentissage et de test est toujours fructueuse, et ne dégrade pas les performances quand les données de test ou d'apprentissage sont propres.

**Débruitage d'un signal de parole (chapitre 6) :**

Dans certaines applications on s'intéresse à la qualité perceptive du signal de parole et non pas à l'extraction automatique du message linguistique qu'il porte. Dans ce cadre nous avons développé une technique de débruitage fondée sur un filtrage dépendant de la gaussienne. Le filtre à appliquer à une trame donnée est calculé en fonction d'une estimation de la densité spectrale du bruit et de la(les) gaussienne(s) du MMCDC à laquelle cette trame appartient. Le filtrage appliqué au signal bruité varie d'une trame à l'autre. Ces filtres sont estimés en utilisant les modèles entraînés sur la parole propre (des MMC Auto-Régressifs : MMCAR) et le modèle correspondant au bruit (gaussienne AR) [18]. L'utilisation des modèles entraînés sur les signaux propres pour faire l'alignement trame/gaussienne du signal bruité donne de très mauvais résultats, notamment dans le cas de signaux très bruités. Afin d'améliorer l'estimation des modèles correspondant au signal propre, nous avons combiné cette technique avec celle de la combinaison parallèle de données (voir chapitre 5). Ce qui donne un alignement trame/gaussienne largement meilleur même pour des signaux très bruités (les expériences ont été réalisées avec un RSB de 1dB).

Cette technique de filtrage augmente considérablement le rapport signal sur bruit et n'introduit pas de distorsions majeures. Une remarque qu'on peut faire, dès la première écoute du signal résultant de ce filtrage dépendant de la gaussienne, est l'absence totale du bruit musical qui a longtemps préoccupé les chercheurs dans ce domaine. Ce gain en qualité du signal est obtenu principalement grâce à l'introduction explicite de l'information *a priori*.

Jusqu'à présent, tous les travaux réalisés dans le domaine de l'augmentation de la robustesse des systèmes de reconnaissance, montrent qu'il n'est pas possible d'aller au delà des performances obtenus en entraînant et en testant le système dans les mêmes conditions (bruitées). En utilisant le système entraîné sur des signaux propres et en débruitant les signaux de test par la technique du filtrage dépendant de la gaussienne, nous avons obtenus des performances légèrement meilleures que celles obtenus avec la CPD. Ceci est certainement dû au fait que la CPD ne résulte pas en des modèles rigoureusement optimaux au sens du maximum de vraisemblance.

**Variabilités résiduelles (chapitre 7) :**

L'adaptation fondée sur la MLLR (*Maximum Likelihood Linear Regression*) [45] compense les différences résiduelles, correspondant aux bruits non représentés dans le modèle du canal de transmission, entre les données d'apprentissage et les données de test. La MLLR a montré un succès considérable pour l'adaptation. Elle consiste à adapter le système à un nouvel environnement par des transformations linéaires des paramètres. Les transformations sont estimées en alignant les données d'adaptation avec les états des modèles. Nous avons étendu cette technique à des transformations plus complexes prenant en compte la variance des données.

Dans les expériences que nous avons réalisées, en dépit de l'augmentation considérable de la vraisemblance des données de test, l'apport de la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}Diag(C) + \mathbf{D}$ ($\mu$ étant la moyenne d'une gaussienne, $C$ sa matrice de covariance, $\mathbf{A}$ et $\mathbf{B}$ des matrices carrées, et $\mathbf{D}$ un vecteur *offset*) est marginal par rapport à la transformation d'adaptation $\mu_{ad} = \mathbf{A}\mu + \mathbf{B}$ où $\mathbf{A}$ est une matrice pleine (dans le cas du bruit de brouhaha le gain est de 3.8%). D'autre part, nous avons montré que l'adaptation conjointe des moyennes et des matrices de covariance apporte un gain non négligeable par rapport à l'adaptation des moyennes uniquement.

## 8.2 Perspectives

Dans l'approche de compensation que nous avons utilisée, la variabilité due à l'effet Lombard n'est pas prise en compte explicitement. Dans les applications réelles, l'effet Lombard a toute son importance, notamment dans le cas des RSB faibles [69, 48]. Il est donc nécessaire de tenir compte de cette variabilité dans nos prochains travaux. Néanmoins, cette approche de compensation a été testée dans le cas de parole bruitée naturellement (donc présence de l'effet Lombard) et a donné de bons résultats [26].

Une autre hypothèse simplificatrice de notre approche consiste à supposer que le bruit convolutif est strictement stationnaire. Dans les expériences réalisées dans cette thèse nous n'avons pas étudié l'effet de cette approximation.

Dans le cadre du développement de notre approche de compensation l'aspect temps réel n'a pas été suffisamment pris en compte. Vu que seul le temps mis pour lire les données d'apprentissage à partir du disque est dominant, la solution au problème du temps réel passe nécessairement par un codage optimal de l'information spectrale (cepstre). Des expériences récentes, réalisées au LIMSI, montrent qu'une réduction importante du nombre de bits pour le codage des coefficients cepstraux est possible tout en gardant des résultats similaires aux codages habituellement utilisés.

Pour le filtrage dépendant de la gaussienne, nous avons supposé que le signal de parole est affecté par un bruit additif et relativement stationnaire durant une phrase. Il est cependant possible d'étendre cette technique de débruitage au cas de présence de bruits additif et convolutif.

Nous pouvons donc conclure ce travail de thèse en affirmant que de nombreuses voies restent à explorer afin d'éliminer au mieux les différentes hypothèses simplificatrices liées à notre démarche théorique et expérimentale.

Concernant la classification des approches de robustesse au bruit présentée dans l'introduction et poursuivie dans la présentation de nos travaux, nous avons pu montrer le succès des techniques de bruitage des références. Les techniques de débruitage seules ne produisent pas les améliorations escomptées. Nous tenons cependant à souligner, à la fin d'une étape que représente forcément un travail de thèse, que le succès incontestable des méthodes de bruitage a permis de faire avancer les approches par débruitage. Dans la dernière partie de nos travaux nous avons montré que le débruitage par filtrage dépendant de la gaussienne permet d'améliorer encore les résultats de reconnaissance et donc la robustesse au bruit. Il faut cependant garder à l'esprit que ce gain n'est pas obtenu par débruitage seul. L'estimation des filtres s'appuie sur les résultats obtenus par reconnaissance avec les modèles adaptés en utilisant la CPD. Cette vision des choses est à exploiter plus profondément dans des travaux futurs afin de déterminer la combinaison optimale entre les méthodes d'adaptation par bruitage et les méthodes d'adaptation par débruitage en fonction du signal à traiter.

Nous tenons à souligner que plus le système de reconnaissance est performant en cas d'absence de bruits, plus il est robuste en cas de présence de bruit. On en déduit que l'une des clés au problème de la robustesse au bruit est l'amélioration des performances des systèmes de reconnaissance en général.

# Annexe A

# Les paramètres statistiques d'une variable aléatoire Log-Normale

Soit $X = (X_1, X_2, \ldots, X_n)$ une variable aléatoire suivant une loi normale $N(\mu, \Sigma)$. La variable aléatoire $Y$ définie par $Y = exp(X)$ $(Y_i = exp(X_i), 1 \leq i \leq n)$ suit une loi Log-Normale. Calculons les moments du premier et second ordre de cette variable aléatoire. Nous avons besoin de ces paramètres lors du passage du domaine *log-spectral* au domaine spectral.

## A.1 Calcul de la moyenne de $Y$

La densité de probabilité de la variable aléatoire $X$ est :

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)' \Sigma^{-1} (\mathbf{x} - \mu) \right\} \tag{A.1}$$

Notons $\mu^l$ la moyenne de la variable aléatoire $Y$ :

$$\mu^l = E[Y] \tag{A.2}$$

avec $E[Y]$ désigne l'espérance de $Y$. Notons $\mu_i^l$ la $i^{\text{ème}}$ composante de $\mu^l$ :

$$\begin{aligned} \mu_i^l &= E[Y_i] = E[exp(X_i)] & \text{(A.3)} \\ &= \int_{\mathbf{R}^n} exp(x_i) \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)' \Sigma^{-1} (\mathbf{x} - \mu) \right\} d\mathbf{x} & \text{(A.4)} \end{aligned}$$

Posons le changement de variable $Z = X - \mu$. Soit $\mathbf{e}_i$ la $i^{\text{ème}}$ colonne de la matrice identité d'ordre $n$, c'est-à-dire que $\mathbf{e}_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)'$ avec 1 est situé à la $i^{\text{ème}}$ position. L'expression de la moyenne devient ainsi :

$$\mu_i^l = \int_{\mathbf{R}^n} \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} exp \frac{1}{2} \left\{ 2(z_i + \mu_i) - (-\mathbf{z})' \Sigma^{-1} \mathbf{z} \right\} d\mathbf{x} \tag{A.5}$$

Après développement, nous obtenons l'égalité suivante :

$$(\mathbf{z} - \Sigma \mathbf{e}_i)' \Sigma^{-1} (\mathbf{z} - \Sigma \mathbf{e}_i) = \mathbf{z}' \Sigma^{-1} \mathbf{z} - 2z_i + \sigma_{ii} \tag{A.6}$$

ce qui implique que :

$$2(z_i + \mu_i) - (-\mathbf{z})'\Sigma^{-1}\mathbf{z} = -(\mathbf{z} - \Sigma\mathbf{e}_i)'\Sigma^{-1}(\mathbf{z} - \Sigma\mathbf{e}_i) + \sigma_{ii} + 2\mu_i. \tag{A.7}$$

en utilisant cette égalité, l'expression de la moyenne devient :

$$\begin{aligned}\mu_i^Y &= exp(\frac{\sigma_{ii}}{2} + \mu_i)\int_{\mathbf{R}^n} \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(\mathbf{x} - \mu - \Sigma\mathbf{e}_i)'\Sigma^{-1}(\mathbf{x} - \mu - \Sigma\mathbf{e}_i)\right\} d\mathbf{x} \\ &= exp(\frac{\sigma_{ii}}{2} + \mu_i) \end{aligned} \tag{A.8}$$

nous avons ainsi exprimé la moyenne de la variable aléatoire $Y$ en fonction des moments du premier et second ordre de la variable aléatoire $X$

## A.2 Calcul de la matrice de covariance de $Y$

Notons $\Sigma^l$ la matrice de covariance de $Y$. Notons $\sigma_{ij}^l$ $(1 \leq i, j \leq n)$ les composantes de la matrice $\Sigma^l$. Par définition, nous avons :

$$E[exp(x_i)exp(x_j)] = \int_{\mathbf{R}^n} exp(x_i + x_j)\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)'\Sigma^{-1}(\mathbf{x} - \mu)\right\} d\mathbf{x} \tag{A.9}$$

avec le même changement de variable que dans le paragraphe précédent, nous avons :

$$E[exp(x_i)exp(x_j)] = \int_{\mathbf{R}^n} exp(z_i + \mu_i + z_j + \mu_j)\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(-\mathbf{z})'\Sigma^{-1}bfz\right\} d\mathbf{x} \tag{A.10}$$

Après développement, nous obtenons l'égalité suivante :

$$(\mathbf{z} - \Sigma(\mathbf{e}_i + \mathbf{e}_j))'\Sigma^{-1}(\mathbf{z} - \Sigma(\mathbf{e}_i + \mathbf{e}_j)) = \mathbf{z}'\Sigma^{-1}\mathbf{z} - 2z_i + \sigma_{ii} - 2z_j + \sigma_{jj} + 2\sigma_{ij}. \tag{A.11}$$

De la même manière que pour la moyenne, on obtient l'expression suivante :

$$E[exp(x_i)exp(x_j)] = exp(\frac{\sigma_{ii}}{2} + \mu_i)exp(\frac{\sigma_{jj}}{2} + \mu_j)exp(\sigma_{ij}) = \mu_i^l\mu_j^l exp(\sigma_{ij}). \tag{A.12}$$

On trouve ainsi :

$$\sigma_{ij}^l = E[exp(x_i)exp(x_j)] - E[exp(x_i)]E[exp(x_j)] \tag{A.13}$$

Finalement on obtient :

$$\sigma_{ij}^l = \mu_i^l\mu_j^l(exp(\sigma_{ij}) - 1) \quad (1 \leq i, j \leq n) \tag{A.14}$$

# Annexe B

# Compensation des paramètres cepstraux dynamiques

Dans le cadre de l'adaptation au bruit en utilisant la Combinaison Parallèle des Modèles (CPM), la compensation des paramètres cepstraux statiques peut être réalisée en utilisant l'approximation *log-Normale*. En ce qui concerne la compensation des paramètres *cepstraux* dynamiques, Gales [25], propose d'utiliser, ou bien l'intégration numérique, ou bien la DDPMC (data driven PMC). Dans nos expériences nous avons utilisé des formules analytiques pour réaliser cette compensation.

Nous allons développer ces formules de compensation dans le cas où le $\Delta$*cepstre* et $\Delta\Delta$*cepstre* sont des simples différences [1] :

$$\Delta O^{cp}(t) = -O^{cp}(t-2) + O^{cp}(t+2) \tag{B.1}$$

et

$$\Delta^2 O^{cp}(t) = -\Delta O^{cp}(t-1) + \Delta O^{cp}(t+1) \tag{B.2}$$

## B.1 Compensation du $\Delta$*cepstre*

$$\begin{aligned} \Delta O_i^{cp}(t) &= -O_i^{cp}(t-2) + O_i^{cp}(t+2) \\ &= C \log \left( \frac{O_i^{sp}(t+2)}{O_i^{sp}(t-2)} \right), \end{aligned} \tag{B.3}$$

l'indice $i$ désigne la $i^{\text{ème}}$ composante. Nous convenons que le $\Delta$*cepstre* dans le domaine spectral est :

$$\Delta O_i^{sp}(t) = \frac{O_i^{sp}(t+2)}{O_i^{sp}(t-2)} \tag{B.4}$$

[1] Dans les formules nous utilisons $\Delta^2$ à la place de $\Delta\Delta$

En remplaçant $O_i^{sp}(t+2)$ et $O_i^{sp}(t-2)$ par leurs valeurs en fonction des densités spectrales du signal propre $S^{sp}$ et du bruit $N^{sp}$, nous obtenons :

$$\Delta O_i^{sp}(t) = \frac{S_i^{sp}(t+2)+N_i^{sp}(t+2)}{S_i^{sp}(t-2)+N_i^{sp}(t-2)} \tag{B.5}$$

$$= \frac{S_i^{sp}(t+2)}{S_i^{sp}(t-2)}\left(\frac{\frac{S_i^{sp}(t-2)}{N_i^{sp}(t-2)}}{1+\frac{S_i^{sp}(t-2)}{N_i^{sp}(t-2)}}\right)+\frac{N_i^{sp}(t+2)}{N_i^{sp}(t-2)}\left(\frac{\frac{N_i^{sp}(t-2)}{S_i^{sp}(t-2)}}{1+\frac{N_i^{sp}(t-2)}{S_i^{sp}(t-2)}}\right) \tag{B.6}$$

Pour pouvoir estimer les moment d'ordre 1 et 2 de la variable aléatoire $\Delta O_i^{sp}(t)$, nous faisons l'hypothèse que le rapport $\frac{S_i^{sp}(t)}{N_i^{sp}(t)}$ est relativement constant et vaut $\frac{\mu_i^{sp}(t)}{\tilde{\mu}_i^{sp}(t)}$. avec $\mu_i^{sp}(t)$, la moyenne dans le domaine spectral du signal propre et $\tilde{\mu}_i^{sp}(t)$, la moyenne dans le domaine spectral du bruit. Cette supposition est réaliste dans le cas d'un bruit stationnaire ou possédant une variance assez petite. Notons $R_{sp}$ la parole bruitée dans le domaine spectral ; notons $\mu^i_{\Delta R_{sp}}$ sa $i^{\text{ème}}$ composante :

$$\mu^i_{\Delta R^{sp}} \approx \alpha_i \mu^i_{\Delta S^{sp}} + (1-\alpha_i)\mu^i_{\Delta N^{sp}} \tag{B.7}$$

Les composantes de la matrice de covariance de la variable aléatoire $\Delta O_i^{sp}(t)$ s'écrivent :

$$\sigma^{ij}_{\Delta R_{sp}} \approx \alpha_i\alpha_j\sigma^{ij}_{\Delta S_{sp}} + (1-\alpha_i)(1-\alpha_j)\sigma^{ij}_{\Delta N_{sp}} \tag{B.8}$$

avec

$$\alpha_i = \left(\frac{1}{1+\frac{\mu^i_{N_{sp}}}{\mu^i_{S_{sp}}}}\right) \tag{B.9}$$

## B.2 Compensation du $\Delta\Delta$*cepstre*

La formule du $\Delta\Delta$*cepstre* que nous utilisons est :

$$\Delta^2 O(t)_i^{cp} = -\Delta O_i^{cp}(t-1) + \Delta O_i^{cp}(t+1) \tag{B.10}$$

$$= C\log\left(\frac{\Delta O_i^{sp}(t+1)}{\Delta O_i^{sp}(t-1)}\right) \tag{B.11}$$

Nous convenons que le $\Delta\Delta$*cepstre* dans le domaine spectral est :

$$\Delta^2 O_i^{sp}(t) = \frac{\Delta O_i^{sp}(t+1)}{\Delta O_i^{sp}(t-1)}. \tag{B.12}$$

En remplaçant les $\Delta\Delta$*cepstres* par leurs valeurs en fonctions des paramètres du signal propre et bruit, calculées dans le paragraphe précédent nous obtenons :

$$\Delta O_i^{sp}(t) = \frac{\Delta S_i^{sp}(t+1)}{\Delta S_i^{sp}(t-1)}\left[\frac{\left(\frac{\frac{S_i^{sp}(t-1)}{N_i^{sp}(t-1)}}{1+\frac{S_i^{sp}(t-1)}{N_i^{sp}(t-1)}}\right)}{\left(\frac{\frac{S_i^{sp}(t-3)}{N_i^{sp}(t-3)}}{1+\frac{S_i^{sp}(t-3)}{N_i^{sp}(t-3)}}\right)+\frac{\Delta N_i^{sp}(t-1)}{\Delta S_i^{sp}(t-1)}\left(\frac{\frac{N_i^{sp}(t-3)}{S_i^{sp}(t-3)}}{1+\frac{N_i^{sp}(t-3)}{S_i^{sp}(t-3)}}\right)}\right]$$

$$+ \quad \frac{\Delta N_i^{sp}(t+1)}{\Delta N_i^{sp}(t-1)} \left[ \frac{\left( \frac{\frac{N_i^{sp}(t-1)}{S_i^{sp}(t-1)}}{1+\frac{N_i^{sp}(t-1)}{S_i^{sp}(t-1)}} \right)}{\left( \frac{\frac{S_i^{sp}(t-3)}{N_i^{sp}(t-3)}}{1+\frac{S_i^{sp}(t-3)}{N_i^{sp}(t-3)}} \right) + \frac{\Delta N_i^{sp}(t-1)}{\Delta S_i^{sp}(t-1)} \left( \frac{\frac{N_i^{sp}(t-3)}{S_i^{sp}(t-3)}}{1+\frac{N_i^{sp}(t-3)}{S_i^{sp}(t-3)}} \right)} \right]$$

En utilisant les mêmes arguments que pour le $\Delta\Delta$*cepstre*, on obtient la formule suivante :

$$\mu^i_{\Delta^2 R_{sp}} \approx \alpha_i \mu^i_{\Delta^2 S_{sp}} + (1-\alpha_i)\mu^i_{\Delta^2 N_{sp}} \tag{B.13}$$

Les composantes de la matrice de covariance de $\Delta R_{sp}$ dans ce cas s'écrivent :

$$\sigma^{ij}_{\Delta^2 R_{sp}} \approx \alpha_i \alpha_j \sigma^{ij}_{\Delta^2 S_{sp}} + (1-\alpha_i)(1-\alpha_j)\sigma^{ij}_{\Delta^2 N_{sp}} \tag{B.14}$$

avec

$$\alpha_i = \frac{1}{1 + \frac{\mu^i_{\Delta N_{sp}} \mu^i_{N_{sp}}}{\mu^i_{\Delta N_{sp}} \mu^i_{N_{sp}}}} \tag{B.15}$$

# Annexe C

# Filtrage de Wiener

## C.1 L'erreur du filtrage de Wiener dans le domaine fréquentiel

Nous rappelons tout d'abord le filtrage optimal de Wiener tel qu'il a été conçu par Wiener pour séparer deux signaux aléatoires stationnaires au 2$^e$ ordre, ergodiques, centrés. La relation qui existe entre le signal propre $s(t)$ et le signal bruité $x(t)$ est donnée comme suit :

$$x(t) = s(t) + b(t) \tag{C.1}$$

où $s(t)$ et $b(t)$ sont deux signaux stationnaires au 2$^e$ ordre, ergodiques, centrés, réels.

Le critère de qualité est l'écart quadratique moyen entre le signal propre $s(t)$ et le signal $x'(t)$ ($x'(t)$ est le résultat du filtrage de $x(t)$ ). On se propose donc de chercher le filtre linéaire et invariant dans le temps (de réponse impulsionnelle $h$) qui, appliqué à $x(t)$, minimise l'écart quadratique moyen entre $x'(t)$ et $s(t)$. Il faut bien préciser que l'écart quadratique moyen est optimisé dans le domaine temporel. L'écart quadratique moyen s'écrit :

$$e^2 = E\left\{|x(t) * h(t) - s(t)|^2\right\} \tag{C.2}$$

$E(X)$ désigne l'espérance de $X$. $*$ désigne le produit de convolution. $e^2$ s'écrit :

$$e^2 = E\left\{\left|\int_{-\infty}^{+\infty} x(t-\theta)h(\theta)d\theta - s(t)\right|^2\right\} \tag{C.3}$$

Ceci peut s'écrire :

$$e^2 = E_1 + E_2 + E_3 \tag{C.4}$$

avec

$$\begin{aligned} E_1 &= E\left\{\left|\int_{-\infty}^{+\infty} x(t-\theta)h(\theta)d\theta\right|^2\right\} && \text{(C.5)} \\ E_2 &= E\left\{|s(t)|^2\right\} && \text{(C.6)} \\ E_3 &= -2E\left\{s(t)\int_{-\infty}^{+\infty} x(t-\theta)h(\theta)d\theta\right\} && \text{(C.7)} \end{aligned}$$

$$\tag{C.8}$$

$$
\begin{aligned}
E_1 &= E\left\{\int\int_{-\infty}^{+\infty} h(\theta)h(\sigma)x(t-\theta)x(t-\sigma)d\theta d\sigma\right\} \\
&= \int\int_{-\infty}^{+\infty} E\left\{h(\theta)h(\sigma)x(t-\theta)x(t-\sigma)\right\} d\theta d\sigma \\
&= \int\int_{-\infty}^{+\infty} h(\theta)h(\sigma)R_{xx}(\theta-\sigma)d\theta d\sigma \qquad (C.9)
\end{aligned}
$$

où $R_{xy}$ désigne la fonction de corrélation de $x$ et $y$. Posons :

$$\theta - \sigma = -u$$

$E_1$ s'écrit dans ce cas :

$$
\begin{aligned}
E_1 &= \int_{-\infty}^{+\infty} R_{xx}(-u)R_{hh}(u)du \\
&= \int_{-\infty}^{+\infty} R_{xx}(u)R_{hh}(u)du \qquad (C.10)
\end{aligned}
$$

$$
\begin{aligned}
E_3 &= -2E\left\{s(t)\int_{-\infty}^{+\infty} x(t-\theta)h(\theta)d\theta\right\} \\
&= -2E\left\{\int_{-\infty}^{+\infty} s(t)x(t-\theta)h(\theta)d\theta\right\} \\
&= -2\int_{-\infty}^{+\infty} h(\theta)E\left\{s(t)x(t-\theta)\right\} d\theta \\
&= -2\int_{-\infty}^{+\infty} h(\theta)R_{sx}d\theta \qquad (C.11)
\end{aligned}
$$

D'après Parseval

$$
\begin{aligned}
E_1 &= \int_{-\infty}^{+\infty} \Gamma_{xx}(\nu)|H(\nu)|^2 d\nu \qquad (C.12) \\
E_2 &= \int_{-\infty}^{+\infty} \Gamma_{ss}(\nu)d\nu \qquad (C.13) \\
E_3 &= -2\int_{-\infty}^{+\infty} \Gamma_{sx}(\nu)H(\nu)d\nu \qquad (C.14)
\end{aligned}
$$

où $\Gamma_{xy}$ désigne la Transformée de Fourier de $R_{xy}$, il s'agit de la densité spectrale d'interaction. $\Gamma_{xx}$ est la densité spectrale d'interaction du signal $x$ avec lui même, appelée habituellement “la densité spectrale de $x$”.

Puisque $b(t)$ et $s(t)$ sont non corrélés alors on a :

$$
\begin{aligned}
\Gamma_{xx}(\nu) &= \Gamma_{ss}(\nu) + \Gamma_{bb}(\nu) \\
\Gamma_{sx}(\nu) &= \Gamma_{ss}(\nu)
\end{aligned}
$$

donc

$$e^2 = \int_{-\infty}^{+\infty} \left[\Gamma_s(\nu) + \Gamma_s(\nu)H(\nu) + (\Gamma_s(\nu) + \Gamma_b(\nu))|H(\nu)|^2\right] d\nu. \tag{C.15}$$

La minimisation de $e^2$ revient à minimiser pour chaque $\nu$ la quantité suivante :

$$e^2(\nu) = \Gamma_s(\nu) + \Gamma_s(\nu)H(\nu) + (\Gamma_s(\nu) + \Gamma_b(\nu))|H(\nu)|^2. \tag{C.16}$$

La formule précédente montre que la minimisation de l'erreur quadratique moyenne dans le domaine temporel n'implique pas la minimisation de l'erreur quadratique moyenne dans le domaine fréquentiel. A titre de vérification, en supposant que le filtre recherché est réel, la minimisation de $e^2(\nu)$ donne exactement le filtre de Wiener classique.

# Bibliographie


[1] A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition", *thesis Department of Electrical and Computer Engineering, CMU university,* Sept. 1990.

[2] A. Acero, R.M. Stern, "Environmental Robustness in Automatic Speech Recognition", *ICASSP-90*, pp. 849-852.

[3] P. Alexandre, P. Lockwood, "Root cepstral analysis: A unified view. Application to speech processing in car noise environments", *Speech Communication* Vol. 12, pp. 277-288, 1993.

[4] T.H. Applebaum, B.A. Hanson, "Regression Features for Recognition of Speech in Quiet and in Noise", *ICASSP-91,* pp. 985-988.

[5] B. Atal, "Effectiveness of Linear Prediction Characteristics of Speech Wave for Automatic Speaker Identification and Verification", *Journal of the Acoustic Society of America*, **55**, pp. 1304-1312, June 1974.

[6] L.R. Bahl, P.V. de Souza, P.S Gopalkrishnan, D. Nahamoo, and M.A Picheny, "Context dependent modelling of phones in continuous speech using decision trees", *In Proceedings DARPA Speech and Natural Language Processing Workshop,* pp. 264-270, 1991.

[7] L.E. Baum, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *Ann. Math. Statist.,* 41 (1970), pp. 164-71.

[8] M. Berouti, R. Schwartz, J. Makhoul, "Enhancement of Speech Corrupted by Acoustic Noise", *ICASSP-79,* pp. 208-211.

[9] S.F Boll, " Suppression of Acoustic Noise in Speech Using Spectral Subtraction", *IEEE*, pp. 113-120, 1979.

[10] O. Cappé, "Techniques de réduction de bruit pour la restauration d'enregistrements musicaux", *thèse TELECOM Paris 93 E 019*, sept 93.

[11] Y.M. Cheng, D. O'Shaughnessy, "Speech Enhancement Based Conceptually on Auditory Evidence", *IEEE trans. on ASSP,* 39(9):1943-1954, September 1991.

[12] D. V. Compernolle, "Noise adaptation in a hidden Markov model speech recognition system", *Computer Speech & Language*, pp. 151-167, 1989.

[13] D. V. Compernolle, "Spectral Estimation Using a Log-Distance Error Criterion Applied to Speech Recognition", *ICASSP-89,* pp.258-261.

[14] S. Das, R. Bakis, A. Nadas, D. Nahamoo, M. Picheny, "Influence of Background Noise and Michrophone on the Performance of the IBM TANGORA Speech Recognition System", *ICASSP-93,* Vol. 2, pp. 71-74.

[15] A.P. Dempster, N.M Laird, B.D. Rubin, "Maximum Likelihood from Incomplete Data via the EM algorithm", *Journal of the Royal Statistical Society,* 39:1-38, 1977.

[16] H. Drucker, "Speech Processing in a High Ambient Noise Environment", *IEEE Trans. Audio and Electroacoustics,* Vol. AU-16, pp. 165-168, June 1968.

[17] ESCA - NATO, Tutorial and Research Workshop on "Robust Speech Recognition for Unknown Communication Channels", *Pont-à-Mousson, France* 17-18 April 1997.

[18] Y. Ephraim, D. Malah, B.H Juang, "On the Application of Hidden Markov Models for Enhancing Speech", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 37. NO. 12. December 1989.

[19] H.F. Fletcher, W.A. Munson, "Loudness, its Definition, Measurement and Calculation", *J. Acoust. Soc. Am.,* Vol. 5, 1933.

[20] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", *IEEE Trans. Acoustics, Speech and Signal Processing ASSP-29(2):254-272,* Apr., 1981.

[21] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum", *IEEE Trans. Acoustics, Speech and Signal Processing ASSP-34:52-59,* 1986.

[22] M. Gales, S. Young, "An improved approach to hidden Markov model decomposition of speech and noise", *ICASSP-92*.

[23] M.J.F. Gales, S.J. Young, "Robust Continuous Speech Recognition using Parallel Model Combination", *Computer Speech & Language*, **9**(4), Oct. 1995.

[24] M.J.F. Gales, S.J. Young, "A fast and flexible implementation of parallel model combination", *ICASSP-95*, pp. 133-136.

[25] M.J.F. Gales, "Model-Based Techniques for Noise Robust Speech Recognition", *thesis Gonville and Caius College*, Sept 95.

[26] J.L. Gauvain, L. Lamel, G.Adda, D.Matrouf, "Developments in Continuous Speech Dictation using the 1995 ARPA NAB News Task", *ICASSP-96*, pp. 73-76.

[27] O. Ghitza, "Auditory nerve representation as a front end for speech recognition in noisy environement", *Computer, Speech and Language,* (1986) Vol. 1, NO. 2, pp. 109-130.

[28] Y. Gong, "Base Transformation for Environment Adaptation in Continuous Speech Recognition", *European Conference on Speech Communication and Technology,* September 1993.

[29] Y. Gong, J-P. Haton, "Stochastic trajectory modeling for speech recognition", *ICASSP-94*, Vol. 1, pp. 57-60.

[30] Y. Gong, W.C Treurniet, "Speech Recognition in Noisy Environments: a survey", *CRC-TN 93-002*, June, 1993.

[31] R.M. Gray "Toeplitz and Circulant Matrices: A review", *Stanford Electron. Lab., CA, Tech. Rep. 6504-1,* Apr. 1977

[32] B.A. Hanson, T.H. Applebaum, "Robust Speeker-independent Word Recognition using Static, Dynamic and Accelaration Features: Experiments with Lombard and Noisy Speech", *ICASSP-90,* pp. 857-860.

[33] B.A. Hanson, T.H. Applebaum, "Subband or Cepstral Domain Filtering for Recognition of Lombard and Channel-distored speech", *ICASSP-93,* Vol. 2, pp 79-82.

[34] H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis", *Journal Acoust. Soc. Am.,* Vol. 87, NO. 4, pp. 1738-1752, 1990.

[35] H. Hermansky, N. Morgan, A. Bayya, P. Kohn, "Compensation for the Effect of Communication Channel in Auditory-like Analysis of Speech (RASTA-PLP)", *European Conference on Speech Technology,* pp. 1367-1370, 1991.

[36] J. Hernando, C. Nadeu, "A comparative study of parameters and distances for noisy speech recognition", *In Proceedings Eurospeech,* pp. 91-94, 1991.

[37] J.N. Holmes, N.C. Sedgwick, "Noise Compensation for Speech Recognition using Probabilistic Models", *ICASSP-86,* pp.741-744.

[38] X. Huang, A. Acero, F. Alleva, M-Y. Hwang, L. Jiang, M. Mahajan "Microsoft windows highly intelligent speech recognizer: whisper", *ICASSP-95*, pp. 93-96.

[39] B.H. Juang, "Speech recognition in adverse environments", *Computer Speech and Language,* (1991)5, pp. 275-294.

[40] B.H. Juang, “Maximum-Liklihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains”, *AT&T Technical Journal,* pp. 1235-1249, July-August 1985.

[41] S.M. Kay, “The effects of noise on autoregressive spectral estimator”, *IEEE Trans. Acoust. Speech Signal Process.,* Vol. 27, No. 5, October 1979.

[42] L. Lamel, S. Rosset, S. Bennacef, H. Bonneau-Maynard, L. Devillers, J.L. Gauvain, “Development of Spoken Language Corpora for Travel Information”, *Eurospeech'95*, pp. 1961-1964.

[43] K.F. Lee, H.W. Hon, “Large-Vocabulary Speaker-Independant Continuous Speech Recognition Using HMM”, *ICASSP-88,* pp. 123-126.

[44] C. Lefebvre, D. Zwierzynski, D. Starks, G. Birch, “Further Optimization of Robust IMELDA Speech Recogniser for Applications with Severely Degraded Speech”, *ICSLP-92,* Vol. 1, pp. 691-694.

[45] C.J. Leggetter, P.C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models”, *Computer Speech & Language*, pp. 171-185, 1995.

[46] S. E. Levinson, “Continuously variable duration hidden Markov models for automatic speech recognition”, *Computer Speech and Language 1986.*

[47] J.S. Lim, A.V. Oppeenheim, “All-pole modeling of degraded speech”, *IEEE Trans. Acoust., Speech, Siganl Process.,* Vol. 26, pp. 197-210, June 1978.

[48] R.P. Lippmann, E.A. Martin, D.B. Paul, “Multi-style training for robust isolated-word speech recognition”, *ICASSP-87*, pp. 705-708.

[49] P. Lockwood, J. Boudy, M. Blanchet: “Non-linear Spectral Subtration (NSS), and Hidden Markov Models for robust speech recognition in car noise environments”, *ICASSP-92.*

[50] B. T. Logan, A. J. Robinson, “Enhancement and Recognition of Noisy Speech within an AutoRegressive HMM Framework Using Noise Estimates from The Noisy Signal”, *ICASSP-97.*

[51] R.J. McAulay, M.L Malpass, “Speech Enhancement using a Soft-decision Noise Suppression filter”, *IEEE Trans. Acoust., Speech, Signal Process.,* Vol. 28, pp. 137-145, Apr 1980.

[52] D. Mansour, B.H Juang, “A Family of Distortion Measures Based Upon Projection Operation for Robust Speech Recognition”, *IEEE Trans. on Acoust., Speech and Signal Processing,* Vol. 37, No. 11, 1989.

[53] D. Mansour, B.H Juang, “The short-time modified coherence representation and Noisy speech recognition”, *IEEE Trans. on Acoust., Speech and Signal Processing,* Vol. 37, No. 6, 1989.

[54] J.D. Markel, A.H. Gray, Jr., "Linear Prediction of Speech", *Communication and Cybernetics, Springer-Verlag,* Berlin Heidelberg New York 1976.

[55] F. Martin, K. Shikano, Y. Minami, "Recognition of Noisy Speech by Composition of Hidden Markov Models", *EuroSpeech'93.*

[56] D. Matrouf, J.L. Gauvain, "Model compensation for noises in test and training data", *ICASSP-97.*

[57] D. Matrouf, J.L. Gauvain, "Techniques de Compensation pour La Reconnaissance de la Parole Bruitée", *JEP-96*, pp. 331-334.

[58] S. Muzita, K. Nakajima, "Optimal Discriminative Training for HMMs to Recognize Noisy Speech", *ICSLP-92,* Vol. 2, pp. 1519-1522.

[59] C. Mokbel, "Reconnaissance de la Parole dans le Bruit : Bruitage / Débruitage", *thèse TELECOM Paris 92 E 008*, Juin 1992.

[60] S. Morii, T. Morii, M. Hoshimi, S. Hiroaka, T. Watanabe, K. Niyada, "Noise Robustness in Speeker Independent Speech Recognition", *ICSLP-90,* pp. 1145-1148.

[61] A. Nádas, D. Nahamoo, M.A Picheny, "Speech Recognition using Noise-adaptative Prototypes", *IEEE trans. on ASSP,* Vol. 29, NO. 3, pp. 351-363, Oct. 1989.

[62] L. Neumeyer, M. Weintraub, "Probabilistic Optimum Filtring for Robust Speech Recognition", *ICASSP-94,* Vol. 1, pp. 417-420.

[63] J. A. Nolazco, S. J. Young, "CSS-PMC: a Combined Enhancement / Compensation Scheme for Continuous Speech Recognition in Noise", *In Technical Report, Cambridge University Engineering Dept., Speech Group,* CUED/F-INFENG/TR.128 June 1993.

[64] K. Ohkura, D. Rainton, M. Sugiyama, "Noise-Robust HMMs Based on Minimum Error Classifier", *ICASSP-93*, Vol. 2, pp. 75-78.

[65] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, A.F. Martin, M.A. Przybocki, "1995 Hub-3 Multiple Microphone Corpus Benchmark Tests", *Proc. ARPA Speech Recognition Workshop,* Harriman, NY, Fév. 1996.

[66] A.B. Poritz, "Linear predictive hidden Markov models and the speech signal", *ICASSP-82,* pp. 1291-1294.

[67] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Computer, Speech and Language,* (1991)5, pp. 275-294.

[68] L.R. Rabiner, B. Gold, "Theory and application of digital signal processing",

[69] P.K. Rajasekaran, G.R. Doddington, J.W. Picone, J.W. Picone, "Recognition of speech under and in noise", *ICASSP-86*, pp. 733-736.

[70] R. A. Redner, H. F. Walker, "Mixture densities, Maximum Likelihood and the EM Algorithm", *Society for Industrial and Applied Mathematics,* Vol. 26, No. 2, April 1984.

[71] D.B. Roe "Speech Recognition with a Noise-adapting Codebook", *ICASSP-87,* No. 27.5.1, pp. 1139-1142.

[72] A. Sankar, C.-H. Lee, "Robust Speech Recognition based on Stochastic Matching", *ICASSP-95*, pp. 121-124.

[73] C.W. Seymour, M. Niranjan, "An HMM based Cepstral-domain speech enhancement scheme", *ICSLP-94,* pp. 1595-1598.

[74] O. Siohan, Y. Gong, J-P. Haton, "Noise adaptation using linear regression for continuous noisy speech recognition", *EuroSpeech'95*, pp. 465-468.

[75] R.M. Stern, A. Acero, "Acoustical Pre-processing for Robust Speech Recognition", *Speech and Natural Language Workshop, Cape Cod, MA,* pp.311-318, Morgan Kaufmann, Oct 1989.

[76] S.S. Stevens, "On The Psychological Law", *Psychological Review* 64, 1957.

[77] N. Suaudeau, R. Andre-obrecht, "An efficient combination of acoustic and supra-segmental informations in a speech recognition system", *ICASSP-94,* pp. 65-68, Vol. 1.

[78] K. Tokuda, T. Kobayashi, S. Imai, "Generalized cepstral analysis of speech-unified approach to LPC and cepstral method", *ICSLP-90.*

[79] J.T. Tou, R.C. Gonzalez "Tattern Recognition Principles", *Addision-Wesley Publishing Company, Advanced Book Program/World Science Division,* pp. 291-298, 1974

[80] A.P. Varga, R.K. Moore, "Hidden Markov model decomposition of speech and noise", *ICASSP-90*, pp. 845-848.

[81] A.P. Varga et al. "The NOISEX-92 study on the effect of additive noise on automatic speech recognition", *In Technical Report, DRA Speech Research Unit,* 1992.

[82] A.J. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm", *IEEE Transactions Information Theory,* 13:260-269, 1982.

[83] D.L. Wang, J.S. Lim, "The Unimportance of Phase in Speech Enhancement", *IEEE Trans. ASSP-82* Vol. 30, No. 4, 1982.

[84] S.J Young, J.J. Odell, P.C Woodland, "Tree-based state tying for high accuracy acoustic modelling", *In Proceedings ARPA Workshop on Human Language Technology,* pp. 307-312, 1994.

[85] S.J Young, P.C Woodland, "The use of state tying in continuous speech recognition", *In Proceeding Eurospeech,* pp. 2207-2210, 1993.

[86] Y. Zaho, "Self-learning speaker adaptation based on spectral variation source decomposition", *Eurospeech'93,* pp. 359-362.

[87] E. Zwicker, R. Feldtkeller, "Psychoacoustique : L'oreille, récepteur d'information", *Masson* 1981.