Speech quality enhancement for mobile radio systems by using a priori information at the receiver side

Christophe Veaux

Thèse

présentée pour obtenir le grade de :

Docteur de l'Ecole Nationale Supérieure

des Télécommunications

Spécialité : Traitement du Signal et des Images

Christophe Veaux

Laboratoire d'accueil : France Télécom R&D – DIH/IPS, Lannion

Étude de traitements en réception pour l'amélioration de la qualité de la parole — Application au GSM

Soutenue le 20 janvier 2005 devant la commission d'examen :

Nicolas Moreau	ENST Paris	Président du Jury
Samir Saoudi	ENST Bretagne	Rapporteur
Pierre Siohan	France Télécom R&D	Rapporteur
André Gilloire	France Télécom R&D	Examinateur
Pascal Scalart	IRISA Lannion	Examinateur

Remerciements

J'ai touché le point final, enfin, au bout de tant d'autres restés en suspension. Cette thèse au long cours s'achève et je tiens ici à remercier toutes les personnes qui m'ont accompagné, soutenu et bien souvent donné d'elles-mêmes durant toutes ces années.

En premier lieu, je ne saurais assez dire ma profonde gratitude envers André Gilloire et Pascal Scalart auxquels le bon aboutissement de cette thèse doit beaucoup. Ils furent tout d'abord des encadrants qui m'ont marqué par leur ouverture d'esprit, leur enthousiasme et leur compétence, alliant curiosité et rigueur scientifique. Je les remercie également pour leurs qualités humaines certaines, leur disponibilité et leur écoute qui ne se sont jamais démenties, et pour toute l'énergie consacrée. Je leur sais gré de m'avoir maintenu leur confiance et j'espère que ce document saura payer leur bienveillance de retour.

J'aimerais ensuite remercier Nicolas Moreau qui m'a fait l'honneur de présider le jury de ma soutenance. L'intérêt qu'il a manifesté à cette occasion pour ce travail de thèse dont il n'avait reçu pour tout écho que les demandes répétées de sursis, est pour moi une vraie source de gratification.

Je remercie chaleureusement Pierre Siohan et Samir Saoudi d'avoir accepté de rapporter ce mémoire. La pertinence de leurs remarques et la précision de leurs corrections m'ont permis d'en améliorer la clarté.

Je tiens à remercier ici M. Bernard Robinet, directeur de l'EDITE, pour m'avoir accordé ces dernières années plusieurs dérogations et avoir autorisé la tenue de ma soutenance de thèse.

Je suis reconnaissant envers Dominique Massaloux de m'avoir accueilli au sein du laboratoire DIH/IPS et permis l'achèvement de cette thèse dans les meilleures conditions possibles. Je remercie Claude Lamblin, Catherine Quinquis et Balasz Kovesi pour leur aide et leurs encouragements tout au long de cette thèse et de la rédaction du manuscrit. Une pensée également pour Janine Denmat et son dévouement quotidien.

J'aurais un remerciement particulier pour Vincent Barriac qui m'a accueilli dans son laboratoire ces dernières années et m'a permis d'aménager mon emploi du temps pour faciliter la poursuite de la rédaction de ce manuscrit.

Je garde un souvenir chaleureux de l'équipe de permanents et de thésards qui m'accueilli à l'ère primaire de ma thèse. Je pense à Rozenn et son indéfectible amitié, à Claude et Marco à qui je fournis pour leur plaisir quelques occasions de mise en boite, à Christophe Beaugeant qui me fit découvrir les joies du théâtre d'avant-garde lannionais. Merci aussi à Valérie Turbin et à David Deléam avec lesquels je me compromis gaiement dans des spectacles un peu moins avant-gardistes mais la découverte du milieu local est parfois à ce prix.

Le tournant du millénaire amena de nouvelles et belles rencontres mais marqua subrepticement pour moi une ère de fossilisation rampante dans une rédaction sans fin. Tous ceux-là eurent à supporter avec moi cette gangue pesante. Je ne sais si la consistance d'un merci tient encore face à leurs efforts conjugués pour m'en extirper. Sachez que votre présence a été un soutien bien plus puissant que vous ne l'imaginez.

Ici, pensée spéciale au "bureau des émotions" tenu par Léti et Valérie, en première ligne de front au quotidien et dont l'empathie sensible et vibrante encaissa tant et encore. Merci à Bobo et son harcèlement régulier dans la dernière ligne droite (s'il y en eu); Léna pour ses visites aux ermites locaux et ses billets d'humeur dans ma boite à mails.

Merci à Noël pour sa dissonance salutaire; à Rapha pour ses caricatures bien senties.

Hervé, Gaël, et Nico furent d'agréables compagnons de voyage quoique peu ponctuels. Le tremblement de rire Rochien, celui vif et narquois de Karin; celui sonore et franc de Marion sans oublier le "mais euh" de Béné résonnent encore à mon cœur comme autant d'échos réjouissants.

Et puis encore, pour votre amitié, vos sourires, et nos souvenirs partagés, merci à David et Elodie, Erwan et Zanou, Cécile, Hélène, Sylvie, Fred et Valérie, David et Nath, JP et Sophie.

Enfin, je songe à la sphère familiale qui malgré la distance (ou peut-être à cause d'elle) a beaucoup porté en plus de me soutenir. Un merci particulier également à mes supporters lyonnais, Marie-Thérèse et René, dont l'intérêt m'a beaucoup touché.

Merci à tous.

Table des Matières

Glossaire

AK0	0th order A priori Knowledge (*modèle a priori d'ordre 0*)
AK1	First-order A priori Knowledge (*modèle a priori d'ordre 1*)
AK2	Second-order A priori Knowledge (*modèle a priori d'ordre 2*)
APRI-VA	A Priori Viterbi Algorithm (*décodage de Viterbi avec a priori*)
AR	Auto Regressif
BFI	Bad Frame Indicator (*indicateur de trame perdue*)
C/I	Carrier to Interference ratio (*rapport porteuse sur interférences*)
CELP	Code Excited Linear Prediction (*prédiction linéaire excitée par codes*)
CRC	Cyclic Redundancy Check (*détection d'erreur par code cyclique*)
EFR	Enhanced Full Rate (*codeur de parole plein débit amélioré*)
GMM	Gaussian Mixture Model (*modèle par mélange de gaussiennes*)
GSM	Global System for Mobile
LPC	Linear Predicitve Coefficient (*coefficient de prédiction linéaire*)
LSF	Lignes Spectrales de Fréquence
LSP	Lignes Spectrales par Paires
LTP	Long-Term Prediction (*prédiction à long-terme*)
MA	Moyenne Ajustée
MAP	Maximum A Posteriori
MMSE	Minimum Mean Square Error (*minimum d'erreur quadratique moyenne*)
MOS	Mean Opinion Score
MV	Maximum de Vraisemblance
PESQ	Perceptual Evaluation of Speech Quality
QV	Quantification Vectorielle
SBSD	Soft-Bit Source Decoding (*décodage source à entrées souples*)
SCCD	Source Controlled Channel Decoding (*décodage canal contrôlé par la source*)
SNR	Signal to Noise Ratio (*rapport signal à bruit*)
SOVA	Soft Output Viterbi Algorithm (*décodage de Viterbi à sorties pondérées*)
TEB	Taux d'Erreur Binaire
TU	Typical Urban (*canal urbain typique*)

Introduction Générale

A ses premiers temps, la téléphonie mobile a dû son essor extraordinaire au seul fait d'apporter une liberté nouvelle. La possibilité inconnue jusqu'alors de communiquer quel que soit l'endroit où l'on se trouve a rendu les utilisateurs de ces systèmes assez indulgents quant à la qualité vocale offerte. En effet, coupures, voix métallique et autres sons artificiels sont les signatures caractéristiques de ces communications et viennent rappeler qu'on ne coupe pas si impunément le cordon fixe. Cependant, le téléphone mobile est de nos jours un objet du quotidien et à l'attrait de la nouveauté succède l'exigence d'une qualité la plus proche possible de celle du téléphone fixe. Ainsi, la qualité de la parole restituée par les mobiles est désormais devenue un enjeu central.

La parole transmise par un réseau radiomobile subit de nombreuses dégradations à des niveaux successifs. La première d'entre-elles est la distorsion introduite par le codeur de parole et qui résulte d'un compromis posé dès la conception du système entre la qualité vocale escomptée et les ressources du réseau allouées à sa transmission. Cette distorsion peut être aggravée par la présence de sources acoustiques interférentes, comme le bruit, puisque le codeur est spécialisé pour un type de source donnée (la parole). Les mécanismes du réseau radiomobile, comme ceux destinés à économiser les ressources radio (transmission discontinue) ou à gérer le transfert inter-cellulaire d'un mobile, mutilent les informations vocales transmises par le mobile et se traduisent souvent par des artefacts dans la parole restituée en sortie de décodeur. Enfin, la liaison radio introduit des erreurs de transmission qui peuvent affecter la parole au point de la rendre inintelligible.

Face à cette diversité de facteurs, il apparaît que la recherche d'une meilleure qualité vocale passe également par des réponses à plusieurs niveaux. Ceci motive notamment l'étude de nouvelles normes de codage de parole afin par exemple, d'en améliorer la robustesse aux entrées bruitées tout comme aux erreurs de transmission. Parallèlement à ces travaux qui supposent des modifications importantes de l'émetteur et de la norme du système, il a semblé intéressant de rechercher des traitements pouvant améliorer la qualité d'un système existant – le GSM – au prix de modifications limitées.

Plus précisément, l'objectif initial de cette thèse était de mettre en œuvre des post-traitements de la parole, situés en aval de la chaîne de réception du GSM. Cette position offrait une vision globale des dégradations introduites par le réseau radio-mobile et permettait une mise en œuvre aisée sur des plate-formes centralisées de traitement du signal.

Derrière ces arguments pratiques perçait également l'ambition d'une méthode universelle, c'est-à-dire pouvant traiter une large diversité de dégradations inconnues. L'inspiration lointaine était le mécanisme de la perception humaine elle-même qui peut détecter une dégradation dans un signal de parole sans aucune autre référence qu'un modèle interne *a priori* de la parole. Après de premières études sur la détection d'artefact dans le signal de parole restitué par le système GSM, il est apparu que la marge de manœuvre offerte à un traitement d'amélioration de la qualité en aval des décodeurs GSM était extrêmement limitée par les mécanismes de masquage mis en oeuvre dans ces mêmes décodeurs. D'autre part, la détection d'artefacts sans autre information que celle issue d'un modèle *a priori* de la parole conduit à un taux de fausses alarmes trop élevé.

Parallèlement, l'idée d'utiliser un modèle *a priori* au niveau du décodeur parole, conjointement avec une information issue du canal, apparaissait dans des recherches sur le masquage « intelligent » des erreurs de transmission. La possibilité d'exploiter une information *a priori* au décodeur peut surprendre à première vue puisque le codeur parole est sensé éliminer toute redondance. Cependant, les contraintes de complexité et de délai font qu'il subsiste en réalité une *redondance résiduelle* en sortie du codeur parole. C'est cette redondance résiduelle qui est modélisée au décodeur.

Le principe consistant à exploiter la redondance résiduelle lors du décodage nous a paru très intéressant car il laisse entrevoir la possibilité d'améliorer la qualité vocale en présence d'erreurs de transmission sans pour autant nécessiter de modifications de la norme du système radiomobile. Ceci nous a conduit à redéfinir la problématique de nos travaux en l'orientant sur des aspects décodage conjoint source-canal, et plus précisément, sur l'exploitation de la redondance résiduelle en réception pour améliorer la robustesse aux erreurs de transmission.

Cette problématique est assez large et laisse volontairement ouverte la question du niveau (décodeur parole ou décodeur canal) auquel exploiter la redondance résiduelle. En effet, on peut distinguer deux approches. La première exploite la redondance résiduelle au niveau du décodeur canal afin de réduire le taux d'erreur binaire. La seconde effectue l'estimation optimale des paramètres du codeur parole afin de minimiser l'impact subjectif des erreurs (masquage « intelligent »).

L'argument en faveur de la première approche est qu'il est préférable d'exploiter simultanément toutes les redondances disponibles (résiduelle et canal) pour la protection aux erreurs. En revanche, en se plaçant au niveau des bits, elle ne modélise qu'une faible partie de la redondance résiduelle disponible en sortie du codeur parole. A contrario, la seconde approche modélise la redondance résiduelle directement au niveau des index de quantification. De plus, elle permet de minimiser le critère qui nous intéresse au final, à savoir la distorsion de la parole décodée.

Nous avons choisi d'étudier successivement ces deux approches ainsi que leur combinaison. Une spécificité de notre contexte d'étude a été d'aborder ces travaux théoriques en essayant de les adapter au cadre très concret qui est celui du système GSM existant. Ceci nous a conduit d'une part, à rechercher des réponses aux problèmes posés par la complexité des méthodes. D'autre part, un certain nombre des améliorations proposées ont avant tout eu pour but de contourner les contraintes imposées par un système réel de radio-communications.

Indépendamment de ces contraintes système, nous avons également cherché à améliorer les performances intrinsèques des méthodes pour chacune des deux approches. Cette amélioration passe dans les deux cas par une meilleure modélisation de la redondance résiduelle. Au niveau du décodeur canal, nous avons ainsi proposé des techniques permettant d'exploiter la corrélation entre les bits d'une même trame durant le processus de décodage. Au niveau du décodeur parole, nous modélisons la redondance résiduelle directement dans l'espace des paramètres, ce qui permet de réduire la complexité de l'estimation au décodeur et délivre une information plus riche sur cette redondance résiduelle.

Organisation du document

Ce document présente un nombre important d'annexes, elles ont pour but de décharger la lecture de la partie centrale de cette thèse, tout en offrant la possibilité d'un niveau de lecture plus détaillé.

Le chapitre 1 pose à un niveau plus approfondi la problématique de cette thèse. Les principes de la transmission de la parole dans un système tel que le GSM sont en premier lieu rappelés. Les motivations de cette étude sont justifiées par une analyse des principaux artefacts rencontrés dans la parole restituée par le GSM EFR. Enfin, on y présente les axes de recherche étudiés ainsi que les critères d'évaluation des algorithmes développés.

Le chapitre 2 décrit les études menées sur les post-traitements de la parole et notamment sur l'étape préalable de détection d'artefacts. Nous discutons ensuite des potentialités de cette approche dans le contexte du GSM et justifions la redéfinition de notre axe d'étude.

Les chapitres 3 à 5 sont consacrés à l'approche de décodage conjoint au niveau du décodeur parole. Un état de l'art des méthodes se rattachant à cette approche est tout d'abord présenté. La mise en œuvre de ces méthodes nécessite un algorithme de décodage canal à sorties souples dont le principe est détaillé en annexe D. Nous analysons ensuite, au chapitre 4, l'information apportée par cette sortie souple ainsi que la redondance résiduelle laissée par le codeur de parole EFR. Enfin, nous développons, au chapitre 5, des propositions d'algorithmes visant à réduire la complexité et à améliorer les performances du décodage.

Les chapitres 6 et 7 sont consacrés aux méthodes de décodage canal contrôlé par la source. Après un état de l'art des diverses techniques développées, nous proposons des modifications permettant une meilleure prise en compte de la redondance entre bits d'une même trame. Enfin, nous abordons la combinaison des approches exploitant la redondance résiduelle au niveau du décodeur parole et au niveau du décodeur canal.

Chapitre 1

Contexte et problématique

1.1 Introduction

Les dégradations de la qualité vocale observées sur la parole transmise par le GSM ont des origines multiples, allant des conditions de prise de son [Beaugeant, 1999] aux problèmes de gestion de l'itinérance du mobile par le réseau [Scalart, 1997]. Notre objectif initial était d'élaborer des post-traitements du signal de parole en sortie du réseau GSM capables de traiter une large gamme de ces dégradations. Cependant, il nous est apparu que parmi l'ensemble des dégradations rencontrées, celles liées aux erreurs de transmission sur le canal radiomobile étaient les plus déterminantes vis-à-vis de la qualité de parole du GSM. Aussi, la présentation du contexte qui est faite ici, insiste plus particulièrement sur la problématique de la transmission sur un canal bruité. Notre objectif est ici de dégager les compromis pratiques d'un système de transmission de la parole comme le GSM, d'illustrer les dégradations de la qualité vocale qui en résultent et d'esquisser les axes de recherches pour l'amélioration de cette qualité en réception.

1.2 Problématique d'une transmission numérique

Nous donnons ici un exposé des *principes* mis en œuvre dans une chaîne de transmission numérique actuelle, les divers éléments de la chaîne du système GSM proprement dit sont détaillés dans les Annexes A à C. Nous rappelons des résultats classiques de la théorie des communications mais qui seront utiles pour mettre en perspective les travaux présentés dans ce document.

A la présentation usuelle des divers éléments (fonctionnalités) de la chaîne, abordés dans le sens de la transmission, nous préférons ici repartir de la modélisation du canal et de la source à transmettre avant d'introduire les concepts de codage de source et codage de canal. Cette présentation très générale a pour avantage de mettre en lumière le *principe de séparation* entre codage de source et codage de canal énoncé par Shannon pour un contexte théorique [Shannon, 1948]. Les limitations de ce principe dans les cas pratiques seront illustrées au travers du système GSM.

Considérons le problème de la transmission d'un signal au travers d'un canal bruité. Dans le cas d'une transmission analogique, les perturbations et les bruits apportés par le canal se répercutent inévitablement sur le signal reconstruit en bout de chaîne. En revanche, dans le cas d'une transmission numérique, il est possible de transmettre un message avec un taux d'erreur aussi faible que l'on veut. C'est un résultat de la théorie de l'information développée pour les communications numériques (qui débouche sur le codage canal et codage source). Cette théorie précise les deux points suivants :

- la quantité d'information (par unité de temps) apportée par un message numérique,
- la quantité d'information transmissible par le canal (par unité de temps).

Il convient d'abord de définir le message (ou source) numérique ainsi que le canal de transmission au sens de la théorie de l'information.

1.2.1 La source

Le signal de parole étant par nature analogique, il doit tout d'abord être converti dans l'objectif d'une transmission numérique. La Figure 1.1 rappelle les étapes de cette conversion analogique-numérique où $F_e = 1/T_e$ désigne la fréquence d'échantillonnage.

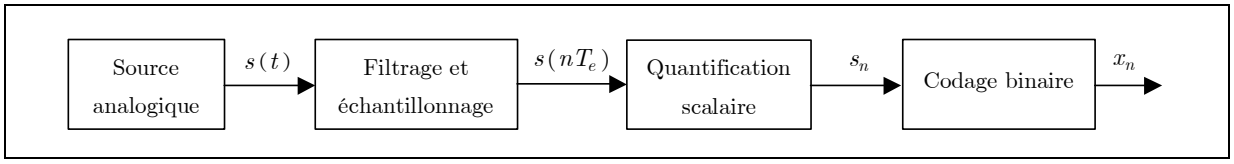

Figure 1.1 : Principe de la numérisation d'une source analogique

Dans le cas de la parole, on distingue couramment deux *gammes de qualités* selon la bande utilisée :

- La *bande téléphonique*, qui correspond à un signal filtré dans la bande $[300 - 3400\ \text{Hz}]$ puis échantillonné à 8 kHz.
- La *bande élargie*, qui correspond à la bande $[50 - 7000\ \text{Hz}]$ et à une fréquence d'échantillonnage de 16 kHz.

Les applications radiomobiles, comme le système GSM, utilisent la bande téléphonique.

La quantification utilisée dans les convertisseurs analogique-numérique est une quantification scalaire de *résolution* suffisamment fine pour limiter le bruit de quantification à un niveau quasiment inaudible. En effet, la numérisation ne vise pas à compresser le signal mais à générer une source numérique de référence (*source non-codée*). Cette source est caractérisée par son *débit binaire D* défini comme le nombre d'éléments binaires émis par unité de temps. En sortie du convertisseur, on a :

$$D = \frac{k}{T_e} \tag{1.1}$$

où k est le nombre d'éléments binaires utilisé pour quantifier chaque échantillon $s(nT_e)$ de la source. Ainsi, la sortie d'un convertisseur analogique-numérique de parole est souvent un signal MIC[1] linéaire correspondant à un signal échantillonné à 8 kHz et quantifié de manière uniforme sur 16 bits. La source numérique ainsi définie a un débit binaire de 128 kbits/s. Dans le cas du GSM, la source numérique est un signal MIC linéaire échantillonné à 8 kHz et quantifié sur 13 bits, ce qui correspond à débit de référence (« source non codée ») de 104 kbits/s.

1.2.2 Transmission sur un canal radio

Considérons un canal de propagation radiomobile caractérisé par sa réponse impulsionnelle *équivalente en bande de base* $h(t)$. Le bruit et les interférences perturbant le canal peuvent être modélisés comme un *bruit additif blanc gaussien* $\eta(t)$ de puissance mono-latérale N_0. On s'intéresse à la transmission d'un message numérique x_n sur ce canal.

Le message x_n, défini comme une suite d'éléments binaires, est un signal abstrait. Pour pouvoir être transmis dans le milieu physique de propagation, ce message doit être véhiculé par un signal physique. C'est l'objet de la *modulation* qui associe une forme d'onde[2] analogique $m(t,\mathbf{x})$ à une séquence d'éléments binaires $\mathbf{x} = [x_1,...,x_n,...]$. En sortie du milieu de transmission, le signal reçu en bande de base et échantillonné à une période T_s, s'écrit :

$$r(nT_s) = h(nT_s) \otimes m(nT_s,\mathbf{x}) + \eta(nT_s) \tag{1.2}$$

où $\otimes$ dénote le produit de convolution.

Le but du récepteur est de retrouver la séquence $\mathbf{x}$ émise à partir de la séquence $\mathbf{r}$ d'échantillons reçus. Le *critère optimal* de détection est celui du *Maximum a Posteriori* (MAP) qui maximise la probabilité $p(\mathbf{x}|\mathbf{r})$:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \{ p(\mathbf{x}|\mathbf{r}) \} = \arg\max_{\mathbf{x}} \{ p(\mathbf{r}|\mathbf{x})\, p(\mathbf{x}) \} \tag{1.3}$$

---

[1] Modulation par impulsions codées

[2] On considère ici la représentation en bande de base de la modulation.

Ce critère exploite donc à la fois une information *a priori* $p(\mathbf{x})$ sur le message et une information *a posteriori* $p(\mathbf{r}|\mathbf{x})$ issue du canal.

La probabilité $p(\mathbf{r}|\mathbf{x})$ peut être calculée en estimant les paramètres du canal $\{h(nT_s);N_0\}$. Une méthode communément utilisée pour estimer ces paramètres est d'émettre un séquence d'apprentissage connue du récepteur[3]. Cette méthode est celle utilisée dans le système GSM (cf. Annexe C). Le schéma de principe de la transmission est alors illustré Figure 1.2. Le récepteur effectue ici implicitement *l'égalisation* et *la démodulation* conjointement à l'exploitation de la redondance du message (« *décodage* »).

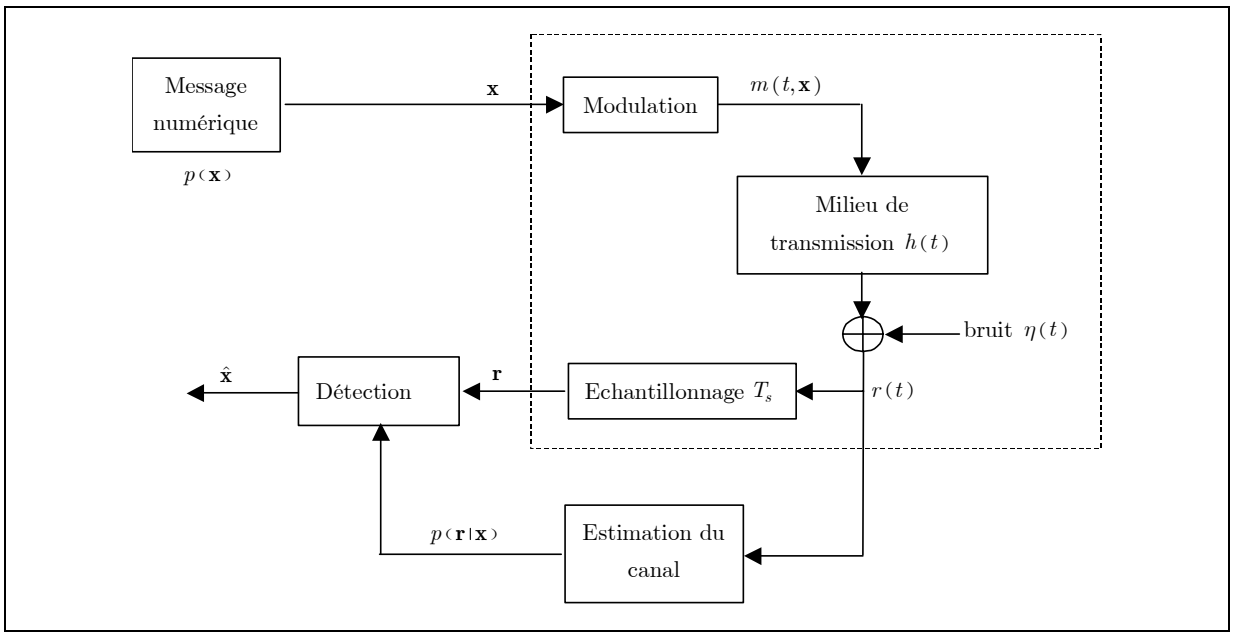

Figure 1.2 : Schéma équivalent en bande de base d'une transmission numérique

On montre que le critère du Maximum a Posteriori (1.3) équivaut à *minimiser la probabilité d'erreur* en réception, de plus la théorie de l'information précise dans quelles conditions cette probabilité d'erreur peut être rendue arbitrairement petite.

Considérons une séquence émise $\mathbf{x}_N$ de N éléments binaires. On dispose au niveau du récepteur d'une connaissance *a priori* $p(\mathbf{x}_N)$ sur les valeurs possibles de cette séquence. Plus précisément, *l'incertitude* au niveau du récepteur sur la séquence $\mathbf{x}_N$ est quantifiée par l'entropie [Moreau, 1995] :

$$H(\mathbf{x}_N) = -\sum_{\mathbf{x}_N} p(\mathbf{x}_N)\log_2\left(p(\mathbf{x}_N)\right) \tag{1.4}$$

où la sommation a lieu sur toutes les séquences possibles $\mathbf{x}_N$.

[3] On notera que dans la pratique, le canal radiomobile n'est pas stationnaire et les paramètres $\{h(nT);N_0\}$ doivent être régulièrement ré-estimés en émettant des séquences d'apprentissage à intervalles réguliers.

La détection au sens du Maximum a Posteriori (1.3) peut être réalisée sans erreurs si l'*information* $I(\mathbf{x}_N, \mathbf{r}_N)$ apportée[4] sur $\mathbf{x}_N$ par la séquence $\mathbf{r}_N$ reçue est égale à *l'incertitude* $H(\mathbf{x}_N)$ que l'on a au récepteur sur la séquence émise $\mathbf{x}_N$:

$$\textit{incertitude au récepteur } H(\mathbf{x}_N) = I(\mathbf{x}_N, \mathbf{r}_N) \textit{ connaissance apportée par } \mathbf{r}_N \tag{1.5}$$

La limite asymptotique de ce résultat (i.e. lorsqu'on considère des séquences de longueur $N \to \infty$) est le théorème de Shannon qui stipule qu'on peut toujours rendre la probabilité d'erreur arbitrairement petite dès lors que le *débit entropique* $\overline{H}(\mathbf{x})$ du message est inférieur à la *capacité* C du canal :

$$\overline{H}(\mathbf{x}) = \lim_{N\to\infty} \frac{1}{N} H(\mathbf{x}_N) = \left(\lim_{N\to\infty} \frac{1}{N} I(\mathbf{x}_N, \mathbf{r}_N) \right) \leq C \tag{1.6}$$

Le débit entropique mesure le débit *moyen* d'information du message à transmettre (en bits par symbole) et la capacité C du canal correspond au maximum d'information que l'on peut transmettre par symbole émis dans le canal[5].

Ces développements montrent l'intérêt *d'exploiter la connaissance a priori sur la source* pour diminuer le taux d'erreurs, ceci repose sur un décodage par séquences au sens du MAP. On remarque également qu'ils supposent un décodeur de *mémoire infinie*.

Lorsque le décodage n'exploite pas cette redondance (décodage au sens du Maximum de Vraisemblance $p(\mathbf{r}|\mathbf{x})$), le message est implicitement supposé i.i.d. (indépendant et identiquement distribué) et on ne peut avoir une transmission sans erreurs que si le débit binaire D vérifie :

$$D \leq C' \text{ (bits/s)} \tag{1.7}$$

Il est également intéressant de préciser les paramètres physiques qui déterminent la capacité C' d'un canal (exprimée en bits/s). Ces paramètres sont le rapport signal à bruit *SNR* et la largeur de bande W du canal. En effet, l'information apportée sur $\mathbf{x}_N$ par la séquence reçue $\mathbf{r}_N$ tend à augmenter avec le *SNR* et avec elle la capacité C en bits par symbole. La largeur de bande W quant à elle, limite le débit binaire en entrée du canal et ainsi la capacité en bits par seconde $(C' = 2WC)$.

On verra en Annexe C que le canal radiomobile présente une forte *sélectivité temporelle*, autrement dit que le rapport signal à bruit *SNR* peut devenir localement très faible. Dans ces conditions, on observera des *paquets d'erreurs* (« bursts ») en sortie du détecteur de la Figure 1.2, même si celui-ci exploite la redondance du message car il travaille en pratique sur une longueur N limitée.

---

[4] L'information mutuelle $I(\mathbf{x}_N, \mathbf{r}_N)$ se quantifie comme étant la différence entre l'entropie $H(\mathbf{x}_N)$ de $\mathbf{x}_N$ et son entropie *connaissant* $\mathbf{r}_N$: $H(\mathbf{x}_N|\mathbf{r}_N) = \sum_{\mathbf{x}_N} \sum_{\mathbf{r}_N} p(\mathbf{x}_N, \mathbf{r}_N) \log_2 \{ p(\mathbf{x}_N|\mathbf{r}_N) \}$.

[5] Ici on rappellera qu'en pratique lorsque le canal est de largeur mono-latérale W, on ne peut transmettre au plus que $2W$ symboles par seconde. On exprime alors la capacité en bits par seconde, selon $C' = 2WC$.

Pour lutter contre ces évanouissements temporels, on utilise des techniques de *diversité*. Dans le cas du système GSM, on utilise ainsi un *entrelacement temporel* des symboles du message $\mathbf{x}$ avant modulation (cf. Annexe C). Cependant l'exploitation de la redondance du message $\mathbf{x}$ (« décodage ») ne peut plus se faire au même niveau que l'égalisation et la démodulation. Ceci conduit à la division du récepteur en deux entités séparées par l'entrelacement des données :

- Un récepteur *interne* qui effectue l'estimation des paramètres du canal, la démodulation et l'égalisation.
- Un récepteur *externe* qui considère la sortie du récepteur interne comme celle d'un canal *sans mémoire*, *de probabilités de transitions connues* (estimées par le décodeur interne) et qui prend une décision sur les symboles de la source. Ce récepteur est abordé dans la section suivante.

Le récepteur interne a donc pour but de présenter un canal « idéal » au récepteur externe, selon le schéma de la Figure 1.3. Il est détaillé en Annexe C dans le cas du GSM.

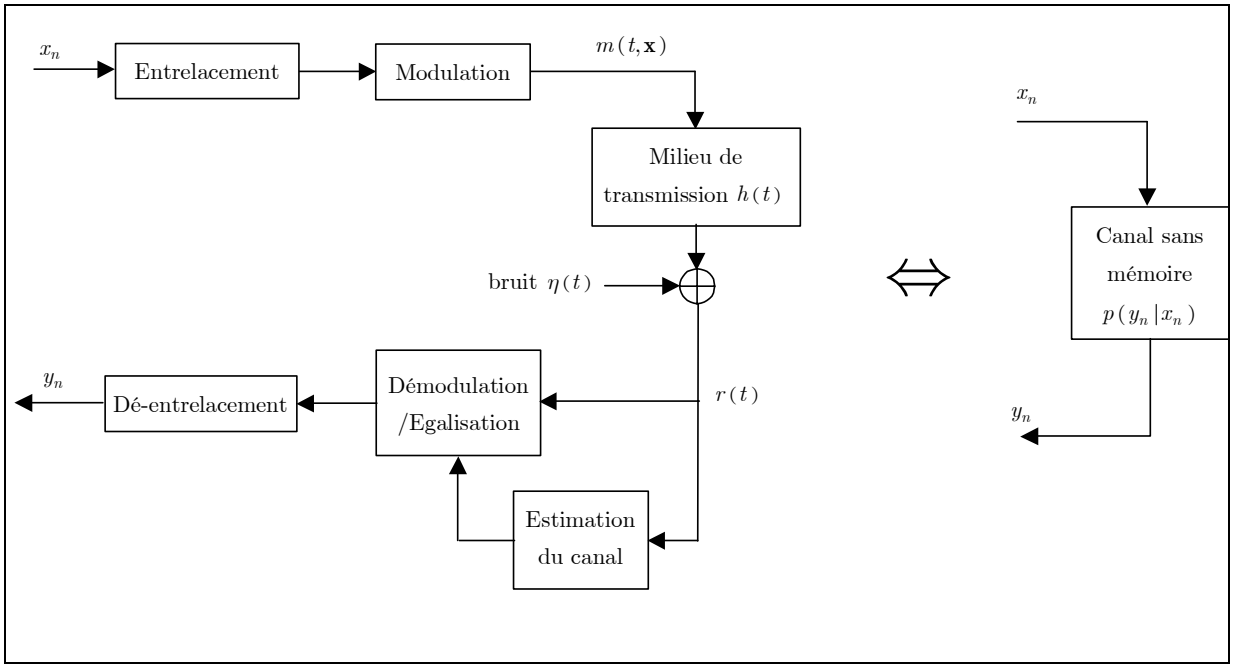

Figure 1.3 : Récepteur interne correspondant au modèle de canal idéal

Le récepteur interne considère que le message émis dans le canal est i.i.d. Autrement dit la démodulation et l'égalisation (conjointes dans le cas du GSM) utilisent le critère du Maximum de Vraisemblance MV.

1.2.3 Principe de séparation entre codage de source et codage de canal

On reprend (Figure 1.4) le schéma d'une réception au sens du Maximum a Posteriori où l'on a remplacé le récepteur interne par le canal idéal équivalent. On a considéré jusqu'ici que la redondance exploitée par le décodeur MAP était celle de la source elle-même ce qui est une simplification. En effet, la condition sur le débit entropique $\overline{H}(\mathbf{x})$ pour une *transmission sans erreurs* (théorème de Shannon) se double d'une condition sur le débit binaire D_x en entrée d'un *canal à bande limitée* W. Soit :

$$\overline{H}(\mathbf{x}) \leq C \ \text{(bits/symbole)} \tag{1.8}$$

et :

$$D_x \leq 2W \ \text{(symbole/s)} \tag{1.9}$$

La seconde condition est rarement vérifiée, notamment pour les canaux radio-mobiles où l'on souhaite économiser la bande passante. Ceci impose alors une *réduction du débit binaire*. Cette réduction de débit *binaire* n'induit pas de distorsions tant que le débit *entropique* reste inchangé (exprimé en bits/s).

Cependant, la première condition est également rarement vérifiée ce qui conduit à accepter une *distorsion* de manière à diminuer le débit *entropique* (en bits/s). Le *principe de séparation* de Shannon affirme alors que cette réduction de débit avec pertes (distorsions) peut s'effectuer en *deux étapes séparées* sans perte d'optimalité.

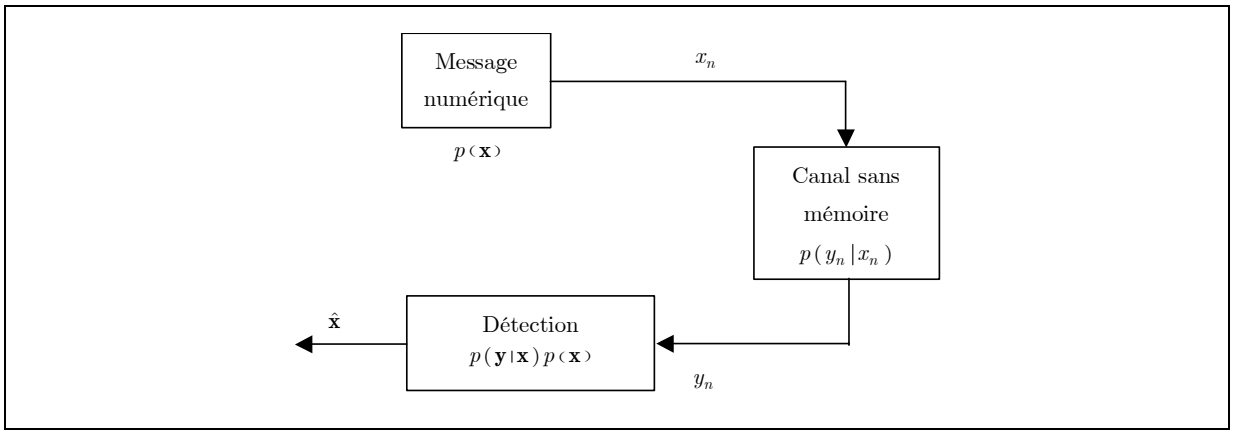

Figure 1.4 : Réception au sens du Maximum a Posteriori

Plus précisément, considérons une source $\mathbf{s}$ de débit *binaire* D_s et de débit entropique par symbole $\overline{H}(\mathbf{s})$, deux cas sont possibles :

- $D_s \bar{H}(\mathbf{s}) \leq 2WC$, alors la source peut être transmise *sans erreurs*, éventuellement après une compression (codage de source) *sans pertes* réduisant le débit binaire si $D_x > 2W$.

- $D_s \bar{H}(\mathbf{s}) > 2WC$, alors on peut effectuer une compression *avec pertes* par un codeur de source dont la sortie u_n vérifie $\bar{H}(\mathbf{u}) = 1$ et $D_u \leq 2WC$ puis transmettre *sans erreurs* les données après un codage canal dont la sortie x_n vérifie $D_x \leq 2W$ avec un débit entropique inchangé (en bits par seconde).

Selon ce **principe de séparation**, le codeur source et le codeur canal ont des rôles distincts mais duaux :

- Le codeur source adapte le débit *entropique* de la source à la capacité du canal (exprimés par unités de temps). Il réalise pour cela un *compromis* entre réduction du *débit entropique* et *distorsion*[6]. De plus, en renvoyant une sortie i.i.d. il permet d'atteindre la limite de capacité du canal.

- Le codeur canal met en forme le message à transmettre dans le canal bruité en rajoutant des symboles de *redondance* (sans augmenter le débit *entropique* par unité de temps). Il permet ainsi le décodage MAP en réception. On appelle *rendement du codeur canal* la quantité $R = D_u / D_x$ et l'on doit avoir $R < C$ pour une transmission sans erreurs. Le codeur canal réalise donc un *compromis* entre *débit binaire* en entrée du canal et *taux d'erreurs binaires* en réception.

Il faut bien garder à l'esprit que l'optimalité de ce principe de séparation n'est garantie que pour un codage/décodage de mémoire infinie (séquences de longueur $N \to \infty$). Néanmoins, ce principe a conduit à optimiser le codeur source et le codeur de canal indépendamment l'un de l'autre. Il est à la base des systèmes de communications numériques actuels et est schématisé Figure 1.5.

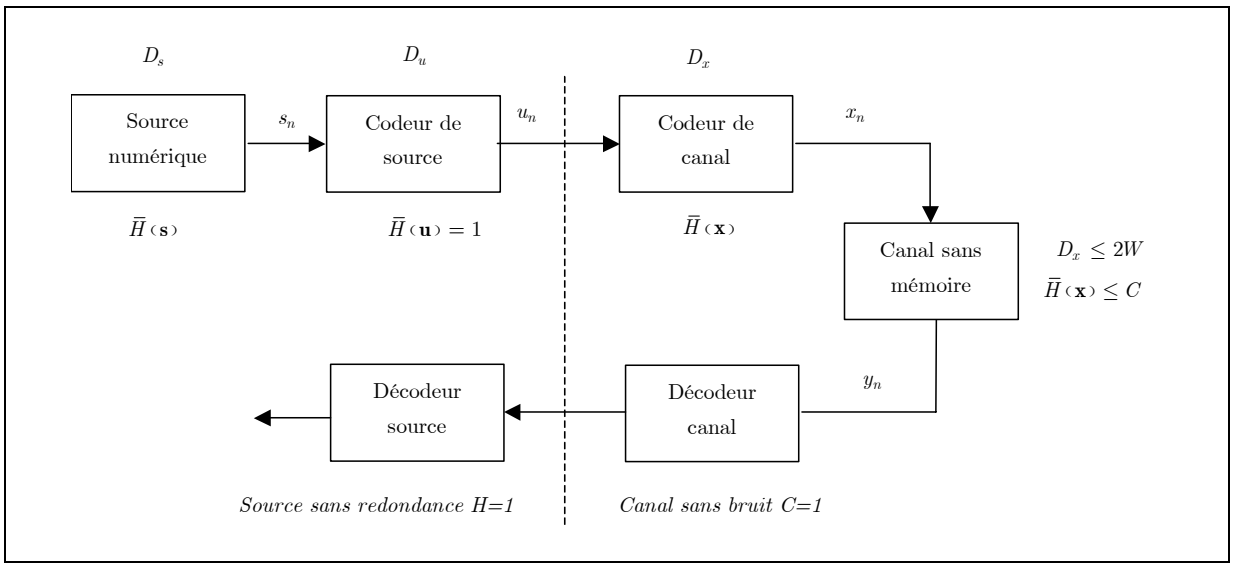

Figure 1.5 : Principe de séparation entre décodeur source et canal

[6] Selon une mesure qui dépend du signal et de l'application. Dans le cas de la parole, le codeur utilise l'erreur quadratique en sortie d'un filtre qui modélise certaines propriétés de l'audition (masquage, cf. Annexe A).

1.3 Mise en oeuvre pratique – le système GSM

Selon le schéma idéal décrit aux paragraphes précédents, la seule *dégradation du signal* associée à la transmission serait la *distorsion* introduite au niveau du codeur de source et donc parfaitement contrôlée. Cependant, comme on l'a déjà noté, ce schéma optimal n'est atteint que pour des codeurs / décodeurs de mémoire infinie, et on peut donc s'attendre à ce qu'il subsiste des *erreurs résiduelles* en sortie du décodeur canal[7]. D'autre part, même à supposer des codeurs d'une grande complexité, le réglage optimal de ces codeurs est effectué pour une capacité du canal *C nominale*. Si la capacité réelle du canal devient inférieure à cette capacité nominale, le réglage n'est plus optimal et les performances du décodeur canal se dégradent très rapidement [Hedelin et al., 1995].

1.3.1 Protection hiérarchique et masquage

Ceci conduit, pour un codeur canal de rendement R fixé, à répartir de manière inégale la redondance introduite afin de la réserver aux éléments binaires les plus importants pour la minimisation de la *distorsion* [Duhamel et al., 1997]. Cette technique connue sous le nom d'*Unequal Error Protection* (UEP) est notamment utilisée dans le GSM. Elle permet d'être moins sensible (en termes de *distorsion*) aux variations de la capacité du canal.

Dans le cas du GSM, cette *protection inégale* des éléments binaires en sortie du codeur parole est complétée par un mécanisme de *masquage*. On ajoute une *détection d'erreur* sur les bits dont l'impact sur la *distorsion* du signal reconstruit est tel qu'il faut absolument éviter de les décoder en présence d'erreurs. Le mécanisme de masquage consiste alors en la substitution de la trame « rejetée » à partir de la dernière trame valide reçue[8]. Une telle procédure exploite implicitement la *redondance résiduelle* présente en sortie du codeur parole. Le principe consistant à utiliser la redondance résiduelle du codeur parole pour réduire l'impact des erreurs résiduelles en sortie du décodeur canal est à la base des algorithmes proposés dans ce document, nous y reviendrons au paragraphe 1.4.

Pour préciser tout ceci, considérons le schéma illustré Figure 1.6 et qui résume la chaîne de transmission de la parole mise en œuvre pour le GSM EFR. Le principe et le fonctionnement de chacun des éléments de cette chaîne sont détaillés en Annexes, nous en présentons juste ici les points clés.

Le codeur de parole de type CELP (cf. Annexe A) analyse la parole par trames de 20 ms. Les paramètres calculés [GSM, 06.60] pour chaque trame correspondent à :

---

[7] De manière duale, il subsiste une *redondance résiduelle* en sortie du codeur parole. Celle-ci sera exploitée par la suite.

[8] Le codeur GSM fonctionne par trame et c'est l'intégralité d'une trame qui est substituée par la procédure de masquage.

- 2 jeux de Lignes Spectrales de Fréquences LSF représentant *l'enveloppe spectrale* du signal analysé [Kleijn et al., 1995] et qui définissent la fonction de transfert du filtre de synthèse.
- Le gain et délai (lag) du dictionnaire adaptatif, actualisés sur des sous-trames de 5 ms, et qui définissent la *partie périodique de l'excitation* en entrée du filtre de synthèse.
- Le gain et l'index du code algébrique définissant la *partie stochastique de l'excitation*. Ils sont également actualisés sur des sous-trames de 5 ms.

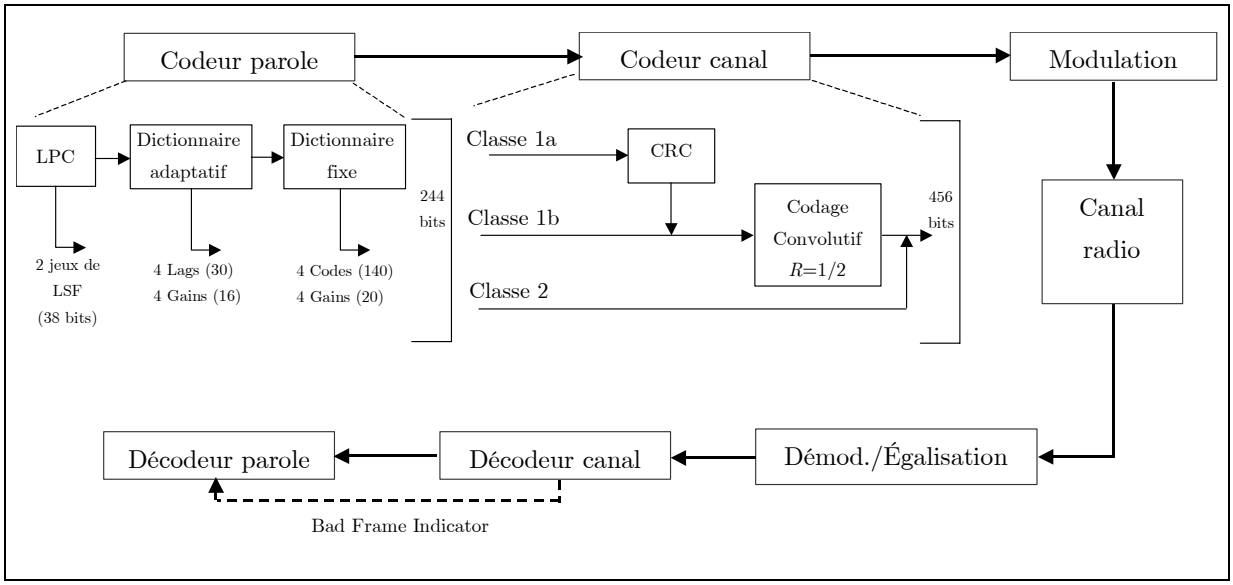

Figure 1.6 : Schéma synoptique de la transmission de la parole par le GSM EFR

Après quantification (cf. Annexe A), ces paramètres sont représentés[9] par une trame de 244 bits en sortie du codeur de parole. Conformément à la stratégie de *protection hiérarchique* UEP présentée plus haut, ces bits sont répartis en trois *classes* selon leur *impact sur la qualité* de la parole [GSM, 05.03] :

- Classe 1a : 65 bits, très sensibles aux erreurs, ils ne doivent pas être mal interprétés.
- Classe 1b : 109 bits, sensibles aux erreurs.
- Classe 2 : 70 bits, les moins sensibles aux erreurs.

Comme l'illustre la Figure 1.6, un codage canal convolutif de rendement ½ (cf. Annexe B) est appliqué aux bits de la Classe 1 (a et b) mais les bits de la Classe 1a sont préalablement protégés par des codes cycliques CRC. Les bits de la Classe 2 sont eux transmis sans la moindre protection.

En réception, lorsqu'une erreur est détectée à l'aide des codes CRC sur les bits de la Classe 1a, un *indicateur de trame invalide* BFI (*Bad Frame Indicator*) est positionné et transmis au décodeur parole

[9] La table d'allocation des bits est donnée en Annexe A.

qui déclenche la procédure de *substitution de trame* [GSM, 06.61]. Celle-ci consiste à *extrapoler* les paramètres de la trame invalide à partir de ceux de la dernière trame valide reçue. Cette extrapolation assure notamment deux effets importants pour la qualité perçue :

- **l'atténuation (muting)**

L'idée à la base de l'atténuation est qu'il convient de ne pas prolonger indéfiniment le signal en cas d'une succession de trames perdues, mais d'amener à une transition douce avec le silence. Dans le cas de l'EFR, le gain de chacun des dictionnaires est ainsi remplacé par la valeur médiane des gains des sous-trames précédentes et un facteur d'atténuation variable est appliqué à cette valeur médiane.

- **l'expansion spectrale**

L'expansion spectrale est complémentaire de l'atténuation, elle permet de tendre progressivement vers un spectre plat en cas d'une succession de trames perdues. Ainsi, le codeur EFR ré-utilise les valeurs passées des LSF en les faisant tendre vers leur valeur moyenne.

Enfin, le délai de pitch de la dernière sous-trame valide est répété pour toutes les sous-trames de la trame substituée. On notera que les index des dictionnaires fixes ne sont pas substitués mais que les valeurs reçues sont utilisées telles quelles.

1.3.2 Analyse des dégradations de la parole décodée

Nous dressons ici un bilan des dégradations de la qualité vocale les plus fréquemment rencontrées dans le contexte de la transmission radiomobile GSM. Bien que la thématique qui sera finalement développée par nos travaux soit centrée sur les erreurs de transmission, nous élargissons la présentation qui est faite ici à l'ensemble des dégradations introduites par le réseau de transmission radiomobile. Ceci permettra de les hiérarchiser les unes par rapport aux autres.

Plus précisément, on peut diviser les dégradations de la parole transmise par le réseau GSM en trois catégories, présentées dans les paragraphes qui suivent.

1.3.2.1 Dégradations liées au codage parole

L'objectif de réduction de débit impose, comme on l'a vu, au codeur parole d'effectuer un codage *avec pertes*. On peut ainsi assimiler le codeur CELP utilisé par le GSM EFR à un *quantificateur* multi-étages opérant dans le domaine du résidu LPC de la parole (Figure A.20).

Cependant, la qualité de parole obtenue avec le codeur EFR, en l'absence d'erreurs de transmission, est jugée satisfaisante [Pascal et al., 1999], et correspond à une note de qualité perçue de 4 MOS[10]. D'autre part, [Cox et al., 1989] dans son étude du CELP montre que ce type de codeur, en tant que

[10] La définition de l'échelle de qualité perçue MOS est précisée au paragraphe 1.5.

codeur forme d'onde, est assez robuste aux entrées bruitées et est surtout sensible aux erreurs de transmission.

1.3.2.2 Dégradations liées aux procédures du réseau

Certaines dégradations sont générées par les procédures du réseau radio-mobile lui-même. Il en est ainsi notamment lorsque le mobile change de cellule [Cruchant et al., 1998]. Il y a alors une période (*handover*) variant entre 40 à 160 ms durant laquelle la parole doit être extrapolée, ce qui dégrade fortement la qualité. Cette dégradation est à rapprocher de celle associée à la *substitution de trame perdue* et qui sera abordée au paragraphe suivant.

On peut également inclure dans cette catégorie des défauts liés au mode de transmission discontinue DTX [Scalart, 1997]. Ce mode optionnel repose sur une détection d'activité vocale DAV pour n'émettre que les segments correspondants à de la parole active. Durant les périodes d'inactivité vocale, le codeur se contente de transmettre (à très bas débit) une information destinée à coder le bruit de fond. Cependant, les erreurs de DAV peuvent entraîner des troncatures de la parole, et le rafraîchissement irrégulier du bruit de confort peut générer une gêne due à une sensation de trop forts contrastes de bruit [Veaux, 1998].

Néanmoins, ces dégradations sont nettement moins fréquentes que celles liées aux erreurs de transmission ou peuvent parfois s'y rattacher, comme dans le cas du *handover*.

1.3.2.3 Dégradations associées aux erreurs de transmission

On considérera ici l'ensemble des dégradations associées à la présence d'erreurs binaires résiduelles en sortie du décodeur canal. Elles correspondent soit à une utilisation directe des bits erronés par le décodeur parole, soit à des artefacts générés par la procédure de masquage. Plus précisément, on peut dégager trois principaux types de dégradations :

1.3.2.3.a Erreurs non détectées par le CRC

Les distorsions les plus gênantes sont celles générées par le décodage de bits erronés au sein de la Classe 1a regroupant les bits les plus *sensibles*. Une telle situation peut se présenter lorsque le code CRC utilisé par l'indicateur de trame invalide BFI fait une erreur de détection. En effet, le pouvoir de détection d'un tel code est limité et dépend du nombre de bits de redondance ajoutés (cf. Annexe B). Cette situation était fréquente pour la première génération de codeur GSM FR qui utilisait un CRC sur 3 bits. Elle résultait en des distorsions non-linéaires de la parole, extrêmement audibles (saturations brusques, sons très artificiels). La Figure 1.7 donne un exemple d'une telle dégradation, rencontrée dans le cas du GSM FR. L'indicateur BFI y est superposé au signal de parole.

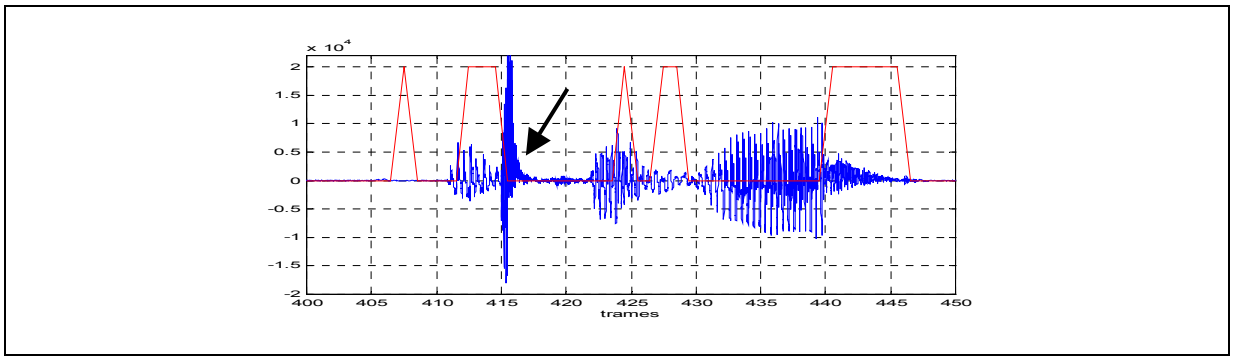

Figure 1.7 : Erreur non détectée parmi la Classe 1a pour le GSM FR (indicateur BFI en rouge)

La détection d'erreurs a cependant été largement améliorée avec le GSM EFR en rajoutant un CRC sur 8 bits [GSM, 05.03]. On peut dès lors considérer que ce type de dégradation n'est quasiment plus jamais rencontré pour le GSM EFR.

1.3.2.3.b Erreurs résiduelles sur les bits non protégés par CRC

Les bits qui ne subissent aucune protection aux erreurs (Classe 2) correspondent uniquement au dictionnaire d'excitation algébrique du codeur CELP. L'impact d'une erreur sur ces paramètres peut apparaître, au niveau du signal de parole décodé, comme équivalent à du bruit de quantification [Sereno, 1991].

Les bits de la Classe 1b sont protégés par le codeur convolutif mais sont toujours décodés (sauf erreur dans la Classe 1a). Ils correspondent pour l'essentiel aux bits de poids faibles codant les gains d'excitation du CELP (dictionnaires adaptatif et algébrique) ainsi qu'aux bits représentant les deux dernières LSF « de hautes fréquences[11] ». La présence d'erreurs résiduelles parmi ces bits engendre des dégradations à une *échelle fine* de la parole. Ainsi, on citera notamment parmi les défauts audibles :

- des défauts de voisement (perte de voisement ou harmoniques de la parole noyées dans du « bruit»).

- des variations de niveau entre les segments de parole reçus sans erreurs et ceux décodés en présence d'erreurs binaires résiduelles et qui apparaissent plus « étouffés »[12].

La Figure 1.8 illustre certaines de ces dégradations. Elle compare le spectre du signal de parole décodé par le GSM EFR en l'absence d'erreurs de transmission avec celui de la parole décodée pour une transmission bruitée (rapport *Porteuse sur Interférences* : $C/I = 2$dB). Parmi d'autres défauts, liés à la procédure de masquage (cf. 1.3.2.3.c), on observe une dégradation de la structure harmonique qui tend à être noyée dans le bruit d'excitation.

---

[11] Celles-ci sont les LSF9 et LSF10 qui définissent la forme de l'enveloppe spectrale LPC dans les hautes fréquences (approximativement dans la gamme [2800-3600] Hz).

[12] Ces variations de niveaux ne sont pas dus à l'atténuation appliquée par la procédure de substitution de trame, celle-ci n'étant pas activée.

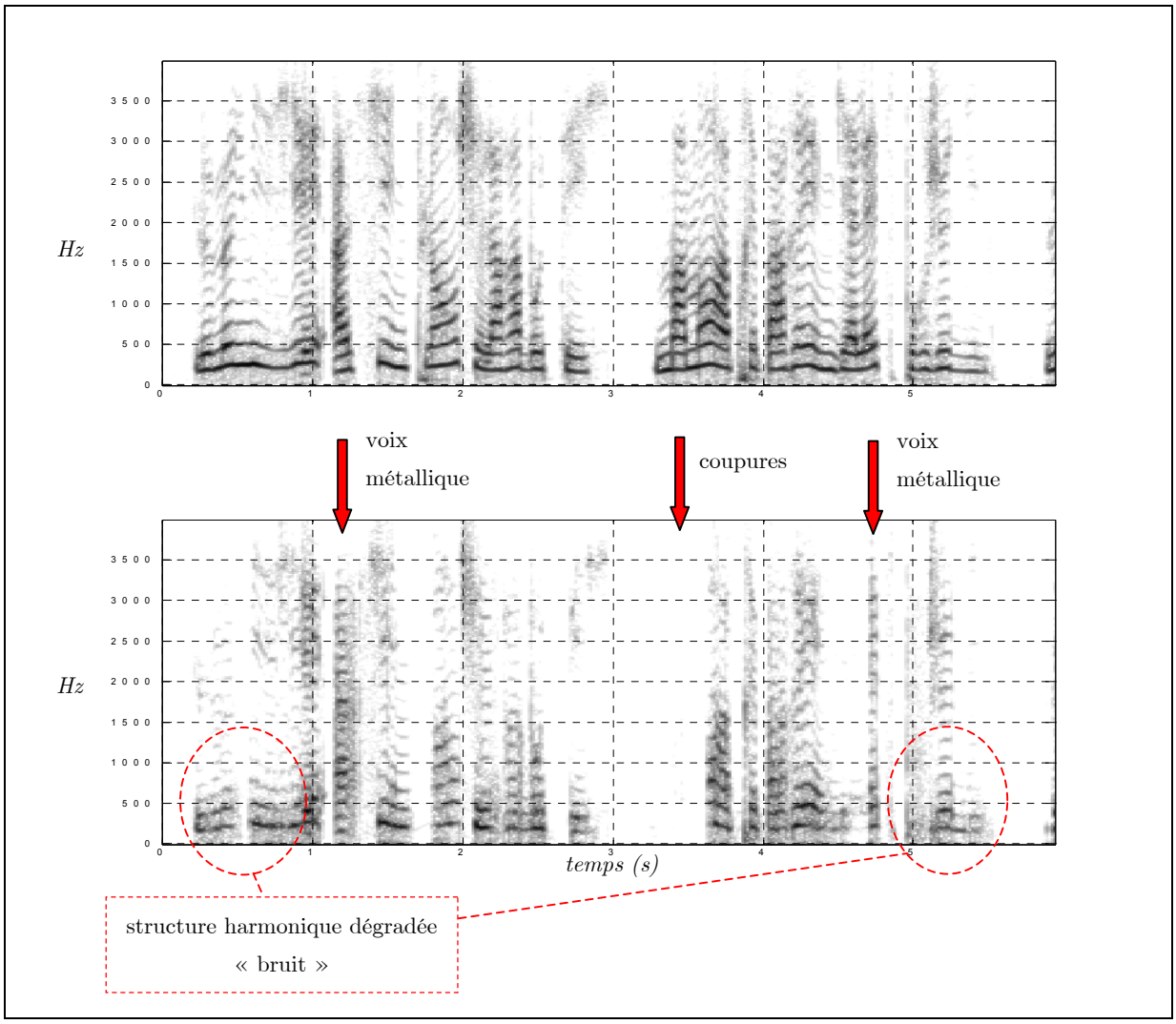

Figure 1.8 : Spectrogrammes de la parole décodée sans erreur et de la parole reçue à C/I=2dB (TU50)

1.3.2.3.c artefacts introduits par la procédure de masquage

Les dégradations associées à la technique de masquage utilisée dans un système tel que le GSM EFR sont de deux ordres. Il y a d'une part, les coupures de parole qui sont intrinsèques à la stratégie même de ne pas décoder les trames binaires reçues lorsque les bits sensibles sont erronés. Cette dégradation ne peut être réduite qu'en envisageant une technique de décodage qui puisse corriger au moins en partie ces erreurs pour les bits les plus sensibles.

Il y a d'autre part, les artefacts générés par la procédure de *substitution de trame*. Ces artefacts sont bien évidemment spécifiques à une procédure donnée, et même à une implémentation donnée du codeur EFR puisqu'elle n'est pas normalisée. Nous avons présenté succinctement celle donnée en exemple dans la norme du GSM EFR [GSM, 06.61]. Elle introduit un effet de « voix métallique » assez audible lorsque la trame extrapolée est une trame voisée. Cet effet est du à la répétition à l'identique du délai de pitch (cf. paragraphe 1.3.1) qui crée une périodicité à long-terme artificiellement

stable de la parole générée. L'introduction d'une *gigue de pitch*[13] permet d'éliminer cet artefact. De nombreux autres mécanismes peuvent être raffinés pour améliorer la qualité perçue de la parole synthétisée par ces procédures de substitution [De Martin et al., 2000]. L'algorithme proposé pour le GSM EFR marquait déjà à cet égard un net progrès par rapport à la technique assez rudimentaire utilisée par le GSM FR et qui produisait un effet « voix de robot » très marqué (cf. Chapitre 2).

Les défauts de coupure et de « voix métallique » apparaissent également sur la Figure 1.8. La « voix métallique » notamment, se manifeste pas des harmoniques invariantes au cours du temps et présentes jusque dans les hautes fréquences.

1.3.3 Discussion

La présentation qui a été faite des dégradations observées sur la parole transmise par le GSM ne prétend pas être exhaustive, néanmoins elle permet de relier la nature et *l'impact* des dégradations à leur *origine*. C'est ainsi que nous avons choisi de centrer nos travaux sur les défauts liés aux *erreurs de transmission* et plus précisément aux erreurs résiduelles en sortie du décodeur canal. Ces défauts nous apparaissent en effet être ceux qui impactent le plus la qualité de la parole dans l'utilisation du GSM au quotidien.

Les techniques de masquage mises en œuvre pour maintenir une qualité acceptable de la parole démontrent qu'il existe des paramètres du codeur parole pour lesquels un modèle d'extrapolation entraîne une *distorsion*[14] plus faible que le décodage de l'information reçue du canal en présence d'erreurs résiduelles. Ces techniques reviennent donc implicitement à s'appuyer sur la *redondance résiduelle* des paramètres du codeur parole pour minimiser la *distorsion* en présence *d'erreurs résiduelles*. Mais cette démarche est de type « tout ou rien », c'est-à-dire que le décodeur parole fait soit totalement confiance à l'information issue du décodeur canal, soit utilise exclusivement le modèle d'extrapolation des paramètres. Il en résulte le phénomène de « coupures » de la parole restituée. L'exploitation conjointe de ces deux types d'informations offrirait un meilleur compromis. Elle constitue un des axes de recherche pour l'amélioration de la qualité vocale, présentés dans la partie suivante de ce chapitre.

[13] Petites variations du pitch autour de sa valeur moyenne (qui est la valeur de la dernière sous-trame valide).

[14] On ne parle pas ici nécessairement d'une distorsion basée sur un critère quadratique, mais d'une distorsion basée sur la perception, qui est le critère à prendre en compte in fine.

1.4 Amélioration de la qualité vocale en réception

Nous présentons ici les différents axes de recherche que nous avons suivis dans le but d'améliorer la *qualité perçue* de la parole restituée par le GSM. Les dégradations de la parole envisagées sont toutes liées aux erreurs de transmission.

Malgré la diversité de leur point de vue, deux aspects fédèrent les méthodes présentées. Le premier aspect est d'ordre pratique, ces méthodes s'appliquent toutes en réception, *sans nécessiter de modifications des codeurs* parole et canal du GSM. Le deuxième aspect correspond à l'idée d'exploiter un *modèle a priori* de la source (parole) pour réduire l'impact qualitatif des dégradations entraînées par les erreurs de transmission.

1.4.1 Post-traitement du signal de parole en sortie du décodeur

Notre première approche était d'effectuer un post-traitement du signal de parole en sortie de la chaîne de réception du GSM. Un exemple de post-traitement appliqué au contexte du GSM est le post-filtrage proposé par [Sereno, 1991]. Celui-ci était destiné à masquer l'effet des erreurs introduites sur les bits codant l'excitation (Classe 2) et qui peut être assimilé à du bruit de quantification. Cependant, ce post-filtre était activé conditionnellement à une mesure d'erreur estimée par le décodeur canal.

Ceci illustre bien le principal problème du post-traitement appliqué aux dégradations introduites par les erreurs de transmission. En effet, ces dégradations sont non-stationnaires et non-linéaires, il est donc très difficile de les estimer à partir du signal reçu. En particulier, les estimateurs linéaires classiquement utilisés pour le rehaussement de la parole [Beaugeant, 1999] ne peuvent être appliqués.

Dans ce contexte, on ne peut effectuer de post-traitement du signal de parole que si l'on dispose d'un modèle *a priori* du défaut à traiter ou d'un modèle *a priori* du signal de parole à reconstruire. De plus, les « traitements » appliqués au signal ne peuvent être, dans le cas général, de simples opérations de filtrage linéaire. Les techniques à envisager relèvent du *masquage* ou du schéma plus général[15] d'analyse - modification - synthèse [Laroche, 1995].

Nous abordons un premier aspect du post-traitement au Chapitre 2, en étudiant la détection d'artefacts sur le signal en sortie de décodeur parole. Cependant, il est progressivement apparu que l'intérêt des post-traitements pour les dégradations associées aux erreurs de transmission se limitaient à celles dues aux erreurs binaires dans les Classes 1b et 2 (cf. paragraphe 1.3.2.3). Ces dégradations ne nous paraissent pas être les plus déterminantes pour la qualité de la parole GSM transmise en présence

[15] Les techniques fréquentielles d'analyse-modification–synthèse du signal pourraient être utilisées pour le masquage ou pour toute procédure visant à reconstruire le signal d'après un modèle *a priori*.

d'erreurs. La contribution principale à la distorsion du signal de parole décodé étant due aux bits de la Classe 1a (ou à la procédure de masquage enclenchée par le décodeur).

Ceci nous a conduit à envisager d'autres approches, situées au niveau des décodeurs parole et canal du GSM EFR.

1.4.2 Décodage parole à entrées souples

Si l'on vise une amélioration significative de la qualité de la parole restituée par le GSM en présence d'erreurs de transmission, il convient de minimiser la distorsion entraînée par le décodage des erreurs binaires résiduelles. Une idée, déjà esquissée par la technique du masquage, est d'utiliser pour ce faire la *redondance résiduelle* laissée par le codeur parole. Cette approche se rattache aux développements récents [Duhamel et al., 1997] sur le codage et décodage *conjoint source-canal*. Elle part du constat que le schéma idéal (Figure 1.5) justifiant la séparation du codage/décodage source et canal n'est jamais atteint dans la pratique et qu'il vaut mieux dès lors faire interagir ces 2 étapes plutôt que de les idéaliser.

Le principe du décodage parole à entrées souples est illustré Figure 1.9. La sortie binaire $\hat{\mathbf{b}}$ du décodeur canal est ici complétée par une information de fiabilité (ou probabilités d'erreur $\mathbf{p}_e$ associées aux bits $\hat{\mathbf{b}}$). Le couple $\left(\hat{\mathbf{b}}, \mathbf{p_e}\right)$ forme la « sortie souple » du décodeur canal (resp. « entrée souple » du décodeur parole). Cette information souple issue du canal est utilisée conjointement au niveau du décodeur de parole avec une information *a priori* sur les paramètres du codeur de parole. Ceci permet *l'estimation optimale* du paramètre transmis [Hedelin et al., 1995], c'est-à-dire la valeur minimisant le critère utilisé pour mesurer la *distorsion*.

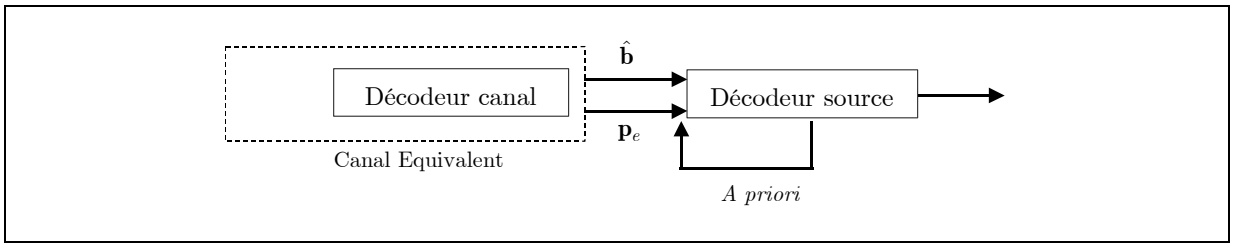

Figure 1.9 : Principe du décodage source à entrées souples

Une telle approche sera développée dans les chapitres 4 et 5 pour le GSM EFR. Notre principale contribution sera d'élaborer un modèle *a priori* des paramètres permettant de réduire la complexité de l'estimation et offrant une caractérisation plus fine de la redondance résiduelle.

1.4.3 Décodage Canal Contrôlé par la Source (SCCD)

Dans le schéma de la Figure 1.9, la sortie du décodeur canal est vue comme celle d'un canal équivalent donné et la redondance résiduelle laissée par le codeur parole sert à compenser les erreurs résiduelles observées en sortie de ce canal. En prolongeant l'idée d'exploiter la redondance résiduelle du codeur parole pour compenser les imperfections du décodeur canal, il paraît intéressant d'utiliser cette redondance résiduelle de « source » conjointement avec la redondance « systématique » introduite par le codeur canal. C'est l'idée à la base du décodage canal contrôlé par la source [Hagenauer, 1995] dont le schéma de principe est illustré Figure 1.10. Le canal équivalent correspond ici au récepteur interne de la Figure 1.3 et les probabilités d'erreur $\mathbf{p_e}$ associées aux sorties binaires $\mathbf{y}$ ne sont autres que les probabilités de transition de ce canal, estimées par le récepteur interne (cf. Annexe C).

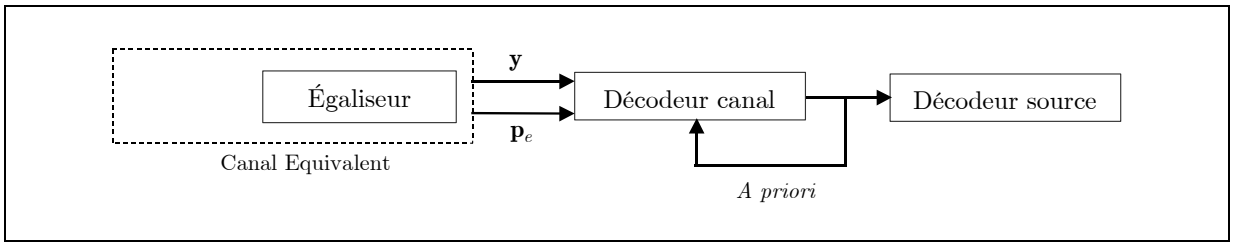

Figure 1.10 : Principe du décodage canal contrôlé par la source

L'information *a priori* issue de la redondance résiduelle est alors exploitée au niveau du décodeur canal. L'objectif étant ici de *minimiser le taux d'erreurs résiduelles* en sortie du décodeur canal plutôt que de minimiser un critère de distorsion sur les paramètres « corrompus » reçus par le décodeur parole. Nous avons jugé intéressant d'étudier l'application de ce principe au GSM EFR, dans le but d'identifier lequel de ces deux points de vue sur l'utilisation de la redondance résiduelle était le plus judicieux et si ils pouvaient éventuellement être complémentaires. Nous nous attacherons à la fois à améliorer l'exploitation de la redondance résiduelle par les algorithmes de SCCD et à les adapter aux contraintes pratiques du codage canal mis en œuvre pour les communications radio-mobiles. Ces développements sont présentés au Chapitre 7.

Pour conclure, on notera qu'on s'est restreint ici aux méthodes situées en réception puisque l'une de nos contraintes était de ne pas modifier la partie codage du système GSM. Cependant de nombreuses voies de recherches s'offrent à ce niveau pour rendre les paramètres transmis par le codeur parole plus robustes aux erreurs de transmission [Duhamel et al., 1997]. On citera notamment les techniques d'optimisation de l'étiquetage des centroïdes du dictionnaire de quantification (*Index Assignement*). L'objectif visé par ces techniques étant de minimiser la distorsion induite dans l'espace des centroïdes[16] par une erreur de transmission sur l'index du centroïde.

[16] C'est-à-dire l'espace des paramètres du codeur parole.

1.5 Critères d'évaluation des méthodes

Nous terminons ce chapitre introductif par une présentation des mesures objectives qui seront utilisées tout au long de ce document pour évaluer les performances des différentes approches développées. En effet, nous comparons des algorithmes implémentés à des niveaux distincts de la chaîne de transmission et il nous faut des *mesures communes* qui nous permettent d'établir une comparaison globale entre ces méthodes.

Nous avons choisi des mesures calculées à partir du *signal de parole* restitué par le décodeur du GSM EFR. Les mesures de distance les plus communément utilisées pour la parole sont la distance log-spectrale et la distance cepstrale [Kleijn et al., 1995]. La distance log-spectrale est considérée comme la plus pertinente au plan subjectif mais on lui préfère dans la pratique la distance cepstrale. En effet celle-ci peut être considérée comme une approximation de la distance log-spectrale mais se calcule beaucoup plus aisément, à partir des coefficients de prédiction linéaire.

Cependant ces distances « classiques » sont surtout adaptées à la mesure de distorsions linéaires ou additives (filtrage, bruit additif, etc.) or la nature des dégradations observées ici est très différente. Dans le cas du masquage de l'EFR, par exemple, la perte de trames successives aboutit au final à l'annulation de tout signal en sortie du décodeur (segment de silence). Une mesure basée sur l'enveloppe spectrale (distance log-spectrale, cepstre) aura alors tendance à diverger sur des défauts de ce type. C'est pourquoi nous avons choisi d'utiliser 2 mesures complémentaires présentées dans les paragraphes qui suivent.

1.5.1 Distance cepstrale

La distance cepstrale est principalement utile pour représenter *la distribution de l'erreur au cours du temps*. En revanche, elle est moins pertinente pour la comparaison avec la procédure de masquage classique[17]. Nous précisons en premier lieu son calcul à partir des coefficients de prédiction linéaire.

La distance cepstrale sera évaluée par *trames* de 16 ms, recouvrantes de moitié, entre le signal de parole $s(t)$ en sortie de décodeur et le signal de référence $s'(t)$ correspondant **au signal décodé** par le GSM EFR **en l'absence d'erreurs de transmission**.

Considérons les jeux de coefficients cepstraux $\left\{c_{n,i}\right\}$ et $\left\{c'_{n,i}\right\}$ calculés respectivement sur les trames d'indice n du signal à évaluer $s(t)$ et de la référence $s'(t)$. La distance cepstrale d'ordre 2 entre ces deux signaux à la trame n est donnée par :

[17] Pour éviter la divergence de la distance cepstrale en cas de trame perdue résultant en un segment de silence, on rajoute un bruit de fond (bruit blanc, SNR=50dB).

$$d_{cep}(n) = \sum_{i=-N}^{N} \left(c_{n,i} - c'_{n,i} \right)^2 \tag{1.10}$$

Les coefficients cepstraux $\{c_i\}$ associés à une trame donnée[18] s'obtiennent à partir des coefficients $\{a_i\}$ de prédiction linéaire LPC calculés sur cette trame de signal à l'aide des relations suivantes [Boite et al., 1987] :

$$c_i = -a_i - \sum_{k=1}^{i-1} \left(1 - \frac{k}{i}\right) c_{i-k} a_k; \quad i > 0 \tag{1.11}$$

avec $c_i = c_{-i}$ et $c_0 = \log\left(\sigma^2\right)$ où σ^2 est la puissance du signal (mesurée sur la trame).

Le nombre N est pris égal au double de l'ordre du modèle auto-régressif de l'analyse LPC (cf. Annexe A). Dans la pratique, la différence d'énergie $\left(c_0 - c'_0\right)^2$ ne sera pas prise en compte dans le calcul (1.10) car peu significative sur le plan perceptif.

La distance cepstrale au cours du temps $d_{cep}(n)$ permet de visualiser la distribution temporelle des dégradations du signal de parole à évaluer par rapport au signal décodé sans erreurs (référence). On peut également obtenir une note unique pour le signal de parole à évaluer en calculant la moyenne des distances cepstrales $d_{cep}(n)$ sur les trames d'activité vocale. Cependant, on a préféré utiliser le critère PESQ, présenté au paragraphe suivant, afin d'obtenir une note d'évaluation globale prenant mieux en compte les pertes de trames tout comme les erreurs introduites durant les segments de non-activité vocale.

1.5.2 Distance perceptuelle PESQ (MOS estimée)

La distance cepstrale tend à pénaliser trop fortement le masquage classique de l'EFR car elle prend des valeurs très importantes dans les périodes de « coupure » associées au mécanisme de masquage d'erreur, lequel annule complètement le signal restitué au bout de plusieurs trames successives perdues.

D'autre part, il serait intéressant de prendre également en compte les dégradations générées par le décodage des erreurs résiduelles dans les zones de non-activité vocale. Mais il faut pour cela disposer d'un modèle permettant de pondérer différemment l'impact des dégradations mesurées selon qu'elles concernent des segments d'activité vocale ou de non-activité vocale. En l'absence d'un tel modèle, le calcul de la moyenne des distances cepstrales est ainsi restreint aux seuls segments d'activité vocale.

Tout ceci nous a conduit à utiliser, comme critère d'évaluation, l'algorithme PESQ dont nous donnons ici une présentation succincte.

---

[18] On omet ici l'indice de trame n pour alléger les notations.

L'algorithme PESQ a été normalisé par l'UIT-T [UIT-T, P.862] pour l'estimation de la qualité vocale téléphonique et est capable de modéliser les distorsions non-linéaires engendrées par le codage ou par les procédures de masquage d'erreur. Très schématiquement, on peut le considérer comme un algorithme calculant une distance spectrale « perceptuelle » suivie d'un modèle « cognitif » qui permet de prendre en compte le fait qu'une dégradation n'a pas le même impact selon qu'elle est additive ou soustractive, ou selon son contexte (segment de parole ou non) et sa distribution (localisée ou non).

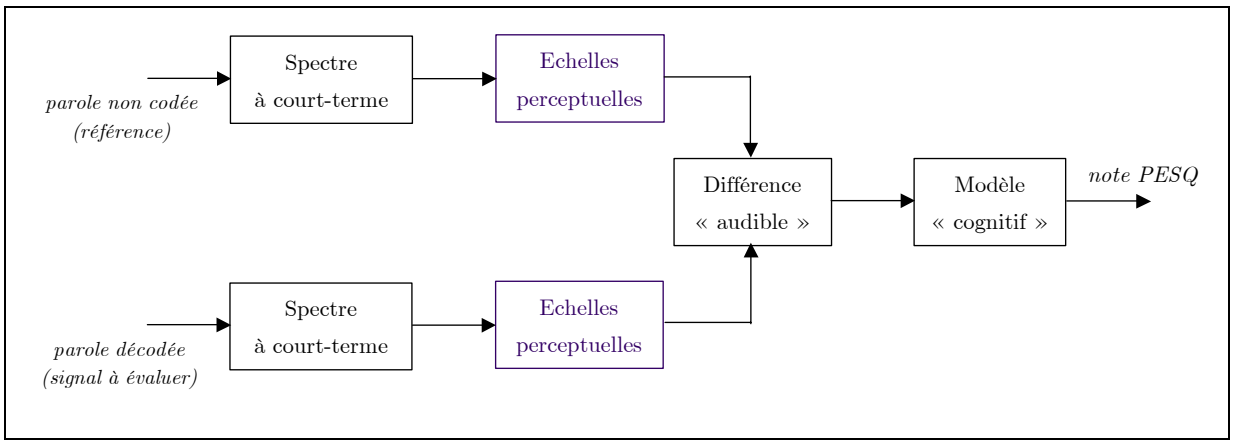

Figure 1.11 : Schéma synoptique du calcul de la distance perceptuelle PESQ

Le schéma de l'algorithme PESQ est illustré Figure 1.11. L'algorithme opère par comparaison du signal à évaluer avec une référence. La **référence** choisie est ici le signal de parole **non codé**. La distance calculée par l'algorithme PESQ entre ces deux signaux est une distance entre leurs représentations « auditives ». Ces représentations auditives sont obtenues par des transformations d'échelles du spectre de puissance calculé à partir de la transformée de Fourier à court-terme. L'utilisation d'échelles de fréquence (Bark) et d'intensité (Sonie) correspondant à des échelles perceptives [Zwicker et al., 1981] permet d'obtenir une « distance spectrale perceptuelle » plus pertinente que la distance cepstrale.

Le modèle cognitif intervient au niveau de l'intégration en temps et en fréquence (échelle des Bark) des différences entre les deux représentations « auditives » (différences « audibles »). Cette intégration permet d'obtenir une *note de qualité globale* (note PESQ) pour l'intégralité du signal à évaluer. Le modèle « cognitif » applique une pondération plus faible aux dégradations sous-tractives (signal filtré fréquentiellement ou atténué temporellement) qu'aux dégradation additives (bruit additif, par exemple). Ceci est sensé reproduire un mécanisme de la perception et permet, par exemple, de ne pas pénaliser trop fortement les dégradations (coupures) générées par le masquage classique de l'EFR. De plus, les dégradations intervenant dans les segments de non-activité vocale sont également prises en compte mais avec une pondération moindre. Enfin, l'intégration en temps et fréquence des « différences audibles » est non-linéaire afin de modéliser le fait que des erreurs isolées (en temps et/ou fréquence) ont un impact perceptif plus fort que des erreurs uniformément réparties.

La *note globale de qualité* renvoyée par l'algorithme PESQ est corrélée avec la note MOS (*Mean Opinion Score*), c'est pourquoi on utilisera indifféremment les termes « note PESQ » ou « MOS estimée » pour la désigner. La note MOS est une note de qualité *subjective* obtenue comme la moyenne des notes fournies par des sujets lors de tests d'écoute. Les notes attribuées par les sujets sont situées sur une échelle discrète (*Opinion Scores*) explicitée par le Tableau 1.1, la moyenne résultante (note MOS) étant à valeurs continues. Bien que l'échelle MOS corresponde par sa terminologie à une échelle de qualité absolue, on insiste sur le fait que la « note MOS » estimée par l'algorithme PESQ (ou note PESQ) n'est utilisée ici que comme une mesure de « distance perceptuelle » par rapport à la référence. Elle n'a donc de sens que pour une **comparaison relative** des algorithmes entre-eux et par rapport à la référence, et pour des **mêmes conditions** de test.

Opinion scores	**Qualité perçue**
5	Excellente
4	Bonne
3	Moyenne
2	Médiocre
1	Mauvaise

Tableau 1.1 : Echelle de qualité (Opinion Scores)

La corrélation moyenne entre la note MOS « réelle » et la note estimée par PESQ a été évaluée à 0,92 pour des conditions correspondant à de la parole transmise par les réseaux radio-mobiles (et pour différents niveaux de brouillage). On considère qu'un écart entre notes MOS estimées (notes PESQ) est **significatif** s'il excède **0,2 MOS**.

Chapitre 2

Détection d'artefacts introduits par le réseau GSM sur le signal de parole

2.1 Introduction

Ce chapitre s'inscrit dans la problématique du *post-traitement* du signal de parole transmis par le réseau GSM. Le but initial visé étant de développer des algorithmes traitant la parole en un point situé en *aval* de la chaîne de réception afin de réduire l'impact subjectif des dégradations rencontrées. L'avantage d'une telle stratégie est son universalité, le positionnement en aval permettant la prise en compte des divers artefacts susceptibles d'être introduits tout au long de la chaîne de transmission. En contre-partie, on dispose du minimum d'information pour traiter les dégradations liées aux erreurs de transmission puisque on a uniquement accès au signal de parole décodé. La détection de ces dégradations depuis le signal de parole constitue alors une étape cruciale conditionnant l'efficacité de tout post-traitement ultérieur.

Le chapitre introductif qui précède a dressé le tableau des principales dégradations de la qualité vocale transmise par le GSM. Notre objectif est ici de dégager les méthodes aptes à détecter ces dégradations. Nous étudions deux approches complémentaires pour détecter l'occurrence d'un artefact dans un signal de parole. La première est basée sur une modélisation du défaut à détecter et ne peut s'appliquer qu'aux artefacts bien caractérisés. Un exemple est l'artefact « voix de robot » du GSM Full Rate (FR) et pour lequel un algorithme de détection est proposé. La seconde approche utilise un modèle *a priori* de la parole et pourrait s'appliquer à une gamme plus large de dégradations. Une analyse des méthodes s'inspirant de ce principe est menée dans la seconde partie de ce chapitre et la pertinence de la stratégie *post-traitement* pour les artefacts dus aux erreurs de transmission est finalement discutée.

2.2 Principe

Nous présentons ici le principe général de la détection d'une dégradation sur un signal de parole. Ceci nous permet de dégager deux grandes catégories de méthodes de détection que nous appliquerons par la suite aux dégradations de la parole transmise par le GSM mises en évidence au chapitre précédent.

La détection de dégradations peut être envisagée comme un problème de classification illustré Figure 2.1. Le principe est d'extraire régulièrement du signal de parole, un certain nombre de paramètres associés aux modèles que l'on se donne du signal de parole attendu ou de défauts connus. On décide alors en fonction de la valeur des paramètres extraits à quelle classe affecter la trame de signal analysée : parole non dégradée ou défaut.

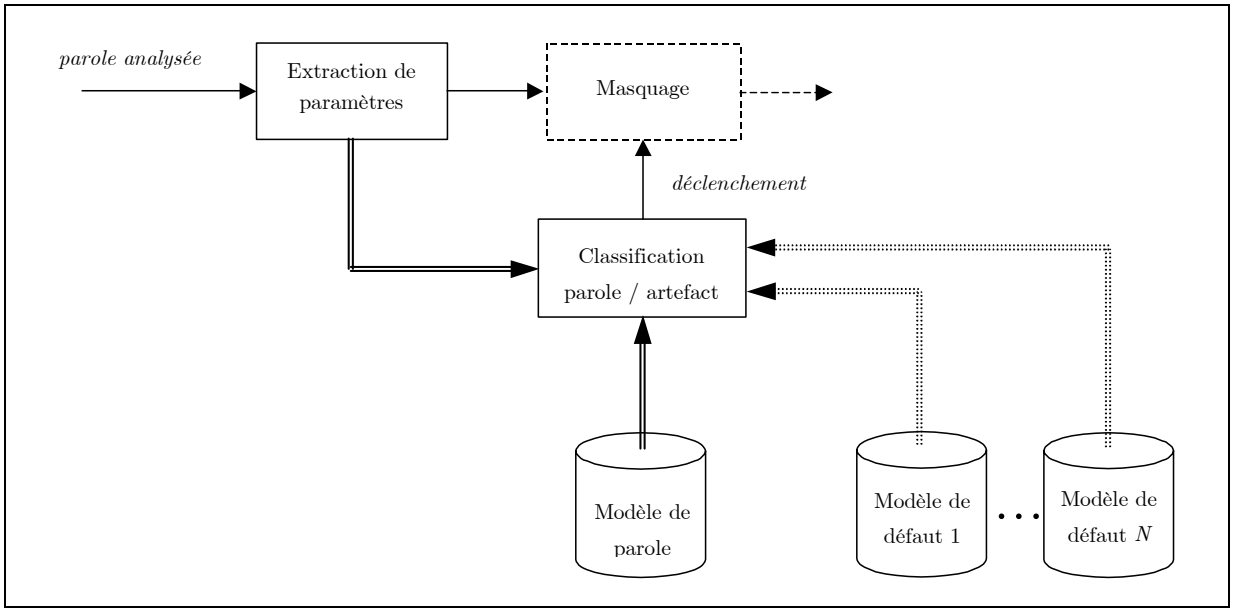

Figure 2.1 : Détection d'artefacts en sortie de décodeur parole

Cette présentation de principe très générale recouvre des situations réelles diverses.

Il est souvent plus facile d'exploiter uniquement le modèle d'un défaut bien caractérisé que l'on cherche à détecter. Le modèle de parole est implicite et réduit à la seule hypothèse qu'il ne recouvre pas celui du défaut. L'avantage de telles méthodes est leur simplicité et leur robustesse. Cette stratégie est mise en œuvre au paragraphe 2.3 pour la détection de la « voix de robot » [Veaux et al., 1999].

A l'inverse, l'idée de détecter n'importe quel type de défaut à partir du seul modèle explicite de la parole apparaît très séduisante dans le contexte de post-traitement qui est le notre. En effet, les dégradations de la parole associées aux erreurs de transmissions sont par nature très diverses et il paraît difficile de les modéliser. D'autre part, le modèle de parole utilisé pour la détection de

dégradations[19] peut être naturellement exploité pour le masquage de ces dégradations. Ceci est schématisé Figure 2.2 où une même loi a priori sur les valeurs d'un paramètre extrait du signal analysé permet de définir le critère de détection d'un artefact (seuil de rejet) ainsi que la valeur de substitution utilisée par la procédure de masquage pour ce paramètre.

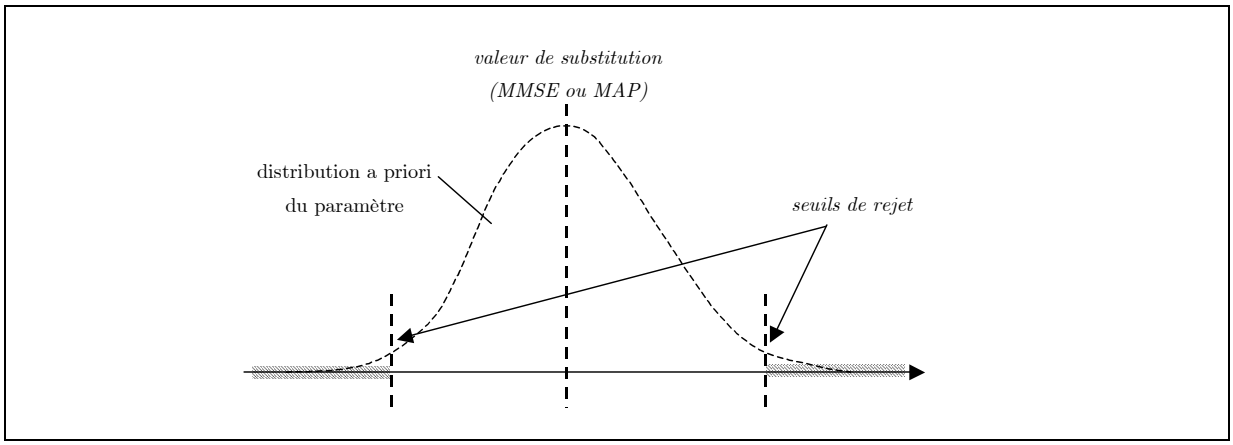

Figure 2.2 : Détection et masquage d'un artefact à partir du modèle a priori d'un paramètre pour la parole

L'application de ces principes de détection aux dégradations de la parole transmise par le GSM est l'objet des développements qui suivent. Nous présentons en particulier un algorithme de détection de la « voix de robot » du GSM FR. Cet algorithme résulte des premières études que nous avons menées afin de valider la stratégie *post-traitement* en aval du système GSM existant. L'enjeu de la présentation qui en est faite ici est d'illustrer une méthode de détection d'un défaut donné, et non de chercher à améliorer une technologie de codage qui a fait son temps.

2.3 Détection d'un artefact caractérisé : « la voix de robot »

La procédure de substitution de trame proposée pour le GSM FR peut se traduire par une perte très sensible du naturel de la voix, celle-ci devenant très « métallique ». C'est cet effet dénommé « voix de robot » que nous cherchons à détecter. Nous montrons qu'il se caractérise par la présence d'une périodicité de 50 Hz correspondant à la répétition de trame (20 ms). Nous proposons ensuite de détecter l'effet « voix de robot » à partir d'une mesure du degré de périodicité de 50 Hz appliquée au signal de parole.

[19] La détection d'une dégradation se fait alors par « rejet », c'est-à-dire lorsque les valeurs des paramètres observés sont très peu probables conditionnellement au modèle de parole. Ceci est illustré Figure 2.2.

2.3.1 Caractérisation de l'effet « voix de robot » du GSM FR

La Figure 2.3 illustre le résultat de la procédure de masquage mise en œuvre par le GSM FR. Le signal de parole décodé en l'absence d'erreur de transmission y est comparé à un segment de plusieurs trames substituées (l'indicateur BFI est superposé au signal). Il apparaît nettement une périodicité égale à la durée d'une trame (20 ms), cette périodicité est responsable de la résonance métallique perçue (50Hz).

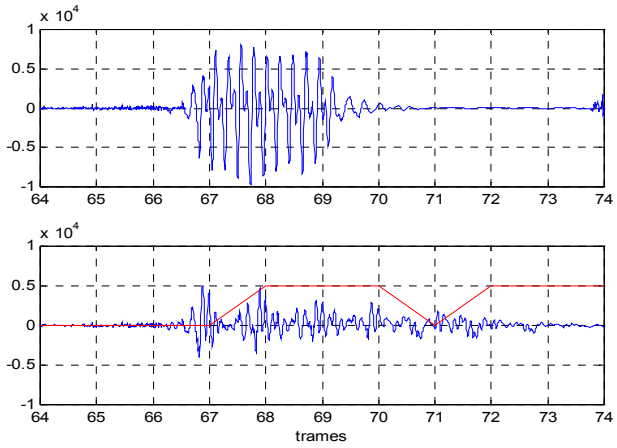

Figure 2.3 : Périodicité-trame introduite par la substitution de trames erronées ; 1- signal reçu sans erreurs ; 2- signal reçu pour C/I=5dB (avec indicateur de BFI)

La procédure de masquage mise en œuvre dans le décodeur GSM FR semble donc être à l'origine de cette périodicité-trame. Nous avançons ici une explication plus précise de ce phénomène [Paping et al., 1997] :

Lors de la substitution de trame, les *paramètres RPE-LTP* (cf. Annexe A) de la dernière trame[20] valide sont *répétés* en entrée du décodeur, mis à part les gains de calibration qui sont diminués à chaque répétition (i.e. toutes les 20 ms) et les grilles des 4 séquences RPE mais celles-ci sont définies par un offset qui ne varie que de 0 à 3 échantillons. Si l'on néglige la variation des grilles RPE, on peut alors considérer que les 4 séquences RPE de la dernière trame valide sont répétées à l'identique pour chaque nouvelle trame substituée, c'est-à-dire que le signal d'excitation $\tilde{r}(nT_e)$[21] du filtre de synthèse LTP est *pseudo-périodique* de période $T_r = 20$ ms. Ceci explique l'apparition d'une périodicité-trame (20ms) dans le signal synthétisé. Plus précisément, *durant la procédure de substitution de trame*, le signal d'excitation $\tilde{r}(nT_e)$ du filtre de synthèse LTP peut être modélisé par :

$$\tilde{r}(nT_e) \simeq \alpha(nT_e)\ r_0(nT_e) \otimes \Pi_{T_r}(nT_e) \qquad (2.1)$$

[20] Le décodeur GSM FR reçoit toutes les 20 ms une trame de 260 bits codant 1 filtre LPC, 4 filtres LTP et 4 séquences d'excitation RPE de 5 ms.

[21] T_e désigne la période d'échantillonnage. La fréquence d'échantillonnage $F_e = 1/T_e$ est ici égale à 8 kHz.

où $\otimes$ désigne l'opérateur de convolution, $r_0(nT_e)$ est le *signal d'excitation* de 20 ms (4 séquences RPE) codé par la dernière trame valide reçue, Π_{T_r} correspond à un *peigne de Dirac* de période $T_r = 20$ ms et $\alpha(nT_e)$ est un *facteur d'atténuation* diminuant par sauts toutes les 20 ms.

Ainsi, si on considère le signal $x_w(nT_e)$ en sortie du décodeur, analysé au travers de la *fenêtre à court-terme* w, son spectre s'écrit :

$$X_w(f) = \alpha\ \omega(f) \otimes [r_0(f)\Pi_{F_r}(f)\frac{1}{P_0(f)}\frac{1}{A_0(f)}],\ F_r = 1/T_r = 50\ \text{Hz}, \tag{2.2}$$

où on a supposé le facteur d'atténuation $\alpha(nT_e)$ constant à l'intérieur de la fenêtre d'analyse et noté $1/P_0(f)$ et $1/A_0(f)$ les fonctions de transfert des filtres LTP et LPC codés par la dernière trame valide[22]. En notant $X_0(f)$, le spectre du signal synthétisé pour la *dernière trame valide* reçue, cette relation devient :

$$X_w(f) = \alpha\ w(f) \otimes [X_0(f)\Pi_{F_r}(f)]. \tag{2.3}$$

Le spectre synthétisé $X_w(f)$ s'apparente donc au *produit* du spectre de la dernière trame valide reçue avec un peigne Π_{F_r}.

Pour vérifier l'hypothèse selon laquelle la variation des grilles RPE n'est pas suffisante pour « détruire » la périodicité $T_r = 20$ ms introduite par la répétition des autres paramètres du codeur, nous avons simulé la procédure de substitution du GSM FR. Ainsi, nous avons d'abord simulé le fonctionnement normal du décodeur RPE-LTP en excitant un filtre de synthèse LTP $1/P(z)$ par des séquences d'impulsions RPE d'amplitude et d'offset aléatoires. Pour simplifier, les paramètres β et P du filtre ont été pris invariants (dans la pratique, ces paramètres sont réactualisés pour chaque séquence RPE de 5 ms, aussi les résultats présentés ici ne sont valables que pour les zones voisées stationnaires de la parole). Le spectre du signal synthétisé pour $\beta = 0.9$ et $P = 42$ (soit une fréquence fondamentale $F_0 = 190Hz$) est illustré sur la Figure 2.4 (haut). On a simulé ici la seule partie LTP du décodeur.

Nous avons ensuite simulé la procédure décrite dans la norme [*GSM*, 06.10] pour masquer les trames perdues (à partir de la seconde trame perdue). C'est-à-dire que pour chaque nouvelle trame, nous avons répété les 4 séquences d'impulsions RPE de la trame précédente avec simplement[23] un offset aléatoire pour les trains d'impulsions (*grille RPE*). Le spectre obtenu en sortie du filtre LTP avec $\beta = 0.9$ et $P = 42$ est illustré sur la Figure 2.4 (bas). On voit clairement que le spectre est modulé par des *harmoniques de 50 Hz*. La procédure de substitution de trames donnée dans la norme GSM

[22] Dans le cas du filtre LTP, il s'agit d'un spectre moyen sur l'ensemble de la trame de 20 ms (i.e. moyenne des 4 filtres LTP).

[23] Nous n'avons pas simulé le facteur d'atténuation car il n'intervient pas vraiment dans l'hypothèse que l'on veut vérifier ici, à savoir que les variations d'offset ne sont pas suffisantes pour détruire la périodicité de l'excitation. Si l'on tenait compte du facteur d'atténuation, on aurait simplement une excitation périodique amortie.

engendre donc bien des distorsions semblables à celles décrites pour les signaux analysés précédemment (« voix de robot »).

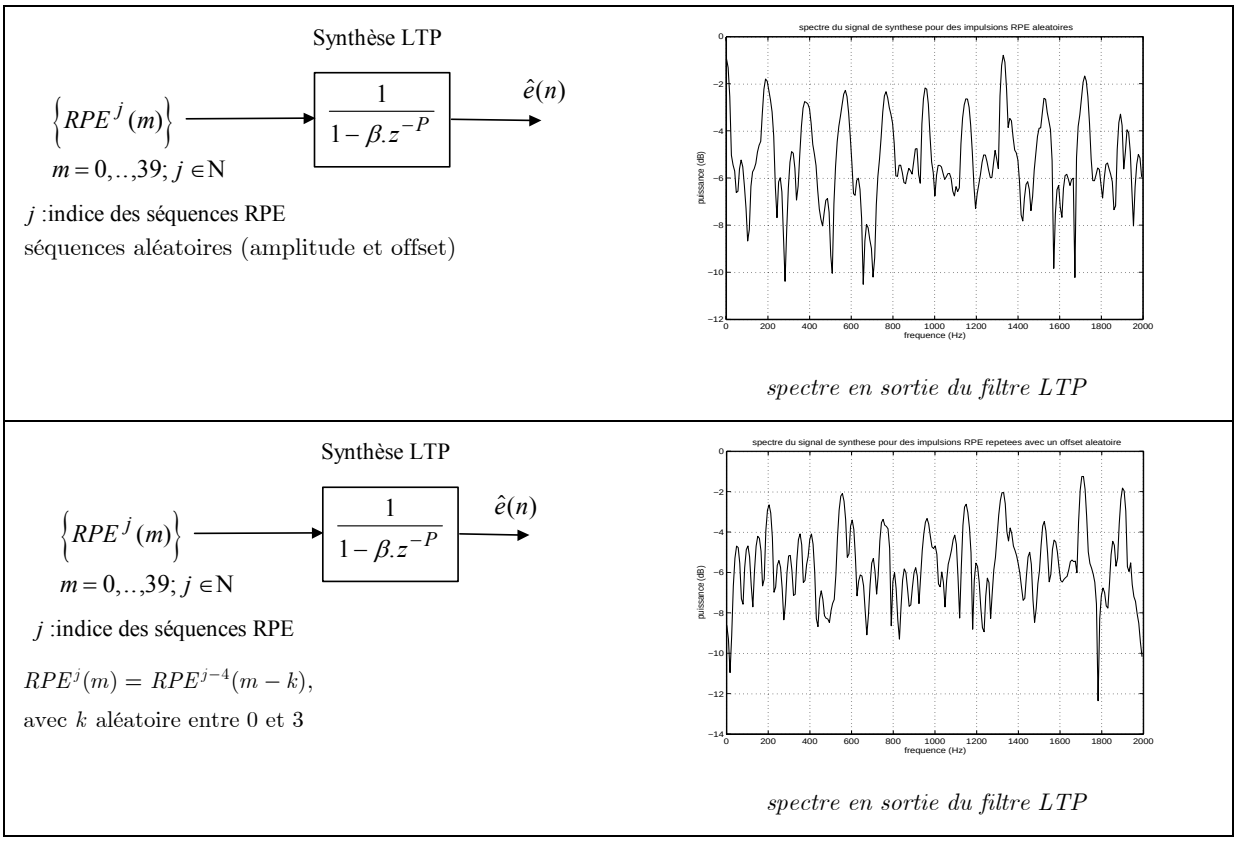

Figure 2.4 : Explication de l'artefact « voix de robot » produit par le GSM FR

2.3.2 Détection des occurrences de « voix de robot »

L'équation (2.3) caractérise l'effet « voix de robot » par la *présence d'harmoniques de 50 Hz* dans le signal en sortie du décodeur parole. La détection de l'artefact « voix de robot » se réduit donc à la détection de telles harmoniques ou de la périodicité $T_r = 20$ ms dans le domaine temporel. On opte pour une méthode temporelle plutôt que fréquentielle et on utilisera l'inter-corrélation normalisée pour mesurer la périodicité à long-terme du signal. Cette méthode est déjà utilisée par le GSM FR pour calculer le filtre LTP et permet une meilleure résolution temporelle qu'une méthode fréquentielle. Plus précisément, on considère des trames de L échantillons du signal de parole $s(n)$:

$$\mathbf{s}_n = [s(nT_e),...,s((n-L+1)T_e)]^T,\ n \in \mathbb{N} \tag{2.4}$$

où T désigne l'opérateur *transposé*.

On définit *l'inter-corrélation normalisée* $\rho_s(n,k)$ à l'instant n et pour le décalage k, selon :

$$\rho_s(n,k) = \frac{\mathbf{s}_n^T.\mathbf{s}_{n-k}}{\|\mathbf{s}_n\|\|\mathbf{s}_{n-k}\|} \text{ avec } P_{\min} \le k \le P_{\max} \tag{2.5}$$

où $\|\mathbf{s}_n\|^2 = \mathbf{s}_n^T\mathbf{s}_n$.

On choisit ici des trames de longueur $L = 160$ échantillons à la fréquence d'échantillonnage $F_e = 8$ kHz. La fonction d'inter-corrélation $\rho_s(n,k)$ est évaluée pour les décalages k compris entre $P_{\min} = 20$ et $P_{\max} = 165$ échantillons de manière à inclure les valeurs *naturelles* du pitch[24] de la parole et la valeur de la périodicité à détecter ($T_r = 20$ ms soit 160 échantillons).

On remarquera que l'inter-corrélation (2.5) diffère de celle calculée par le codeur du GSM FR (cf. Annexe A) puisqu'on n'a pas effectué la simplification $\|\mathbf{s}_n\| = \|\mathbf{s}_{n-k}\|$ au dénominateur. Cette simplification n'est en effet plus valide lors du masquage de trame puisque les trames substituées sont progressivement atténuées.

A partir de l'expression (2.5), on définit *un critère de détection* de l'effet « voix de robot » à l'instant n, selon :

$$\rho_s(n, k = N_r) > \lambda \text{ avec } 0 < \lambda < 1 \tag{2.6}$$

où N_r correspond à la périodicité T_r exprimée en nombre d'échantillons : $N_r = T_r F_e = 160$.

On remarque que l'évènement « voix de robot » est associé au *dépassement d'un seuil* λ par l'inter-corrélation $\rho_s(n,k)$ évaluée *pour le décalage k associé à la période* N_r et non au fait que le *maximum* selon k de la fonction d'inter-corrélation $\rho_s(n,k)$ coïncide avec la période N_r. La justification de ce choix tient au fait que pour un instant n donné, la fonction d'inter-corrélation $\rho_s(n,k)$ peut présenter à la fois des pics pour la période N_r et pour le pitch P_0 de la trame répétée par la procédure de substitution. Suivant les caractéristiques de la trame répétée (degré de voisement) et l'atténuation appliquée aux trames substituées, le maximum de l'inter-corrélation peut correspondre au pitch P_0 ou à la période de répétition N_r.

Pour évaluer les performances de détection d'un tel critère, on a calculé sa *caractéristique optimale de réception* (courbe COR) qui représente les couples $\left(P_{fa}, P_d\right)$ de probabilités de fausses alarmes et de détection correcte obtenues en variant le seuil de décision λ. Pour cela, il est nécessaire de disposer d'une référence indiquant l'occurrence de l'artefact à détecter. On utilisera ici l'indicateur BFI renvoyé par le décodeur canal, couplé à une détection d'activité vocale DAV. A partir de ces deux indicateurs, on considérera que l'effet voix de robot est présent (indicateur RV) dès que *deux trames de parole successives* sont perdues :

$$RV_n = \left(DAV_n \cup DAV_{n-1}\right) \cap \left(BFI_n \cap BFI_{n-1}\right) \tag{2.7}$$

[24] La justification de ce choix apparaîtra au paragraphe 2.3.2.1.

Les résultats de détection du critère (2.6) sont illustrés Figure 2.5. On constate que ce critère est peu discriminant, la détection de l'effet voix de robot s'accompagne d'un taux de fausses alarmes rapidement élevé.

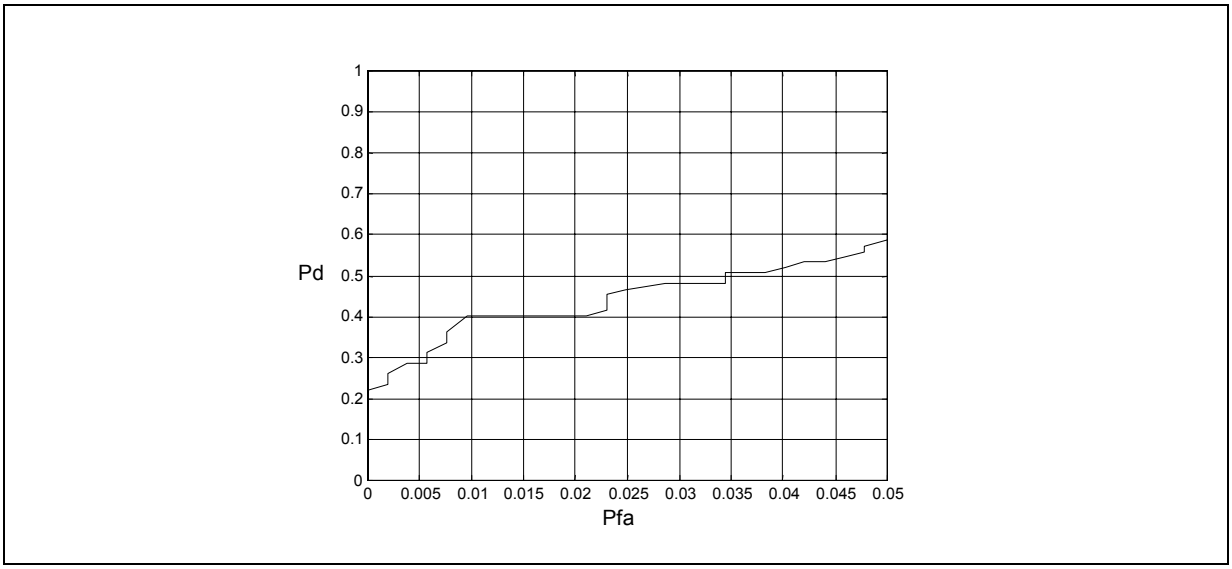

Figure 2.5 : Courbe COR du détecteur basé sur la règle de décision (2.6)

Ce phénomène s'explique de la façon suivante :

Pour un signal de périodicité à long-terme P, l'inter-corrélation $\rho_s(n,k)$ est elle-même pseudo-périodique de période P selon k. Ainsi, un pic de l'inter-corrélation en $k = N_r$ peut correspondre aussi bien à une périodicité due à la répétition de trame qu'à une trame de parole dont le *pitch* P est un sous-multiple de N_r. Pour lever cette ambiguïté, il est nécessaire d'adjoindre à la règle de décision (2.6) une *estimation du pitch* du signal de parole analysé.

2.3.2.1 Réduction des fausses alarmes par estimation robuste du pitch

Afin de diminuer le taux de fausses alarmes, on propose désormais la *règle de détection* suivante pour l'artefact « voix de robot » :

$$\rho_s(n,k = N_r) > \lambda \text{ et } \hat{P}(n) \neq N_r / l \ ; \ l = 2,3,... \tag{2.8}$$

où $\hat{P}(n)$ correspond à la valeur (en échantillons) du *pitch estimé* à partir de la trame de parole $\mathbf{s}_n$ à l'instant n. Le cas $\hat{P} = N_r$ sera considéré comme une *répétition de trame* car c'est une valeur assez improbable pour le pitch naturel de la parole. L'estimée $\hat{P}(n)$ du pitch est ainsi utilisée pour distinguer entre la « périodicité-pitch » et une périodicité multiple du pitch.

Une première idée est d'obtenir l'estimée $\hat{P}(n)$ comme le maximum *absolu* de l'inter-corrélation $\rho_s(n,k)$:

avec $$\hat{P}(n) = \underset{P_{\min}<k<P_{\max}}{\arg\max}\ \rho_s(n,k) \tag{2.9}$$

Cependant, la corrélation à court-terme du signal de parole vient moduler l'amplitude des pics de l'inter-corrélation ρ_s si bien que l'estimée (2.9) ne peut elle-même pas toujours discriminer la valeur réelle du pitch de celle de l'un de ses multiples. C'est pourquoi on effectue généralement [Hess, 1983] un *pré-traitement* du signal de parole destiné à éliminer la contribution des formants (associés à la corrélation à court-terme du signal de parole, cf. Annexe A).

2.3.2.1.a Blanchiment linéaire à court-terme

Le pré-traitement communément utilisé pour éliminer la contribution des formants consiste en un filtrage prédictif à court-terme du signal $s(n)$ selon :

$$e(n) = s(n) - \sum_{k=1}^{r} a_k s(n-k) \tag{2.10}$$

où les coefficients a_k sont les coefficients de l'analyse LPC d'ordre r (cf. Annexe A).

L'inter-corrélation à l'instant n et pour le décalage k est alors calculée sur le signal résiduel selon :

$$\rho_e(n,k) = \frac{\mathbf{e}_n^T \mathbf{e}_{n-k}}{\|\mathbf{e}_n\|\|\mathbf{e}_{n-k}\|}, \text{ avec } \mathbf{e}_n = [e(n),...,e(n-L+1)]^T \tag{2.11}$$

L'inter-corrélation du signal résiduel $\rho_e(n,k)$ est alors substituée à l'inter-corrélation du signal de parole $\rho_s(n,k)$ dans l'estimateur (2.9) du pitch. L'application de la règle de décision (2.8) avec cette nouvelle estimation du pitch conduit aux résultats illustrés Figure 2.6. On constate une légère amélioration des performances de détection mais celle-ci présente encore un taux de fausses alarmes trop élevé.

En fait, le blanchiment à court-terme réduit légèrement les erreurs d'estimation entre valeur réelle du pitch et valeurs multiples mais rehausse en contre-partie les composantes de bruit qui viennent alors masquer la périodicité à long-terme du résidu $e(n)$.

Ceci nous a conduit à proposer un nouvel estimateur de pitch. L'idée est de limiter la recherche du maximum de l'inter-corrélation (2.11) à l'intérieur d'une plage restreinte autour d'une *première estimée* assez grossière du pitch. Cet estimateur procède donc en deux étapes. La première étape ne cherche pas à renvoyer une estimation fine du pitch mais doit parfaitement discriminer les valeurs multiples du pitch. Elle est basée sur un blanchiment non-linéaire du signal de parole $s(n)$.

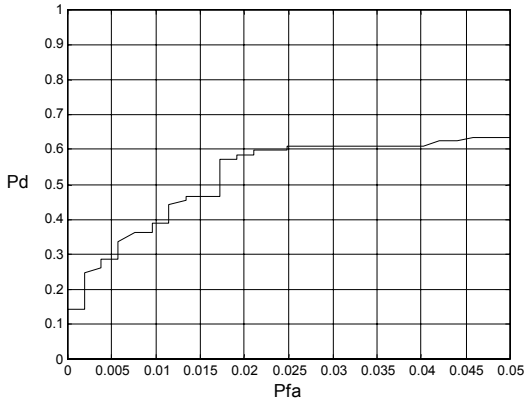

Figure 2.6 : Courbe COR du détecteur basé sur la règle de décision (2.8) et une estimation du pitch sur le résidu de prédiction à court-terme

2.3.2.1.b Blanchiment non-linéaire à court-terme

Dans la première étape, l'estimation du pitch se fait directement sur la forme d'onde du signal après transformation non-linéaire. Il s'agit en fait d'une *détection des impulsions glottiques*, la transformation non-linéaire ayant pour but de réduire la contribution des formants d'où l'expression de « blanchiment non-linéaire ». Cette transformation non-linéaire a été originellement proposée par [Dogan, 1992] comme un estimateur à court-terme du cumulant d'ordre 3 normalisé. Nous l'interpréterons ici plutôt comme un *filtrage adapté* (2.12) après *expansion cubique* (2.13) du signal selon :

$$y(n) = h_1(n) \otimes z(n) \tag{2.12}$$

avec
$$z(n) = s^3(n) / E_n \tag{2.13}$$

et
$$E_n = h_0(n) \otimes s^2(n)\,. \tag{2.14}$$

où E_n est l'énergie du signal mesurée à l'intérieur d'une fenêtre de Hamming h_0 de 100 points centrée en n et h_1 est une fenêtre de Harris de 11 points. Ces deux fenêtres sont illustrées Figure 2.7.

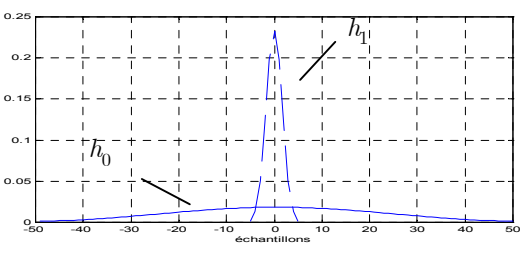

Figure 2.7 : Fenêtres utilisées pour le rehaussement des impulsions glottiques

De part la non-linéarité cubique, les zones de forte amplitude relative à l'intérieur de la fenêtre h_0 sont fortement rehaussées. On peut ainsi réduire la contribution des formants par rapport aux impulsions glottiques en choisissant une longueur adaptée pour la fenêtre h_0 (on prend ici le pitch moyen de la parole). On effectue ainsi une sorte de blanchiment non-linéaire du signal. L'effet de l'expansion cubique (2.13) sur le signal dans le domaine temporel est représenté Figure 2.8.

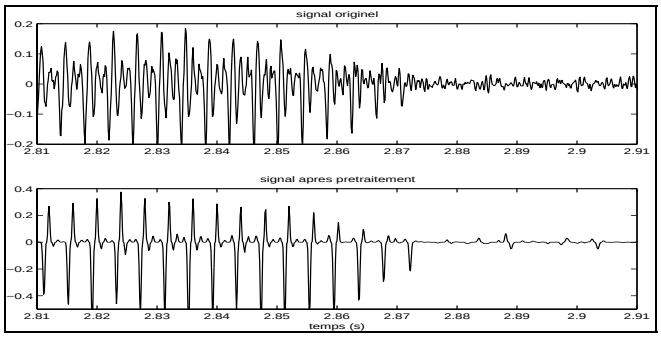

Figure 2.8 : Rehaussement des impulsions glottiques du signal par expansion cubique (haut : signal $s(n)$ analysé, bas : signal transformé $z(n)$)

Ce rehaussement apparaît de manière encore plus explicite sur les spectres des signaux. La Figure 2.9 compare ainsi le spectre du signal après expansion cubique $z(n)$ avec celui du signal originel $s(n)$ et celui du résidu LPC $e(n)$ qui correspond à la technique de blanchiment linéaire. La transformation (2.12) permet d'améliorer très nettement le rapport signal à bruit pour les zones voisées, elle possède également un effet blanchissant puisque la contribution du conduit vocal a été quasiment éliminée dans le signal $z(n)$. On remarquera par contre que l'analyse LPC, si elle permet une meilleure égalisation de l'amplitude des harmoniques, dégrade très sensiblement le rapport signal à bruit.

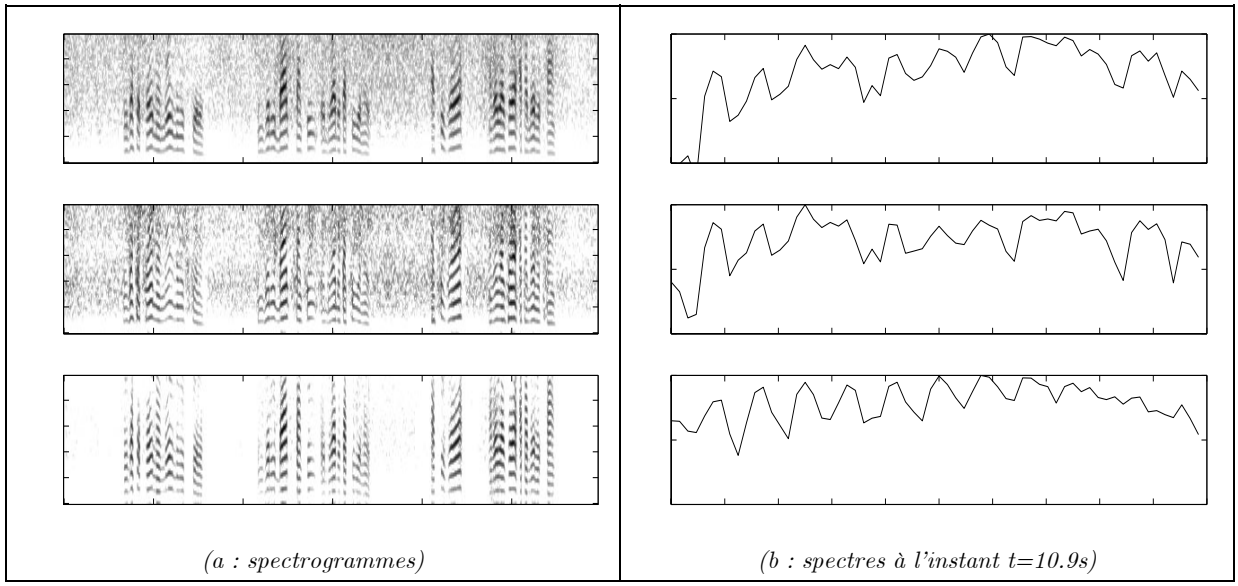

Figure 2.9 : Comparaison dans le domaine spectral des blanchiment linéaire (LPC) et non-linéaire ; (haut : signal $s(n)$ analysé, milieu : résidu LPC $e(n)$, bas : signal $z(n)$)

2.3.2.1.c Méthode combinée

A partir du signal transformé $y(n)$, on peut désormais détecter de manière fiable les impulsions glottiques en utilisant un simple seuil adaptatif (représenté en pointillés Figure 2.10). Ce seuil est calculé par lissage à oubli exponentiel de l'enveloppe de puissance à court-terme de $y(n)$. A l'intérieur d'une trame de 160 échantillons (20ms), on sélectionne le train d'impulsions de même signe *le plus énergétique* et on estime le pitch comme la distance moyenne entre ces impulsions supposées correspondre aux impulsions glottiques.

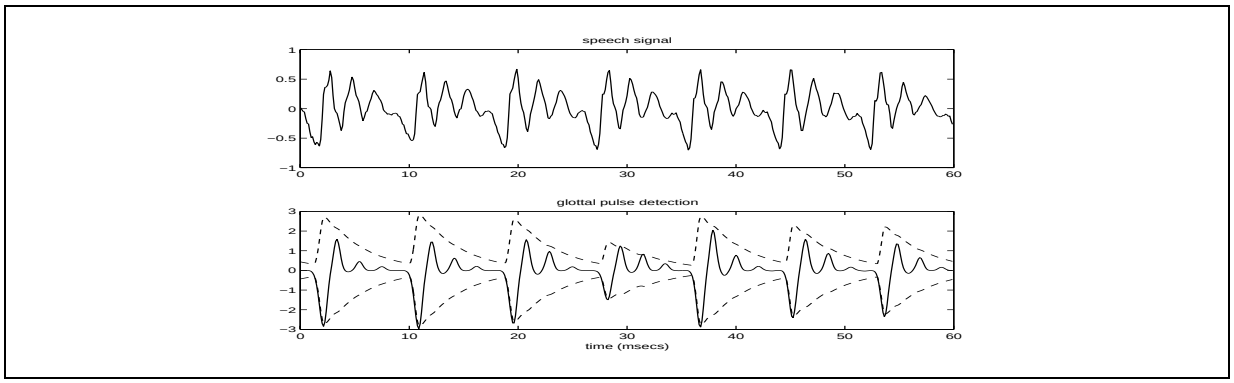

Figure 2.10 : Détection des impulsions glottiques sur le signal transformé $y(n)$ (bas) ; le signal analysé $s(n)$ est représenté pour comparaison (haut)

Cette méthode s'inspire des *méthodes temporelles à deux seuils* [Laroche, 1995] classiquement utilisées pour estimer le pitch avec une complexité réduite. Elle renvoie une estimation assez grossière du pitch mais qui n'est plus sujette aux erreurs d'estimation entre valeurs multiples et sous-multiples. Cette estimée est utilisée lors de la seconde étape de l'estimation combinée dont le diagramme est illustré Figure 2.11.

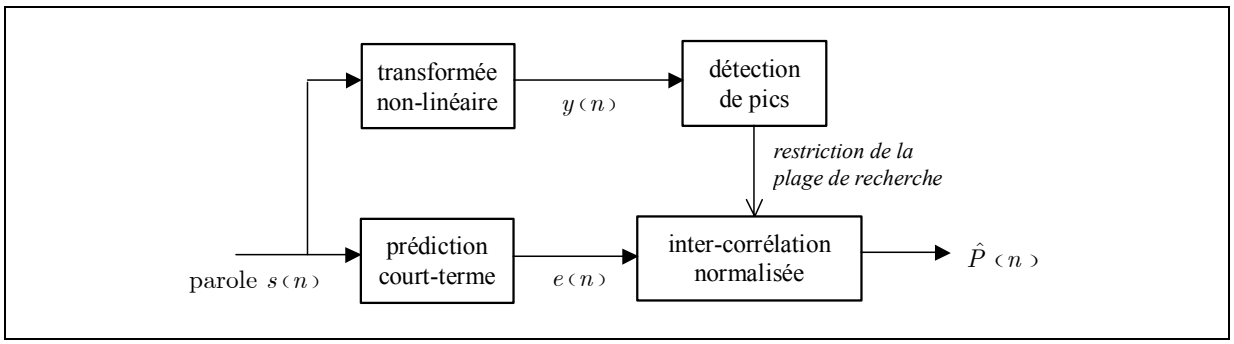

Figure 2.11 : Diagramme de l'estimation robuste du pitch

L'estimée préliminaire du pitch est raffinée par calcul de l'inter-corrélation $\rho_e(n,k)$ sur le résidu de prédiction linéaire. La recherche du maximum de cette inter-corrélation $\rho_e(n,k)$ est restreinte à une plage de décalages k centrée sur l'estimée préliminaire. Les performances de cet estimateur de pitch sont comparées Figure 2.12 à celles de l'estimation basée sur l'inter-corrélation (2.5) uniquement.

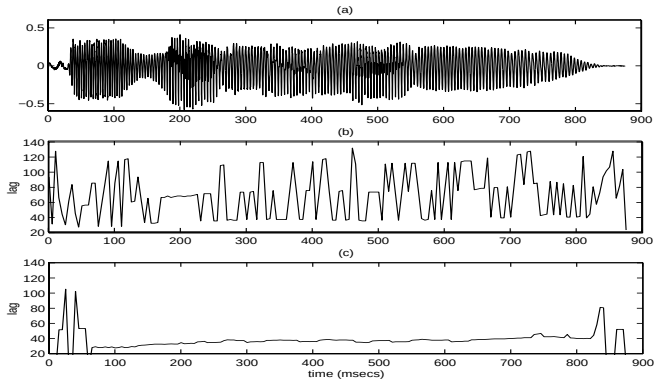

Figure 2.12 : Performances de l'estimateur de pitch proposé ;
1-signal de parole ; 2-pitch estimée par inter-corrélation
seule ;3- pitch estimé par méthode combinée

On constate que toute ambiguïté sur la valeur réelle du pitch est levée, l'estimée $\hat{P}(n)$ du pitch peut ainsi être efficacement utilisée dans la règle de décision (2.8). Les performances de la détection de l'effet voix de robot (2.8) utilisant ce nouvel estimateur de pitch sont représentées par la courbe COR illustrée Figure 2.13. On observe une diminution de taux de fausses alarmes puisque l'on peut détecter près de 40% des artefacts « voix de robot » sans faire de fausse alarmes. En revanche, la probabilité de détection ne croit ensuite que très lentement au-delà de la valeur de 50%. Il convient cependant de rappeler que la référence des évènements « voix de robot » utilisée dans le calcul des probabilités de détection présentées ici est obtenue « artificiellement » à partir de l'indicateur BFI selon la règle (2.7) et non pas par étiquetage manuel des signaux analysés. La limite supérieure (proche de 65%) systématiquement observée pour la valeur de la probabilité de bonne détection de chacun des critères étudiés ici s'expliquerait ainsi par les imperfections de l'étiquetage automatique (2.7).

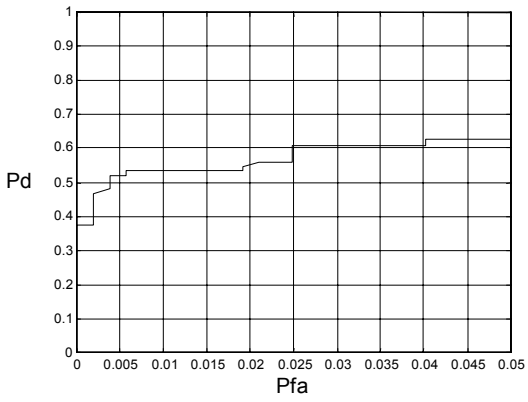

Figure 2.13 : Courbe COR du détecteur basé sur la règle de décision (2.8) et l'estimation robuste de pitch

2.3.3 Discussion

Les développements qui précèdent sont à considérer comme une première étude d'un cas simple de détection d'artefacts introduits par le réseau GSM sur le signal de parole. La détection mise en œuvre s'appuie sur un modèle *a priori* de l'artefact « voix de robot » caractérisé par la présence d'une périodicité de 50 Hz dans le signal. Cet artefact est évidemment très spécifique puisqu'il est engendré par la procédure de substitution du GSM FR. Il n'en constitue pas moins un des défauts les plus fréquemment rencontrés dans la parole GSM FR transmise sur un canal bruité et dont le masquage apparaissait intéressant dans une problématique de post-traitement du signal de parole.

D'autre part, on a vu au Chapitre 1 que la procédure de substitution du GSM EFR introduisait elle-même un défaut de type « voix métallique » dans le signal de parole décodé. L'algorithme de détection de la « voix de robot » du GSM FR n'est pas directement applicable à la « voix métallique » du GSM EFR dont les caractéristiques sont différentes[25]. Néanmoins, certaines conclusions faites ici sont généralisables au cas du GSM EFR. En particulier, ces deux types d'artefacts coïncident avec une *substitution* de trame et le seul post-traitement envisageable se limite à la mise en oeuvre d'une *version améliorée de substitution* de trame. Un déclenchement intempestif de cette substitution de trame par le post-traitement aurait un impact extrêmement négatif sur la qualité perçue. C'est pourquoi on recherche en priorité une procédure de détection minimisant le *taux de fausses alarmes*.

Sous ces conditions, l'algorithme développé pour détecter la « voix de robot » ne permet pas de détecter plus de 50% des occurrences de cet artefact. Néanmoins, le critère de détection utilisé mesure le degré de périodicité de 50 Hz dans le signal et a donc une signification sur le plan perceptif. Aussi, on peut considérer que les 50% d'occurrences détectées correspondent à celles qui sont les plus perceptibles. Une amélioration de la qualité de la parole est donc envisageable par la mise en œuvre d'une technique de substitution de trame améliorée conditionnée aux détections de cet algorithme.

Cependant, la stratégie de détection basée sur une connaissance *a priori* des défauts rencontrés a ses limites. Elle s'applique très bien au cas des artefacts générés systématiquement par des procédures implémentées en amont comme les procédures de substitution des GSM FR et EFR. En revanche, les dégradations associées au décodage direct de paramètres du codeur corrompus par des erreurs de transmission apparaissent difficilement modélisables *a priori*. Une stratégie de détection de dégradations par rejet à partir d'un modèle de parole *a priori* paraît pouvoir s'appliquer avec plus de généralité à la diversité des dégradations susceptibles d'être rencontrées en aval de la chaîne de réception.

---

[25] Ceci explique les nuances de vocabulaire utilisées ici pour désigner ces deux types d'artefacts. Sur un plan perceptif, la « voix de robot » est plus perçue comme une vibration artificielle (rugosité) alors que la « voix métallique » s'apparente à un excès de composantes tonales (harmoniques pures).

2.4 Exploitation d'un modèle a priori sur la parole

L'objectif de cette partie n'est pas de faire un état de l'art exhaustif des méthodes proposées pour la détection de dégradations à partir d'un modèle de parole. Nous cherchons plutôt à en dégager les principaux axes, en mettant en évidence les potentialités et les limitations de cette stratégie et en nous replaçant dans la problématique plus large du *post-traitement*. Nous tentons notamment d'analyser les conditions et la pertinence d'une mise en œuvre de telles méthodes couplées à des procédures de masquage des dégradations en aval de la chaîne de réception GSM.

2.4.1 Modèles pour la détection de dégradations

Nous présentons ici différentes techniques de détection des artefacts de la parole à partir d'un modèle *a priori* de la parole. On distinguera les méthodes qui exploitent uniquement la *non-uniformité* d'un paramètre extrait de la parole et les méthodes qui prennent en compte la *corrélation temporelle* de la suite de ses valeurs.

2.4.1.1 Exploitation de la non-uniformité des paramètres de la parole

La technique la plus simple consiste à considérer la valeur d'un paramètre à un instant donné indépendamment de ses valeurs passées. On décide alors de la présence d'un artefact lorsque la valeur observée est très peu probable pour la parole. Ceci n'est évidemment possible que si certaines valeurs sont nettement moins probables que d'autres sous l'hypothèse de la parole. Autrement dit, on exploite la *non-uniformité* de la distribution du paramètre pour la parole.

Cette distribution *a priori* des paramètres de la parole peut être apprise à partir d'une base de donnée de parole. Ainsi, pour évaluer les dégradations audibles, [Bayya et al., 1996] mesurent une distance entre les spectres LPC estimés sur le signal de parole analysé et ceux appris par l'algorithme de la K-moyenne (cf. Annexe A) sur une base de donnée de parole.

Le modèle *a priori* des paramètres peut également correspondre à un modèle physique de production de la parole. Une dégradation est alors détectée comme étant une configuration inadmissible du modèle de production. Ainsi, une paramétrisation du conduit vocal, obtenue à partir des coefficients de corrélation partielle PARCOR (cf. Annexe A) de la parole, peut être utilisée pour caractériser une dégradation associée à une violation des contraintes physiques du conduit vocal [Gray et al., 2000].

On peut s'interroger sur la pertinence de telles méthodes dans le cas de la parole codée par le GSM EFR puisque les paramètres spectraux reçus (LSF) sont toujours les éléments d'un dictionnaire de quantification vectorielle (QV) appris sur la parole. On rappelle cependant que la quantification des coefficients LSF n'est pas conjointe pour des raisons de complexité. Plus précisément, dans le cas du GSM EFR, les coefficients LSF sont divisés en cinq sous-ensembles quantifiés chacun par un index

transmis séparément (cf. Annexe A). Ainsi, il n'y a aucune garantie que les coefficients LSF utilisés par le décodeur correspondent à une configuration admissible pour la parole.

2.4.1.2 Exploitation de la corrélation temporelle des paramètres de la parole

Pour certaines dégradations, les valeurs prises isolément au cours du temps par les différents paramètres de la parole peuvent sembler vraisemblables mais c'est la séquence de ces valeurs au cours du temps qui ne l'est pas. Pour détecter ce type de dégradations, le modèle a priori doit prendre en compte les observations passées.

Ce principe a notamment été mis en œuvre pour une détection d'erreur sur les paramètres reçus au niveau du décodeur parole[26]. Les méthodes proposées méritent d'être exposées ici car elles sont facilement transposables à une approche de type post-traitements. Ainsi, dans le cas d'un codeur de type CELP, [Görtz, 1997] montre que des paramètres comme le délai LTP, le gain d'excitation, et la 1ère LSF exhibent une forte corrélation temporelle. Cette corrélation temporelle est exploitée pour une détection d'erreur complémentaire au niveau du décodeur parole. Le principe utilisé consiste simplement à comparer les *variations des paramètres* reçus à un seuil et à décider qu'il s'agit d'une erreur dès que le seuil est franchit. De la même façon, [Hindelang et al., 1997] améliorent la détection d'erreur au niveau décodeur GSM FR en observant les *variations de l'énergie* (estimée dans ce contexte à partir des paramètres du codeur parole) et en comparant ces variations à des seuils moyens pour la parole. Ces méthodes basées sur une statistique des variations de paramètres ont l'avantage de la simplicité, cependant elles conduisent toutes à des taux de fausse alarme élevés.

Les approches se situant au niveau du signal de parole lui-même peuvent exploiter des modèles plus globaux de la parole car ils ne sont pas assujettis aux paramètres spécifiquement calculés par un codeur. Des modèles de corrélation entre vecteurs successifs de coefficients spectraux extraits du signal de parole peuvent être appris sur une base de parole et utilisés pour la détection de séquences d'erreurs. Ainsi, [Lindblom et al., 2000] modélisent la loi jointe de l'ensemble des coefficients LSF extraits sur deux trames successives de parole. Cette loi jointe est représentée par un modèle multi-gaussien [Hedelin et al., 2000] ce qui permet de réduire la complexité de l'apprentissage. Enfin, les modèles de production de la parole peuvent à nouveau être exploités, notamment pour prendre en compte les contraintes articulatoires dans la détection d'erreurs par séquences [Gray et al., 2000] .

2.4.2 Pertinence d'une mise en œuvre aval de ces modèles

Le bref aperçu qui a été dressé des méthodes exploitant un modèle de parole pour la détection de dégradations permet néanmoins de dégager des conclusions quant à leur intérêt dans le contexte du post-traitement de la parole transmise par le GSM.

---

[26] Nous développerons ce type d'approches au chapitre suivant.

On a présenté au paragraphe 2.2, les caractéristiques qui rendent ces méthodes particulièrement intéressantes pour la détection des dégradations observées en aval d'une chaîne de réception :

- La possibilité de détecter une large classe de dégradations puisque le modèle a priori porte sur la parole et non sur les dégradations.

- La possibilité d'utiliser le modèle *a priori* sur les paramètres de la parole pour un masquage ultérieur des dégradations. Ceci est surtout le cas des méthodes exploitant la corrélation temporelle des paramètres.

Cependant, il apparaît que les paramètres modélisés par ces méthodes sont essentiellement les paramètres spectraux (ou les caractéristiques du conduit vocal). Ceci n'est pas étonnant car ces paramètres, qui sont les plus importants vis-à-vis de la perception, sont aussi les plus redondants. Ils correspondent à une description de la parole à une échelle la plus grossière, qui est celle de *l'enveloppe spectrale*. Or ces paramètres sont aussi ceux qui sont les plus protégés par le codage canal du GSM (Classe 1a) et pour lesquels une procédure de substitution est mise en œuvre en cas d'erreur détectée. La portée de ces méthodes se voit ainsi réduite au cas où la détection d'erreur mise en œuvre aux décodeurs est défectueuse. Cette situation, qui était assez fréquente pour le GSM FR, est beaucoup plus rare pour le GSM EFR dont les mécanismes de détection d'erreur ont été améliorés (cf. Annexe B).

Autrement dit, les méthodes de détections basées sur un modèle de parole apparaissent séduisantes mais le décodeur GSM limite d'emblée leur utilité pour un post-traitement en ne transmettant pas les principales dégradations qu'elles seraient susceptibles de traiter.

La mise en œuvre de procédures de détection de défauts demeure intéressante pour traiter toutes les dégradations à un *niveau d'échelle plus fin* du signal de parole, et qui sont associées aux erreurs de transmission introduites sur les paramètres considérés moins sensibles par le codeur. Certaines de ces dégradations ont été recensées au Chapitre 1. Outre le problème lié à leur détection, la principale gageure est alors le masquage de ces dégradations. En effet, toute procédure de masquage par substitution de trame est à exclure puisqu'elle serait plus préjudiciable à la qualité et à l'intelligibilité de la parole que les dégradations que l'on cherche à masquer. Un schéma de type « Analyse – Modification – Synthèse » [Laroche, 1995] semble une approche pertinente dans ce cas puisqu'il permet des modifications non-linéaires du signal et que les dégradations à traiter sont elles-mêmes de type non-linéaire.

2.5 Conclusion

Dans l'optique d'un post-traitement de la parole transmise par le système GSM, nous avons étudié le problème de la détection d'artefacts sur le signal de parole décodé. Nous nous sommes focalisés sur les artefacts liés aux erreurs de transmission car ils sont les principaux facteurs de la dégradation de la qualité vocale. Nous avons ainsi développé un algorithme de détection de la « voix de robot » qui est liée à une périodisation du signal de parole engendrée par la procédure de substitution de trame du GSM Full Rate. Ces artefacts introduits par les mécanismes de protection aux erreurs du réseau lui-même sont les plus faciles à détecter car aisément caractérisables. Cependant, la détection de défauts caractérisés souffre d'un manque évident de généralité.

A l'inverse, les méthodes de détection de dégradations basées sur un modèle *a priori* des paramètres de la parole apparaissent très séduisantes. Un de leurs attraits est qu'en modélisant « l'attente » (le signal de parole) plutôt que les évènements inattendus (artefacts), elles se rapprochent à bas niveau du mécanisme de la perception humaine d'une dégradation. Ceci doit permettre de détecter une large diversité de défauts. L'autre intérêt de cette approche est qu'elle fournit naturellement un modèle pour le masquage des artefacts rencontrés. Cependant, la stratégie de masquage déjà mise en œuvre en amont par le décodeur GSM EFR réduit le champ d'application de ces méthodes de détection en aval du décodeur parole.

Il s'avère ainsi que le post-traitement de la parole transmise par le réseau GSM est surtout intéressant pour les dégradations à une échelle fine du signal. Cependant, l'impact de ces dégradations sur la qualité de parole est nettement moindre que celui lié aux pertes de trames. C'est pourquoi, il nous est apparu plus intéressant d'essayer d'appliquer, au niveau du décodeur parole lui-même, les méthodes de détection et masquage de dégradations basées sur un modèle *a priori* des paramètres de la parole. C'est la direction suivie dans la suite de ce document.

Chapitre 3

Décodage source à entrées souples : Introduction et état de l'art

3.1 Introduction

Les applications, telles les communications radio-mobiles, pour lesquelles le risque d'erreurs résiduelles est non-négligeable, ont conduit à ajouter une fonctionnalité de *masquage d'erreur* au niveau du décodeur parole. Cette approche revient déjà à exploiter un modèle de prédiction des paramètres de la parole. Les techniques de masquage existantes sont cependant souvent empiriques et le modèle *a priori* qu'elles utilisent reste implicite et assez rudimentaire. Parallèlement, des techniques de détection d'erreur sur les paramètres de parole, dont certaines ont été présentées au chapitre précedent, ont été mises en œuvre au niveau du décodeur parole. Une tendance s'affirme donc pour faire converger la détection et le masquage d'erreur à partir d'une connaissance *a priori* sur la parole au niveau du décodeur.

Le décodage de parole à *entrées souples* généralise et formalise cette démarche. Il représente une nouvelle conception du décodage, qui ne se limite plus à une simple lecture dans une table de quantification mais qui devient un *estimateur optimal* du paramètre transmis. Il constitue l'objet d'étude principal de ce chapitre. Les travaux sont envisagés ici dans un cadre théorique général et ne se limitent pas au contexte particulier du système GSM. Nous présentons, dans un premier temps, des procédures de masquages améliorées avant de développer plus particulièrement les techniques de *décodage parole à entrées souples*.

3.2 Améliorations de la procédure de masquage du décodeur

Les procédures de masquage au niveau du décodeur parole reposent sur deux éléments :

- Une information sur la *validité* des données reçues en entrée du décodeur de parole.
- Un mécanisme de *substitution* des paramètres pour lesquels les données reçues ont été déclarées « invalides ».

Dans le cas du GSM, l'information de validité est apportée par l'indicateur BFI (*Bad Frame Indicator*) calculé au niveau du décodeur canal et transmis en entrée du décodeur parole. Plus précisément, le BFI indique la présence d'une erreur résiduelle parmi les bits considérés comme les plus sensibles vis-à-vis de la qualité de la parole décodée. Cette détection est basée sur un test CRC (*Cyclic Redundancy Check*) décrit en Annexe B. Lorsque une erreur est détectée parmi ces bits, c'est *l'intégralité* de la trame[27] reçue en entrée du décodeur parole qui est alors *substituée*. La technique de substitution mise en œuvre par le décodeur parole EFR est décrite en détail dans l'Annexe A.

Les recherches en vue d'améliorer la procédure de masquage portent sur ces deux éléments, avec pour but de rendre d'une part, l'information de validité exploitée plus *sélective* et plus *robuste*, et d'autre part, la parole engendrée par le mécanisme de substitution plus *naturelle*.

3.2.1 Améliorations de la substitution de trame

Les premières procédures de substitution étaient très rudimentaires, comme celle du GSM Full Rate présentée en Annexe A. Une analyse des artefacts très audibles (« voix de robot ») qu'elle engendrait a été présentée au Chapitre 2. Cependant, la procédure de substitution du GSM EFR introduit également des artefacts de type « voix métallique ». Cet artefact, qui se traduit par un excès d'harmoniques dans le spectre du signal, est engendré par la répétition à l'identique de la valeur du paramètre « délai LTP » (pitch) lors de la substitution de trame. Pour éviter cela, on peut légèrement modifier la valeur du pitch répétée à chaque nouvelle trame substituée. Ainsi, la procédure de masquage du G.729 [Salami et al., 1996] incrémente le délai LTP d'une unité et [De Martin et al., 2000] introduit une *gigue de pitch*. On peut également introduire des procédures d'extrapolation différentes (notamment en ce qui concerne le délai LTP) selon que la dernière trame valide était *voisée ou non*.

---

[27] On rappelle que le codeur de parole fonctionne par *trames*, c'est-à-dire que les paramètres du codeur sont estimés périodiquement, sur des segments de parole de *durée fixe* (cf. Annexe A). On parle ainsi de « trame de parole » (segment sur lequel les paramètres sont calculés), de « trame de paramètres » (paramètres calculés), et de « trame de bits » (bits codant les index de quantification des paramètres).

D'autres modifications visent à améliorer les *transitions* entre les trames extrapolées et les premières trames valides qui leur succèdent. En effet, beaucoup des paramètres de l'EFR sont quantifiés avec un effet mémoire, et de ce fait, les dernières trames substituées influencent les premières trames valides suivantes. Ceci peut entraîner des artefacts notamment par la propagation aux nouvelles trames de l'atténuation appliquée aux gains des dictionnaires fixes et adaptatifs. Afin d'obtenir une transition douce, il peut être alors nécessaire de rajouter un contrôle de gain sur le signal reconstruit [De Martin et al., 2000].

Enfin les *modèles a priori* utilisés pour l'extrapolation des paramètres sont assez rudimentaires dans le cas des codeurs GSM FR et EFR. Nous avons déjà passé en revue des modèles plus élaborés au chapitre précédent. Un modèle plus complexe mais moins empirique puisque obtenu par apprentissage est la prédiction des LSF à partir d'une loi *a priori* jointe modélisée par *mélange de gaussiennes* (GMM) comme proposé par [Lindblom et al., 2000]. Une telle méthode permet aussi de tenir compte de la corrélation entre paramètres de diverse nature comme les LSF et le gain LTP, il suffit pour cela d'étendre la loi jointe aux nouveaux paramètres considérés. On peut également modéliser l'évolution des valeurs d'un paramètre au cours du temps à l'aide d'une chaîne de Markov. Une chaîne de Markov du 1er ordre ne permet d'extrapoler que sur un horizon d'une trame, cependant on peut étendre cet horizon en utilisant une chaîne de Markov d'ordre plus élevé. Ainsi, [Kohler et al., 2000] propose une procédure de substitution basée sur l'approximation d'une telle chaîne de Markov.

3.2.2 Masquage par paramètre

Les procédures de masquage actuelles, dont celles du GSM, affectent l'intégralité de la trame du codeur. Autrement dit, si une erreur est détectée dans une trame, toute la trame de paramètres est déclarée perdue et se voit substituée. Des recherches récentes visent à améliorer la *sélectivité* de la procédure de masquage en considérant les paramètres individuellement et non plus la trame entière. On peut ainsi conserver les paramètres non-corrompus et exploiter la *corrélation intra-trame* entre paramètres pour extrapoler les paramètres erronés de la même trame. Ceci suppose néanmoins une détection d'erreur *par paramètres*, ce que ne permet pas le codage détecteur d'erreur (CRC) utilisé par le GSM (cf. Annexe B).

Pour éviter d'avoir à modifier le codeur canal, certains auteurs ont mis en œuvre, au niveau du décodeur parole, des techniques de *détection d'erreur sans redondance ajoutée* [Görtz, 1997]. Ces techniques exploitent uniquement la redondance des paramètres du codeur parole, elles ont été en partie présentées au chapitre précédent puisqu'elles permettent une détection en aveugle des dégradations. On citera notamment [Atungsiri et al., 1990] qui affine la localisation d'une erreur détectée globalement sur les LSP (par vérification de leur relation d'ordre) à l'aide de statistiques *a priori* sur les LSP individuelles. De la même façon, [Görtz, 1997] exploite la corrélation temporelle des LSF considérées individuellement, ainsi que celles du gain LTP et du délai LTP, pour détecter une erreur sur chacun de ces paramètres. Cependant, ces techniques de détection s'avèrent insuffisantes, leur taux élevé de fausses alarmes pouvant dégrader la qualité de la parole en l'absence d'erreur de

transmission. La détection d'erreur basée sur la *redondance résiduelle* des paramètres du codeur parole doit donc être combinée avec une détection d'erreur basée sur la *redondance ajoutée* par le codeur canal. Ainsi, [Görtz, 1998] modifie le codeur canal en rajoutant des CRC individuels pour les paramètres sensibles comme les LSP, le gain et le délai LTP ou le gain d'excitation.

Un autre procédé permettant un masquage individuel des paramètres d'une même trame de parole est de les étaler sur plusieurs trames lors de la transmission, une erreur dans une trame transmise étant détectée par CRC. [Martin et al., 2001] applique ce principe à un codeur CELP dans le contexte de la voix sur IP. Il peut ainsi masquer individuellement les jeux de résidus LSF du codeur en exploitant leur corrélation mutuelle pour l'extrapolation. Cette corrélation est exprimée par la loi conditionnelle d'un jeu de résidus LSF sachant les valeurs prises par les autres jeux de résidus LSF. Cette loi conditionnelle est elle même calculée à partir de la loi jointe des résidus LSF, modélisée par un mélange de gaussiennes (GMM). Si cette méthode ne peut être transposée au GSM sans modification de la norme, elle est intéressante pour le modèle *a priori* utilisé afin d'exploiter la corrélation entre jeux de résidus LSF.

3.2.3 Amélioration de la détection d'erreurs résiduelles

Le mécanisme de détection d'erreur (CRC) utilisé par le GSM pour pré-positionner ou non l'indicateur BFI est présenté en Annexe B. Ce mécanisme a un pouvoir de détection limité, autrement dit, il peut exister des erreurs résiduelles *non-détectées* par le BFI. On présente ici des techniques permettant d'améliorer cette détection d'erreur *sans avoir à modifier le codeur canal.*

Une première voie possible est d'adjoindre à la détection d'erreur issue du décodeur canal, une détection exploitant la *redondance résiduelle* des paramètres du codeur parole. Une telle procédure est mise en œuvre sur les paramètres reçus au niveau du décodeur parole. Pour la plupart des procédures développées, la détection d'erreur est basée uniquement sur la corrélation temporelle des paramètres du codeur, considérés individuellement. Ceci permet le masquage par paramètre présenté au précédent paragraphe [Görtz, 1997]. En revanche la corrélation entre les paramètres d'une même trame n'est pas exploitée par de telles procédures de détection.

Si l'on se restreint à une détection d'erreur par trame, il est alors possible d'utiliser la redondance entre les paramètres afin de détecter des configurations inadmissibles. En présence d'une telle configuration, la trame entière est rejetée. Ce principe a été mis en œuvre pour le GSM FR en dérivant une estimée de l'énergie du signal de parole calculée à partir des coefficients LARs, du gain LTP et du gain d'excitation. Une statistique sur les variations admissibles de l'énergie est alors utilisée pour détecter une erreur conjointement sur ces paramètres [Hindelang et al., 1997].

Une autre voie d'amélioration de la détection d'erreur consiste à exploiter la métrique de Viterbi calculée au niveau du décodeur canal. Ainsi, l'idée développée par [Sereno, 1991] est de comparer cette métrique de Viterbi à un seuil variable dépendant du rapport C/I estimé. Une trame est alors considérée comme corrompue si la métrique de Viterbi est inférieure à ce seuil ou si une erreur est

détectée par le CRC (détection BFI classique). Une telle procédure pourrait être mise en oeuvre au niveau du décodeur parole à condition de connaître la métrique de Viterbi à ce niveau. Cependant, les résultats obtenus montrent ici aussi une augmentation non négligeable du taux de fausses alarmes, ce qui peut nuire à la qualité de parole en condition de transmission non-bruitée.

3.2.4 Convergence vers un masquage souple

En conclusion, les approches développées pour l'amélioration du masquage au décodeur concernent trois points principaux :

- Le modèle *a priori* des paramètres du codeur.
- La sélectivité du masquage (masquage par paramètres).
- Une détection d'erreur exploitant conjointement la redondance ajoutée par le codeur canal et la redondance résiduelle des paramètres du codeur parole.

Enfin on peut voir dans les travaux de [Sereno, 1991], une tentative d'exploiter une information souple issue du canal afin d'améliorer la détection du BFI.

Toutes ces approches convergent donc vers une procédure qui exploite la redondance résiduelle des paramètres du codeur comme connaissance *a priori* afin de rendre le masquage plus sélectif parce que individualisé à chacun des paramètres, et plus souple parce que l'information binaire renvoyée par le CRC n'est plus la seule information de confiance utilisée pour la détection d'erreur. On peut voir le *décodage à entrées souples* [Fingscheidt et al., 2001] comme l'aboutissement de ces approches. C'est vers ce concept que nous avons orienté nos travaux, aussi nous le présentons en détail dans ce qui suit.

3.3 Décodage source à entrées souples

3.3.1 Principe

Avant d'introduire le concept de décodage à entrées souples, nous rappelons en premier lieu le schéma d'une transmission avec décodage source classique telle que celle du GSM. On s'intéresse ici à la transmission d'un paramètre ou d'un vecteur $\mathbf{v}$ de *paramètres* du codeur parole, comme par exemple les résidus LSF du GSM EFR. Le schéma de transmission est représenté de manière très synthétique par la Figure 3.1 sur laquelle on a surtout fait apparaître l'utilisation des paramètres reçus par le décodeur source.

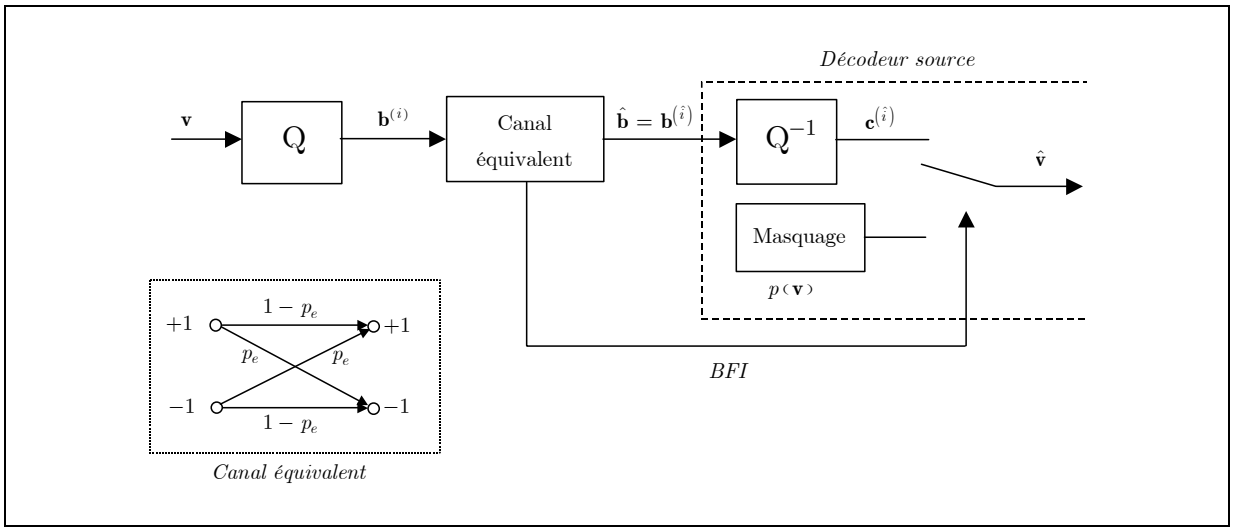

Figure 3.1 : Transmission avec décodage source conventionnel

Le quantificateur Q associe au paramètre $\mathbf{v}$, un élément $\mathbf{c}^{(i)}$ du dictionnaire de quantification $\mathbf{C} = \left\{\mathbf{c}^{(0)},...,\mathbf{c}^{(N-1)}\right\}$ de taille $N = 2^M$. Puisque l'on connaît le dictionnaire de quantification au décodeur, on transmet simplement l'index i avec $i \in \left\{0,1,...,2^M - 1\right\}$. Plus précisément, on notera $\mathbf{b}^{(i)} = \left[b_0^{(i)},...,b_m^{(i)},...,b_{M-1}^{(i)}\right]$, la combinaison de bits codant[28] l'index de quantification i. Cette combinaison de bits $\mathbf{b}^{(i)}$ est parfois appelée « mot de code source ».

Les bits $\mathbf{b}^{(i)}$ sont transmis au travers d'un *canal équivalent* qui englobe le codeur canal, l'émetteur, le canal de transmission radiomobile ainsi que le démodulateur et le décodeur canal (cf. Chapitre 1). Ce canal équivalent peut être considéré comme un *canal sans mémoire*[29]*, binaire symétrique* mais dont la probabilité d'erreur $p_e(m)$ associée à chaque bit $\hat{b}_m$ reçu est inconnue. La seule information sur l'état du canal est celle apportée par l'indicateur BFI de trame perdue. Lorsque cet indicateur n'est pas positionné (trame valide), l'opération effectuée par le décodeur source se réduit à la recherche dans la table de quantification de l'élément $\mathbf{c}^{(\hat{i})}$ associé à l'index $\hat{i}$ codé par la combinaison de bits reçus $\hat{\mathbf{b}} = \left[\hat{b}_0,...,b_m,...,\hat{b}_{M-1}\right]$. A l'inverse, lorsque l'indicateur BFI est positionné (trame invalide), le paramètre $\hat{\mathbf{v}}$ est calculé par une procédure de masquage d'erreur qui s'appuie[30] sur un modèle *a priori* $p(\mathbf{v})$ du paramètre.

L'indicateur BFI est un indice de confiance *binaire*, c'est-à-dire de type « tout ou rien » puisqu'il résulte d'une *détection* d'erreur. Comme tout résultat d'un processus de détection, le BFI peut donc générer des fausses alarmes et des non-détections. De plus, cet indice de confiance n'est pas *individuel*

---

[28] On considèrera ici un codage de l'index i selon le *code binaire naturel*. On notera que l'attribution d'une valeur d'index i à chaque centroïde peut être le résultat d'une procédure d'optimisation (« *Index Assignement* ») dans le but de minimiser dans le domaine des centroïdes, l'impact d'une erreur sur les bits $\mathbf{b}^{(i)}$ [Hedelin et al., 1995].

[29] La sortie du canal équivalent est considérée sans mémoire car celui-ci inclut un égaliseur (la démodulation et l'égalisation sont conjointes dans le cas du système GSM).

[30] Comme on l'a mentionné précédemment, le modèle *a priori* correspond ici au modèle mis en oeuvre pour *l'extrapolation* des paramètres de la trame effacée à partir de ceux de la dernière trame valide.

à chaque bit reçu $\hat{b}_m$ (peu sélectif) puisqu'il est calculé globalement sur une *trame* du codeur parole. Ainsi, une trame peut être déclarée perdue alors qu'un faible nombre de bits a été corrompu, et inversement, peut être considérée valide alors que certains bits sont incorrects. Cependant, le décodeur source fait entièrement confiance aux données issues du canal lorsque le BFI n'indique pas d'erreur alors qu'en cas d'erreur détectée, il utilise exclusivement la redondance de paramètre $\mathbf{v}$ au travers du modèle *a priori* $p(\mathbf{v})$.

L'idée à la base du décodage à entrées souples est d'exploiter une estimée de la probabilité d'erreur $p_e(m)$ lors du décodage source. La connaissance de cet indice de confiance *souple* (non binaire) et *instantané* (propre à chaque bit reçu $\hat{b}_m$) permet d'utiliser *conjointement* les données issues du canal et la connaissance *a priori* sur la source $\mathbf{v}$ pour l'estimation optimale du paramètre transmis. Le principe du décodage à entrées souples est illustré Figure 3.2, le décodeur parole reçoit désormais, en sortie du canal équivalent, la combinaison de bits $\hat{\mathbf{b}} = \left[\hat{b}_0,...,\hat{b}_{M-1}\right]$ codant l'index de quantification[31] j, ainsi que les estimées des probabilités d'erreur associées $\mathbf{p}_e = [\,p_e(0),...p_e(M-1)]$. Nous précisons le calcul de ces probabilités d'erreur dans le paragraphe qui suit.

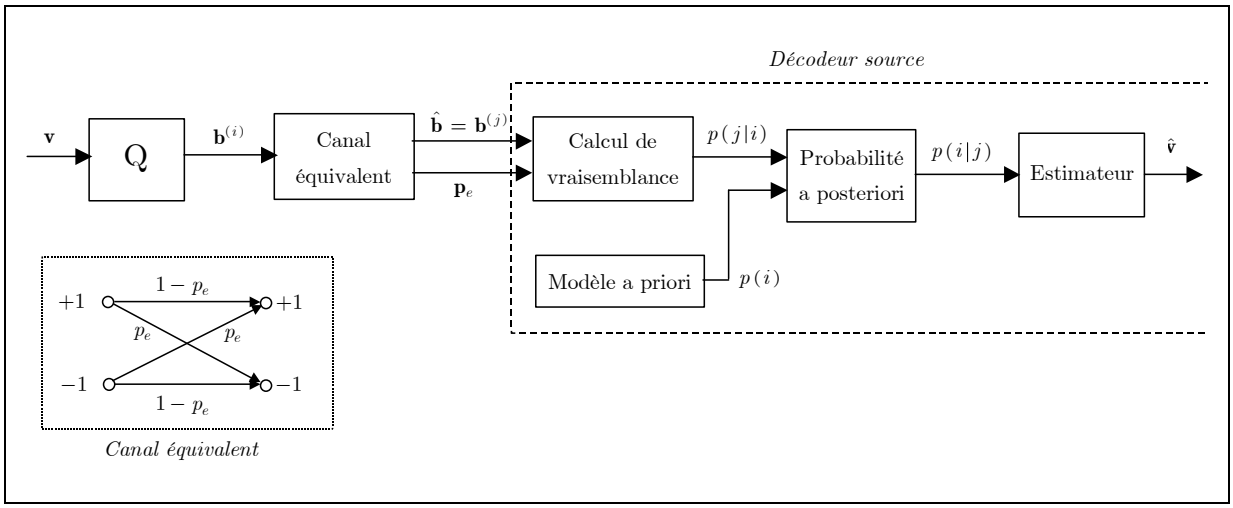

Figure 3.2 : Principe du décodage à entrées souples

3.3.1.1 Canal à sorties souples

Comme indiqué Figure 3.2, les probabilités d'erreur $p_e(m)$ correspondent aux *probabilités de transition du canal binaire équivalent*, ce sont des probabilités *instantanées* variant en fonction de l'instant[32] m. Lorsque le canal équivalent inclut un codeur/décodeur canal, ce qui est le cas du GSM, on peut les obtenir en utilisant un *décodeur canal à sorties souples* (cf. Annexe D) dont la sortie

[31] Nous notons j la valeur de l'index codé par la combinaison de bits *reçus* et non plus $\hat{i}$ comme pour le décodeur classique. Ceci afin d'indiquer qu'aucune décision *ferme* sur la valeur de l'index transmis n'a été prise à ce niveau.

[32] L'instant m réfère ici à la position du bit considéré au sein du mot de code source transmis $\mathbf{b}$ (resp. reçu $\hat{\mathbf{b}}$)

$L(b_m)$ à l'instant m s'interprète comme le logarithme du rapport des probabilités *a posteriori* des valeurs du bit b_m :

$$L(b_m) = \log \frac{p(b_m = +1|\mathbf{Y})}{p(b_m = -1|\mathbf{Y})} \tag{3.1}$$

où $\mathbf{Y}$ est la séquence de symboles reçus en entrée du décodeur canal. On notera qu'en règle générale[33], le rapport des probabilités *a posteriori* dans (3.1) se réduit à un rapport de *vraisemblances* puisque le décodeur canal ne fait pas d'hypothèses sur la valeur *a priori* du bit b_m.

La *décision* $\hat{b}_m$, en sortie du décodeur canal, sur la valeur du bit transmis à l'instant m est directement fournie par le signe de la *valeur souple* $L(b_m)$:

$$\hat{b}_m = \text{sign}\,(L(m)) \tag{3.2}$$

Puisque l'on considère des éléments binaires à valeur dans $\{+1,-1\}$, on peut ré-écrire la valeur souple $L(b_m)$ en faisant apparaître la probabilité d'erreur $p_e(m)$ associée à la décision $\hat{b}_m$:

$$\begin{aligned} L(b_m) &= \log \frac{p(b_m = +1|\mathbf{Y})}{p(b_m = -1|\mathbf{Y})} \\ &= \hat{b}_m \log \frac{p\left(\hat{b}_m|\mathbf{Y}\right)}{1 - p\left(\hat{b}_m|\mathbf{Y}\right)} = \hat{b}_m \log \frac{1 - p_e(m)}{p_e(m)} \end{aligned} \tag{3.3}$$

On en déduit ainsi l'expression de la probabilité d'erreur en fonction de la valeur souple $L(b_m)$:

$$p_e(m) = \frac{1}{1 + \exp|L(m)|} \tag{3.4}$$

3.3.1.2 Vraisemblance de l'index de quantification transmis

La première étape du décodage consiste à formuler les *probabilités de transition* $p(j|i)$ entre une valeur quelconque d'index de quantification i transmise et la valeur d'index j associée à la combinaison de bits $\hat{\mathbf{b}} = \left[\hat{b}_0,...,\hat{b}_{M-1}\right]$ reçus en sortie du *canal binaire équivalent*.

Les probabilités de transition du *canal binaire équivalent* s'expriment pour chaque bit à l'instant m :

$$p\left(\hat{b}_m \middle| b_m^{(i)}\right) = \begin{cases} 1 - p_e(m) \text{ si } \hat{b}_m = b_m^{(i)} \\ p_e(m) \text{ si } \hat{b}_m \neq b_m^{(i)} \end{cases} \tag{3.5}$$

[33] Le cas contraire est le décodage canal contrôlé par la source, présenté au Chapitre 6, qui a justement pour objet d'exploiter un *a priori* sur les bits b_m.

En raison de l'entrelacement des bits (cf. Annexe C) utilisé dans la chaîne de transmission GSM, on peut considérer[34] que les probabilités d'erreur $p_e(m)$ sont indépendantes entre-elles. On peut alors former la probabilité de transition entre un index i transmis et l'index j codé par les bits reçus :

$$p(j|i) = \prod_{m=0}^{M-1} p\left(\hat{b}_m \middle| b_m^{(i)}\right) \tag{3.6}$$

La probabilité de transition $p(j|i)$ doit être évaluée sur l'ensemble des valeurs $i \in \{0,1,...,2^M - 1\}$ afin d'obtenir la *distribution de vraisemblance* de l'index i transmis, étant donnés la valeur d'index j codée par les bits reçus $\hat{\mathbf{b}}$, et les probabilités d'erreur associées $\mathbf{p}_e$.

3.3.1.3 Probabilité a posteriori de l'index de quantification

Dans une seconde étape, la distribution de vraisemblance $p(j|i)$ de l'index de quantification transmis i est combinée avec une distribution *a priori* de l'index i. Ceci permet le calcul des *probabilités a posteriori* $p(i|j)$ qui expriment la probabilité d'avoir transmis le paramètre quantifié d'index i sachant l'index reçu j et la connaissance *a priori* au niveau du décodeur :

$$\begin{aligned} p(i|j) &= \frac{p(i,j)}{p(j)} \\ &= C\ p(j|i)\,p(i) \end{aligned} \tag{3.7}$$

où C est une constante de normalisation (puisque indépendante de i sachant l'index reçu j).

3.3.1.4 Estimation du paramètre transmis

Enfin, la dernière étape est celle de *l'estimation du paramètre* transmis. C'est uniquement à ce niveau final qu'est prise la décision sur la valeur du paramètre, à la différence du décodage classique pour lequel cette valeur est décidée en sortie du canal équivalent (au niveau décodeur canal). En effet, les probabilités *a posteriori* $p(i|j)$ permettent d'estimer la valeur optimale $\hat{\mathbf{v}}$ du paramètre en fonction d'une mesure d'erreur choisie. Les critères d'optimalité considérés sont le *Maximum a Posteriori* (MAP) et le *Minimum d'Erreur Quadratique Moyenne* (MMSE en anglais) :

- Dans le cas d'une **estimation MAP**, la valeur estimée $\hat{\mathbf{v}}$ est l'élément du dictionnaire de quantification (centroïde) tel que :

$$\hat{\mathbf{v}} = \mathrm{Q}^{-1}\left(\hat{i}\right) = \mathbf{c}^{(\hat{i})} \text{ avec } \hat{i} = \arg\max_i\left(p(i|j)\right) \tag{3.8}$$

---

[34] Pour que la relation (3.6) soit pleinement valide, il faudrait un entrelacement des bits entre le codeur parole et le codeur canal alors que, dans le système GSM, l'entrelacement intervient entre le codeur canal et la modulation (émetteur). Cependant, le multiplexage et la redistribution des bits au sein de la trame codée avant le codage canal (cf. Annexe B) joue ici approximativement le même rôle qu'un entrelacement.

- Dans le cas d'une **estimation MMSE**, la valeur $\hat{\mathbf{v}}$ correspond à la moyenne des centroïdes $\mathbf{c}^{(i)}$ pondérés par la distribution *a posteriori* $p(i|j)$ des index i :

$$\hat{\mathbf{v}} = E(\mathbf{v}|j) \simeq \sum_i E(\mathbf{v}|i)\,p(i|j) = \sum_i \mathbf{c}^{(i)} p(i|j) \tag{3.9}$$

On notera que le paramètre estimé au sens du MMSE n'appartient plus forcément au dictionnaire de quantification, autrement dit le *décodage est à entrées souples et sorties souples*, on parle alors plus simplement de *décodage souple* [Skoglund, 1999].

La Figure 3.3 illustre de manière intuitive le mécanisme du décodage à entrée souple, selon le type d'estimateur utilisé. On y a représenté la distribution de probabilité *a priori* $p(i)$ d'un index de quantification i en regard de sa distribution de vraisemblance $p(j|i)$ en sortie du canal équivalent.

Lorsque les probabilités d'erreur $\mathbf{p_e}$ associées aux bits $\hat{\mathbf{b}} = \mathbf{b}^{(j)}$ reçus en sortie du canal équivalent augmentent, la vraisemblance $p(j|i)$ de l'index transmis i tend vers la distribution uniforme. La contribution de l'information apportée par la distribution *a priori* $p(i)$ devient alors prépondérante dans l'estimation (MAP ou MMSE) du paramètre $\hat{\mathbf{v}}$. A l'inverse, dans le cas où il n'y a pas d'erreur de canal alors les estimées MAP et MMSE sont confondues et coïncident avec la sortie $\hat{\mathbf{v}} = \mathbf{c}^{(j)}$ du décodeur source conventionnel.

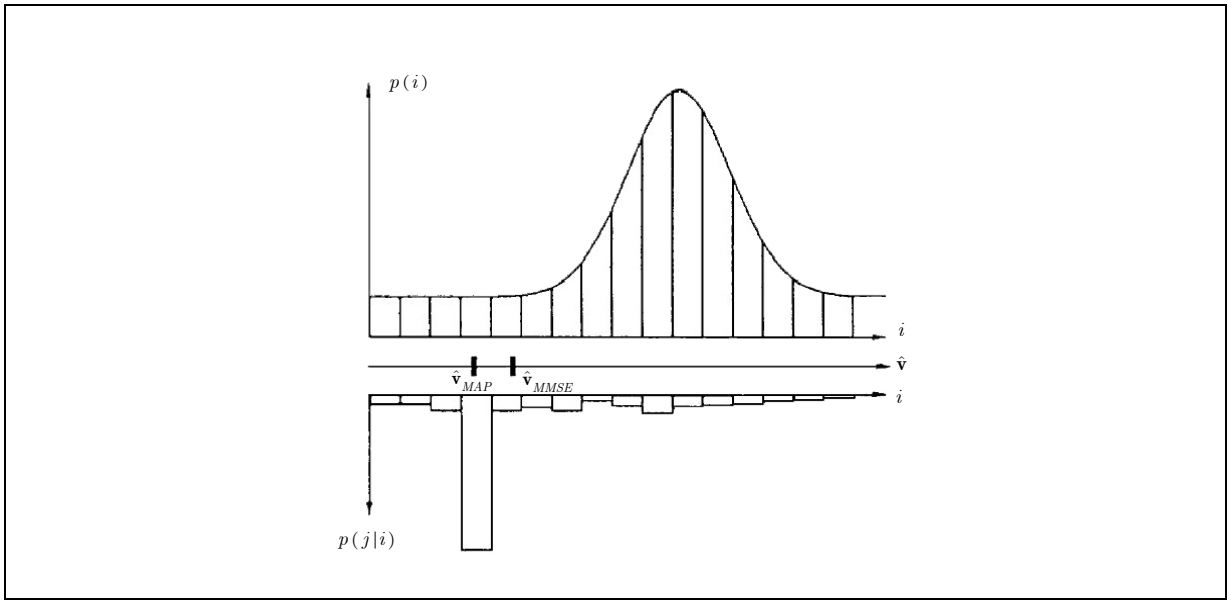

Figure 3.3 : Estimation de paramètre à partir de la probabilité *a priori* et de la vraisemblance

3.3.2 Structure de la probabilité *a posteriori*

On considère désormais la suite i_n des index transmis, où on a fait apparaître l'indice temporel n qui peut être un indice de *trame* ou de *sous-trame* selon la périodicité avec laquelle est le paramètre $\mathbf{v}$ est

quantifié. Dans la présentation de principe qui précède, nous n'avons pas précisé la forme exacte de la loi *a priori* $p(i_n)$, cette loi peut être invariante et déterminée par avance, ou au contraire dépendre des valeurs reçues aux instants précédents. Le calcul de cette loi, et par conséquent celui de la probabilité *a posteriori* $p(i_n|j_n)$, dépend du modèle *a priori* utilisé pour représenter la *redondance résiduelle* des index de quantification i_n. Nous présentons dans ce qui suit les différents modèles adoptés dans la littérature ainsi que le calcul des probabilités *a posteriori* correspondantes.

3.3.2.1 Décodage souple sans *a priori*

Considérons le décodeur présenté Figure 3.2 dans le cas où l'on ne dispose d'aucune connaissance *a priori* sur le paramètre transmis. Ceci revient à supposer par défaut une distribution *a priori* $p(i_n)$ uniforme et la probabilité *a posteriori* (3.7) se réduit à la vraisemblance fournie par le canal :

$$p(i_n|j_n) = C\ p(j_n|i_n) \tag{3.10}$$

où C est une constante de normalisation (relativement à l'index i_n recherché).

Si le critère d'estimation utilisé est le MAP selon l'équation (1.3), alors la sortie du décodeur est identique à celle du décodeur conventionnel illustré Figure 3.1 puisque le critère MAP se réduit au *Maximum de Vraisemblance* (MV). En revanche, dans le cas du critère MMSE, la sortie de l'estimateur (3.9) demeure une sortie souple qui correspond à la moyenne des éléments du dictionnaire de quantification pondérés par leur vraisemblance :

$$\hat{\mathbf{v}}_n = E(\mathbf{v}_n|j_n) \simeq \sum_i \mathbf{c}^{(i)} p(j_n|i_n = i) \tag{3.11}$$

3.3.2.2 Exploitation de la non-uniformité (AK0)

La distribution des index en sortie du quantificateur est rarement uniforme. En effet, les algorithmes de quantification de type Lloyd-Max minimisent un critère de distorsion moyenne (cf. Annexe A) plutôt que de chercher une distribution de probabilités $p(i_n)$ uniforme. Cette forme de redondance résiduelle est la plus simple à modéliser. Ainsi, [Fingscheidt et al., 1997] utilisent la probabilité *a priori* $p(i_n)$ correspondant à l'histogramme des paramètres quantifiés $\mathbf{c}^{(i)}$ appris sur une large base de données. La probabilité *a posteriori* s'obtient alors directement selon l'équation (3.7). L'algorithme résultant est dénommé **AK0** (*0th order a priori knowledge*) pour indiquer qu'il exploite une distribution *a priori* $p(i_n)$ invariante selon l'instant n.

3.3.2.3 Exploitation de la corrélation inter-trame (AK1)

Comme présenté en Annexe A, le codage parole est confronté à un certain nombre de limitations pratiques comme la complexité, le délai maximal admissible ou encore les risques de propagation d'erreur en cas d'erreur de transmission. De part ces limitations, les paramètres quantifiés (ou de

manière équivalente les index associés) présentent une redondance résiduelle. Une part importante de cette redondance est représentée par la corrélation temporelle entre les trames (ou sous-trames) successives de paramètres. Différents modèles ont été proposés pour modéliser cette corrélation dans le cadre du décodage source à entrées souples.

3.3.2.3.a Processus de Markov

La corrélation temporelle de la suite i_n des index de quantification transmis est entièrement représentée par la probabilité conditionnelle $p(i_n|i_1,...,i_{n-1})$. On modélise ici cette corrélation en se limitant à un *processus de Markov d'ordre 1* :

$$p(i_n|i_1,...,i_{n-1}) = p(i_n|i_{n-1}) \tag{3.12}$$

c'est-à-dire qu'on ne considère que la corrélation entre trames (ou sous-trames) adjacentes [Sayood et al., 1991]. Les *probabilités de transition a priori* $p(i_n|i_{n-1})$ entre index (ou paramètres quantifiés) sont apprises sur une base de données de parole et doivent être stockées au décodeur.

Considérons à nouveau la transmission de l'index de quantification i_n selon le schéma illustré Figure 3.2. La suite des indices j_n en sortie du canal discret sans mémoire et de probabilités de transition $p(j_n|i_n)$ peut être décrite par une *Chaîne de Markov Cachée d'ordre 1* [Miller et al., 1998]. Les états de cette chaîne correspondent aux différentes valeurs $i \in \{0,1,...,2^M-1\}$ de l'index de quantification i_n et l'observation associée à chaque état est décrite par la probabilité de transition $p(j_n|i_n)$ du canal discret.

Le calcul de la *probabilité a posteriori* $p(i_n|j_1,...,j_n)$ s'apparente alors à celui de la variable *forward* (ou *induction avant*) $\alpha_n(i)$ dans un treillis, définie par :

$$p(i_n = i|j_1,...,j_n) = C\ \alpha_n(i) \tag{3.13}$$

avec
$$\alpha_n(i) = p(j_1,...,j_n,i_n = i) \tag{3.14}$$

et C est une constante de normalisation.

Cette variable peut se calculer de manière itérative comme suit :

$$\begin{aligned} &\alpha_n(i) = p(j_n|i_n = i)\sum_{i'}\alpha_{n-1}(i')\ p(i_n = i|i_{n-1} = i') \\ &\alpha_0(i) = \pi_i \end{aligned} \tag{3.15}$$

où π_i désigne la probabilité *a priori* de l'état i à l'instant initial $n = 0$.

Cette récursion est à la base des modèles proposés par [Phamdo et al., 1994], [Fingscheidt et al., 1997], [Miller et al., 1998]. On désignera par **AK1** (*first-order a priori knowledge*), cet algorithme exploitant un *a priori* d'ordre 1 (plus exactement, la corrélation temporelle à l'ordre 1).

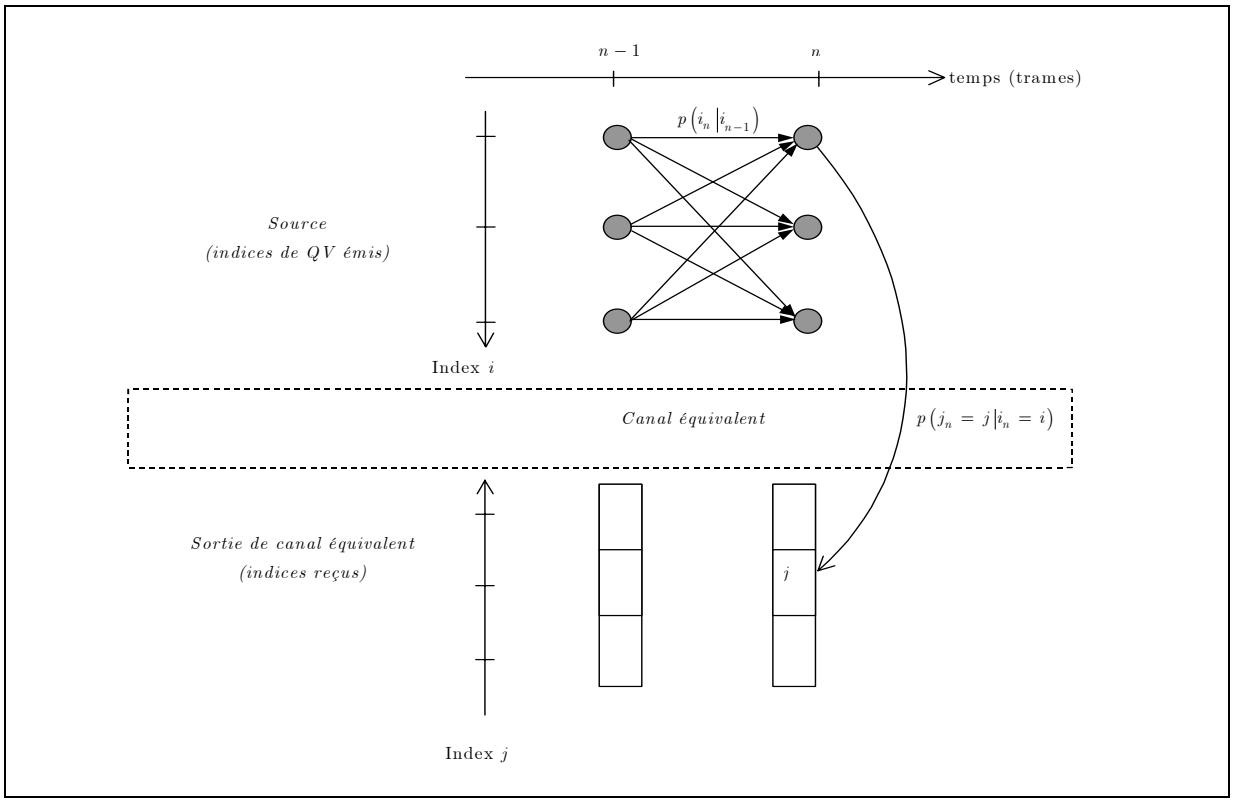

Figure 3.4 : Interprétation de la sortie du canal comme une Chaîne de Markov Cachée

La complexité de calcul de la probabilité *a posteriori* $p\left(i_n \middle| j_1,...,j_n\right)$ selon l'équation (3.15) est de l'ordre $O\left(N^2\right)$ où $N = 2^M$ est la taille du dictionnaire de quantification. Cette complexité apparaît rédhibitoire pour un codeur tel que l'EFR puisque la taille des dictionnaires de quantification peut être aussi élevée que $N = 512$ pour certains paramètres[35].

La complexité élevée de l'algorithme AK1 vient du fait qu'on ne fait *aucune hypothèse sur les valeurs précédentes* de l'index i_n dans le calcul de $p\left(i_n \middle| j_1,...,j_n\right)$. Des approches *sous-optimales* ont été introduites pour réduire la complexité. Elles reposent sur l'idée d'une prédiction de l'index i_n uniquement à partir des valeurs précédemment estimées. Plus précisément, on peut décomposer la probabilité $p\left(i_n \middle| j_1,...,j_n\right)$ de la façon suivante :

$$p\left(i_n \middle| j_1,...,j_n\right) = p\left(i_n \middle| \mathbf{j}_1^n\right) = C\ p\left(j_n \middle| i_n\right) \sum_{\mathbf{i}_1^{n-1}} p\left(i_n \middle| \mathbf{i}_1^{n-1}\right) p\left(\mathbf{i}_1^{n-1} \middle| \mathbf{j}_1^{n-1}\right) \tag{3.16}$$

[35] La table d'allocation des bits pour les paramètres de l'EFR est rappelée par le Tableau A.2 en Annexe A.

où $\mathbf{i}_1^{n-1} = [i_1,...,i_{n-1}]$ désigne la *séquence* d'index pour les instants précédents et C est une constante de normalisation (sachant l'index reçu j_n).

Suivant le critère d'optimalité (MAP ou MMSE) employé pour l'estimation du paramètre, on peut alors simplifier l'équation (3.16) de deux façons.

- **Critère MAP :**

  On cherche à maximiser l'expression (3.16), on peut alors utiliser l'approximation dite de la *séquence dominante*, c'est-à-dire qu'on réduit la sommation sur toutes les séquences $\mathbf{i}_1^{n-1}$ dans (3.16) à la séquence la plus probable. C'est la séquence $\hat{\mathbf{i}}_1^{n-1}$ *en sortie de l'estimateur MAP*. Une telle approximation est valable pour les faibles taux d'erreur mais entraîne une divergence dès que le taux d'erreur s'élève.

- **Critère MMSE :**

  Une approximation de la probabilité (3.16) peut être obtenue en remplaçant la probabilité marginale obtenue par sommation sur toutes les séquences $\mathbf{i}_1^{n-1}$ par une « *probabilité prédictive* » conditionnée aux valeurs du paramètre précédemment estimées selon le critère MMSE :

$$p(i_n|\mathbf{j}_1^n) \simeq p(j_n|i_n)\, p(i_n|E(\mathbf{v}|j_{n-1}),...,E(\mathbf{v}|j_1)) \tag{3.17}$$

  C'est cette approximation qui est utilisée dans les modèles basés sur la prédiction linéaire présentés dans ce qui suit.

3.3.2.3.b Prédiction Linéaire

On modélise ici la corrélation temporelle du *paramètre* $\mathbf{v}_n$ et non plus celle de *l'index de quantification* i_n. Ainsi, [Gerlach, 1993] et [Fingscheidt et al., 1997] décrivent $\mathbf{v}_n$ à partir d'un processus auto-régressif d'ordre r :

$$\mathbf{v}_n = \mathbf{A}.[\mathbf{v}_{n-1},...,\mathbf{v}_{n-r}]^T + \mathbf{w}_n \tag{3.18}$$

où $\mathbf{A}$ est la matrice des coefficients de prédiction linéaire et $\mathbf{w}$ le signal d'erreur. Les coefficients $\mathbf{A}$ sont fixes et doivent être appris à l'avance sur une base de données. Comme le paramètre $\mathbf{v}_n$ est rarement un processus auto-régressif gaussien, la distribution $p_{\mathbf{w}}(\mathbf{w})$ du signal d'erreur doit également être apprise puis stockée au décodeur.

Le décodeur utilise le modèle *a priori* (3.18) pour prédire une valeur du paramètre $\mathbf{v}_n$ d'après les précédentes valeurs estimées au sens du MMSE :

$$\mathbf{v}_n^{(PL)} = \mathbf{A}[\hat{\mathbf{v}}_{n-1}^{(MMSE)},...,\hat{\mathbf{v}}_{n-r}^{(MMSE)}]^T \tag{3.19}$$

On peut en déduire une probabilité *a priori* sur l'index de quantification i_n à l'instant n, selon :

$$p\left(i_n = i \middle| E\left(\mathbf{v}_{n-1} \middle| j_{n-1}\right),...,E\left(\mathbf{v}_{n-r} \middle| j_{n-r}\right)\right) = \int_{\mathbf{v}\in\Omega^{(i)}} p_{\mathbf{w}}\left(\mathbf{v}_n^{(PL)} - \mathbf{v}\right) d\mathbf{v} \tag{3.20}$$

où $\Omega^{(i)}$ désigne la cellule de quantification associée à l'index i.

La probabilité *a posteriori* $p\left(i_n \middle| \mathbf{j}_{n-p}^{n}\right)$ s'obtient alors à partir de l'équation (3.17).

On remarquera que l'emploi d'une prédiction linéaire fixe à partir des données précédemment estimées risque également d'entraîner une propagation d'erreurs[36].

3.3.2.3.c Chaîne de Markov Cachée

Un autre défaut inhérent à la description de l'index de quantification i_n par une chaîne de Markov est la très grande dimension prise par la table des probabilités de transition $p\left(i_n \middle| i_{n-1}\right)$ dès que la résolution du quantificateur augmente.

En fait, lorsque la résolution du quantificateur est élevée, ce qui est le cas du GSM, il est beaucoup plus pertinent de décrire i_n à l'aide d'une *chaîne de Markov Cachée*. Ce type de modélisation est d'ailleurs utilisé pour les paramètres spectraux en reconnaissance de la parole. En utilisant une chaîne de Markov Cachée, la relation entre l'index i_n et les états finis de la chaîne de Markov est beaucoup plus souple. En effet, un *état* q de la chaîne n'est plus lié de manière déterministe à une *valeur donnée* prise par l'index i_n mais définit une distribution $p(i_n|q_n = q)$ de l'index i_n à l'instant n. Ceci permet de réduire la dimension de la chaîne de Markov. Cependant, la difficulté dans le cas du décodeur souple est que les états q sont *doublement cachés* puisque l'index i_n est lui-même inobservable et que l'on ne dispose que de sa vraisemblance $p(j_n|i_n)$ d'après les index j_n reçus en sortie du canal. En fait, ce processus équivaut à une chaîne de Markov Cachée dont les lois d'observation associées aux états q sont données par :

$$p(j_n|q_n = q) = \sum_i p(j_n|i_n = i)\, p(i_n = i|q_n = q) \tag{3.21}$$

Ce calcul nécessite l'intégration sur l'ensemble du dictionnaire de quantification, ce qui demeure d'une complexité assez élevée. Une simplification est possible si l'on dispose d'une expression analytique de la loi $p(j_n|i_n)$ puisque l'équation (3.21) peut alors être résolue analytiquement en utilisant un modèle multi-gaussien pour la loi $p(i_n|q_n)$. C'est la démarche utilisée par [Ligdas et al., 1997].

A partir des lois d'observation calculées selon l'équation (3.21), on peut utiliser la *récursion avant* entre les états q du treillis pour obtenir la probabilité *a posteriori* des états $p\left(q_n \middle| j_n,...,j_1\right)$. La probabilité *a posteriori* d'avoir transmis l'index i à l'instant n s'obtient ensuite comme la somme des lois $p(i_n = i|q_n)$ pondérées par la probabilité des états $p\left(q_n \middle| j_n,...,j_1\right)$:

---

[36] On peut rapprocher ceci du fait qu'on emploie, au codeur parole, une prédiction MA des LSF et non un modèle AR, justement afin d'éviter la propagation d'erreur.

$$p\left(i_n = i \middle| j_n,...,j_1\right) = C \sum_q p\left(i_n = i | q_n = q\right) p\left(q_n = q \middle| j_n,...,j_1\right) \tag{3.22}$$

L'idée sous-jacente à cette approche, qui est de réduire la dimension de la chaîne de Markov en introduisant des « états » intermédiaires, rejoint l'approche que nous avons développée et que nous exposons au Chapitre 5.

3.3.2.4 Exploitation de la corrélation intra-trame (AK2)

Les méthodes présentées jusqu'ici n'exploitent que la *corrélation inter-trame* du paramètre $\mathbf{v}_n$ ou de l'index de quantification i_n. Par corrélation inter-trame, on entend ici corrélation entre les valeurs d'un même paramètre (ou d'un même index de quantification) pour des trames *successives*, l'indice n désignant la trame. Cependant, dans les schémas de codage tels que celui du GSM EFR, il existe également une corrélation entre les différents paramètres (resp. index de quantification) au sein d'une même trame. Cette *corrélation intra-trame* provient par exemple d'une QV sous-optimale (pour des raisons de complexité) comme la QV des LSF pour le GSM. La corrélation intra-trame peut aussi simplement correspondre à une corrélation entre des paramètres identiques de différentes sous-trames. Il s'agit alors d'une corrélation temporelle mais celle-ci n'est pas entièrement exploitée par les méthodes précédentes qui ne permettent pas *l'interpolation* entre sous-trames.

La corrélation intra-trame peut être modélisée par un processus de Markov, de manière similaire à la corrélation inter-trame. Ceci permet notamment de dériver une probabilité *a posteriori* tenant compte *simultanément* de la redondance *inter-* et *intra-trame* [Adrat et al., 2000], [Lahouti et al., 2001].

On notera $\mathbf{I}_n = \left(i_{n,1},...,i_{n,k},...,i_{n,L}\right)$, une *trame* de L index de quantification en sortie du codeur parole à l'instant n et où l'indice k note la position au sein de la trame. Soit $\mathbf{J}_n = \left(j_{n,1},...,j_{n,k},...,j_{n,L}\right)$, la trame d'index reçus en sortie du canal équivalent. Considérons le calcul de la *probabilité a posteriori* $p\left(i_{n,k} \middle| \mathbf{J}_1,...,\mathbf{J}_n\right)$ de l'index $i_{n,k}$ sachant les trames reçues à l'instant n et aux instants précédents.

Pour réduire la complexité du calcul, on modélise la corrélation inter-trame et la corrélation intra-trame de manière *indépendante*. Ceci signifie qu'on néglige la corrélation entre index *différents* de trames *successives*. On a alors :

$$p\left(i_{n,k} \middle| \mathbf{I}_{n-1},...,\mathbf{I}_1\right) = p\left(i_{n,k} \middle| i_{n-1,k},...,i_{1,k}\right) \tag{3.23}$$

D'autre part la corrélation inter-trame et les corrélations intra-trame sont modélisées chacune comme des processus de Markov indépendants d'ordre 1 :

- inter-trame : $$p\left(i_{n,k} \middle| \mathbf{I}_{n-1},...,\mathbf{I}_1\right) = p\left(i_{n,k} \middle| i_{n-1,k},...,i_{1,k}\right) = p\left(i_{n,k} \middle| i_{n-1,k}\right) \tag{3.24}$$

- intra-trame : $$p\left(i_{n,k} \middle| i_{n,k-1},...,i_{n,1}\right) = p\left(i_{n,k} \middle| i_{n,k-1}\right) \tag{3.25}$$

On s'est ainsi ramené à une structure en treillis sur les index $i_{n,k}$ et on peut calculer la *probabilité a posteriori* $p\left(i_{n,k}\left|\mathbf{J}_1,...,\mathbf{J}_n\right.\right)$ en introduisant les variables d'induction avant $\alpha_{n,k}(i)$ et d'induction « latérale » $\beta_{n,k}(i)$ et $\beta'_{n,k}(i)$, définies par :

$$\alpha_{n,k}(i) = p\left(j_{1,k},...,j_{n,k},i_{n,k}=i\right) \tag{3.26}$$

$$\beta_{n,k}(i) = p\left(j_{n,k+1},...,j_{n,L}\left|i_{n,k}=i\right.\right) \tag{3.27}$$

$$\beta'_{n,k}(i) = p\left(j_{n,1},...,j_{n,k-1}\left|i_{n,k}=i\right.\right) \tag{3.28}$$

La Figure 3.5 illustre l'exploitation de la corrélation inter-trame et intra-trame par les variables d'induction $\alpha_{n,k}(i)$, $\beta_{n,k}(i)$et $\beta'_{n,k}(i)$ pour le calcul de la probabilité *a posteriori*.

Les variables $\alpha_{n,k}(i)$, $\beta_{n,k}(i)$et $\beta'_{n,k}(i)$ s'obtiennent par les récursion avant et « latérale » :

$$\alpha_{n,k}(i) = p\left(j_{n,k}\left|i_{n,k}=i\right.\right)\sum_{i'}\alpha_{n-1,k}\left(i'\right)\,p\left(i_{n,k}=i\left|i_{n-1,k}=i'\right.\right) \tag{3.29}$$

$$\beta_{n,k}(i) = \sum_{i'}p\left(i_{n,k+1}=i'\left|i_{n,k}=i\right.\right)p\left(j_{n,k+1}\left|i_{n,k+1}=i'\right.\right)\beta_{n,k+1}\left(i'\right) \tag{3.30}$$

Enfin, on peut calculer la probabilité *a posteriori* selon :

$$p\left(i_{n,k}=i\left|\mathbf{J}_1,...,\mathbf{J}_n\right.\right) = p\left(i_{n,k}\left|j_{1,k},...,j_{n-1,k-1},\mathbf{J}_n\right.\right) = C\;\alpha_{n,k}(i)\beta_{n,k}(i)\beta'_{n,k}(i) \tag{3.31}$$

On restreint les récursions latérales à l'ordre 1 afin de limiter la complexité [Adrat et al., 2000], [Lahouti et al., 2001]. On dénommera cet algorithme **AK2** (*second-order a priori knowledge*) dans le reste de ce document.

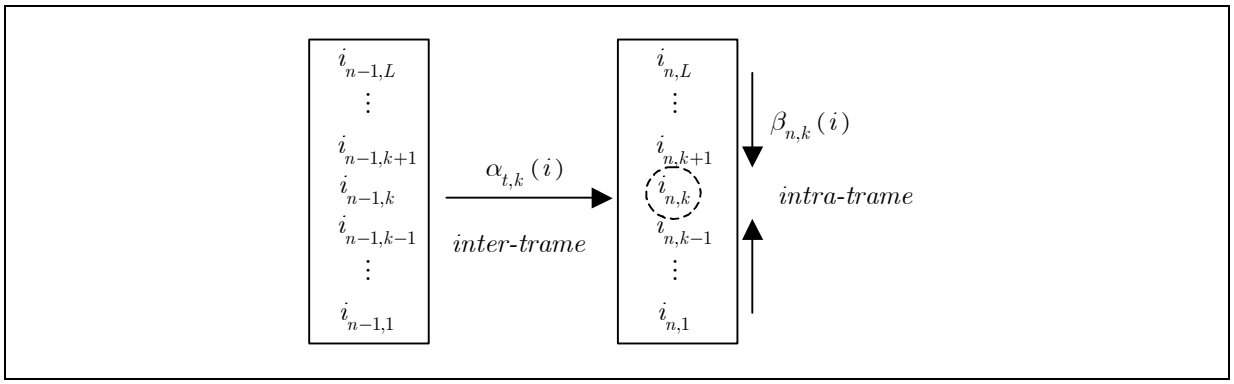

Figure 3.5 : Prise en compte de la redondance inter et intra-trame par l'algorithme forward-backward

3.4 Conclusion

Les travaux présentés ci-dessus ouvrent une nouvelle voie pour l'amélioration de la qualité de la parole reçue, intermédiaire entre le *masquage d'erreur* classiquement mis en œuvre au décodeur parole et la *correction d'erreur* habituellement cantonnée au décodeur canal. Cependant, la plupart de ces études se limitent à des cas simples comme la transmission de parole codée PCM [Fingscheidt et al., 1997] ou à des codeurs appliquant une quantification scalaire comme le GSM Full Rate [Fingscheidt et al., 1997]. L'efficacité de telles méthodes reste à démontrer pour des codeurs plus performants comme l'EFR, c'est-à-dire laissant moins de redondance résiduelle.

3.4.1.1 Le problème d'un modèle de prédiction fixe

Qu'il s'agisse d'une chaîne de Markov ou d'une prédiction linéaire de coefficients fixes, les modèles utilisés pour prendre en compte la corrélation résiduelle en sortie du codeur parole font implicitement l'hypothèse d'une source *stationnaire*. Ceci n'est évidemment pas le cas de la parole. On pourra remarquer que les techniques de masquage d'erreur classiques utilisent généralement elles-mêmes des modèles de prédiction fixe pour les paramètres spectraux[37]. Le modèle *a priori* utilisé dans les approches de décodage souple ne fait donc que reprendre le modèle implicite des procédures de masquage.

L'utilisation d'un modèle aussi simple pour la parole est acceptable dans une stratégie *masquage*, c'est-à-dire pour les bas niveaux de C/I, pour lesquels il n'est plus question de récupérer les paramètres transmis mais simplement de minorer l'impact subjectif des erreurs. La prédiction fixe des paramètres doit alors s'accompagner d'une *procédure d'affaiblissement* des gains d'excitation afin de ne pas prolonger trop artificiellement un segment de parole.

A l'inverse, lorsque la vraisemblance des paramètres reçus en sortie du canal est élevée, le modèle *a priori* a une contribution négligeable dans la probabilité *a posteriori* (3.7) calculée par le décodeur souple, et donc sur les paramètres estimés. Le décodeur souple est alors « transparent ».

Le problème se pose pour les niveaux de C/I intermédiaires, la question est alors de savoir si les modèles présentés plus haut sont efficaces dans une stratégie *correction d'erreur*. C'est-à-dire s'ils n'introduisent pas plus d'erreurs qu'ils n'en corrigent, du fait de la prédiction fixe utilisée.

On peut ici tenter un parallèle entre l'emploi d'un modèle de prédiction *des paramètres* au codeur (prédiction des LSF, du gain d'excitation) et la prédiction utilisée au niveau du décodeur *souple* :

- Au codeur, la redondance temporelle des paramètres est modélisée par une *prédiction fixe* mais le biais entre le comportement de ces paramètres pour la parole (non-stationnaire) et le modèle de

[37] Ainsi, la procédure de substitution des LSP donnée en exemple pour l'EFR correspond à un modèle de prédiction *auto-régressive* AR des LSP (cf. Annexe A).

prédiction fixe est pris en compte par le signal résiduel, qui est transmis au décodeur après quantification. Ainsi, l'EFR (cf. Annexe A) utilise une prédiction en *moyenne adaptée* MA d'ordre 1 pour les LSF et une prédiction MA d'ordre 4 pour le gain d'excitation (dans le domaine logarithmique). Il est préférable d'éviter l'emploi d'un modèle de prédiction trop « contraint » tel un modèle *auto-régressif* AR, qui, du fait de sa mémoire infinie, entraînerait une accumulation d'erreurs au décodeur en cas d'erreur de transmission sur les résidus [Skoglund et al., 1997].

- Au niveau du décodeur souple, c'est l'information apportée par le canal qui permet de corriger le biais entre la prédiction fixe d'un paramètre et sa trajectoire réelle pour la parole (non-stationnaire). Le modèle de prédiction est nécessairement plus « contraint » qu'au codeur afin de modéliser la redondance résiduelle. Ceci explique l'emploi de modèles AR ou de chaîne de Markov d'ordre 1. Il y a alors *propagation d'erreur* si ces modèles de prédiction exploitent uniquement les valeurs précédemment estimées des paramètres et non leur probabilité.

3.4.1.2 Le problème de la complexité

Un autre problème est celui de la *complexité* des méthodes de décodage souple. Le tableau 3.1 fait un bilan de la complexité des méthodes AK0, AK1 et AK2 en fonction de la taille N du dictionnaire de quantification et du nombre Q de paramètres exploités pour la prédiction intra-trame. On constate que les méthodes AK1 et AK2 deviennent très complexes à mettre en œuvre pour des codeurs utilisant de larges dictionnaires comme l'EFR[38]. Quant aux méthodes sous-optimales proposées pour réduire la complexité, telles la prédiction linéaire, elles comportent le risque d'une propagation d'erreur comme on l'a vu plus haut.

Complexité des algorithmes	**AK0**	**AK1**	**AK2**
Calcul de la probabilité *a posteriori*	0	$O(N^2)$	$O(QN^2)$
Estimation (MAP ou MMSE)	$O(N)$	$O(N)$	$O(N)$

Tableau 3.1 : Complexité en fonction de la taille N du dictionnaire de quantification et du nombre P de paramètres exploités par la prédiction intra-trame (algorithme AK2)

En résumé, le décodage souple ne peut être une solution intéressante dans le cas pratique du GSM EFR que si *la redondance résiduelle est réellement exploitable* et à condition de réduire la *complexité*

[38] Certains paramètres du codeur EFR comme le 3ième jeu de résidus LSF ou le délai de pitch peuvent avoir des dictionnaires de quantification de taille $N = 512$. (cf. Table A.2 d'allocation des bits en Annexe A).

des méthodes d'estimation. Nos travaux, présentés dans les chapitres qui suivent, s'attachent à répondre à ces conditions.

Chapitre 4

Décodage source à entrées souples : Application au GSM EFR

4.1 Introduction

Les techniques de décodage à entrées souples reposent sur l'utilisation conjointe d'une information sur la *fiabilité des données reçues* et d'une *prédiction des données à recevoir*. Ces informations sont respectivement obtenues à partir d'une estimée du canal et d'une modélisation de la redondance des paramètres du codeur. Leur nature et leur richesse dépend donc totalement du contexte dans lequel ces techniques sont mises en oeuvre. Notre contexte étant celui du système GSM, nous analysons dans ce chapitre la redondance résiduelle des paramètres du codeur EFR ainsi que l'information exploitable en sortie du décodeur canal. Nous proposons ensuite une mise en œuvre des algorithmes AK0 et AK1 abordés au chapitre précédent.

4.2 Redondance résiduelle du codeur EFR

Nous devons tout d'abord nous assurer qu'il existe bel et bien une redondance résiduelle en sortie du codeur de parole EFR. Mesurer la redondance suppose déjà que l'on se donne un modèle permettant de la représenter. Nous reprenons ici l'approche utilisée par [Alajaji et al., 1996] pour l'étude de la redondance du codeur CELP FS1016. Celle-ci consiste à caractériser les 3 formes de redondance (non-uniformité, mémoire et corrélation entre paramètres distincts) par des mesures d'entropies conditionnelles.

4.2.1 Modèle utilisé pour caractériser la redondance résiduelle

Plus précisément, nous noterons dans ce qui suit $i_n^{(\mathbf{v})}$ l'index de quantification associé au paramètre $\mathbf{v}_n$ calculé par le codeur de parole à l'instant[39] n, et définirons les 3 formes de redondance de la manière suivante :

- **Non uniformité :** Supposons que l'index de quantification $i_n^{(\mathbf{v})}$ est codé sur M bits. La redondance exploitable par la connaissance de la distribution $p(i_n^{(\mathbf{v})})$ en sortie de codeur peut se mesurer selon :

$$R0 = H_0(i_n^{(\mathbf{v})}) - H(i_n^{(\mathbf{v})}) \tag{4.1}$$

où $H_0(i_n^{(\mathbf{v})}) = M$ et $H(i_n^{(\mathbf{v})}) = -\sum_i p(i_n^{(\mathbf{v})} = i)\log_2(p(i_n^{(\mathbf{v})} = i))$.

L'entropie $H(i_n^{(\mathbf{v})})$ exprime le nombre, non nécessairement entier, de bits nécessaires pour représenter $i_n^{(\mathbf{v})}$ connaissant sa distribution *a priori*. $R0$ mesure donc le *nombre de bits redondants en sortie du codeur* si l'on connaît la distribution *a priori* $p(i_n^{(\mathbf{v})})$.

- **Mémoire :** La redondance en sortie de codeur due à la *corrélation temporelle* entre les index de quantification successifs $i_n^{(\mathbf{v})}$ et $i_{n-1}^{(\mathbf{v})}$ s'exprime selon :

$$R1 = H(i_n^{(\mathbf{v})}) - H\left(i_n^{(\mathbf{v})} \middle| i_{n-1}^{(\mathbf{v})}\right) \tag{4.2}$$

où
$$H\left(i_n^{(\mathbf{v})} \middle| i_{n-1}^{(\mathbf{v})}\right) = -\sum_i \sum_{i'} p\left(i_n^{(\mathbf{v})} = i\ ,\ i_{n-1}^{(\mathbf{v})} = i'\right)\log_2\left(p\left(i_n^{(\mathbf{v})} = i \middle| i_n^{(\mathbf{v})} = i'\right)\right) \tag{4.3}$$

est l'entropie conditionnelle de l'index $i_n^{(\mathbf{v})}$ sachant $i_{n-1}^{(\mathbf{v})}$. $R1$ mesure le nombre de bits redondants en sortie du codeur si l'on connaît la loi jointe $p\left(i_n^{(\mathbf{v})}, i_{n-1}^{(\mathbf{v})}\right)$.

- **Corrélation entre index de QV :** Considérons des paramètres distincts $\mathbf{v}_n$ et $\mathbf{v}'_n$ calculés au même instant n (trame ou sous-trame) et quantifiés séparément. On cherche à évaluer la corrélation résiduelle entre les index $i_n^{(\mathbf{v})}$ et $i_n^{(\mathbf{v}')}$ issus de ces processus de quantification séparés. On peut mesurer leur redondance selon :

$$R2 = H(i_n^{(\mathbf{v})}) - H\left(i_n^{(\mathbf{v})} \middle| i_n^{(\mathbf{v}')}\right) \tag{4.4}$$

où $H\left(i_n^{(\mathbf{v})} \middle| i_n^{(\mathbf{v}')}\right)$ est l'entropie conditionnelle de l'index de quantification du paramètre $\mathbf{v}_n$ sachant l'index de quantification du paramètre $\mathbf{v}'_n$, c'est-à-dire le nombre de bits nécessaires pour représenter $i_n^{(\mathbf{v})}$ connaissant la loi jointe $p\left(i_n^{(\mathbf{v})}, i_n^{(\mathbf{v}')}\right)$.

---

[39] L'instant n désignera la trame ou la sous-trame selon le paramètre considéré.

On notera que l'on a mesuré ici indépendamment la redondance *temporelle*, modélisée par une loi jointe de la forme $p\left(i_n^{(\mathbf{v})}, i_{n-1}^{(\mathbf{v})}\right)$, et la redondance entre *paramètres distincts*, mesurée par la loi de forme $p\left(i_n{}^{(\mathbf{v})}, i_n{}^{(\mathbf{v}')}\right)$. Ce choix a été imposé afin de limiter la dimension des lois jointes à apprendre sur notre base de données. En contre-partie, ce modèle ne permet pas de dire dans quelle mesure ces deux redondances ne se recouvrent pas.

Une autre limitation de ce modèle est qu'on calcule uniquement des *statistiques moyennes*, ceci s'applique bien à un signal stationnaire mais ce n'est évidemment pas le cas du signal de parole. Ainsi, les corrélations mesurées seront une moyenne entre les corrélation élevées des paramètres pour les segments stationnaires (segments voisés) de la parole avec celles, certainement plus faibles correspondant aux non-stationnarités (transitions, plosives, etc.). Cependant, les algorithmes présentés au chapitre précédent reposent également sur ce modèle (probabilités de transition *moyennes* entre index de quantification), nous avons jugé intéressant de l'appliquer au cas de l'EFR afin de pouvoir par la suite en dériver une mise en œuvre de ces algorithmes.

4.2.2 Résultats obtenus

Nous pouvons estimer les 3 formes de redondance résiduelle en calculant les probabilités $p(i_n{}^{(\mathbf{v})})$, $p\left(i_n^{(\mathbf{v})}, i_{n-1}^{(\mathbf{v})}\right)$, $p\left(i_n{}^{(\mathbf{v})}, i_n{}^{(\mathbf{v}')}\right)$ sur une base de données. Les caractéristiques de la base de donnée utilisée sont reportées sur le Tableau 4.1. **Cette base de données sera la base d'apprentissage utilisée pour toute la suite de nos travaux.**

Corpus pour moitié en langue anglaise et pour moitié en français, enregistré par 8 locuteurs (2 hommes et 2 femmes / anglais ; 2 hommes et 2 femmes / français).
Echantillons de parole constitués de doubles phrases phonétiquement équilibrées.
Restriction aux périodes *d'activité vocale* uniquement (21 minutes au total).

Tableau 4.1 : Corpus de parole utilisé pour l'apprentissage

Cette base de données permet de générer 65000 trames de paramètres du codeur EFR. Le codeur EFR est présenté en détail en Annexe A, on rappellera juste ici que les paramètres quantifiés en sortie de ce codeur sont les résidus de prédiction MA des LSF (quantification vectorielle QV par jeux de 5 paires), les résidus de prédiction MA du gain de dictionnaire fixe (quantification scalaire QS), le gain de dictionnaire adaptatif et le délai de pitch (quantification scalaire QS). Dans cette étude de la redondance résiduelle, on s'intéresse plus précisément aux *index de quantification* associés à ces paramètres quantifiés. La table 4.2 résume ces index de quantification et leur notations.

$i_n^{(LSF_k)}$	Index de QV du $k^{ième}$ *jeu de résidus LSF* à l'instant n, $(k = 1,2,...,5)$
$i_n^{(gp)}$	Index de QS du gain de dictionnaire adaptatif à l'instant n
$i_n^{(gc)}$	Index de QS du résidu de gain de dictionnaire fixe à l'instant n
$i_n^{(lag)}$	Index de QS du délai (pitch) à l'instant n

Tableau 4.2 : Paramètres étudiés en sortie du codeur EFR

Les entropies estimées sur la base de données sont reportées sur les tables 4.3 à 4.6 pour ces 4 types de paramètres du codeur EFR. Les *redondances résiduelles* associées à la non-uniformité, la mémoire ou la corrélation inter-paramètres se déduisent par comparaison avec l'entropie H_0, c'est-à-dire le nombre de bits effectivement utilisé pour coder l'index de quantification. L'indice n désigne la trame dans le cas des résidus LSF et la sous-trame pour les autres paramètres[40].

	$H_0\left(i_n^{(LSF_k)}\right)$	$H\left(i_n^{(LSF_k)}\right)$	$H\left(i_n^{(LSF_k)} \middle\| i_{n-1}^{(LSF_k)}\right)$	$H\left(i_n^{(LSF_k)} \middle\| i_n^{(LSF_{k-1})}\right)$
$k = 1$	7	5.37	4.71	
$k = 2$	8	7.01	5.80	6.05
$k = 3$	9	8.25	6.25	6.97
$k = 4$	8	7.38	6.40	6.10
$k = 5$	6	5.39	5.06	4.92

Tableau 4.3 : Entropies des indices de quantification des LSF (non-uniformité, mémoire, corrélation entre jeux de résidus LSF)

Dans le cas des LSF, la redondance résiduelle provient essentiellement de la corrélation temporelle et de la corrélation entre jeux de résidus de prédiction des LSF. Cette redondance est assez significative. Ceci s'explique par la forme sous-optimale de quantification utilisée[41] qui modélise imparfaitement la forte corrélation temporelle des LSF dans les segments stationnaires de la parole et la corrélation entre LSF due à leur relation d'ordre.

[40] Dans le cas du délai de pitch, les sous-trames 2 et 4 ne sont pas étudiées car elles sont codées en différentiel par rapport aux sous-trames 1 et 3, c'est-à-dire qu'elle intègrent déjà la corrélation entre sous-trames.

[41] La prédiction MA modélise moins bien la corrélation temporelle qu'une prédiction AR, et la matrice formée par les 2 vecteurs de LSF calculés à chaque trame est quantifiée par blocs (5 blocs) et non de manière totalement conjointe (cf. Annexe A).

$H_0\left(i_n^{(gp)}\right)$	$H\left(i_n^{(gp)}\right)$	$H\left(i_n^{(gp)}\middle\vert i_{n-1}^{(gp)}\right)$
4	3.92	3.53
$H_0\left(i_n^{(gc)}\right)$	$H\left(i_n^{(gc)}\right)$	$H\left(i_n^{(gc)}\middle\vert i_{n-1}^{(gc)}\right)$
5	4.11	3.93

Tableau 4.4 : Entropies des index de gains (non-uniformité, mémoire)

$H\left(i_n^{(gp)}\middle\vert i_n^{(gc)}\right)$	$H\left(i_n^{(gc)}\middle\vert i_n^{(gp)}\right)$
3.78	3.98

Tableau 4.5 : Entropies des index de gains (corrélation entre gains)

$H_0\left(i_n^{(lag)}\right)$	$H\left(i_n^{(lag)}\right)$	$H\left(i_3^{(lag)}\middle\vert i_1^{(lag)}\right)$
9	8.70	5.55

Tableau 4.6 : Entropies du délai de pitch (non-uniformité, mémoire)

Pour le délai de pitch, la redondance entre les sous-trames 1 et 3 était prévisible car celui-ci est très stationnaire dans les segments voisés.

Nous avons étudié la corrélation mutuelle des gains de dictionnaire fixe et adaptatif, en plus de leur non-uniformité et de leur corrélation temporelle. Ceci était motivé par le fait que certains codeurs comme le G.729 [ITU-T, G.729] quantifient conjointement ces deux paramètres. On constate cependant que l'essentiel de la redondance est due à la non-uniformité alors que les corrélations mutuelle et temporelle sont assez peu significatives.

Pour conclure, il existe une forme plus générale de redondance entre paramètres que nous n'avons pas étudiée, il s'agit de l'information apportée par la classification *voisé / non-voisée* de la trame. En effet, le comportement des LSF n'est pas le même selon que la trame est voisée ou non (certains codeurs intègrent cette information voisé / non-voisé). Un modèle *a priori* caractérisant la redondance des paramètres séparément pour ces deux états de la parole, avec des probabilités de transition d'un état à l'autre, permettrait également une meilleure prise en compte du caractère non-stationnaire de la parole. Nous reviendrons par la suite sur un tel modèle.

4.3 Vraisemblance en sortie du canal équivalent

Pour mettre en œuvre une technique de décodage de parole à entrées souples, il est nécessaire de disposer de la *vraisemblance* (3.6). du paramètre transmis (ou de son index de quantification). Nous avons implémenté pour cela un décodeur canal *à sorties souples* de type SOVA (*Soft Output Viterbi Algorithm*) à la place du décodeur canal classique de l'EFR. Cet algorithme est décrit en Annexe D. Il renvoie une estimée de la probabilité d'erreur $p_e(m)$ associée à chaque bit $\hat{b}_m$ décodé. La connaissance de la probabilité d'erreur $p_e(m)$ pour chacun des bits codant un index de quantification permet alors de calculer la vraisemblance de cet index de quantification à l'aide des relations (3.5) et (3.6).

Afin de vérifier la validité de l'estimation fournie par le SOVA, nous avons comparé le *Taux d'Erreur Binaire* TEB effectivement *mesuré* sur l'ensemble des bits en sortie du décodeur canal, avec la *moyenne* des probabilités d'erreur *estimées* par le SOVA pour ces bits (p_e *moyen*). Cette comparaison est illustrée Figure 4.1 pour un canal de type TU50 (cf. Annexe C) et des niveaux de C/I compris entre 2 et 8 dB. La très forte similitude entre le TEB mesuré et la moyenne des probabilités d'erreur estimées permet de conclure que le SOVA délivre une estimée *non-biaisée* de la probabilité d'erreur $p_e(m)$ associée à chaque bit $\hat{b}_m$. En revanche, les approximations posées par l'algorithme SOVA (cf. Annexe D) doivent conduire à une estimée *bruitée* de la probabilité d'erreur individuelle $p_e(m)$. Cependant, la variance de l'estimée $p_e(m)$ ne peut être évaluée dans notre contexte de simulation[42].

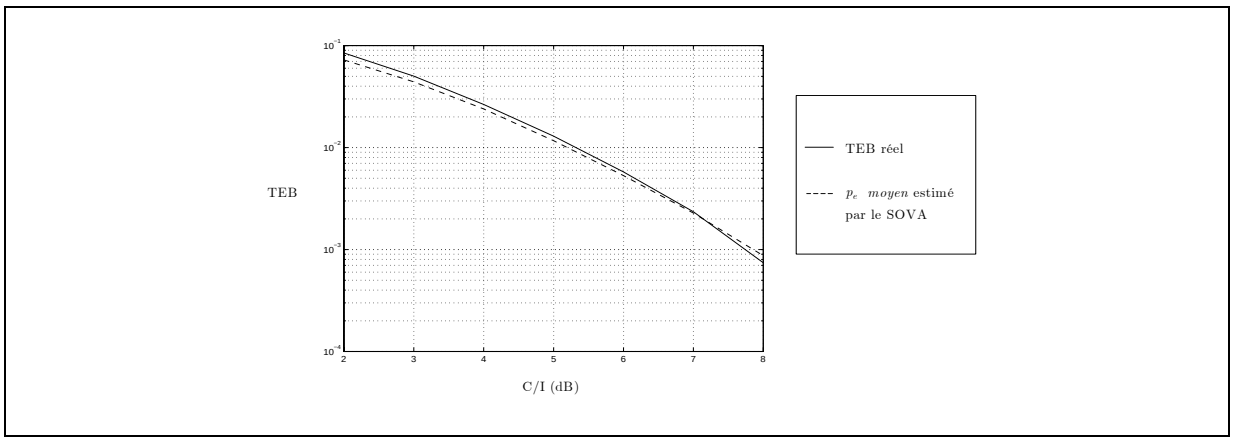

Figure 4.1 : Comparaison du TEB effectif en sortie du décodeur canal et de la probabilité d'erreur estimée par le SOVA

[42] Nous utilisons des « patterns d'erreurs » fixes (cf. Annexe C) pour modéliser les erreurs introduites par le canal radio-mobile. La nature déterministe des erreurs ainsi introduites ne permet pas de faire une statistique individuelle des bits décodés, on doit se limiter à des moyennes d'ensemble (sur l'ensemble des bits de la trame).

Au-delà de la simple vérification de sa validité, il est intéressant d'étudier plus en détail la nature de l'information apportée par la vraisemblance en sortie du canal équivalent du GSM. En effet, l'emploi de la vraisemblance (3.6) en remplacement de l'information « tout ou rien » sur la qualité du canal, utilisée classiquement par le décodeur de parole (indicateur BFI), est motivée par les deux potentialités suivantes :

- **Masquage souple** par l'exploitation conjointe d'une information a priori avec celle issue du canal.
- **Masquage sélectif** puisque l'on calcule une vraisemblance par paramètre (ou index de quantification associé).

Cependant, la notion de masquage sélectif n'a d'intérêt que si les paramètres d'une même trame ne sont pas simultanément corrompus en présence d'erreurs de transmission. La vraisemblance (3.6) permet alors d'exploiter cette sorte de « diversité de réception[43] » entre paramètres. Cette hypothèse est utilisée par l'algorithme de décodage AK2 présenté au chapitre précédent et qui exploite la corrélation entre les paramètres d'une même trame pour corriger ceux d'entre eux qui sont erronés. Or, les erreurs introduites par un canal radio-mobile sont essentiellement de type « *burst* », c'est-à-dire très regroupées temporellement, et l'hypothèse d'une « diversité de réception » entre paramètres doit être vérifiée. Ceci est l'objet de l'analyse menée dans les paragraphes suivants.

Afin d'étudier si les paramètres d'une même trame sont simultanément corrompus ou non, nous formons un *indicateur de paramètre invalide* BPI (par analogie avec l'indicateur de trame invalide BFI) à partir de la vraisemblance de l'indice de quantification associé à ce paramètre. Plus précisément, considérons à nouveau l'index de quantification $i_n^{(\mathbf{v})}$ du paramètre $\mathbf{v}_n$, codé sur M bits, l'indicateur BPI du paramètre $\mathbf{v}_n$ à l'instant n est défini selon :

$$BPI(\mathbf{v}_n) = H_{vrais}(i_n^{(\mathbf{v})})/M \tag{4.5}$$

avec :
$$H_{vrais}(i_n^{(\mathbf{v})}) = -\sum_i p(j_n^{(\mathbf{v})}|i_n^{(\mathbf{v})} = i)\log_2(p(j_n^{(\mathbf{v})}|i_n^{(\mathbf{v})} = i)) \tag{4.6}$$

où $p(j_n^{(\mathbf{v})}|i_n^{(\mathbf{v})})$ est la vraisemblance calculée selon (3.6) de l'index de quantification émis $i_n^{(\mathbf{v})}$ étant donné l'index reçu $j_n^{(\mathbf{v})}$ en sortie de canal équivalent (sortie du SOVA). Autrement dit, $H_{vrais}(i_n^{(\mathbf{v})})$ mesure l'entropie de la distribution de vraisemblance de l'index $i_n^{(\mathbf{v})}$. Cette entropie est dépendante de la valeur de l'index reçu $j_n^{(\mathbf{v})}$ et des probabilités d'erreur calculées par le SOVA mais on omettra d'y faire référence dans les notations.

L'indicateur $BPI(\mathbf{v}_n)$ s'interprète donc comme le rapport du nombre de bits qui seraient nécessaires pour coder le paramètre $\mathbf{v}_n$ connaissant les informations reçues en sortie de canal équivalent, au

[43] Par analogie avec la diversité de réception en traitement d'antennes exploitant des trajets de propagation différemment brouillés.

nombre de bits M utilisés au codeur de parole. Il constitue ainsi une mesure *normalisée*[44] du degré de dégénérescence du paramètre reçu (1 pour un paramètre totalement dégénéré, c'est-à-dire pouvant prendre toutes les valeurs possibles de son dictionnaire de quantification, 0 si la valeur reçue est certaine). A la différence de l'indicateur BFI, l'indicateur BPI renvoie une valeur souple, variant continûment entre 0 et 1.

La Figure 4.2 illustre la correspondance entre l'indicateur BPI (en bas) calculé selon (4.5) et la distribution de vraisemblance de l'index de quantification (au milieu) calculée selon (3.6). L'intensité de niveau de gris code l'amplitude de la vraisemblance (l'échelle utilisée est logarithmique). On vérifie que l'indicateur BPI correspond bien à une mesure d'entropie de la loi de vraisemblance. L'index de quantification est dans cet exemple celui du 1er jeu de résidus LSF quantifiés (noté LSF1). Les artefacts dus aux erreurs de transmission peuvent être repérés directement sur le signal de parole décodé sans procédure de masquage (illustré en haut).

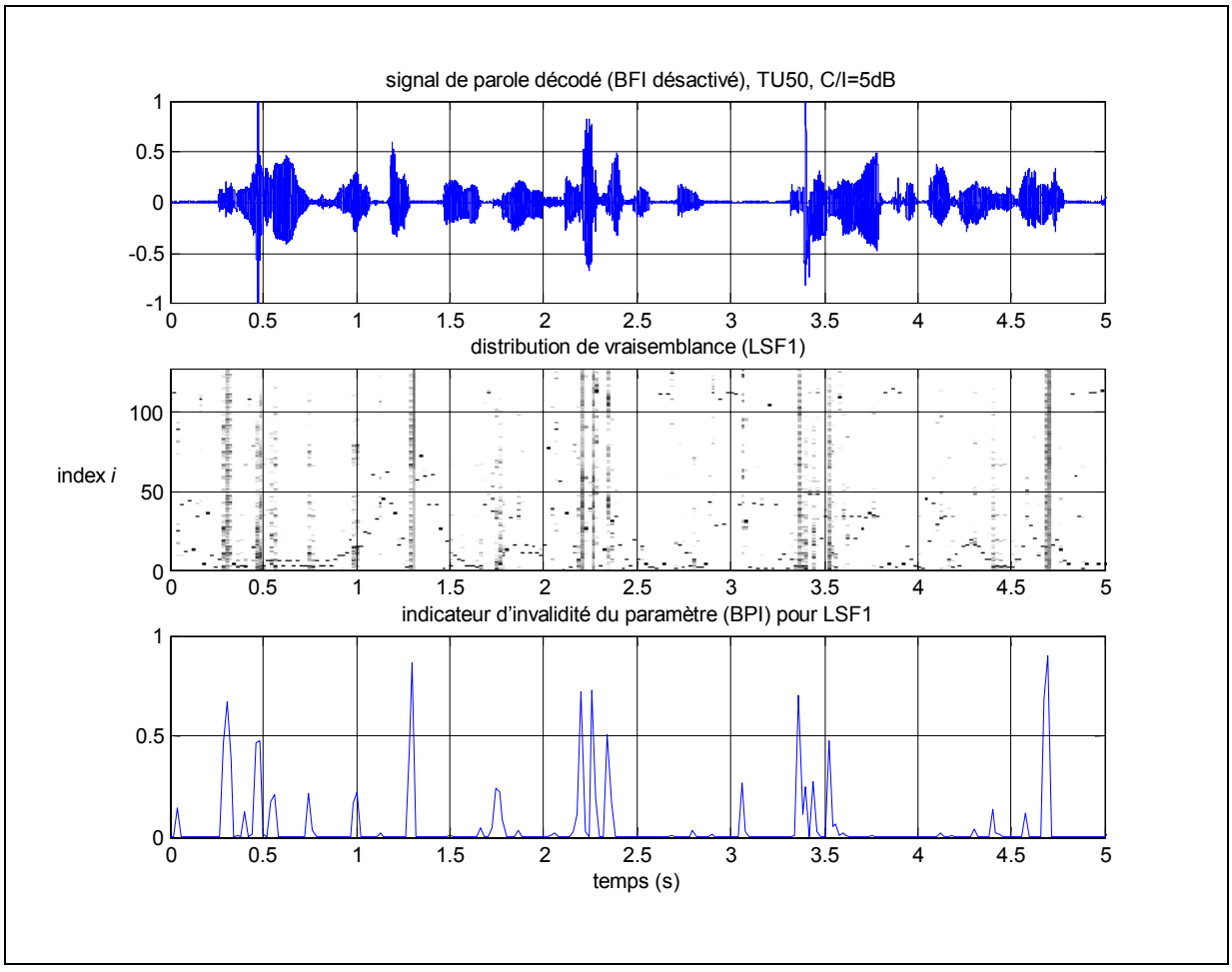

Figure 4.2 : Signal de parole décodé (masquage désactivé), distribution de log-vraisemblance (milieu) et indicateur d'erreur BPI (bas) associé à l'index de quantification LSF1 reçu au cours du temps (TU50, $C/I = 5$dB)

[44] L'équation (4.5) est à rapprocher de celles définissant les mesures de redondance (4.1) à (4.4), la soustraction est remplacée par un quotient afin d'obtenir une grandeur normalisée, indépendante du nombre de bits M.

Pour vérifier l'existence éventuelle d'une « diversité de réception » entre paramètres, on calcule le coefficient de corrélation entre les indicateurs BPI obtenus pour chaque paramètre. Un coefficient de corrélation proche de l'unité signifie que les paramètres sont affectés de manière uniforme par les erreurs introduites par le canal radio-mobile. A l'inverse, si les BPI sont décorrélés entre eux, l'exploitation de la redondance résiduelle intra-trame et le masquage sélectif sont justifiés.

Le coefficient de corrélation entre BPI a été mesuré sur des communications d'une durée de 32 s (1600 trames), pour des niveaux de C/I compris entre 2 et 7 dB et un canal de type TU50 (cf. Annexe C). La Figure 4.3 illustre les résultats obtenus pour les 2 premiers jeux de résidus LSF quantifiés (notés LSF1 et LSF2). Les valeurs moyennes des BPI ont été retranchées avant calcul de leur inter-corrélation normalisée (coefficient de corrélation). Ces valeurs moyennes sont représentées sur le graphique gauche, elles permettent de vérifier d'une part la cohérence du comportement du BPI avec le niveau de C/I, et d'autre part l'invariance du niveau moyen entre paramètres (le BPI étant un critère normalisé). Les coefficients de corrélation sont représentés sur le graphique droit, on constate que ceux-ci sont très élevés sur toute la plage des niveaux de C/I évalués. On en conclut que les paramètres d'une même trame semblent être simultanément corrompus par les erreurs introduites par le canal radio-mobile et de manière relativement uniforme. Ceci est dû à la statistique des erreurs du canal radio-mobile et paraît minorer les gains à attendre d'une prise en compte de la redondance intra-trame dans le cas du GSM.

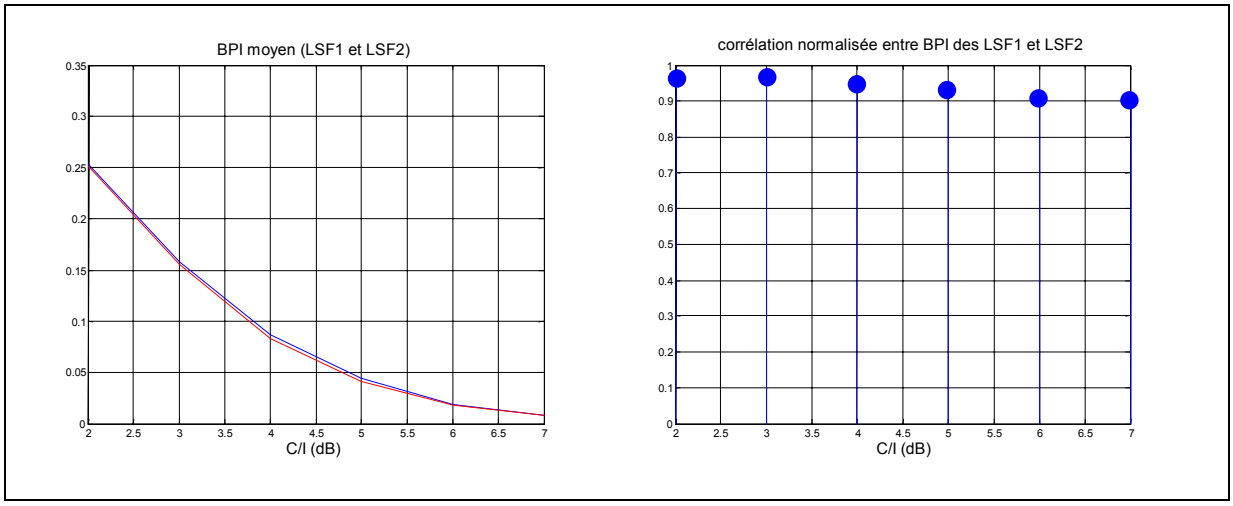

Figure 4.3 : Inter-corrélation normalisée et valeur moyenne des indicateurs d'erreur BPI associés aux index de quantification LSF1 et LSF2 (TU50, C/I compris entre 2 et 7 dB)

4.4 Mise en œuvre du décodage souple

Les analyses précédentes ont montré qu'il existait une redondance résiduelle en sortie du codeur de parole EFR. Nous proposons ici d'exploiter cette redondance au niveau du décodeur parole, en mettant en oeuvre les algorithmes de décodage souple présentés au paragraphe 3.3. Nous envisageons successivement les 3 types d'algorithmes suivants :

- Décodage souple sans *a priori* (paragraphe 3.3.2.1)
- Décodage « AK0 » exploitant la non-uniformité (paragraphe 3.3.2.2)
- Décodage « AK1 » exploitant la corrélation temporelle entre trames (paragraphe 3.3.2.3)

L'algorithme AK2 n'a pas été mis en œuvre ici car jugé trop complexe dans le cas de l'EFR.

Les performances de ces algorithmes seront évaluées à l'aide des *critères objectifs* présentés au Chapitre 1 pour différents niveaux d'interférences C/I compris entre 2dB et 10dB et pour la configuration typique de canal radiomobile TU50 avec saut de fréquence idéal. Les notes MOS estimées par l'algorithme PESQ et la distance cepstrale sont dans chaque cas moyennées sur l'ensemble des échantillons de parole d'une *base de test* dont les caractéristiques sont reportées Tableau 4.7. Cette base de test est distincte de la base d'apprentissage utilisée pour estimer les modèles *a priori* des algorithmes.

Corpus enregistré par 4 locuteurs (2 hommes et 2 femmes / français).
18 doubles phrases de 8 secondes chacune (correspondant à 90 secondes d'activité vocale).

Tableau 4.7 : Corpus de parole utilisé pour l'évaluation des algorithmes

4.4.1 Décodage souple sans a priori

Comme on l'a vu, le décodage parole « classique » re-synthétise le signal de parole à partir de paramètres estimés au sens du Maximum de Vraisemblance MV puisque la décision a été prise par l'algorithme de Viterbi au niveau du décodeur de canal. Le décodage souple sans *a priori* consiste à effectuer l'estimation des paramètres au sens du MMSE à partir des vraisemblances calculées par un décodeur canal à sorties souples (SOVA) selon l'équation (3.11). L'argument sous-jacent est qu'il est plus pertinent pour améliorer la qualité perçue de la parole, de *minimiser une erreur quadratique* dans le domaine des paramètres (point de vue « codage parole »), que de *minimiser un taux d'erreur binaire* (point de vue « codage canal »).

Cependant, la distance quadratique n'est pas forcément le critère optimal d'un point de vue subjectif. Ainsi, [Fingscheidt et al., 1997] observent que l'estimateur du Maximum *A Posteriori* MAP est plus adapté pour le délai de pitch dans le cas du GSM Full-Rate. D'autre part, dans le cas des LSF, la distance utilisée lors du processus de quantification n'est pas la simple distance quadratique mais fait intervenir une *pondération perceptuelle* de chaque LSF, qui est fonction de l'écartement avec les LSF adjacentes. Enfin, le gain de dictionnaire fixe[45] est quantifié dans le domaine logarithmique. Dans la mise en œuvre du décodeur souple, c'est donc dans le domaine logarithmique que nous appliquerons le critère MMSE pour le gain de dictionnaire fixe. En revanche, le calcul des *pondérations perceptuelles* pour les LSF ne pourrait être mis en oeuvre que dans le cadre d'un décodage *conjoint* des LSF, beaucoup trop complexe pour être envisagé et l'on se limitera donc à la distance quadratique entre LSF.

Dans ce qui suit, nous comparons les performances du décodeur souple (MMSE) au décodeur « classique » (MV) avec ou sans la procédure de masquage donnée en exemple par la norme du GSM EFR [GSM, 06.61]. Deux versions du décodeur souple sont évaluées, selon que le délai de pitch est estimé comme les autres paramètres par le critère MMSE, ou traité à part (estimation MAP[46] du délai de pitch).

Les conditions étudiées correspondent à des niveaux de C/I variant par pas de 1dB sur la plage [2dB-7dB] et au-delà pour la seule valeur de C/I égale à 10 dB (afin de vérifier les performances asymptotiques pour les faibles niveaux de brouillage). Pour chacune de ces conditions, les performances des décodeurs sont évaluées à l'aide des deux critères présentés au Chapitre 1 :

- La *note MOS* estimée par l'algorithme PESQ. On rappellera que cette note est ici utilisée en tant que *distance « perceptuelle »* par rapport au signal de parole *non codé* et qu'un **écart** entre notes peut être considéré comme **significatif à partir de 0.2 MOS**. Les notes MOS obtenues sont reportées sur la Figure 4.4.

- La *distance cepstrale moyenne* par rapport au signal de parole *décodé sans erreurs* (C/I infini). La moyenne est ici restreinte aux périodes d'activité vocale uniquement. La distance cepstrale moyenne est représentée Figure 4.5.

[45] Plus exactement, c'est le résidu de la prédiction MA du gain de dictionnaire fixe qui est quantifié. Cette prédiction MA s'effectue justement dans le domaine logarithmique.

[46] Dans le cas présent, l'estimation MAP se réduit à l'estimation MV puisque aucun *a priori* sur les paramètres du codeur n'est utilisé.

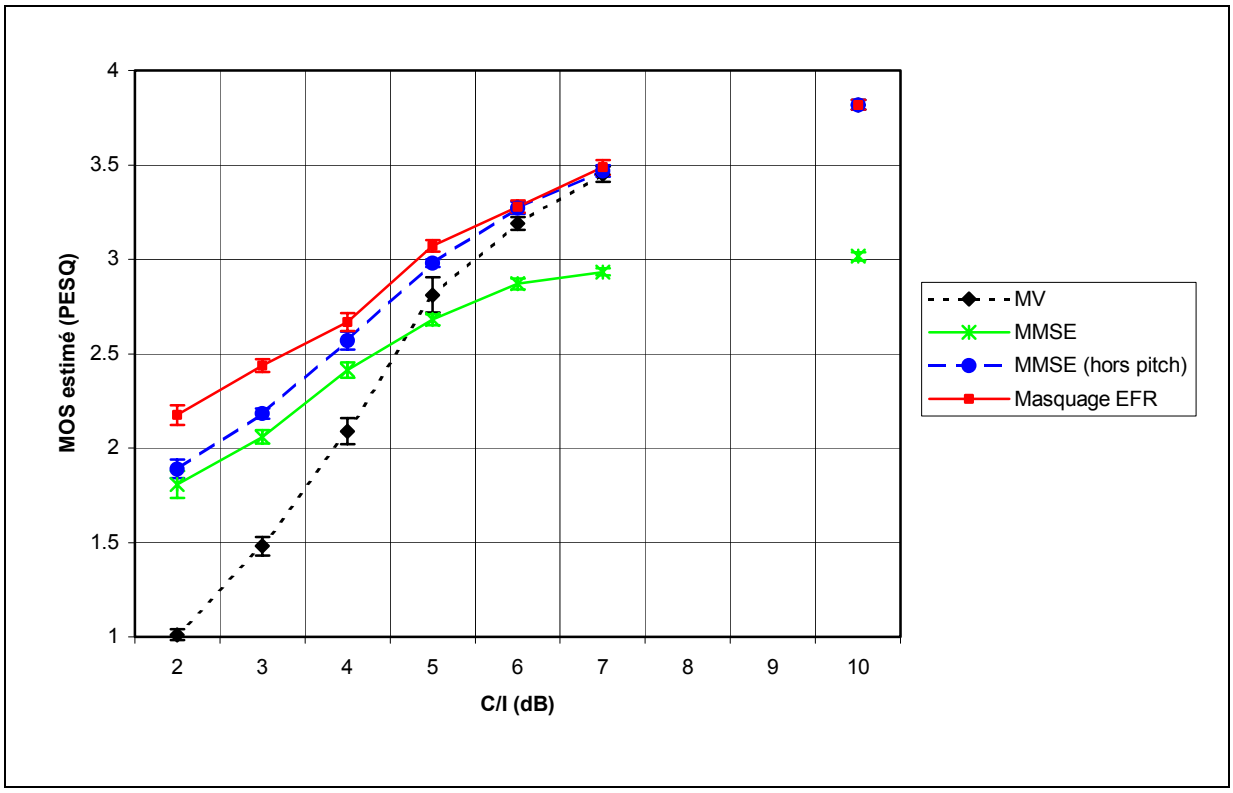

Figure 4.4 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

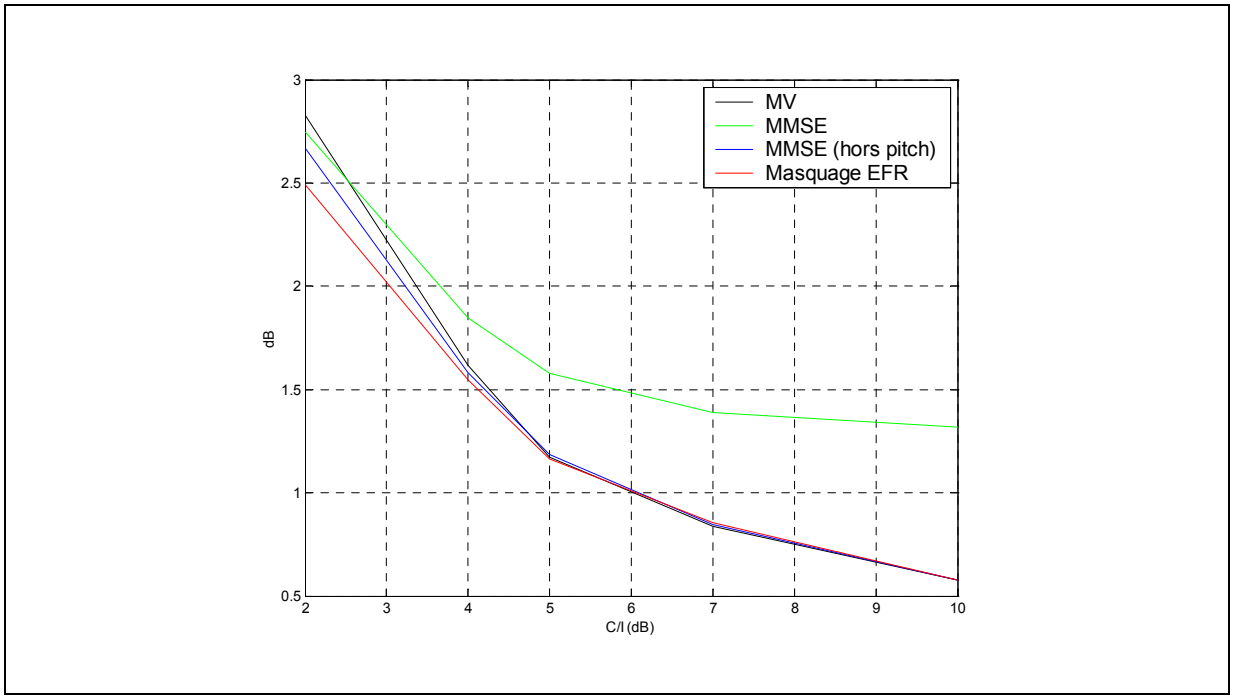

Figure 4.5 : Distance cepstrale moyenne sur les trames d'activité vocale en fonction du C/I

On notera en premier lieu que le décodage direct (MV), c'est-à-dire le décodeur classique de l'EFR sans procédure de masquage, est plus fortement pénalisé par le critère PESQ que par la distance cepstrale

moyenne. Ceci s'explique par le fait que la note PESQ prend en compte l'intégralité du signal[47] alors que la distance cepstrale moyenne est restreinte aux instants d'activité vocale. Cette dernière ne prend donc pas en compte les artefacts générés par le décodeur MV dans les périodes de non-activité vocale et bien visibles sur les signaux illustrés Figure 4.11.

Il apparaît très clairement qu'il est préférable d'utiliser l'estimateur MAP pour le délai de pitch. On constate en effet que les performances du décodeur mettant en oeuvre l'estimation MMSE du pitch ne convergent pas vers celles du décodeur classique lorsque le C/I augmente mais présentent un effet de saturation. A l'inverse, le décodeur exploitant le critère MMSE pour tous les paramètres à l'exclusion du pitch (MMSE hors pitch) converge bien vers les performances du décodeur classique pour les C/I élevés. Une explication possible est que le biais de l'estimateur MAP tend rapidement vers zéro à mesure que la confiance dans les données reçues du canal augmente alors qu'il subsiste un biais dans le cas de l'estimateur MMSE. La présence d'un biais sur le délai estimé de pitch tend à détériorer la structure harmonique des segments voisés[48], comme l'illustre la Figure 4.6 où le décodeur utilisant l'estimation MMSE du pitch est comparé au décodeur classique pour un C/I égal à 10 dB.

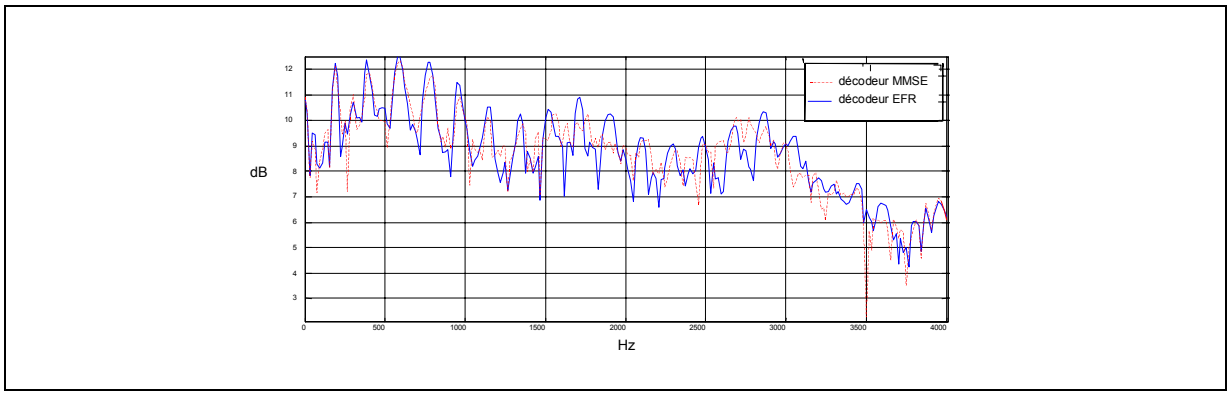

Figure 4.6 : Spectre d'un segment voisé (rouge : décodeur MMSE, bleu : décodeur EFR classique)

Dans tout ce qui suit, nous utiliserons systématiquement l'estimateur MAP pour le délai de pitch et en l'absence d'ambiguïté, dénommerons simplement « MMSE » le décodeur souple utilisant le critère MMSE pour tous les paramètres à l'exclusion du pitch.

On observe que ce décodage MMSE qui n'exploite pourtant aucun *a priori* sur la redondance résiduelle se rapproche des performances de la procédure de masquage classique de l'EFR. En fait, l'impact subjectif des dégradations est très différent entre ces deux approches de décodage. Dans le cas du décodeur MMSE, les erreurs sont moindres mais réparties plus uniformément au cours du temps, donnant l'impression d'un signal corrompu par du bruit. Le masquage classique de l'EFR crée la

[47] La distance spectrale « perceptuelle » calculée par l'algorithme PESQ est moyennée sur toute la durée du signal mais avec une pondération différente pour les segments d'activité vocale et les segments de non-activité.

[48] Le délai de pitch détermine la période avec laquelle le signal d'excitation (dictionnaire fixe) est répété pour construire le dictionnaire adaptatif (excitation périodique). Un biais sur le délai de pitch va entraîner un déphasage entre les composantes résiduelles du pitch présentes dans le signal d'excitation et donc détruire la structure harmonique.

sensation bien connue de « trous » dans le flux de parole liés aux pertes de trames. La Figure 4.11 illustre ceci par des exemples de signaux de parole décodés pour un C/I de 2 dB, ainsi que la distribution de l'erreur cepstrale au cours du temps.

L'utilisation de la seule information de vraisemblance en sortie du décodeur canal par l'estimateur MMSE permet déjà d'obtenir un gain qualitatif par rapport au décodeur MAP sans procédure de masquage. Si l'on considère que la procédure de masquage de l'EFR est une technique exploitant un modèle *a priori* empirique des paramètres du codeur, on peut s'attendre à surpasser cette méthode en utilisant le décodeur souple (MMSE) avec une information *a priori*. Ceci conduit aux algorithmes AK0 et AK1 exposés précédemment.

4.4.2 Décodage AK0

La mise en œuvre du décodage AK0 est très simple, elle nécessite seulement de stocker au niveau du décodeur parole, les probabilités *a priori* des indices de quantification $p(i_n)$ sous forme d'histogrammes. Les Figures 4.7 et 4.8 comparent les performances des estimateurs MV, MMSE (hors pitch), AK0 (MMSE hors pitch) avec celles du masquage « classique » de l'EFR.

La note MOS estimée fait apparaître une légère supériorité de l'estimateur AK0 par rapport au masquage de l'EFR sans que celle-ci puisse être considérée comme réellement significative. Cette supériorité est plus clairement confirmée par la distance cepstrale moyenne.

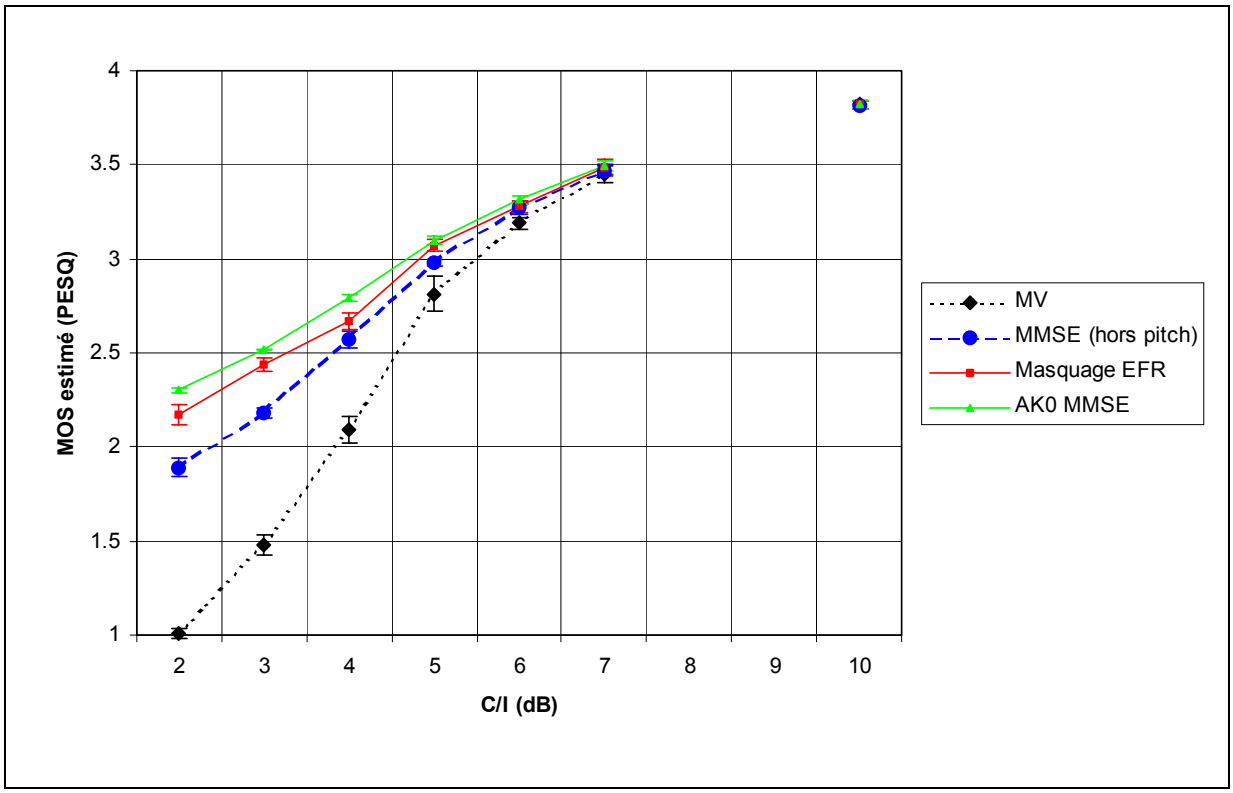

Figure 4.7 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

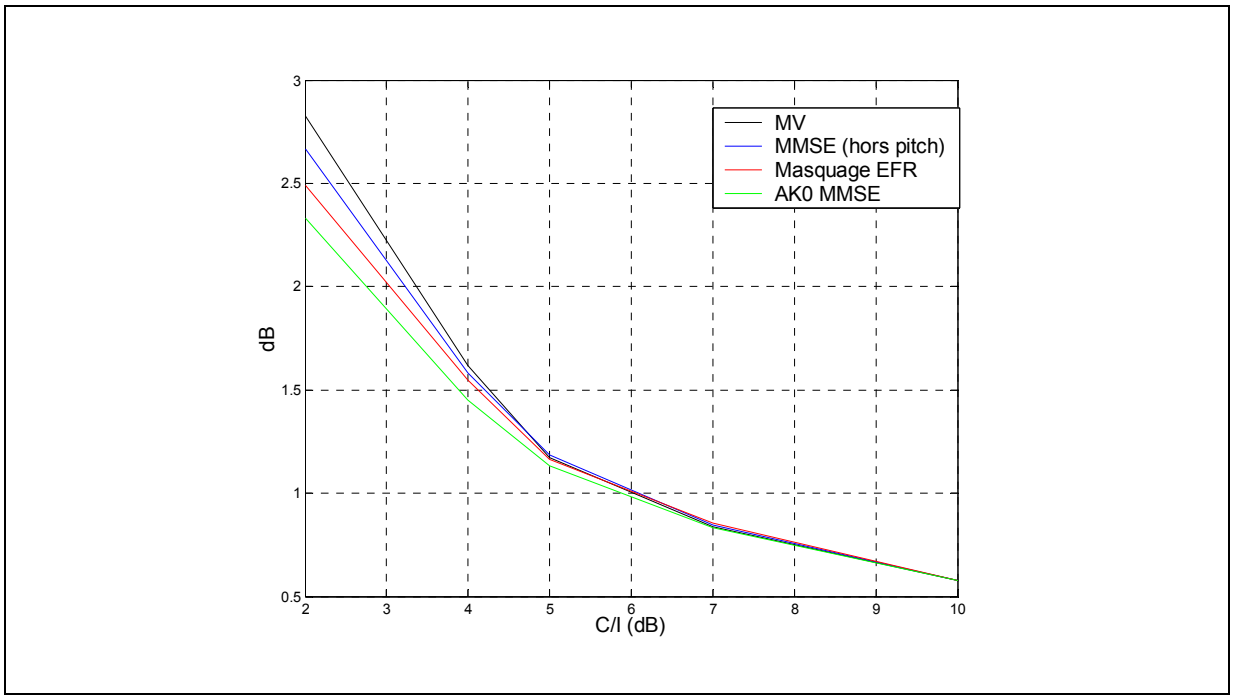

Figure 4.8 : Distance cepstrale moyenne sur les trames d'activité vocale en fonction du C/I

Des exemples de signaux de parole décodés selon ces algorithmes sont illustrés Figure 4.11 avec, pour chacun d'eux, la distance cepstrale évaluée par rapport à la parole codée de référence. On constate, en premier lieu, la présence d'artefacts dans les zones de silence pour le décodeur AK0. Le masquage de l'EFR, qui enclenche une substitution de trame, n'a évidemment pas ce problème. En revanche, le signal de parole apparaît étonnamment bien restauré, si on le compare au décodage direct MV. On observe également une réduction des « bruits impulsifs[49] » dans la parole par rapport au décodeur MMSE. Ceci montre bien que la prise en compte de la redondance permet un masquage des artefacts.

A l'écoute, le signal de parole apparaît plus « bruité » en sortie du décodeur AK0 mais il est également toujours intelligible, ce qui n'est pas le cas avec la stratégie de masquage de l'EFR. Comme on l'a noté précédemment pour le MMSE, la gêne perçue pour les faibles niveaux de C/I est très différente de celle du masquage classique. Dans le cas des décodeurs MMSE et AK0, le signal de parole apparaît entaché de « bruits impulsifs ». Ce bruit correspond en réalité à des distorsions non-linéaires mais celles-ci sont nettement plus atténuées que celles associées au décodage direct MV.

Cette sensation de bruit est mieux acceptée que celle d'un signal discontinu associé au masquage de l'EFR. Cet effet d'ordre cognitif n'est pas pris en compte par les critères objectifs utilisés ici et plaide en faveur du décodage souple.

[49] Par commodité, on appelera ainsi les pics impulsionnels visibles sur la forme temporelle des signaux de parole. Il s'agit d'un abus de language car la distortion associée n'est pas linéaire.

4.4.3 Décodage AK1

A partir des probabilités $p(i_n)$ et $p(i_n, i_{n-1})$ apprises sur la base de données, nous pouvons mettre en œuvre l'algorithme AK1 selon la récursion (3.15). Les figures 4.9 et 4.10 présentent l'évaluation des performances de cet algorithme au regard des critères MOS estimés (PESQ) et distance cepstrale.

Les performances de l'algorithme AK1 sont inférieures à celles de AK0 et du masquage classique. Ce résultat est surprenant car le modèle *a priori* utilisé par l'algorithme AK1 inclut la non-uniformité exploitée par l'algorithme AK0 et devrait donc être au moins aussi performant que ce dernier. Deux hypothèses peuvent être avancées pour expliquer ce résultat :

- Le modèle de corrélation temporelle utilisé (probabilité de transition *fixes*) introduit plus d'erreurs qu'il n'en corrige car il n'est pas adapté au caractère non-stationnaire de la parole.

- La dimension des données à estimer pour le modèle *a priori* AK1 (matrice des probabilités de transition entre index de quantification) requiert une quantité de données d'apprentissage nettement plus élevée que la base de données dont nous disposons. Dès lors, si le modèle est incomplètement appris, il ne peut offrir ses performances optimales sur une base de test.

On remarquera que les performances de l'algorithme AK1 sont meilleures avec le critère de distance cepstrale moyenne (pour lequel elles paraissent confondues avec celles du Masquage EFR) qu'avec le critère PESQ. Ceci tend à signifier que les dégradations apportées par l'algorithme AK1 (relativement à AK0) concernent principalement les zones de non-activité vocale puisque la distance cepstrale moyenne est restreinte aux instants d'activité vocale au contraire de la note PESQ. En considérant les exemples de signaux décodés illustrés Figure 4.11, il apparaît également que l'algorithme AK1 introduit plus de « bruit impulsif » dans les zones de silence en comparaison avec l'algorithme AK0. Un tel comportement plaide plutôt pour l'hypothèse d'un mauvais conditionnement du modèle AK1 appris sur notre base de données (puisque les zones de non-activité vocale ont été exclues de cette base d'apprentissage). Cette hypothèse sera validée par les résultats expérimentaux obtenus pour les algorithmes développés au Chapitre suivant.

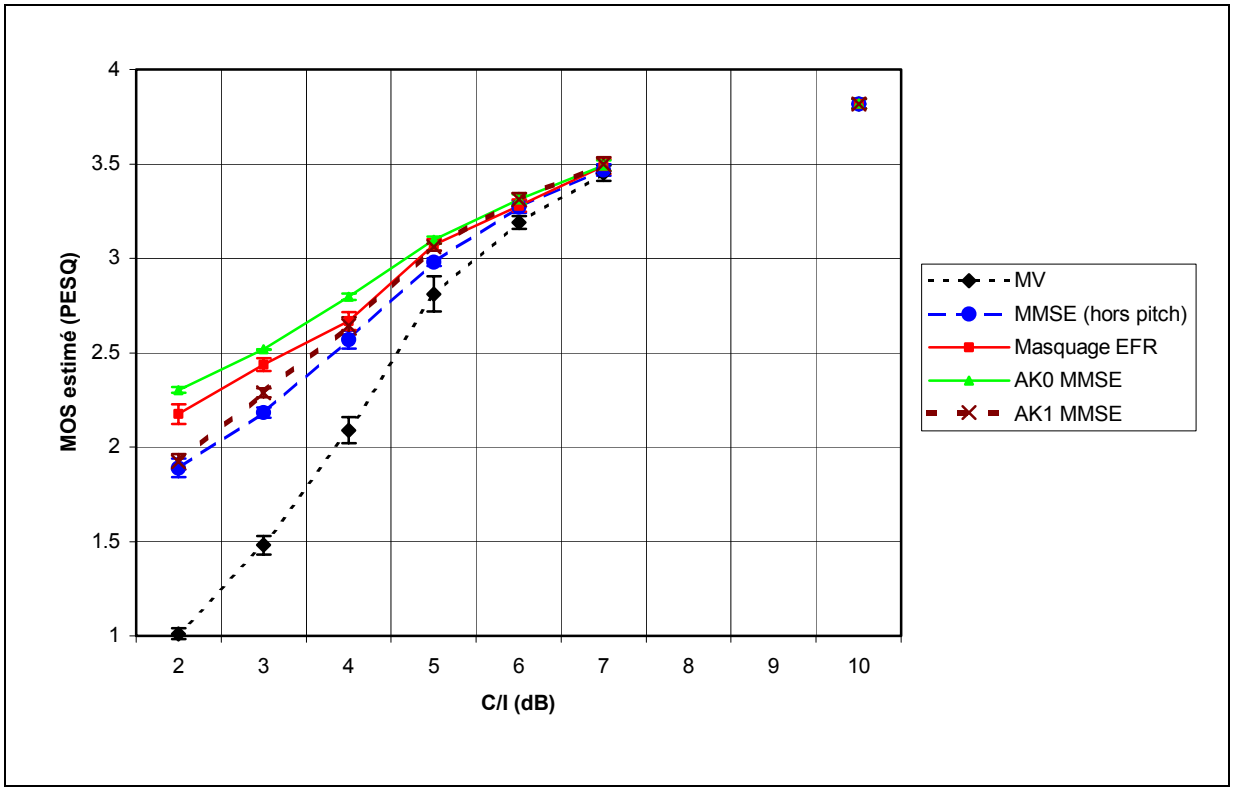

Figure 4.9 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

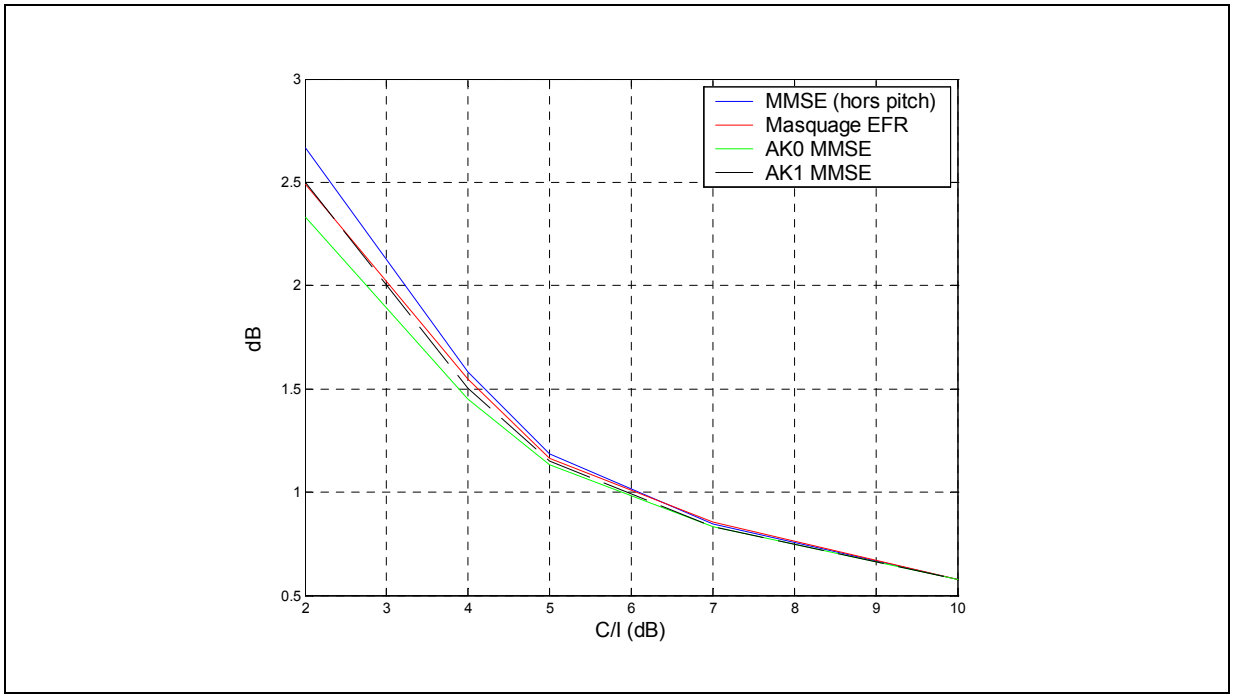

Figure 4.10 : Distance cepstrale moyenne sur les trames d'activité vocale en fonction du C/I

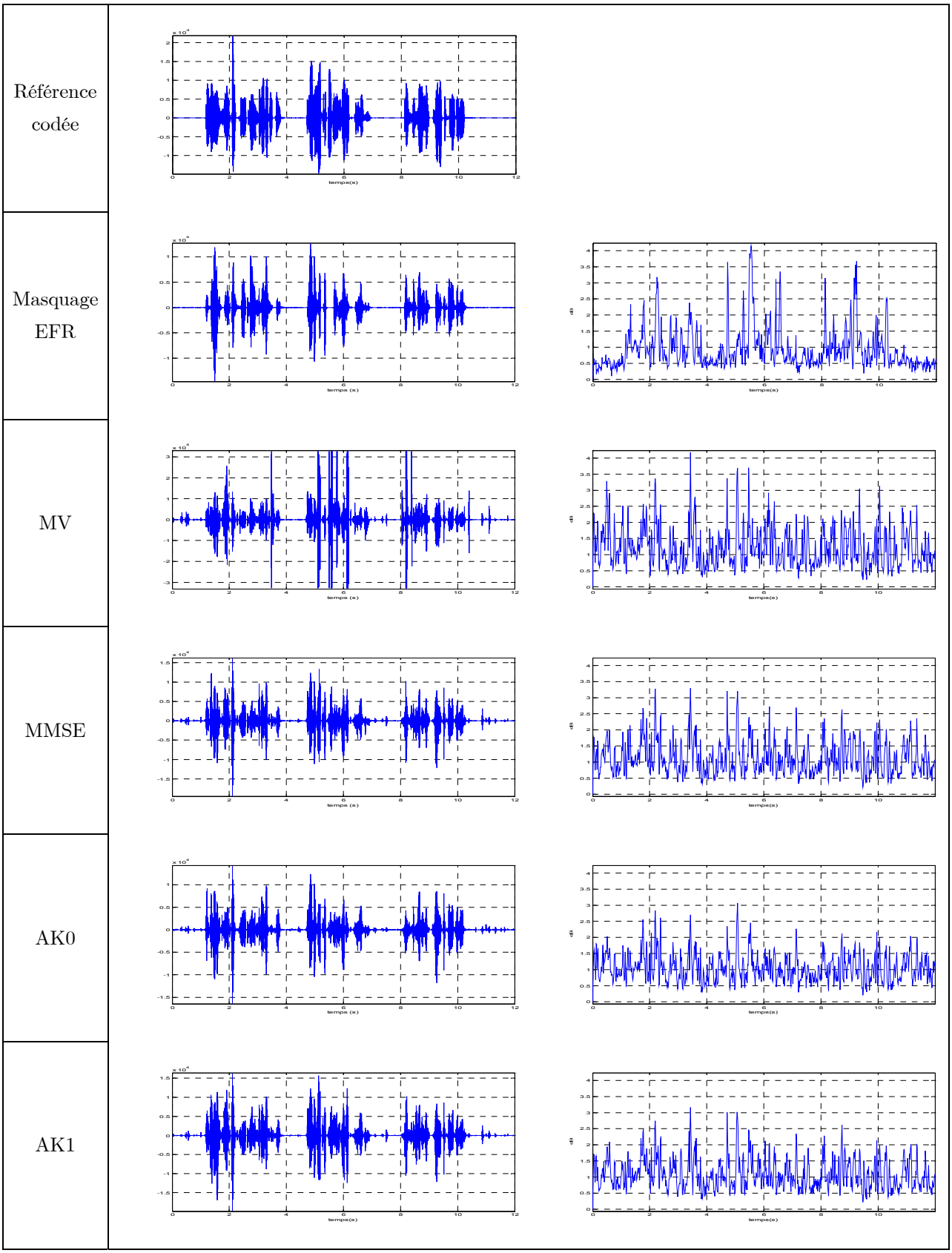

Figure 4.11 : Exemples de signaux décodés et distances cepstrales correspondantes ($C/I = 2$dB)

4.5 Conclusion

La mise en œuvre des algorithmes de décodage souple sur le codeur GSM EFR fait d'abord ressortir l'intérêt d'une *estimation souple* des paramètres tenant compte des probabilités d'erreur estimées par un décodeur canal à sortie souple. Elle permet de maintenir une « continuité » de signal à l'écoute tout en atténuant fortement les distorsions de parole. C'est cette estimation souple (critère MMSE) qui fournit l'essentiel du gain de qualité perçue par rapport à un décodage direct des paramètres reçus.

Il y a également une information à retirer de la redondance des paramètres du codeur EFR. La prise en compte de la non-uniformité de leur distribution apporte déjà un gain supplémentaire de qualité mais c'est au niveau de la corrélation inter-trame que l'on peut espérer les améliorations les plus importantes. En effet, le modèle de corrélation temporelle utilisé, malgré ses limitations, capture déjà une redondance significative des paramètres spectraux (LSF).

Cependant, l'exploitation de ce modèle par l'algorithme de décodage AK1 ne donne pas de résultats satisfaisants. La nécessité d'estimer de larges matrices de probabilités de transition pose un problème de conditionnement et donc de robustesse du décodeur face à des signaux de parole hors base d'apprentissage. D'autre part, si ce modèle permet de caractériser le niveau *moyen* de corrélation entre paramètres successifs, ceci n'implique pas qu'on puisse obtenir un gain de qualité *en moyenne* en l'utilisant pour la prédiction des paramètres du codeur. Cette hypothèse ne tient en effet pas compte de la non-stationnarité de ces paramètres pour la parole.

Chapitre 5

Décodage source à entrées souples : Etude de nouveaux algorithmes

5.1 Introduction

Les méthodes de décodage de parole à entrées souples étudiées aux chapitres 3 et 4 sont en réalité la simple transposition au domaine de la parole, de développements effectués dans un cadre bien plus général, qui est celui du codage et du décodage conjoint source - canal. En particulier, les probabilités de transition entre indices de quantification utilisées pour représenter la redondance des paramètres du codeur parole sont emprunts d'une approche « théorie de l'information » bien plus que du souci de modéliser l'objet spécifique qu'est la parole (ou ses paramètres). Il apparaît pourtant intéressant si l'on souhaite modéliser finement la redondance d'une source (ici, la parole) de se munir d'un modèle prenant en compte les spécificités de celle-ci. D'autre part, sur un plan strictement pratique, les méthodes proposées[50] se révèlent bien trop complexes pour une mise en œuvre sur des décodeurs tels le que celui du GSM EFR.

On propose ici d'apporter une réponse commune à ces deux limitations en se donnant un *modèle analytique* de la corrélation entre les paramètres du codeur de parole. Ce modèle est cherché en premier lieu pour des considérations de complexité puisque nous visons à l'appliquer à un système du type GSM. On verra cependant que la modélisation analytique qu'il fournit pour les paramètres du codeur correspond également à une modélisation plus « physique » de ces derniers et de leur comportement.

[50] On considère surtout ici la méthode de prédiction inter-trame AK1.

5.2 Réduction de la complexité

Les dictionnaires de quantification du codeur LSF comptent jusqu'à 512 éléments pour certains jeux de résidus de prédiction LSF et pour le délai de pitch. Les algorithmes AK1 et AK2 sont alors d'une complexité rédhibitoire pour une mise en œuvre pratique. Aussi nous recherchons une méthode de décodage souple de complexité réduite. Contrairement à la prédiction linéaire, cette méthode ne doit pas faire d'hypothèses sur les valeurs des paramètres déjà décodés, afin d'éviter tout risque de propagation d'erreur. Elle doit également permettre d'exploiter la redondance inter-trame (AK1) tout comme la redondance intra-trame (AK2).

C'est le calcul de la *probabilité a posteriori* qui représente la partie la plus complexe du décodeur souple. En reprenant les notations du paragraphe 3.3, les deux principaux facteurs de la complexité du calcul de cette probabilité *a posteriori* $p(i_n | j_n,...,j_1)$ sont les suivants :

- Le modèle a priori basé sur des probabilité de transition $p(i_n | i_{n-1})$ entre indices de quantification et qui exige de parcourir l'ensemble des éléments du dictionnaire pour calculer une « probabilité prédictive » $p(i_n | j_{n-1},...,j_1)$.

- La combinaison de la « probabilité prédictive » $p(i_n | j_{n-1},...,j_1)$ avec la vraisemblance a posteriori $p(j_n | i_n)$ qui doit être évaluée explicitement pour chacun des éléments du dictionnaire.

Ces deux éléments réunis expliquent la complexité en $O(N^2)$ du calcul de la probabilité *a posteriori* par **AK1**.

5.2.1 Recherche d'un modèle analytique

La première idée serait d'exploiter des expressions analytiques pour le modèle *a priori* et pour la vraisemblance. Ceci conduirait à une expression analytique de la *probabilité a posteriori* sans avoir à parcourir explicitement les éléments du dictionnaire. Le calcul des estimées MMSE ou MAP en serait lui-même simplifié. L'inspiration est ici l'algorithme de Kalman qui combine un modèle interne des données à estimer (modèle *a priori*) et un modèle de perturbation des données reçues (bruit additif gaussien).

Il y a deux possibilités pour rechercher une expression analytique de la vraisemblance et de la probabilité *a priori*, selon le domaine dans lequel on choisit de calculer ces probabilités :

- dans le domaine des indices de quantification i

- dans le domaine des paramètres (ou centroïdes $\mathbf{c}^{(i)}$)

L'indexation (*Index Assignement*) d'un paramètre quantifié $\mathbf{c}^{(i)}$ est rarement une transformation linéaire, hormis dans le cas d'une quantification scalaire où l'index i peut être simplement la valeur quantifiée du paramètre. Dans le cas d'une quantification vectorielle, comme c'est le cas pour les LSF dans l'EFR, cette linéarité est évidemment perdue. Ceci signifie que si une loi de type multi-gaussiennes peut être utilisée dans l'espace des *paramètres* pour modéliser la distribution *a priori*, on ne peut absolument pas en déduire un modèle analytique pour la loi *a priori* exprimée dans le domaine des *indices de quantification*. Ceci est illustré par l'exemple de la Figure 5.1 où le paramètre est un des résidus de prédiction des LSF.

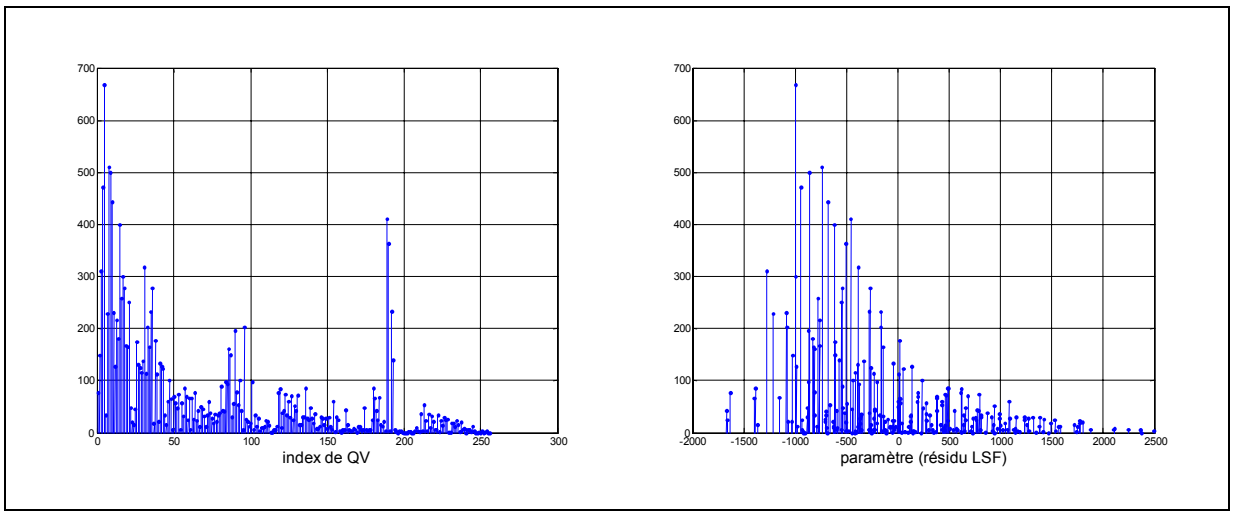

Figure 5.1 : Distribution *a priori* exprimée dans le domaine de l'index de quantification (gauche) ou selon les valeurs du paramètre associé (droite)

A l'inverse, si la sortie du SOVA peut être modélisée analytiquement (en l'assimilant à la sortie d'un *canal à bruit additif gaussien*[51] *CABG*), un modèle analytique de la vraisemblance n'est pas disponible dans le domaine des paramètres. En effet, il n'existe pas de transformation[52] *linéaire* permettant d'obtenir les centroïdes $\mathbf{c}^{(i)}$ à partir des bits codant l'indice i.

En conclusion, on ne peut faire l'économie du calcul explicite de la vraisemblance, néanmoins on peut modéliser analytiquement la loi *a priori* en se plaçant dans le *domaine des paramètres*. L'exploitation d'un tel modèle *a priori* pourrait permettre de réduire la complexité du calcul de la probabilité *a posteriori*. C'est cette approche que nous développons dans ce qui suit.

---

[51] Comme il est présenté en Annexe C, la sortie du SOVA peut être interprétée comme un CABG ou comme un canal binaire symétrique CBS de probabilités d'erreur *instantanées* connues. C'est cette seconde interprétation qu'on utilise dans tout ce document mais l'interprétation CABG aurait pu être intéressante pour en dériver un modèle analytique de la vraisemblance.

[52] Il est montré dans [Hedelin et al., 1995] que les centroïdes peuvent s'obtenir à partir des bits codant leur index de quantification i par l'intermédiaire d'une transformée de Hadamard et que celle-ci est généralement non-linéaire.

5.2.2 Modèle a priori dans le domaine des paramètres

On modélise désormais la connaissance *a priori* dans le domaine des paramètres à valeurs continues $\mathbf{v} \in R^d$, c'est-à-dire des paramètres *non quantifiés*. Ce choix permet d'utiliser des densités continues comme les multi-gaussiennes pour représenter la distribution *a priori*. En revanche, au niveau du décodeur souple, c'est toujours la probabilité *a posteriori* des paramètres *quantifiés* $\mathbf{c}^{(i)}$ (ou de manière équivalente, de leur index de quantification i) que l'on cherche à évaluer[53]. La première étape consiste donc à exprimer les probabilités *a priori* des index de quantification à partir du modèle *a priori* défini sur les paramètres. Pour préciser tout ceci, considérons les différentes formes de redondance, à savoir, non-uniformité, corrélation temporelle, et corrélation entre paramètres distincts :

- **Non uniformité :** Le modèle[54] *a priori* dont on dispose ici est la distribution du paramètre $p(\mathbf{v}_n|\lambda)$ supposée invariante au cours du temps n. On en déduit une loi *a priori* sur les index de quantification selon :

$$p(i_n = i|\lambda) = \int_{\mathbf{v}\in\Omega^{(i)}} p(\mathbf{v}_n = \mathbf{v}|\lambda)\,d\mathbf{v} \tag{5.1}$$

  où $\Omega^{(i)}$ désigne la *cellule de quantification* associée à la valeur i de l'index de quantification, c'est-à-dire l'ensemble des valeurs $\mathbf{v}$ telles que $Q(\mathbf{v}) = i$.

  Dans la pratique, on posera l'hypothèse simplificatrice suivante :

$$\begin{aligned} p(i_n = i|\lambda) &= \int_{\mathbf{v}\in\Omega^{(i)}} p(\mathbf{v}_n = \mathbf{v}|\lambda)\,d\mathbf{v} \\ &\simeq C\ r^d(i)\,p\left(\mathbf{v}_n = \mathbf{c}^{(i)}|\lambda\right) \end{aligned} \tag{5.2}$$

  où C est une constante de normalisation, d est la dimension du paramètre $\mathbf{v}$, et $r(i)$ est le « rayon moyen » de la cellule de quantification $\Omega^{(i)}$. Le rayon moyen $r(i)$ est estimé à partir de la base de donnée de parole, comme étant l'écart type de la distribution du paramètre $\mathbf{v}$ à l'intérieur de la cellule de quantification $\Omega^{(i)}$. Cette approximation revient à supposer la probabilité $p(\mathbf{v}_n|\lambda)$ constante à l'intérieur de la cellule de quantification.

- **Mémoire :** De même que précédemment, on se limitera à la corrélation entre 2 valeurs *successives* du paramètre. On modélise donc la loi jointe $p\left(\mathbf{v}_n, \mathbf{v}_{n-1}|\lambda\right)$. On en déduit :

---

[53] Ceci parce que les vraisemblances en sortie de canal équivalent sont définies sur les index de quantification transmis.

[54] On notera ici par λ, l'ensemble des paramètres définissant le modèle *a priori* utilisé.

$$\begin{aligned} p\big(i_n = i, i_{n-1} = i'|\lambda\big) &= \int_{\mathbf{v}\in\Omega^{(i)}} \int_{\mathbf{v}'\in\Omega^{(i')}} p\big(\mathbf{v}_n = \mathbf{v}, \mathbf{v}_{n-1} = \mathbf{v}'|\lambda\big) d\mathbf{v}\, d\mathbf{v}' \\ &\simeq C\ r^d(i) r^d\big(i'\big) p\Big(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')}|\lambda\Big) \end{aligned} \tag{5.3}$$

- **Corrélation entre paramètres $\mathbf{v}_{n,k}$ et $\mathbf{v}_{n,\ell}$:** Elle est modélisée par la loi jointe $p\Big(\mathbf{v}_{n,k}, \mathbf{v}_{n,\ell}\Big)$, on supposera de la même façon qu'on a la relation :

$$p\Big(i_{n,k} = i, i_{n,\ell} = i'|\lambda\Big) \simeq C\ r^d(i) r^d\big(i'\big) p\Big(\mathbf{v}_{n,k} = \mathbf{c}_k^{(i)}, \mathbf{v}_{n,\ell} = \mathbf{c}_\ell^{(i')}|\lambda\Big) \tag{5.4}$$

où $\mathbf{c}_k$ et $\mathbf{c}_\ell$ sont les éléments des dictionnaires de quantification des paramètres $\mathbf{v}_{n,k}$ et $\mathbf{v}_{n,\ell}$ respectivement.

5.2.3 Prédiction inter-trame par multi-gaussiennes

On peut maintenant exprimer les probabilités *a posteriori* des index de quantification émis à l'instant n à partir du modèle *a priori* sur les paramètres. On considèrera en premier lieu le cas de la *prédiction inter-trame*[55]. La probabilité *a posteriori* se décompose alors de la façon suivante :

$$\begin{aligned} p\big(i_n\big|j_n,...,j_1\big) &= C\ p\big(i_n, j_n,...,j_1\big) \\ &= C\ p\big(j_n|i_n\big) p\big(i_n, j_{n-1},...,j_1\big) \\ &= C'\ p\big(j_n|i_n\big) p\big(i_n\big|j_{n-1},...,j_1\big) \end{aligned} \tag{5.5}$$

où C et C' sont des constantes de normalisation.

Le premier facteur correspond aux probabilités de transition $p\big(j_n|i_n\big)$ du canal équivalent évaluées d'après la relation **(3.6)**. Le second facteur $p\big(i_n\big|j_{n-1},...,j_1\big)$ est la « *probabilité prédictive* » de i_n sachant $j_{n-1},...,j_1$. On s'intéresse dans ce qui suit au calcul de cette probabilité prédictive.

Comme on modélise uniquement la corrélation temporelle à l'ordre 1, c'est-à-dire entre trames *successives* $n-1$ et n, la probabilité prédictive s'exprime de façon analogue au second membre de la *variable d'induction avant* (3.15) dans un treillis :

$$\begin{aligned} p\big(i_n = i\big|j_{n-1},...,j_1,\lambda\big) &= \sum_{i'} p\big(i_n = i\big|i_{n-1} = i', j_{n-1},...,j_1,\lambda\big) p\big(i_{n-1} = i'\big|j_{n-1},...,j_1,\lambda\big) \\ &= \sum_{i'} p\big(i_n = i\big|i_{n-1} = i',\lambda\big) p\big(i_{n-1} = i'\big|j_{n-1},...,j_1,\lambda\big) \end{aligned} \tag{5.6}$$

On introduit maintenant la probabilité *a priori* définie sur les paramètres. D'après (5.3), il vient :

[55] La prédiction inter-trame modélise aussi implicitement la non-uniformité du paramètre.

$$p\big(i_n = i \big| i_{n-1} = i' | \lambda\big) = \frac{p\big(i_n = i, i_{n-1} = i' | \lambda\big)}{p\big(i_{n-1} = i' | \lambda\big)} \simeq C\ r^d(i) \frac{p\Big(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)}{p\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)} \tag{5.7}$$

où C est une constante de normalisation et $p\big(\mathbf{v}_{n-1} | \lambda\big)$ est la loi marginale associée à $p\big(\mathbf{v}_n, \mathbf{v}_{n-1} | \lambda\big)$.

On obtient finalement une expression définissant la *probabilité prédictive* $p\big(i_n \big| j_{n-1}, ..., j_1, \lambda\big)$ à partir de la *probabilité a posteriori* $p\big(i_{n-1} \big| j_{n-1}, ..., j_1, \lambda\big)$ à l'instant $n-1$ et de la loi jointe *a priori* $p\big(\mathbf{v}_n, \mathbf{v}_{n-1} | \lambda\big)$ apprise sur les paramètres, ainsi que la loi marginale qui s'en déduit $p\big(\mathbf{v}_{n-1} | \lambda\big)$:

$$p\big(i_n = i \big| j_{n-1}, ..., j_1, \lambda\big) \simeq C\ r^d(i) \sum_{i'} \frac{p\Big(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)}{p\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)} p\big(i_{n-1} = i' \big| j_{n-1}, ..., j_1, \lambda\big) \tag{5.8}$$

Jusqu'ici le calcul de la probabilité *a posteriori* est strictement identique à celui proposé par [Fingscheidt et al., 1997] et l'on s'est contenté d'un jeu de substitution d'une loi *a priori* sur les indices de quantification par une loi *a priori* sur les paramètres. L'intérêt de la formulation présente apparaît lorsqu'on choisit de modéliser la probabilité *a priori* $p\big(\mathbf{v}_n, \mathbf{v}_{n-1} | \lambda\big)$ par un *mélange de gaussiennes*. Ceci permet à la fois de modéliser de manière compacte n'importe quelle forme de distribution et de réduire la complexité puisque seuls les paramètres définissant les gaussiennes sont à actualiser dans la récursion (5.8) définissant $p\big(i_n \big| j_{n-1}, ..., j_1, \lambda\big)$.

5.2.3.1 Modèle multi-gaussien

Nous modélisons la loi jointe $p\big(\mathbf{v}_n, \mathbf{v}_{n-1} | \lambda\big)$ par un *mélange de gaussiennes* GMM définies sur $R^d \times R^d$:

$$\begin{aligned} &p\big(\mathbf{v}_n, \mathbf{v}_{n-1} | \lambda\big) = \sum_{m=1}^{K} w_m g_m\big([\mathbf{v}_n, \mathbf{v}_{n-1}]\big) \\ &\text{avec } g_m(\mathbf{X}) = N(\mathbf{X}, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m),\ \mathbf{X} \in R^d \times R^d \end{aligned} \tag{5.9}$$

où w_m sont les poids des gaussiennes $N(\mathbf{X}, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ de moyenne $\boldsymbol{\mu}_m$ et de matrice de covariance $\boldsymbol{\Sigma}_m$, et K est le nombre de gaussiennes utilisées. On note $\lambda = \{K, w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ l'ensemble de ces paramètres spécifiant la GMM.

On choisit ici des gaussiennes de **matrice de covariance diagonale**. L'hypothèse de covariance diagonale permet à la fois d'éviter les problèmes de conditionnement lors de l'apprentissage et d'exprimer chaque gaussienne g_m comme le produit des lois *marginales* selon $\mathbf{v}_n$ et $\mathbf{v}_{n-1}$:

$$g_m\big([\mathbf{v}_n, \mathbf{v}_{n-1}]\big) = N\big(\mathbf{v}_n, \boldsymbol{\mu}_m^{(0)}, \boldsymbol{\Sigma}_m^{(0)}\big) N\big(\mathbf{v}_{n-1}, \boldsymbol{\mu}_m^{(1)}, \boldsymbol{\Sigma}_m^{(1)}\big) \tag{5.10}$$

où les indices 0 et 1 indiquent la restriction à l'espace des $\mathbf{v}_n$ et des $\mathbf{v}_{n-1}$ respectivement. Les matrices de covariances $\boldsymbol{\Sigma}_m^{(0)}$ et $\boldsymbol{\Sigma}_m^{(1)}$ étant elles mêmes diagonales.

Compte tenu du modèle multi-gaussien choisi (5.9), la probabilité *a posteriori* $p\big(i_n \big| j_{n-1},...,j_1,\lambda\big)$ peut se ré-écrire :

$$p\big(i_n = i \big| j_{n-1},...,j_1,\lambda\big) = C\ r^d(i) \sum_{i'} \sum_m w_m g_m\Big(\big[\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')}\big]\Big) \frac{p\big(i_{n-1} = i' \big| j_{n-1},...,j_1,\lambda\big)}{p\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)} \quad (5.11)$$

La factorisation (5.10) des gaussiennes individuelles[56] g_m permet de simplifier cette expression et d'aboutir à une *probabilité prédictive* sous la forme d'un *mélange de gaussiennes* :

$$p\big(i_n = i \big| j_{n-1},...,j_1,\lambda\big) = C\ r^d(i) \sum_m w'_m N\Big(\mathbf{v}_n = \mathbf{c}^{(i)}, \boldsymbol{\mu}_m^{(0)}, \boldsymbol{\Sigma}_m^{(0)}\Big) \quad (5.12)$$

avec :
$$w'_m = w_m \left(\sum_{i'} N\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')}, \boldsymbol{\mu}_m^{(1)}, \boldsymbol{\Sigma}_m^{(1)}\Big) \frac{p\big(i_{n-1} = i' \big| j_{n-1},...,j_1,\lambda\big)}{p\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big)} \right) \quad (5.13)$$

5.2.3.2 Interprétation de la modélisation proposée

Les expressions (5.12) et (5.13) donnent le détail du calcul de la loi prédictive à partir de la loi multi-gaussienne utilisée pour modéliser la connaissance *a priori*. On peut cependant les reformuler de manière à en dégager une interprétation plus intuitive. Pour cela, on remarque que l'on a :

$$p\big(i_{n-1} = i' | g_m\big) \simeq C\ r^d\big(i'\big) N\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')}, \boldsymbol{\mu}_m^{(1)}, \boldsymbol{\Sigma}_m^{(1)}\Big) \quad (5.14)$$

$$p\big(i_{n-1} = i' | \lambda\big) \simeq C\ r^d\big(i'\big) p\Big(\mathbf{v}_{n-1} = \mathbf{c}^{(i')} | \lambda\Big) \quad (5.15)$$

$$w_m = p(g_m | \lambda) \quad (5.16)$$

Après quelques manipulations, la relation (5.13) de mise à jour du poids des gaussiennes s'écrit :

$$\begin{aligned} w'_m &= \sum_{i'} p\big(g_m \big| i_{n-1} = i'\big) p\big(i_{n-1} = i' \big| j_{n-1},...,j_1,\lambda\big) \\ &= p\big(g_m \big| j_{n-1},...,j_1,\lambda\big) \end{aligned} \quad (5.17)$$

et la probabilité prédictive (5.12) prend la forme suivante :

$$p\big(i_n = i \big| j_{n-1},...,j_1,\lambda\big) = \sum_m w'_m p(i_n = i | g_m) \quad (5.18)$$

Les gaussiennes définissent un ensemble de « *classes* » dans l'espace joint $\big(\mathbf{v}_n, \mathbf{v}_{n-1}\big)$. Comme elles sont de *covariance diagonale*, la distribution de l'index i_n (ou de manière équivalente, des centroïdes

[56] La loi jointe obtenue par mélange des gaussiennes individuelles n'est pas factorisable en elle-même.

émis à l'instant n) s'obtient à partir du mélange des lois *projetées* sur l'axe $\mathbf{v}_n$ des gaussiennes individuelles.

Ainsi, la corrélation temporelle avec l'index i_{n-1} n'est pas modélisée directement par une loi de transition $p(i_n|i_{n-1})$ mais indirectement par la probabilité $p(g_m|i_{n-1})$ d'être dans la « classe » associée à la gaussienne g_m sachant l'index précédent i_{n-1}.

Aucune hypothèse n'est faite sur les données précédemment décodées puisqu'on utilise la probabilité *a posteriori* de l'index i_{n-1} (calculée à l'itération précédente), conjointement avec la probabilité $p(g_m|i_{n-1})$ pour estimer la probabilité $p(g_m|j_{n-1},...,j_1)$ d'être dans la « classe » définie par la gaussienne g_m sachant les données reçues $j_{n-1},...,j_1$. C'est cette probabilité qui est utilisée comme *pondération* des gaussiennes dans le mélange (5.18) donnant la distribution de l'index i_n.

La Figure 5.2 compare l'approche proposée avec celle utilisant les probabilités de transition $p(i_n|i_{n-1})$ entre index de quantification. On y a représenté l'espace joint formé par les index de quantification (resp. centroïdes) émis à l'instant n et $n-1$. Le calcul de la probabilité prédictive $p(i_n = i|j_{n-1},...,j_1)$ est schématiquement décomposé de manière à faire apparaître les relations entre les probabilités calculés et les couples (i_n, i_{n-1}) (resp. $(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')})$).

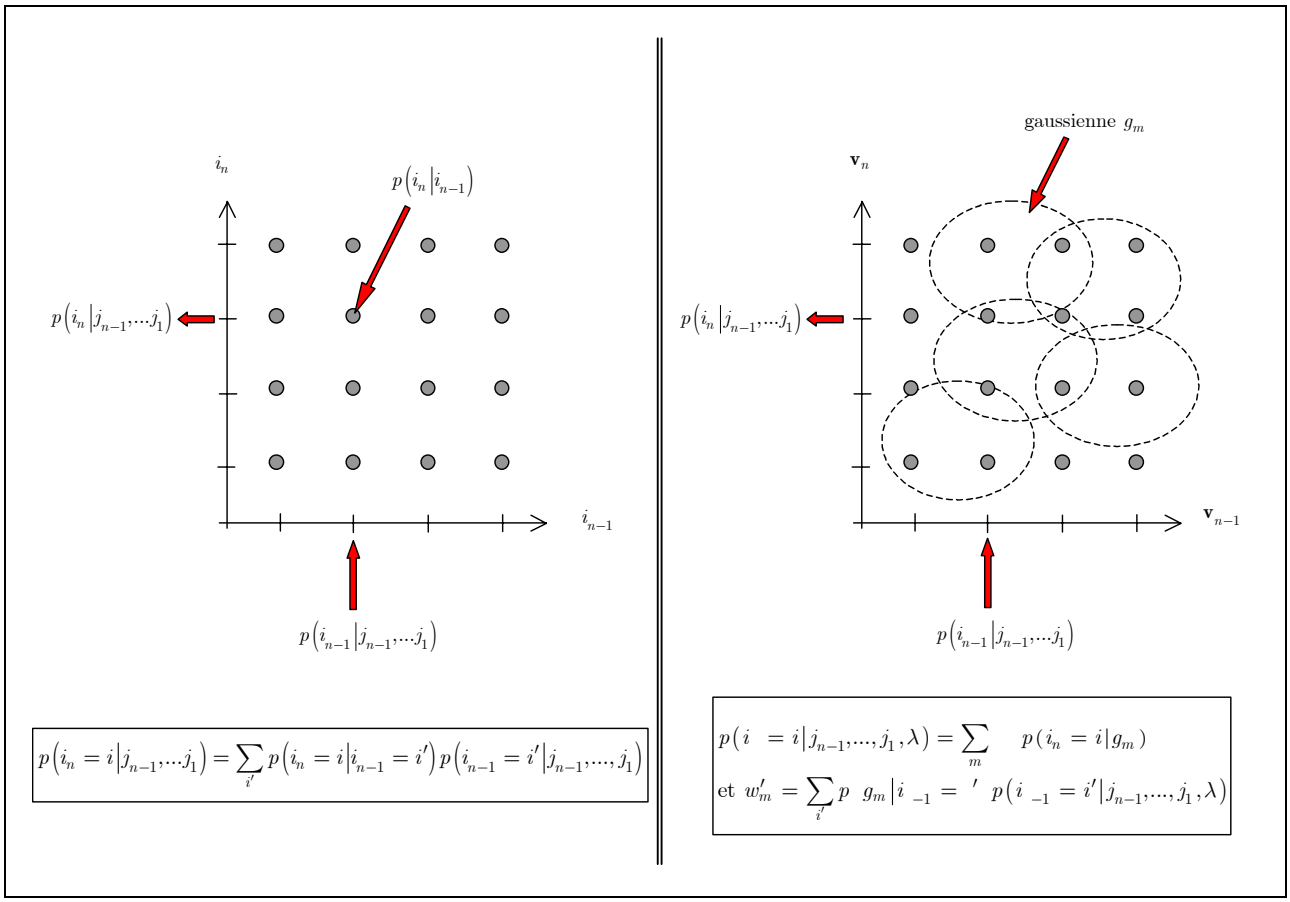

Figure 5.2 : Comparaison du calcul de la probabilité prédictive

On voit que la modélisation par un mélange de gaussiennes s'interprète comme un sous-échantillonnage de l'espace des couples $\left(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')}\right)$ en « classes » à l'intérieur desquelles la distribution des points $\left(\mathbf{v}_n = \mathbf{c}^{(i)}, \mathbf{v}_{n-1} = \mathbf{c}^{(i')}\right)$ est décrite par une gaussienne. C'est ce sous-échantillonnage qui permet une réduction de la complexité du calcul de la probabilité *a posteriori* et du stockage de la loi *a priori*.

5.2.3.3 Complexité du calcul de la probabilité a posteriori

La probabilité *a posteriori* $p(i_n = i | j_n, ..., j_1)$ peut finalement être calculée à partir de la récursion définie par les équations (5.5), (5.12) et (5.13). La Figure 5.3 représente le diagramme synoptique de cette récursion dans le cas de la prédiction inter-trame.

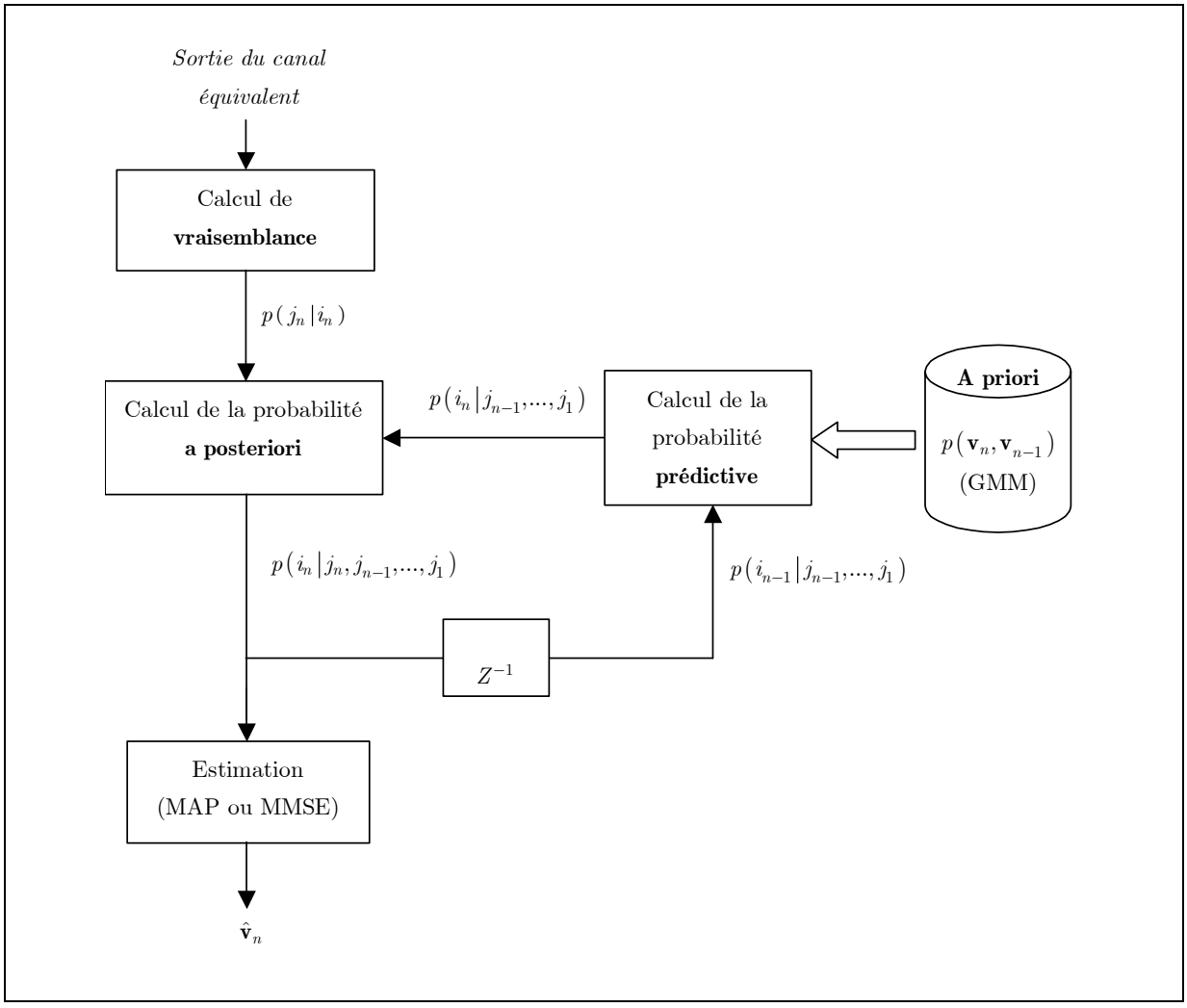

Figure 5.3 : Algorithme de décodage avec prédiction inter-trame(AK1 GMM)

La complexité totale du calcul de la probabilité *a posteriori* est de l'ordre $O(KN)$ où K est le nombre de gaussiennes utilisée dans la loi *a priori*. La réduction de complexité par rapport au schéma de

prédiction inter-trame proposé par [Fingscheidt et al., 1997] est ainsi d'un facteur d'ordre $O\left(\frac{N}{K}\right)$. Ceci correspond au facteur de « sous-échantillonnage » du dictionnaire de quantification opéré par les multi-gaussiennes. La démarche proposé n'a donc d'intérêt que pour $N >> K$, ce qui sera le cas pour les LSF dont les dictionnaires de quantification ont une dimension importante.

5.2.4 Prédiction intra-trame par multi-gaussiennes

On peut étendre ce principe de modélisation de la distribution *a priori* par un mélange de gaussiennes au cas de la prédiction entre paramètres distincts au sein d'une même trame. En reprenant les notations précédemment utilisées, considérons la *trame* $\mathbf{I}_n = \left(i_{n,1},...,i_{n,k},...,i_{n,L}\right)$ des indices de quantification en sortie du codeur à l'instant n, et la trame $\mathbf{J}_n = \left(j_{n,1},...,j_{n,k},...,j_{n,L}\right)$ des indices reçus en sortie du canal équivalent. On modélise la corrélation intra-trame *au niveau des paramètres* associés $\mathbf{V}_n = \left(\mathbf{v}_{n,1},...,\mathbf{v}_{n,k},...,\mathbf{v}_{n,L}\right)$. Afin de limiter la complexité, on se limite à la corrélation entre couples de paramètres $\left(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\right)$, représentée par la loi jointe $p\left(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\,|\lambda\right)$. Cette loi jointe est définie par un mélange de gaussiennes g_m de matrice de covariances diagonales, de manière similaire à (5.9).

La prédiction intra-trame est modélisée par l'intermédiaire de la *probabilité* $p\left(i_{n,k}\middle|j_{n,\ell}\right)$ de l'index $i_{n,k}$ associé au paramètre $\mathbf{v}_{n,k}$ sachant l'index *reçu* à l'instant n pour le paramètre $\mathbf{v}_{n,\ell}$. On ne pose donc aucune hypothèse sur la valeur du paramètre décodé $\hat{\mathbf{v}}_{n,\ell}$. On a :

$$p\left(i_{n,k}\middle|j_{n,l}\right) = C\ p\left(j_{n,l}\middle|i_{n,k}\right)p\left(i_{n,k}\right) \tag{5.19}$$

où C est une constante de normalisation. La probabilité $p\left(i_{n,k}\right)$ représente la connaissance *a priori* que l'on a de $i_{n,k}$ indépendamment de la prédiction intra-trame, il peut s'agir soit d'une distribution invariante au cours du temps (prise en compte de la non-uniformité uniquement), soit de la probabilité *a posteriori* $p\left(i_{n,k}\middle|j_{n,k},...,j_{1,k}\right)$ issue de la prédiction inter-trame. Ceci sera précisé par la suite.

On peut décomposer la vraisemblance $p\left(j_{n,\ell}\middle|i_{n,k}\right)$ de la façon suivante :

$$p\left(j_{n,\ell}\middle|i_{n,k}\right) = \sum_{i'} p\left(j_{n,\ell}\middle|i_{n,\ell}=i'\right)p\left(i_{n,\ell}=i'\middle|i_{n,k}\right) \tag{5.20}$$

Cette expression qui fait intervenir les probabilités de transition $p\left(j_{n,\ell}\,|i_{n,\ell}\right)$ du canal équivalent et les probabilités de transitions entres indices de quantification $p\left(i_{n,\ell}\,|i_{n,k}\right)$ correspond à la relation (3.30) définissant la variable d'induction latérale pour la prédiction intra-trame (limitée à l'ordre 1), utilisée par [Lahouti et al., 2001].

De même que pour la prédiction inter-trame, on substitue les probabilités de transition entre indices $p\left(i_{n,\ell}\,|i_{n,k}\right)$ par la loi jointe sur les paramètres $p\left(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\,|\lambda\right)$ selon la relation (5.4). La loi

$p\big(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\,|\lambda\big)$ étant un mélange de gaussiennes de *covariances diagonales*, on aboutit finalement à l'expression de la vraisemblance $p\big(j_{n,\ell}\big|i_{n,k}\big)$ comme un mélange de gaussiennes :

$$p\big(j_{n,\ell}\big|i_{n,k}=i,\lambda\big)=C\sum_m w'_m\,\frac{N\big(\mathbf{v}_{n,k}=\mathbf{c}_k{}^{(i)},\boldsymbol{\mu}_m^{(0)},\boldsymbol{\Sigma}_m^{(0)}\big)}{p\big(\mathbf{x}_{n,k}=\mathbf{c}_k{}^{(i)}\,|\lambda\big)} \tag{5.21}$$

avec :
$$w'_m=w_m\left(\sum_{i'}r^d\,(i')\,N\big(\mathbf{x}_{n,\ell}=\mathbf{c}_\ell{}^{(i')},\boldsymbol{\mu}_m^{(1)},\boldsymbol{\Sigma}_m^{(1)}\big)\,p\big(j_{n,\ell}\big|i_{n,\ell}=i'\big)\right) \tag{5.22}$$

En faisant apparaître explicitement les gaussiennes g_m dans ces équations, on aboutit à une interprétation similaire à celle de la prédiction inter-trame :

$$p\big(j_{n,\ell}\big|i_{n,k}=i,\lambda\big)=C\sum_m p\big(j_{n,\ell}\,|g_m\big)\,p\big(g_m\,|i_{n,k}=i\big) \tag{5.23}$$

avec :
$$p\big(j_{n,\ell}\,|g_m\big)=\sum_{i'}p\big(j_{n,\ell}\big|i_{n,\ell}=i'\big)p\big(i_{n,\ell}=i'|g_m\big) \tag{5.24}$$

La corrélation intra-trame est modélisée uniquement au travers des gaussiennes g_m qui définissent des « classes » dans l'espace joint $\big(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\big)$. La vraisemblance $p\big(j_{n,\ell}\big|i_{n,k}\big)$ s'obtient comme la somme des vraisemblances de l'indice $i_{n,k}$ pour chaque « classe » (gaussienne g_m) pondérée par la vraisemblance de la classe (gaussienne g_m) pour l'indice reçu $j_{n,\ell}$. Cette classification et le « sous-échantillonnage » qu'elle définit, ne sont aisément réalisables que dans l'espace des paramètres. Ici aussi, la complexité du calcul de la vraisemblance $p\big(j_{n,\ell}\big|i_{n,k}\big)$ est de l'ordre de $O(KN)$ où K est le nombre de gaussiennes utilisée dans la loi *a priori*.

5.2.5 Combinaison avec la prédiction inter-trame

Comme on l'a vu au Chapitre 3, la prédiction inter-trame peut être combinée avec la prédiction intra-trame. Considérons le calcul de la *probabilité a posteriori* $p\big(i_{n,k}\big|\mathbf{J}_1,...,\mathbf{J}_n\big)$ de l'index $i_{n,k}$ sachant les trames reçues à l'instant n et aux instants précédents. En négligeant la corrélation *temporelle* entre paramètres *distincts*, on retrouve une expression similaire à la probabilité (3.31) :

$$p\big(i_{n,k}=i\big|\mathbf{J}_n,...,\mathbf{J}_1\big)=p\Big(i_{n,k}\Big|\big\{j_{n,\ell}\big\}_{\ell\neq k},j_{n,k},j_{n-1,k},...,j_{1,k}\Big)=C\prod_{\ell\neq k}p\big(j_{n,\ell}\big|i_{n,k}\big)p\big(i_{n,k}\big|j_{n,k},j_{n-1,k},...,j_{1,k}\big) \tag{5.25}$$

Autrement dit, la probabilité *a posteriori* $p\big(i_{n,k}\big|\mathbf{J}_1,...,\mathbf{J}_n\big)$ est le produit de la probabilité *a posteriori* issue de la prédiction *inter-trame* $p\big(i_{n,k}\big|j_{n,k},...,j_{1,k}\big)$ et des vraisemblances $p\big(i_{n,\ell}\,|i_{n,k}\big)$ calculées d'après les corrélations *intra-trame* entre couples $\big(\mathbf{v}_{n,k},\mathbf{v}_{n,\ell}\big)$. Le schéma du décodeur souple exploitant la corrélation inter et intra-trame est représenté Figure 5.4.

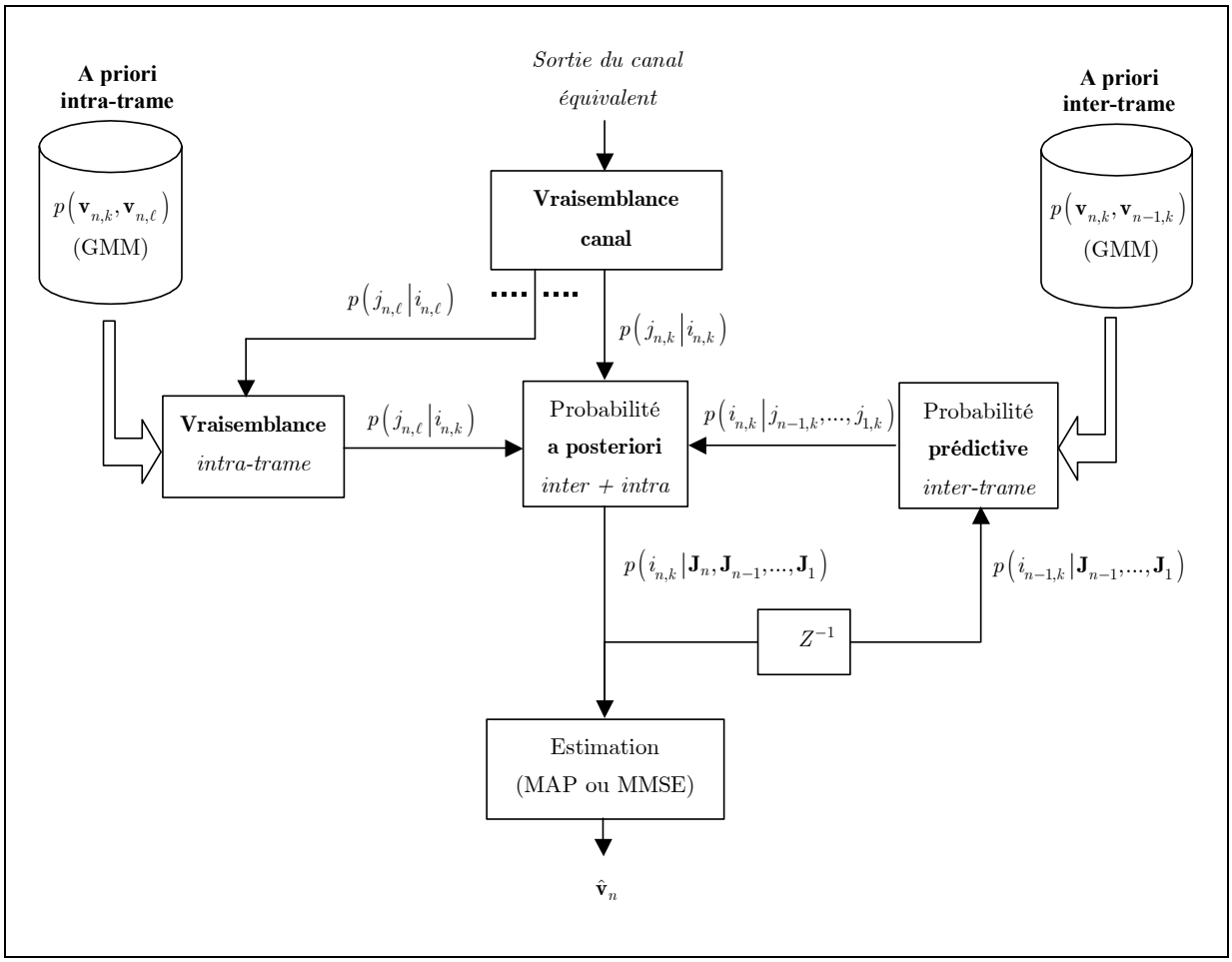

Figure 5.4 : Algorithme de décodage avec prédictions inter-trame et intra-trame (AK2 GMM)

5.3 Mise en oeuvre des modèles proposés

5.3.1 Apprentissage du modèle multi-gaussien

Pour mettre en œuvre les algorithmes proposés, nous devons au préalable définir et effectuer l'apprentissage des modèles multi-gaussiens des paramètres du codeur EFR. Comme on l'a vu on modélise la distribution de probabilité de couples de paramètres $(\mathbf{v}, \mathbf{v}')$ où $\mathbf{v}'$ désigne soit la valeur précédente du paramètre $\mathbf{v}$ (*corrélation inter-trame*), soit un paramètre distinct de la même trame (*corrélation intra-trame*). Les couples de paramètres modélisés sont reportés sur le Tableau 5.1 où

l'indice temporel n réfère à la trame dans le cas des LSF et à la sous-trame pour les autres paramètres[57].

Paramètres	Corrélation inter-trame	Corrélation intra-trame
LSF	$\left(LSF_{n,k}, LSF_{n,k+1}, LSF_{n+1,k}, LSF_{n+1,k+1}\right), k = 1,3,...,9$	$\left(LSF_{n,k}, LSF_{n,k+1}\right), k = 2,4,...,8$
Gain code fixe	$\left(gc_n, gc_{n-1}\right)$	
Gain code adaptatif	$\left(gp_n, gp_{n-1}\right)$	
Délai de pitch	$\left(lag_n, lag_{n-2}\right), n = 1,3$	

Tableau 5.1 : Couples de paramètres du codeur EFR modélisés par GMM

Les paramètres modélisés ici sont les paramètres *non quantifiés*, ceci pour faciliter la convergence de l'apprentissage des lois multi-gaussiennes. Plus précisément :

a) $LSF_{n,k}$ désigne la $k^{ième}$ LSF définie sur les sous-trames 3 et 4 de la trame n puisque 2 jeux de LSF sont calculés par trame. Comme les index de quantification transmis sont associés à des *paires* de LSF, on utilise le même appariement $\left(LSF_{n,k}, LSF_{n,k+1}, LSF_{n+1,k}, LSF_{n+1,k+1}\right)$ avec $k = 1,3,...,9$ pour modéliser la corrélation inter-trame. Ceci permet de modéliser la relation d'ordre entre LSF au sein d'une même paire conjointement avec la corrélation temporelle[58]. Dès lors, il est seulement nécessaire de modéliser la corrélation entre LSF adjacentes de paires distinctes $\left(LSF_{n,k}, LSF_{n,k+1}\right)$ avec $k = 2,4,...,8$ pour prendre en compte l'intégralité de la corrélation *intra-trame*.

b) gc_n désigne le *résidu* de la prédiction MA du gain de dictionnaire fixe[59] effectuée au codeur, pour la sous-trame n. Ici, la distinction entre corrélation inter-trame et intra-trame n'a plus lieu d'être puisqu'elles recouvrent toutes deux la *corrélation temporelle* entre sous-trames.

c) gp_n et lag_n sont respectivement le gain de dictionnaire adaptatif et le délai de pitch pour la sous-trame n. Ici également, on modélise la corrélation temporelle entre sous-trames consécutives (sous-trames 1 et 3 dans le cas du délai de pitch puisque les sous-trames 2 et 4 sont codées en différentiel).

---

[57] On modélise la corrélation temporelle entre trames pour les LSF et la corrélation temporelle entre sous-trames pour les autres paramètres.

[58] Une manière optimale de modéliser simultanément corrélation temporelle et relation d'ordre des LSF (corrélation intra-trame) serait de modéliser par GMM la distribution jointe des vecteurs LSF de deux trames consécutives. Ceci n'est pas envisageable dans la pratique pour des raisons de complexité.

[59] On rappelle que cette prédiction MA s'effectue sur le gain exprimé dans le domaine logarithmique.

5.3.1.1 Le choix d'un domaine pour modéliser la redondance

On remarquera que dans le cas des LSF, les paramètres modélisés ne sont pas exactement les paramètres qui sont quantifiés au niveau du codeur. En effet, le codeur de parole prend déjà en compte, bien qu'imparfaitement, la corrélation temporelle entre LSF en effectuant une prédiction MA d'ordre 1 et ce sont les résidus de prédiction qui sont quantifiés et transmis. Puisqu'on cherche à modéliser la redondance laissée par le codeur de parole, il peut sembler plus logique de la modéliser à partir du signal résiduel de la prédiction MA plutôt qu'à partir des LSF elles-mêmes. Cependant, un point important doit être considéré, l'algorithme de décodage souple limite la prédiction temporelle à *l'ordre 1* pour des raisons de complexité. Dès lors, le domaine le mieux adapté pour la modélisation par GMM est celui dans lequel un modèle prédictif d'ordre 1 suffit pour capturer la redondance non modélisée par le codeur de parole. Ce n'est pas forcément le domaine du signal résiduel en sortie de la prédiction MA. Ceci est illustré par la Figure 5.5 représentant l'évolution d'une LSF au cours du temps et le signal résultant de la prédiction MA. Il apparaît que la redondance restante dans le signal en sortie de prédiction MA n'est pas associée à une corrélation à très court-terme (ordre 1) mais plutôt à moyen-terme (fluctuations autour d'une moyenne). Un prédicteur d'ordre supérieur à 1 serait alors nécessaire pour extraire cette redondance. A l'inverse, une prédiction AR d'ordre 1 sur les LSF permettrait, par exemple, une meilleure prise en compte de la corrélation temporelle que la prédiction MA utilisée par le codeur (et donc une modélisation de la redondance laissée par le codeur). L'algorithme que nous proposons d'utiliser peut être vu comme une prédiction non-linéaire à l'ordre 1 puisque n'importe quelle forme de distribution des couples $\left(LSF_n, LSF_{n-1}\right)$ peut être modélisée par les multi-gaussiennes. Dès lors, sa « puissance de prédiction » ne dépend que du nombre de gaussiennes utilisées et l'on peut parfaitement modéliser la redondance non capturée par la prédiction MA dans le domaine des LSF.

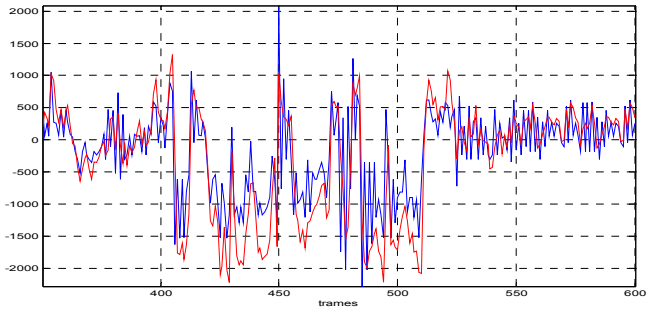

Figure 5.5 : Trajectoire temporelle d'une LSF (rouge) et du résidu de prédiction MA (bleu)

L'approche ainsi choisie nécessite de re-synthétiser les LSF lors du calcul des probabilités *a posteriori* (5.5). Ceci ne pose pas de problèmes car la prédiction MA est d'ordre 1 et qu'à un index de quantification correspond deux valeurs successives de LSF, c'est-à-dire qu'on peut associer une valeur de LSF à chaque valeur d'index de quantification.

Les même arguments pourraient être employés pour le gain de dictionnaire fixe mais dans ce cas, la prédiction MA effectuée par le codeur parole est d'ordre 4 et un index de quantification est associé à chaque valeur de la suite temporelle du résidu de prédiction (quantification scalaire). Il n'est alors pas possible de re-synthétiser la valeur du gain lors du calcul (5.5) sans faire d'hypothèses sur les valeurs précédemment décodées du gain, ce qui comporte le risque d'une propagation d'erreur.

5.3.1.2 Résultats de l'apprentissage

L'apprentissage des lois multi-gaussiennes est effectué à partir de la base de parole présentée au Tableau 4.1. L'algorithme de la K-moyenne présenté en Annexe A est utilisé pour initialiser les paramètres de poids w_m, de moyennes $\boldsymbol{\mu}_m$ et de covariances $\boldsymbol{\Sigma}_m$ des multi-gaussiennes. L'estimation finale des ces paramètres est ensuite réalisée par l'algorithme EM [Hedelin et al., 2000] qui recherche, par étapes successives[60] k, un maximum local de la vraisemblance $p\left(\mathbf{X}_1^L \middle| \lambda_k\right)$ évaluée sur la base d'apprentissage $\mathbf{X}_1^L = \left\{\mathbf{x}_1, ..., \mathbf{x}_L\right\}$.

En revanche, le nombre de gaussiennes K doit être fixé *a priori*. Ce choix est un des points critiques de la modélisation par multi-gaussiennes et peut être rapproché du problème du choix de l'ordre d'un modèle AR. Il résulte d'un compromis entre la *maximisation* de la vraisemblance du modèle et la *complexité* avec les K croissants (alliée au risque d'un *sur-apprentissage*). Un critère classiquement utilisé pour résoudre ce compromis est le Minimum Description Length MDL [Rissanen, 1978] :

$$\varphi(K) = -\log p\left(\mathbf{X}_1^L \middle| \lambda^{(K)}\right) + \frac{\alpha(K)}{2}\log L \tag{5.26}$$

où $\alpha(K)$ est une mesure de complexité de la multi-gaussienne de K composantes $\lambda^{(K)}$. Dans notre cas, $\alpha = 2dK$ puisqu'il faut estimer K moyennes et K variances (matrices diagonales) dans R^d.

La valeur optimale du nombre K de gaussiennes est sensée correspondre au minimum du critère (5.26). Ce critère a été évalué pour chacune des multi-gaussiennes associées à un couple de paramètres du Tableau 5.1. Les résultats de cette analyse sont illustrés Figure 5.6. On remarquera que les courbes associées aux « gain de code fixe » g_c et « gain de dictionnaire adaptatif » g_p présentent des minimum locaux. En fait, ces deux paramètres ont la particularité d'être bornés comme on peut le constater sur la Figure 5.7 illustrant leur distribution. Le critère MDL tend alors à surestimer le nombre de gaussiennes nécessaires afin de modéliser les frontières de la distribution. On choisira donc systématiquement le premier minimum local.

---

[60] Les étapes Estimation- Maximisation de l'algorithme EM peuvent être comparées aux étapes « estimation du centroïde » et « recherche du plus proche voisin » de l'algorithme de la K-moyenne (cf. Annexe A), en remplaçant les distances euclidiennes utilisées par une pondération par la vraisemblance.

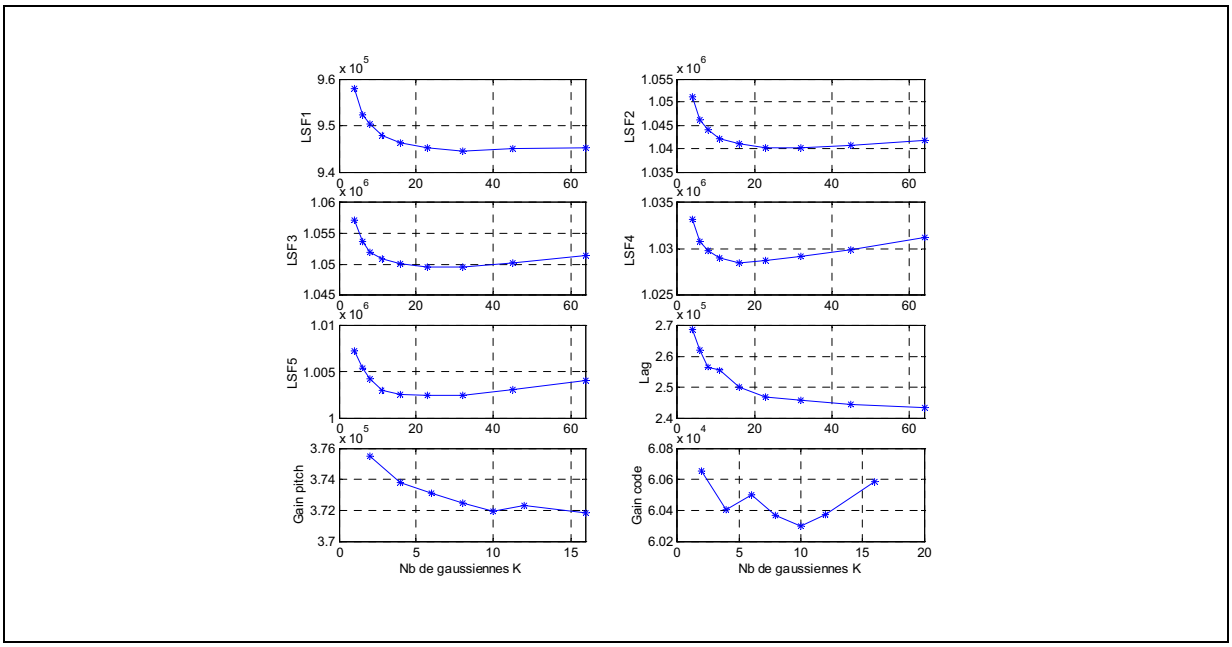

Figure 5.6 : Critères MDL calculés pour estimer le nombre optimal de gaussiennes pour chacun des paramètres modélisés

Finalement, le nombre K de gaussiennes des modèles appris pour chaque type de paramètre est reporté sur le Tableau 5.2 où on a également rappelé la taille N des dictionnaires de quantification du codeur EFR pour chacun de ces paramètres. On peut évaluer la *réduction globale de complexité* R_{AK1} apportée par l'algorithme AK1 GMM par rapport à l'algorithme AK1 basé sur les probabilités de transition entre index de quantification proposé par [Fingscheidt et al., 1997]. On a vu que pour un paramètre donné, cette réduction de complexité est de l'ordre $O\left(\frac{N}{K}\right)$ où N est la taille du dictionnaire de quantification. A partir des valeurs reportées sur le Tableau 5.2, on constate une *réduction globale de complexité* de l'ordre de $R_{AK1} \simeq 10$.

Paramètres	Nombre K de gaussiennes du modèle		Taille N des dictionnaires de quantification
LSF	Inter-trame : 22, 22, 32, 16, 16	Intra-trame : 22, 22, 22, 16	128, 256, 512, 256, 64
Gain code fixe	4		32
Gain code adaptatif	8		16
Délai de pitch (sous-trames 1 et 3)	32		512

Tableau 5.2 : Nombre de gaussiennes utilisées pour modéliser les redondances résiduelles dans l'espace des paramètres et comparaison avec la taille des dictionnaires de quantification

La superposition des multi-gaussiennes estimées (*courbes de niveau*) avec les distributions des couples de paramètres modélisés (*nuages de points*) est illustrée par les graphiques de la Figure 5.7 pour la 5$^{\text{ième}}$ LSF (corrélation inter-trame et intra-trame) ainsi que pour les gains de dictionnaires fixe et adaptatif (corrélation inter-trame).

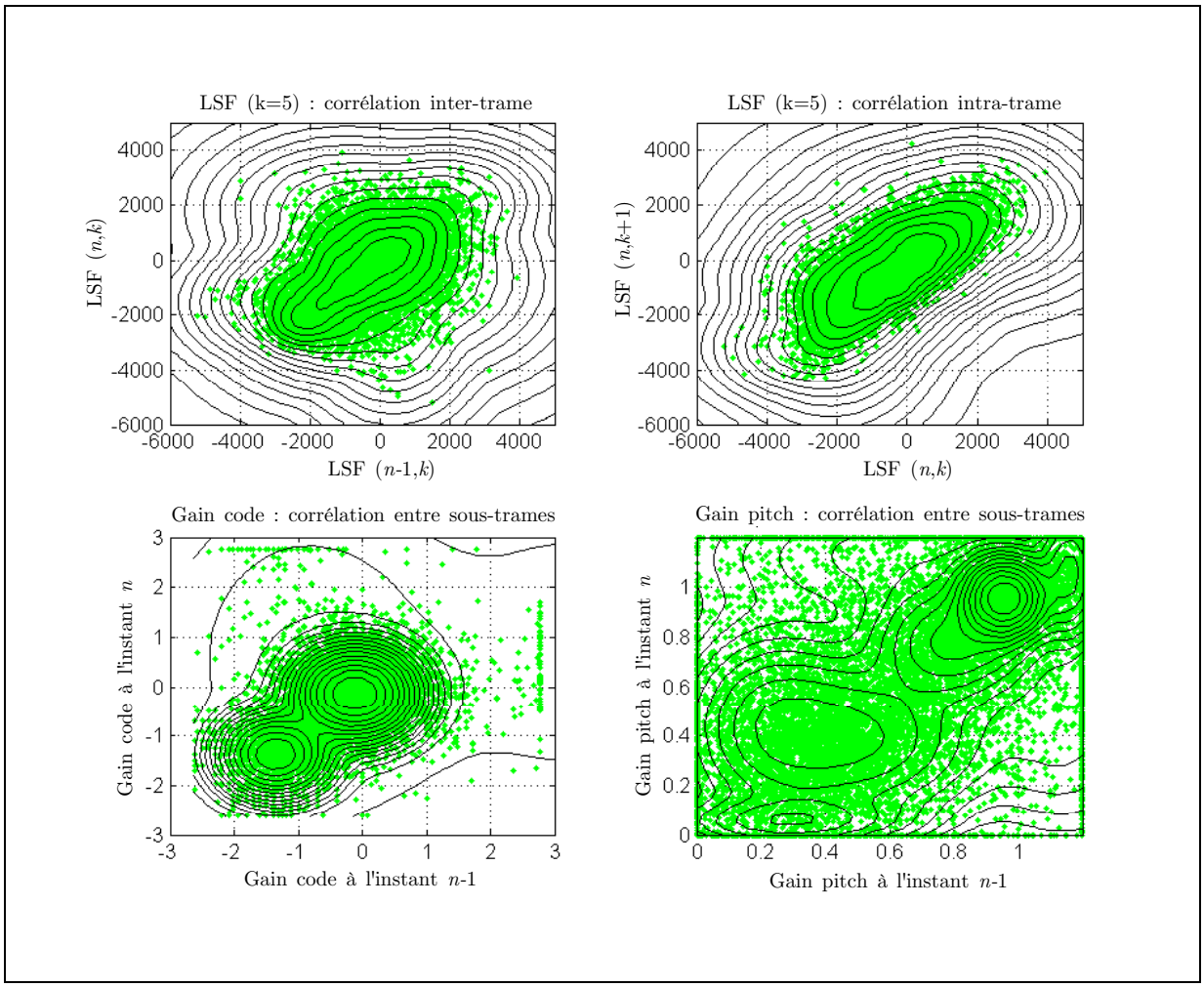

Figure 5.7 : Modélisation par GMM des distributions de couples de valeurs successives de paramètres (corrélation inter-trame) ou de LSF adjacentes (corrélation intra-trame)

5.3.2 Performances des algorithmes proposés

A partir des modèles multi-gaussiens estimés, nous pouvons mettre en oeuvre les algorithmes de décodage souple exploitant la non-uniformité (AK0 GMM), la corrélation temporelle (AK1 GMM) et la corrélation intra-trame, réduite ici à la corrélation entre LSF adjacentes (AK2 GMM). Les Figures 5.8 à 5.10 présentent les notes MOS estimées à partir de l'algorithme PESQ pour les algorithmes testés, en comparaison avec le masquage classique de l'EFR et les algorithmes de décodage souple utilisant un modèle *a priori* sur les index de quantification (AK0_Hist et AK1_Ptrans). Le canal

simulé est du type TU50 (cf. Annexe C) et le niveau de C/I est varié entre 2 et 7dB par pas de 1dB. Les résultats obtenus pour un C/I égal à 10dB sont également représentés afin de vérifier la convergence des algorithmes vers la qualité nominale du GSM EFR lorsque le niveau d'interférences diminue.

Dans le cas du décodage AK0 (exploitation de la non-uniformité), l'utilisation de multi-gaussiennes apporte un léger gain par rapport aux performances de l'algorithme AK0_Hist (histogramme des index de quantification). Nous y voyons l'influence du meilleur conditionnement de la GMM, qui permet notamment une bonne robustesse de l'algorithme pour des signaux non-appris. Ce phénomène sera clairement illustré avec la comparaison des signaux temporels (Figure 5.11) sur laquelle nous reviendrons par la suite.

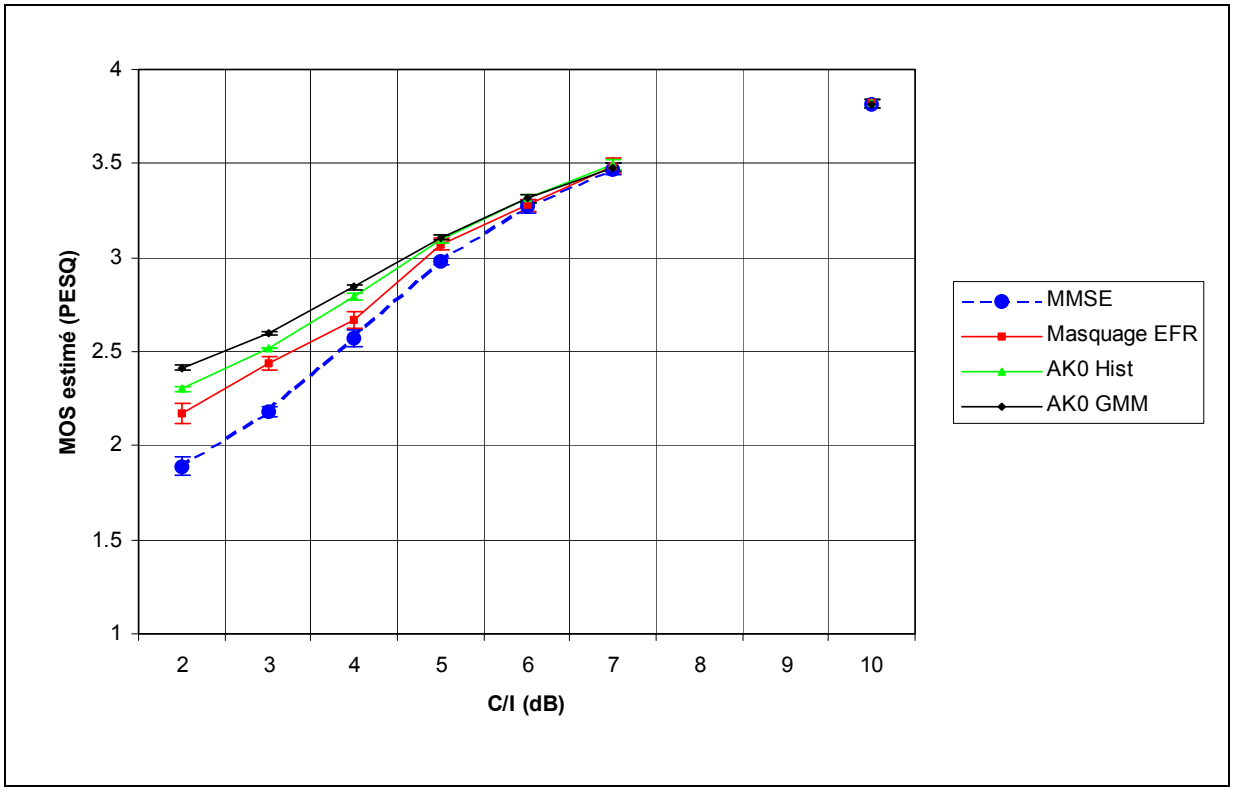

Figure 5.8 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

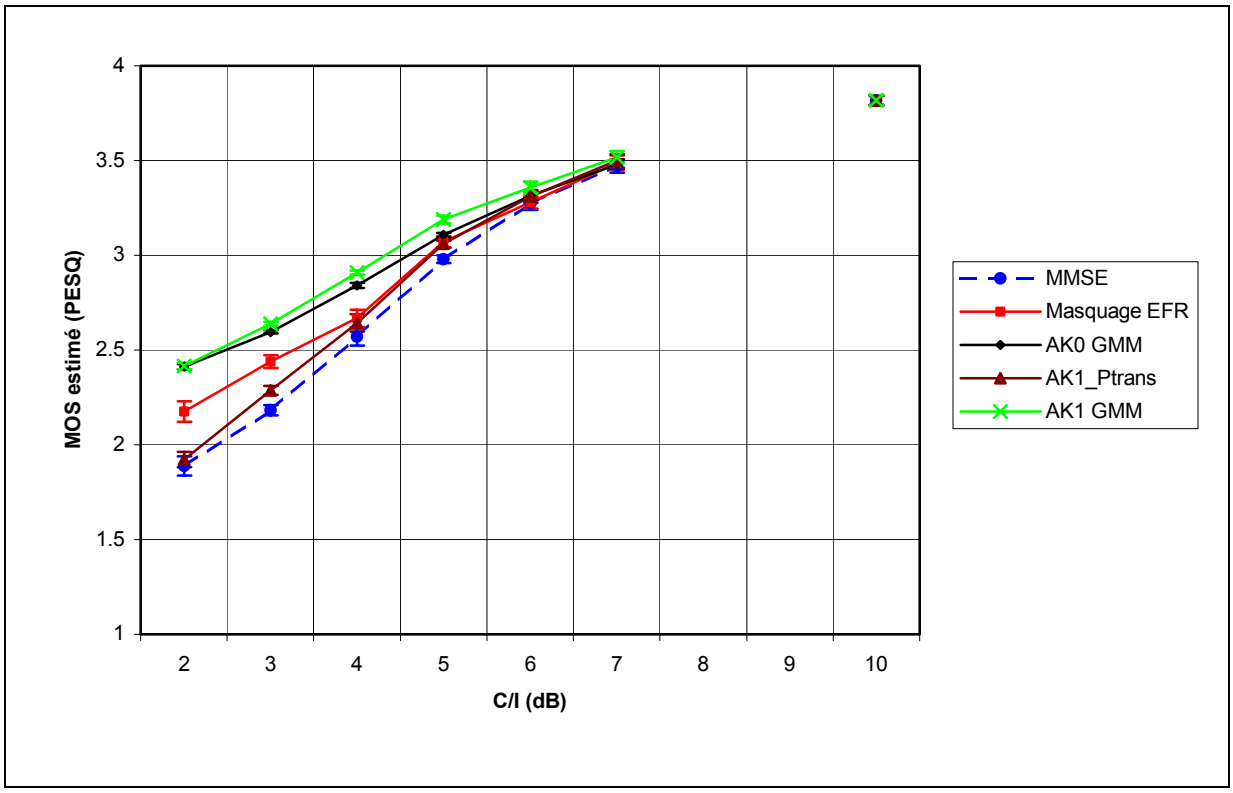

Figure 5.9 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

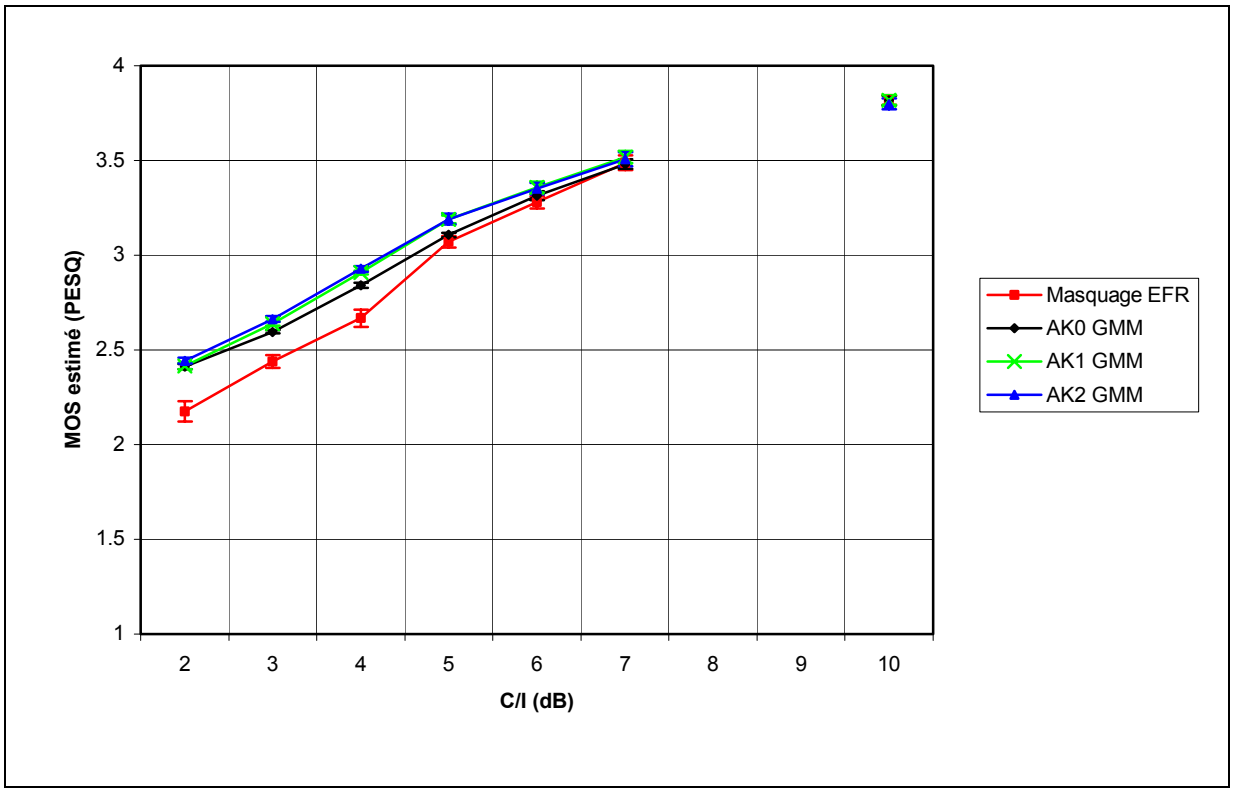

Figure 5.10 : Notes MOS estimées (PESQ) en fonction du niveau de C/I

L'intérêt de la modélisation par GMM apparaît encore plus clairement pour l'algorithme AK1 (prédiction inter-trame). Ici, d'une part, la réduction de complexité avec les paramètres choisis (cf. Tableau 5.2) est de l'ordre d'un facteur 10, mais d'autre part, l'apprentissage du modèle est mieux conditionné et les performances de l'estimateur qui en dérive permettent un gain de l'ordre de 0.4 MOS par rapport au masquage classique de l'EFR pour les valeur de C/I comprises entre 2dB et 4dB. On observe cependant que la prédiction inter-trame AK1 n'est intéressante que pour les C/I intermédiaires (compris entre 3dB et 6dB). Ceci signifie que pour les très bas niveaux de C/I, la probabilité *a posteriori* $p(i_n = i | j_n,...,j_1)$ est trop dégénérée (au sens de l'entropie) pour apporter une information, via le modèle de corrélation inter-trame, sur la trame à venir à l'instant $n+1$. La redondance exploitée par le modèle AK1 se réduit alors à la non-uniformité (modèle AK0).

L'exploitation de la redondance intra-trame (AK2 GMM), qui est ici essentiellement due à la relation d'ordre entre les LSF, n'apporte aucun gain en termes de qualité perçue. Ceci confirme les constatations faites au paragraphe 4.3 sur l'information exploitable au travers de la vraisemblance délivrée par le SOVA. Dans le cas du canal radiomobile, les erreurs sont de type « burst » (erreurs par paquet) et il apparaît que lorsqu'un burst affecte une trame, quasiment tous les paramètres sont corrompus. Ceci limite l'intérêt de la prédiction intra-trame dans le contexte de transmission radiomobile. Néanmoins, cette prédiction intra-trame pourrait être intéressante pour d'autres applications.

Enfin, une comparaison plus qualitative des performances des différents algorithmes est proposée Figure 5.11, où sont illustrés des exemples de signaux décodés selon chacune des méthodes pour un C/I égal à 2dB. Les erreurs cepstrales sont également calculées sur chacun des ces signaux.

On constate en premier lieu, que les algorithmes basés sur la modélisation GMM apportent une nette réduction des artefacts dans les zones de silence, en comparaison avec les algorithmes étudiés au chapitre précédent (Figure 4.11). Ce résultat semble assez surprenant puisque les zones de silence ont justement été exclues de la base d'apprentissage. En fait, nous voyons là une preuve du meilleur conditionnement du modèle appris par GMM[61]. Cet avantage de la modélisation par GMM avait déjà été mis en évidence sur les courbes de notes MOS estimées.

D'autre part, on observe une légère diminution *en moyenne* du nombre de pics d'erreur cepstrale (associés à une erreur localisée) entre le décodeur AK1 GMM et le décodeur AK0 GMM. Ceci apporte la preuve du *pouvoir de prédiction* de l'algorithme AK1 GMM. Cependant, le gain apporté par la prédiction inter-trame demeure limité et quelques pics voient leurs amplitudes augmenter, ce qui signifie qu'elle introduit de nouvelles erreurs. Un modèle de prédiction inter-trame mieux adapté au comportement des paramètres de la parole reste encore à obtenir.

---

[61] Puisque l'on a appris une loi multi-gaussienne (distribution continue), on est capable d'exprimer une probabilité *a priori* y compris pour des valeurs du paramètre non-apprises. Cette capacité de généralisation n'est cependant valable que si le nombre de gaussiennes utilisé ne « sur-modélise » par la distribution apprise.

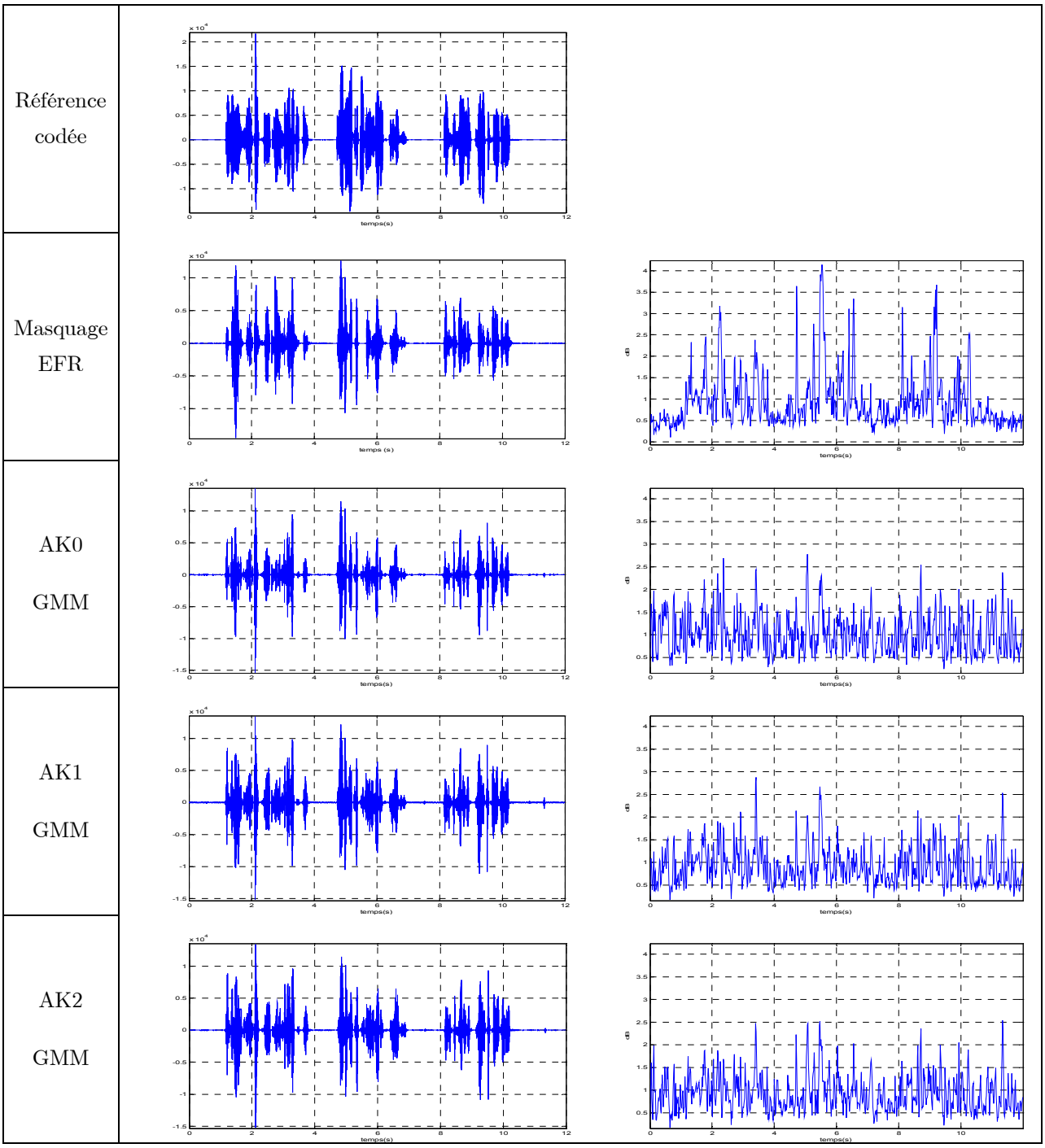

Figure 5.11 : Exemples de signaux décodés et distances cepstrales correspondantes (C/I=2dB, TU50)

5.4 Extensions du modèle de prédiction

Nous revenons ici sur la modélisation par multi-gaussiennes de la loi jointe $p(\mathbf{v}_n, \mathbf{v}_{n-1})$ afin de lui donner une interprétation plus physique. En effet, l'intérêt de cette modélisation ne réside pas uniquement dans la réduction de complexité. Elle effectue naturellement une classification dans l'espace joint $(\mathbf{v}_n, \mathbf{v}_{n-1})$ en « états » définis par une gaussienne ou un regroupement de gaussiennes.

Ceci est parfaitement visible sur la Figure 5.12 où on a représenté le calcul de la loi *prédictive* (5.12) à partir de la loi *a posteriori* $p(i_{n-1}|j_{n-1},...,j_1,\lambda)$ et de la loi jointe $p(\mathbf{v}_n, \mathbf{v}_{n-1})$. Le centre de chaque gaussienne est représenté par un point de largeur proportionnelle au poids w_m de la gaussienne. On y distingue un état « voisé » associé à une valeur du gain de pitch proche de l'unité et un état « non-voisé ».

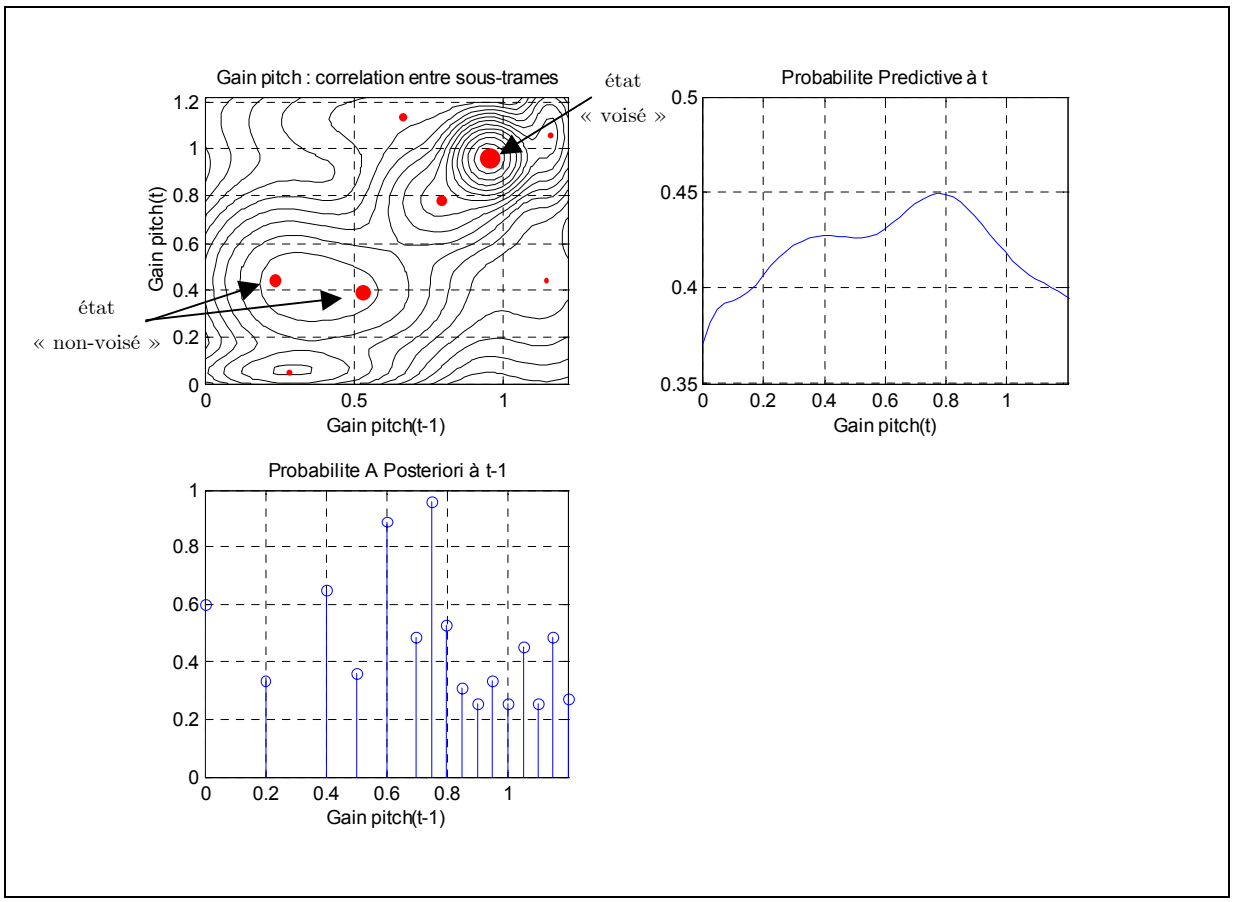

Figure 5.12 : Prédiction de la loi à l'instant n (haut – droite) à partir de la loi a posteriori à l'instant n-1 (bas – gauche) et du modèle de corrélation par GMM (haut – gauche)

Il pourrait être intéressant d'exploiter cette information de classification au niveau des autres paramètres. Par exemple, le comportement des LSF n'est pas le même selon que la trame de parole est voisée ou non et un codeur comme le G.729 exploite cette information de classification lors du masquage [ITU-T, G.729]. D'autre part, on peut généraliser la notion « d'état » de la parole en y incluant d'autres attributs que la seule information « voisé » / « non-voisé ».

5.4.1 Modélisation par HMM

Une manière de procéder serait de définir un état $\mathbf{S}$ en l'associant à un *jeu de probabilités a priori* des gaussiennes de chacune des lois jointes des paramètres définis par le Tableau 5.1 :

$$\text{état } \mathbf{S} \leftrightarrow \left\{ p\left(g_m^{(LSF_k)} | \mathbf{S} \right), p\left(g_{m'}^{(gc)} | \mathbf{S} \right), p\left(g_{m''}^{(gp)} | \mathbf{S} \right) \right\} \tag{5.27}$$

où $p(g_m^{(\mathbf{x})}|\mathbf{S})$ désigne le poids de la gaussienne g_m (conditionnellement à l'état $\mathbf{S}$) dans la GMM modélisant la loi jointe $p(\mathbf{v}_n, \mathbf{v}_{n-1})$ du paramètre $\mathbf{v}$.

L'état $\mathbf{S}$ introduit ainsi une *dépendance* entre les lois modélisant la *corrélation temporelle* $p(\mathbf{v}_n, \mathbf{v}_{n-1})$ des différents paramètres, par l'intermédiaire du poids de leurs gaussiennes. Plus précisément, en prenant l'exemple d'une modélisation en états « voisés » et « non-voisés » de la parole :

On introduit deux états $\mathbf{S}_0$ et $\mathbf{S}_1$ associés respectivement à « voisé » et à « non-voisé » et définissant chacun un jeu de probabilité *a priori* des gaussiennes dans les lois $p(\mathbf{v}_n, \mathbf{v}_{n-1})$, pour tous les paramètres $\mathbf{v}$ dont on souhaite prendre en compte la dépendance à l'état voisé / non-voisé.

Le modèle doit être complété par les probabilités a priori des états $\mathbf{S}_0$ et $\mathbf{S}_1$ ainsi que par les probabilités de transition entre états :

$$\begin{aligned} \pi_0 &= p(\mathbf{S}_0) \\ \pi_1 &= p(\mathbf{S}_1) \\ a_{ij} &= p\left(\mathbf{S}_i \middle| \mathbf{S}_j\right);\ i,j \in \{0,1\} \end{aligned} \tag{5.28}$$

Un tel modèle peut être appris par un algorithme de type *segmental K-mean* [Rabiner, 1989] où la segmentation voisé / non-voisé initiale est obtenue d'après un modèle *a priori*. Ce modèle *a priori* peut par exemple être fourni par la classification voisé / non-voisé obtenue à partir de la loi $p(gp_n, gp_{n-1})$ illustrée Figure 5.12.

Les développements qui précèdent n'ont pas d'autre ambition que d'ouvrir une perspective et mériteraient un approfondissement. On pourra remarquer cependant que le modèle de HMM obtenu ici correspond à une proposition de [Wellekens, 1987] pour prendre en compte la corrélation temporelle dans une HMM en modélisant la loi d'émission associée à chaque état par une loi jointe.

5.5 Conclusion

Les méthodes de décodage souple généralisent au niveau du décodeur parole, une approche déjà répandue dans tous les autres éléments de la chaîne de réception. Cette approche consiste à utiliser des entrées souples et, si possible, à générer des sorties souples, de manière à limiter les erreurs de décision. Cependant, l'application de cette approche au décodeur parole se révèle très complexe en raison de la taille des dictionnaires de quantification à parcourir.

Après avoir vérifié l'existence d'une redondance résiduelle laissée par le codeur de parole EFR, nous avons proposé une méthode permettant de réduire d'un facteur 10 la complexité par rapport aux approches de l'état de l'art. De plus, cette méthode, basée sur une modélisation par mélange de gaussiennes de la distribution *a priori* des paramètres, offre un meilleur conditionnement des estimateurs. Les performances des algorithmes ainsi proposés permettent un gain de l'ordre de 0,4 MOS par rapport à la procédure de masquage classique de l'EFR pour des niveaux de C/I compris entre 2dB et 4dB, tout en convergeant vers la qualité nominale de l'EFR dans le cas où le canal n'introduit pas d'erreur.

Cependant, le modèle de prédiction fixe AK1 utilisé pour exploiter la corrélation inter-trame n'est pas pertinent et limite le gain relatif observé par rapport à l'algorithme AK0 exploitant la seule non-uniformité. En fait, la prise en compte de la corrélation inter-trame n'est vraiment intéressante que pour les niveaux de C/I intermédiaires, c'est-à-dire entre 3dB et 6dB. Pour les niveaux inférieurs de C/I, seule l'information de non-uniformité (AK0) est exploitable puisque la faible confiance dans les données reçues limite l'information réellement apportée par la prédiction inter-trame. Or le modèle de prédiction fixe utilisé par l'algorithme AK1 s'inspire des modèles (invariants) d'extrapolation de trame perdue, c'est-à-dire de procédures développées pour les bas niveaux de C/I. Il est nécessaire de chercher un modèle de prédiction mieux adapté au comportement non-stationnaire des paramètres de la parole afin d'obtenir un gain significatif de qualité perçue pour les niveaux de C/I intermédiaires. Une voie d'amélioration de la prédiction inter-trame pourrait être la modélisation par « états » introduite en fin de ce chapitre.

Chapitre 6

Décodage canal contrôlé par la source : Principe et état de l'art

6.1 Introduction

Les méthodes de décodage souple de parole étudiées aux chapitres précédents visent à exploiter la redondance résiduelle des paramètres du codeur parole pour lutter contre les erreurs de transmission en sortie d'un canal équivalent qui, dans le cas du GSM, inclut un décodeur canal. Il semble naturel d'essayer d'exploiter cette *redondance résiduelle de source* directement au niveau du décodeur canal, c'est-à-dire conjointement avec la *redondance systématique* introduite par le codeur canal. C'est l'idée à la base des techniques de *décodage de canal contrôlé par la source*.

Si la démarche peut sembler parallèle avec celle du décodeur souple de parole, le point de vue avec lequel la redondance résiduelle est exploitée diffère sensiblement entres ces deux approches. Dans le cas du décodage canal contrôlé par la source (SCCD), l'objectif est celui de la *correction d'erreur binaire*, l'information *a priori* issue de la redondance étant exploitée au niveau des bits. En revanche, le décodeur souple de parole, lorsqu'il réalise l'estimation des paramètres au sens du MMSE, peut être vu comme un intermédiaire entre la correction d'erreur et le *masquage*, la redondance étant alors utilisée pour minimiser l'impact perceptif des erreurs plutôt que pour les annuler.

Un des arguments en faveur du décodage de canal contrôlé par la source (SCCD) est qu'il est sensé être plus robuste pour les niveaux d'interférences C/I intermédiaires, pour lesquels la correction d'erreur binaire est effective alors que les paramètres estimés par le décodeur souple sont, eux, déjà biaisés. D'autre part, les approches SCCD et décodage de parole souple peuvent être complémentaires même si ce point reste à vérifier.

Nous analysons dans ce chapitre les différentes techniques proposées dans le domaine du décodage canal contrôlé par la source. Nous évaluerons tout particulièrement leur pertinence vis-à-vis de la modélisation de la redondance résiduelle du codeur parole et vis-à-vis des contraintes imposées par la stratégie de codage canal du GSM.

6.2 Principe du décodage canal contrôlé par la source

Considérons le schéma de transmission illustré Figure 6.1 où l'on a fait cette fois-ci apparaître explicitement le codeur/décodeur canal. Le *canal équivalent* correspond alors au récepteur interne présenté au Chapitre 1 et détaillé en Annexe C. Il regroupe l'opération d'entrelacement, le modulateur, le milieu de transmission, le démodulateur/égaliseur et enfin, le dé-entrelacement. On assimilera ici ce canal équivalent à un *canal sans mémoire*, *binaire symétrique*, dont les probabilités de transition *instantanées* p_e peuvent être estimées à partir de la *sortie souple* de l'égaliseur[62] (cf. Annexe C).

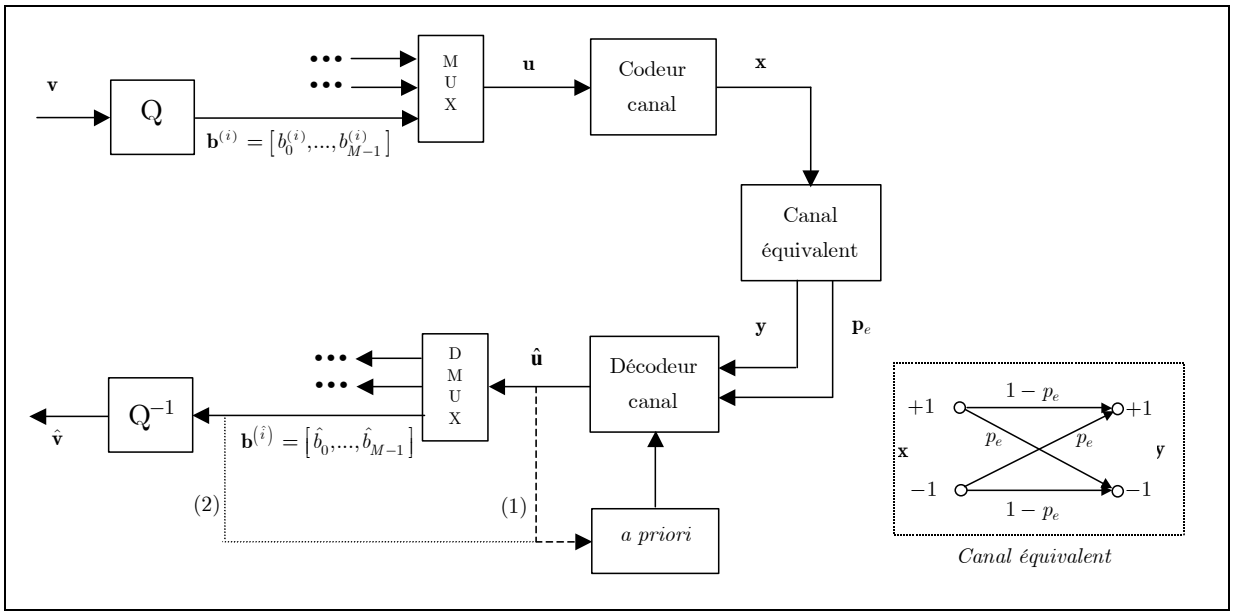

Figure 6.1 : Principe du décodage canal contrôlé par la source

[62] Le canal équivalent associé au récepteur interne est généralement modélisé comme un canal à bruit additif gaussien (CABG). Nous adoptons ici un point de vue différent (Canal Binaire Symétrique) afin de maintenir un parallèle avec la démarche du décodage parole à entrées souples présentée au Chapitre 3. De plus, ceci correspond à l'interprétation exacte de la sortie calculée par l'égaliseur qui est ici un égaliseur de Viterbi de type SOVA (cf. Annexe C).

Le codeur canal reçoit en entrée une trame de *bits d'information* $\mathbf{u} = \left[u_1,...,u_k,...,u_L\right]$, ces bits sont associés aux index de quantification en sortie de codeur parole mais peuvent avoir subi des opérations diverses (codage préliminaire, ré-ordonnancement). Nous reviendrons par la suite sur la relation entre la trame de bits d'information **u** et la trame de bits en sortie de codeur parole **b**. On considérera ici des bits u_k à valeur dans $\{+1,-1\}$. Le codeur canal est un codeur convolutif de *rendement* $\frac{1}{N}$ et de mémoire ν, qui associe à chaque bit d'information u_k, un *symbole canal* $\mathbf{x}_k = \left\{x_{k,1},...,x_{k,N}\right\}$. Comme on l'a présenté en Annexe B, un tel codeur convolutif peut être décrit par un *treillis* dont les états q_k à l'étape k sont définis par les bits d'information $\left[u_{k-1},...,u_{k-\nu}\right]$ précédemment entrés dans le codeur (effet mémoire) et dont les transitions entre états $\left(q_{k-1},q_k\right)$ sont associées au bit d'information u_k en entrée. Un symbole canal $\mathbf{x}_k$ est délivré en sortie du codeur pour chaque transition $\left(q_{k-1},q_k\right)$ déclenchée par le bit d'information u_k entré. Ceci est rappelé schématiquement sur la Figure 6.2 .

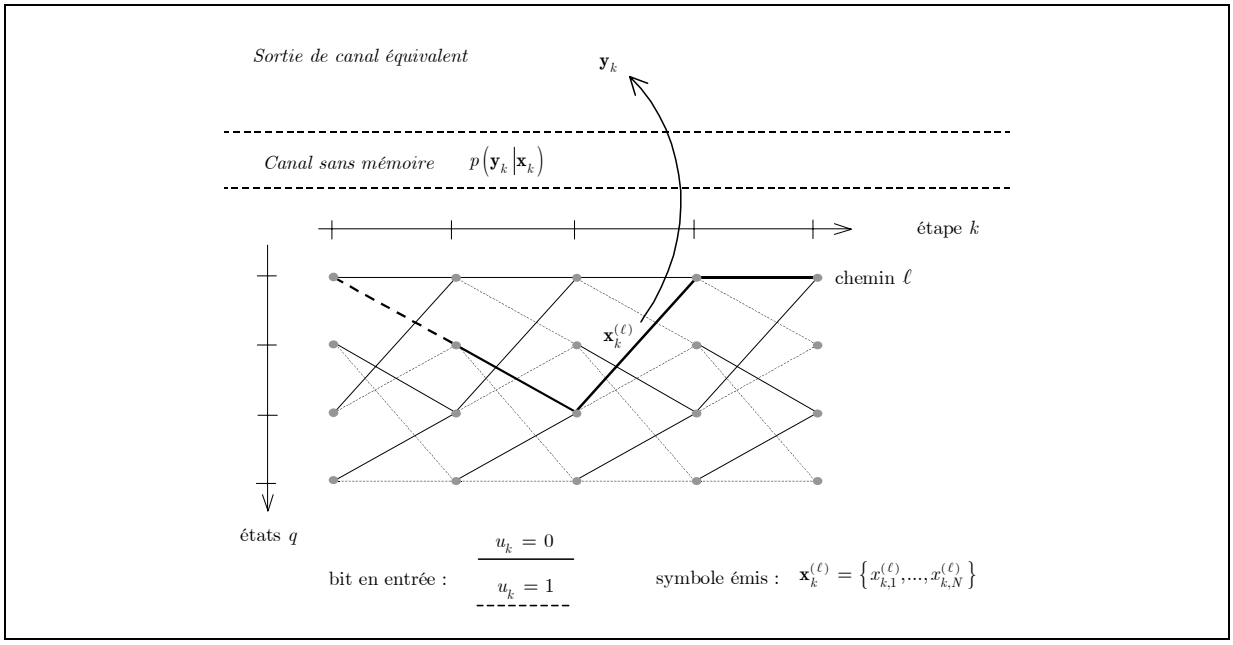

Figure 6.2 : Codage convolutif et observation associée en sortie de canal équivalent

On s'intéresse ici au décodage canal par séquences[63]. A partir de la séquence $\mathbf{Y} = \left\{\mathbf{y}_1,...,\mathbf{y}_L\right\}$ observée en sortie de canal équivalent, le décodeur cherche à retrouver la séquence des états q_k afin d'estimer les bits d'information u_k en entrée. Plus exactement, l'algorithme de Viterbi recherche le chemin $\hat{\mathbf{q}}^{(\ell)}$ de probabilité *a posteriori* maximale :

$$\hat{\mathbf{q}}^{(\ell)} = \arg\max_{\ell} p\left(\mathbf{q}^{(\ell)}\middle|\mathbf{Y}\right) = \arg\max_{\ell} p\left(\mathbf{Y}\middle|\mathbf{q}^{(\ell)}\right) p\left(\mathbf{q}^{(\ell)}\right) \tag{6.1}$$

[63] Il existe aussi l'algorithme de Bahl, qui minimise la probabilité d'erreur par symbole et non par séquence (cf. Annexe D), mais celui-ci est peu utilisé en pratique, du fait de sa complexité.

Cependant, dans la mise en œuvre classique de l'algorithme de Viterbi, on fait l'hypothèse supplémentaire *d'équiprobabilité* et *d'indépendance* des bits d'information u_k, autrement dit, tous les chemins $\mathbf{q}^{(\ell)}$ ont la même probabilité *a priori*. Comme le canal équivalent est supposé sans mémoire, le terme à maximiser dans l'équation (6.1) se réduit au produit des *vraisemblances des transitions* entre états :

$$p\left(\mathbf{Y}\middle|\mathbf{q}^{(\ell)}\right) = \prod_k p\left(\mathbf{y}_k\middle|q_{k-1}^{(\ell)}, q_k^{(\ell)}\right) = \prod_k p\left(\mathbf{y}_k\middle|\mathbf{x}_k^{(\ell)}\right) \tag{6.2}$$

La correction d'erreur est ici possible du fait de la *contrainte* imposée par la mémoire du codeur sur les séquences d'états q_k *admissibles* (structure du treillis).

Cependant, on a vu qu'il existait une redondance résiduelle au niveau des index de quantification en sortie du codeur parole. Celle-ci se retrouve au niveau des trames de bits d'information u_k avec pour conséquence que les chemins parcourus dans le treillis ne sont pas équiprobables. L'idée du *décodage contrôlé par la source* (SCCD) est d'utiliser la *probabilité a priori* des chemins $p\left(\mathbf{q}^{(\ell)}\right)$ afin de rajouter une contrainte supplémentaire lors du décodage par *Maximum a Posteriori* (6.1) pour améliorer la *correction d'erreur*.

Les approches proposées dans la littérature se différencient par le modèle de probabilité *a priori* $p\left(\mathbf{q}^{(\ell)}\right)$ qu'elles exploitent. Le problème est ici que la maximisation directe de la probabilité (6.1) n'est pas possible pour des raisons de complexité. L'algorithme de Viterbi la résout par maximisation *récursive* sur les instants k, ce qui signifie qu'il faut scinder la probabilité *a priori* $p\left(\mathbf{q}^{(\ell)}\right)$ en probabilités élémentaires *indépendantes* associées aux branches $\left(q_{k-1}^{(\ell)}, q_k^{(\ell)}\right)$ du treillis, de manière à pouvoir prendre une décision sur le chemin de métrique maximale à chaque état q_k. Autrement dit, puisque chaque branche du codeur est associée à un bit d'information u_k, la redondance de la trame $\mathbf{u}$ doit être modélisée par des probabilités *a priori au niveau des bits individuels* u_k. Deux modèles sont alors possibles :

- Les bits u_k sont supposés *indépendants* et on ne modélise que la non-uniformité de la distribution $p\left(u_k\right)$.
- La corrélation entre bits d'une même trame $\mathbf{u}$ (*corrélation intra-trame*) est prise en compte par des lois conditionnelles de la forme $p\left(u_k\middle|\left\{u_j\right\}_{j\neq k}\right)$.

Dans les deux cas, les lois *a priori* peuvent être actualisées en fonction de la trame décodée précédente, afin de prendre en compte la corrélation temporelle (*corrélation inter-trame*). En revanche, seule la seconde approche permet de modéliser la redondance au niveau des *index de quantification* du codeur de parole, en exploitant, au moins partiellement, la loi *jointe* $p\left(u_{k_0}, ..., u_{k_{M-1}}\right)$ des bits codant un même index de quantification. Ces approches sont présentées de manière plus approfondie dans les paragraphes qui suivent.

6.3 Non-uniformité et corrélation temporelle des bits individuels

6.3.1 Métrique modifiée de l'algorithme de Viterbi

Dans cette approche, initialement proposée par [Hagenauer, 1995], on exploite la redondance résiduelle sous le seul aspect de la non-uniformité des bits *individuels* u_k en supposant que les valeurs $+1$ et -1 ne sont pas équiprobables, au moins pour certains bits :

$$p\big(u_k = +1\big) \neq p\big(u_k = -1\big) \tag{6.3}$$

On montre ici qu'il est possible d'intégrer cette information *a priori* moyennant une très légère modification de l'algorithme de Viterbi. Pour cela, nous repartons de la probabilité *a posteriori* (6.1) et introduisons la variable :

$$\alpha_k^{(\ell)} = \; p\big(\mathbf{y}_1,...,\mathbf{y}_k,q_1^{(\ell)},...,q_{k-1}^{(\ell)},q_k^{(\ell)}\big) \tag{6.4}$$

qui s'interprète, à une constante près, comme la probabilité *a posteriori* du *chemin partiel* ℓ jusqu'à l'étape k. Le chemin optimal de l'étape initiale $k = 1$ à l'étape finale $k = L$ du treillis, s'écrit :

$$\hat{\mathbf{q}} = \arg\max_{\ell} p\big(\mathbf{q}^{(\ell)}|\mathbf{Y}\big) = \arg\max_{\ell} \alpha_L^{(\ell)} \tag{6.5}$$

La variable $\alpha_k^{(\ell)}$ peut s'écrire en fonction de sa valeur à l'étape précédente $\alpha_{k-1}^{(\ell)}$ selon :

$$\alpha_k^{(\ell)} = \alpha_{k-1}^{(\ell)} p\big(\mathbf{y}_k\big|q_{k-1}^{(\ell)},q_k^{(\ell)}\big)\, p\big(q_k = q_k^{(\ell)}\big|q_{k-1}^{(\ell)},...,q_1^{(\ell)}\big) \tag{6.6}$$

Comme on ne remet pas ici en cause l'hypothèse d'*indépendance* des bits u_k à l'intérieur d'une trame $\mathbf{u}$, on a :

$$p\big(q_k = q_k^{(\ell)}\big|q_{k-1}^{(\ell)},...,q_1^{(\ell)}\big) = p\big(u_k = u_k^{(\ell)}\big) \tag{6.7}$$

La relation (6.6) s'écrit alors :

$$\alpha_k^{(\ell)} = \alpha_{k-1}^{(\ell)} p\big(\mathbf{y}_k\big|\mathbf{x}_k^{(\ell)}\big)\, p\big(u_k = u_k^{(\ell)}\big) \tag{6.8}$$

soit, en passant dans le domaine logarithmique avec la *métrique* $M_k^{(\ell)} = \log \alpha_k^{(\ell)}$:

$$M_k^{(\ell)} = M_{k-1}^{(\ell)} + \sum_{r=1}^{N} \log p\left(y_{k,r} \middle| x_{k,r}^{(\ell)}\right) + \log p\left(u_k = u_k^{(\ell)}\right) \tag{6.9}$$

On reconnaît le calcul de métrique de l'algorithme de Viterbi classique auquel on a ajouté un terme sur la distribution *a priori* du bit u_k. L'incrément de métrique ne dépend que de la branche $\left(q_{k-1}^{(\ell)}, q_k^{(\ell)}\right)$ du treillis, le long du chemin ℓ considéré, et des données reçues à l'étape k. Il en résulte que la *maximisation* de la métrique $M_k^{(\ell)}$ peut se faire de manière *récursive* sur les instants k. On retrouve ainsi le fonctionnement de l'algorithme de Viterbi qui conserve, à chaque étape k et pour chaque état, uniquement le meilleur chemin m aboutissant à cet état. L'algorithme résultant a été appelé APRI-VA par [Hagenauer, 1995], il effectue un décodage par séquence au sens du Maximum a Posteriori et non plus seulement au sens du Maximum de Vraisemblance comme le décodeur canal classique.

6.3.1.1 Valeurs souples et interprétation

Il est pratique de reformuler la métrique (6.9) dans le domaine des *valeurs souples*. On rappelle qu'on définit la *valeur souple* $L(b)$ associée à un bit b à valeur dans $\{+1,-1\}$ comme le logarithme du rapport des probabilités $p(b=+1)$ et $p(b=-1)$. La valeur souple $L(b)$ représente donc *la connaissance* que l'on a sur un bit b non encore observé (valeur souple *a priori*) ou observé au travers d'un canal introduisant des erreurs (valeur souple *a posteriori*). Une représentation équivalente de la valeur souple est fournie par le couple $\left(\hat{b}, p_e\right)$ correspondant à la *décision ferme* sur la valeur du bit b et à la probabilité d'erreur associée à cette décision. On a ainsi la relation :

$$\begin{aligned} L(b) &= \log \frac{p(b=+1)}{p(b=-1)} \\ &= \hat{b} \log \frac{1-p_e}{p_e} \end{aligned} \tag{6.10}$$

Les « valeurs souples » en sortie du canal équivalent sont définies par les couples $\left(y_{k,r}, p_{e_{k,r}}\right)$ des décisions fermes $y_{k,r}$ en sortie du canal et de leur probabilités d'erreur estimées $p_{e_{k,r}}$. On a :

$$p\left(y_{k,r} \middle| x_{k,r}^{(\ell)}\right) = \begin{cases} 1 - p_{e_{k,r}} & \text{si } y_{k,r} = x_{k,r}^{(\ell)} \\ p_{e_{k,r}} & \text{si } y_{k,r} \neq x_{k,r}^{(\ell)} \end{cases} \tag{6.11}$$

En remarquant qu'on peut respectivement remplacer les conditions $y_{k,r} = x_{k,r}^{(\ell)}$ et $y_{k,r} \neq x_{k,r}^{(\ell)}$ par $x_{k,r}^{(\ell)} y_{k,r} = +1$ et $x_{k,r}^{(\ell)} y_{k,r} = -1$, il vient après quelques manipulations :

$$p\left(y_{k,r} \middle| x_{k,r}^{(\ell)}\right) = \frac{1}{2}\left(x_{k,r}^{(\ell)} y_{k,r} \log \frac{1-p_{e_{k,r}}}{p_{e_{k,r}}} + C_{y_{k,r}} \right) \tag{6.12}$$

avec $C_{y_{k,r}} = \log p_{e_{k,r}} + \log\Big(1 - p_{e_{k,r}}\Big)$.

En suivant une démarche similaire, on peut exprimer la probabilité *a priori* $p\big(u_k = u_k^{(\ell)}\big)$ en fonction de la valeur souple *a priori* du bit u_k :

$$p\big(u_k = u_k^{(\ell)}\big) = \frac{1}{2}\left(u_k^{(\ell)} \log \frac{p\big(u_k = +1\big)}{p\big(u_k = +1\big)} + C_{u_k} \right) \tag{6.13}$$

avec $C_{u_k} = \log p\big(u_k = +1\big) + \log p\big(u_k = -1\big)$.

On remarquera que les quantités $C_{y_{k,r}}$ et C_{u_k} sont indépendantes de la branche du treillis $\big(q_{k-1}^{(\ell)}, q_k^{(\ell)}\big)$ considérée à une étape k donnée. Elles n'interviennent donc pas dans la maximisation de la métrique $M_k^{(\ell)}$ relativement aux chemins ℓ. Il en résulte qu'une métrique équivalente à la métrique (6.9) est donnée par la récursion suivante :

APRI-VA :
$$M_k^{(\ell)} = M_{k-1}^{(\ell)} + \sum_{r=1}^{N} x_{k,r}^{(\ell)} L_{c_{k,r}} y_{k,r} + u_k^{(\ell)} L\big(u_k\big) \tag{6.14}$$

où $L_{c_{k,r}} = \log \dfrac{1 - p_{e_{k,r}}}{p_{e_{k,r}}}$ représente la *fiabilité* des bits $y_{k,r}$ reçus en sortie du canal équivalent et $L\big(u_k\big)$ est la valeur souple *a priori* du bit d'information u_k. Le calcul de cette métrique est illustré Figure 6.3.

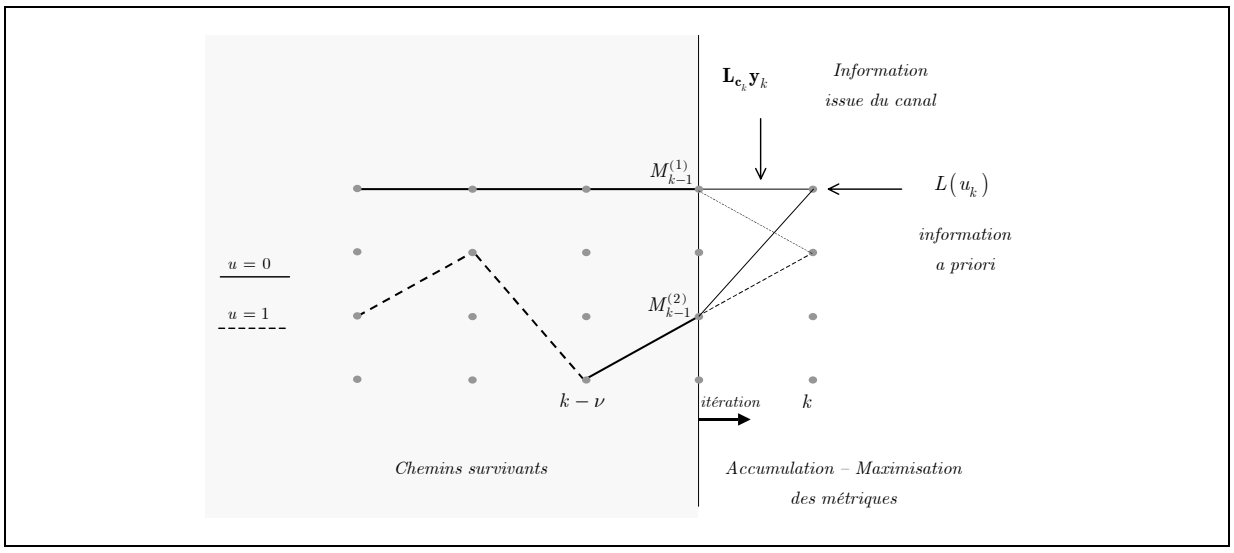

Figure 6.3 : Formation des métriques des branches du treillis par l'APRI-VA

Ainsi, lorsque la confiance $L_{c_{k,r}}$ dans les données issues du canal est élevée, la contribution de l'information *a priori* est négligeable et la métrique converge vers celle du décodeur de Viterbi classique (Maximum de Vraisemblance). A l'inverse, lorsque la confiance dans le canal $L_{c_{k,r}}$ diminue,

c'est l'information *a priori* $L(u_k)$ qui permet de maintenir une pondération non-uniforme des branches, permettant la correction d'erreur. On retrouve donc au niveau de chaque branche du treillis, un mécanisme similaire à celui mis en œuvre par les algorithmes de décodage souple de parole abordés aux chapitres précédents. Ce mécanisme n'est autre que la formation d'une *probabilité a posteriori* à partir d'une vraisemblance issue du canal et d'une probabilité *a priori* issue d'un modèle de la source.

6.3.2 Calcul des valeurs souples a priori des bits d'information

On n'a pas précisé jusqu'à maintenant comment obtenir la valeur souple *a priori* $L(u_k)$ du bit d'information. Cette valeur peut être *fixe*, apprise sur une base de données pour chaque bit de la trame $\mathbf{u}$, dans ce cas on exploite uniquement la non-uniformité de la distribution *moyenne* de chaque bit individuel u_k. Ceci peut être intéressant par exemple pour les bits de poids forts en sortie d'un quantificateur scalaire, la probabilité d'un paramètre diminuant vers les valeurs extrêmes de sa plage de quantification. Dans le cas d'une quantification vectorielle, comme c'est la cas pour les LSF du codeur EFR, l'intérêt semble nettement moins évident. En revanche, la valeur souple *a priori* $L(u_k)$ peut être *prédite* à partir des bits d'information décodés pour la trame précédente afin d'exploiter la redondance liée à la *corrélation temporelle* résiduelle en sortie du codeur. On a recensé deux approches distinctes proposées pour modéliser cette corrélation *inter-trame* au niveau des bits d'information u_k, nous les détaillons dans les paragraphes suivants.

6.3.2.1 Modélisation de la corrélation temporelle entre bits individuels

Cette approche a été introduite par [Hagenauer, 1995]. On désignera ici par n l'indice de trame et on notera $\mathbf{u}_n = \left[u_{n,1},...,u_{n,k},...,u_{n,L}\right]$, la trame de bits d'information reçue à « l'instant trame » n. On s'intéresse à la corrélation temporelle entre bits de *même position* k appartenant à des trames successives. L'information véritablement exploitable au travers de la corrélation temporelle des bits pris *individuellement* est celle de *l'invariance* de chaque bit au cours du temps. En effet, le postulat de la démarche présentée ici est que pour les zones stationnaires de la parole, ou les segments de silence, certains bits $u_{n,k}$ conservent la même valeur pour des instants trame n successifs. Aussi, la corrélation temporelle des bits $u_{n,k}$ est modélisée ici par l'intermédiaire d'un bit de *changement de signe* $c_{n,k}$ entre la valeur $u_{n-1,k}$ et $u_{n,k}$:

$$u_{n,k} = u_{n-1,k} + c_{n,k} \tag{6.15}$$

$c_{n,k}$ est à valeurs dans $\{+1,-1\}$ et l'addition utilisée ici est celle définie dans le groupe GF2 (cf. Annexe B, Table B.1 avec la correspondance $0 \rightarrow 1$ et $1 \rightarrow -1$)[64].

[64] L'addition dans GF2 correspond au « ou exclusif » dans $\{0,1\}$, ou encore à la multiplication dans $\{+1,-1\}$.

La *connaissance a priori* sur la corrélation (probabilité de changement de signe) entre $u_{n,k}$ et $u_{n-1,k}$ est apportée par la valeur souple *a priori* du bit $c_{n,k}$:

$$L\left(c_{n,k}\right) = \log\frac{1 - Pc_{n,k}}{Pc_{n,k}} \tag{6.16}$$

où $Pc_{n,k}$ est la probabilité de changement de signe. Ce modèle de corrélation temporelle apparaît équivalent à un modèle de *Markov d'ordre 1* de probabilités de transition $p\left(u_{n,k}\middle|u_{n-1,k}\right)$ *symétriques*. Il est illustré Figure 6.4.

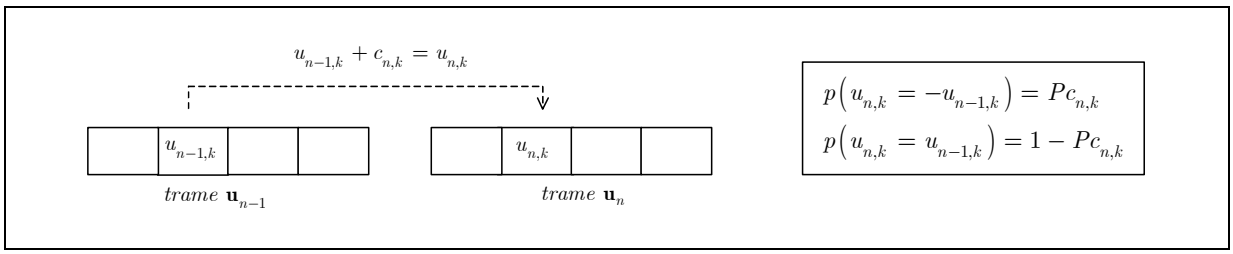

Figure 6.4 : Modèle de corrélation inter-trame par addition d'un bit de changement de signe

L'intérêt de ce modèle par addition d'un bit de changement de signe est que l'on peut en déduire une relation très simple entre les valeurs souples correspondantes. On montre en effet [Hagenauer, 1995] que la valeur souple $L\left(u_{n,k}\right)$ peut être estimée à partir de (6.15) selon :

$$L\left(u_{n,k}\right) \simeq \operatorname{sign}\left(L\left(c_{n,k}\right)\right)\operatorname{sign}\left(L\left(u_{n-1,k}\right)\right)\min\left(\left|L\left(c_{n,k}\right)\right|,\left|L\left(u_{n-1,k}\right)\right|\right) \tag{6.17}$$

Ainsi, connaissant la valeur souple du bit $u_{n-1,k}$ décodé à la trame *précédente* et le modèle de corrélation *a priori*, c'est-à-dire la valeur souple *a priori* du bit $c_{n,k}$, on peut prédire une valeur souple *a priori* du bit $u_{n,k}$ que l'on s'apprête à décoder. Ceci est illustré Figure 6.5 où l'on a mis en évidence les valeurs *a posteriori* et les valeurs *a priori* entrant en jeu dans la prédiction inter-trame. La valeur souple du bit décodé à la trame précédente $u_{n-1,k}$ est disponible si l'on utilise un décodeur canal à sortie souple (algorithmes SOVA ou Max-Log MAP, présentés en Annexe D).

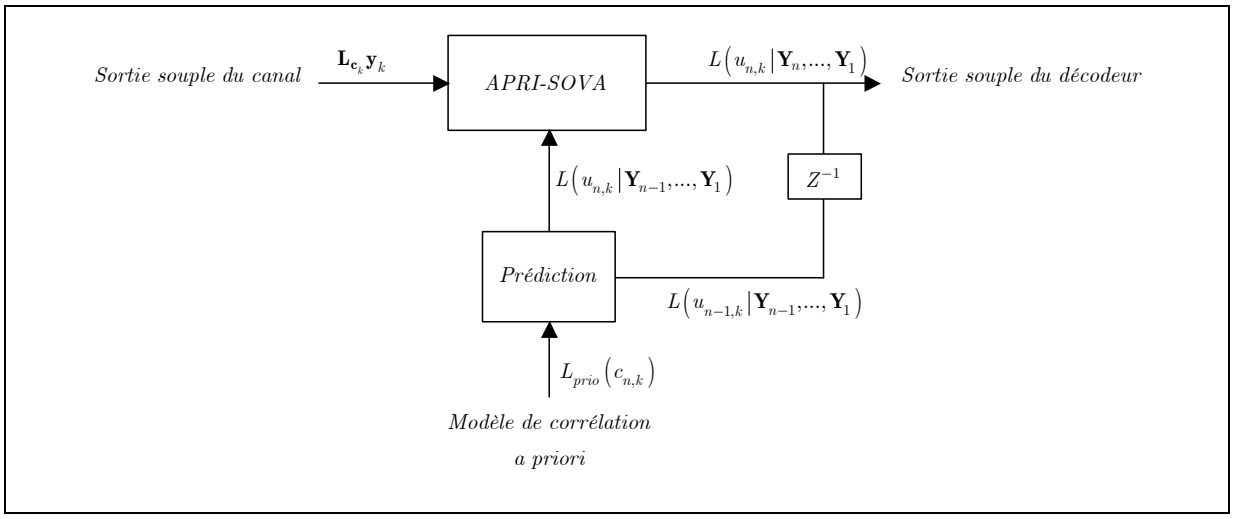

Figure 6.5 : Prédiction inter-trame au niveau des bits d'information

Le mécanisme de *prédiction inter-trame* mis en œuvre ici au niveau des bits individuels $u_{n,k}$ prend en compte l'information de *confiance* dans la valeur *décodée* $\hat{u}_{n-1,k}$ pour la trame passée, il évite ainsi le phénomène de propagation d'erreur.

Le paramètre qui gouverne la prédiction inter-trame est la probabilité *a priori* de changement de signe, représentée par l'intermédiaire de $L_{prio}\left(c_{n,k}\right)$. On peut envisager deux manières d'estimer ce paramètre :

- **Prédiction fixe**

La probabilité *a priori* de changement de signe peut être apprise en *moyenne* sur une base de parole. Ceci aboutit à un schéma de prédiction inter-trame basé sur des probabilités de transitions $Pc_k^{(prio)}$ *invariantes* au cours du temps n, comme le sont les modèles AK1 de prédiction sur les index de quantification, étudiés aux chapitres précédents.

- **Prédiction adaptative**

Le schéma de prédiction proposé par [Hagenauer, 1995] est, lui, basé sur une estimation adaptative de la valeur souple $L_{prio}\left(c_{n,k}\right)$ du bit de changement de signe. L'argument mis en avant par Hagenauer est que la parole est un processus non-stationnaire et qu'il faut actualiser le modèle de prédiction, autrement dit $L_{prio}\left(c_{n,k}\right)$, en fonction de la statistique *à court-terme* de la parole transmise. Il propose pour cela une méthode empirique consistant à augmenter ou diminuer d'un certain facteur la valeur souple $L_{prio}\left(c_{n,k}\right)$, en fonction des changements de signes observés entre les valeurs *décodées* $\hat{u}_{n-1,k}$ et $\hat{u}_{n,k}$. La valeur absolue de $L_{prio}\left(c_{n,k}\right)$ est bornée de manière à ne pas diverger dans les périodes très stationnaires comme les plages de silence. Cet algorithme a été appliqué par [Hindelang et al., 1997] au GSM Full Rate.

6.3.2.2 Lois marginales calculées à partir de la loi de l'index de quantification

Cette méthode part de l'idée qu'il est préférable de modéliser la redondance résiduelle au niveau des index de quantification plutôt qu'entre bits individuels $u_{n,k}$. Cependant, la métrique de branche (6.14) de l'APRI-VA s'exprime uniquement en fonction du bit d'information $u_{n,k}$, on choisit donc ici une voie intermédiaire consistant à prédire *le bit* $u_{n,k}$ de la trame n à partir de la valeur à la trame $n-1$ de *l'index de quantification* auquel il est associé [Fingscheidt et al., 2000].

Considérons un index de quantification i_n à la trame n codé par la combinaison de bits $\mathbf{b}_n$ (ou « mot de code source ») :

$$i_n \leftrightarrow \mathbf{b}_n = \left[b_{n,0},...,b_{n,M-1} \right] \tag{6.18}$$

Après multiplexage avec les bits codant les autres paramètres (cf. Figure 6.1), on obtient la trame $\mathbf{u}_n$ des bits d'information. On notera u_{n,k_m} le bit correspondant au bit $b_{n,m}$ à l'intérieur de la trame $\mathbf{u}_n$:

$$b_{n,m} \underset{\text{démultiplexage}}{\overset{\text{multiplexage}}{\rightleftarrows}} u_{n,k_m} \tag{6.19}$$

où k_m désigne la position du bit d'information dans la trame $\mathbf{u}_n$.

La corrélation inter-trame est alors modélisée par les probabilités de transition :

$$p\left(u_{n,k_m} = \xi \middle| i_{n-1}\right) = p\left(b_{n,m} = \xi \middle| \mathbf{b}_{n-1}\right) \tag{6.20}$$

définies pour chaque bit $b_{n,m}$ codant l'index de quantification i_n. Ces probabilités *a priori* se déduisent directement des probabilités de transition entre valeurs des *index de quantification* $p\left(i_n \middle| i_{n-1}\right)$, ou de manière équivalente, entre mots de « code source » $\mathbf{b}$ associés, soit $p\left(\mathbf{b}_n \middle| \mathbf{b}_{n-1}\right)$:

$$p\left(b_{n,m} = \xi \middle| \mathbf{b}_{n-1}\right) = \sum_{\mathbf{b}_n \in I_m(\xi)} p\left(\mathbf{b}_n \middle| \mathbf{b}_{n-1}\right) \tag{6.21}$$

avec :
$$p\left(\mathbf{b}_n = \mathbf{b}^{(i)} \middle| \mathbf{b}_{n-1} = \mathbf{b}^{(i')}\right) = p\left(i_n = i \middle| i_{n-1} = i'\right) \tag{6.22}$$

et où $I_m(\xi)$ désigne l'ensemble des valeurs d'index pour lesquels le bit b_m est égal à ξ.

Les probabilités de transition $p\left(i_n \middle| i_{n-1}\right)$ correspondent au modèle AK1 présenté aux chapitres précédents pour le décodage de parole souple. Elles sont apprises sur une base de données de parole, le modèle de prédiction (6.20) est donc *invariant* dans le temps.

La probabilité *a priori* du bit $b_{n,m}$ s'exprime alors à partir du modèle de corrélation inter-trame (6.20) selon :

$$p\left(b_{n,m}\left|\mathbf{Y}_{n-1},...,\mathbf{Y}_1\right.\right) = \sum_{\mathbf{b}_{n-1}} p\left(b_{n,m}\left|\mathbf{b}_{n-1}\right.\right) p_{unif}\left(\mathbf{b}_{n-1}\left|\mathbf{Y}_{n-1},...,\mathbf{Y}_1\right.\right) \tag{6.23}$$

La probabilité *a posteriori* $p_{unif}\left(\mathbf{b}_{n-1}\left|\mathbf{Y}_{n-1},...,\mathbf{Y}_1\right.\right)$ à la trame $n-1$ s'obtient à partir des probabilités *a posteriori* des bits $b_{n-1,m}$ lesquelles se déduisent de la valeur souple en sortie du décodeur canal (SOVA) à la trame $n-1$. Plus précisément, on a pour une trame n donnée :

$$p_{unif}\left(\mathbf{b}_n\left|\mathbf{Y}_n,...,\mathbf{Y}_1\right.\right) = \prod_{m=0}^{M-1} p\left(b_{n,m}\left|\mathbf{Y}_n,...,\mathbf{Y}_1\right.\right) \tag{6.24}$$

en supposant indépendantes[65] les probabilités d'erreurs pour les bits $\left[b_{n,0},...,b_{n,M-1}\right]$.

La probabilité (6.24) est à rapprocher de la *vraisemblance* calculée par le décodeur de parole souple (cf. chapitre 3). Plus précisément, la probabilité (6.24) se réduit à la vraisemblance du mot de « code source » $\mathbf{b}_n$, ou de manière équivalente de l'index i_n, dans le cas ou on utilise un décodeur canal sans *a priori* puisqu'on a alors :

$$p\left(b_{n,m}\left|\mathbf{Y}_n,...,\mathbf{Y}_1\right.\right) = p\left(b_{n,m}\left|\mathbf{Y}_n\right.\right) \propto p\left(\mathbf{Y}_n\left|b_{n,m}\right.\right) \tag{6.25}$$

Dans le cas présent, le décodeur canal exploite la probabilité *a priori* des bits individuels $b_{n,m}$ sachant les trames reçues précédemment (corrélation inter-trame), la probabilité (6.24) s'interprète alors comme la probabilité *a posteriori* du mot de « code source » $\mathbf{b}_n$ (resp. de l'index i_n) sachant les trames reçues précédemment et la trame reçue à l'instant n. Cependant, cette probabilité *a posteriori* ne prend pas en compte la distribution *conjointe a priori* du mot de « code source » $\mathbf{b}_n$ (*non-uniformité* de l'index i_n) mais seulement celles des bits $b_{n,m}$, supposés *indépendants* entre eux. Elle est donc à voir comme un intermédiaire entre une *vraisemblance* et la probabilité *a posteriori* telle que calculée par le modèle AK1 au chapitre 3.

---

[65] L'hypothèse d'indépendance des erreurs en sortie du décodeur canal est vérifiée si les bits considérés n'appartiennent pas à la même région de décision de l'algorithme de Viterbi. On considère que c'est le cas statistiquement lorsqu'ils sont distants d'au moins 5 fois la longueur de contrainte du code. Le multiplexage permet, entre autres, de satisfaire cette condition.

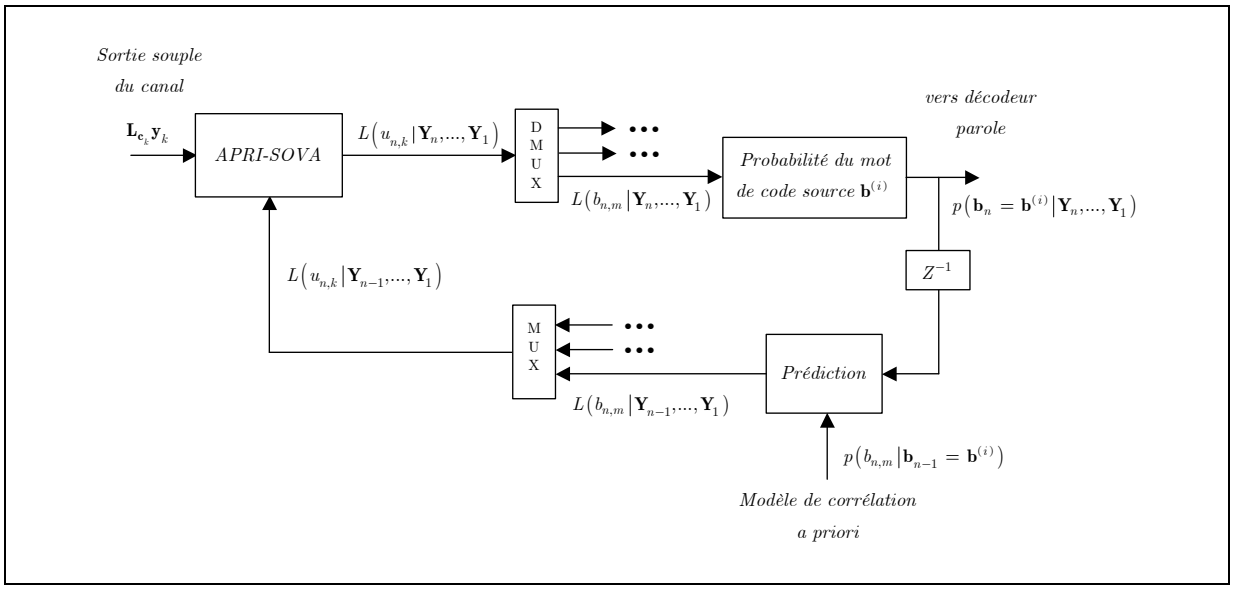

Figure 6.6 : Prédiction inter-trame des bits d'information à partir des index de QV

Finalement, la *probabilité a priori* (6.23) permet le calcul de la *valeur souple a priori* du bit $b_{n,m}$, ou de manière équivalente, du bit u_{n,k_m} après démultiplexage. Le schéma de cette méthode de prédiction *inter-trame* est représenté Figure 6.6.

6.3.3 Discussion

Nous étudierons, dans la suite de ce document, les performances de ces méthodes appliquées au codeur GSM EFR. On peut cependant discuter de la pertinence des modèles utilisés par ces méthodes et de leur hypothèses implicites.

Le premier point est le choix du niveau auquel représenter la corrélation *inter-trame*. Il est évident que les probabilités de transition $p(b_{n,m}|\mathbf{b}_{n-1})$ entre bit $b_{n,m}$ et mot de « code source » $\mathbf{b}_{n-1}$ (autrement dit, valeur de l'index de quantification i_{n-1}) caractérisent mieux la redondance que les probabilités de transition entre bits individuels $p(u_{n,k}|u_{n-1,k})$. Néanmoins c'est au prix d'une complexité nettement plus élevée pour l'actualisation de la probabilité *a priori* (6.23) à chaque trame décodée. A contrario, le modèle très simple de corrélation temporelle par *bit de changement de signe* peut se justifier lorsqu'on a optimisé l'attribution des indices de quantification de manière à minimiser la distance dans GF2 (distance de Hamming) entre deux éléments « proches » du dictionnaire (*Index Assignement*). Dans un tel cas, la corrélation temporelle au niveau des paramètres (centroïdes du dictionnaire) se traduira par *l'invariance* au cours du temps d'une partie des bits codant l'index associé aux centroïdes.

Le second point est le choix d'un prédicteur[66] fixe ou adaptatif. On peut s'attendre à ce qu'une méthode de prédiction fixe possède un pouvoir de correction limité puisqu'elle exploite une redondance

[66] Par prédicteur, on entend ici le modèle de corrélation basé sur les probabilités de transition *a priori*.

moyennée sur une base de parole. Cependant, la stratégie adaptative proposée dans [Hagenauer, 1995] est susceptible d'engendrer des erreurs lors des *transitions* entre segments de parole de nature très différentes (voisé / non voisé, parole /silence) puisqu'elle se contente d'une adaptation *a posteriori* du prédicteur (probabilité de changement de signe).

Pour finir, on insistera sur le fait qu'aucun des modèles présentés ici ne prend en compte la corrélation entre bits $b_{n,m}$ codant le même index de quantification i_n. La prise en compte de cette corrélation *intra-trame* permettrait de modéliser la non-uniformité des index de quantification eux-mêmes. C'est l'objet des approches présentées dans la seconde partie de cet état de l'art.

6.4 Corrélation intra-trame entre bits

La redondance au niveau des bits, considérés séparément, n'est qu'une conséquence de la redondance au niveau des paramètres (resp. index de quantification) observée en sortie de codeur parole. Au chapitre 4, nous avons caractérisé cette redondance par la probabilité *a priori* $p(i_n)$ qui peut être soit fixe (non-uniformité uniquement), soit actualisée à partir des données précédentes et de la probabilité de transition $p(i_n|i_{n-1})$ pour prendre en compte la corrélation temporelle. Pour modéliser cette redondance au niveau du *décodeur canal*, on doit donc exploiter la loi *jointe* sur les bits d'information codant l'index i_n :

$$\begin{aligned} p(i_n) &= p\Big(b_{n,0},...,b_{n,M-1}\Big) \\ &= p\Big(u_{n,k_0},...,u_{n,k_{M-1}}\Big) \end{aligned} \tag{6.26}$$

en reprenant les notations du paragraphe précédent.

Les bits associés à des *paramètres distincts* seront ici supposés *indépendants* à l'intérieur d'une même trame, c'est-à-dire qu'on ne modélise pas la corrélation résiduelle entre les différents paramètres du codeur. Dans tout ce qui suit, on s'intéressera à un seul paramètre du codeur, d'index de quantification associé i_n. Pour alléger les notations, on omettra l'indice temporel n lorsqu'il n'est pas nécessaire de faire référence explicite à la dépendance temporelle.

Le problème est alors de reformuler le critère MAP (6.1) à partir de la loi jointe (6.26) et d'en dériver un algorithme récursif permettant sa résolution dans la pratique.

6.4.1 Métrique de branche associée aux paramètres

On part ici à nouveau de la probabilité $\alpha_k^{(\ell)}$ du chemin partiel ℓ jusqu'à l'étape k, introduite en (6.4), et de l'équation :

$$\hat{\mathbf{q}} = \arg\max_{\ell} p\big(\mathbf{q}^{(\ell)}|\mathbf{Y}\big) = \arg\max_{\ell} \alpha_L^{(\ell)} \tag{6.5}$$

On a vu que le principe de la maximisation récursive mise en œuvre dans l'algorithme de Viterbi est que la probabilité du chemin partiel ℓ jusqu'à l'étape k se décompose deux probabilités indépendantes :

$$\alpha_k^{(\ell)} = \alpha_{k-1}^{(\ell)} f\big(\mathbf{y}_k, q_{k-1}^{(\ell)}, q_k^{(\ell)}\big) \tag{6.27}$$

où le second terme ne dépend que de la branche $\big(q_{k-1}^{(\ell)}, q_k^{(\ell)}\big)$ et des données reçues $\mathbf{y}_k$ associées à cette branche.

Autrement dit, la *métrique de branche* $\log\big(f\big(\mathbf{y}_k, q_{k-1}^{(\ell)}, q_k^{(\ell)}\big)\big)$ doit être *indépendante* des états parcourus par le chemin aux instants passés $k_j < k-1$ et futurs $k_j > k$, pour pouvoir formuler la métrique totale $\log\big(\alpha_L^{(\ell)}\big)$ du chemin ℓ comme l'accumulation des métriques de branches. Dès lors, il y a deux solutions pour intégrer la probabilité jointe (6.26) dans un calcul récursif de $\alpha_L^{(\ell)}$:

- Choisir un code convolutif tel que les bits $u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}$ codant un même valeur d'index i, soient associés à une même branche $\big(q_{k-1}^{(\ell)}, q_k^{(\ell)}\big)$ du treillis. On a alors :

$$\begin{aligned}\alpha_k^{(\ell)} &= \alpha_{k-1}^{(\ell)} p\big(\mathbf{y}_k \big| q_{k-1}^{(\ell)}, q_k^{(\ell)}\big) p\big(q_k = q_k^{(\ell)} \big| q_{k-1}^{(\ell)},...,q_1^{(\ell)}\big) \\ &= \alpha_{k-1}^{(\ell)} p\big(\mathbf{y}_k \big| q_{k-1}^{(\ell)}, q_k^{(\ell)}\big) p\Big(u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}\Big) \\ &= \alpha_{k-1}^{(\ell)} p\big(\mathbf{y}_k \big| \mathbf{x}_k^{(\ell)}\big) p(i)\end{aligned} \tag{6.28}$$

  La modification de la métrique de branche s'apparente alors à celle de l'APRI-VA (6.9) où la probabilité *a priori* est désormais celle de l'index de quantification i.

  Une approche de ce type a été mise en œuvre par [Alajaji et al., 1996] pour modéliser les 3 bits de poids fort des LSF du codeur FS CELP à 4.8kbits/s. Ceci suppose l'utilisation d'un codeur convolutif de rendement K/N où K est égal au nombre de bits dont on exploite les valeurs conjointes.

- Si les bits $u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}$ sont rentrés *séquentiellement* dans le codeur, on peut alors regrouper les branches contiguës associées à ces bits et calculer l'incrément de métrique sur le *tronçon* ainsi formé. Ceci revient à changer le pas de la récursion définissant $\alpha_k^{(\ell)}$:

$$\alpha_k^{(\ell)} = \alpha_{k-M}^{(\ell)} p\big(\mathbf{y}_{k-M+1},...,\mathbf{y}_k \big| q_{k-M}^{(\ell)},...,q_k^{(\ell)}\big) p\big(q_{k-M+1}^{(\ell)},...,q_k^{(\ell)} \big| q_{k-M}^{(\ell)},...,q_1^{(\ell)}\big) \tag{6.29}$$

  en supposant que les indices $\big(k_0,...,k_{M-1}\big)$ correspondent aux étapes $(k,...,k-M+1)$ dans le treillis. On a alors :

$$p\big(q_{k-M+1}^{(\ell)},...,q_k^{(\ell)} \big| q_{k-M}^{(\ell)},...,q_1^{(\ell)}\big) = p\Big(u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}\Big) \tag{6.30}$$

On obtient ainsi un nouvel incrément de métrique, associé au *tronçon* $\left(q_{k-M}^{(\ell)},...,q_k^{(\ell)}\right)$:

$$\begin{aligned} f\left(\mathbf{y}_{k-M+1},...,\mathbf{y}_k,q_{k-M}^{(\ell)},...,q_k^{(\ell)}\right) &= p\left(\mathbf{y}_{k-M+1},...,\mathbf{y}_k\left|q_{k-M}^{(\ell)},...,q_k^{(\ell)}\right.\right)p\left(u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}\right) \\ &= \prod_{k'=k-M+1}^{k} p\left(\mathbf{y}_{k'}\left|\mathbf{x}_{k'}^{(\ell)}\right.\right)p(i) \end{aligned} \tag{6.31}$$

Les étapes successives d'accumulation – maximisation de l'algorithme de Viterbi se font désormais par *groupes de bits* codant un même paramètre et non pour chaque bit d'information u_k. Ceci revient à définir un nouveau treillis, dont les branches sont associées aux bits $u_{k_0}^{(i)},...,u_{k_{M-1}}^{(i)}$. Dans ce nouveau treillis, la métrique de branche (6.31) s'apparente alors à celle de l'APRI-VA utilisant la probabilité *a priori* $p(i)$. Cette démarche de transposition du treillis dans le domaine des paramètres peut être étendue [Heinen et al., 2000] au décodage par l'algorithme de Bahl.

6.4.2 Métrique de branche conditionnée aux états précédents

Les deux méthodes présentées plus haut aboutissent donc à la même formulation d'un treillis dont les branches sont associées aux paramètres. L'algorithme APRI-VA étant alors applicable à ce treillis pour prendre en compte la probabilité *a priori* des paramètres (resp. index de quantification).

Cependant, dans les deux cas, la complexité augmente du fait de la croissance du nombre d'états (méthode 1), ou du nombre de branches du treillis (méthode 2).

De plus, aucune de ces méthodes n'est applicable au GSM sans modification de la norme du codeur. En effet, le GSM utilise un codeur convolutif de rendement $1/N$ et les bits en sortie du codeur parole ne sont pas rentrés séquentiellement dans le codeur convolutif. A contrario, le GSM applique une procédure de *ré-ordonnancement* des bits dont le principe est illustré Figure 6.7. Celle-ci permet une meilleure protection des bits « sensibles » en les redistribuant en début et en fin de la trame en entrée du codeur convolutif, c'est-à-dire là où la probabilité d'erreur au décodage est la plus faible[67].

Ce procédé n'est pas spécifique au GSM et se retrouve dans de nombreux systèmes de transmission mettant en œuvre un codeur convolutif fonctionnant par trames. En revanche, il engendre une *dispersion* des bits codant un même index de quantification sur toute la longueur de trame $\mathbf{u}$ (cf. Figure 6.7).

[67] L'état initial et final des chemins est imposé au codeur (par des bits de bourrage), ce qui a pour effet de réduire les degrés de liberté sur les chemins possibles en début et fin de trame.

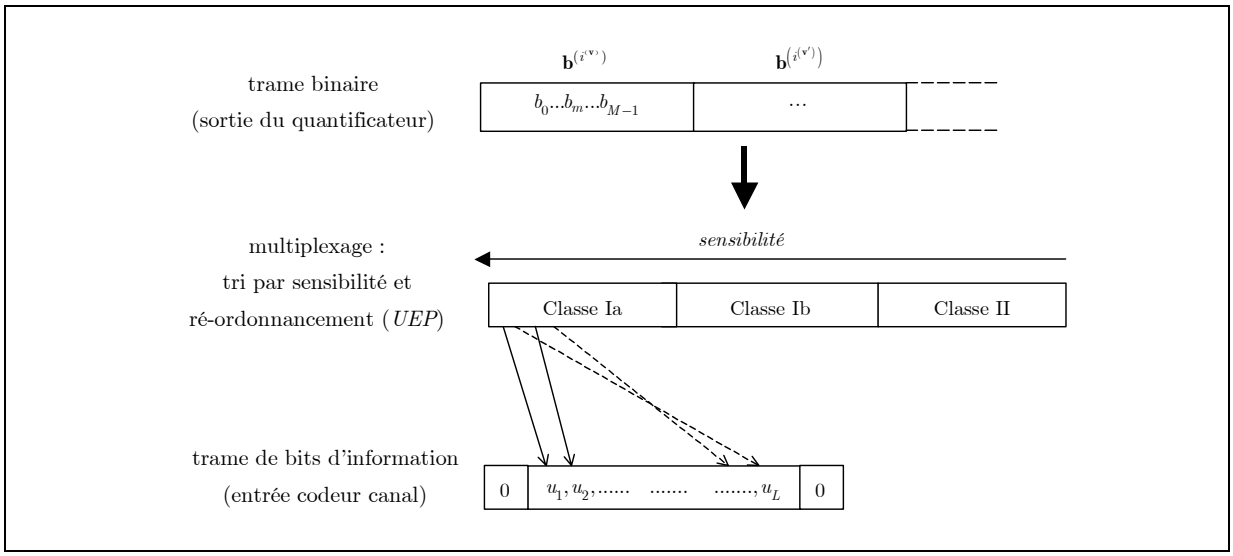

Figure 6.7 : Ré-ordonnancement des bits avant codage canal

Dans ce contexte, puisqu'il n'est pas possible de formuler une métrique de branche en fonction de la loi jointe $p\left(u_{k_0},...,u_{k_{M-1}}\right)$, [Heinen et al., 1997] propose une approche sous-optimale exploitant des lois conditionnelles. Plus précisément, la branche du treillis associée au bit d'information u_{k_m} est pondérée par la loi conditionnelle du bit u_{k_m} sachant les valeurs des bits $\left\{u_{k_j}^{(\ell)};k_j<k_m\right\}$ déjà décodés par le chemin survivant ℓ testé. Le calcul de métrique résultante correspond alors à une légère modification de l'APRI-VA. On a, pour $k=k_m$:

$$\begin{aligned} M_k^{(\ell)} &= M_{k-1}^{(\ell)} + \sum_{n=1}^{N} \log p\left(y_{k,n}\left|x_{k,n}^{(\ell)}\right.\right) + \log p\left(u_k = u_k^{(\ell)}\right) \\ \text{avec } p\left(u_k = u_k^{(\ell)}\right) &= p\left(u_{k_m} = u_k^{(\ell)}\left|\left\{u_{k_j}^{(\ell)};k_j<k_m\right\}\right.\right) \end{aligned} \tag{6.32}$$

Les lois conditionnelles $p\left(u_{k_m}\left|\left\{u_{k_j};k_j<k_m\right\}\right.\right)$ s'obtiennent à partir de la loi jointe (6.26).

Pour comprendre la nature de l'approximation posée par le calcul de métrique (6.32), on considère à nouveau la décomposition de la probabilité $\alpha_k^{(\ell)}$ du chemin partiel l aboutissant à l'état $q_k^{(\ell)}$, on a :

$$\begin{aligned} \alpha_k^{(\ell)} &= \alpha_{k-1}^{(\ell)} p\left(\mathbf{y}_k\left|q_{k-1}^{(\ell)},q_k^{(\ell)}\right.\right) p\left(q_k = q_k^{(\ell)}\left|q_{k-1}^{(\ell)},...,q_1^{(\ell)}\right.\right) \\ &= \alpha_{k-1}^{(\ell)} p\left(\mathbf{y}_k\left|\mathbf{x}_k^{(\ell)}\right.\right) p\left(u_k = u_k^{(\ell)}\left|u_{k-1}^{(\ell)},...,u_1^{(\ell)}\right.\right) \end{aligned} \tag{6.33}$$

Comme on suppose indépendants entre eux les groupes de bits codant différents paramètres, on a pour $k=k_m$:

$$p\left(u_k = u_k^{(\ell)}\left|u_{k-1}^{(\ell)},...,u_1^{(\ell)}\right.\right) = p\left(u_{k_m} = u_k^{(\ell)}\left|\left\{u_{k_j}^{(\ell)};k_j<k_m\right\}\right.\right) \tag{6.34}$$

d'où, pour $k = k_m$:

$$\alpha_k^{(\ell)} = \alpha_{k-1}^{(\ell)} p\left(\mathbf{y}_k \left| \mathbf{x}_k^{(\ell)} \right.\right) p\left(u_{k_m} = u_k^{(\ell)} \left| \left\{ u_{k_j}^{(\ell)}; k_j < k_m \right\} \right.\right) \tag{6.35}$$

ce qui, dans le domaine logarithmique, correspond à la métrique (6.32).

Le problème ici est que la *métrique de branche* à l'étape k dépend des états précédents parcourus par le chemin ℓ aboutissant à l'état $q_{k-1}^{(\ell)}$. La *maximisation récursive* de la métrique (6.32), par sélection d'un *unique chemin survivant* à chaque étape et pour chaque état n'est alors plus équivalente à la maximisation globale de la métrique du chemin complet. Autrement dit, la récurrence (6.32) maximise, à chaque étape k, la probabilité conditionnelle du bit u_k sans remettre en question les décisions prises aux étapes précédentes or ceci n'est pas équivalent à la maximisation de la loi jointe des bits.

On remarquera cependant que l'approximation effectuée par la récurrence (6.32) est d'autant plus faible que les bits $u_{k_0}, ..., u_{k_{M-1}}$ sont proches, c'est-à-dire appartiennent à la même *région de décision*. Le cas limite est celui pour lequel la distance entre les bits est inférieure ou égale à la longueur de contrainte du code, dans ce cas la maximisation récursive (6.32) est parfaitement valide.

6.4.3 Décodage canal en deux étapes

Les schémas de prédiction *inter-trame* présentés au §6.3.2.1 évitent la propagation d'erreur car ils prennent en compte la probabilité d'erreur des bits décodés à la trame précédente pour estimer la probabilité *a priori* du bit que l'on s'apprête à décoder dans la trame courante. Dans le cas de la corrélation *intra-trame*, le problème est que la probabilité d'erreur n'est disponible pour aucun bits avant le décodage complet de la trame, c'est pourquoi la métrique de Heinen fait intervenir les valeurs binaires des bits et non leur valeurs souples. Une solution, proposée par [Ruscitto et al., 1997], consiste à effectuer le décodage en deux étapes :

- Décodage préliminaire (SOVA) :

  La trame reçue $\mathbf{Y}_n$ est décodée une première fois par un décodeur canal à sortie souples *sans information a priori*. La sortie souple de ce décodeur est alors utilisée pour calculer les probabilités *a priori* individuelles des bits d'après le modèle de corrélation *intra-trame*.

- Décodage final (APRI-SOVA) :

  La trame $\mathbf{Y}_n$ est décodée une nouvelle fois par un décodeur de type APRI-SOVA exploitant les *probabilités a priori* des bits calculées à la première étape.

Le schéma proposé ici pour exploiter la corrélation intra-trame est donc directement transposé du schéma de prédiction inter-trame. La seule différence est que l'information *a priori* est estimée d'après un premier décodage de la trame courante $\mathbf{Y}_n$ et non d'après le décodage de la trame précédente $\mathbf{Y}_{n-1}$.

Les *modèles de corrélation* exploités pour la prédiction inter-trame sont également directement transposables à la prédiction intra-trame suivant ce schéma en deux étapes. En particulier, on retrouve les modèles exploitant la corrélation entre bits deux à deux et les modèles utilisant la probabilité marginale des bits sachant l'index de quantification associé.

6.4.3.1 Corrélation entre bits deux à deux

[Strauch et al., 1998] et [Ruscitto et al., 1997] modélisent la corrélation entre les deux premiers bits de poids forts u_{n,k_1} et u_{n,k_2} codant un même paramètre dans le cas d'une quantification scalaire. A la différence du modèle (6.15) proposé pour la corrélation temporelle, les probabilités de transition $p\left(u_{n,k_1}\middle|u_{n,k_2}\right)$ de leur modèle ne sont pas symétriques car ce n'est pas l'information d'invariance entre u_{n,k_1} et u_{n,k_2} qui est pertinente dans ce cas mais la *non-uniformité* du couple $\left(u_{n,k_1},u_{n,k_2}\right)$. Dans une démarche similaire à Hagenauer, les probabilités de transition $p\left(u_{n,k_1}\middle|u_{n,k_2}\right)$ sont *actualisées* à partir des décisions en sortie du décodeur canal (à l'issue des deux étapes). Ruscitto calcule pour cela un histogramme à la volée des valeurs *binaires* des bits u_{n,k_1} et u_{n,k_2}. Cependant l'argument de cette démarche adaptative n'est pas ici le suivi des non-stationnarités de la parole mais l'adaptation aux conditions *moyennes* du signal de parole transmis[68].

Ce modèle très simple prend donc en compte de manière incomplète la non-uniformité de l'index de quantification, il a été proposé pour le GSM Full Rate utilisant une quantification scalaire. Son application au cas d'une quantification vectorielle paraît moins justifiée. D'autre part, l'actualisation du modèle de prédiction (probabilités de transition) à partir des *décisions fermes* en sortie de codeur présente des risques de propagation d'erreur.

6.4.3.2 Loi marginale des bits sachant l'index de quantification

Afin de tirer plus pleinement parti de la non-uniformité *au niveau paramètre* (index de quantification), [Fingscheidt et al., 2000] modélise la relation entre un bit u_{k_m} codant l'index de quantification i, et l'ensemble des $M-1$ autres bits $\left\{u_{k_j};k_j \neq k_m\right\}$ codant le même index. Le modèle *a priori* est alors défini par les probabilités conditionnelles de la forme :

$$p\left(u_{k_m}\middle|\mathbf{b}_{\setminus m}\right) \tag{6.36}$$

[68] L'histogramme des valeurs des bits est estimé sur une fenêtre temporelle d'une durée de l'ordre de plusieurs secondes.

ou l'on note : $\mathbf{b}_{\setminus m} = \left\{ b_j \right\}_{0 \le j \le M-1 \text{ et } j \ne m}$

Les probabilités *a priori* (6.36) se déduisent la probabilité *a priori* du mot de code source $\mathbf{b}^{(i)}$ (index i) selon :

$$p\left(u_{k_m} \left| \mathbf{b}_{\setminus m} \right. \right) = \frac{p(\mathbf{b})}{p\left(\mathbf{b}_{\setminus m} \right)} = \frac{p(\mathbf{b})}{\sum_{b_m=0}^{1} p(\mathbf{b})} \tag{6.37}$$

avec $\mathbf{b} = b_0,...,b_m,...,b_{M-1}$.

La probabilité du mot de code source $p(\mathbf{b})$ coïncide avec celle de l'index i associé. Cette probabilité $p(i)$ est apprise sur une base de parole, de manière similaire au modèle AK0 étudié au chapitre 3.

A partir de la sortie souple du décodeur *préliminaire*, on peut estimer la *probabilité a posteriori* $p_{unif}(\mathbf{b}|\mathbf{Y})$ du mot de code source $\mathbf{b}$ (index i) selon la relation (6.24). Comme on l'a vu, cette probabilité *a posteriori* est homogène à une *vraisemblance* puisqu'on ne fait aucune hypothèse sur la distribution *a priori* de l'index i lors du décodage préliminaire. La prise en compte de cette distribution *a priori* se fait dans un second temps, au travers des relations (6.36) entre les bits codant i. Plus exactement, on calcule la *probabilité a priori* de chaque bit u_{k_m} selon :

$$p_{prio}\left(u_{k_m} | \mathbf{Y} \right) = \sum_{\mathbf{b}_{\setminus m}} p\left(u_{k_m} \left| \mathbf{b}_{\setminus m} \right. \right) p_{unif}\left(\mathbf{b}_{\setminus m} | \mathbf{Y} \right) \tag{6.38}$$

avec

$$p_{unif}\left(\mathbf{b}_{\setminus m} | \mathbf{Y} \right) = \sum_{b_m=0}^{1} p_{unif}(\mathbf{b}|\mathbf{Y}) \tag{6.39}$$

On remarquera que la probabilité (6.38) s'interprète comme la probabilité *a priori*[69] du bit u_{k_m} sachant les probabilités *a posteriori* des bits $\left\{ u_{k_j}; k_j \ne k_m \right\}$ estimées par le décodeur *préliminaire*. Dans la deuxième étape, cette probabilité est utilisée dans la métrique (6.9) de l'APRI-SOVA afin de réduire le taux d'erreur binaire par la prise en compte de la corrélation intra-trame entre bits.

En comparaison avec le décodeur proposé par Heinen, cette méthode permet de prendre en compte toutes les corrélations entre bits codant un même index, puisque le décodage en deux étapes supprime toute contrainte sur les positions respectives des bits dans le modèle de prédiction utilisé (6.36). De plus, la « prédiction » intra-trame, selon l'équation (6.38), prend en compte la probabilité d'erreur des bits estimée par le décodeur préliminaire, on évite ainsi toute propagation d'erreur. Cependant, if faut bien voir que le décodage en deux étapes est une technique *sous-optimale* qui ne permet pas un décodage *conjoint* des bits corrélés, puisque les branches du treillis du second décodeur sont

[69] Il peut sembler surprenant d'interpréter cette probabilité comme une probabilité a priori alors qu'elle est conditionnée aux données reçues $\mathbf{Y}$. Cependant, il faut bien voir que la vraisemblance du bit u_{k_m} est exclue de la sommation (6.38) qui correspond donc bien à une prédiction.

simplement pondérées par des lois *marginales* les bits d'informations $p_{prio}\left(u_{k_m}|\mathbf{Y}\right)$. Autrement dit, la corrélation *intra-trame* entre les bits n'est pas exploitée *au moment* du choix du chemin optimal mais à une étape antérieure à cette décision, pour établir une pondération initiale des branches qui n'est ensuite pas remise en cause au fur et à mesure du processus de décodage dans le treillis.

Enfin, le calcul de la probabilité *a priori* (6.38) présente une complexité élevée puisqu'il exige d'intégrer sur le dictionnaire de quantification pour chaque bit u_{k_m} considéré.

6.5 Bilan et discussion

Il est tout d'abord intéressant de positionner les techniques de décodage canal contrôlé par la source présentées ici par rapport aux méthodes de décodage souple de la parole abordées aux chapitres précédents. Dans les cas pratiques, il apparaît que le décodage canal contrôlé par la source (SCCD) ne peut exploiter pleinement la redondance résiduelle existante au *niveau des index de quantification*. D'autre part, le critère MAP mis en œuvre au décodeur canal est moins pertinent que le critère MMSE vis-à-vis de la minimisation de *l'impact subjectif* des erreurs de transmission. Dès lors, on peut considérer les mises en œuvre pratiques du SCCD comme *sous-optimales* au regard de l'exploitation de la redondance de source par rapport aux algorithmes de décodage souple de la parole. Des algorithmes de décodage *conjoint* source – canal exploitant la redondance au niveau des index de quantification et le critère MMSE ont été proposés [Heinen et al., 2000] mais ces méthodes se révèlent complexes (dimension du treillis, algorithme de décodage de Bahl). De plus, elles supposent implicitement la suppression du multiplexage des bits avant codage canal, or ce multiplexage permet en pratique d'améliorer la protection des bits sensibles (re-distribués en extrémités de trame).

Plus précisément, le SCCD paraît intéressant pour les niveaux de C/I intermédiaires, pour lesquels seul un nombre limité de chemins du treillis sont vraisemblables. La pondération additionnelle de ces chemins apportée par l'information *a priori* sur la source pourrait alors permettre une réduction effective du taux d'erreur. Pour des C/I plus faibles, la stratégie « correction d'erreur » (décodage des séquences binaires au sens du MAP) n'est plus suffisante et il est préférable d'appliquer le critère MMSE au niveau des paramètres pour réduire l'impact subjectif des erreurs (décodage souple de parole). Ainsi, plutôt que de considérer le SCCD comme une alternative aux méthodes de décodage souple de parole présentées auparavant, nous l'envisagerons ici comme une technique *complémentaire*.

Dans cette optique, on recherchera des algorithmes de SCCD de *complexité limitée*. Ainsi, les méthodes basées sur les corrélations entre bits individuels, seront étudiées malgré leur sous-optimalité évidente. Comme on l'a mentionné, la modélisation de la corrélation temporelle au niveau des bits individuels peut se justifier dans le cas d'une quantification scalaire (gains et délai de pitch de l'EFR) ou, de manière plus générale, en concomitance avec une procédure de type « Index Assignement » [Hedelin et al., 1995]. Nous proposerons dans la suite, une technique de prédiction *intra-trame*, permettant de modéliser notamment la corrélation deux à deux entre bits de sous-trames successives ou de

paramètres distincts. L'intérêt de ces méthodes est donc d'exploiter, de manière sous-optimale mais très simple, une information de corrélation qui n'est modélisable qu'au prix d'une complexité élevée au niveau du décodeur souple de parole (lequel met en œuvre une approche optimale sur les *paramètres*).

A l'inverse, nous ne développerons pas ici les algorithmes utilisant la probabilité (6.24) des index de quantification en sortie de décodeur canal. D'une part, ces algorithmes présentent une complexité élevée et la nature du modèle *a priori* qu'ils exploitent est redondante avec celle du décodage souple de parole. L'intérêt de leur emploi *combiné* avec une technique décodage souple semble donc moins évident, et l'étude menée par [Fingscheidt et al., 2000] révèle même une diminution des performances (SNR mesuré sur les paramètres) lorsqu'on ajoute ces méthodes de SCCD en amont d'une technique de décodage souple comme celles présentées aux chapitres précédents.

En revanche, l'approche proposée par Heinen (6.32) paraît mieux justifier l'intérêt du SCCD à la place ou en combinaison avec un décodeur souple car la redondance résiduelle au niveau des index de quantification y est exploitée *conjointement* avec une information uniquement disponible au décodeur canal, c'est-à-dire l'effet mémoire induit par le code convolutif. En effet, la structure du treillis intervient dans la probabilité conditionnelle utilisée par la métrique. Cet algorithme, qui présente une complexité relativement limitée, fera également l'objet d'une étude dans la suite de ce document et nous en proposerons des améliorations permettant une meilleure prise en compte de la corrélation intra-trame.

Chapitre 7

Décodage canal contrôlé par la source : Proposition d'algorithmes

7.1 Introduction

Nous avons étudié plus particulièrement deux axes d'amélioration des techniques de décodage canal contrôlé par la source. Le premier porte sur les méthodes exploitant la corrélation deux à deux entre bits. La redondance en sortie du codeur parole n'est ici que très partiellement exploitée mais ces méthodes pourraient offrir une alternative à des méthodes plus optimales mais qui sont également beaucoup plus complexes. Dans ce cadre, nous proposons une technique permettant de prendre en compte la corrélation deux à deux entre bits d'une même trame, simultanément au processus de décodage de Viterbi. Ceci permet notamment d'utiliser une information supplémentaire sur la redondance au niveau « paramètre[70] » lorsque les deux bits considérés codent le même index de quantification.

Le second axe étudié concerne les algorithmes exploitant la loi jointe de l'ensemble des bits codant un même « paramètre ». La prise en compte de cette loi permet une modélisation optimale de la redondance au niveau « paramètre » mais les contraintes imposées par le codeur canal du GSM limitent l'emploi de tels algorithmes. Nous recherchons des extensions à ces méthodes ainsi qu'une alternative au décodage de Viterbi afin de contourner ces contraintes.

[70] Dans tout ce chapitre, nous nous plaçons au niveau du décodeur canal, et nous ne distinguerons pas entre « paramètre » et « index de quantification ». La redondance au niveau « paramètre » désigne ici la connaissance de la loi jointe des bits codant l'index de quantification du paramètre.

Toutes les méthodes proposées ici seront évaluées avec les deux critères du *taux d'erreur binaire* en sortie du décodeur canal et de la *qualité perçue* (note MOS estimée) de la parole décodée. Ceci nous permet d'établir une comparaison avec les performances des algorithmes de décodage parole à entrées souples développées au Chapitre 5, et d'étudier les différentes manières de combiner ces deux approches.

7.2 Etude de la prédiction au niveau des bits individuels

Nous proposons ici d'étudier, dans le cas du GSM EFR, l'apport des méthodes de décodage canal exploitant un *a priori* individuellement sur les bits. Ces méthodes, présentées au §6.3, sont sous-optimales vis-à-vis de la modélisation de la redondance résiduelle mais ont l'avantage de pouvoir être très facilement mises en oeuvre. Nous quantifions en premier lieu, la redondance résiduelle caractérisable à ce niveau des bits individuels puis appliquons la méthode de prédiction inter-trame proposée par [Hagenauer, 1995]. Une extension de cette méthode à la corrélation intra-trame et ne nécessitant pas de décodage canal en deux étapes est ensuite proposée.

7.2.1 Analyse de la redondance résiduelle au niveau bit

Au chapitre 4, nous avons caractérisé la redondance résiduelle en sortie du codeur parole au niveau des paramètres (ou plus exactement, des index de quantification associés). Il est clair qu'une partie de l'information de redondance est perdue en passant d'un modèle *a priori* exploitant une loi jointe (bits codant un même paramètre) à un modèle basé sur des lois marginales (bits individuels), néanmoins on a noté précédemment que pour certaines configurations, comme une quantification scalaire ou une indexation optimisée (*Index Assignment*), une redondance significative pourrait demeurer au niveau des bits individuels. Nous cherchons ici à mesurer cette redondance dans le cas du GSM EFR. Les bits étudiés sont ceux codant les paramètres principaux de l'EFR, ils sont rappelés par le Tableau 7.1.

$\mathbf{lsf}_k(m)$	bit de poids m codant l'index de QV du $k^{ième}$ *jeu de résidus LSF*
$gp_k(m)$	bit de poids m codant le *gain de dictionnaire adaptatif* (sous-trame k)
$gc_k(m)$	bit de poids m codant le résidu de *gain de dictionnaire fixe* (sous-trame k)
$lag_k(m)$	bit de poids m codant le *délai de pitch* (sous-trame k)

Tableau 7.1 : Notations des bits alloués aux paramètres (index de quantification)

Les bits codant le dictionnaire d'excitation ne sont pas étudiés, leur redondance étant *a priori* très faible[71]. On rappellera que les paramètres spectraux (LSF) sont quantifiés *conjointement* (QV) par jeux de 5 paires après avoir été (partiellement) décorrélés par une prédiction MA. Une quantification *scalaire* (QS) est par contre utilisée pour les gains[72] et le délai de pitch. La table d'allocation des bits pour ces paramètres est donnée en Annexe A.

Pour caractériser la redondance au niveau des bits individuels, nous avons exploité un extrait de la base de parole présentée au chapitre 4. Les données considérées ici ne sont plus les index de quantification en sortie du codeur parole mais les *trames binaires* **d** formées par multiplexage des bits codant les index et tri décroissant selon leur *sensibilité* (Table 6 de la norme GSM EFR). On considérera ici uniquement les 80 premiers bits de cette trame ordonnée par sensibilité, c'est-à-dire les bits pour lesquels l'impact subjectif d'une erreur est le plus important. Les corrélations inter-trame et intra-trame sont mesurées sur 6000 trames correspondant à des **périodes d'activité vocale uniquement**. Plus exactement, on calcule les corrélations normalisées suivantes :

Inter-trame :
$$r_k = \frac{1}{N}\sum_{n=1}^{N} d_{n,k} d_{n-1,k} \tag{7.1}$$

Intra-trame :
$$r_{k_i,k_j} = \frac{1}{N}\sum_{n=1}^{N} d_{n,k_i} d_{n,k_j} \tag{7.2}$$

où $d_{n,k}$ désigne le bit de position k dans la trame $\mathbf{d}_n$ et N est le nombre de trames utilisées pour l'estimation. Ces expressions supposent que le bit $d_{n,k}$ est à valeur dans $\{+1,-1\}$

Les résultats de cette analyse sont illustrées Figure 7.1 et Figure 7.2 pour les corrélations inter-trame et intra-trame, respectivement. Le Tableau 7.2 présente, pour les bits les plus corrélés, les correspondances entre les positions des bits sur ces figures (position dans la trame **d** ordonnée par sensibilité) et leur signification, c'est-à-dire *leur poids* dans l'index de quantification associé.

Sensibilité (position k dans la trame réordonnée d**)**	**Poids dans l'index de quantification associé**
1,2,3,4,5,6	$lag_1\,(8,7,6,5,4,3)$
7,8,9,10,11,12	$lag_3\,(8,7,6,5,4,3)$
13,14 et 15,16	$lag_2\,(5,4)$ et $lag_4\,(5,4)$
17, 19, 58, 59	$gp_k\,(3);\ k=1,..,4$
25, 26, 51	$\mathbf{lsf}_1\,(5,4,6)$
(18, 62) – (20, 63) – (60, 64) – (61,65)	$gc_k\,(4,3);\ k=1,..,4$

Tableau 7.2 : Correspondance entre position et signification des bits les plus corrélés

[71] Le dictionnaire d'excitation représente la partie non-modélisable de la parole, autrement dit « aléatoire ».

[72] Dans le cas du gain de dictionnaire fixe, c'est le résidu de prédiction MA dans le domaine logarithmique qui est quantifié (cf. Annexe A).

En ce qui concerne la corrélation temporelle, on constate que les bits les plus corrélés correspondent majoritairement aux bits de *poids forts* des index issus d'une quantification *scalaire*. Un seul bit associé aux LSF apparaît vraiment corrélé, il s'agit du bit de poids fort de l'index du premier jeu de résidus LSF.

On remarquera que la corrélation temporelle est essentiellement positive, ce qui signifie qu'elle correspond à *l'invariance* des bits entre deux trames successives. La modélisation de cette corrélation par l'intermédiaire de la probabilité de *changement de signe*, proposée par Hagenauer, est donc pertinente dans ce cas.

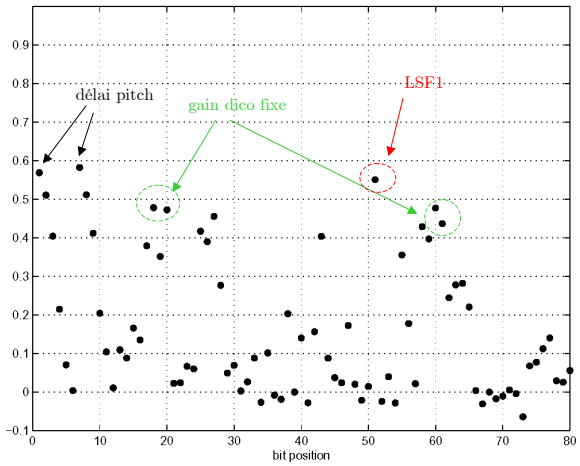

Figure 7.1 : Corrélation normalisée entre bits de trames successives du GSM EFR

Pour la corrélation intra-trame également, les bits les plus corrélés sont ceux associés à une quantification scalaire. Il s'agit alors des bits de poids forts d'un même index de quantification ou encore des bits de même signification à l'intérieur de sous-trames successives. Dans le premier cas, la corrélation entre bits traduit la *non-uniformité* de la distribution de l'index de quantification[73], dans le second cas, elle correspond à la *corrélation temporelle* entre sous-trames. En revanche, les bits associés à des paramètres distincts sont peu corrélés entre eux. Ceci rejoint l'analyse de redondance faite au chapitre 4 qui montrait notamment que les gains (dictionnaires fixe et adaptatif) étaient peu corrélés entre eux. Les seuls paramètres présentant une corrélation intra-trame significative sont les LSF

[73] La non-uniformité de la distribution d'un index implique que les bits codant cet index ne sont pas équiprobaux dans $\{+1,-1\}$. La non-uniformité d'un index résulte du fait que les cellules de quantification ne sont pas « adaptées » à la distribution du paramètre. C'est notamment le cas pour la quantification scalaire du gain du dictionnaire fixe comme l'illustre la table 4.4 présentée au Chapitre 4.

(relation d'ordre) mais cette corrélation n'apparaît plus au niveau des bits après quantification vectorielle.

En conclusion, la corrélation observée au niveau des bits individuels pour le GSM EFR concerne essentiellement les gains et le pitch. La redondance des paramètres spectraux (LSF) n'est que très partiellement prise en compte à ce niveau. Ceci semble limiter *a priori* les possibilités de rehaussement de la qualité perçue par des méthodes exploitant la corrélation entre bits. Cependant, on rappellera que selon le point de vue « correction d'erreur » qui est celui de SCCD, l'existence de bits redondants peut permettre une correction des autres bits situés dans une région proche du treillis (en raison de l'effet mémoire du codeur). Or, comme les bits sensibles sont regroupés en début et fin de trame, l'exploitation de la redondance des bits associés aux gains et au pitch pourrait induire une correction des bits codant les LSF.

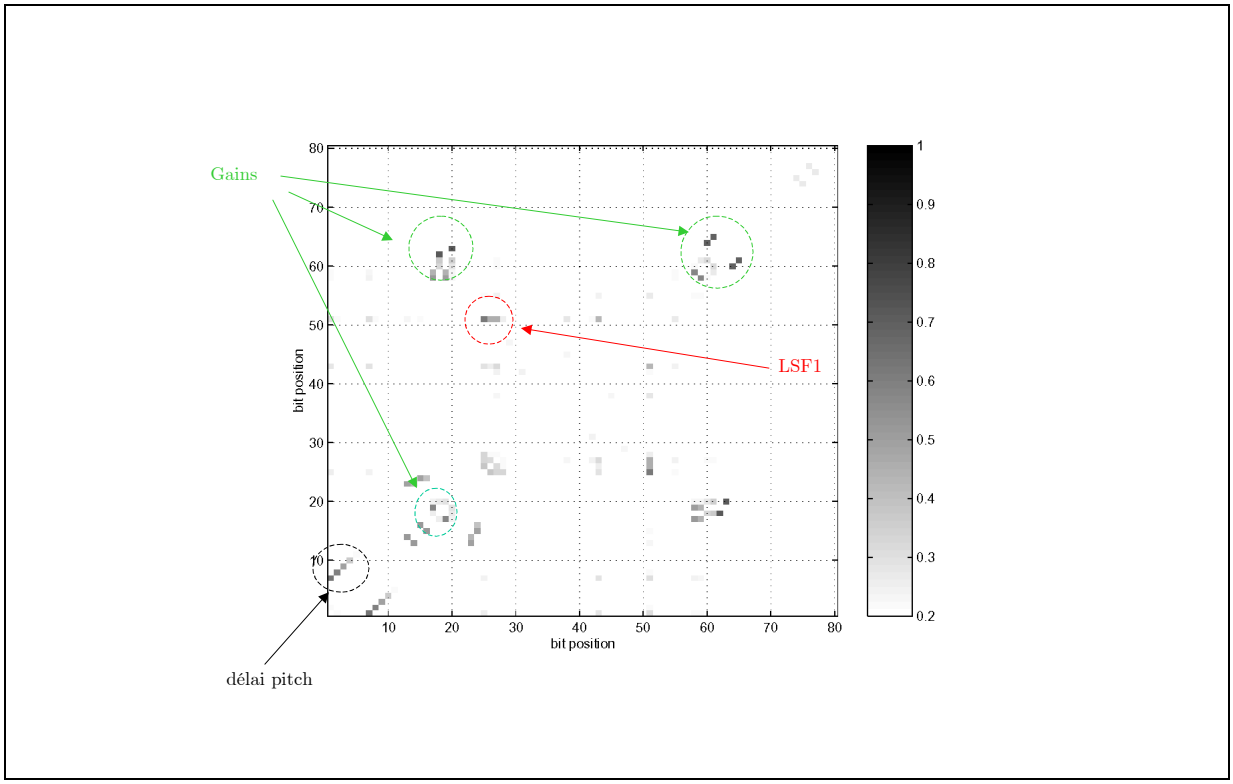

Figure 7.2 : Corrélation croisée (valeur absolue normalisée) entre bits d'une même trame (GSM EFR)

7.2.2 Prédiction inter et intra-trame au niveau bit pour le GSM EFR

Nous proposons ici une technique de décodage canal contrôlé par la source (SCCD) exploitant les corrélations entre bits individuels mises en évidence au paragraphe précédent. Nous partons pour cela de l'algorithme APRI-VA proposé par [Hagenauer, 1995] et qui consiste, comme on l'a vu, à intégrer à la *métrique de branche*, la valeur souple *a priori* $L_{prio}(u_k)$ du bit d'information, en plus des valeurs souples $\mathbf{L}_{\mathbf{c}k}\mathbf{y}_k$ en sortie d'égaliseur (canal équivalent) :

APRI-VA : $$M_k^{(\ell)} = M_{k-1}^{(\ell)} + \sum_{r=1}^{N} x_{k,r}^{(\ell)} L_{c_{k,r}} y_{k,r} + u_k^{(\ell)} L_{prio}(u_k) \qquad \textbf{(6.14)}$$

Dans une étape préliminaire, on considère le cas de la corrélation inter-trame et l'on étudie quel type de prédiction (fixe ou adaptative) de la valeur souple $L_{prio}(u_k)$ est la plus pertinente.

7.2.2.1 Conditions de simulations et critère d'évaluation

Les performances sont ici évaluées en termes de *taux d'erreur binaire* sur les 50 bits de la classe 1a du codeur EFR. On rappelle que cette classe recouvre les bits dont l'impact sur la qualité perçue est le plus élevé. Au niveau du décodeur de parole « classique » du GSM EFR, la présence *d'erreur* sur cette classe déclenche la *procédure de masquage* (cf. Annexe B). Le corpus de parole et les conditions de simulation utilisés pour générer la base de test sont résumés Tableau 7.3. Cette base de test est celle utilisée pour évaluer les performances (en termes de taux d'erreurs binaires) de l'ensemble des algorithmes étudiés dans ce chapitre.

2 séquences de 30 s de parole, multi-locuteurs, restreintes aux **périodes d'activité vocale uniquement**. Pour chaque séquence, on simule *3 itérations* de transmission au travers du canal.
La transmission dans le canal est simulée par l'application de *Pattern d'Erreurs fixes* avec un *offset aléatoire*.
Les *Pattern d'Erreurs* utilisés ont été générés par un modèle de canal type TU50 (modèle urbain, vitesse 50 km/h, Sauts de fréquences idéal) détaillé en Annexe C. Le seul paramètre que l'on fait varier est le C/I par pas de 1dB dans la gamme [2dB – 7dB].

Tableau 7.3 : Base de test utilisée pour l'estimation du TEB

7.2.2.2 Prédiction inter-trame

Les corrélations observées Figure 7.1 traduisent *l'invariance* des bits d'une trame à l'autre, c'est-à-dire l'absence de changement de signe. On reprend donc ici le modèle (6.15) de corrélation par l'intermédiaire d'un bit de changement de signe, soit au niveau des valeurs souples :

$$\begin{aligned} L_{prio}\left(u_{n,k}\right) &= L\left(u_{n,k}\middle|\mathbf{Y}_1^{n-1}\right) \\ &\simeq \operatorname{sign}\left(L_{prio}\left(c_{n,k}\right)\right)\operatorname{sign}\left(L\left(u_{n-1,k}\middle|\mathbf{Y}_1^{n-1}\right)\right)\min\left(\left|L_{prio}\left(c_{n,k}\right)\right|,\left|L\left(u_{n-1,k}\middle|\mathbf{Y}_1^{n-1}\right)\right|\right) \end{aligned} \tag{6.17}$$

où $c_{n,k}$ est le bit de changement de signe entre $u_{n,k}$ et $u_{n-1,k}$, et où on a fait apparaître explicitement le conditionnement aux sorties du canal afin de dissocier valeurs souples *a priori* $L\left(u_{n,k}\middle|\mathbf{Y}_1^{n-1}\right)$ à la trame n et valeurs souples *a posteriori* $L\left(u_{n-1,k}\middle|\mathbf{Y}_1^{n-1}\right)$ à la trame $n-1$. La valeur souple du bit de changement de signe est également une valeur *a priori* puisqu'elle provient du modèle de corrélation choisi.

Hagenauer actualise la valeur $L_{prio}\left(c_{n,k}\right)$ d'après la statistique à court-terme des décisions *fermes* $\hat{u}_{n,k}$ en sortie de décodeur canal. Nous pensons qu'il est préférable de prendre en compte la *fiabilité* des bits en sortie de décodeur afin d'éviter toute propagation d'erreur lors de l'actualisation de $L_{prio}\left(c_{n,k}\right)$. Nous proposons pour ce faire un algorithme [Veaux et al., 2000] un peu plus complexe que la méthode « empirique » d'Hagenauer. D'autre part, nous comparons la pertinence de cette démarche adaptative par rapport à une prédiction fixe.

7.2.2.2.a Probabilité de changement de signe actualisée d'après les sorties souples

Considérons la sortie souple du décodeur canal $L\left(u_{n,k}\middle|\mathbf{Y}_1^{n}\right)$ et la décision binaire associée :

$$\hat{u}_{n,k} = \operatorname{sign}\left(L\left(u_{n,k}\middle|\mathbf{Y}_1^{n}\right)\right) \tag{7.3}$$

On peut obtenir la valeur *a posteriori* du bit de changement de signe entre les valeurs *décodées* $\hat{u}_{n,k}$ et $\hat{u}_{n-1,k}$ selon :

$$c_{n,k} = \hat{u}_{n,k} + \hat{u}_{n-1,k} \tag{7.4}$$

ce qui permet de relier la valeur souple *a posteriori* $L_{post}\left(c_{n,k}\right)$ du bit de changement de signe aux valeurs souples en sortie du décodeur canal. En effet, de la même manière que la relation d'addition[74] entre bits (6.15) peut être approximée par la relation (6.17) entre valeurs souples [Hagenauer, 1995], on déduit de la relation d'addition (7.4), l'expression de la valeur souple $L_{post}\left(c_{n,k}\right)$ suivante :

---

[74] On rappelle ici que l'addition dans GF2 (addition binaire) correspond au « ou exclusif » dans $\{0,1\}$, ou encore à la multiplication dans $\{+1,-1\}$.

$$L_{post}\left(c_{n,k}\right) \simeq \text{sign}\left(L\left(u_{n,k}\middle|\mathbf{Y}_1^n\right)\right)\text{sign}\left(L\left(u_{n-1,k}\middle|\mathbf{Y}_1^{n-1}\right)\right)\min\left(\left|L\left(u_{n,k}\middle|\mathbf{Y}_1^n\right)\right|,\left|L\left(u_{n-1,k}\middle|\mathbf{Y}_1^{n-1}\right)\right|\right) \quad (7.5)$$

On peut alors adapter le modèle de corrélation *a priori* $L_{prio}\left(c_{n,k}\right)$ à la statistique à court-terme des sorties observées, en estimant la probabilité *a priori* de changement de signe $Pc_{n,k}^{(prio)}$ comme la moyenne à court-terme des probabilités *instantanées* (ou *a posteriori*) $Pc_{n,k}^{(post)}$:

$$Pc_{n,k}^{(prio)} = \lambda Pc_{n,k}^{(prio)} + (1-\lambda)\,Pc_{n,k}^{(post)} \quad (7.6)$$

où λ est un facteur d'oubli, et avec la relation (6.16) entre probabilité de changement de signe et valeur souple.

7.2.2.2.b Comparaison entre prédictions fixe et adaptative

La valeur du facteur d'oubli λ diffère selon *l'objectif assigné à l'actualisation du modèle* $L_{prio}\left(c_{n,k}\right)$. Ainsi, Hagenauer motive l'emploi d'une prédiction adaptative par la nécessité de suivre les variations de la statistique à court-terme de la parole, considérée comme « stationnaire par morceaux ». Ceci doit correspondre à des valeurs assez faibles du facteur d'oubli (constante de temps inférieure à 100 ms). A l'inverse, si l'objectif visé est simplement de s'adapter à la statistique moyenne du signal afin d'éviter les biais avec le modèle appris, par exemple pour s'adapter aux caractéristiques d'un locuteur, le facteur d'oubli doit prendre des valeurs plus proches de l'unité (constante de temps de l'ordre de la seconde). Dans tous les cas, on initialise le modèle à partir d'une valeur apprise sur une base de données. Cette valeur initiale de $L_{prio}\left(c_{n,k}\right)$ se déduit[75] de l'inter-corrélation moyenne illustrée Figure 7.1. On limitera également les déviations de $L_{prio}\left(c_{n,k}\right)$ par rapport à sa valeur apprise sur la base de données afin d'éviter une divergence dans les intervalles de non-activité vocale (silence, bruit de fond). Enfin, le prédicteur fixe correspond bien sur à un facteur d'oubli λ égal à l'unité.

Nous avons étudié les performances de la prédiction inter-trame pour différentes valeurs du facteur d'oubli. Nous présentons Figure 7.3, les résultats correspondant respectivement à des valeurs du facteur d'oubli de 50 ms et de 500 ms. La première valeur (50 ms) correspond à la stratégie de suivi de la statistique à court-terme de la parole, tandis que la seconde (500 ms) correspond plus à une adaptation aux conditions rencontrées (locuteur, filtrage, etc.). Les valeurs du facteur d'oubli supérieures à la seconde n'ont pu être envisagées ici en raison de la structure de la base de donnée utilisée (séquences constituées de phrases de locuteurs différents, concaténées).

---

[75] Comme les bits sont à valeurs dans $\{+1;-1\}$, l'addition des bits dans GF(2) équivaut au produit dans $\mathbb{R}$, et l'inter-corrélation moyenne des bits $u_{n,k}$ et $u_{n-1,k}$ correspond à la moyenne statistique du bit $c_{n,k}$. On en déduit la valeur souple a priori du bit $c_{n,k}$.

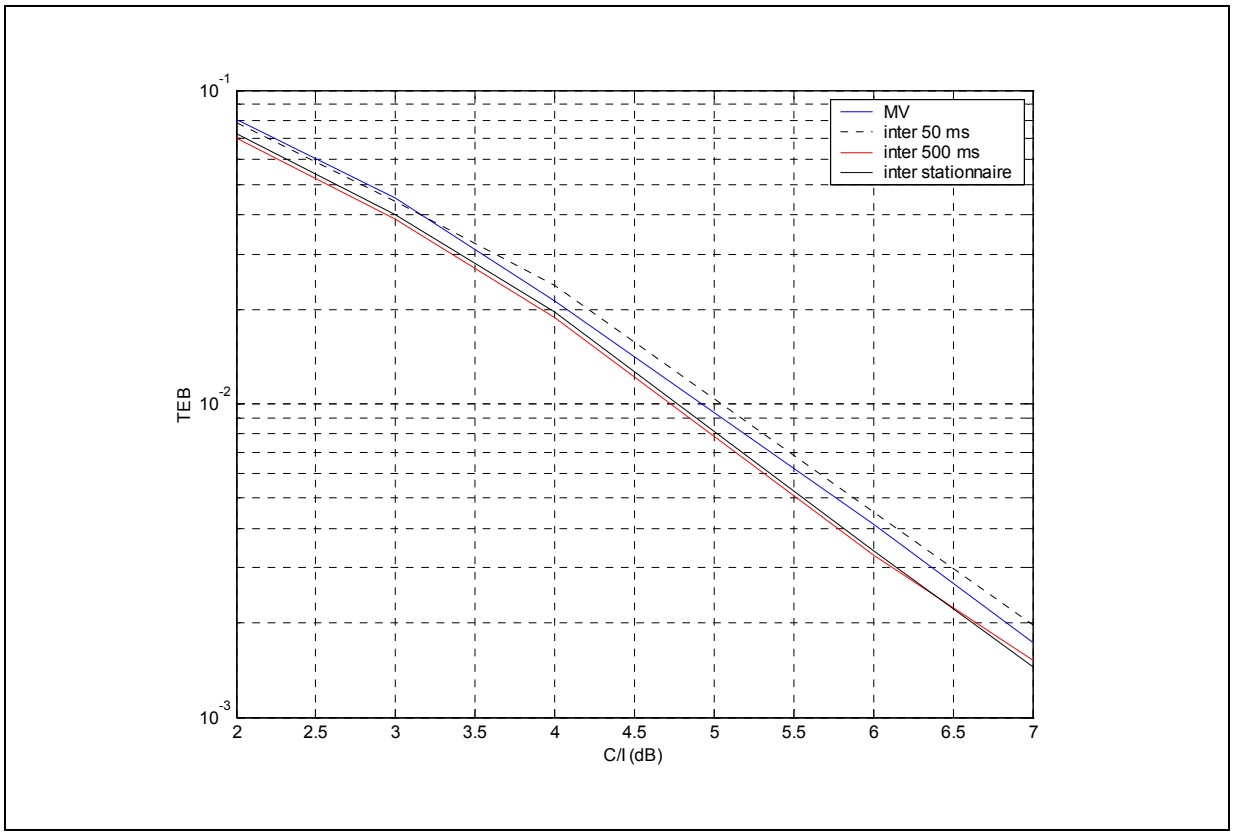

Figure 7.3 : Comparaison du décodeur de Viterbi classique (MV) avec l'APRI-VA exploitant la prédiction inter-trame adaptative, pour différents facteur d'oubli (TU50, Class1a)

L'adaptation à *très court-terme* du modèle de corrélation $L_{prio}\left(c_{n,k}\right)$ conduit à une dégradation du taux d'erreur binaire (TEB). Ce résultat n'est pas surprenant, comme mentionné au §6.3.3, cette stratégie engendre des erreurs aux transitions entre segments de statistiques très différentes et il apparaît finalement que les erreurs ainsi introduites sont plus nombreuses que les erreurs corrigées. Les performances du prédicteur *fixe* et de l'adaptation à *moyen terme* de $L_{prio}\left(c_{n,k}\right)$ sont, elles, très semblables et correspondent à une légère amélioration du TEB. Plus précisément, on observe dans ce cas, un gain[76] de l'ordre de 0.15 dB à TEB constant égal à 10^{-2}.

La base de test utilisée pour évaluer les performances des algorithmes limite certainement les différences entre le prédicteur fixe et le prédicteur adaptatif mais il ressort que l'actualisation du modèle *a priori* présente un intérêt pour diminuer la sensibilité de l'algorithme vis-à-vis d'un biais entre les conditions de parole transmise et celles de la base d'apprentissage. La possibilité d'une mise en œuvre adaptative simple est donc un avantage des méthodes exploitant le modèle de corrélation par bit de changement de signe (6.15). Dans les développements suivants, on se limitera néanmoins au cas d'un modèle *a priori fixe*, étant donné le faible écart de performance observé sur notre base de test, et

[76] Ce gain est l'accroissement du niveau d'interférences I qu'il faut appliquer pour retrouver le même taux d'erreur que le décodeur classique. Autrement dit, c'est le gain en robustesse vis à vis du niveau d'interférences.

afin de permettre une comparaison avec les méthodes exploitant un *a priori* (fixe) au niveau des index de quantification.

7.2.2.3 Prédiction intra-trame en parallèle au calcul de la métrique

On cherche maintenant à prendre en compte la corrélation résiduelle existante entre bits au sein d'une *même trame* **u** (on omet ici la référence à l'indice de-trame n pour alléger les notations).

On a vu qu'une part importante de cette corrélation intra-trame correspondait en fait à la corrélation temporelle entre sous-trames. Les approches basées sur une modélisation *jointe* des bits corrélés [Heinen et al., 2000] peuvent difficilement être étendues à cette corrélation entre sous-trames car la dimension de la loi jointe deviendrait trop élevée. Une méthode exploitant uniquement la corrélation entre *bits individuels* peut alors être justifiée afin de prendre en compte l'information de *stationnarité* entre sous-trames de manière simple. Cependant, les schémas existants de prédiction intra-trame entre bits individuels s'avèrent également complexes car ils nécessitent un décodage canal en deux étapes [Ruscitto et al., 1997]. Aussi, nous proposons ici une technique de prédiction intra-trame en *parallèle* au processus de décodage du chemin optimal dans le treillis [Veaux et al., 2000]. Cette technique peut être vue comme une extension de l'approche proposée par Hagenauer au cas intra-trame.

7.2.2.3.a Principe

On se restreint ici à la corrélation deux à deux entre bits $\{u_k ; u_{k'}\}$ avec une **contrainte supplémentaire d'ordre** $k > k'$ à l'intérieur de la trame **u**. Considérons l'extension du chemin candidat ℓ, à l'étape k, on utilise une métrique de branche similaire à celle de l'APRI-VA (ici formulée en faisant apparaître la probabilité plutôt que la valeur souple) :

$$M_k^{(\ell)} = M_{k-1}^{(\ell)} + \sum_{r=1}^{N} \log p\left(y_{k,r} \middle| x_{k,r}^{(\ell)}\right) + \log p_{prio}\left(u_k = u_k^{(\ell)}\right) \tag{6.9}$$

La probabilité *a priori* du bit u_k est ici estimée à partir de la valeur du bit $u_{k'}^{(\ell)}$ *décodée par le chemin candidat* ℓ à l'étape k' et des probabilités de transition *a priori* $p\left(u_k \middle| u_{k'}\right)$:

$$p_{prio}\left(u_k\right) = p\left(u_k \middle| u_{k'} = u_{k'}^{(\ell)}\right)\left(1 - \hat{p}_{k'}^{(\ell)}\right) + p\left(u_k \middle| u_{k'} = -u_{k'}^{(\ell)}\right)\hat{p}_{k'}^{(\ell)} \tag{7.7}$$

où $\hat{p}_{k'}^{(\ell)}$ est une estimée de la *probabilité d'erreur* associée à la décision $u_{k'}^{(\ell)}$ prise par le chemin ℓ à l'étape k'. Une estimée de cette probabilité d'erreur $\hat{p}_{k'}^{(\ell)}$ est justement disponible à chaque étape $k > k'$ lorsqu'on utilise l'algorithme SOVA de décodage à sorties souples. Cet algorithme et son interprétation sont détaillées en Annexe D. Une implémentation particulière de cet algorithme exploite des mémoires associées aux chemins survivants [Hagenauer et al., 1989]. Pour chaque chemin survivant

ℓ à l'étape k, le SOVA stocke, dans ces mémoires, les décisions fermes[77] $u_{k'}^{(\ell)}$ prises le long de ce chemin pour les instants $k' < k$, ainsi qu'une information de *fiabilité* associée à chacune de ces décisions :

$$\hat{L}_{k'}^{(\ell)} = \log \frac{1 - \hat{p}_{k'}^{(\ell)}}{\hat{p}_{k'}^{(\ell)}} \tag{7.8}$$

Les fiabilités (7.8) des décisions $u_{k'}^{(\ell)}$ prises le long du chemin ℓ sont ensuite *actualisées* à chaque nouvelle étape k, c'est-à-dire à chaque choix de chemin survivant (cf. Annexe D).

Plus précisément, les fiabilités (7.8) sont mises à jour pour les bits qui sont décodés différemment par le chemin survivant et le chemin éliminé, autrement dit, ceux qui seraient affectés par une erreur sur le choix du chemin survivant. Ces fiabilités sont initialisées au maximum de leur dynamique[78] tant que la distance entre les étapes k et k' est strictement inférieure à la *longueur de contrainte* $(\nu + 1)$ du codeur convolutif, puisque dans ce cas, le choix du chemin survivant n'affecte pas encore la valeur du bit $u_{k'}^{(\ell)}$. On remarquera que la probabilité *a priori* (7.7) se réduit alors à la probabilité de transition $p\left(u_k \middle| u_{k'} = u_{k'}^{(\ell)}\right)$. Dans ce cas, la métrique utilisée peut être vue comme une restriction de la métrique (6.32) proposée par Heinen, au couple de bits $\left\{u_k ; u_{k'}\right\}$. A l'inverse, lorsque la distance entre les étapes k et k' excède le *délai de décision* de l'algorithme, on considère que tous les chemins candidats décodent la même valeur pour le bit $u_{k'}$ et que la fiabilité (7.8) a convergé vers une valeur fixe. Cette valeur correspond à la sortie souple $L\left(u_{k'}\right)$ relâchée par le SOVA pour le bit $u_{k'}$. Dans ce second cas, notre métrique s'identifie très exactement à celle utilisée par Hagenauer pour l'inter-trame.

La Figure 7.4 illustre le mécanisme de prédiction intra-trame proposé. Cette technique exploite donc la corrélation intra-trame conjointement avec l'effet mémoire du code convolutif (contrainte sur les chemins dans le treillis). Elle prend en compte une information de fiabilité estimée par le SOVA lorsque la contrainte apportée par l'effet mémoire du code diminue (les chemins fusionnent). Ceci permet un décodage conjoint du couple de bits $\left\{u_k ; u_{k'}\right\}$ tant qu'il n'y a qu'un seul chemin possible entre u_k et $u_{k'}$ dans le treillis, puis limite la propagation d'erreur lorsque des chemins décodant une valeur différente de $u_{k'}$ ont fusionnés avant l'étape k.

---

[77] Ceci revient à stocker les états formant le chemin et évite ainsi l'opération de *traceback* une fois le chemin optimal décodé.

[78] Autrement dit, les probabilités d'erreur $\hat{p}_{k'}^{(\ell)}$ sont initialisées à zéro.

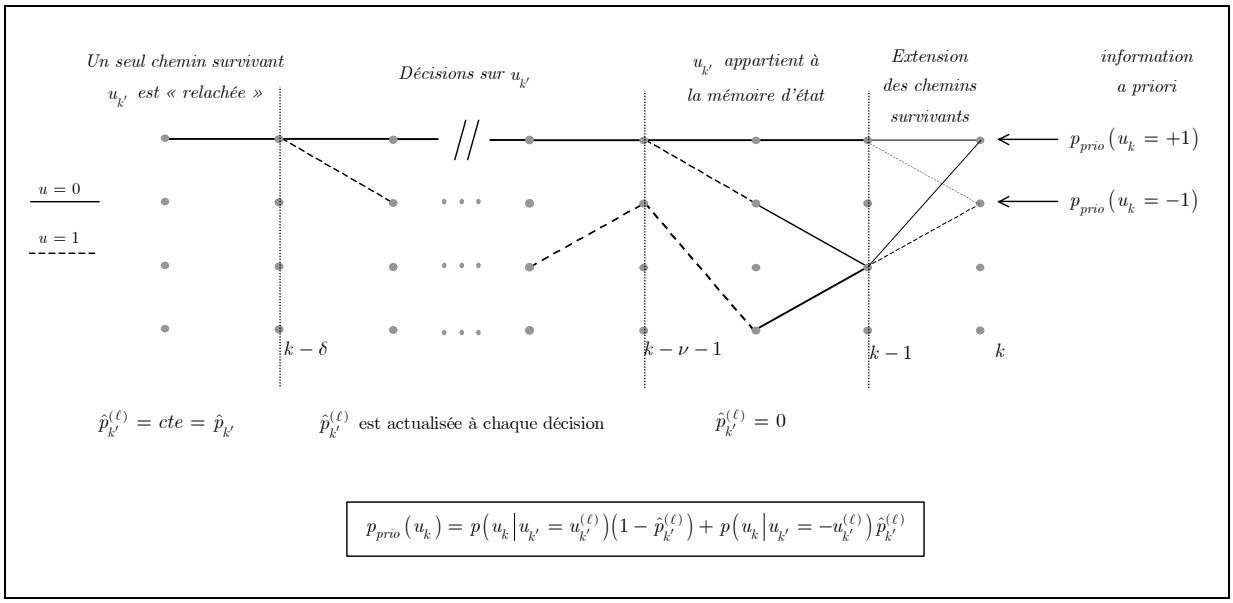

Figure 7.4 : Principe de la prédiction intra-trame avec prise en compte de la fiabilité des décisions

7.2.2.3.b Modèle de corrélation et combinaison des prédictions

Afin de simplifier le calcul de l'information *a priori* (7.7), on supposera que les probabilités de transition $p\left(u_k \middle| u_{k'}\right)$ sont *symétriques*. Ceci correspond au modèle de corrélation par *addition d'un bit de signe* déjà utilisé pour l'inter-trame. On a vu qu'un tel modèle était surtout justifié pour représenter l'invariance au cours du temps de la valeur d'un bit, cependant son emploi pour la corrélation intra-trame nous permet de ré-utiliser la relation très simple [Hagenauer, 1995] traduisant dans le domaine des valeurs souples, la relation d'addition binaire. Cette relation entre valeurs souples s'écrit dans le cas intra-trame :

$$L_{prio}(u_k) \simeq \text{sign}\left(L_{prio}\left(c_k^{k'}\right)\right)\text{sign}\left(L\left(\hat{u}_{k'}^{(\ell)}\right)\right)\min\left(\left|L_{prio}\left(c_k^{k'}\right)\right|, \left|L\left(\hat{u}_{k'}^{(\ell)}\right)\right|\right) \tag{7.9}$$

où $c_k^{k'}$ désigne le bit de changement de signe entre les bits d'information u_k et $u_{k'}$. On a ici omit l'indice temporel n de la trame considérée pour alléger les notations.

La valeur *a priori* $L_{prio}\left(c_k^{k'}\right)$ se déduit de la corrélation intra-trame calculée sur la base de données et illustrée Figure 7.2. Cependant, toutes les corrélations intra-trame entre bits ne peuvent être exploitées puisqu'on doit respecter la relation d'ordre $k > k'$. La Figure 7.5 représente les corrélations exploitables lorsqu'on applique cette *contrainte d'ordre*. La valeur représentée Figure 7.5 est la valeur souple du bit de changement de signe, elle est équivalente, à une transformation près, à la corrélation normalisée. Enfin, on rappellera que les abscisses et ordonnées correspondent ici à la position des bits triés par *sensibilité*, cette position diffère de la position k des bits dans la trame d'information **u**.

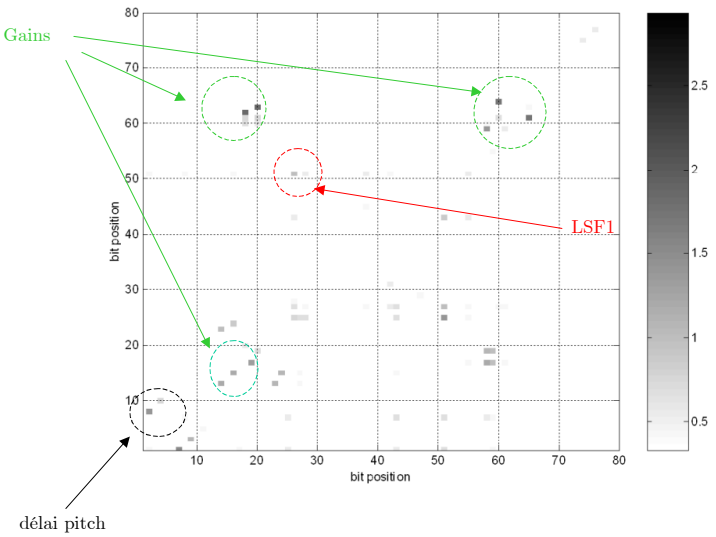

Figure 7.5 : Redondance entre bits exploitée par la prédiction intra-trame

On remarque que la symétrie de la Figure 7.2 représentant l'ensemble des corrélations intra-trame n'est plus présente sur la Figure 7.5 illustrant les corrélations *exploitées* par la prédiction intra-trame. Ceci est une conséquence de la relation d'ordre $k > k'$ nécessaire pour que la prédiction puisse se faire au fur et à mesure du calcul de la métrique par l'algorithme de Viterbi.

Les corrélations exploitées par la prédiction intra-trame concernent essentiellement les bits codant les gains (dictionnaires fixe et adaptatif) et le pitch. Ces corrélations correspondent, soit à des couples de bits codant un même index (non-uniformité), soit à des couples de bits de même position dans des sous-trames successives (mémoire). Ainsi, on peut retrouver un même bit membre de différents couples, ce qui traduit simplement le fait que la probabilité *jointe* d'un groupe de bits corrélés a été scindée en une série de couples (probabilités *marginales).*

Afin d'exploiter l'ensemble de ces corrélations intra-trame, on combine les prédictions concernant un même bit u_k. On reprend pour ce faire la relation additive utilisée par [Strauch et al., 1998] pour combiner les « prédictions » (valeurs souples *a priori*) inter-trame et intra-trame. Dans le cas intra-trame qui nous concerne ici, la valeur souple *a priori* $L_{prio}(u_k)$ intervenant dans la métrique s'obtient à partir des valeurs souples prédites *individuellement* depuis chaque bit $u_{k'}$ corrélé à u_k selon :

$$L_{prio}(u_k) = \sum_{k'<k} L\left(u_k \middle| u_{k'}\right) \tag{7.10}$$

où $L\left(u_k \middle| u_{k'}\right)$ désigne la valeur souple *a priori* du bit u_k, prédite d'après le bit $u_{k'}$ d'après la relation (7.9). La relation (7.10) est « empirique », puisqu'il n'est pas possible de dériver une relation de combinaison entre les probabilités sous-jacentes (probabilités conditionnées à des événements distincts). Elle s'obtient en considérant que les « informations » prédites depuis des bits $u_{k'}$ distincts s'additionnent, ce qui suppose implicitement *l'indépendance* des valeurs souples prédites $L\left(u_k \middle| u_{k'}\right)$. Ce n'est pas le cas en pratique, c'est pourquoi on limite l'amplitude de la valeur absolue de $L_{prio}\left(u_k\right)$ afin d'éviter qu'elle ne prenne des valeurs trop élevées lorsque les valeurs souples $L\left(u_k \middle| u_{k'}\right)$ sont redondantes entre elles.

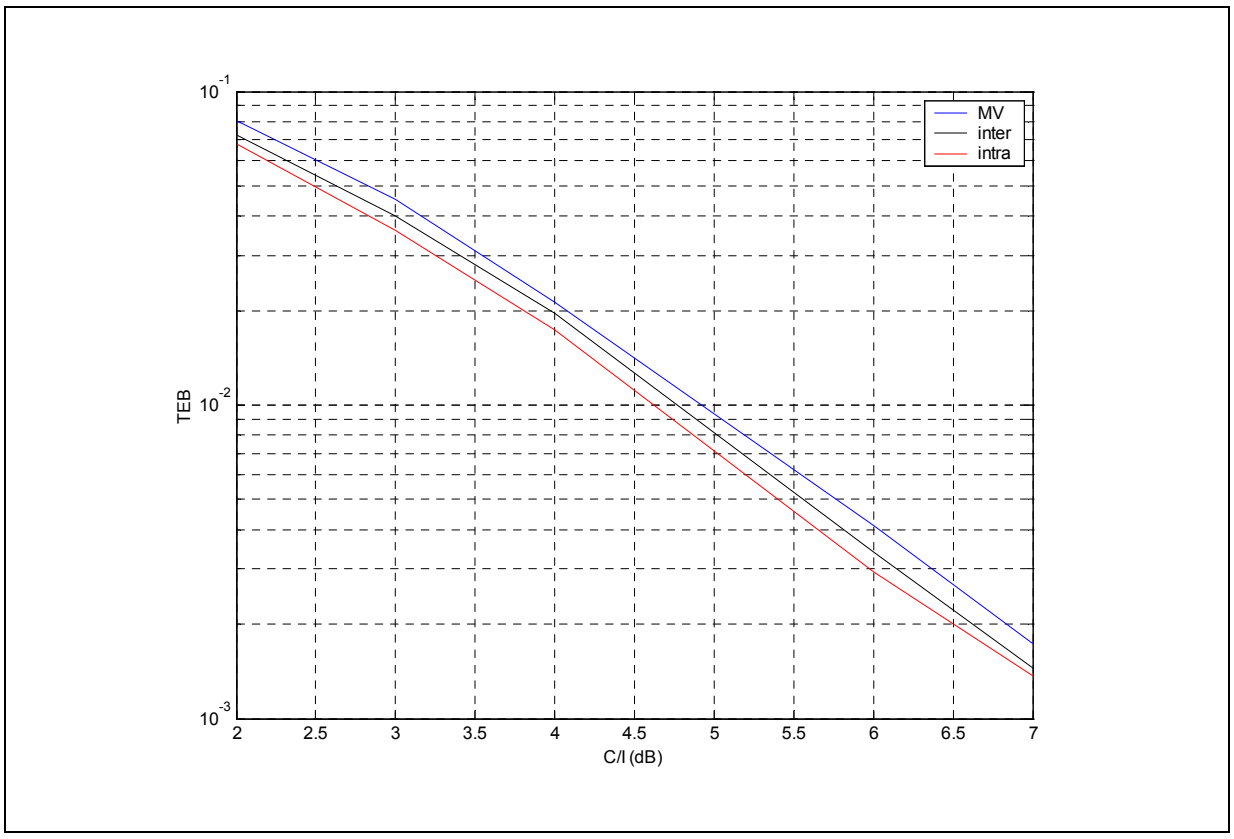

Figure 7.6 : TEB comparés des prédictions inter-, intra-trame et du décodeur de Viterbi classique (TU50, Class1a)

Les performances de la prédiction intra-trame proposée sont comparées Figure 7.6, à celles du décodeur de Viterbi classique (MV) et de la prédiction inter-trame (fixe) présentée au paragraphe précédent. A niveau d'erreur binaire constant, le gain en robustesse par rapport au décodeur classique est le double de celui obtenu par la prédiction inter-trame.

Le mécanisme de prédiction intra-trame étant une extension de la prédiction inter-trame, il est naturel de combiner les deux pour exploiter le maximum de corrélations résiduelles entre bits. Nous présentons les performances de cette combinaison au paragraphe suivant.

7.2.2.4 Combinaison inter-trame et intra-trame

Nous utilisons une relation d'additivé similaire à (7.10) pour combiner la valeur souple prédite à partir de la corrélation inter-trame $L\left(u_{n,k}\middle|u_{n-1,k}\right)$ et celles issues des corrélations intra-trame $L\left(u_{n,k}\middle|u_{n,k'}\right)$:

$$L_{prio}\left(u_k\right) = L\left(u_{n,k}\middle|u_{n-1,k}\right) + \sum_{k'<k} L\left(u_{n,k}\middle|u_{n,k'}\right) \tag{7.11}$$

La valeur souple résultante, bornée en amplitude absolue, est utilisée dans la métrique de l'APRI-VA. La Figure 7.7 illustre les performances de cette combinaison des prédictions inter-trame et intra-trame. On constate que le gain additionnel est assez faible, en particulier il ne correspond pas à la somme des gains apportés par la prédiction inter-trame seule et la prédiction intra-trame seule. Ceci provient certainement du fait que les valeurs souples $L\left(u_{n,k}\middle|u_{n-1,k}\right)$ et $L\left(u_{n,k}\middle|u_{n,k'}\right)$ sont partiellement redondantes. Il semble par exemple évident que la corrélation entre sous-trames apporte une information de même nature que celle entre trames (*invariance* du bit u_k au cours du temps).

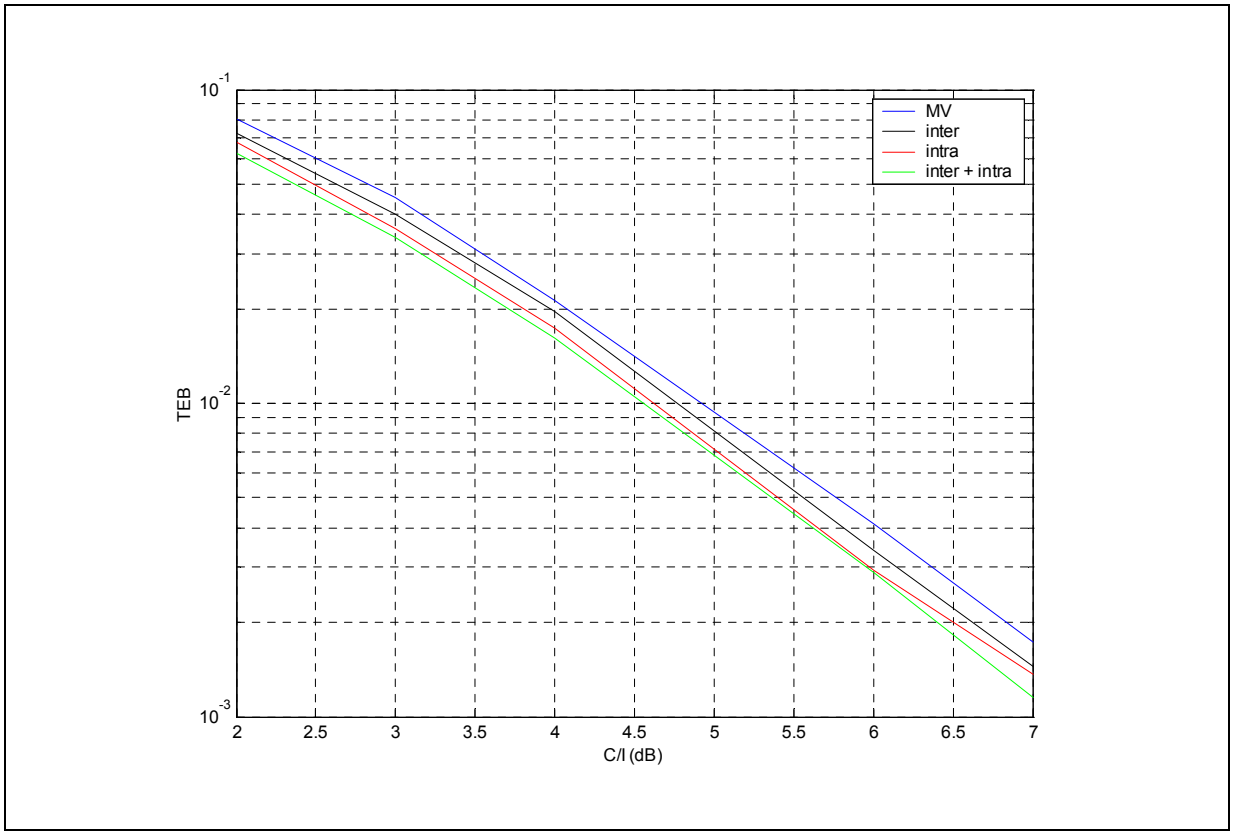

Figure 7.7 : TEB de la combinaison des prédictions inter- et intra-trame (TU50, Class1a)

Au final, la combinaison des prédictions inter-trame et intra-trame permet un gain de l'ordre de 0.4dB pour un TEB constant égal à 10^{-2} parmi les bits de la Classe 1a.

On a vu que les bits de la Classe 1a sont les plus importants pour la reconstruction de la parole, néanmoins, tous les bits de cette classe n'ont pas le même impact sur la qualité perçue et il est nécessaire de mesurer sur le signal de parole décodé lui-même, l'amélioration de qualité effectivement apportée par ces méthodes. Nous utilisons pour cela, l'algorithme PESQ, déjà mis en œuvre aux chapitres précédents pour évaluer les méthodes de décodage souple. Nous disposons ainsi d'un critère commun permettant de comparer les méthodes de SCCD, celles de décodage souple, et le masquage « classique » de l'EFR [GSM, 06.61].

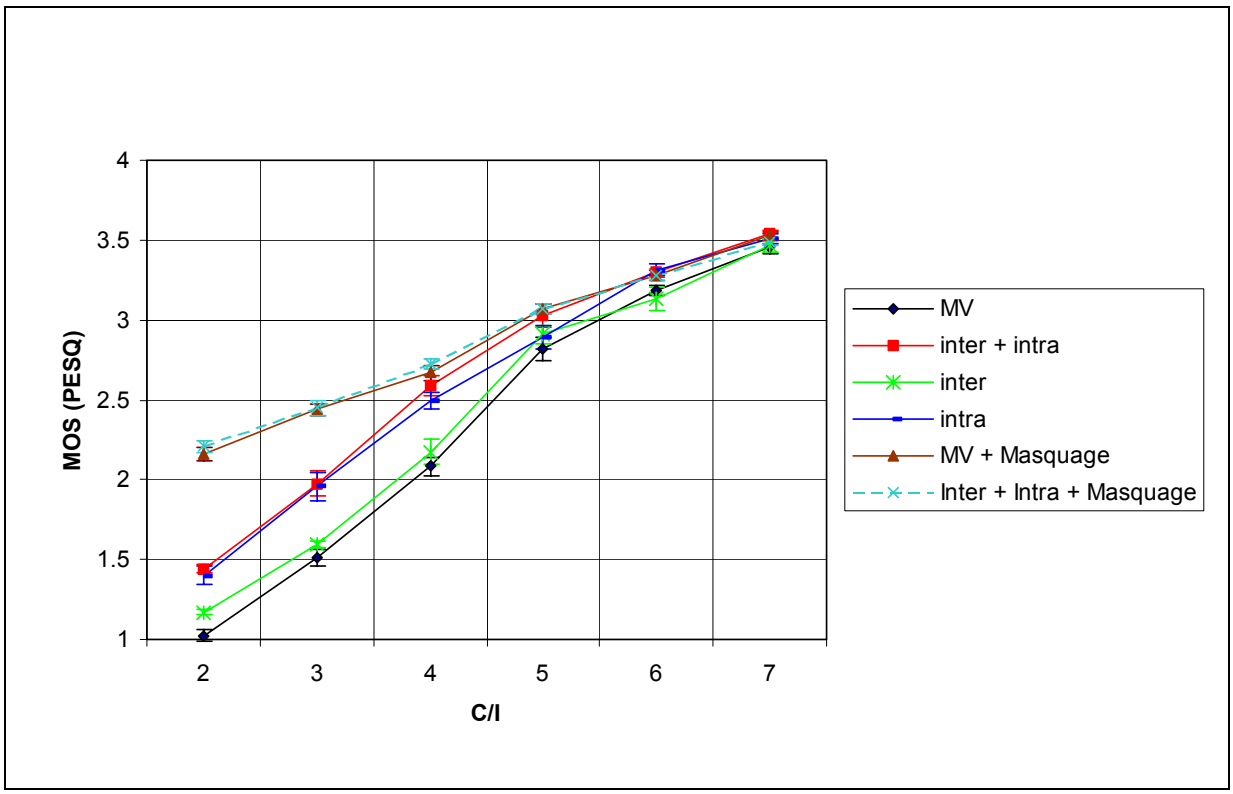

Figure 7.8 : Comparaison des méthodes de SCCD exploitant la redondance des bits individuels et du décodeur de Viterbi classique (MV), avec et sans substitution de trame (notes MOS *estimées*, TU50)

Nous avons tout d'abord comparé l'APRI-VA avec prédiction inter-trame et intra-trame au décodeur de Viterbi classique, *en débrayant le mécanisme de substitution de trame* au décodeur parole. Ceci permet d'évaluer dans quelle mesure les bits corrigés par les méthodes proposées impactent la qualité perçue, et complète la comparaison entre les performances respectives des algorithmes. Nous avons également étudié leurs performances avec la procédure de masquage mise en œuvre au niveau du décodeur parole. Les courbes de notes MOS estimées par l'algorithme PESQ sont reportées Figure 7.8, où l'extension (+ Masquage) signifie que le mécanisme de déclenchement de la procédure de masquage par l'indicateur BFI (détection de trame perdue) est activé.

Au niveau de leurs performances dans l'absolu (BFI débrayé), on retrouve une hiérarchie identique à celle observée précédemment sur le TEB, entre les prédictions inter-trame, intra-trame, et combinaison des deux. L'apport de la corrélation intra-trame est cependant plus marqué au niveau de la qualité

perçue (MOS estimée) qu'au niveau du taux d'erreur binaire (TEB). Nous interprétons ce résultat comme la preuve qu'une information sur la *non-uniformité* au niveau paramètre est exploitée par la prédiction intra-trame proposée. Cette information est modélisée par la corrélation entre les deux bits de poids forts codant les index de quantification scalaire. Elle vient s'ajouter à l'information de *stationnarité temporelle* modélisée par la corrélation entre bits de *sous-trames* successives. Cette dernière est de même nature que celle exploitée par la prédiction inter-trame, c'est pourquoi la combinaison des deux prédictions apporte un gain très faible au niveau de la qualité perçue, comme c'était également le cas au niveau du TEB.

Même dans la meilleure configuration (combinaison inter-trame et intra-trame), les algorithmes proposés ici ne sont pas destinés à être employés *seuls* car ils n'égalent pas le niveau de qualité perçue offert par la combinaison du décodeur de Viterbi classique *avec une procédure de masquage* déclenchée par *détection d'erreur* en sortie du décodeur convolutif. Les performances de cette combinaison sont représentées par la courbe (MV+Masquage) sur la Figure 7.8. Cependant les algorithmes de SCCD se placent justement dans un point de vue « correction d'erreur » qui est tout à fait complémentaire avec l'emploi, en aval, d'une procédure de masquage déclenchée par détection d'erreur. Les bits corrigés par ces algorithmes font justement partie de la Classe 1 sur laquelle est définie la détection d'erreur du GSM EFR. L'emploi des méthodes de SCCD permettrait alors simplement de diminuer le nombre de trames détectées en erreur et par là d'améliorer la qualité perçue, les trames non-corrigées étant, elles, toujours masquées au décodeur parole.

Les performances obtenues en combinant l'APRI-VA exploitant la corrélation inter- et intra-trame avec la procédure de masquage « classique » de l'EFR sont illustrées Figure 7.8 (courbe Inter + Intra + Masquage). On n'observe aucun gain significatif par rapport à la combinaison du décodeur de Viterbi classique avec la procédure de masquage. Il apparaît ainsi que les capacités de correction des méthodes proposées sont trop faibles pour pouvoir diminuer sensiblement le nombre de trames détectées comme « perdues ». Elles sont surtout trop limitées[79] à un nombre réduit de bits de la Classe 1. En effet, on a vu que seuls les bits de poids fort associés aux gains et au pitch étaient modélisés par ces techniques de prédiction bit à bit. L'hypothèse d'une correction induite des autres bits (comme ceux codant les LSF) par la prise en compte de l'effet mémoire du code convolutif (structure des chemins) ne paraît pas vérifiée ou son effet demeure très limité.

En conclusion, il apparaît nécessaire d'étendre la modélisation de l'information *a priori* à d'autres bits de la Classe 1, notamment à ceux codant les paramètres spectraux (LSF). C'est l'objectif des méthodes présentées dans les paragraphes suivants.

---

[79] Les courbes de taux d'erreur binaires (TEB) présentées plus haut ne donnaient pas d'information sur la *distribution* des erreurs corrigées au sein de la Classe 1a. L'absence d'amélioration avec procédure de masquage indique que les méthodes proposées influent très peu la détection d'erreur sur l'ensemble de la Classe 1, autrement dit, elles ne peuvent corriger qu'un nombre *limité* de bits au sein de la Classe 1.

7.3 Exploitation d'un a priori sur les index de quantification

Les développements précédents ont montré que des méthodes exploitant simplement la corrélation résiduelle au niveau des *bits individuels* permettaient une réduction du taux d'erreur binaire. Cependant, l'impact de ces méthodes très simples semble limité aux bits associés à une quantification *scalaire*. Or, dans le cas du GSM EFR, les paramètres spectraux (résidus LSF), qui sont parmi les plus importants pour la reconstruction de la parole, subissent une quantification *vectorielle*. Il apparaît donc nécessaire, pour ces paramètres, de modéliser la redondance résiduelle au niveau des *index de quantification*, ou plus précisément, d'exploiter la loi jointe du *groupe de bits* codant un même index i :

$$\begin{aligned} p(i) &= p\left(b_0,...,b_{M-1}\right) \\ &= p\left(u_{k_0},...,u_{k_{M-1}}\right) \end{aligned} \tag{6.26}$$

Notre objectif est ici d'analyser l'apport éventuel des méthodes de SCCD, par rapport à celles de décodage souple, pour une information *a priori de même nature*. On se limitera donc au modèle *a priori* AK0 (cf. chapitre 3), autrement dit à des lois invariantes au cours du temps. Cependant, la loi jointe (6.26) pourrait aussi bien être actualisée à chaque trame, selon le modèle de prédiction inter-trame AK1.

Les méthodes de SCCD exploitant un *a priori* au niveau paramètre (index de quantification) ont été présentées au paragraphe 6.4. Parmi les approches applicables au GSM EFR, nous ne développerons pas ici celles basées sur un décodage canal en deux étapes [Fingscheidt et al., 2000]. En effet, on a vu que ces méthodes sont d'une complexité élevée et ne permettent pas d'exploiter les contraintes du treillis (mémoire induite par le code) *conjointement* à la redondance résiduelle entre bits (corrélation intra-trame). Nous pensons que la spécificité et l'intérêt du SCCD est justement de s'appuyer sur la structure du treillis pour l'exploitation de la redondance intra-trame, ceci afin de diminuer la complexité (contrainte sur les combinaisons de bits envisagées) et d'améliorer la réduction d'erreur. Aussi, on s'intéressera dans ce qui suit à l'approche proposée par [Heinen et al., 1997].

7.3.1 Métrique conditionnée aux états précédents

On rappelle ici succinctement la métrique proposée par Heinen. Considérons à nouveau l'extension du chemin candidat ℓ, à l'étape k. A partir de la loi *a priori* (6.26) apprise sur la base de donnée de parole, on déduit la probabilité du bit u_k que l'on s'apprête à décoder sachant les bits déjà décodés le

long du chemin ℓ. Cette probabilité conditionnelle est utilisée pour pondérer la métrique de branche. On a pour l'étape $k = k_m$:

$$\begin{aligned} M_k^{(\ell)} &= M_{k-1}^{(\ell)} + \sum_{r=1}^{N} \log p\left(y_{k,r} \left| x_{k,r}^{(\ell)}\right.\right) + \log p\left(u_k = u_k^{(\ell)}\right) \\ \text{avec } p\left(u_k = u_k^{(\ell)}\right) &= p\left(u_{k_m} = u_k^{(\ell)} \left| \left\{ u_{k_j}^{(\ell)}; k_j < k_m \right\}\right.\right) \end{aligned} \tag{6.32}$$

On a vu que le chemin trouvé en appliquant l'algorithme de Viterbi pour maximiser cette métrique n'est le chemin optimal que lorsque les bits considérés sont étalés sur une distance inférieure à la longueur de contrainte du code $(\nu + 1)$. Ceci nous amènera à proposer un algorithme de décodage alternatif à l'algorithme de Viterbi. D'autre part, la contrainte d'ordre $k_j < k_m$ limite la quantité d'information exploitable. Nous proposerons au paragraphe 7.3.2 une extension permettant de modéliser la corrélation de l'ensemble des bits codant un même index, sans contrainte d'ordre. Une autre limitation est le risque de propagation d'erreur, puisque contrairement à l'approche intra-trame proposée au paragraphe 7.2.2.3, la *fiabilité* des décisions déjà opérées le long du chemin ℓ pour les bits $\left\{ u_{k_j}^{(\ell)}; k_j < k_m \right\}$ n'est pas prise en compte. Deux alternatives sont possibles pour limiter cette propagation d'erreur :

- Prédire une loi *a priori* sur u_k en généralisant l'équation (7.7) à l'ensemble des bits $\left\{ u_{k_j}^{(\ell)}; k_j < k_m \right\}$. Ceci revient à intégrer sur une partie du dictionnaire de quantification et la complexité croit très vite avec le nombre de bits considérés.

- Augmenter la profondeur de décodage effective de l'algorithme de Viterbi de manière à ce que les décisions sur les bits précédents ne soient pas définitives au moment on l'on s'apprête à décoder u_k. C'est-à-dire qu'il subsiste plusieurs chemins à l'étape k, décodant des *valeurs différentes* pour les bits $\left\{ u_{k_j}; k_j < k_m \right\}$. C'est le sens de la démarche qui sera proposée au paragraphe 7.3.3.

Les performances de la métrique de Heinen, en termes de *taux d'erreur binaire* au sein de la Classe 1a, sont illustrées Figure 7.9 (courbe « Viterbi Cond »), où elles sont comparées au décodeur de Viterbi classique (MV) ainsi qu'aux décodeurs exploitant la prédiction bit à bit inter- et intra-trame. Le gain apporté par la métrique de Heinen apparaît moindre que celui de la prédiction intra-trame entre bits individuels. Une explication pourrait être le phénomène de propagation d'erreur, qui limiterait les performances du décodeur (6.32), cependant ce phénomène devrait entraîner une dégradation *relative* d'autant plus grande que le C/I est faible. Or les courbes de TEB de la métrique (6.32) et de la prédiction intra-trame sont parallèles. Une autre explication est que la prédiction intra-trame entre bits individuels modélise également la *corrélation temporelle* (bit à bit) entre sous-trames alors que la métrique (6.32) ne prend en compte que la *non-uniformité*. Cependant, on peut s'attendre à ce que les performances de la métrique de Heinen en termes de qualité perçue soient meilleures car cette information de non-uniformité est désormais modélisée pour les LSF. On vérifiera ce point par la suite.

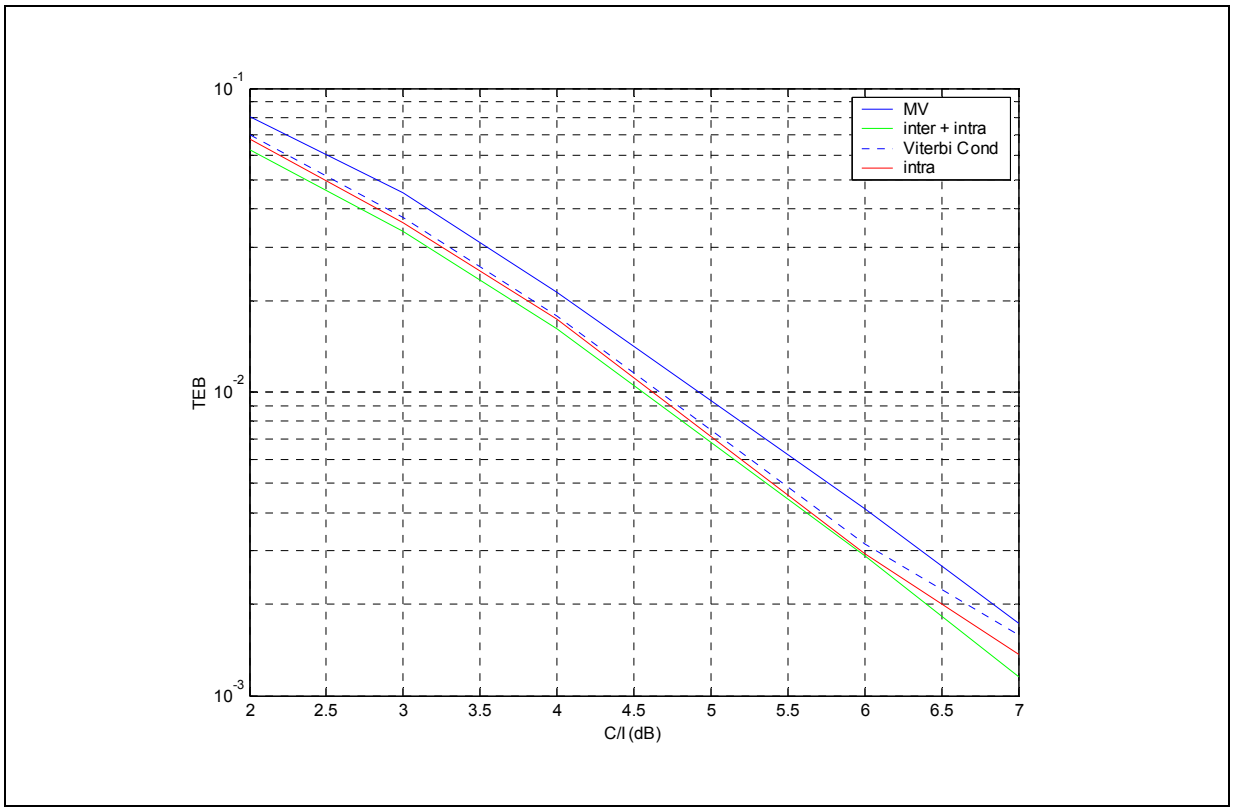

Figure 7.9 : TEB dans la Classe 1a (TU50)

7.3.2 Extension à l'algorithme du Max-Log-MAP

7.3.2.1 Principe

La contrainte d'ordre $k_j < k_m$ résulte du fait que l'algorithme de Viterbi accumule les métriques des chemins selon les k croissants. La prise en compte de l'ensemble des bits corrélés $\left\{u_{k_j}; k_j \neq k_m\right\}$ est possible avec l'algorithme de Bahl, présenté en Annexe D. Cet algorithme parcourt le treillis dans les sens « avant » (k croissants) et « arrière » (k décroissants), mais il est généralement jugé trop complexe puisque les variables calculées à chaque étape sont des sommes dans le domaine des probabilités. Cependant, comme il est remarqué en Annexe D, la maximisation des métriques dans le sens des k croissants par l'algorithme de Viterbi peut être vue comme une simplification de la variable d'induction « avant » utilisée par l'algorithme de Bahl et on déduit de cette analogie une implémentation particulière de l'algorithme de Viterbi basée sur une maximisation « avant » et « arrière ». Plus exactement, considérons les variables :

Métrique « avant » $$\alpha_k^{(\ell)} = p\left(\mathbf{y}_1,...,\mathbf{y}_k, q_1^{(\ell)},...,q_{k-1}^{(\ell)}, q_k^{(\ell)}\right) \tag{7.12}$$

Métrique « arrière » $$\beta_k^{(\ell)} = p\left(\mathbf{y}_L,...,\mathbf{y}_{k+1}, q_L^{(\ell)},...,q_{k+1}^{(\ell)} \middle| q_k^{(\ell)}\right) \tag{7.13}$$

La métrique totale du chemin ℓ peut alors se décomposer à chaque étape k, selon les métriques avant et arrière :

$$\begin{aligned} M_k^{(\ell)} &= \log p\big(\mathbf{y}_1,...,\mathbf{y}_L,q_1^{(\ell)},...,q_L^{(\ell)}\big) \\ &= \log \alpha_k^{(\ell)} + \log \beta_k^{(\ell)} \end{aligned} \tag{7.14}$$

Comme, les variables $\alpha_k^{(\ell)}$ et $\beta_k^{(\ell)}$ sont indépendantes, pour un état $q_k^{(\ell)}$ donné, il en résulte que la maximisation de la métrique $M_k^{(\ell)}$ peut se scinder en deux maximisations séparées selon les variables $\alpha_k^{(\ell)}$ et $\beta_k^{(\ell)}$. Ce principe est à la base de l'algorithme du Max-Log-MAP présenté plus en détail en Annexe D :

- pour chaque état q_k, on calcule la métrique du meilleur chemin ℓ passant par cet état
- La décision ferme $\hat{u}_k$ relâchée par l'algorithme correspond au bit d'information associé au « meilleur état » q_k, c'est-à-dire celui par lequel passe le chemin de métrique maximum.
- La valeur souple $L\big(u_k\big)$ est définie par la différence de métrique entre les meilleurs chemins passant respectivement par un état q_k pair (i.e. $u_k = +1$) et impair (i.e. $u_k = -1$).

L'algorithme du Max-Log-MAP revient donc à calculer deux algorithmes de Viterbi, l'un dans le sens des k croissants, l'autre dans le sens des k décroissants. En contrepartie, l'opération de « traceback » n'est plus nécessaire pour retrouver les états du chemin optimal.

On montre ici que l'introduction de la variable $\beta_k^{(\ell)}$ permet de prendre en compte les corrélations avec les bits $\big\{u_{k'};k'>k\big\}$. Pour cela, on introduit la variable arrière légèrement modifiée suivante :

$$\tilde{\beta}_k^{(\ell)} = p\big(\mathbf{y}_L,...,\mathbf{y}_{k+1},q_L^{(\ell)},...,q_{k+1}^{(\ell)},q_k^{(\ell)}\big) \tag{7.15}$$

La variable $\tilde{\beta}_k^{(\ell)}$ peut alors se calculer par une récurrence selon les k *décroissants* :

$$\tilde{\beta}_k^{(\ell)} = \tilde{\beta}_{k+1}^{(\ell)} p\big(\mathbf{y}_{k+1}\big|q_k^{(\ell)},q_{k+1}^{(\ell)}\big)\, p\big(q_k = q_k^{(\ell)}\big|q_{k+1}^{(\ell)},...,q_L^{(\ell)}\big) \tag{7.16}$$

ce qui correspond, dans le domaine logarithmique, à une accumulation de métrique le long du chemin ℓ parcouru dans le sens rétrograde :

$$\begin{aligned} \underset{\leftarrow}{M}_k^{(\ell)} &\triangleq \log \tilde{\beta}_k^{(\ell)} \\ &= \underset{\leftarrow}{M}_{k+1}^{(\ell)} + \Delta\underset{\leftarrow}{M}\big(q_k^{(\ell)},q_{k+1}^{(\ell)}\big) \end{aligned} \tag{7.17}$$

où $\Delta\underset{\leftarrow}{M}\big(q_k^{(\ell)},q_{k+1}^{(\ell)}\big)$ désigne *l'incrément de métrique* associé à la branche $\big(q_k^{(\ell)},q_{k+1}^{(\ell)}\big)$ du treillis :

$$\begin{aligned} \Delta\underset{\leftarrow}{M}\big(q_k^{(\ell)},q_{k+1}^{(\ell)}\big) &= \log p\big(\mathbf{y}_{k+1}\big|q_k^{(\ell)},q_{k+1}^{(\ell)}\big) + \log p\big(q_k = q_k^{(\ell)}\big|q_{k+1}^{(\ell)},...,q_L^{(\ell)}\big) \\ &= \log p\big(\mathbf{y}_{k+1}\big|\mathbf{x}_{k+1}^{(\ell)}\big) + \log p\big(u_{k-\nu+1} = u_{k-\nu+1}^{(\ell)}\big|u_{k-\nu+2}^{(\ell)},...,u_k^{(\ell)},...,u_L^{(\ell)}\big) \end{aligned} \tag{7.18}$$

On utilise alors l'algorithme de Viterbi pour maximiser récursivement la métrique $\underset{\leftarrow}{M}_k^{(\ell)}$ selon les k *décroissants*. Comme pour le calcul de la métrique « avant », l'emploi de l'algorithme de Viterbi conduit à une approximation puisque les métriques de branches à des étapes successives $(k+1)$ et k ne sont pas indépendantes entre elles. La nature de cette approximation a été étudiée au §6.4.2 pour la métrique « avant ». Elle correspond à une exploitation sous-optimale de la loi jointe (6.26) des bits corrélés et on a vu les risques de propagation d'erreur qu'elle pouvait induire.

On remarquera que l'information *a priori* utilisée dans la métrique (7.18) de la branche $\left(q_k, q_{k+1}\right)$ du treillis est la probabilité conditionnelle du bit $u_{k-\nu+1}$ et non celle du bit d'information u_k associé à cette branche par le codeur convolutif. En effet, du point de vue de la récursion « arrière » (7.16), c'est le bit $u_{k-\nu+1}$ qui est en « entrée » de registre d'état lorsqu'on étend le chemin ℓ de l'état q_{k+1} à l'état q_k. Cependant, la probabilité conditionnelle $p\left(u_k = u_k^{(\ell)} \middle| u_{k+1}^{(\ell)},...,u_L^{(\ell)}\right)$ est déjà prise en compte à une étape antérieure[80] du calcul de la métrique « arrière ». La dépendance entre le bit u_k et les bits $\left\{u_{k'}; k' > k\right\}$ est donc effectivement exploitée lorsqu'on maximise la métrique $\underset{\leftarrow}{M}_k^{(\ell)}$ relativement à l'état q_k.

Finalement, de manière similaire à (7.14), la *métrique totale* d'un chemin l du treillis se décompose à chaque étape k selon les métriques *« avant »* et *« arrière »*. On a en effet d'après (7.12) et (7.15) :

$$\begin{aligned} M_k^{(\ell)} &\triangleq \log p\left(\mathbf{y}_1,...,\mathbf{y}_L, q_1^{(\ell)},...,q_L^{(\ell)}\right) \\ &= \log \alpha_k^{(\ell)} + \log \beta_k^{(\ell)} - \log p\left(q_k^{(\ell)}\right) \\ &= \underset{\rightarrow}{M}_k^{(\ell)} + \underset{\leftarrow}{M}_k^{(\ell)} - \log p\left(u_{k-\nu+1}^{(\ell)},...,u_k^{(\ell)}\right) \end{aligned} \tag{7.19}$$

Le dernier terme de l'expression (7.19) s'explique par le fait que la probabilité de l'état $q_k^{(\ell)}$ est prise en compte deux fois, par la métrique « avant » $\underset{\rightarrow}{M}_k^{(\ell)}$ et par la métrique « arrière » $\underset{\leftarrow}{M}_k^{(\ell)}$.

On considère alors que la métrique totale du *meilleur chemin* passant par *l'état* $q_k^{(\ell)}$ s'obtient comme la somme des métriques $\underset{\rightarrow}{M}_k^{(\ell)}$ et $\underset{\leftarrow}{M}_k^{(\ell)}$ *maximisées* par les mises en œuvre « avant » et « arrière » de l'algorithme de Viterbi. Cette démarche s'inspire de celle de l'algorithme Max-Log-Map en y introduisant un terme *a priori*. Cependant on insistera à nouveau sur le fait qu'on pose ici une hypothèse simplificatrice puisque l'algorithme de Viterbi ne tient pas compte de la dépendance entre les termes *a priori* introduits dans les métriques de branche.

Ainsi, l'algorithme présenté ici constitue une amélioration de l'approche proposée par Heinen puisque la *décision* sur le bit d'information u_k à l'étape k, qui s'obtient par *maximisation de la métrique* (7.19) *relativement aux états* $q_k^{(\ell)}$, prend désormais en compte la probabilité *a priori* $p\left(u_k = u_k^{(\ell)} \middle| \left\{u_{k'}^{(\ell)}; k' \neq k\right\}\right)$. Cependant, cet algorithme demeure sous-optimal puisqu'il ne permet pas un décodage *conjoint* de l'ensemble des bits corrélés entre eux. Cette limitation provient comme on l'a déjà mentionné des conditions de mise en oeuvre de l'algorithme de Viterbi. Elle était déjà présente dans l'approche de Heinen. Nous reviendrons sur ce point par la suite.

---

[80] Cette étape « antérieure » est l'étape $(k+\nu-1)$ puisqu'on parcourt le treillis dans le sens des k décroissants.

En termes de complexité, on notera que la démarche proposée nécessite de stocker les décisions prises le long de chaque chemin, pour les sens « avant » et « arrière » alors que l'algorithme du Max-Log-Map permettait de s'affranchir de toute mémorisation des décisions. La complexité de la méthode présentée se rapproche donc de celle d'un décodage en *deux étapes*, cependant il n'est ici nécessaire de stocker que les seuls bits pour lesquels on exploite une information *a priori*. D'autre part, on évite ici le parcours du dictionnaire de quantification selon **(6.38)** en considérant uniquement les combinaisons de bits associées aux chemins survivants à l'étape k dans les sens « avant » et « arrière ».

7.3.2.2 Mise en œuvre

L'algorithme mettant en œuvre la métrique (7.19) sera désigné « Max-Log-Map Conditionné » tout au long de ce qui suit. Nous comparons ici ses performances à celles des algorithmes de décodage canal précédemment étudiés. L'amélioration apportée par cette extension de la métrique de Heinen apparaît assez peu significative lorsqu'on considère le taux d'erreur binaire dans la Classe 1a, illustré Figure 7.9. Elle permet simplement de retrouver un taux d'erreur similaire à celui de la prédiction intra-trame. Cependant, on rappellera ici encore que les techniques de prédiction inter-trame et intra-trame exploitent une information supplémentaire à la seule non-uniformité, qui est la *stationnarité* temporelle. L'intérêt des méthodes exploitant un *a priori* au niveau des *index de quantification* apparaît de manière bien plus évidente lorsqu'on s'intéresse à la qualité perçue du signal de parole décodé. Les notes de qualité *estimées* par l'algorithme PESQ sont reportées Figure 7.11, elles correspondent à une mise en oeuvre des algorithmes *sans procédure de masquage en aval*. La prise en compte de la non-uniformité des paramètres spectraux (LSF) par la métrique de Heinen (« algorithme de « Viterbi Conditionné ») ainsi que par son extension à l'algorithme du Max-Log-Map, apportent cette fois-ci un gain par rapport à la technique de prédiction combinée inter- et intra-trame. La métrique (7.19) représente elle-même une amélioration par rapport à la métrique de Heinen pour les bas niveaux de C/I.

Employés seuls, c'est-à-dire sans procédure de masquage en aval, les algorithmes de « Viterbi conditionné » et du « Max-Log-Map conditionné » permettent d'atteindre un niveau de qualité perçue équivalent à celui du décodeur GSM EFR avec procédure de masquage (courbe « Masquage ») pour les niveaux de C/I intermédiaires (entre 4dB et 7dB). Leur performance devient inférieure en deçà de 4dB. On n'a cependant pas constaté d'amélioration significative lorsque on les combine avec la procédure de masquage déclenchée par indicateur de trame perdue BFI. Ici aussi, l'explication est que la proportion de bits corrigés par ces méthodes au sein de la classe 1 est trop restreinte pour avoir un impact important sur le nombre de trames détectées en erreur et substituées. L'intérêt éventuel de ces méthodes de SCCD réside donc dans leur emploi combiné avec un décodeur souple de parole. Ceci sera étudié dans la suite de ce chapitre.

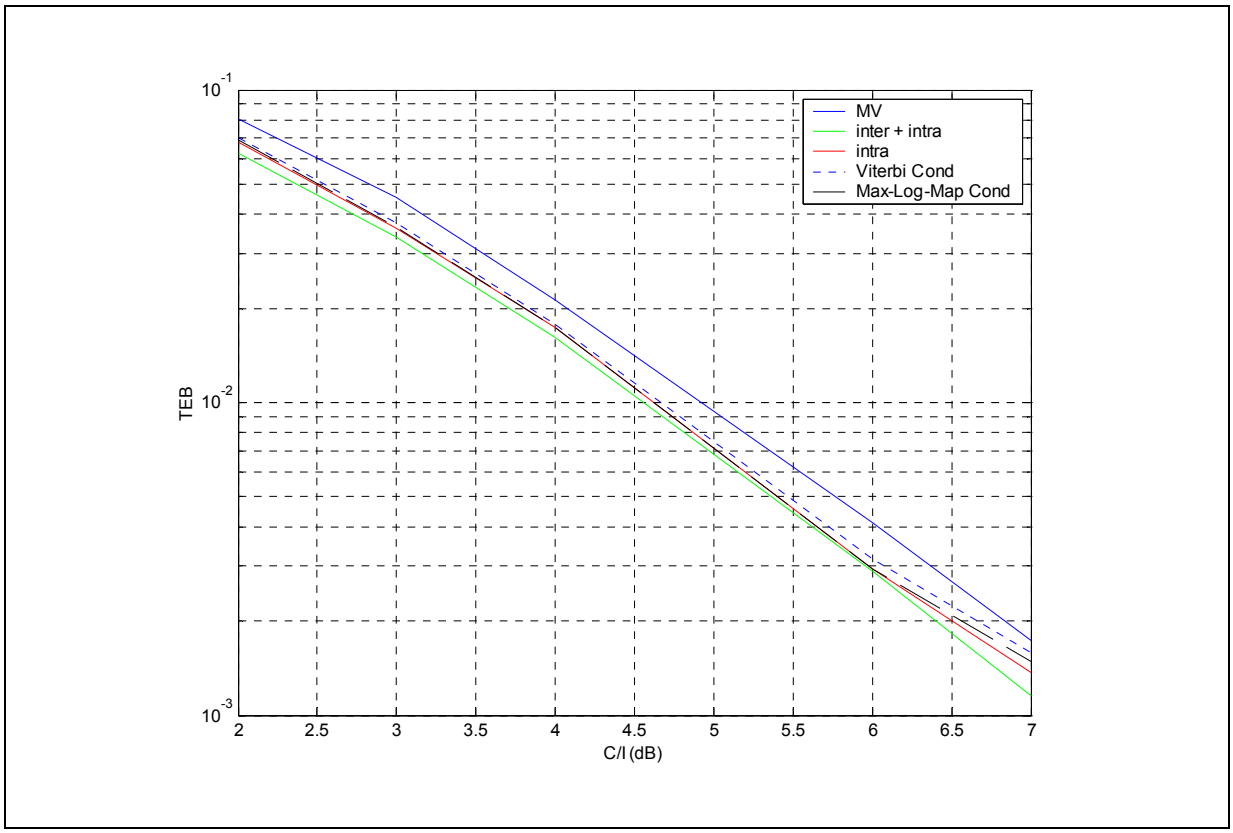

Figure 7.10 : Taux d'erreur binaire dans la Classe 1a (TU50)

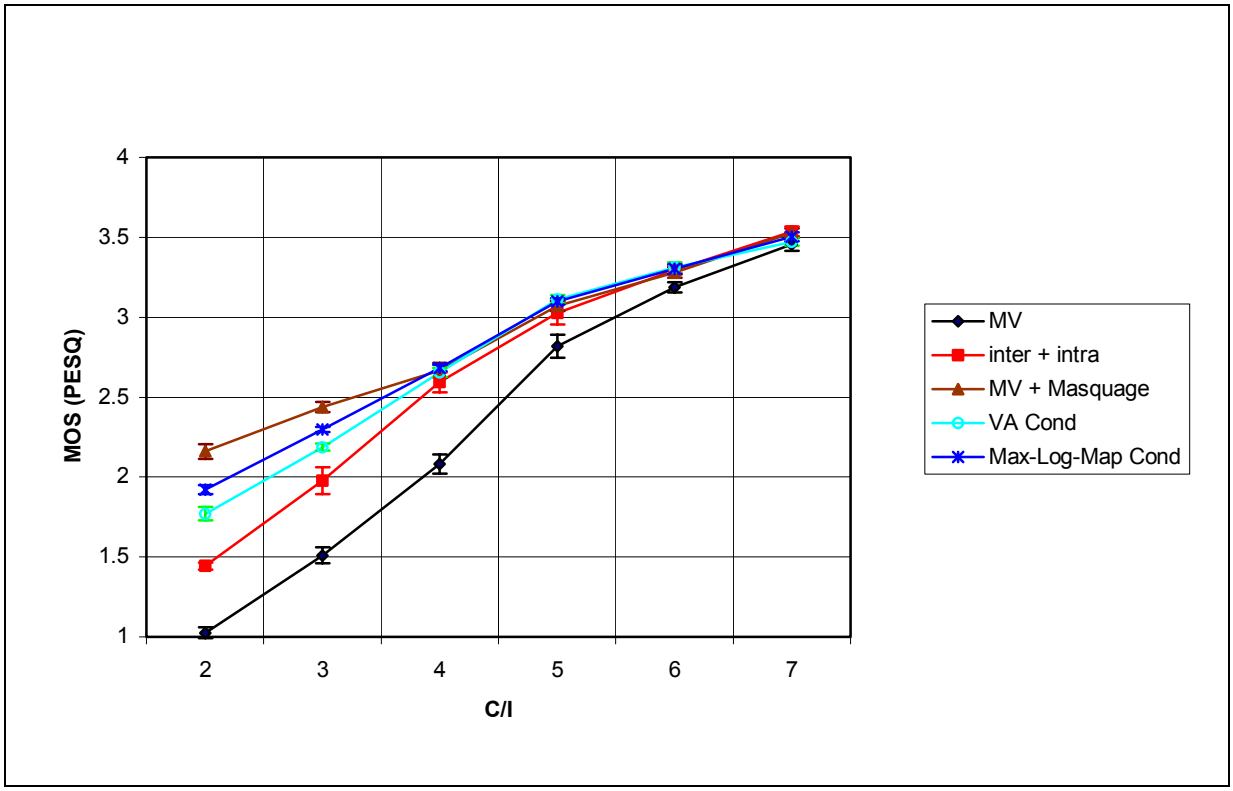

Figure 7.11 : Notes MOS *estimées* pour les méthodes de SCCD étudiées (TU50)

7.3.3 Augmentation de la profondeur de décodage à l'aide d'un GVA

L'approche présentée au paragraphe précédent ainsi que celle de Heinen sont *sous-optimales* parce que l'algorithme de Viterbi ne permet pas un décodage *conjoint* des bits étalés sur une longueur supérieure à la *longueur de contrainte* $D = (\nu + 1)$ où ν désigne la mémoire d'état du treillis. L'étalement des bits corrélés, qui résulte entre autres, du ré-ordonnancement des bits avant codage canal, est illustré Figure 7.12. L'exploitation *optimale* de la redondance résiduelle par l'algorithme de Viterbi suppose alors, comme on l'a vu au §6.4.1, de reformuler un treillis dont les états ou les branches *regroupent* l'ensemble des bits dont on cherche à exploiter la corrélation. On notera que la structure d'un tel treillis « étendu » coïncide alors avec celle du treillis associé à un code de longueur de contrainte égale à l'étalement D_M des bits corrélés. Ceci signifie simplement qu'au niveau du décodeur canal, la redondance résiduelle est exploitée de la même façon qu'une redondance induite par un code convolutif. Cependant la taille de ce treillis croit exponentiellement avec l'étalement D_M ce qui rend irréalisable la mise en oeuvre de l'algorithme de Viterbi. Une possibilité cependant serait d'appliquer à un tel treillis « étendu », un algorithme de décodage de complexité moindre que l'algorithme de Viterbi. C'est l'idée à la base des développements présentés dans ce paragraphe.

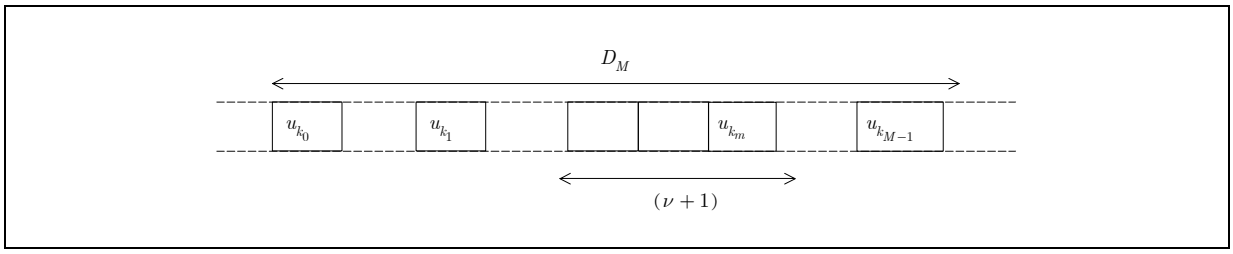

Figure 7.12 : Etalement des bits codant un même index, comparé à la longueur de contrainte

7.3.3.1 Principe

On se place ici du point de vue du décodeur canal où l'on traite la *redondance résiduelle* comme une redondance qui serait induite par un code de *longueur de contrainte* D_M. Le treillis dont il est question dans ce qui suit est donc le treillis associé à un tel code.

La complexité de mise en oeuvre de l'algorithme de Viterbi dans le cas où la longueur de contrainte devient élevée a conduit au développement d'algorithmes de décodage sous-optimaux, ayant une relation plus « lâche » avec la structure du treillis. En effet, l'algorithme de Viterbi explore tous les chemins du treillis en conservant, à chaque étape k, *un chemin survivant par état*. Cependant, il existe des conditions pour lesquelles il est raisonnable de considérer que seuls quelques uns de ces chemins ont une probabilité élevée. Dans ce cas, un parcours exhaustif du treillis n'est pas nécessaire et l'on peut conserver uniquement un nombre plus réduit de *chemins dominants*. Les algorithmes procédant

de cette démarche peuvent être regroupés sous l'approche générique de l'algorithme de « Viterbi généralisé » (GVA) proposée par [Hashimoto, 1987]. Le principe de cet algorithme pour un code de longueur de contrainte D est le suivant :

- **Réduction de la dimension du « treillis » considéré :**

  Au lieu de considérer individuellement les 2^{D-1} états du treillis, on considère des états « dégénérés » ou *labels* définis en tronquant la mémoire d'état à une taille $(L-1)$. Un label g_k à l'étape k, regroupe donc les 2^{D-L} états qui partagent les mêmes $(L-1)$ derniers bits $\left(u_{k-L+2},...,u_k\right)$ entrés dans leur registre.

- **Relâchement de contrainte sur les chemins survivants :**

  La sélection de chemins survivants ne se fait plus séparément pour chaque état du treillis, mais par *labels*. En contre-partie, plusieurs chemins survivants sont retenus par labels. Les chemins survivants au label g_k sont sélectionnés par tri de la *liste* de chemins candidats aboutissant à g_k, c'est-à-dire des chemins se terminant par $\left(u_{k-L+2},...,u_k\right)$.

Le nombre S de chemins survivants par label est choisi *inférieur ou égal* au nombre d'états regroupés. Les performances de l'algorithme GVA sont d'autant meilleures que le nombre d'états regroupés 2^{D-L} est faible et que le ratio entre le nombre S d'états survivants et le nombre d'états regroupés est proche de l'unité.

On justifie maintenant l'intérêt de l'algorithme GVA pour l'exploitation de la redondance résiduelle. Considérons à nouveau la métrique (6.32). Cette métrique exploite la redondance des bits d'information sur une longueur D_M et le décodeur optimal associé à cette métrique serait un algorithme de Viterbi défini sur un treillis de dimension 2^{D_M-1}. Cependant, le décodeur met en oeuvre en pratique l'algorithme de Viterbi sur le treillis associé au code convolutif du GSM EFR, c'est-à-dire un treillis de dimension 2^{ν} avec $\nu = 4$. En reprenant les notations de l'algorithme GVA, ceci peut être vu comme une approximation par rapport au décodeur *optimal* où l'on poserait $L = \nu + 1$ et $S = 1$. Une manière d'améliorer les performances de l'algorithme, c'est-à-dire la prise en compte de la redondance résiduelle[81], est donc d'augmenter le nombre le chemins survivants S par état du treillis utilisé (treillis du code convolutif).

Le fait de choisir plusieurs survivants par états du treillis peut sembler paradoxal, mais il faut bien considérer ces états comme les *états « dégénérés »* ou *labels* d'un treillis modélisant l'effet mémoire induit par la *redondance résiduelle* entre bits d'information. L'augmentation du nombre de survivants S permet ici d'augmenter *la profondeur de décodage* pour tendre vers un décodage joint des bits corrélés. Cette démarche n'a évidemment de sens que si la métrique exploitée prend en compte la

[81] La *redondance systématique* introduite par le code convolutif est bien sur parfaitement prise en compte par l'algorithme de Viterbi tel qu'utilisé par le décodeur puisqu'il exploite la structure du treillis associée au code. L'objectif d'une mise en oeuvre du GVA n'est pas ici de réduire le nombre d'états de ce treillis, ce qui conduirait à une diminution des performances nominales du décodeur canal EFR.

redondance résiduelle entre bits d'une même trame d'information (redondance *intra-trame*), le cas contraire aboutirait à considérer plusieurs chemins strictement parallèles. On se propose ici de mettre en œuvre un algorithme GVA avec la métrique (6.32) exploitant la corrélation entre les bits codant un même index de quantification.

7.3.3.2 Mise en oeuvre

Comme on l'a vu, l'algorithme GVA est défini par deux paramètres :

- le nombre S de survivants par labels ;
- la longueur L définie, par analogie avec la longueur de contrainte, telle que $(L-1)$ est la taille de la mémoire associée aux labels.

Dans l'approche proposée ici, la longueur L est nécessairement prise égale à la longueur de contrainte $\nu+1$ du code convolutif, afin de ne pas dégrader les performances spécifiées pour le GSM EFR. Le choix du nombre S de survivants par labels (ici, états du treillis) est alors limité par des considérations de complexité. Les deux facteurs limitant sont le *nombre total Q de chemins* candidats à gérer et la *complexité algorithmique du tri* effectué pour chaque liste de candidats par labels. On a :

$$Q = 2^L S \tag{7.20}$$

et l'algorithme de tri optimal requière de l'ordre de $O(n \log n)$ opérations pour chacun des 2^{L-1} labels, où n est ici égal à $2S$. Par comparaison l'algorithme de Viterbi gère un total de 2^L chemins candidats et la complexité de la sélection des survivants est de l'ordre $O\left(2^L\right)$ pour l'ensemble des états. Dans ces conditions, la complexité de l'algorithme GVA croit très vite avec le nombre S de survivants et devient rédhibitoire à partir de $S = 8$. On a choisit ici $S = 4$, la complexité de l'algorithme résultant est donc nettement plus élevée que l'algorithme de Viterbi (d'un facteur supérieur à 10 sur la sélection de survivant), mais le but de l'étude menée ici est simplement d'évaluer si un gain est possible en augmentant la profondeur de décodage de façon à permettre (au moins partiellement) un décodage conjoint des bits corrélés.

La Figure 7.13 illustre les taux d'erreur binaires obtenus pour les bits de la Classe 1a et compare les performances de l'algorithme GVA dans la configuration retenue $(S = 4, L = 5)$ à celles de la métrique (7.19) (« Max-Log-Map conditionné ») et des prédictions inter- et intra-trame. Comme c'était le cas pour les algorithmes de « Viterbi conditionné » (métrique (6.32)) et « Max-Log-Map conditionné », les taux d'erreur binaire obtenus avec l'algorithme GVA, au sein de la Classe 1a, sont plus élevés que ceux obtenus avec la prédiction inter- et intra-trame. La raison est que les algorithmes de prédiction inter- et intra-trame intègrent une information sur la *corrélation temporelle* (entre trames ou sous-trames), alors que nous nous somme limités à un modèle (6.26) invariant au cours du temps (AK0) pour les algorithmes exploitant la distribution des index de quantification. Comme on l'a déjà mentionné, la loi jointe (6.26) peut être actualisée au cours du temps afin de modéliser la dépendance temporelle, néanmoins c'est au prix d'une complexité nettement accrue (cf. chapitre 3).

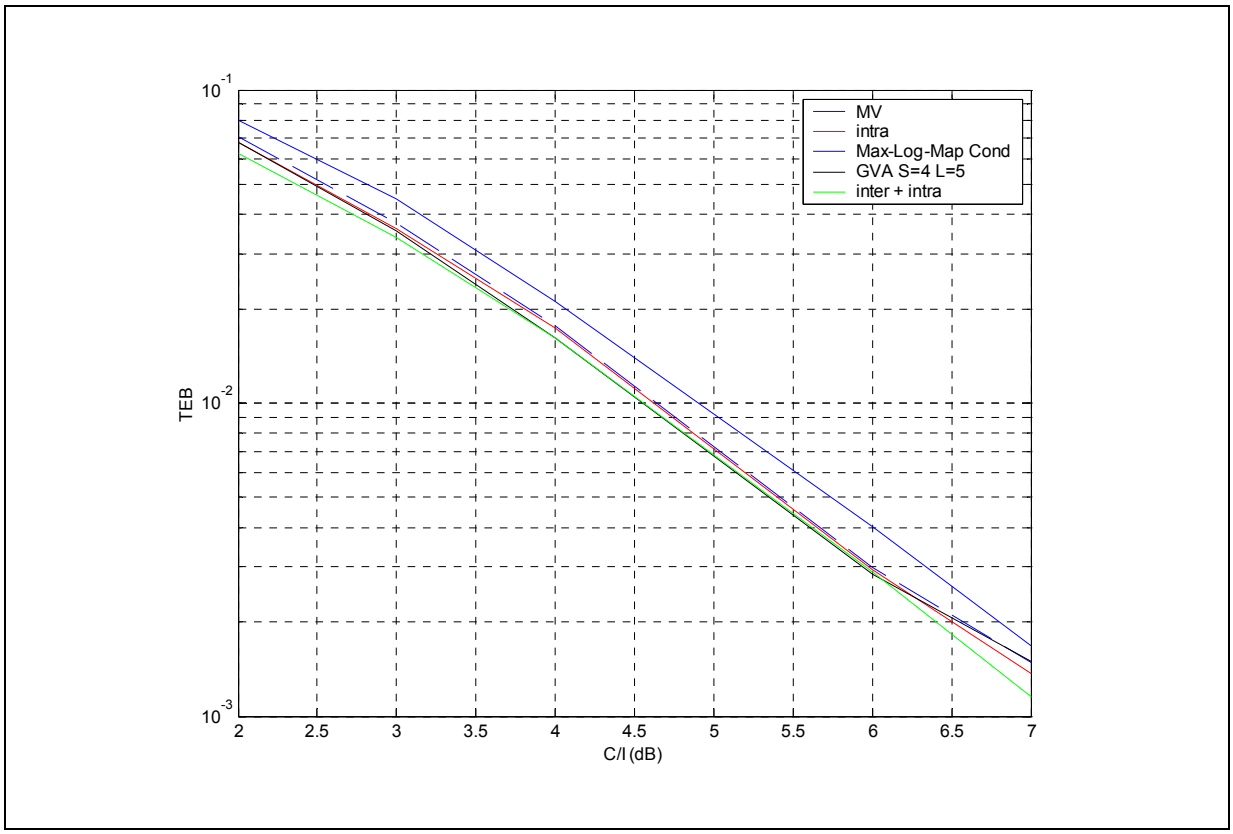

Figure 7.13 : TEB comparés pour la Classe 1a (TU50)

Sur l'ensemble de la Classe 1a, la mise en œuvre de l'algorithme GVA apporte un faible gain relativement à l'algorithme « Max-Log-Map conditionné », ce gain semble surtout effectif pour les niveaux C/I compris entre 3dB et 6dB.

On n'observe pas non plus de gain réellement significatif lorsqu'on considère, Figure 7.14, les notes de qualité *estimées* sur le signal de parole décodé. Les performances du GVA sont même en retrait par rapport à celles du « Max-Log-Map conditionné » pour les très bas niveaux de C/I. En revanche, la mise en œuvre du GVA avec la procédure de masquage de l'EFR, déclenchée par détection d'erreur dans la Classe 1, se traduit par une légère amélioration des notes MOS *estimées* pour ces très bas niveaux de C/I. Une explication pourrait être que la *proportion* des bits corrigés par le GVA au sein de la Classe 1 est plus élevée que pour les autres algorithmes, ce qui tend à réduire le nombre de trames déclarées comme « perdues ». Ceci signifierait que l'algorithme GVA permet d'améliorer le phénomène de correction induite, c'est-à-dire la possibilité de corriger d'autre bits que ceux pour lesquels on exploite la redondance résiduelle, en profitant de l'augmentation de la profondeur de décodage. L'amélioration observée est cependant très faible au regard de la complexité requise par la mise en œuvre de l'algorithme GVA.

En conclusion, l'emploi d'un algorithme GVA pour améliorer la prise en compte de la redondance résiduelle suppose certainement de choisir un nombre de survivants S plus élevé pour obtenir des résultats convaincants. Ceci conduit à un algorithme trop complexe pour être intéressant dans la pratique.

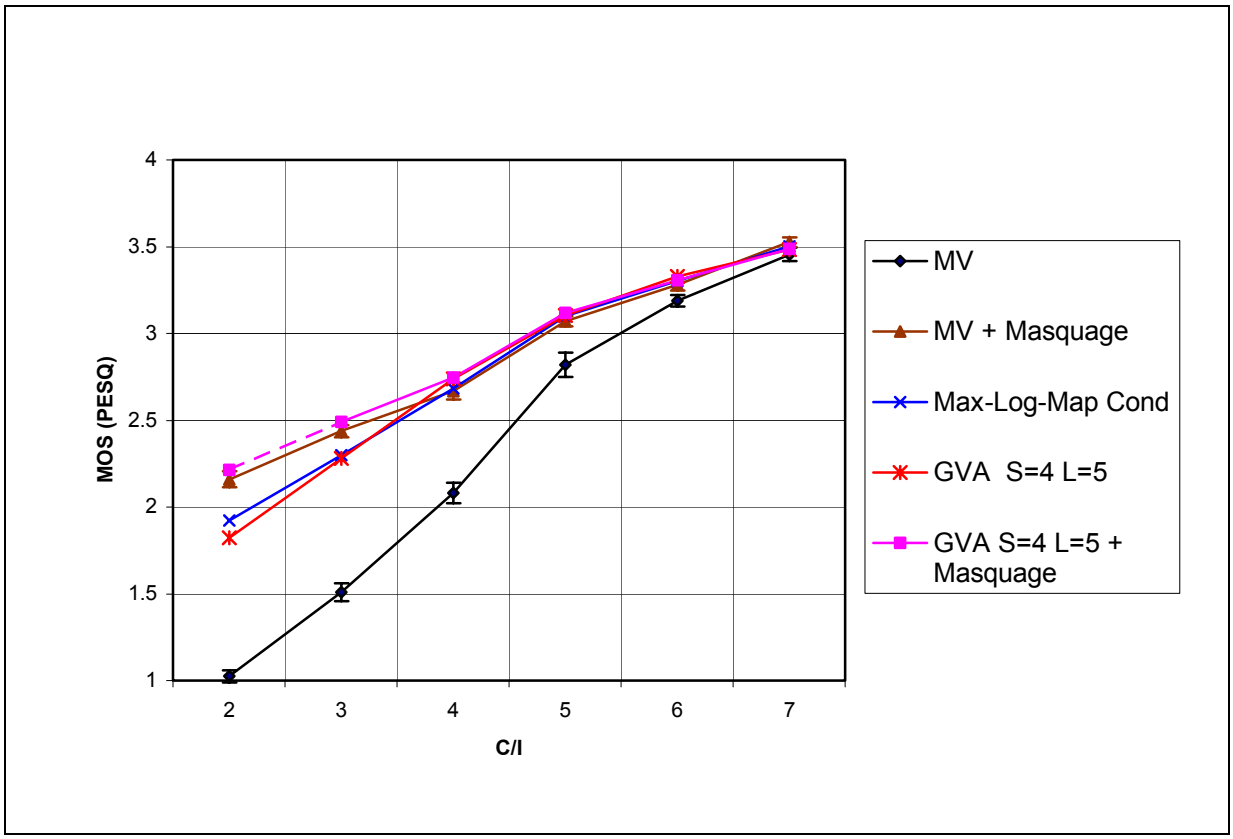

Figure 7.14 : Notes MOS estimées (TU50)

7.4 Combinaison du SCCD et du décodage de parole souple

On s'intéresse maintenant à la combinaison du SCCD et du décodage souple. La redondance résiduelle en sortie du codeur parole est alors exploitée à deux niveaux successifs de la chaîne de réception. Deux configurations sont envisageables :

- le décodeur canal et le décodeur parole exploitent une information *a priori* de *même nature*. Par exemple, la non-uniformité d'un paramètre. Cette information *a priori* étant modélisée de manière optimale au décodeur parole (probabilité du paramètre) et sous-optimale au niveau du décodeur canal (loi conditionnelles entre les bits codant le paramètre).

- le décodeur canal et le décodeur de parole exploitent des informations *a priori* de nature *complémentaires*. Par exemple, la non-uniformité pour le décodeur parole et la corrélation temporelle pour le décodeur canal.

Dans le premier cas, le problème posé est de savoir si une même information utilisée deux fois mais à des niveaux distincts peut apporter un gain de performance. L'intérêt du deuxième cas est qu'il permet le choix du niveau le plus adapté pour prendre en compte une information *a priori* donnée. Ainsi, on peut utiliser la méthode de SCCD avec prédiction inter-trame pour prendre en compte la *stationnarité temporelle* d'un paramètre (quantifié scalairement ou avec « Index Assignement ») au bénéfice d'une complexité réduite par rapport au modèle AK1 présenté pour le décodeur souple.

On se limitera ici à l'étude de la combinaison des algorithmes de SCCD avec le décodeur AK0[82]. D'une part, cette configuration est suffisante pour nous permettre de répondre à la première question posée, et d'autre part, elle constitue la combinaison la plus intéressante en pratique dans le cas où l'on mise sur la complémentarité des informations exploitées au décodeur parole et au décodeur canal.

La Figure 7.15 compare pour les principaux algorithmes de SCCD étudiés dans ce chapitre, les performances de leur combinaison avec le décodeur souple AK0. Les résultats obtenus avec le décodeur de Viterbi classique (MV) suivi du mécanisme de substitution de trame (MV+Masquage) ou du décodeur AK0 (MV+AK0) y sont également représentées.

On constate que les différents algorithmes de SCCD combinés avec le décodeur souple AK0 aboutissent à des performances similaires, légèrement supérieures à celles du décodeur souple AK0 utilisé seul. Cependant le gain maximal obtenu est de l'ordre de 0,1 MOS et n'est donc pas significatif.

De plus, il est surprenant de ne pas retrouver entre les différents algorithmes de SCCD, une hiérarchie conforme à celle obtenue Figure 7.11 lorsqu'ils étaient utilisés seuls. Il nous semble ainsi que les performances observées Figure 7.15 sont essentiellement dictées par le décodeur AK0, ce qui explique que les courbes correspondantes soient parfaitement parallèles à celle du décodeur AK0. Les méthodes de SCCD étudiées ici n'apportent qu'une amélioration marginale. Autrement dit, la réduction du taux d'erreur résiduelles parmi les bits de la Classe 1a qu'elles permettent se traduit en une réduction de la distorsion dans le domaine des paramètres qui est négligeable face à celle obtenue par AK0.

Il apparaît ainsi que dans le contexte étudié, la prise en compte d'une information de redondance résiduelle doit être effectuée au niveau du codeur parole et non à celui du décodeur canal. Ceci s'explique par les contraintes imposées par le schéma de codage canal du GSM. Il en est ainsi, par exemple, du ré-ordonnancement des bits qui limite la possibilité de prendre en compte la redondance au niveau des index de quantification. Or, dans le codeur de parole EFR, les paramètres spectraux (LSF), qui sont les plus importants pour la qualité de la parole restituée, subissent une quantification vectorielle et leur redondance ne peut pas être modélisée au simple niveau des bits individuels.

[82] L'algorithme de décodage souple utilisé ici est celui basé sur la modélisation par GMM. On le désigne simplement par AK0.

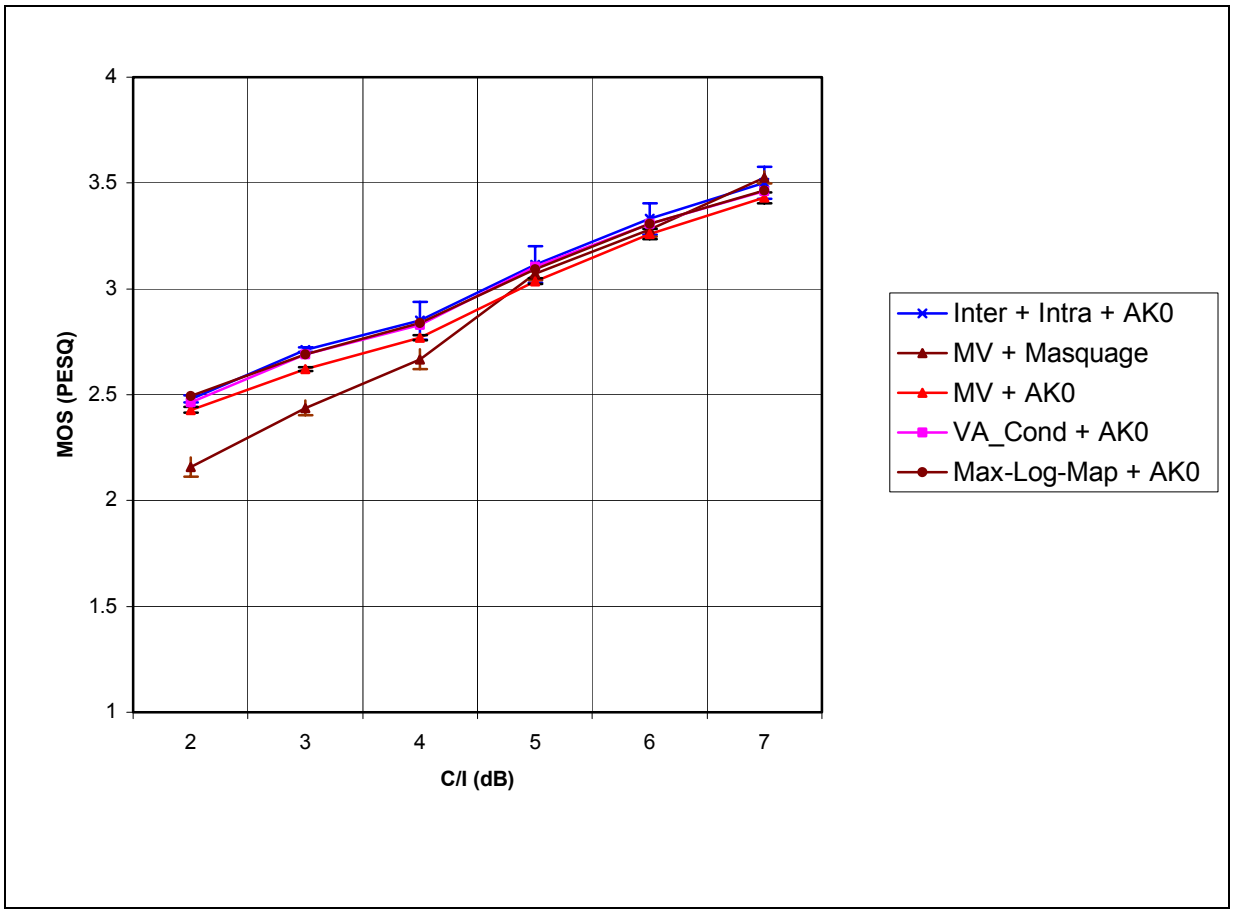

Figure 7.15 : Notes MOS estimées (TU50)

7.5 Conclusion

Les résultats obtenus dans ce chapitre montrent que l'exploitation de la redondance résiduelle au niveau du décodeur canal est moins performante qu'au niveau du décodeur de parole. L'utilisation des méthodes de SCCD *en complément* du décodeur souple de parole n'apporte pas non plus de gain décisif. Parmi les facteurs limitant les performances, on peut dissocier ceux qui résultent des contraintes spécifiquement imposées par le GSM EFR, de ceux qui sont intrinsèques au principe même des algorithmes utilisés. Ainsi, le ré-ordonnancement et le multiplexage des bits avant codage canal dans le système GSM ne permet pas d'exploiter pleinement la redondance au niveau *paramètre*, c'est-à-dire de décoder de manière *conjointe* le groupe de bits codant un même paramètre. En ce qui concerne les méthodes de prédiction entre bits individuels, leur efficacité est restreinte aux seuls paramètres de gains et de délai de pitch puisque les paramètres spectraux (LSF) du codeur de parole GSM EFR subissent une quantification vectorielle (QV) et qu'aucune procédure d'optimisation de l'attribution des indices de QV (« Index Assignment ») n'est utilisée.

Au niveau des limitations propres aux méthodes de SCCD mises en oeuvre, on notera d'une part que l'algorithme de Viterbi utilisé renvoie un chemin sous-optimal lorsqu'on utilise des métriques de branches conditionnées aux états précédents parcourus dans le treillis. D'autre part, l'effet de masquage intrinsèque du décodeur souple, dû au critère MMSE utilisé, ne se retrouve plus au niveau du décodeur SCCD qui utilise le critère MAP par séquence.

En conclusion, l'attrait principal des méthodes SCCD par rapport au méthodes de décodage souple est leur complexité réduite puisqu'elles remplacent le parcours exhaustif du dictionnaire de quantification par le parcours d'un treillis pour l'estimation des index de quantification transmis (i.e. des bits les codant). Néanmoins, l'exploitation au niveau du décodeur canal, d'un *a priori* sur les index de quantification suppose que les bits codant les index soient peu dispersés. Ceci s'oppose au principe du multiplexage qui cherche à étaler les bits afin d'uniformiser la distribution des erreurs. Paradoxalement, l'emploi de méthodes utilisant un *a priori* individuellement sur les bits (prédiction inter ou intra-trame) nous paraît être plus adapté dans la pratique au niveau du décodeur canal. L'efficacité de telles méthodes repose alors sur une proximité de la topologie dans le domaine des bits codant les index (distance de Hamming) avec celle dans le domaine des centroïdes (distance euclidienne). C'est justement l'objectif des techniques d'attribution d'indices (« Index Assignment »), nous pensons que l'emploi de telles méthodes avec les techniques SCCD de prédiction au niveau des bits individuels pourrait être intéressant. Cependant, les performances des méthodes de SCCD en termes de qualité perçue restent limitées par le critère MAP exploité par le décodeur de Viterbi. Une utilisation du critère MMSE a été proposée dans [Heinen et al., 2000] mais elle ramène la complexité du SCCD à celui du décodeur souple.

Conclusion et perspectives

Rappel de la problématique et principaux résultats

Dans le contexte des communications radio-mobiles, la qualité de la parole restituée ne dépend pas uniquement de la distorsion introduite pour la réduction de débit au niveau du codeur parole mais est très fortement impactée par les *erreurs résiduelles* en sortie du décodeur canal. Celles-ci résultent de la très grande fluctuation de la qualité du canal radio ainsi que des contraintes de complexité limitant les performances du codage canal pour un rendement donné. Parallèlement, ces mêmes contraintes de complexité et de délai font qu'il subsiste une *redondance résiduelle* en sortie du codeur parole. Les études présentées dans ce document visent à exploiter cette redondance résiduelle afin de combattre l'impact des erreurs résiduelles. Elles s'inscrivent dans le cadre plus général du *décodage conjoint source-canal*.

Nous avons étudié successivement deux types d'approches selon que la distorsion de la parole en présence d'erreurs résiduelles est minimisée directement au niveau du *décodeur parole* (approche SBSD) ou indirectement par la réduction du taux d'erreurs résiduelles au niveau du *décodeur canal* (approche SCCD). Comme ils s'appliquent en réception, les algorithmes développés pour ces deux approches peuvent être mis en oeuvre dans un système tel que le GSM sans modification de la norme. Leurs performances ont été évaluées par un critère commun (note PESQ) calculé à l'aide d'un algorithme normalisé pour l'estimation de la qualité vocale transmise par un réseau radio-mobile.

Approche SBSD

L'estimation optimale des paramètres à partir de la sortie souple du décodeur canal et de l'information issue de la redondance résiduelle permet un masquage « intelligent » dont le comportement est très différent de celui du masquage classique de l'EFR En effet, le décodage de parole à entrées souples maintient une « continuité » de signal à l'écoute tout en atténuant fortement les distorsions de parole.

Cependant, les méthodes proposées dans l'état de l'art sont trop complexes pour une application à un système tel que le GSM EFR. Notre principale contribution a été de réduire la *complexité* du décodage de parole à entrées souples. Nous modélisons pour cela la redondance résiduelle au niveau des paramètres du codeur de parole à l'aide de *mélanges de gaussiennes* (GMM). La compacité de ce modèle de la redondance résiduelle permet de réduire la complexité de l'estimation des paramètres transmis d'un facteur 10. De plus, l'utilisation de densités continues (gaussiennes) améliore les capacités de généralisation du modèle appris sur un corpus de parole donné, et par conséquence, le conditionnement de l'estimation des paramètres. Par ailleurs, la nature analytique de la modélisation effectuée permet d'extraire des informations supplémentaires comme celles relatives à la classification en états de la parole (par exemple, « voisé » et « non-voisé »).

Des algorithmes basés sur ce principe de modélisation par GMM ont été proposés dans le cadre du GSM EFR afin de prendre en compte la corrélation temporelle (AK1) des paramètres ainsi que la corrélation entre paramètres d'une même trame (AK2). L'algorithme exploitant la corrélation temporelle (AK1) offre un gain de l'ordre de 0,4 MOS en qualité perçue (note PESQ) par rapport à la procédure de masquage classique de l'EFR pour des niveaux de C/I compris entre 2dB et 4dB tout en convergeant vers la qualité nominale de l'EFR en l'absence d'erreur. Cependant, le modèle de corrélation temporelle fixe utilisé parait trop rudimentaire pour améliorer de manière significative la qualité de la parole pour les niveaux de C/I intermédiaires (entre 3dB et 6dB).

Dans le cas du canal GSM, l'intérêt de la modélisation de la corrélation intra-trame (AK2) est moindre. En effet, on observe que les paramètres d'une même trame sont simultanément corrompus par les erreurs introduites sur le canal radio-mobile et cela de manière relativement uniforme.

Approche SCCD

Les travaux effectués dans le cadre de l'approche SCCD ont poursuivis deux buts. D'une part, celui de la complexité minimale, et d'autre part, celui de la prise en compte de la corrélation entre bits d'une même trame afin de modéliser la non-uniformité des index de quantification. Cependant, les performances des algorithmes de SCCD sont limitées par le critère même qu'elle utilisent (taux d'erreur binaire) qui est nettement moins corrélé avec la qualité perçue que le critère MMSE mis en œuvre par le SBSD. D'autre part, la prise en compte de toutes les corrélations intra-trame entre bits n'est pas compatible, pour une complexité raisonnable, avec le multiplexage et la dispersion des bits à l'intérieur d'une trame effectués avant codage canal dans le GSM.

On notera cependant que l'algorithme très peu complexe exploitant la corrélation bit à bit entre trames (ou sous-trames) conserve un intérêt si l'on a optimisé l'étiquetage des centroïdes de la quantification vectorielle (*Index Assignment*) de sorte que la topologie dans le domaine des bits codant les index de quantification soit proche de celle dans le domaine des centroïdes.

Discussion par rapport aux développements récents et perspectives

Nous analysons ici les travaux présentés dans ce document à la lumière des derniers développements recensés dans le domaine du *décodage conjoint source–canal* et nous esquissons les perspectives qui nous semblent intéressantes à suivre.

Approche SBSD

L'approche proposée pour la réduction de complexité demeure relativement originale. Les autres approches publiées à ce jour [Fingscheidt et al., 2000], [Lahouti, 2003, Report] correspondent à un sous-échantillonnage des dictionnaires de quantification en classes. Les probabilités de transition entre éléments du dictionnaire de quantification se réduisent alors aux probabilités de transition entre les classes obtenues, c'est-à-dire qu'elles sont supposées invariantes à l'intérieur d'une classe. Ceci peut sembler similaire avec notre approche, cependant cette hypothèse est complètement empirique alors que dans notre cas, la distribution multi-gaussienne (GMM) effectue automatiquement la partition de l'espace joint des paramètres dont on modélise la corrélation. De plus, la GMM permet d'introduire la notion de probabilité a priori d'une « classe » (associée à une gaussienne) ce qui peut être utile pour modéliser des « états » de parole.

Comme on l'a mentionné, un modèle de corrélation temporelle (AK1) prenant en compte le comportement non-stationnaire des paramètres de la parole paraît nécessaire pour améliorer le gain par rapport à la prise en compte de la seule non-uniformité (AK0). C'est dans ce but que la perspective d'une modélisation des états « voisé »/« non-voisé » a été proposée au Chapitre 5. Une autre voie pour l'amélioration de la prise en compte de la corrélation temporelle est de considérer des corrélations sur un horizon supérieur à une trame. Ceci serait notamment pertinent pour les résidus de prédiction de manière à pouvoir modéliser la corrélation au niveau du signal reconstruit et non plus au niveau du résidu. [Lahouti, 2003] propose ainsi une décodeur MMSE par séquence mais cet algorithme présente une complexité élevée et impose un délai de décodage.

Approche SCCD

Il y a forcément intérêt à prendre en compte la redondance *résiduelle* du codeur parole avec la redondance *systématique* introduite par le codeur canal. En effet, l'hypothèse d'une *correction induite* demeure valide [Hindelang, 2000]. Cette correction induite signifie que l'introduction d'une information a priori sur un bit permet de réduire également le taux d'erreur binaire des bits voisins dans la trame codée (par l'intermédiaire de l'effet mémoire du code convolutif). En fait, la démarche proposée pour le SCCD dans ce document ne se révèle pas être la bonne.

En premier lieu, il est vain d'essayer de modéliser la redondance intra-trame au niveau du décodeur canal. Comme on l'a observé, le multiplexage et la dispersion des bits qui en résulte à l'intérieur de la trame contraint cette modélisation de la redondance intra-trame or ce multiplexage est nécessaire pour assurer l'indépendance des erreurs pour les bits codant un même paramètre, ce qui est souhaitable. A l'époque où nous avons développé les algorithmes présentés dans ce document, le décodage SCCD était vu comme une alternative concurrente du SBSD. Ceci expliquait les tentatives de modéliser intégralement la redondance résiduelle au niveau du décodeur canal. Cependant, comme on l'a vu, le critère d'erreur utilisé par le SCCD le disqualifie par rapport au SBSD lorsqu'on s'intéresse à la qualité perçue. Une solution est de mettre en œuvre un critère MMSE au niveau du décodeur canal comme cela a été proposé par [Heinen et al., 2000] mais cette solution est très complexe et nécessite également que les bits codant un même paramètre ne soient plus dispersés à l'intérieur de la trame codée.

La combinaison du SCCD avec le SBSD par simple concaténation comme envisagée au Chapitre 7 ne conduit pas non plus à des résultats satisfaisants parce que l'information *a priori* est comptabilisée deux fois (au niveau de chacun des décodeurs). L'approche pertinente qui a été développée depuis consiste à appliquer le principe des Turbo-codes à la combinaison SCCD et SBSD. Plus précisément, la redondance résiduelle est traitée comme un *code externe* et la redondance systématique (introduite par le codeur canal) est vue comme un *code interne*. La mise en série du SCCD et du SBSD signifie que ces deux codes sont concaténés. On applique alors le principe du décodage itératif des codes concaténés (Turbo-Codes) à cette combinaison [Hagenauer et al., 2003]. Cette approche est désormais très développée et permet d'obtenir des gains substantiels au bout de quelques itérations.

Remarques générales

Comme on l'a déjà mentionné, l'approche conjointe source–canal peut également être mise en œuvre à l'émetteur. On a ainsi cité l'étiquetage optimal des centroïdes de la quantification vectorielle (*Index Assignment*) ou la quantification par algorithme LBG prenant en compte les probabilités de transition du canal (*Channel Optimized VQ*). Une autre technique développée au niveau de l'émetteur est celle de l'allocation optimale de débit binaire (ou *équilibrage des rendements*) entre codeur de source et codeur canal en fonction de la qualité du canal observée et de la distorsion visée. Cette technique d'équilibrage adaptatif des rendements est notamment utilisée par l'AMR (*Adaptive Multi-Rate*). Les algorithmes de décodage conjoint source–canal présentés dans ce document peuvent être combinés avec ces techniques mises en œuvre à l'émetteur. Ainsi, l'étiquetage des centroïdes peut être optimisé pour augmenter les performances du décodage SCCD [Hindelang, 2000].

On notera cependant qu'un inconvénient de l'utilisation d'un modèle *a priori* au niveau des décodeurs est que l'on se spécialise sur la classe de signaux modélisés par cette information *a priori*. Autrement dit, la robustesse du décodage conjoint source–canal lorsque le signal transmis est assez éloigné de ceux utilisés pour l'apprentissage (cas d'un signal bruité par exemple) reste à valider.

Annexes

Annexe A

Le codage de parole dans le GSM

A.1 Principes et stratégies du codage de parole

Nous présentons ici, de manière très synthétique, les principes de base du codage de source. Le but est de faire ressortir les éléments clés d'un codeur de parole que sont la *quantification* et la *modélisation*. Nous rappelons d'abord le rôle du codeur source.

Nous considérons ici la *numérisation* comme un processus en amont du *codage source*. La source échantillonnée à la période T_s est quantifiée scalairement sur d éléments binaires $\{0;1\}$ avec une résolution suffisamment fine pour être considérée comme « source non-codée ». Le débit binaire de la source « non codée » $s(nT_s)$ est $D_s = d/T_s$. Le rôle du codeur source est de *réduire le débit binaire* de la source à transmettre, comme schématisé Figure A.1.

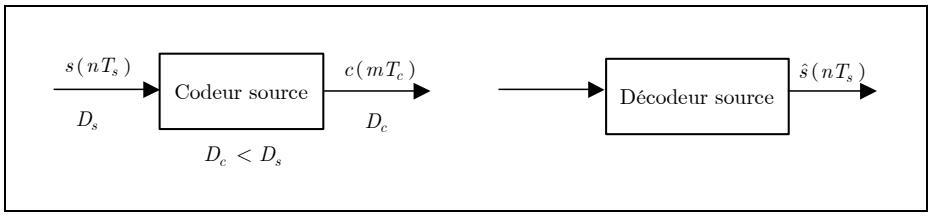

Figure A.1 : Codage de source

Le codeur associe à la séquence de symboles $s(nT_s)$, une séquence de symboles $c(mT_c)$, parfois appelés *mots de code source*. On notera que l'indice m ne réfère pas forcément aux instants d'échantillonnage mais éventuellement à des multiples de ces instants $(T_c = NT_s)$. La séquence

$c(mT_c)$ doit avoir un débit *moyen* $D_c < D_s$. Le décodeur est chargé de reconstruire une approximation $\hat{s}(nT_s)$ de la source numérique à partir de la séquence $c(mT_c)$, éventuellement bruitée. Formellement, la réduction du débit binaire peut s'effectuer de deux manières :

- Codage sans pertes

On réalise un codage sans pertes s'il y a bijection entre les $K = 2^d$ symboles d'entrée s et les L symboles de sortie c du codeur[83]. La réduction du débit *moyen* s'effectue alors en utilisant des symboles c de longueur variable et en associant aux symboles d'entrée les plus probables, les mots de code les plus courts. C'est le principe du codage d'Huffman. L'entropie de la source est inchangée.

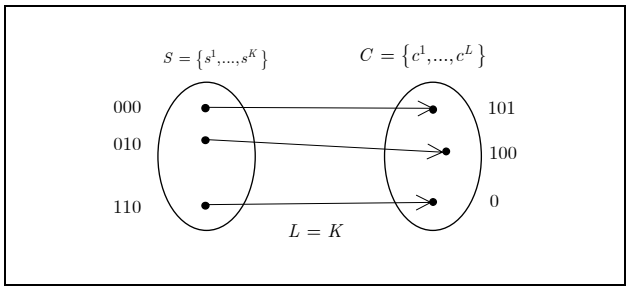

Figure A.2 : Codage sans pertes

- Codage *avec distorsion* ou quantification

On considère dans ce cas des symboles de sortie c de longueur fixe, égale à k éléments binaires. Le seul moyen de diminuer le débit binaire est d'avoir :

$$k < \frac{T_c}{T_s} d \tag{A.1}$$

Dans le cas $T_c = T_s$, un symbole c est délivré en sortie pour chaque symbole en entrée mais un même symbole c peut coder plusieurs symboles en entrée. Ceci correspond à une *quantification scalaire.*

Dans le cas $T_c / T_s = N$, avec N entier, on code une séquence de N symboles d'entrée par un seul symbole c en sortie. L'équation (A.1) signifie alors qu'un même symbole c peut représenter plusieurs séquences, ou vecteurs de dimension N, de symboles d'entrée. Ceci correspond à une *quantification vectorielle.* En considérant des séquences de symboles d'entrée, la quantification vectorielle prend en compte la mémoire de la source, ce qui permet de minimiser la distorsion [Moreau, 1995]. Le codage de source est donc fondamentalement une opération de quantification, comme l'illustre schématiquement la Figure A.3 où les symboles d'entrée peuvent être des scalaires ou des vecteurs.

[83] Une autre manière d'effectuer un codage sans pertes est la transmission discontinue (T_c variable).

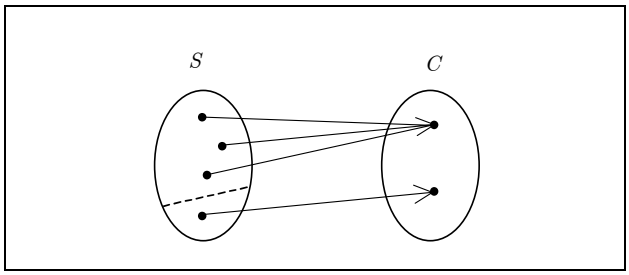

Figure A.3 : Codage avec distorsion

La distinction entre quantification vectorielle et scalaire n'est pas le seul discriminant entre approches de codage. Afin de situer l'approche retenue pour les codeurs GSM par rapport aux autres, nous envisageons succinctement les principaux schémas de codage que l'on peut bâtir autour d'une quantification.

A.1.1 Schémas de codage de la parole

L'élément de base du codage source est la quantification. On distingue généralement deux grandes classes de codeurs de parole, selon la manière dont est utilisée la quantification (indépendamment du fait qu'elle soit scalaire ou vectorielle).

- **Les codeurs paramétriques**

Ces codeurs sont entièrement basés sur un *modèle de production* de la parole. Le codeur estime les paramètres de ce modèle et les quantifie en minimisant une distorsion dans l'espace des paramètres. Au décodeur, le modèle de production est utilisé avec les paramètres quantifiés pour synthétiser un signal de parole. Ce schéma est illustré Figure A.4. De part les approximations du modèle de production utilisé, le signal reconstruit ne converge pas forcement vers le signal de parole originel lorsqu'on augmente la résolution du quantificateur. Ces codeurs permettent de préserver l'intelligibilité de la parole à des débits réduits (<4kbit/s) au prix d'une dégradation importante du naturel de la voix.

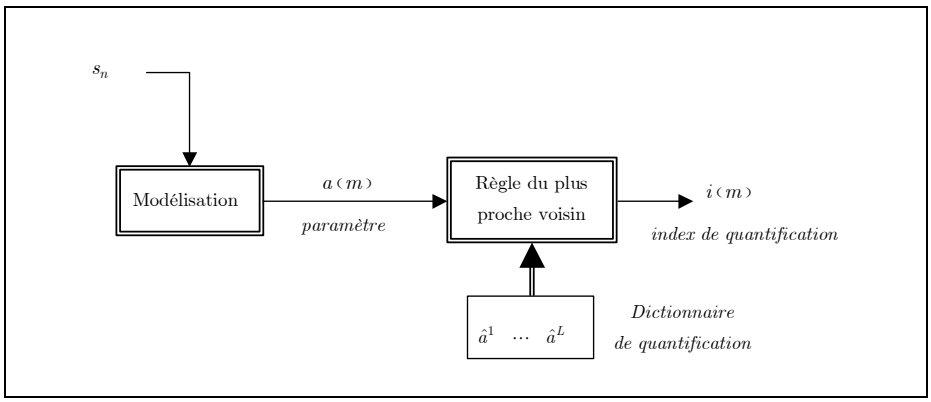

Figure A.4 : Schéma d'un codeur paramétrique

- **Les codeurs de formes d'onde**

Pour ces codeurs, le quantificateur minimise une distance dans l'espace du signal de parole (ou d'une transformation linéaire de ce signal). A l'émetteur, on considère un vecteur[84] de N échantillons à transmettre. On recherche son plus proche voisin dans un dictionnaire composé de L vecteurs représentants selon une distance que nous préciserons par la suite. Le symbole transmis par le codeur est l'indice i du vecteur choisi, cet indice est codé par k éléments binaires ($L = 2^k$). Au récepteur, on dispose du même dictionnaire de représentants et le décodage se résume à la recherche du vecteur d'indice reçu au sein du dictionnaire. Le schéma ainsi décrit correspond aux éléments en traits pleins de la Figure A.5. Ce schéma permet d'exploiter la redondance temporelle de la source si $N > 1$, dans le cas scalaire seule la non-uniformité de la source est exploitée. De plus, l'utilisation d'un dictionnaire fixe est mal adaptée à des signaux non-stationnaires comme la parole.

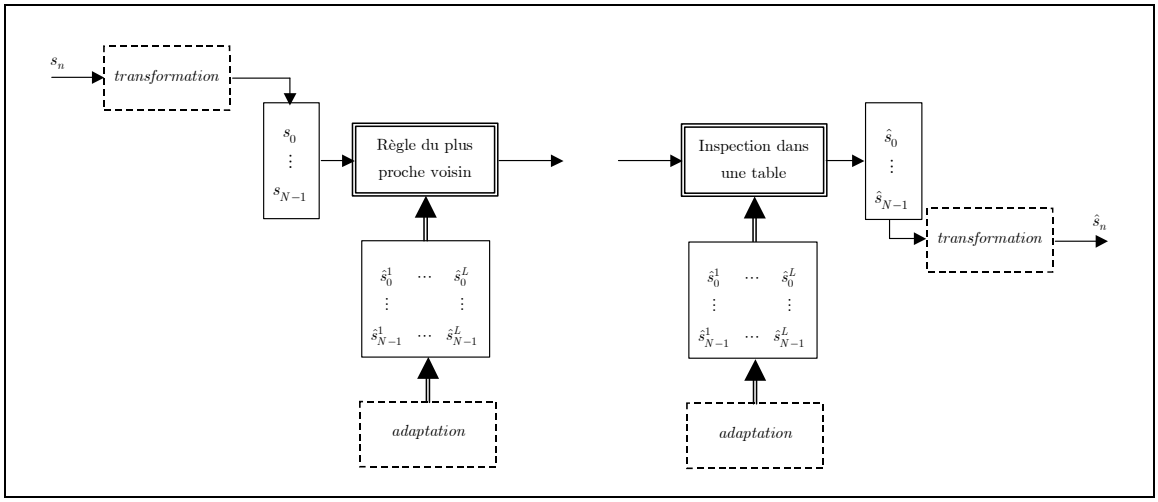

Figure A.5 : Forme générique d'un codeur forme d'onde

Pour tenir compte de la non-stationnarité du signal et mieux exploiter sa redondance (*éviter des N élevés*), on rajoute des traitements autour de la quantification. Ceux-ci correspondent aux blocs en traits pointillés de la Figure A.5. On peut, soit transformer le signal à quantifier de manière à l'adapter au dictionnaire fixe utilisé (bloc *transformation*), soit modifier régulièrement le dictionnaire de quantification de façon à ce qu'il soit adapté à chaque instant à la statistique du signal (bloc *adaptation*).

La première méthode cherche à décorréler au maximum le signal pour pouvoir utiliser un quantificateur simple (de type scalaire). Cette décorrélation peut se faire, soit par application d'une

[84] On se place ici dans le cas d'une quantification vectorielle, le cas scalaire s'obtient pour des vecteurs de dimension $N = 1$.

transformée (transformée en cosinus discrète, décomposition en sous-bandes), soit par une prédiction[85] du signal comme le codeur MICDA à 32kbits/s.

La seconde méthode s'utilise avec un quantificateur vectoriel et permet d'adapter la corrélation modélisée par les vecteurs du dictionnaire à celle observée dans la source. Il existe de nombreuses façons de modifier le dictionnaire. La règle d'adaptation peut être prédéterminée comme, par exemple, dans le cas d'une quantification vectorielle à états finis. Le dictionnaire adaptatif peut aussi être généré à l'aide d'un modèle de production du signal, dont on estime régulièrement les paramètres. Ces paramètres doivent généralement être transmis au décodeur. On choisit de les quantifier séparément selon le schéma de la Figure A.4, c'est-à-dire en minimisant une distance sur les paramètres et non pas sur la parole. On parle alors de codeurs hybrides. Cette approche est celle des codeurs CELP (Code Excited Linear Prediction) illustrés Figure A.6. Ces codeurs permettent de conserver une bonne qualité du signal de parole jusqu'à des débits de 8kbit/s et sont utilisés pour les applications mobiles du type GSM.

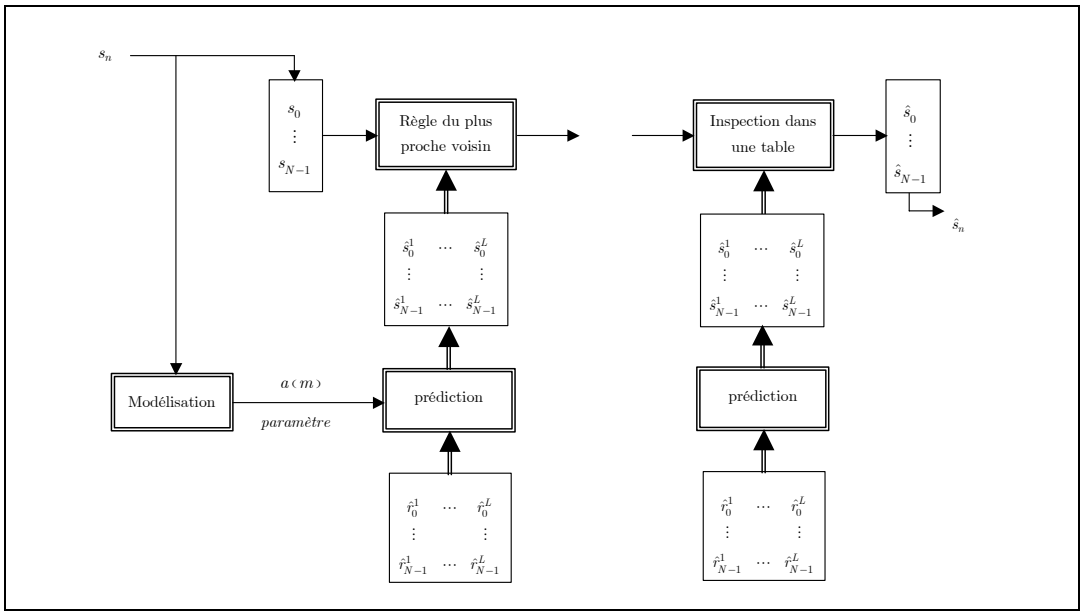

Figure A.6 : Principe d'un codeur CELP

A.1.2 Compromis du codage source

On a vu que la réduction du débit binaire entraîne, le plus souvent, une distorsion avec le signal reconstruit. Le premier compromis auquel le codeur source doit satisfaire est donc un *compromis entre débit et distorsion*. Cependant d'autres critères interviennent également dans ce compromis. Ainsi,

[85] Pour éviter la propagation d'erreur, la prédiction doit se faire à partir de $\hat{s}(n)$. On parle de schéma en boucle fermée, ceci sera précisé au paragraphe A.4.

dans le cas des communications radiomobiles, les principales contraintes pour le codeur de parole sont les suivantes :

- Débit binaire faible (inférieur à 16 kbits/s) afin de minimiser l'occupation spectrale. L'efficacité spectrale est en effet primordiale en raison des restrictions sur le plan des fréquences radio.
- Qualité subjective de la parole la plus proche possible de la qualité téléphonique standard (« bande téléphonique »).
- Délai faible afin de permettre une communication 2 voies (*full duplex*)
- Robustesse aux erreurs de transmission, ce point est particulièrement important pour les transmissions radiomobiles qui introduisent des erreurs par paquets.

Ces contraintes ont orienté le choix vers les familles de codeurs prédictifs excités par impulsions (GSM FR) ou excités par codes (GSM EFR). Avant d'aborder précisément ces codeurs, nous étudions les deux éléments clés sur lesquels ils sont basés, c'est-à-dire la *quantification* et la *modélisation par prédiction linéaire*.

A.2 Quantification vectorielle

Notre description d'un quantificateur s'est jusqu'ici restreinte à un niveau formel, comme l'application de la règle du plus proche voisin (minimisation d'une distance donnée) et la recherche d'un vecteur représentant dans un dictionnaire. Nous précisons ici quelques points concernant la mise en œuvre d'un quantificateur vectoriel. Nous nous intéressons notamment à la construction du dictionnaire de représentants et à la réduction de la complexité.

A.2.1 Conditions d'optimalité

Au niveau du codeur, la quantification vectorielle peut-être vue comme une opération de classification. Elle effectue une partition de l'espace S des vecteurs d'entrée $\mathbf{s} = \left(s_0 ... s_{N-1}\right)$ en L cellules $\left\{\Omega^1,...,\Omega^L\right\}$. Une cellule est représentée par un symbole ou indice i transmis en sortie du quantificateur. Au décodeur, on associe à l'indice i reçu, un vecteur donné, appelé *vecteur représentant* $\hat{\mathbf{s}}^i$. Cette opération de codage – décodage doit *minimiser la distorsion moyenne* :

$$D = E\left\{d(\mathbf{s}, \hat{\mathbf{s}})\right\} \tag{A.2}$$

où $d(\mathbf{s}, \hat{\mathbf{s}})$ est la mesure de distorsion choisie.

Cependant, il n'existe pas de solutions pour trouver conjointement la partition (*codeur*) et les représentants (*décodeur*) minimisant D. Il existe par contre deux conditions nécessaires d'optimalité. L'une spécifie la structure du codeur optimal étant donné le décodeur, et l'autre celle du décodeur étant donné le codeur :

- **Condition du plus proche voisin**

Etant donné un décodeur et son dictionnaire de représentants $\{\hat{\mathbf{s}}^1,...,\hat{\mathbf{s}}^L\}$, la partition réalisé au codeur doit satisfaire :

$$\Omega^i = \{\mathbf{s} \big| d(\mathbf{s},\hat{\mathbf{s}}^i) \leq d(\mathbf{s},\hat{\mathbf{s}}^j); \forall j\} \tag{A.3}$$

où i désigne l'indice de la cellule.

- **Condition du centroïde**

Etant donné la partition $\{\Omega^1,...,\Omega^L\}$ du codeur, les représentants optimaux satisfont :

$$\hat{\mathbf{s}}^i = \arg\min_{\mathbf{y}} E\big(d(\mathbf{s},\mathbf{y}) \big| \mathbf{s} \in \Omega^i\big) \tag{A.4}$$

Dans le cas usuel où la distance $d(\mathbf{s},\hat{\mathbf{s}})$ correspond à l'erreur quadratique moyenne, éventuellement pondérée, le centroïde correspond au centre de gravité de la cellule Ω^i :

$$\hat{\mathbf{s}}^i = E\big(\mathbf{s} \big| \mathbf{s} \in \Omega^i\big) \tag{A.5}$$

La classe des quantificateurs qui satisfont à ces critères est celle des *quantificateurs statistiques*. Leur dictionnaire modélise la distribution des vecteurs $\mathbf{s}$. Nous décrivons dans ce qui suit une méthode de construction de ce dictionnaire.

A.2.2 Construction du dictionnaire

On utilise une base d'apprentissage caractéristique de la source à coder afin d'estimer empiriquement les statistiques. L'algorithme de classification le plus connu pour la construction du dictionnaire est l'algorithme de *Lloyd-Max généralisé*, également appelé algorithme de la *K-moyenne*. C'est un algorithme itératif vérifiant successivement les deux conditions d'optimalité :

(1) Initialisation : On choisit un dictionnaire initial de L centroïdes $\{\hat{\mathbf{s}}^1,...,\hat{\mathbf{s}}^L\}$.

(2) Partition : Les vecteurs d'apprentissage sont répartis dans les L classes définies par le dictionnaire $\{\hat{\mathbf{s}}^1,...,\hat{\mathbf{s}}^L\}$ en appliquant la règle du plus proche voisin (A.3).

(3) Actualisation : On définit un nouveau dictionnaire en mettant à jour le centroïde à l'intérieur de chaque classe à l'aide de la condition du centroïde (A.5).

(4) Critère d'arrêt : On estime la distorsion moyenne (A.2) obtenue sur la base d'apprentissage. On arrête l'algorithme si la décroissance de la distorsion devient inférieure à un seuil, sinon on passe à l'itération suivante en procédant à une nouvelle partition *(2)* de la base d'apprentissage.

Cet algorithme assure la décroissance de la distorsion moyenne mais ne tend seulement vers un minimum local. Ceci pose la question du choix du dictionnaire initial, choix délicat puisque deux dictionnaires initiaux distincts peuvent conduirent à des minima différents. L'algorithme *LBG* (*Linde, Buzo et Gray*) permet de résoudre ce problème. Son principe est le suivant :

(1) Initialisation : On choisit un dictionnaire initial composé d'un seul vecteur $\hat{\mathbf{s}}^1$ minimisant la distorsion moyenne. C'est le centroïde (A.5) de l'ensemble de la base d'apprentissage.

(2) Division : Chaque centroïde $\hat{\mathbf{s}}$ du dictionnaire courant génère deux vecteurs $\hat{\mathbf{s}}+\mathbf{e}$ et $\hat{\mathbf{s}}-\mathbf{e}$ où $\mathbf{e}$ représente un faible variation dans $\mathbb{R}^N$. Le nouveau dictionnaire ainsi obtenu constitue le dictionnaire initial de l'algorithme de *Lloyd-Max*.

(3) Apprentissage : On applique l'algorithme de *Lloyd-Max* jusqu'à atteindre un minimum local.

(4) Arrêt : On arrête l'algorithme si le dictionnaire a atteint la taille L désirée, sinon on réitère la procédure à partir de l'étape *(2)*.

Le dictionnaire d'un quantificateur statistique n'est absolument pas contraint puisqu'il n'est fonction que de la distribution des vecteurs de la base d'apprentissage. Aussi, un quantificateur statistique nécessite, au codage, une recherche exhaustive parmi le dictionnaire du représentant $\hat{\mathbf{s}}$ minimisant la distance $d(\mathbf{s},\hat{\mathbf{s}})$. Or, la taille des dictionnaires demandée en pratique peut être très large afin de minimiser la distorsion pour un débit fixé. Cette considération motive l'utilisation de dictionnaires possédant une structure imposée. Les quantificateurs ainsi définis sont sous-optimaux au sens de la distorsion moyenne (A.2) mais permettent un codage beaucoup moins complexe.

A.2.3 Réduction de la complexité

Plusieurs méthodes sous-optimales de quantification ont été proposés afin d'éviter une croissance exponentielle de la complexité avec la dimension N des vecteurs d'entrée. On donne ici une description sommaire de certaines d'entre elles, notamment celles utilisées par le GSM.

- **quantification vectorielle par produit cartésien (Split-VQ)**

Plutôt que de considérer un vecteur $\mathbf{s}$ de dimension N élevée, on le décompose en plusieurs sous-vecteurs $\mathbf{s}=[\mathbf{s}_1,...,\mathbf{s}_m]$ de dimensions éventuellement différentes et on quantifie indépendamment chacun des sous-vecteurs. On perd cependant la possibilité d'exploiter la redondance entre sous-vecteurs. Le codeur CELP du GSM EFR utilise une forme dérivée de cette méthode pour quantifier les paramètres du filtre de prédiction.

- **quantification forme-gain**

Dans ce schéma illustré Figure A.7, une quantification vectorielle est utilisée conjointement avec une quantification scalaire. De cette façon, la quantification vectorielle utilise un dictionnaire de vecteurs normés (*forme*) et l'énergie du vecteur est quantifiée indépendamment par la quantification scalaire (*gain*). Cette méthode est relativement optimale car l'énergie d'un vecteur et sa corrélation sont souvent des quantités indépendantes.

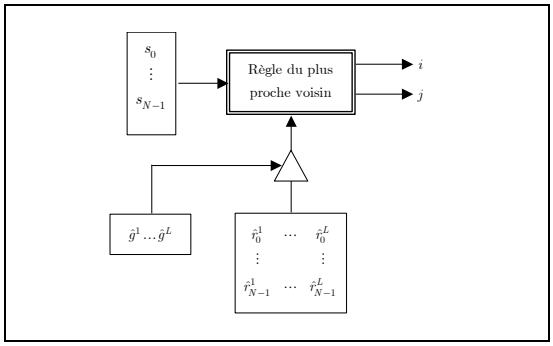

Figure A.7 : quantification de type forme-gain

- **quantification vectorielle multi-étages**

On procède par approximations successives en mettant en cascade plusieurs quantificateurs vectoriels. Chaque étage i quantifie un vecteur résiduel $\mathbf{r}_i$ correspondant à la différence entre le vecteur d'entrée $\mathbf{s}$ et la somme des sorties des quantificateurs précédents :

$$\mathbf{r}_i = \mathbf{s} - \sum_{j=1}^{i-1} \hat{\mathbf{r}}_j \tag{A.6}$$

Le vecteur quantifié $\hat{\mathbf{s}}$ s'obtient par la somme des vecteurs résiduels quantifiés :

$$\hat{\mathbf{s}} = \sum_{i=1}^{m} \hat{\mathbf{r}}_i \tag{A.7}$$

La quantification du signal d'excitation d'un codeur CELP comme le GSM EFR est une forme hybride entre les quantifications de type forme-gain et multi-étage.

- **quantification vectorielle algébrique**

Il ne s'agit plus ici d'un quantificateur statistique. Son dictionnaire n'est pas construit à partir d'une base d'apprentissage, il est prédéterminé. Il consiste à répartir les vecteurs de reproduction de façon régulière dans l'espace. Ces dictionnaires sont souvent utilisés pour modéliser l'excitation des codeurs CELP comme on le verra par la suite.

A.3 Modélisation du signal de parole

La quantification considère un signal dont les propriétés statistiques sont *invariantes* au cours du temps. De plus, si la quantification vectorielle permet de modéliser la corrélation entre les composantes du vecteur d'entrée, elle ne prend pas en compte la corrélation entre vecteurs et la dimension des vecteurs est limitée par la complexité.

Cependant, le signal de parole est non-stationnaire et peut être fortement corrélé sur des segments de plusieurs dizaines de millisecondes. On doit donc se ramener à un signal plus simple, c'est-à-dire plus facilement modélisable par le quantificateur. Ceci peut s'envisager de deux points de vue selon qu'on se place dans l'approche *transformation* ou dans l'approche *adaptation* évoquées Figure A.5. Dans l'approche *transformation*, on extrait la redondance du signal de parole au moyen d'une *prédiction linéaire* afin de quantifier un résidu décorrélé. Dans l'approche *adaptation*, on utilise un *modèle de production* capable de reproduire les propriétés statistiques du signal de parole par *filtrage* du dictionnaire d'un quantificateur. La modélisation *auto-régressive* est souvent choisie car elle se rapproche du modèle du conduit vocal. L'équivalence de la modélisation auto-régressive et de la prédiction linéaire font de celle-ci un élément central de nombreux codeurs de parole. Elle est en particulier utilisée par les codeurs du GSM. Nous présentons ici en détail l'application de la prédiction linéaire au codage de la parole.

A.3.1 Caractéristiques du signal de parole

Considérons le signal de parole représenté sur la Figure A.8. Il est clair que ce signal est *non-stationnaire*; cette caractéristique se traduit par des propriétés statistiques qui varient au cours du temps. Cependant, on peut approximativement le considérer comme *localement stationnaire* (au second ordre) sur des intervalles de temps de l'ordre de quelques dizaines de millisecondes (généralement 20 à 30 ms).

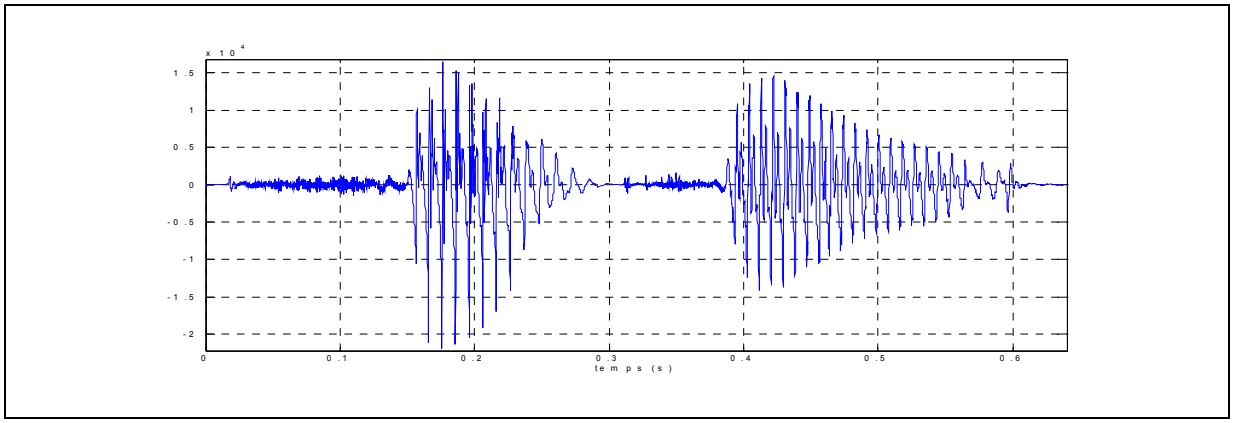

Figure A.8 : Forme temporelle d'un signal de parole

Dès lors une approche commune consiste à segmenter le signal de parole en *trames* sur lesquelles on applique les outils classiques de l'analyse des signaux stationnaires. Une trame de parole est isolée par application d'une fenêtre de pondération $w(n)$ à un instant d'analyse k sur le signal de parole $s(n)$:

$$s_{w_k}(n) = w(k-n)s(n) \tag{A.8}$$

On définit ainsi une auto-corrélation et un spectre à court-terme, pour chaque instant d'analyse k :

$$\gamma_k(m) = \sum_{-\infty}^{\infty} s_{w_k}(n)s_{w_k}(n-m) \tag{A.9}$$

$$\Gamma_k(f) = \left| \sum_{n=-\infty}^{\infty} s_{w_k}(n)e^{-j2\pi f} \right|^2 \tag{A.10}$$

Il existe une relation de dualité entre l'auto-corrélation (A.9) et la densité spectrale (A.10) puisque cette dernière est la transformée de Fourier de l'auto-corrélation. Le spectre à court-terme d'un signal de parole est représenté sur la Figure A.9 pour deux trames successives de 20 ms. On peut y distinguer deux composantes :

- *L'enveloppe moyenne*, représentée en pointillés sur la Figure A.9.
- *La structure fine*, qui correspond aux variations autour de l'enveloppe moyenne.

Par dualité de la transformée de Fourier, *l'enveloppe moyenne* du spectre correspond à une *auto-corrélation à court-terme* du signal de parole et *la structure fine* du spectre à une *auto-corrélation à long-terme*. Ces deux composantes sont décorrélées entre elles, par exemple lorsque la structure fine du spectre exhibe une périodicité fréquentielle (présence d'harmoniques), celle-ci varie souvent plus lentement que l'enveloppe moyenne du spectre.

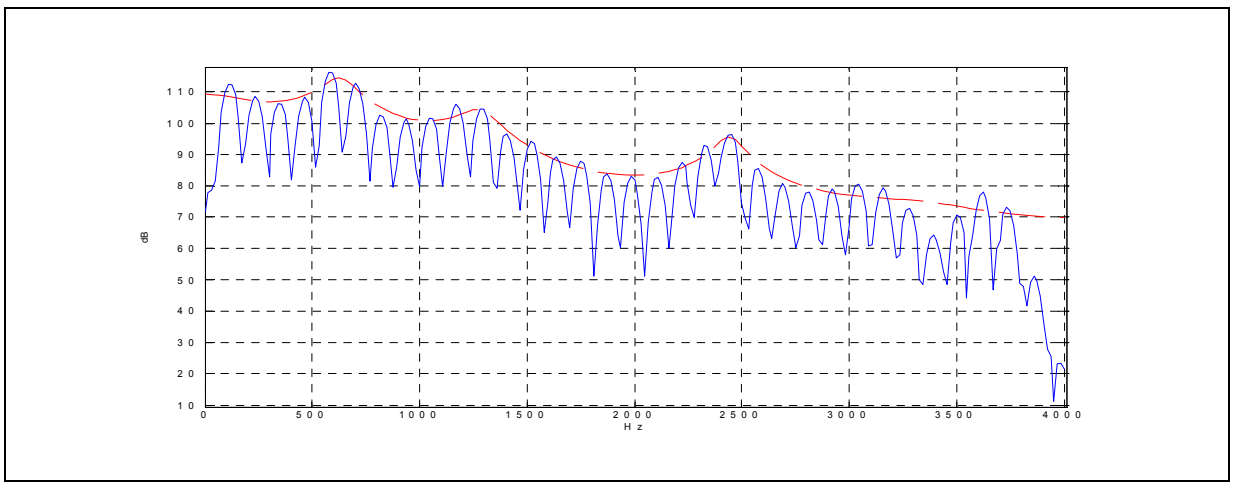

Figure A.9 : Densité spectrale de puissance d'un signal de parole

Au niveau de la perception, l'enveloppe moyenne du spectre est associée au *timbre* d'un son. Dans le cas de la parole, c'est elle qui contient l'essentiel de l'information acoustique pertinente pour la discrimination des phonèmes[86], et donc pour la compréhension de la parole.

Selon la structure fine du spectre, on classe les signaux de parole en deux catégories, les sons *voisés* et les sons *non-voisés.* Pour les sons voisés, la structure fine du spectre contient des raies *harmoniques* associées à la présence d'impulsions périodiques à long-terme dans le signal de parole. L'espacement entre harmoniques définit la *fréquence fondamentale* de la parole. Au niveau de la perception, celle-ci est associée à la *hauteur* de la voix. Pour la voix parlée, elle varie de 80 à 200 Hz chez les hommes, de 150 à 450 Hz chez les femmes et de 200 à 600 Hz chez les enfants. Pour les sons non-voisés, la structure fine du spectre est irrégulière, ces sons correspondent aux *fricatives* et s'apparentent à du bruit.

La plupart des segments de parole[87] peuvent rentrer dans cette classification entre *voisés* et *non-voisés.* Il existent cependant des segments *mixtes* pour lesquels le spectre contient à la fois des harmoniques (aux basses fréquences) et présente une structure irrégulière (aux hautes fréquences). Enfin, la catégorie des *plosives* caractérisées par la présence d'une période de silence suivie d'une impulsion est problématique pour le codage car elle représente une non-stationnarité forte de la parole.

Ces caractéristiques du signal de parole reflètent son mode de production par l'appareil phonatoire illustré Figure A.10. La production du signal de parole peut être divisée en 2 étapes :

- Génération d'un *signal d'excitation* par l'ensemble poumons – cordes vocales.

Les poumons génèrent un souffle d'air circulant à travers les cordes vocales. Lorsque celles-ci sont tendues, le souffle d'air les met en mouvement oscillatoire, il en résulte un signal d'excitation pseudo-périodique. La fréquence d'oscillation des cordes vocales (*fréquence fondamentale*) est contrôlée par leur tension. Lorsque les cordes vocales sont détendues, le souffle d'air n'est plus modulé et le signal d'excitation est assimilable à un bruit[88]. Le signal d'excitation détermine donc la *structure fine du spectre* de la parole.

- *Filtrage* du signal d'excitation par le conduit vocal.

Le conduit vocal, qui regroupe la cavité buccale, la cavité nasale, la langue et les lèvres, joue le rôle d'un filtre amplifiant certaines fréquences appelées *formants.* Les formants permettent d'identifier complètement une voyelle. C'est la déformation du conduit vocal qui produit l'articulation de la parole. Le filtre du conduit vocal est associé à *l'enveloppe moyenne du spectre* de la parole.

---

[86] Un phonème correspond à l'unité acoustique élémentaire convoyant un sens linguistique.

[87] Les sons voisés représentent en moyenne 80% du temps de phonation.

[88] Le signal d'excitation pour les sons non-voisés ne contient pas d'information perceptuelle et peut être remplacé indistinctement par un bruit blanc.

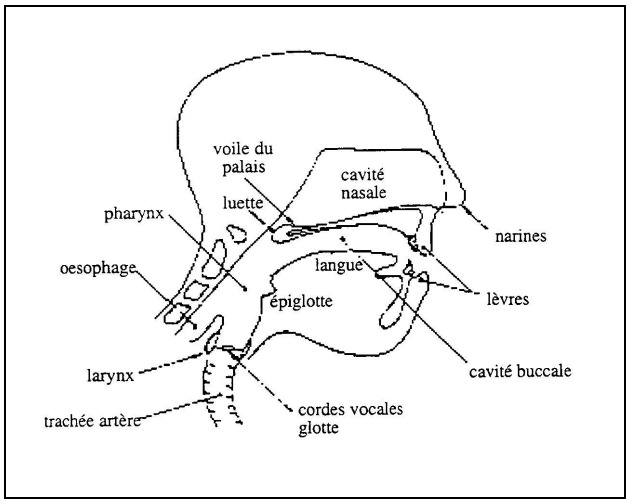

Figure A.10 : l'appareil phonatoire

Ce bref aperçu des caractéristiques du signal de parole a permis de préciser la nature de la redondance présente dans le signal de parole. Nous avons, de plus, relié la corrélation à court-terme à une opération de filtrage induite par le conduit vocal. En reprenant, les deux point de vue possibles de la prédiction linéaire, nous montrons dans ce qui suit comment celle-ci peut fournir un modèle du *filtre* du conduit vocal ou être utilisée pour décorréler le signal de parole.

A.3.2 Prédiction linéaire de la parole

La prédiction linéaire joue un rôle prépondérant dans le codage de la parole. Nous détaillons ici ses relations avec le signal de parole ainsi que sa mise en oeuvre. Nous présentons en premier lieu le modèle source-filtre qui assimile le conduit vocal à un filtre auto-régressif. Ce modèle source-filtre fournit une interprétation spectrale des coefficients de prédiction linéaire en les reliant au spectre à court-terme et aux *formants*. Nous rappelons ensuite brièvement le principe de la prédiction linéaire avant d'aborder le problème de la quantification des coefficients de prédiction linéaire.

A.3.2.a Modélisation du conduit vocal

La distinction des sons entre les deux grandes catégories voisés et non-voisés conduit au modèle de production (simplifié) illustré Figure A.11. Dans ce modèle, l'excitation est représentée par un train d'impulsions périodiques pour les sons voisés ou par un bruit proche d'un bruit blanc pour les sons non-voisés. La glotte et le conduit vocal sont modélisés par un filtre variant dans le temps et de fonction de transfert $H(z)$ de la forme :

$$H(z) = \frac{1 + \sum_{l=1}^{Q} b_l z^{-l}}{1 + \sum_{k=1}^{P} a_k z^{-k}} \tag{A.11}$$

Les zéros du filtre $H(z)$ permettent principalement de modéliser la contribution du conduit nasal lors des sons nasalisés et de certaines fricatives. Les pôles de $H(z)$ sont associés aux formants du conduit vocal, caractéristiques des voyelles.

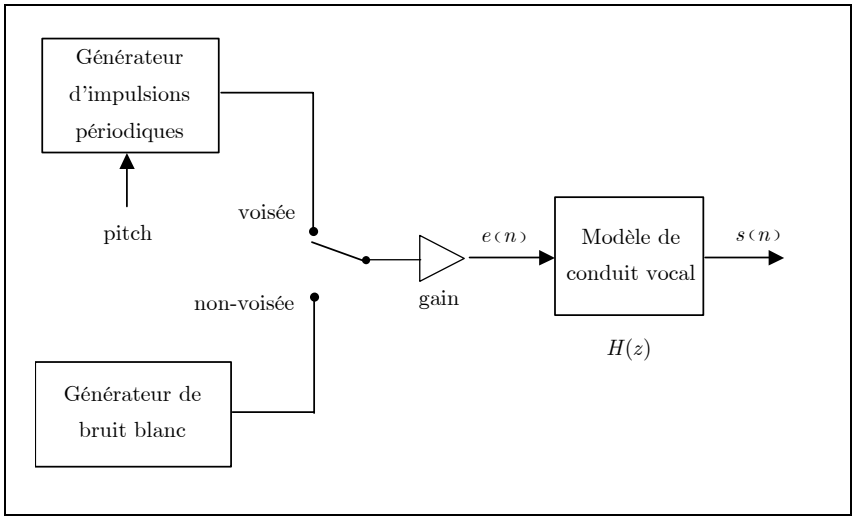

Figure A.11 : Modèle source-filtre de production de la parole.

L'estimation des coefficients du modèle (A.11) nécessite la résolution d'équations non-linéaires. Aussi, pour simplifier, on ne retient que les pôles de la fonction de transfert. Celle-ci correspond alors à un *modèle auto-régressif AR* :

$$H(z) = \frac{1}{A(z)} \tag{A.12}$$

Les coefficients a_k s'obtiennent par résolution d'un système d'équations linéaires comme nous le verrons par la suite. L'ordre P du filtre est généralement choisi de façon à pouvoir représenter un formant par une paire de pôles. La parole présente trois à quatre formants dans la bande téléphonique $[300 - 3400Hz]$, on utilise alors un ordre P compris entre 8 et 16. Les pôles additionnels permettant de modéliser les spectres présentant des zéros (sons nasalisés).

A.3.2.b Analyse par prédiction linéaire

Il y a équivalence entre la modélisation *auto-régressive AR* et la *prédiction linéaire* dont le principe est illustré Figure A.12. En effet, les coefficients de prédiction linéaire s'obtiennent en minimisant l'énergie σ_e^2 du signal d'erreur $e(n)$ défini selon :

$$e(n) = s(n) - \sum_{k=1}^{P} a_k s(n-k) \tag{A.13}$$

Or, d'après le théorème de Parseval [Picinbono, 1989], la norme est invariante par passage du domaine temporel au domaine fréquentiel, soit :

$$\sigma_e^2 = \sum_{-\infty}^{+\infty} e^2(n) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \left|E\left(e^{jw}\right)\right|^2 dw \tag{A.14}$$

où $\left|E\left(e^{jw}\right)\right|^2$ est le spectre d'énergie du signal $e(n)$.

D'après les équations (A.13) et (A.12), le spectre d'énergie $\left|E\left(e^{jw}\right)\right|^2$ peut s'exprimer en fonction des spectres d'énergie $\left|S\left(z=e^{jw}\right)\right|^2$ et $\left|H\left(z=e^{jw}\right)\right|^2$ du signal de parole et du filtre $H(z)$ respectivement. Ceci permet finalement d'aboutir à l'identité suivante :

$$\sigma_e^2 = \sum_{-\infty}^{+\infty} e^2(n) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \frac{\left|S\left(e^{jw}\right)\right|^2}{\left|H\left(e^{jw}\right)\right|^2} dw \tag{A.15}$$

Ainsi, minimiser l'erreur de prédiction linéaire revient à identifier le spectre du signal de parole par celui d'un filtre *AR*.

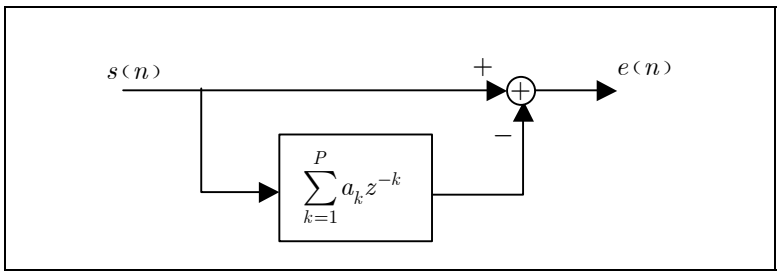

Figure A.12 : prédiction linéaire

Il est alors possible d'estimer les coefficients a_k en minimisant l'expression :

$$\sigma_e^2 = \sum_{-\infty}^{+\infty} e^2(n) = \sum_{-\infty}^{+\infty}\left[s(n) - \sum_{k=1}^{P} a_k s(n-k)\right]^2 \tag{A.16}$$

ce qui conduit au système d'équation :

$$\sum_{k=1}^{P} a_k \gamma_n(i,j) = \gamma_n(i,0)\text{, pour } i = 1,...,P \tag{A.17}$$

avec
$$\gamma_n(i,j) = \sum_{-\infty}^{+\infty} s(n-i)s(n-j)\text{, pour } i = 1,...,P\,;\ j = 1,...,P \tag{A.18}$$

On s'est placé ici dans le cas général de signaux stationnaires, cependant on a vu que le signal de parole ne pouvait être considéré stationnaire que sur un horizon de courte durée. Aussi, on doit borner l'intervalle de sommation dans (A.18). Ceci peut s'envisager de deux manières, correspondant à la *méthode de l'auto-corrélation* et à la *méthode de la covariance*.

- *méthode de l'auto-corrélation*

On limite l'horizon du signal $s(n)$ dans le calcul de la corrélation (A.18) en appliquant une fenêtre d'analyse $w(n)$ de durée N au signal de parole :

$$s_w(n) = w(n)s(n) \tag{A.19}$$

La corrélation (A.18) devient alors :

$$\gamma_n(i,j) = \sum_{-\infty}^{+\infty} s_w(n-i)s_w(n-j), \text{ pour } i = 1,...,P;\ j = 1,...,P \tag{A.20}$$

Comme $s_w(n)$ est nul en dehors de $[0,...,N-1]$, la corrélation ne dépend plus que de $|i-j|$ et la matrice associée au système d'équations (A.17) possède une structure de Toeplitz. Il existe alors un algorithme rapide pour résoudre ce système, c'est l'algorithme de *Levinson-Durbin* [Moreau, 1995].

- *méthode de la covariance*

Au lieu de tronquer le signal $s(n)$ dans (A.18), on borne directement la sommation à l'intervalle $[0,...,N-1]$, ce qui revient à appliquer une fenêtre d'analyse $w(n)$ au signal d'erreur :

$$\sigma_e^2 = \sum_{-\infty}^{+\infty} w(n)e^2(n) \tag{A.21}$$

La corrélation (A.18) s'écrit :

$$\gamma_n(i,j) = \sum_{-\infty}^{+\infty} w(n)s(n-i)s(n-j), \text{ pour } i = 1,...,P;\ j = 1,...,P \tag{A.22}$$

La matrice associée au système d'équations (A.17) n'a cette fois-ci pas de structure de Toeplitz mais le système peut être résolu par décomposition de Cholesky [Markel et al., 1976].

La méthode de l'auto-corrélation, bien que moins précise, est le plus souvent préférée. Elle est peu complexe et garantie que le filtre AR estimé $H(z)$ est stable[89].

Deux méthodes additionnelles sont utilisées pour améliorer l'estimation des paramètres, il s'agit de la *compensation aux hautes fréquences* et de *l'expansion spectrale*. La première applique au signal de parole un filtre de pré-accentuation afin de compenser la décroissance du spectre aux abords de la fréquence de coupure $F_e/2$ du filtre d'échantillonnage. La seconde évite la surestimation des pics spectraux, notamment pour les sons très voisés. Elle consiste à élargir la largeur de bande des formants. Dans le cas de la méthode de l'auto-corrélation, une technique très utilisée [Kleijn et al., 1995] est de pondérer les coefficients d'auto-corrélation par une fenêtre gaussienne, ce qui revient à convoluer le spectre estimé par une gaussienne.

---

[89] La méthode de la covariance nécessite d'être modifiée [Kay, 1988] afin de garantir un filtre stable.

Lorsque la prédiction linéaire est utilisée au codage, les coefficients de *prédiction linéaire LP* sont régulièrement estimés à chaque nouvelle trame de parole et transmis au décodeur[90]. Ceci pose le problème de la quantification des coefficients LP.

A.3.2.c Quantification des coefficients de prédiction linéaire

Comme on l'a vu, la prédiction linéaire peut être intégrée au schéma de codage Figure A.5 comme une opération *d'analyse* (bloc *transformation*) ou comme une opération de *synthèse* (bloc *adaptation*). Dans les deux cas, les coefficients LP sont quantifiés « hors boucle », c'est-à-dire indépendamment du critère d'erreur utilisé pour la quantification du signal résiduel (approche *analyse*) ou du signal de parole reconstruit (approche *synthèse*). On cherchera plutôt à minimiser la *distorsion spectrale* entre les spectres des filtres de synthèse $H(z) = 1/A(z)$ avant et après quantification. Cette approche se justifie par l'interprétation perceptuelle qu'on peut donner au spectre $H\left(e^{jw}\right)$. Une bonne représentation $\left(f\left(a_1\right),...,f\left(a_k\right)\right)$ des coefficients LP doit satisfaire les critères suivants :

- *sensibilité spectrale uniforme* : on souhaite une relation linéaire entre la distance définie sur les coefficients $\left(f\left(a_1\right),...,f\left(a_k\right)\right)$ et une distance sur les spectres des filtres de synthèse associés.

- *critère de stabilité* : on doit pouvoir vérifier la stabilité du filtre de synthèse à partir des coefficients quantifiés.

Les représentations les plus utilisées sont les *coefficients de corrélation partielle PARCOR* et les *Lignes Spectrales par Paires LSP.*

- **Coefficients de Corrélation Partielle**

Les coefficients PARCOR sont des variables intermédiaires de l'algorithme de *Levinson*, ils apparaissent notamment dans l'implémentation du prédicteur sous forme de filtre en treillis [Kondoz, 1994]. Ils sont définis par la récursion :

$$E_0 = \gamma(0), \tag{A.23}$$

$$k_j = \frac{1}{E_{j-1}}\left[\gamma(j) - \sum_{i=1}^{j-1} a_i^{(j-1)}\gamma(j-i)\right], \text{ pour } j = 1,..,P \tag{A.24}$$

où $\gamma(j)$ est l'auto-corrélation (A.9) et les coefficients $\left(a_1^{(j)},...,a_j^{(j)}\right)$ sont ceux du prédicteur optimal d'ordre *j.* Une des propriétés intéressantes des coefficients PARCOR est que la stabilité du filtre $H(z) = 1/A(z)$ est assurée par la condition :

$$\left|k_j\right| < 1 \quad \text{pour } j = 1,...,P \tag{A.25}$$

---

[90] On ne considérera pas l'estimation « backward » des coefficients LP, peu utilisée pour les débits inférieurs à 16kHz.

ce qui procure un test simple de stabilité.

Les coefficients PARCOR n'ont cependant pas une sensibilité spectrale uniforme, c'est pourquoi on leur applique une transformation non-linéaire avant quantification. On utilise le plus souvent les coefficients LAR (Log Area Ratio) définis par :

$$g_j = \log\left(\frac{1-k_j}{1+k_j}\right),\ j = 1,...,P \tag{A.26}$$

Ces coefficients sont ceux utilisés par le GSM Full Rate.

- **Lignes Spectrales par Paires**

Les Lignes Spectrales par Paires [Erkelens et al., 1995] ou de manière équivalente, les Lignes Spectrales de Fréquence LSF, ont été introduites par Itakura comme nouvelle représentation des coefficients LP . On considère le polynôme $A_P(z)$ d'ordre P associé aux coefficients LP et son polynôme réciproque (filtre rétrograde) :

$$B_P(z) = z^{-(P+1)}A_P(z^{-1}) \tag{A.27}$$

On peut alors exprimer le prédicteur d'ordre $P+1$ connaissant le coefficient PARCOR k_{P+1} :

$$A_{P+1}(z) = A_P(z) + k_{P+1}B_P(z) \tag{A.28}$$

On considère les deux configurations $k_{P+1} = +1$ et $k_{P+1} = -1$ définissant les polynômes :

$$P(z) = A_P(z) + B_P(z) \tag{A.29}$$

$$Q(z) = A_P(z) - B_P(z) \tag{A.30}$$

Le filtre $1/P(z)$ correspond au modèle d'un conduit vocal fermé au niveau de la glotte alors que le filtre $1/Q(z)$ est associé à un conduit vocal ouvert au niveau de la glotte.

On montre que les zéros de $P(z)$ et $Q(z)$ sont *entrelacés sur le cercle unité*. Deux zéros sont situés en $\omega = 0$ et $\omega = \pi$, ils sont associés aux valeurs de k_{P+1}. Les P lignes spectrales LSP sont les positions angulaires $0 < w_k < \pi$ des P zéros restants (ou les fréquences $w_k/2\pi$ pour les LSF). Ces paramètres ont des propriétés très intéressantes pour le codage :

- La stabilité du filtre $1/A_P(z)$ est assurée si $0 < \omega_1 < w_2 < ..... < \omega_P < \pi$.
- Les LSF ont une correspondance directe dans le domaine spectral ou elles se regroupent autour des pics spectraux, ceci vient du fait que les LSP correspondent aux formants de deux configurations particulières du conduit vocal.

Leur sensibilité spectrale est donc uniforme et *localisée*. Aussi, une erreur de quantification sur une LSF ne se répercutera que dans le voisinage spectral de cette LSF. Les LSF sont devenues la représentation de prédilection des coefficients LP. Elles sont utilisés par la plupart des codeurs CELP comme le codeur EFR du GSM.

A.4 Codage prédictif de la parole

Nous avons présenté les deux éléments de base d'un codeur prédictif de la parole, à savoir la *prédiction linéaire* et la *quantification*. Le codage prédictif de la parole englobe la plupart des techniques utilisées actuellement pour des débits allant de 16kbits/s à 5kbits/s. Cependant, différentes mises en œuvre du codage prédictif sont possibles, on distingue notamment les méthodes *d'analyse et synthèse*, de celles *d'analyse par synthèse*. Le codeur Full Rate du GSM se classe dans la première catégorie alors que les techniques d'analyse par synthèse correspondent à la famille des codeurs CELP dont le GSM EFR est un représentant. Nous précisons, dans ce qui suit, les mises en œuvre du codage prédictif, notamment en ce qui concerne la détermination de l'excitation.

A.4.1 Schémas de quantification prédictive

Comme nous l'avons vu, la prédiction linéaire peut être utilisée pour extraire la redondance du signal de parole avant quantification du signal résiduel avec un dictionnaire fixe ou pour adapter régulièrement un dictionnaire de quantification aux statistiques du signal de parole. La première approche, représentée par le bloc *transformation* de la Figure A.5, est celle des codeurs dits à *analyse et synthèse*. La seconde approche, représentée par le bloc *adaptation* de la Figure A.5, correspond aux codeurs à *analyse par synthèse*. La première approche implique une optimisation indépendante de chacun des éléments de la chaîne de décodeur. La seconde permet théoriquement l'optimisation jointe de ces éléments. Cependant, pour des raisons de complexité, le prédicteur à court-terme est quantifié indépendamment dans la mise en œuvre de l'approche « analyse par synthèse ». Aussi dans la pratique, la principale distinction entre ces deux approches est la quantification du signal d'excitation. Dans l'approche *analyse et synthèse*, le signal d'excitation est cherché par *minimisation d'une erreur dans le domaine du résidu*. Dans l'approche *analyse par synthèse*, le signal d'excitation est cherché par *minimisation d'une erreur dans le domaine de la parole*.

A.4.1.a Analyse et synthèse

Dans ce schéma, la parole est d'abord analysée et ses paramètres d'analyse quantifiés. Elle est ensuite filtrée par un prédicteur dont les coefficients correspondent aux paramètres d'analyse quantifiés. Le résidu décorrélé en sortie du prédicteur est enfin quantifié. Ainsi dans le schéma d'analyse et synthèse,

la quantification est locale à chaque élément. L'analyse peut être en *boucle ouverte* ou en *boucle fermée*.

- **Analyse en boucle ouverte**

La structure en boucle ouverte, illustrée par la Figure A.13, est la plus immédiate. On notera $s(n|n-1,...,n-P)$, la prédiction du signal $s(n)$ à l'instant n d'après les P échantillons précédents :

$$s(n|n-1,...,n-P) = \sum_{k=1}^{P} a_k s(n-k) \tag{A.31}$$

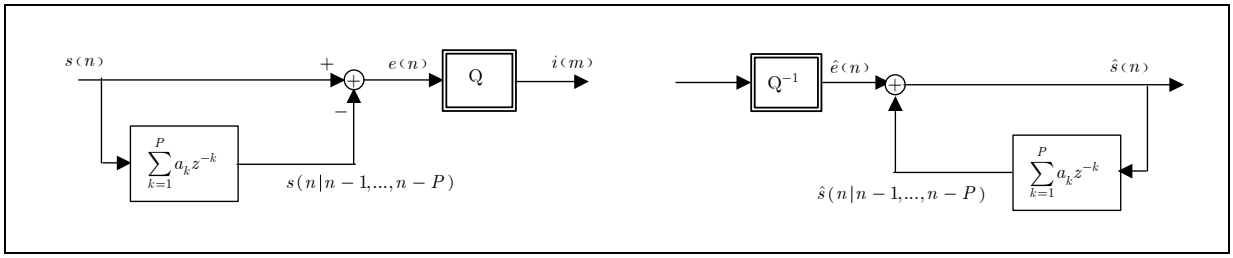

Figure A.13 : Prédiction en boucle ouverte

Cependant, cette configuration « boucle ouverte » peut entraîner une propagation d'erreur au décodage. Considérons l'erreur de quantification du résidu $\varepsilon(n) = e(n) - \hat{e}(n)$, on a pour la structure en boucle ouverte ;

$$\hat{s}(n) - s(n) = \varepsilon(n) + \sum_{k=1}^{P} a_k (\hat{s}(n-k) - s(n-k)) \tag{A.32}$$

Autrement dit, une erreur de quantification constante au codeur peut faire diverger le décodeur. Cette propagation d'erreur sera *d'autant plus forte que le gain du prédicteur linéaire est élevé*.

- **Analyse en boucle fermée**

Cette propagation d'erreur peut être évitée en considérant une structure en « boucle fermée » comme illustrée Figure A.14. Dans une structure en boucle fermée, un *décodeur local* est intégré au codeur[91]. On peut ainsi prédire l'échantillon courant du signal de parole $s(n)$ à partir des échantillons reconstruits précédents $\hat{s}(n-1),...,\hat{s}(n-P)$. On a alors ;

$$\hat{s}(n) - s(n) = \varepsilon(n) = \hat{e}(n) - e(n) \tag{A.33}$$

Cette relation montre qu'une *erreur de quantification dans le domaine du résidu entraîne une erreur identique dans le domaine de la parole*. Elle démontre aussi l'intérêt de la quantification sur le résidu

[91] La présence d'un décodeur local au codeur ne signifie pas que l'on effectue une analyse par synthèse puisqu'ici la parole reconstruite n'est pas utilisée dans le critère d'erreur de la quantification.

de la prédiction linéaire plutôt que directement sur le signal de parole, puisque l'erreur de quantification est proportionnelle à la puissance du signal à quantifier et que la puissance du résidu est inférieure à celle du signal de parole.

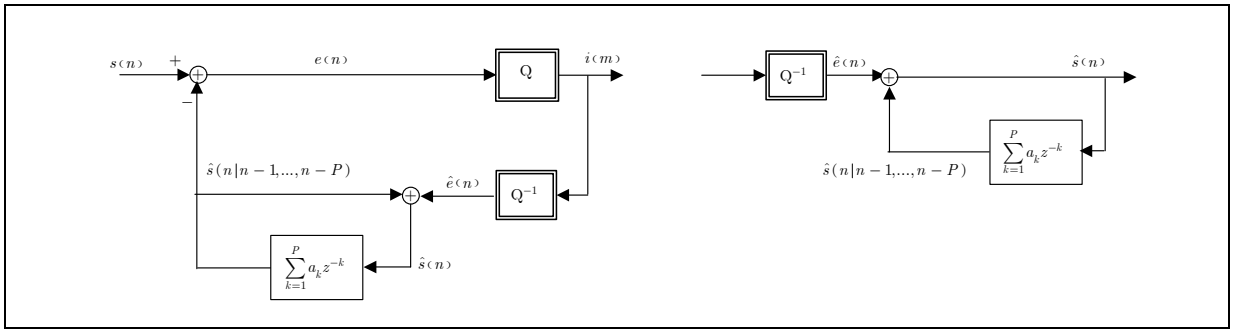

Figure A.14 : Prédiction en boucle fermée avec minimisation de l'erreur sur le résidu

D'un point de vue strictement mathématique, il revient au même, d'après la relation (A.33), de rechercher le signal d'excitation en minimisant une erreur dans le domaine du résidu de prédiction ou une erreur dans le domaine de la parole. Dans la pratique, ceci est faux pour deux raisons. D'une part, la relation (A.33) ne prend pas en compte l'erreur de quantification des paramètres du prédicteur. D'autre part, d'un point de vue perceptuel, c'est l'erreur dans le domaine de la parole qui nous intéresse. Ces argument conduisent aux techniques d'analyse par synthèse.

A.4.1.b Analyse par synthèse

Dans cette approche, on utilise comme critère d'optimisation l'erreur dans le domaine de la parole. Ceci a deux avantages majeurs :

- Minimiser une erreur dans le domaine de la parole permet de quantifier le signal d'excitation en tenant compte de l'erreur introduite par la quantification (séparée dans la pratique) des paramètres du prédicteur.
- Le domaine de la parole se prête bien à l'introduction d'une mesure de distorsion perceptuelle.

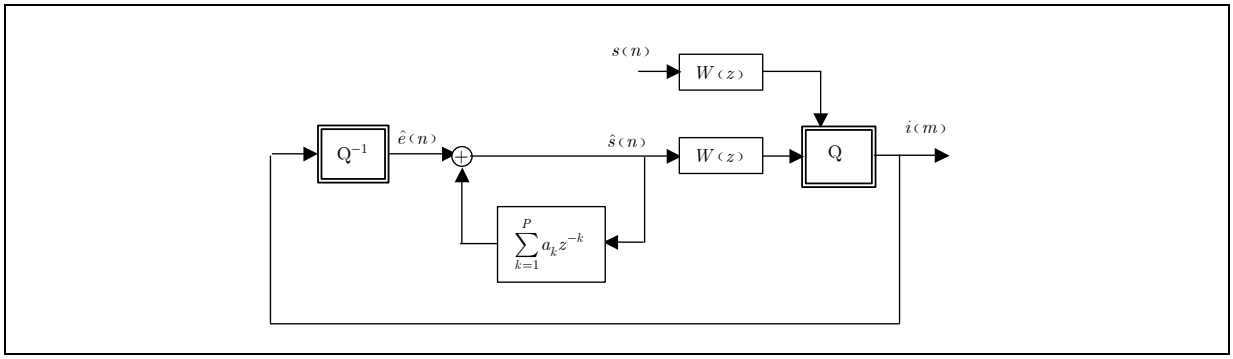

Figure A.15 : Prédiction en boucle fermée avec minimisation de l'erreur sur la parole

La structure d'un codeur à analyse par synthèse est schématisée Figure A.15. Elle utilise également un *décodeur local* au codeur mais considère directement le signal synthétisé $\hat{s}(n)$ pour comparaison avec le signal de parole $s(n)$. La quantification du signal d'excitation utilise comme critère d'erreur une distance perceptuellement pondérée entre $s(n)$ et $\hat{s}(n)$. Cette pondération perceptuelle s'effectue par filtrage des signaux $s(n)$ et $\hat{s}(n)$ par un filtre $W(z)$ modélisant certaines caractéristiques du système auditif. Ainsi, on modélise le phénomène de *masquage fréquentiel* en atténuant la contribution des zones spectrales les plus énergétiques. En effet, l'erreur de quantification est moins perceptible dans ces zones car elle est masquée par le signal. Le filtre $W(z)$ est le plus souvent déterminé à partir du filtre de synthèse LPC et est de la forme :

$$W(z) = \frac{A\left(z/\gamma_1\right)}{A\left(z/\gamma_2\right)}, \text{ avec } 0 \leq \gamma_2 \leq \gamma_1 \leq 1. \tag{A.34}$$

A.4.2 Détermination de l'excitation

Les codeurs prédictifs par analyse-synthèse, comme le GSM FR, et à analyse par synthèse, comme le GSM EFR, diffèrent par leur mode de calcul de l'excitation. On étudie ici les différentes manières de déterminer cette excitation.

A.4.2.a Modélisation de la périodicité à long-terme de l'excitation

Lors des sons voisés, le signal d'excitation présente une forte périodicité à long-terme. Les codeurs par analyse-synthèse utilisent un *filtre prédicteur* afin d'extraire cette redondance à long-terme avant quantification. Les codeurs à analyse par synthèse modélisent le plus souvent cette périodicité à l'aide d'un *dictionnaire adaptatif*.

- **Approche filtrage**

Dans cette approche, la modélisation de la corrélation à long-terme du signal d'excitation $e(n)$ est similaire à celle employée pour la corrélation à court-terme de la parole $s(n)$. Commençons par décrire l'analyse en boucle ouverte selon le schéma de la Figure A.13 où le signal d'entrée à analyser est désormais le résidu de prédiction à court-terme $e(n)$ et le prédicteur utilisé, un prédicteur à long-terme de la forme :

$$P(z) = 1 - \beta z^{-D} \tag{A.35}$$

où β est le coefficient de prédiction et D le délai du prédicteur. Le délai D correspond au pitch estimé et β indique le niveau de voisement associé. Ils s'obtiennent en minimisant l'erreur de prédiction :

$$E_{LTP} = \sum_{-\infty}^{\infty} [e_w(n) - \beta e_w(n-D)]^2 \tag{A.36}$$

où $e_w(n)$ est le résidu de prédiction à court-terme $e(n)$, pondéré par une fenêtre d'analyse. En dérivant E_{LTP} par rapport à β, on obtient[92] :

$$\beta(D) = \frac{\sum_{-\infty}^{\infty} e_w(n)e_w(n-D)}{\sum_{-\infty}^{\infty} e_w^2(n-D)} \tag{A.37}$$

Puis on remplace β par sa nouvelle valeur pour calculer D :

$$E_{LTP}(D) = \sum_{-\infty}^{\infty} e_w^2(n) - \frac{\left[\sum_{-\infty}^{\infty} e_w(n)e_w(n-D)\right]^2}{\sum_{-\infty}^{\infty} e_w^2(n-D)} \tag{A.38}$$

On considère souvent comme constant le numérateur de cette expression. L'estimation du délai D se limite alors à la *recherche du maximum de l'auto-corrélation* de $e_w(n)$. En pratique, le pitch de la parole n'est pas forcément un multiple de la période d'échantillonnage. On utilise alors un délai D avec une *résolution fractionnaire* [Kleijn et al., 1995], en le décomposant en une composante entière et une composante fractionnelle.

Le gain β du prédicteur à long-terme peut être très élevé, ce qui rend instable le schéma en boucle ouverte. Aussi, le prédicteur à long-terme $P(z)$ est toujours implanté selon le schéma en boucle fermée illustré par la Figure A.16. Le critère d'erreur doit donc être modifié pour intégrer le signal d'excitation reconstruit $\hat{e}(n)$:

$$E_{LTP} = \sum_{-\infty}^{\infty} [e_w(n) - \beta\hat{e}_w(n-D)]^2 \tag{A.39}$$

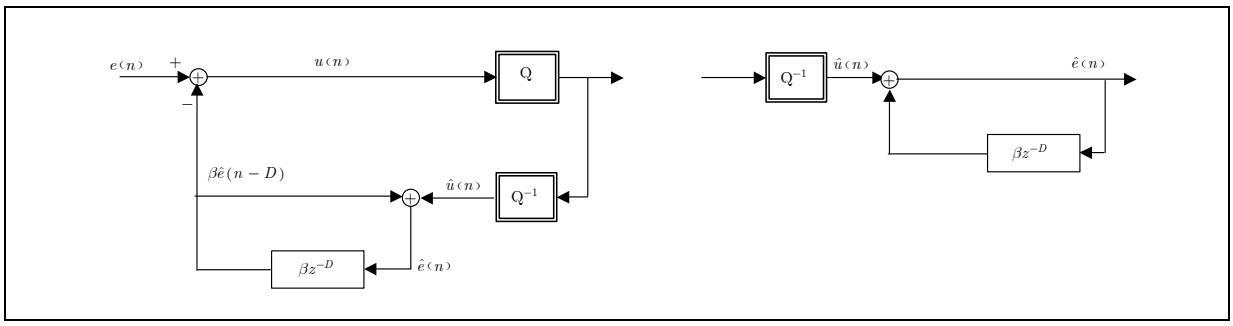

Figure A.16 : Prédiction à long-terme en boucle fermée

[92] On reconnaît ici un cas trivial de la méthode de l'auto-corrélation.

Dès lors un problème se pose si le délai D est inférieur à la longueur L de la fenêtre d'analyse $w(n)$. En effet, compte-tenu de la mémoire du filtre, l'équation (A.39) devient non linéaire en β. Ceci *restreint la recherche du délai* aux valeurs supérieures à la longueur de la trame d'analyse.

Pour contourner cette contrainte, les paramètres du prédicteur à long-terme $P(z)$ sont généralement estimés sur des sous-trames de 5ms, c'est-à-dire 40 échantillons à 8kHz. Une autre motivation du calcul par sous-trames vient du critère d'erreur quadratique moyenne utilisé (A.36), ce critère est mal adapté au pitch puisqu'une faible variation du délai D peut se traduire par une déviance très forte de l'erreur. Certains procédés de codage comme le RCELP proposent de relaxer ce critère d'erreur [Kleijn et al., 1995].

- **Approche dictionnaire adaptatif**

Pour modéliser la périodicité à long terme, on remplace l'approche filtrage par l'utilisation d'un dictionnaire adaptatif peuplé des échantillons $\hat{e}(n)$ du signal d'excitation synthétisé. Ceci est illustré Figure A.17. L'indice dans le dictionnaire est associé à la valeur du pitch D et le gain λ joue un rôle équivalent au coefficient β du filtre prédicteur $P(z)$. Le signal $\hat{u}(n)$ représenté Figure A.17 peut être comparé au résidu du prédicteur à long-terme $P(z)$, il modélise les composantes non-périodiques de l'excitation $\hat{e}(n)$.

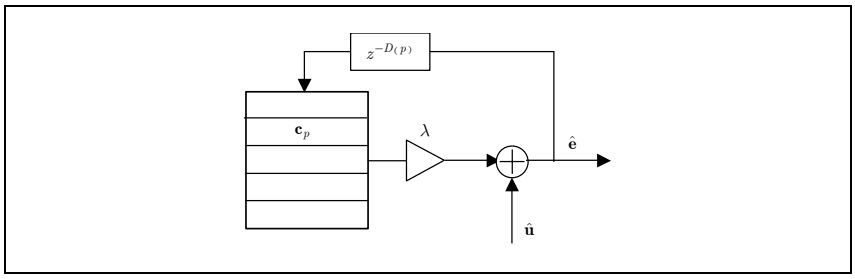

Figure A.17 : Synthèse d'une excitation voisée par dictionnaire adaptatif

Le dictionnaire adaptatif apparaît ainsi comme un buffer à décalage dont chaque vecteur d'indice p est formé par :

$$\mathbf{c}_p = [\hat{e}(n - D(p)), \hat{e}(n - D(p) + 1) ..., \hat{e}(n - D(p) + L - 1)] \tag{A.40}$$

où L correspond à la longueur d'une sous-trame et $D(p)$ au délai (éventuellement fractionnaire) associé à l'indice p. On peut, avec cette approche, modéliser des *délais D inférieurs à la longueur L de sous-trame* en extrapolant le dictionnaire par périodisation [Moreau, 1995] pour les délais $D(p) < L$. Aussi, l'utilisation d'un dictionnaire adaptatif pour modéliser les composantes périodiques de l'excitation est devenue l'approche standard pour les codeurs CELP. Elle est notamment utilisée dans le codeur GSM EFR.

Le délai D (indice dans le dictionnaire) et le gain λ sont cherchés en minimisant une distance perceptuelle sur la parole $d(s_W, \hat{s}_W)$, selon le schéma d'analyse par synthèse illustré Figure A.18. La

quantification du vecteur **u** de *composantes non-périodiques* n'est pas représentée pour plus de clarté. Cette quantification utilise un *dictionnaire fixe associé à un gain*, elle est effectuée après que la contribution du dictionnaire adaptatif ait été fixée. Il s'agit d'une quantification *multi-étages*.

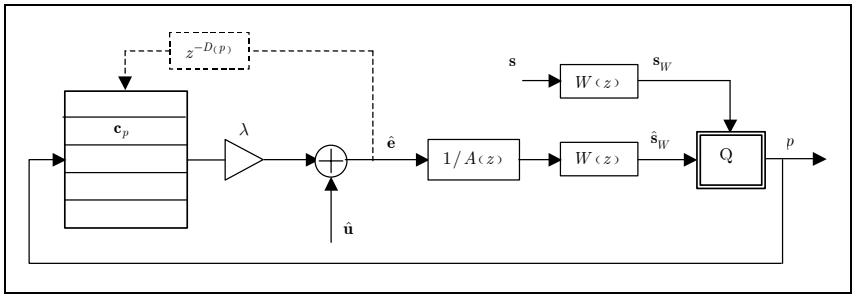

Figure A.18 : Détermination de la composante périodique de l'excitation

La recherche du dictionnaire adaptatif peut être vue comme une forme de quantification *forme-gain*. En effet, on peut écrire :

$$\hat{s}_W = \lambda H c_p + \hat{s}_0 \tag{A.41}$$

où H est la matrice associée à la réponse impulsionnelle tronquée $[h_0, h_1, ..., h_L]$ du filtre $W(z)/A(z)$ et $\hat{s}_0$, la réponse libre[93] du filtre, c'est-à-dire sa réponse à un vecteur d'excitation $\hat{\mathbf{e}}$ nul. Comme $\hat{s}_0$ ne dépend pas de la trame d'excitation courante, la recherche du dictionnaire adaptatif doit minimiser l'erreur :

$$d(s_W - \hat{s}_0, \lambda H \mathbf{c}_p) = \left\| (s_W - \hat{s}_0) - \lambda H \mathbf{c}_p \right\|^2 \tag{A.42}$$

Le gain λ optimal pour une forme $H\mathbf{c}_p$ fixée s'obtient par le théorème de la projection orthogonale :

$$\lambda_p = \frac{\langle \mathbf{s}_W - \hat{\mathbf{s}}_0, H\mathbf{c}_p \rangle}{\left\| H\mathbf{c}_p \right\|^2} = \frac{c_p^T H^T (\mathbf{s}_W - \hat{\mathbf{s}}_0)}{c_p^T H^T H c_p} \tag{A.43}$$

et la forme $H\mathbf{c}_p$ optimale maximise le coefficient de corrélation avec le vecteur cible $(\mathbf{s}_W - \hat{\mathbf{s}}_0)$:

$$p = \arg\max_p \frac{\left| \langle s_W - \hat{s}_0, H\mathbf{c}_p \rangle \right|^2}{\left\| H\mathbf{c}_p \right\|^2} = \max_p \frac{\left[c_p^T H^T (\mathbf{s}_W - \hat{\mathbf{s}}_0) \right]^2}{c_p^T H^T H c_p} \tag{A.44}$$

On reconnaît des relations similaires à celles de l'approche « filtrage ».

Le deuxième étage de quantification, c'est-à-dire la recherche du dictionnaire fixe et de son gain s'effectue de manière analogue. Nous présentons dans ce qui suit les principaux dictionnaires fixes utilisés.

---

[93] Cette réponse représente la mémoire du filtre puisque $W(z)/A(z)$ a une réponse impulsionnelle infinie.

A.4.2.b Modélisation de l'excitation résiduelle

La quantification de l'excitation résiduelle est très importante pour restituer le naturel de la parole. Elle nécessite cependant l'allocation d'un grand nombre de bits puisqu'elle ne contient pas de redondance modélisable. Aussi, c'est un domaine qui fait l'objet de nombreuses recherches et c'est le plus souvent par leur modélisation de l'excitation résiduelle (ou excitation secondaire) que les codeurs se différencient. Les codeurs à analyse par synthèse comme le CELP utilisent des dictionnaires fixes associés à un gain alors que les codeurs par analyse-synthèse utilisent simplement des modèles d'excitation (qu'on peut considérer comme des dictionnaires implicites). Parmi les modèles les plus courants, on trouve :

- les codeurs MPE (Multi-Pulse Excited) où le résidu est modélisé par un train d'impulsions dont le nombre est fixé. L'amplitude et la position de ces impulsions sont laissées libres et subissent une quantification scalaire.

- Les codeurs RPE (Regular Pulse Excited), qui est un modèle sous-optimal du précédent. On choisit la première impulsion mais le nombre d'impulsions et leur espacement est entièrement déterminé.

Les principaux types de dictionnaires utilisés par les codeurs de type CELP correspondent à des séquences de bruit gaussien [Schroeder et al., 1985], à des dictionnaires stochastiques appris sur une base de données, ou bien encore à des impulsion normalisées (+1,-1) dans le cas des dictionnaires algébriques. Une autre variante consiste à utiliser des dictionnaires engendrés par une base restreinte de formes d'ondes combinée à une contrainte sur les coefficients de combinaisons linéaires entre ces formes.

A.5 Le codeur du GSM Full-Rate

Le codeur RPE-LTP [*GSM*, 06.10] analyse des trames de parole de 20 ms (160 échantillons à 8 kHz). Les coefficients du prédicteur à court-terme LPC sont calculés sur l'ensemble de la trame de 20 ms puis transformés en coefficients LAR avant d'être quantifiées. Les paramètres du prédicteur à long-terme LTP et de son signal d'excitation sont réactualisés sur des sous-trames de 5 ms (40 échantillons). Le modèle utilisé pour chaque sous-trame de 40 échantillons du signal d'excitation est un modèle RPE (Regular Pulse-Excited) formé d'impulsions régulièrement espacées tous les 3 échantillons. Les paramètres de ce modèle sont la position de la première impulsion dans la sous-trame (offset ou grille RPE) qui est comprise entre 0 et 3 échantillons, ainsi que l'amplitude de chaque impulsion.

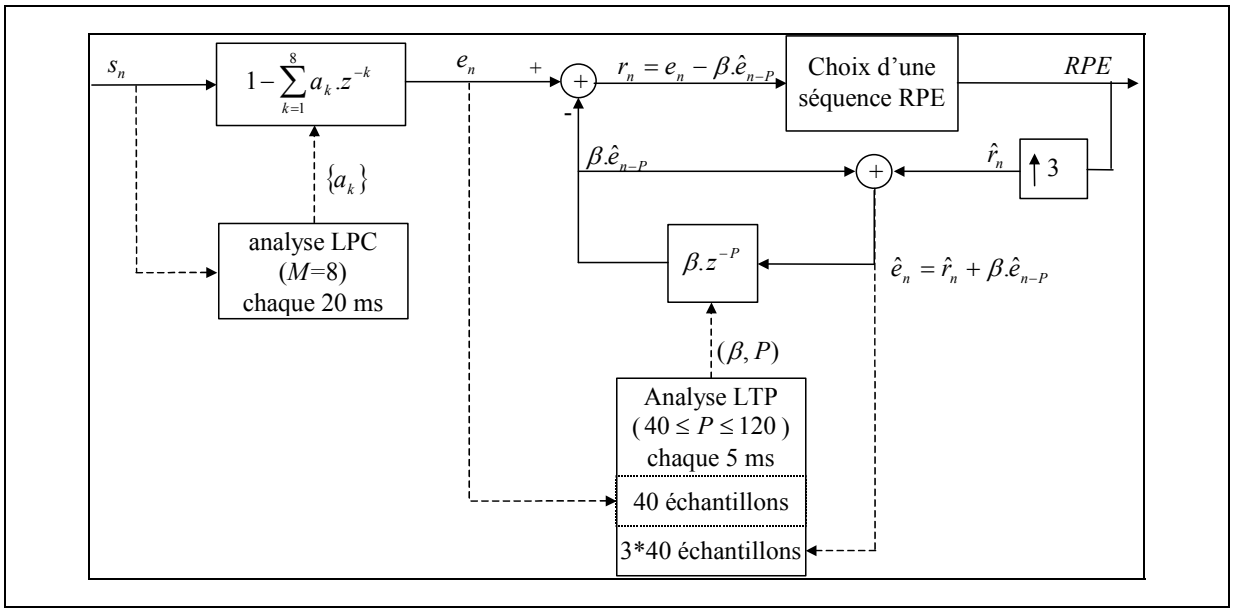

Figure A.19 : Codeur RPE-LTP (analyse en boucle fermée)

L'ensemble des paramètres RPE-LTP définissant 1 jeu de coefficients du filtre LPC, les 4 jeux successifs de coefficients du filtre LTP et les 4 séquences de 5 ms d'excitation RPE associées sont quantifiés sur un total de 260 bits et transmis toutes les 20 ms au décodeur. Ceci correspond à un débit de 13kbit/s. Pour tous les coefficients, la quantification est *scalaire*.

Paramètres	par sous-trame	Total par trame
8 coefficients LAR		36
gain β (filtre LTP)	2	8
délai P (filtre LTP)	7	28
gain de calibration	6	24
grille RPE	2	8
13 amplitudes	39	156
Total		260

Tableau A.1 : Allocation des bits par trame de 20 ms

A.6 Le codeur du GSM Enhanced-Full-Rate

Le codeur du GSM EFR [GSM, 06.60] est basé sur le principe des codeurs CELP illustré Figure A.20. Nous en décrivons ici succinctement les paramètres calculés. Le codeur opère sur des *trames de 20 ms* (160 échantillons à 8 kHz), divisées en quatre *sous trames de 5 ms* (40 échantillons) pour la détermination de l'excitation.

- **Analyse LPC**

Le filtre de prédiction à court-terme LPC comprend 10 coefficients, qui sont estimés deux fois par trame sur des fenêtres asymétriques centrées respectivement sur les sous-trames 2 et 4, et de longueur 30 ms (mémorisation de la trame passée).

Les coefficients LPC sont ensuite transformés en coefficients LSP (Line Spectral Pairs [Erkelens et al., 1995]). La suite formée par les vecteurs de coefficients LSP est partiellement décorrélée en appliquant une prédiction temporelle en *moyenne adaptée MA d'ordre 1*. Les vecteurs de résidus de prédiction ainsi obtenus subissent alors une quantification *vectorielle par produit cartésien* (cf. A.2.3).

Plus précisément, les deux jeux de 10 résidus de prédiction MA (associés aux deux jeux de coefficients LSP par trame) sont divisés en 5 matrices de dimension 2x2, regroupant les valeurs $\left(r_{t,k}^{(LSP)}; r_{t,k+1}^{(LSP)}; r_{t+1,k}^{(LSP)}; r_{t+1,k+1}^{(LSP)}\right)$ où l'indice t est l'indice temporel (demi-trame) et k désigne le $k$ième coefficient du vecteur de résidu $\mathbf{r}_t^{(LSP)}$. Chacune des ces sous-matrices est alors quantifiée séparément.

Les LSP des sous-trames 1 et 3 n'ayant pas été déterminées par analyse LPC du signal, elles seront estimées par interpolation entre les sous-trames précédente et suivante.

- **Dictionnaires d'excitation**

Deux fois par trame, on effectue une première estimation en boucle ouverte du pitch, ce qui permet de restreindre la plage de recherche du pitch par la procédure d'analyse par synthèse. Les index et gains des dictionnaires *adaptatifs et fixes* sont calculés et transmis pour *chaque sous trame*. Le délai de pitch (*lag*) subit une quantification scalaire. Le gain du dictionnaire adaptatif (*gain pitch*) est quantifié en utilisant une quantification scalaire non-uniforme sur 4 bits dans l'intervalle [0 –1.2]. Le gain du dictionnaire fixe (*gain code*) est partiellement décorrélé par une prédiction temporelle MA d'ordre 4. Cette prédiction s'effectue dans le domaine logarithmique. Le résidu de prédiction est alors quantifié sur 5 bits. Enfin, le dictionnaire fixe est représenté par un dictionnaire algébrique de 35 bits [Järvinen et al., 1997].

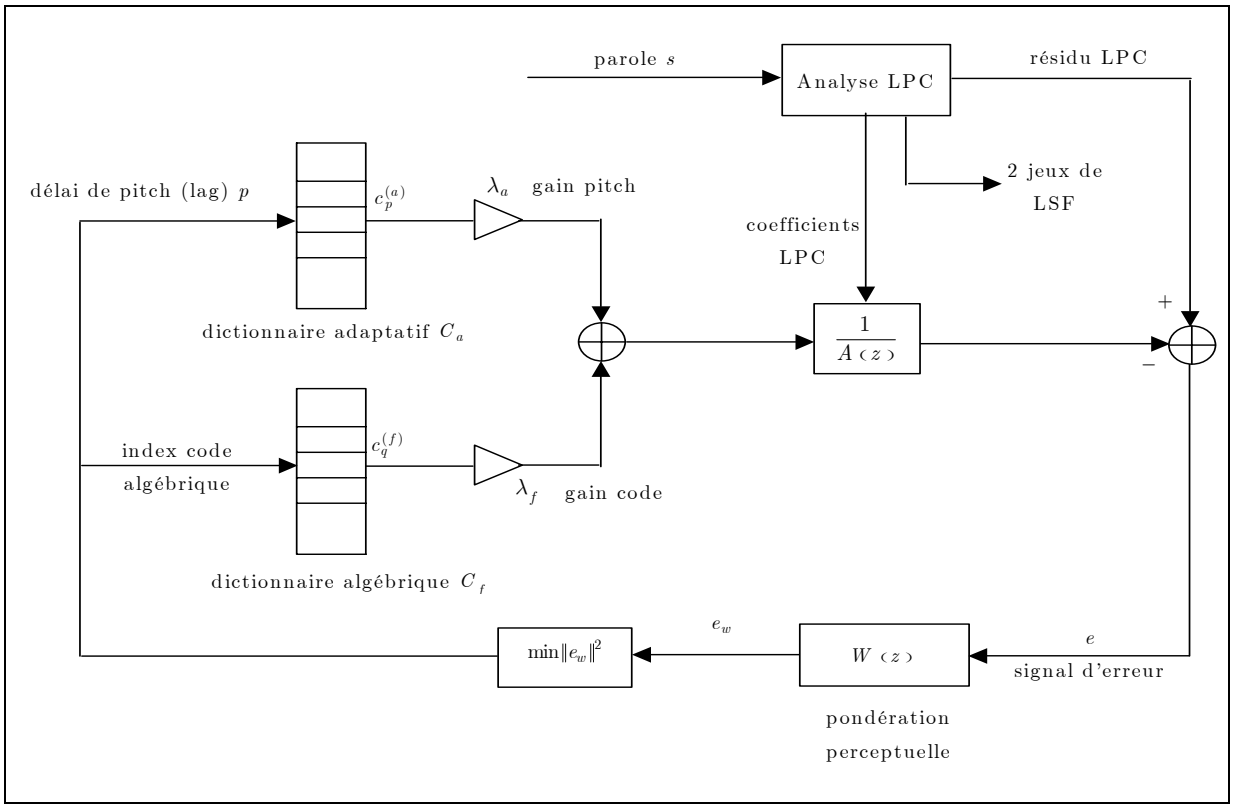

Figure A.20 : Schéma du codeur EFR (analyse par synthèse)

Finalement, le codeur CELP de l'EFR produit 244 bits par trame de 20ms (cf. tableau d'allocation des bits), ce qui correspond à un débit de 12,2 kbit/s.

Paramètres	1ère et 3ème sous trame	2ème et 4ème sous trame	Total par trame
LSP	7 ; 8 ; 9 ; 8 ; 6		38
délai de pitch	9	6	30
Gain pitch	4	4	16
Dictionnaire fixe	35	35	140
Gain code	5	5	20
total			244

Tableau A.2 : Allocation des bits par trame de 20 ms

Annexe B

Le codage canal dans le système GSM

B.1 Principes et stratégies de codage canal

L'objet du codage canal [Proakis, 1989] est de mettre en forme le message binaire à transmettre dans le canal de manière à pouvoir détecter et éventuellement corriger les erreurs introduites lors de sa transmission. Le principe est d'introduire une redondance dans le message issu du codeur source de manière à ce que certaines *configurations* d'éléments binaires (séquences) soient *impossibles* en sortie du codeur canal (diminution d'entropie). Une erreur de transmission sera détectée si une séquence interdite est reçue en sortie du canal. Cette erreur pourra éventuellement être corrigée en recherchant la séquence admissible la plus proche de la séquence reçue.

Plus précisément, considérons un message d'information constitué par une suite d'éléments binaires mutuellement indépendants et prenant leur valeur de manière équiprobable dans l'alphabet {0,1} noté $\mathbf{F}_2$. Toutes les séquences de K éléments binaires ont la même probabilité aussi la détection d'erreurs est impossible sur ce message transmis tel quel. Le codeur canal va produire pour chaque bloc $\mathbf{m}$ de K *éléments binaires d'information* en entrée, un bloc de sortie $\mathbf{x}$ de N *éléments binaires codés* avec $N > K$ pour satisfaire à la condition de redondance. Le rapport $R = K/N$ est appelé le *rendement* du code ou taux de codage. La sortie du codeur est obtenue par combinaisons linéaires d'éléments binaires du message d'information.

On rappelle ci-dessous les opérations d'additions et de multiplication dans $\mathbf{F}_2$:

a	b	a+b	ab
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table B.1 : Opérations dans le corps $\mathbf{F}_2$

On distingue deux grands types de codage canal :

- Les *codes en bloc linéaires* définis par une application linéaire g de la forme :

$$\begin{aligned} g : \mathbf{F}_2^K &\rightarrow \mathbf{F}_2^N \\ \mathbf{m} &\rightarrow \mathbf{x} = g(\mathbf{m}) \end{aligned} \tag{B.1}$$

Les blocs $\mathbf{x}$ (appelés dans ce cas *mots de code*) forment un sous-espace vectoriel $E_{N,K}$ de dimension K dans l'ensemble $\mathbf{F}_2^N$ des N-uplets binaires (g doit être de rang plein). Une erreur est détectable dès lors que le mot $\mathbf{r}$ reçu en sortie du canal n'appartient pas au sous-espace vectoriel $E_{N,K}$. Cette erreur pourra être corrigée en prenant le projeté orthogonal de $\mathbf{r}$ sur le sous-espace $E_{N,K}$ (minimisation de la distance de Hamming entre $\mathbf{r}$ et le mot de code estimé $\hat{\mathbf{x}}$).

- Les *codes convolutifs* (ou récurrents) définis par une application linéaire f de la forme :

$$\begin{aligned} h : \mathbf{F}_2^{K.D} &\rightarrow \mathbf{F}_2^N \\ [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_D] &\rightarrow \mathbf{x} = f([\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_D]) \end{aligned} \tag{B.2}$$

Un bloc de sortie $\mathbf{x}$ (appelé dans ce cas *symbole canal*) est produit en considérant non seulement le bloc $\mathbf{m}_1$ présent en entrée du codeur mais également les $(D-1)$ blocs précédemment entrés. Il y a donc un effet mémoire qui engendre une corrélation de forme connue entre les symboles (blocs) $\mathbf{x}$ émis par le codeur canal. Autrement dit, certaines séquences de symboles $\mathbf{x}$ sont interdites ce qui permettra la détection et la correction d'erreurs.

Le choix d'un procédé de codage dépend du type de canal et de la stratégie de protection contre les erreurs. Deux stratégies complémentaires sont généralement utilisées :

- Détection des erreurs

On veut détecter la présence d'erreurs avec la plus grande fiabilité possible. Cela s'applique notamment à des données sensibles. En cas d'erreur, le récepteur peut éventuellement demander à l'émetteur une retransmission de l'information (*Automatic Repeat Request*). On utilise ici essentiellement des codes en

blocs car leur performance de détection est peu sensible à la statistique des erreurs engendrées par le canal.

- Correction des erreurs (*Forward Error Control*)

On doit ici réaliser un compromis entre le *rendement* R et le *taux d'erreurs binaires* (*TEB*) résiduelles en sortie du décodeur canal. Pour une qualité de transmission donnée (rapport signal à bruit fixé), le *TEB* décroît lorsque le rendement R diminue, autrement dit lorsque le débit binaire augmente en sortie du codeur canal. Les codes convolutifs sont bien adaptés si on accepte un rendement de codage pas trop élevé.

Dans la norme GSM, ces deux stratégies sont utilisées :

- La détection des erreurs utilise un type particulier de codes en blocs linéaires, les codes cycliques. La procédure d'*ARQ* n'est pas applicable en raison du retard qu'elle implique mais elle est remplacée par une procédure de *substitution de trame* au niveau du décodeur.

- La correction d'erreurs est effectuée par un codeur convolutif.

Nous présentons maintenant plus en détail ces deux types de codages.

B.2 Les codes cycliques

Les codes cycliques représentent la classe la plus importante des codes en bloc linéaires. Ils sont utilisées de manière quasi-universelle dans les réseaux pour détecter des erreurs de transmission. Nous rappelons d'abord quelques définitions concernant les codes en blocs.

B.2.1 Capacité de détection des codes en blocs linéaires

On a vu que les 2^K mots de codes d'un code en bloc linéaire $E_{N,K}$ constituent un sous-espace vectoriel de dimension K dans $\mathbf{F}_2^N$. Les paramètres essentiels du code $E_{N,K}$ sont son *rendement* $R = K/N$ et sa *distance minimale* qui est la distance de Hamming minimale entre deux mots de codes distincts $\mathbf{x}_i$ et $\mathbf{x}_j$:

$$d_{\min} = \min_{i \neq j} \left\{ d_H \left(\mathbf{x}_i, \mathbf{x}_j \right) \right\} \tag{B.3}$$

On remarque que :

$$d_{\min} = \min_{\mathbf{x} \neq \mathbf{0}} \left\{ p_H (\mathbf{x}) \right\} \tag{B.4}$$

où $p_H(\mathbf{x})$ est le *poids* du mot de code $\mathbf{x}$ (nombre d'éléments non-nuls dans $\mathbf{x}$) et $\mathbf{0}$ désigne le mot de code nul.

La distance minimale permet d'évaluer les capacités de détection/correction du code. En effet, soit $\mathbf{r}$ le mot reçu en sortie du canal :

$$\mathbf{r} = \mathbf{x} + \mathbf{e} \tag{B.5}$$

où $\mathbf{x}$ est le mot de code émis et $\mathbf{e}$ représente les éventuelles erreurs de transmission.

On a une configuration d'erreur indétectable lorsque $\mathbf{e}$ coïncide avec un mot du code mais un tel cas ne peut se présenter que si $p_H(\mathbf{e}) \geq d_{\min}$. Le code $E_{N,K}$ peut donc *détecter* toutes les configurations de *M erreurs* dans un bloc de N éléments binaires avec :

$$M = d_{\min} - 1 \tag{B.6}$$

Intuitivement, on comprend que la distance minimale $d_{\min}$ augmente avec le nombre d'éléments redondants $(N-K)$ ajoutés par le codeur canal, on peut montrer que la probabilité P_{nd} de non-détection d'erreurs admet, pour un code en blocs, la borne suivante :

$$P_{nd} \leq 2^{-(N-K)} \tag{B.7}$$

B.2.2 Représentation polynomiale des codes cycliques

Un code en blocs linéaires est *cyclique* si toute permutation circulaire à gauche d'un mot de code $\mathbf{x} = [x_1,....,x_N]$ est aussi un mot de code. Pour les codes cycliques, on utilise généralement un formalisme polynomial plutôt que le formalisme vectoriel.

Dans la représentation polynomiale, chaque bloc de K éléments binaires d'information $\mathbf{m} = [m_1, m_1,...,m_K]$ est représenté par le polynôme de degré $(K-1)$ en t :

$$m(t) = m_1 + m_2 t + ... + m_K t^{K-1} \tag{B.8}$$

Un code cyclique $E_{N,K}$ est alors entièrement généré par un *polynôme générateur* de degré $(N-K)$ et de la forme générale :

$$g(t) = g_0 + g_1 t + ... + g_{N-K-1} t^{N-K-1} + t^{N-K} \tag{B.9}$$

En effet, chaque mot $\mathbf{x} = [x_1,....,x_N]$ du code cyclique peut être obtenu à partir du produit d'un polynôme d'information $m(t)$ avec le polynôme $g(t)$ générateur du code selon :

$$x(t) = m(t)g(t) \tag{B.10}$$

où $x(t)$ est le polynôme de degré $(N-1)$ représentant le mot de code **x**.

Pour détecter une erreur de transmission, il suffit alors de vérifier si le polynôme $r(t)$ associé au mot **r** reçu en sortie du canal est un multiple du polynôme générateur $g(t)$.

B.2.3 Code cyclique sous forme systématique

On utilise souvent un codage dit *systématique* pour lequel les éléments binaires d'informations apparaissent explicitement dans les mots de code. Ainsi les mots d'un code cyclique sous forme systématique s'écrivent :

$$c(t) = m(t)t^{N-K} + v(t) \tag{B.11}$$

où $m(t)$ est le polynôme associé aux éléments binaires d'information à coder et $v(t)$, le polynôme de degré au plus égal à $(N-K-1)$ associé aux éléments binaires de redondance.

Le calcul du mot de code $x(t)$ est ici encore très simple. Comme les mots du code cyclique sont générés par le polynôme générateur (i.e. $x(t)$ est un multiple de $g(t)$), on a :

$$m(t)t^{N-K} = f(t)g(t) + v(t) \tag{B.12}$$

Les éléments binaires de redondances sont donc obtenus comme le reste $v(t)$ de la *division euclidienne* de $m(t)t^{N-K}$ par $g(t)$.

La détection d'erreur est inchangée, elle consiste à vérifier que le mot reçu est bien un mot du code, autrement dit qu'il est multiple de son polynôme générateur $g(t)$.

Le système GSM utilise des codes cycliques systématiques pour la détection d'erreurs (*Cyclic Redundant Check*). Ils indiquent au récepteur la présence d'erreurs non corrigées par le codeur convolutif.

B.3 Les codes convolutifs

La différence fondamentale entre les codes convolutifs et les codes en blocs est que les premiers exploitent la notion de temps. Ils utilisent *un effet mémoire* sur les blocs de K éléments binaires présentés à l'entrée du codeur pour produire chaque bloc de N éléments binaires en sortie. Ceci permet notamment le codage à flot continu en choisissant des blocs d'entrée de longueur unité $(K=1)$. De plus, pour un code convolutif, la correspondance entre le message d'information en entrée et les mots de code (ou *symboles canal*) émis est très structurée ce qui conduit à des techniques de décodage très différentes de celles utilisées pour les codes en blocs.

B.3.1 Principe du codage convolutif

Le principe d'un codeur convolutif est illustré par le schéma de la Figure B.1. Les éléments binaires d'information en entrée du codeur sont décalés, au sein d'un registre à décalage, par blocs de longueur K. Pour chaque bloc de K éléments binaires introduits dans le registre à décalage, le codeur produit en sortie un bloc (symbole canal) de N éléments binaires. Les N éléments binaires d'un symbole canal $\mathbf{x}$ sont obtenus par *combinaisons linéaires* des éléments binaires d'information du bloc $\mathbf{m}_1$ en entrée ainsi que des $(D-1)$ blocs précédemment mémorisés.

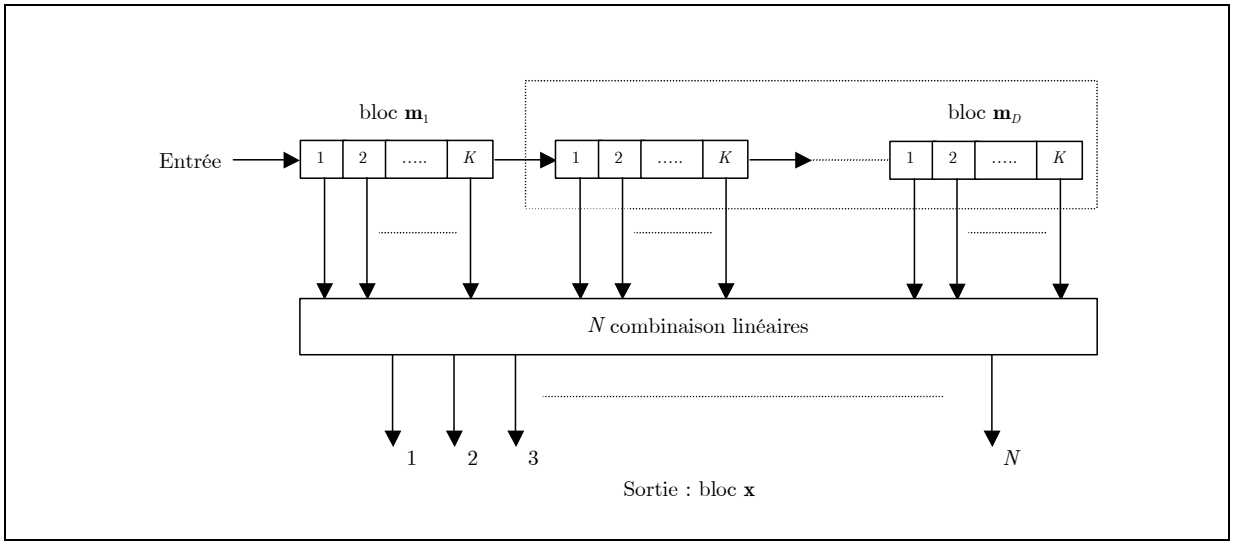

Figure B.1 : Principe d'un codeur convolutif

La structure d'un codeur convolutif est caractérisée par :

- Le *rendement du code*, défini comme pour les codes en blocs par $R = K/N$.
- La *longueur de contrainte* D qui correspond à la *durée de corrélation* entre un élément binaire d'information entré dans le codeur et la suite de symboles $\mathbf{x}$ émis en sortie.
- La mémoire *d'état du codeur* qui stocke les $\nu = K(D-1)$ éléments binaires d'information précédemment entrés.

Le codeur convolutif apparaît comme un automate à 2^{ν} états, la sortie $\mathbf{x}$ du codeur étant entièrement déterminée par la connaissance de l'état du codeur et du bloc d'information $\mathbf{m}_1$ présent en entrée. En utilisant la représentation vectorielle des blocs $\mathbf{m}_l = [m_1^l, m_1^l, ..., m_K^l]$ et $\mathbf{x} = [x_1, ..., x_N]$, on a :

$$\mathbf{x}^T = \mathbf{G}[\mathbf{m}_1, ..., \mathbf{m}_D]^T \tag{B.13}$$

où T est l'opérateur de transposition et $\mathbf{G}$ une matrice à KD colonnes et N lignes ayant pour éléments $g_{i,j} \in \mathbf{F}_2$.

Si on fait apparaître explicitement la suite temporelle d_k d'éléments binaires en entrée, on a :

$$\left[\mathbf{m}_1,...,\mathbf{m}_D\right] = \left[d_k, d_{k-1},...,d_{(k-l)K},...,d_{(k-D)K}\right] \tag{B.14}$$

On voit donc que la relation (B.13) correspond à une opération de convolution vectorielle entre la séquence d_k en entrée du codeur et les N vecteurs lignes $\mathbf{g}_i$ de $\mathbf{G}$. Les vecteurs $\mathbf{g}_i$ sont appelés séquences génératrices du code.

B.3.2 Représentation d'un code convolutif

Pour modéliser le fonctionnement d'un codeur convolutif, on utilise des représentations graphiques sous forme *d'arbre*, de *treillis* ou encore de *diagramme d'états*. Chacune de ces représentations est elle-même à la base d'algorithmes de décodage spécifiques. Ainsi la représentation sous forme d'arbre conduit à des algorithmes de décodage séquentiel comme l'algorithme de Fano. La représentation en treillis est associée à l'algorithme de Viterbi [Forney, 1973] alors que l'algorithme de Bahl est plutôt basé sur le diagramme d'états [Bahl et al., 1974].

B.3.2.a Représentation en arbre

La représentation en arbre repose sur une description des *suites de symboles possibles* $\mathbf{x}_k = \left[x_{k,1},...,x_{k,N}\right]$ en sortie du codeur. A partir d'une racine (qui correspond à un état initial du codeur, choisi nul en général), l'arbre se divise en deux branches à chaque nouvel élément binaire entré dans le registre du codeur. Une branche est associée à l'entrée d'un 0 et l'autre à celle d'un 1. On obtient ainsi une arborescence.

Nous ne développerons pas plus cette représentation car nous ne l'utiliserons pas par la suite.

B.3.2.b Représentation en diagramme d'états

Le diagramme d'états, illustré Figure B.2, ne fait pas apparaître explicitement le temps mais seulement les transitions possibles entre états du codeur (vu comme un automate à 2^{ν} états). Les branches en traits pointillés correspondent à une transition déclenchée par l'entrée d'un élément binaire d'information égal à 0. Les branches en traits pleins correspondent à un élément binaire en entrée égal à 1. A chaque transition, on a associé le symbole canal $\mathbf{x}$ émis en sortie du codeur.

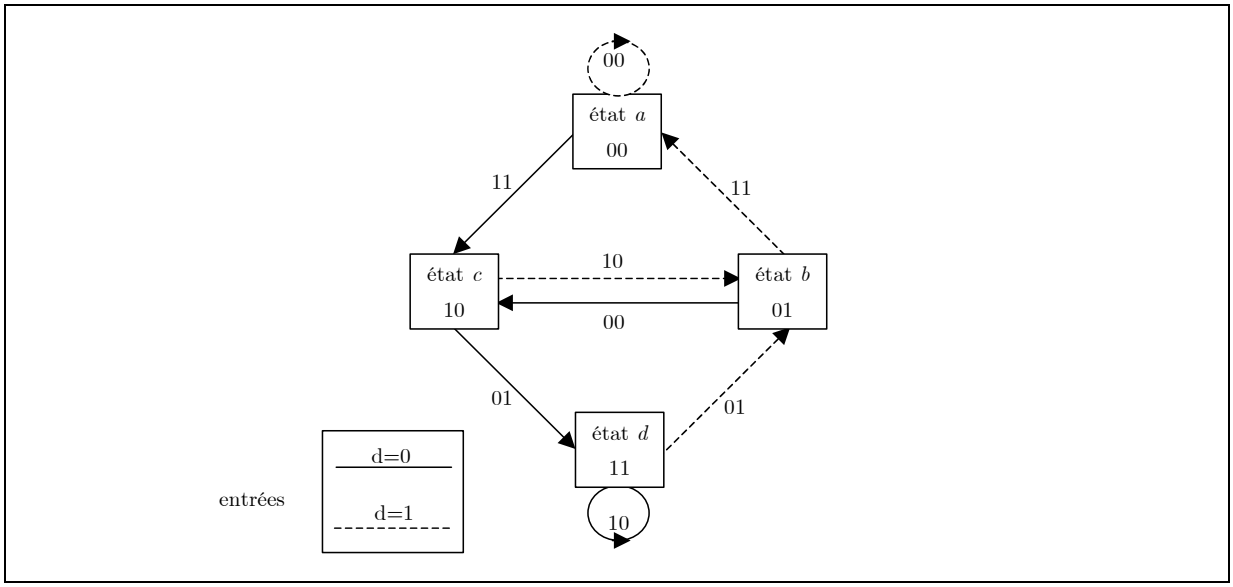

Figure A.2 : Diagramme d'état d'un codeur convolutif (*K*=1,*N*=2)

Pour un code convolutif de rendement K/N, 2^K branches partent de chaque état et 2^K branches aboutissent à un même état.

A.3.2.c Représentation en treillis

Le diagramme en treillis représente *les suites d'états possibles* au cours du temps (et non de symboles comme le diagramme en arbre). On peut le voir comme le déroulement au cours du temps du diagramme d'états. Pour illustrer le principe du treillis, on a représenté sur la Figure A.3, le treillis du codeur convolutif (7,5) = [111] [101].

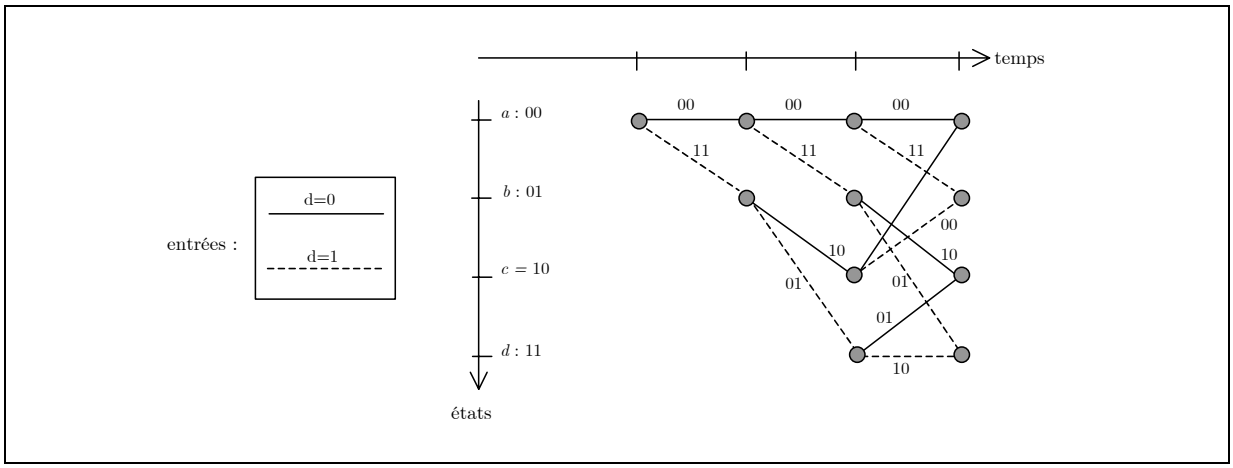

Figure A.3 : Diagramme en treillis

Après D décalages, quel que soit l'état initial du codeur, le motif du treillis se répète. En effet, la *longueur de contrainte* D correspond à la durée nécessaire pour renouveler entièrement le contenu du

registre du codeur, c'est donc la longueur minimale au bout de laquelle deux chemins issus d'un même nœud (état) peuvent reboucler.

B.3.3 Capacité de correction d'un code convolutif

Pour les codes en blocs, les performances de détection/correction d'erreurs étaient mesurées à partir de la distance minimale entre deux mots de code. Cette distance dépendant essentiellement du rendement.

Conditionnellement à un état donné, un codeur convolutif peut être vu comme un code en bloc dont les mots de code (symboles canal) appartiennent à un sous-espace de dimension K dans $\mathbf{F}_2^N$. Ceci constitue une première contrainte (fonction du rendement $R = K / N$) indispensable à la détection/correction des erreurs. Mais le codeur convolutif rajoute une contrainte sur les transitions entre états qui se traduit par la *longueur de contrainte* D, longueur minimale pour que deux chemins qui ont divergé convergent de nouveau. Le pouvoir de correction d'un codeur convolutif pourra donc être mesuré par *la plus petite distance de Hamming qui existe entre deux chemins qui divergent puis convergent de nouveau*[94], c'est la *distance libre* du code convolutif. Intuitivement, on voit que la distance libre d_{libre} dépendra du rendement R et de la longueur de contrainte D.

D'après la définition de la longueur de contrainte D, l'influence d'un élément binaire d'information ne perdure que sur D symboles canal, autrement dit sur DN éléments binaires émis en sortie du codeur. Aussi, les capacités de correction d'un code convolutif sont limitées, en particulier en ce qui concerne les erreurs groupées. *Un code convolutif ne peut corriger les paquets d'erreurs de longueur supérieure à* $(D-1)N$.

Comme on le verra par la suite, on a $D = 5$ et $N = 2$ pour le codeur convolutif mis en œuvre dans le système GSM. Celui-ci ne pourra donc corriger des séquences d'erreurs de longueur supérieure à 8 bits. Pour éviter cet inconvénient, on utilise la technique de *l'entrelacement* présentée en Annexe C. Sous certaines conditions, cette technique permet de transformer un canal à paquets d'erreurs (comme le canal radiomobile) en canal à erreurs indépendantes.

B.3.4 Décodage du code convolutif

De manière générale, on peut concevoir le décodage canal selon deux stratégies distinctes.

- Le critère d'optimalité porte sur les mots de code du codeur canal :

  On estime les *mots de code émis* $\mathbf{x}_k$ à partir des mots $\mathbf{y}_k$ reçus et on en déduit la séquence de bits d'information $\mathbf{u}_k$. Dans le cas des codes en blocs, cette estimation s'effectue mot par mot

[94] Poids minimal d'une erreur non-détectable de manière analogue à $d_{\min}$ pour les codes en blocs.

(mots de codes indépendants). Pour un codeur convolutif, les mots de code $\mathbf{x}_k$ sont corrélés et seules certaines séquences de mots sont possibles en sortie du décodeur, elles correspondent aux chemins autorisés dans le diagramme en treillis du code. Dans ce cas, l'estimation du mot de code $\mathbf{x}_k$ émis à l'étape k ne peut se faire indépendamment de l'estimation des mots précédents $\{\mathbf{x}_0,...,\mathbf{x}_{k-1}\}$. Pour décoder une séquence de mots $\mathbf{y}_k$, il est nécessaire de considérer la séquence reçue dans son ensemble, c'est la démarche de l'algorithme de Viterbi [Forney, 1973] qui choisit le meilleur chemin dans le treillis (selon le critère d'optimalité du Maximum a Posteriori). On parle alors de *minimisation de la probabilité d'erreur par séquence.*

- Le critère d'optimalité porte sur les bits d'informations décodés :

  On cherche ici à *minimiser la probabilité d'erreur sur chaque bit d'information* décodés (plutôt que de rechercher la meilleure *séquence* de bits décodés). Ceci nécessite d'intégrer tous les chemins du treillis décodant une même valeur du bit d'information considéré. Cet algorithme, beaucoup plus complexe a été proposé par [Bahl et al., 1974].

On se place dans ce qui suit selon la première approche (*minimisation de la probabilité d'erreur par séquence*). Avant de présenter l'algorithme de Viterbi, il convient de définir la nature exacte des informations reçues du canal.

B.3.4.a Canal à sorties souples

Le décodage canal est très souvent présenté dans le cadre d'un canal additif à bruit gaussien (*CABG*). On adopte ici un formalisme différent permettant de donner une interprétation en termes de probabilité d'erreur de la sortie souple d'un canal (cf. Annexes C et D). Ainsi, on modélise le *canal équivalent* formé par l'ensemble (entrelacement - canal radio - égalisation – dé-entrelacement) comme un **canal binaire symétrique (CBS) sans mémoire** et de probabilité d'erreur *instantanée* connue. Ceci est illustré par le schéma de transmission Figure B.4.

On notera qu'on peut définir un tel canal à partir d'un canal additif à bruit gaussien (*CABG*) en considérant les décisions *fermes* en sortie du canal. Dans ce cas, la probabilité d'erreur varie au cours du temps et doit être estimée pour chaque bit, c'est pourquoi on parle de probabilité d'erreur *instantanée*. Nous présentons en Annexe C une implémentation de l'égaliseur permettant d'estimer une probabilité d'erreur pour chaque bit.

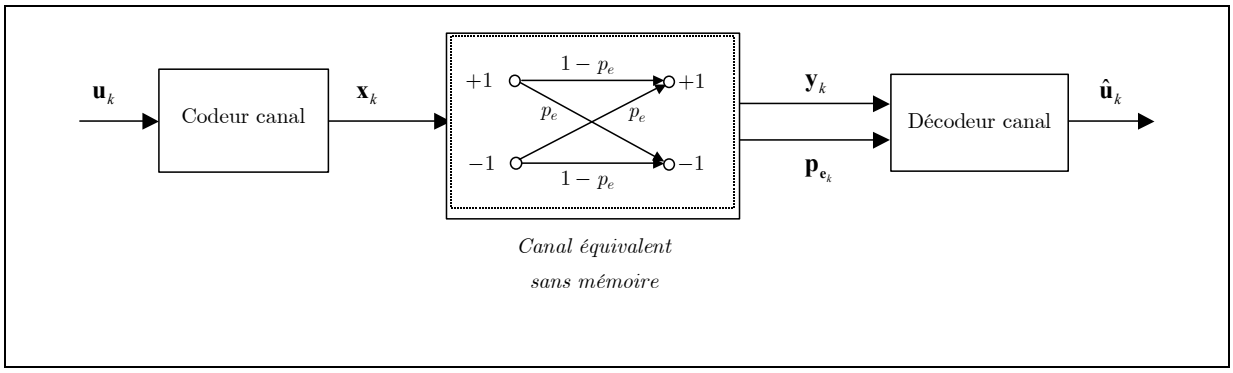

Figure B.4 : Modèle de canal utilisé pour le décodage canal

La connaissance des probabilités d'erreur $\mathbf{p}_{\mathbf{e}_k} = \left[p_{e_{k,1}},...,p_{e_{k,r}},...,p_{e_{k,N}} \right]$ associées aux bits du mot de code $\mathbf{y}_k = \left[y_{k,1},...,y_{k,r},...,y_{k,N} \right]$ reçu à l'étape k définit la *sortie souple* du canal équivalent. Plus exactement, on définit la sortie souple $L\left(x_{k,r}\right)$ du canal équivalent d'entrée $x_{k,r}$, de la façon suivante :

$$\begin{aligned} L\left(x_{k,r}\right) &= \log \frac{p\left(x_{k,r} = +1\right)}{p\left(x_{k,r} = -1\right)} \\ &= y_{k,r} \log \frac{1 - p_{e_{k,r}}}{p_{e_{k,r}}} = y_{k,r} L_{c_{k,r}} \end{aligned} \tag{B.15}$$

Le signe de la sortie souple correspond à la décision ferme $y_{k,r}$ en sortie du canal équivalent binaire symétrique alors que la valeur absolue de la sortie souple mesure la confiance $L_{c_{k,r}}$ dans cette décision ferme.

B.3.4.b Principe du décodage par séquences

Le décodeur cherche la séquence de symboles canal $\mathbf{X} = \left\{ \mathbf{x}_1,...,\mathbf{x}_k,...,\mathbf{x}_L \right\}$ (ou de manière équivalente, le chemin ℓ dans le treillis) maximisant la probabilité *a posteriori* (conditionnellement à la séquence reçue $\mathbf{Y} = \left\{ \mathbf{y}_1,...,\mathbf{y}_k,...,\mathbf{y}_L \right\}$) :

$$\hat{\mathbf{X}} = \underset{\mathbf{X}_{\text{admissibles}}}{\arg\max}\, p(\mathbf{X}|\mathbf{Y}) = \underset{\mathbf{X}^{(\ell)}}{\arg\max}\, p\left(\mathbf{X}^{(\ell)}|\mathbf{Y}\right) \tag{B.16}$$

où l'on a étiqueté la séquence de symboles émise par le chemin ℓ correspondant dans le treillis. On notera $\mathbf{x}_k = \left(x_{k,1},...,x_{k,N}\right)$ et $\mathbf{y}_k$, les symboles canal émis (resp. reçus) à l'étape k du treillis. On a :

$$p\left(\mathbf{X}^{(\ell)}|\mathbf{Y}\right) = p\left(\mathbf{Y}\middle|\mathbf{X}^{(\ell)}\right) \frac{p\left(\mathbf{X}^{(\ell)}\right)}{p(\mathbf{Y})} \tag{B.17}$$

On a fait l'hypothèse que *les symboles d'informations étaient indépendants et uniformément distribués*, il en résulte que les chemins du treillis sont équiprobaux. Comme la probabilité $p(\mathbf{Y})$ est une

constante relativement au chemin ℓ, le critère (B.16) du Maximum a Posteriori (MAP) est équivalent au Maximum de Vraisemblance (MV) :

$$\arg\max_{\ell} p\big(\mathbf{X}^{(\ell)}|\mathbf{Y}\big) = \arg\max_{\ell} p\big(\mathbf{Y}\big|\mathbf{X}^{(\ell)}\big) \tag{B.18}$$

De plus, on a supposé un canal *sans mémoire*, donc:

$$\log\big\{p\big(\mathbf{Y}|\mathbf{X}^{(\ell)}\big)\big\} = \sum_{k} \log\big\{p\big(\mathbf{y}_k\big|\mathbf{x}_k^{(\ell)}\big)\big\}$$

Le critère (B.18) revient ainsi à maximiser la métrique *récursive* :

$$M_k^{(\ell)} = M_{k-1}^{(\ell)} + \log\big\{p\big(\mathbf{y}_k\big|\mathbf{x}_k^{(\ell)}\big)\big\} \tag{B.19}$$

L'incrément de métrique $\Delta M_k^{(\ell)} = \log\big\{p\big(\mathbf{y}_k\big|\mathbf{x}_k^{(\ell)}\big)\big\}$ ne dépend que de la branche du treillis et des données reçues à l'étape k du treillis, on l'appelle *métrique de branche*. On peut l'exprimer (cf. Chapitre 6) en fonction de la *sortie souple* $L_{c_{k,r}} y_{k,r}$ du canal selon :

$$\Delta M_k^{(\ell)} = \log\big\{p\big(\mathbf{y}_k\big|\mathbf{x}_k^{(\ell)}\big)\big\} = \sum_{n=1}^{N} x_{k,n}^{(\ell)} L_{c_{k,n}} y_{k,n} \tag{B.20}$$

On a ainsi formulé le problème d'estimation en un problème de recherche du chemin de meilleur coût dans un treillis. Un tel problème est adressé par l'algorithme de Viterbi.

B.3.4.c Algorithme de Viterbi

L'algorithme de Viterbi [Forney, 1973] dérive du principe de la *programmation dynamique* et permet de déterminer le chemin de métrique maximale dans le treillis. Cet algorithme, divisé en quatre étapes, a la forme suivante :

A chaque symbole $\mathbf{y}_k$ reçu à l'étape k :

- *Calcul des métriques de branches* : $\Delta M_k^{(\ell)}$
- *Cumul des métriques* : pour chaque chemin partiel m : $M_k^{(\ell)} = M_{k-1}^{(\ell)} + \Delta M_k^{(\ell)}$
- *Choix du chemin survivant à l'étape k* : En chaque nœud, il converge 2^K chemins. Le chemin le plus vraisemblable jusqu'à un nœud correspond au chemin de métrique la plus forte. On conserve donc en ce nœud la valeur de la métrique la plus forte ainsi que l'indice du nœud dont est issue à la branche du treillis choisie.
- On passe à l'étape suivante $k+1$.

Jusque là, on a donc besoin de conserver la métrique pour chaque nœud, ce qui correspond à un vecteur de longueur 2^{ν}, ainsi que de l'indice du nœud précédent chaque nœud à l'étape k, ce qui correspond à une matrice de 2^{ν} lignes et L colonnes. Lorsqu'on arrive au bout du treillis (dernier symbole reçu), il suffit de remonter le long du chemin de métrique la plus élevée, dans le sens inverse, opération aussi appelée «*trace back*» :

- *Traceback* : Parmi les distances cumulées calculées à la réception du dernier symbole, celle de plus forte valeur correspond au chemin de vraisemblance maximale. Le nœud précédent correspondant a été mis en mémoire, ainsi que le nœud précédent ce nœud et ainsi de suite. On retrouve le chemin parcouru dans le treillis et de ce fait la séquence émise la plus vraisemblable.

B.4 La protection aux erreurs de transmission du GSM FR et EFR

Contrairement au codage source, le codage de canal n'a pas évolué ente le GSM EFR et le GSM FR [GSM, 05.03]. Pour garder le même débit en entrée du codeur de canal (trame de 260 bits), on a donc rajouté un « codage de canal préliminaire », constitué de 16 bits de redondance, répartis en 8 bits résultant d'un codage CRC et 8 bits de répétition. Les 260 bits de la trame codée sont ensuite répartis en trois classes selon leur *impact sur la qualité* de la parole [Scalart, 1997] :

- Classe I.a : 50 bits, très sensibles aux erreurs, ils ne doivent pas être mal interprétés.
- Classe I.b : 132 bits, sensibles aux erreurs.
- Classe II : 78 bits, les moins sensibles aux erreurs.

Les 50 bits de la classe I.a sont protégés par un code cyclique CRC de 3 bits. Si, à la réception, une erreur est détectée sur cette partie de la trame, la trame complète est rejetée (*perte de trame)* et une technique de masquage est utilisée (cf. Chapitre 1).

Les 53 bits résultants de l'application du CRC aux éléments de la classe I.a sont groupés avec les 132 bits de classe I.b pour former un bloc de 185 bits auquel est appliqué un code convolutif de *rendement R=1/2* et de *mémoire* $\nu = 4$. Il en résulte un bloc *protégé* de taille 2*(185+4) soit 378 bits.

En ajoutant à ces 378 bits protégés, les 78 bits de la classe II, *non protégés*, on obtient le bloc de 456 bits produit en sortie du codeur de canal du GSM. Ces opérations sont schématisées par la Figure B.5.

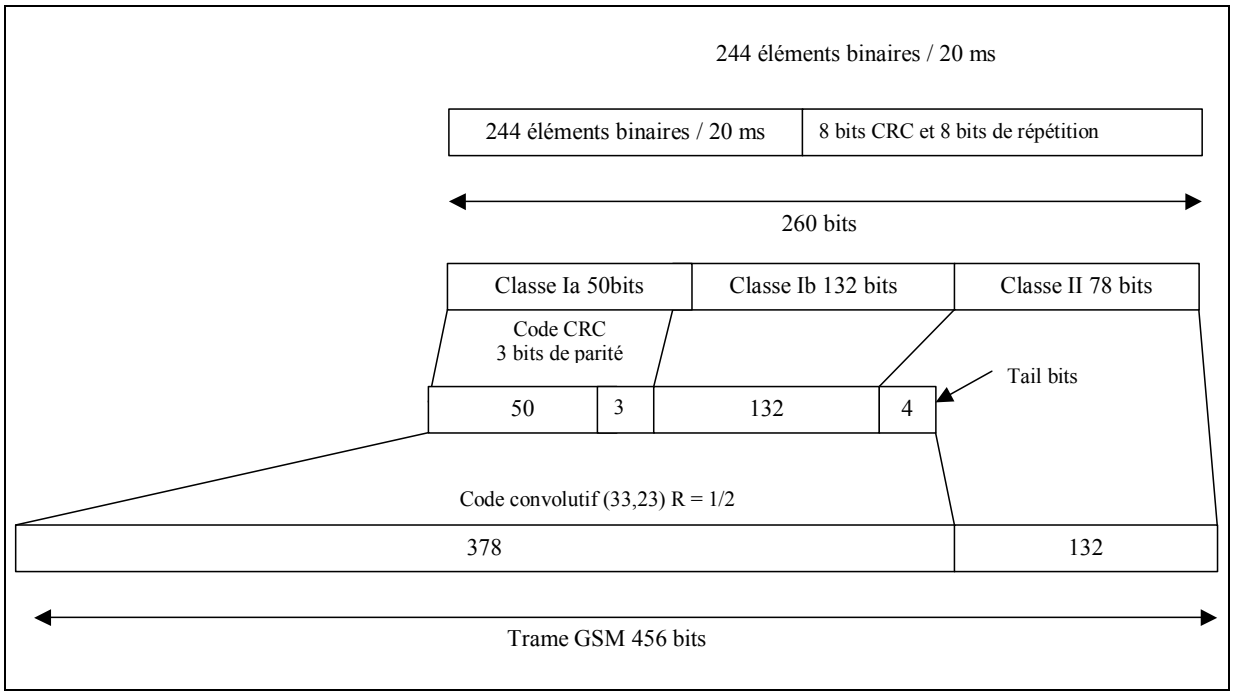

Figure B.5 : Répartition des bits après codage de canal

Annexe C

Simulation du canal de transmission

Le canal de transmission, tel que nous le considérons ici, englobe les fonctionnalités d'émission et de réception de part et d'autre du canal physique de transmission proprement dit (canal radio). Nous présentons ici ces éléments et principalement le récepteur. Ce dernier est appelé *récepteur interne* par opposition au *récepteur externe* formé par les décodeurs de canal et de source (cf. Chapitre 1). Son rôle est de faire apparaître un canal idéal du point de vue du décodeur externe, c'est-à-dire un canal sans mémoire et de probabilités de transitions connues à chaque instant n comme illustré Figure C.1.

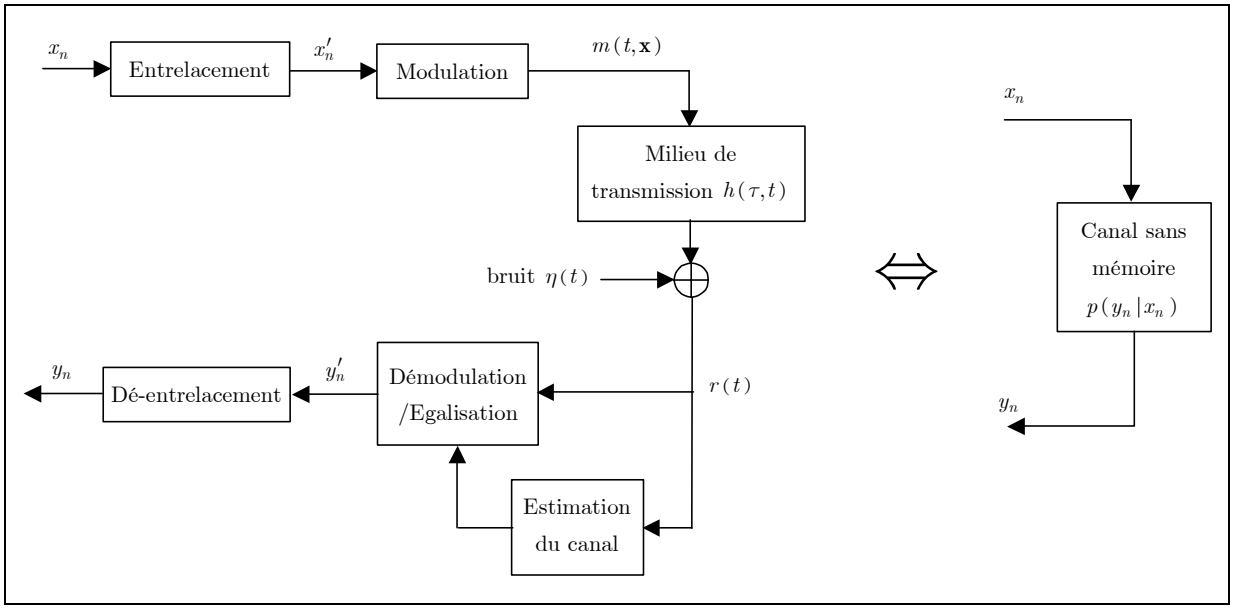

Figure C.1 : Canal équivalent associé au récepteur interne

C.1 Caractéristiques du milieu de transmission

Le canal de propagation radiomobile introduit de nombreuses perturbations sur le signal transmis. En effet, la transmission radiofréquence entre un mobile et une station de base est perturbée par la présence d'obstacles tel que le relief naturel ou bien encore les bâtiments. Une partie des ondes réfléchies ou diffractées par ces obstacles sera captée par le récepteur, on parle alors de transmission par *trajets multiples*. Cette caractéristique introduit deux effets majeurs :

- les évanouissements sélectifs (*fading*) qui correspondent à des interférences destructives entre trajets.
- l'interférence entre symboles (*IES*) du à la dispersion temporelle des signaux provenant des différents trajets.

Ces effets sont de plus *non-stationnaires* puisque les trajets fluctuent avec le mouvement du mobile.

Une autre cause majeure de perturbations est liée au concept cellulaire. La réutilisation de fréquences entre cellules génère un *taux de brouillage* (rapport signal/interférences C/I) qui peut être élevé pour un réseau dense de cellules.

C.1.1 Sélectivité en fréquence

La réponse impulsionnelle *en bande de base* d'un canal multi-trajets s'écrit selon :

$$h(\tau,t) = \sum_{p=0}^{N-1} c_p(\tau)\delta\big(t - d_p(\tau)\big) \tag{C.1}$$

Ce modèle intègre la variabilité temporelle du canal liées aux variations (en τ) des caractéristiques des trajets (atténuation $c_p(\tau)$ et retard $d_p(\tau)$). Sa fonction de transfert est de la forme :

$$H(f,\tau) = \sum_{p=0}^{N-1} c_p(\tau)e^{-j2\pi f d_p(\tau)} \tag{C.2}$$

Le module de cette fonction de transfert présente des « trous » à certaines fréquences (évanouissement en fréquences). Autrement dit, certaines fréquences subissent une très forte atténuation du fait de l'interférence entre trajets, ceci caractérise la *sélectivité en fréquence* d'un canal multi-trajets.

On appelle *bande de cohérence* du canal, l'écart fréquentiel Δf_c entre deux évanouissements. Elle est liée au *temps de dispersion de groupe* σ_t de la réponse impulsionnelle par :

$$\Delta f_c \approx \frac{1}{2\pi\sigma_t} \tag{C.3}$$

Le canal de propagation sera *sélectif en fréquence* si la largeur de bande W du signal émis excède la bande de cohérence Δf_c, soit :

$$W \gg \Delta f_c \tag{C.4}$$

Dans le cas contraire, toutes les fréquences de la bande W subissent globalement la même atténuation ("*fading plat*"). Cette atténuation peut cependant être très forte (lorsque la bande W correspond à une zone d'évanouissement fréquentiel).

Dans le domaine temporel, la *sélectivité en fréquence* se traduit par *l'interférence entre symboles* (*IES*). En effet, puisque $W \propto 1/T_s$ où T_s est la durée d'un symbole, la relation (C.4) s'écrit :

$$\sigma_t \gg T_s \tag{C.5}$$

ce qui signifie que le retard "moyen" des trajets secondaires par rapport au trajet de référence est important comparé à la durée T_s d'un symbole. Pour les canaux radiomobiles, le temps de dispersion de groupe σ_t est de l'ordre de 5 à 10 μs en zone urbaine et de 0.7 μs en zone rurale.

C.1.2 Variabilité temporelle

On a pour l'instant considéré une réponse impulsionnelle (C.1) déterminée, autrement dit à un instant t fixé. En fait, les trajets varient en fonction du déplacement du mobile. Ceci se traduit par des modifications de la fonction de transfert, la position et la profondeur des évanouissements variant dans le temps.

L'évolution dans le temps du canal est directement liée à la rapidité du mobile. Ainsi, le déplacement du mobile va induire pour chaque trajet n, le décalage Doppler:

$$f_d = \frac{f_0 \|\vec{v}\|}{c} \cos \varphi_n \tag{C.6}$$

$\vec{v}$ étant la vitesse du mobile et φ_n, l'angle du trajet n par rapport au vecteur $\vec{v}$.

Plusieurs définitions du *temps de cohérence* Δt_c peuvent être rencontrées, ici on le définira selon :

$$\Delta t_c \approx \frac{1}{2 f_{d_{\max}}} \tag{C.7}$$

Le canal sera considéré comme introduisant des *évanouissements rapides* s'il varie sensiblement pendant la durée d'un symbole, soit :

$$T_s \gg \Delta t_c \tag{C.8}$$

Dans le cas radiomobile, on a $f_d \leq 250\ Hz$, soit $\Delta t_c \geq 2\ ms$.

En conclusion, on voit que la capacité C du canal radiomobile dépend de la durée symbole T_s choisie. En effet, pour un niveau d'interférences et de bruit donné, la capacité C va dépendre de l'amplitude du signal reçu ainsi que de l'interférence entre symboles *IES*. Pour *maximiser la capacité C*, on doit respecter la contrainte suivante :

$$\sigma_t \ll T_s \ll \Delta t_c \tag{C.9}$$

La durée symbole du GSM est $T_s = 3.69\mu s$. Le canal de propagation GSM est donc fortement sélectif en fréquence (en zone urbaine) mais présente peu d'évanouissements rapides.

C.1.3 Modèles de canaux

On a jusqu'ici défini la réponse impulsionnelle du canal de façon *déterministe*. Dans la réalité, les paramètres définissant la réponse (C.1) ne sont absolument pas maîtrisables, on doit donc utiliser des modèles de canaux. On présente ici les deux modèles utilisés dans la norme du GSM.

- Modèle de Rayleigh (TU) :

Ce modèle suppose un canal non sélectif (évanouissements plats) et sans évanouissements rapides. Le signal reçu en bande de base à un *instant t donné* peut être modélisé comme une *variable aléatoire* de la forme :

$$r(t) = R_a e^{j\theta} \tag{C.10}$$

où θ est distribuée uniformément entre 0 et 2π et l'amplitude R_a (réel positif) suit une loi de Rayleigh :

$$p_{R_a}(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \ ;\ x > 0 \tag{C.11}$$

Ce modèle correspond aux modèles **TU** du GSM (zones urbaines).

- Modèle de Rice (RU):

Ce modèle correspond au cas où un des trajets reçus présente une atténuation constante (trajet dominant). L'amplitude R_a du signal reçu $r(t)$ suit alors la loi :

$$p_{R_a}(x) = \frac{x}{\sigma^2} e^{-\frac{x^2+\beta^2}{2\sigma^2}} I_0\left(\beta \frac{x}{\sigma^2}\right) \ ;\ x > 0 \tag{C.12}$$

Ce modèle correspond au modèle **RU** du GSM (zones rurales).

C.2 Fonctionnalités de l'émetteur

La principale opération réalisée par l'émetteur, appelée modulation, consiste à associer au message numérique issu du codeur de canal, un signal analogique de forme adaptée au milieu de propagation utilisé (*fonction d'adaptation*). Parmi les autres traitements effectués par l'émetteur, on présentera le saut de fréquences qui permet de lutter contre les interférences (*diversité fréquentielle*). Enfin, bien qu'elle soit en pratique réalisée au niveau du codeur canal, on a intégré l'opération d'entrelacement à l'émetteur. En effet, l'entrelacement peut être vu comme une technique de *diversité temporelle* destiné à lutter contre les évanouissements.

C.2.1 Entrelacement (diversité temporelle)

Les évanouissements sélectifs du canal radiomobile introduisent, au niveau du récepteur, des erreurs de décodage se présentant le plus souvent sous la forme de "paquets" d'erreurs groupées. Or le codeur convolutif utilisé dans le GSM ne peut corriger une séquence d'erreurs de longueur supérieure à 8 éléments binaires (cf. Annexe B). L'entrelacement vise donc à *transformer un canal à paquets d'erreurs en un canal à erreurs indépendantes* en fragmentant ces paquets d'erreurs.

C.2.1.a Entrelacement bloc

Dans le système GSM, chaque séquence de 456 éléments binaires provenant du codeur canal est insérée dans une matrice de 57 lignes et 8 colonnes. Les éléments binaires sont rentrés ligne par ligne et ensuite lus colonne par colonne. Chaque colonne correspond à un sous-bloc (ou à un *demi-burst*).

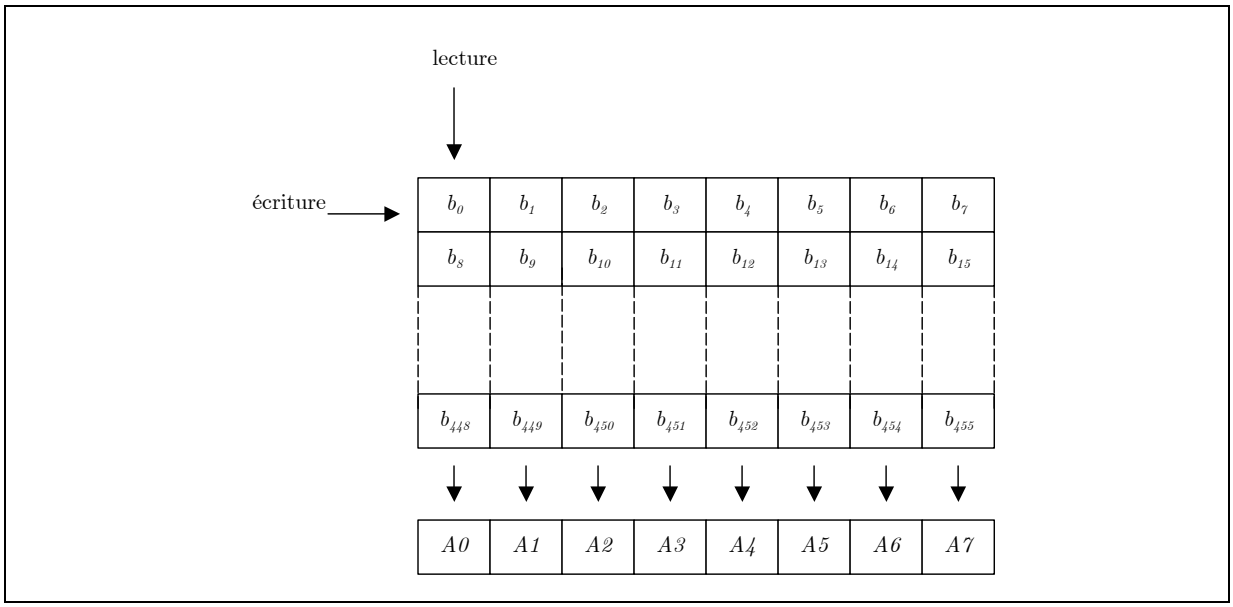

Figure C.2 : Entrelacement bloc

C.2.1.b Entrelacement convolutif

Suite à l'entrelacement bloc, la transmission des 8 sous-blocs obtenus est étalée dans le temps de manière à lutter contre les évanouissements sélectifs et les brouilleurs. Ainsi, chaque sous-bloc est associé avec un sous-bloc de la trame précédente (pour les sous-blocs A0 à A3) ou de la trame suivante (pour les sous-blocs A4 à A7) pour former un bloc de 114 éléments binaires (*burst*). On remarquera que cet entrelacement dit diagonal (ou convolutif) introduit *un retard d'une trame* dans la transmission.

Enfin, à l'intérieur d'un burst de 114 éléments, les éléments binaires provenant de la trame de parole la plus récente (sous-blocs A0 à A3) et de la trame précédente (sous-blocs A4 à A7) sont alternés.

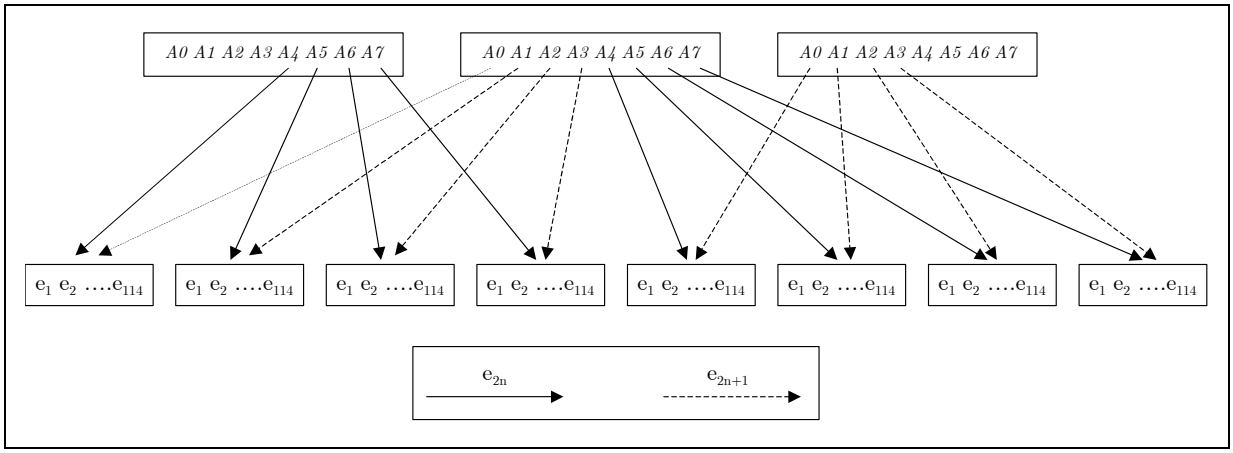

Figure C.3 : Entrelacement convolutif

C.2.2 Modulation

La modulation joue un rôle particulièrement important dans les performances d'un système de transmission. Deux critères principaux régissent le choix d'un type de modulation :

- La bande passante, déterminée par le spectre de puissance du signal modulé. Dans les applications radiomobiles, le premier objectif est de réduire *l'occupation spectrale*.

- Le taux d'erreurs binaires (TEB) en sortie du canal physique. Intuitivement, on perçoit que les erreurs introduites lors de la transmission seront d'autant plus faibles que la distance entre les états de modulations du signal propagé sera grande[95]. Pour la parole numérisée, des taux d'erreurs de l'ordre de 10^{-3} sont acceptables.

La modulation utilisée dans le GSM est la modulation *GMSK*. On peut la considérer comme une modulation *MSK* dont les transitions de phase seraient adoucies afin de diminuer l'occupation spectrale. En fait, il s'agit d'une modulation de fréquence, le signal modulé s'écrit :

$$m(t,\mathbf{x}) = \cos\left(2\pi f_0 t + \int_{\tau=0}^{t} v(\tau,\mathbf{x})\,d\tau\right) \tag{C.13}$$

le signal $v(t,\mathbf{x})$ étant le résultat d'un pré-filtrage de la séquence[96] à transmettre $\mathbf{x} = \left\{x\left(nT_b\right)\right\}_{n\in\mathbb{N}}$ par un filtre passe-bas gaussien $g(t)$ défini par son produit BT_s.

On se bornera ici à rappeler quelques caractéristiques essentielles de cette modulation :

- La modulation GMSK peut être linéarisée et ainsi être traitée de manière analogue à une modulation antipodale comme la *MDP*-2.

- La durée symbole T_s est ici égale à la durée d'un élément binaire codé T_b.

- La réponse du filtre s'étale sur 3 éléments binaires, le pré-filtrage introduit donc de l'interférence entre symboles (IES). En fait, l'occupation spectrale est d'autant plus faible que le produit BT_s diminue mais c'est au prix d'une dégradation du TEB engendrée par l'augmentation de l'IES. Ainsi, l'utilisation d'un égaliseur en réception apparaît indispensable même lorsque le canal radiomobile est peu sélectif en fréquence (zones rurales).

---

[95] On perçoit également que le TEB et l'occupation spectrale sont des critères antinomiques puisqu'il est nécessaire d'avoir des transitions les plus douces possibles entre les états de modulations si l'on souhaite minimiser l'occupation spectrale.

[96] T_b .dénote ici la durée d'un élément binaire alors que T_s est celle d'un *symbole de modulation*.

C.3 Fonctionnalités du récepteur

Le récepteur *interne* est la partie qui nous concerne le plus dans cette présentation car son rôle est de reproduire un *canal idéal* vis à vis du récepteur externe (décodeurs canal et source). Il se compose essentiellement de deux fonctions, une fonction d'*estimation* de la réponse du canal et une fonction *d'ajustement* au canal (synchronisation, égalisation). La présentation succincte qui est faite ici vise surtout à fournir une interprétation de la sortie renvoyée par le récepteur interne.

C.3.1 Estimation de la réponse impulsionnelle du canal

Afin d'estimer la réponse impulsionnelle du canal, on insère dans les bursts émis une séquence d'apprentissage connue du récepteur. Une propriété très intéressante pour une séquence d'apprentissage est d'avoir une fonction d'auto-corrélation proche du Dirac $\delta(n)$. Ce type de séquence permet alors de synchroniser finement chaque burst et de sonder le canal de propagation radiomobile par filtrage adapté. En effet, la fonction d'inter-corrélation $\Gamma_{x',r}(n)$ évaluée entre le signal émis x'_n en *entrée du modulateur* et le signal reçu *en sortie du filtre de réception* $r(nT_s)$ et échantillonné à la période T_s est donnée par :

$$\Gamma_{x',r}(n) = \Gamma_{x',x'}(n) \otimes h(nT_s) \tag{C.14}$$

où $\otimes$ désigne l'opération de convolution et $h(nT_s)$ est la réponse impulsionnelle du canal *incluant les filtres de modulation et de réception* et évaluée aux instants d'échantillonnage.

Donc si l'on a $\Gamma_{x',x'}(n) \approx \delta(n)$, l'inter-corrélation $\Gamma_{x',r}(n)$ fournit une estimée de la réponse impulsionnelle $h(nT_s)$.

Il convient de remarquer que pour un canal non-stationnaire comme le canal radio-mobile, les éléments binaires en début et en fin de burst (donc les plus éloignés de la séquence d'apprentissage) subiront des taux d'erreurs binaires plus importants.

C.3.2 Egaliseur de Viterbi

On suppose désormais disposer d'une estimée de la réponse impulsionnelle en bande de base $\hat{h}_n = \hat{h}(nT_s)$ du canal de transmission *incluant les filtres de modulation et de réception.* L'égalisation de la séquence reçue $r_n = r(nT_s)$ est un problème de *déconvolution* qui peut être traiter de manière analogue au décodage d'un code *convolutif.* A ce titre, le canal apparaît comme un code de rendement unité et de polynôme générateur $\hat{h}_n$. On peut facilement établir la *métrique de branche* du treillis associé à ce code :

$$\Delta M_n = \left| r_n - \sum_{l=0}^{L-1} \hat{h}_l x'_{n-l} \right|^2 \tag{C.15}$$

où $x'_n = \pm 1$ est le signal numérique émis en entrée du modulateur.

Il paraît essentiel d'utiliser un algorithme de décodage générant des *sorties souples* dans la mesure où celles-ci sont ensuite (après dé-entrelacement) placées en entrée du décodeur canal. Il existe plusieurs algorithmes de décodage convolutif fournissant des sorties souples (Annexe D). Le point commun à tous ces algorithmes est que la *sortie souple* du décodeur convolutif peut s'interpréter selon :

$$L\big(x'_n\big) = \log \frac{p\big(x'_n = +1 | \mathbf{r}\big)}{p\big(x'_n = -1 | \mathbf{r}\big)} \tag{C.16}$$

où $\mathbf{r}$ désigne la séquence des échantillons r_n en entrée de l'égaliseur (cf. Figure C.4).

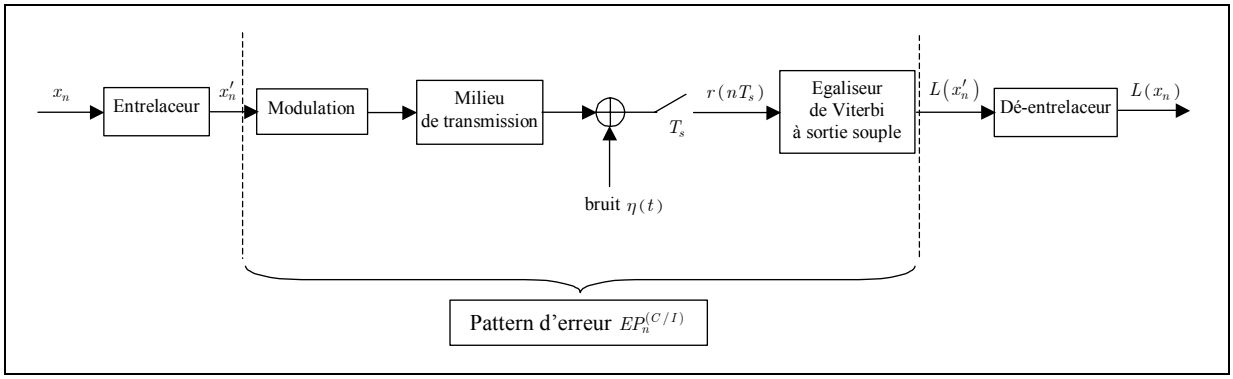

Figure C.4 : Récepteur interne à sortie souple

D'un point de vue plus physique, la sortie souple $L(x_n)$ *après dé-entrelacement* peut être interprétée de deux façons :

- **CABG :** La séquence $L(x_n)$ apparaît comme une variable aléatoire gaussienne décorrélée (en raison du dé-entrelacement), de moyenne $m = E(L(x_n))$ et de variance $\sigma^2 = \mathrm{var}(L(x_n))$. La sortie du récepteur *interne* illustré Figure C.4 apparaît comme la sortie d'un *canal additif à bruit gaussien* (CABG) sans mémoire.

- **CBS :** On peut former les décisions *fermes* $y_n = \mathrm{sign}(L(x_n))$ en sortie du canal équivalent. Celui-ci s'interprète alors comme un *canal binaire symétrique* sans mémoire (en raison du dé-entrelacement) de probabilité d'erreur *instantanée* p_{e_n} connue. Cette probabilité d'erreur associée à la décision y_n dérive de la sortie souple $L(x_n)$ selon (cf. Chapitre 3) :

$$L(x_n) = \log \frac{p(x_n = +1 | \mathbf{r})}{p(x_n = -1 | \mathbf{r})} = y_n \log \frac{p(y_n | \mathbf{r})}{1 - p(y_n | \mathbf{r})} = y_n \log \frac{1 - p_{e_n}}{p_{e_n}} \tag{C.17}$$

soit : $$p_{e_n} = \frac{1}{1 + \exp|L(x_n)|} \quad \text{(C.18)}$$

Si le point de vue CABG est celui généralement adopté pour interpréter les entrées du décodeur canal (sorties souples du canal de transmission équivalent), le point de vue CBS est en revanche celui avec lequel le décodeur de parole souple (cf. Chapitre 3) interprète la sortie du décodeur de canal à sorties souples (SOVA). Aussi, c'est ce **point de vue CBS que nous adopterons** systématiquement afin d'utiliser le même formalisme en entrée du décodeur canal et en entrée du décodeur souple de parole.

C.4 Simulation du canal par insertion d'erreur

Les conditions de transmission utilisées pour évaluer les algorithmes étudiés dans ce document ont toutes été obtenues à partir d'une *simulation du canal de propagation radio*. Le simulateur utilisé permet de régler les paramètres principaux suivants :

- Rapport *signal utile* (porteuse) sur *bruit* (interférences) C/I.
- Nombre de trajets du canal radio-mobile.
- Vitesse du mobile.
- Saut de fréquence (idéal, cyclique ou désactivé).

Afin de pouvoir comparer les algorithmes sur les mêmes *configurations d'insertion d'erreur* engendrées par la transmission radio, le simulateur de canal est utilisé pour générer des « pattern d'erreur ». Ces patterns d'erreur représentent la *relation entrée-sortie* du système comprenant le modulateur, le canal radio et le démodulateur-égaliseur à sortie souple, tel qu'illustré Figure C.4. Plus précisément, cette relation entrée-sortie instantanée est donnée à l'instant d'échantillonnage n par :

$$EP_n^{(C/I)} = \frac{L\left(x'_n\right)}{x'_n} \text{ avec } x'_n \in \{+1;-1\} \quad \text{(C.19)}$$

Les patterns d'erreur sont ainsi générés une fois pour toutes, pour un ensemble de conditions de transmission fixé. Ces patterns sont ensuite utilisés pour obtenir les valeurs souples $L\left(x'_n\right)$ mises en entrée du dé-entrelacement afin de simuler une condition donnée de transmission.

Les conditions de transmission utilisées pour les évaluations des algorithmes sont les suivantes :

- Rapport C/I variant dans la plage de 2dB à 10dB
- Canal TU50 (12 trajets et vitesse du mobile de 50km/h)
- Saut de fréquence idéal.

On notera que le rapport C/I ne varie pas au cours d'une même simulation de transmission. Autrement dit, chaque pattern d'erreur est associé à un rapport C/I fixe, c'est pourquoi on les désigne sous la terminologie « patterns d'erreurs fixes ».

Annexe D

Décodage convolutif à sorties souples

On s'intéresse ici au problème de l'obtention d'une information de *fiabilité* sur les bits décodés en sortie d'un décodeur convolutif. Les principaux algorithmes basés sur la structure du treillis sont présentés ainsi que l'interprétation exacte de l'information de fiabilité qu'ils renvoient.

D.1 Classes d'algorithmes de décodage à sorties souples

On distingue deux grandes classes d'algorithmes de décodage à sorties souples s'appuyant sur la structure du treillis :

- Algorithmes délivrant des listes [Hashimoto, 1987]

  Ces algorithmes, basés sur une généralisation de l'algorithme de Viterbi (GVA), renvoient la liste des M meilleurs chemins dans le treillis au lieu de se borner au chemin le plus vraisemblable. Ainsi, pour chaque symbole à décoder, on dispose de M décisions fermes, pondérées par les métriques des chemins associés. Un étage postérieur de traitement pourra ensuite utiliser cette liste pour générer une valeur souple du symbole décodé. On ne s'étendra pas plus ici sur cette classe d'algorithmes.

- Algorithmes délivrant des symboles « souples »

L'algorithme optimal est ici l'algorithme du Maximum a Posteriori MAP [Bahl et al., 1974]. Cet algorithme délivre, en chaque instant, la probabilité (marginale) *a posteriori* du symbole à décoder. Il permet donc de minimiser le taux d'erreur par symbole décodé et non la probabilité d'erreur par séquence comme l'algorithme de Viterbi. Cependant, cet algorithme est très complexe et pose des difficultés d'implémentation. Aussi, des algorithmes sous-optimaux mais ne nécessitant pas de

modification majeure de l'algorithme de Viterbi ont été proposés. On présentera ici l'algorithme du Max-Log-MAP [Koch et al., 1990] et l'algorithme SOVA (*Soft Output Viterbi Algorithm*) [Hagenauer et al., 1989].

On considère dans tout ce qui suit un code convolutif de mémoire ν et de rendement $R = 1/N$. On suppose également un fonctionnement du codeur par trames de longueur L, ce qui correspond au mode utilisé pour les applications de type GSM. De plus, l'état initial et l'état final est forcé à zéro par des *tail bits* (bits nuls placés en fin de message, cf. Annexe B).

On notera $\mathbf{q} = [q_0,...,q_k,...,q_L]$ une séquence d'états du codeur. Chaque transition (q_{k-1}, q_k), représentée par une branche du treillis, est associée à une valeur u_k du bit d'information rentré à l'étape k et à une valeur de symbole canal émis $\mathbf{x}_k = [x_{k,1},...,x_{k,n},...,x_{k,N}]$. Enfin, $\mathbf{Y} = [\mathbf{y}_1,...,\mathbf{y}_k,...,\mathbf{y}_L]$ est la séquence de symboles reçus en sortie du canal équivalent.

D.1.1 L'algorithme MAP

Le principe de cet algorithme est de minimiser la probabilité d'erreur sur chaque bit décodé, soit de manière équivalente, de trouver les états q_k qui sont *individuellement* les plus probables. Pour cela, on calcule la probabilité *a posteriori* :

$$\gamma_k(i) \triangleq p(q_k = q|\mathbf{Y}) \ ; \ q \in \mathbb{N} \tag{D.1}$$

c'est-à-dire la probabilité d'être dans l'état q du treillis à l'étape k conditionnellement à la séquence reçue. Les probabilités *a posteriori* pour le bit d'information u_k s'en déduisent alors immédiatement[97] :

$$p(u_k = 0|\mathbf{Y}) = \sum_{q \text{ pair}} \gamma_k(q) \tag{D.2}$$

$$p(u_k = 1|\mathbf{Y}) = \sum_{q \text{ impair}} \gamma_k(q) \tag{D.3}$$

Le calcul de la probabilité (D.1) s'effectue par la procédure *forward-backward* qui exploite les propriétés du treillis. Introduisons les variables *forward* $\alpha_k(q)$ et *backward* $\beta_k(q)$ définie par :

$$\alpha_k(q) = p(\mathbf{y}_1,...,\mathbf{y}_k, q_k = q) \tag{D.4}$$

---

[97] On suppose ici que les bits des registres d'état du codeur convolutif correspondent à la représentation binaire naturelle des états q et que ces registres sont à décalage vers la gauche, c'est-à-dire que le bit d'information u_k est rentré à droite du registre à l'étape k.

et
$$\beta_k(q) = p\big(\mathbf{y}_{k+1},...,\mathbf{y}_L \big| q_k = q\big) \tag{D.5}$$

Ces variables peuvent se calculer de manière itérative comme suit :

$$\alpha_k(q) = \sum_{q'} \alpha_{k-1}(q') p\big(\mathbf{y}_k \big| q_{k-1} = q', q_k = q\big) p\big(q_k = q\big) \tag{D.6}$$

$$\beta_k(q) = \sum_{q'} \beta_{k+1}(q') p\big(\mathbf{y}_{k+1} \big| q_k = q, q_{k+1} = q'\big) p\big(q_{k+1} = q'\big) \tag{D.7}$$

où q' parcourt les états reliés à l'état q par une branche du treillis.

Les états sont considérés ici comme équiprobables, ce qui signifie que les termes $p\big(q_k = q\big)$ et $p\big(q_{k+1} = q'\big)$ sont vus comme constants et peuvent être ignorés. La probabilité $p\big(\mathbf{y}_k \big| q_{k-1} = q', q_k = q\big)$ apparaît être la *vraisemblance de la transition* $\big(q_{k-1}, q_k\big)$, son logarithme n'est autre que la *métrique de branche* du treillis. D'autre part, l'état initial et l'état final étant fixés à zéros, on a :

$$\alpha_0 = [1,0,...,0] \text{ et } \beta_0 = [1,0,...,0] \tag{D.8}$$

La récurrence sur α_k se fait dans le sens des k croissants (*forward*) et celle sur β_k se fait en sens rétrograde (*backward*). Une fois calculées ces quantités, on peut exprimer $\gamma_k(q)$ par :

$$\gamma_k(q) = \frac{\alpha_k(q)\beta_k(q)}{\sum_{q=0}^{2^\nu - 1} \alpha_k(q)\beta_k(q)} \tag{D.9}$$

En définitive, l'algorithme MAP renvoie comme valeur souple, le logarithme du rapport des probabilités *a posteriori* (D.2) et (D.3) :

$$L\big(u_k\big) = \log \frac{p\big(\hat{u}_k = 0 | \mathbf{Y}\big)}{p\big(\hat{u}_k = 1 | \mathbf{Y}\big)} \tag{D.10}$$

Cependant cet algorithme est peu utilisé dans la pratique du fait de la complexité de la procédure *forward-backward* qui ne peut être implémentée dans le domaine logarithmique. Une simplification de cet algorithme consiste à ne garder dans les sommations (D.2) et (D.3) que les termes prépondérants, ce qui conduit à l'algorithme du Max-Log-MAP.

D.1.2 L'algorithme Max-Log-MAP

La complexité de l'algorithme MAP provient de la sommation sur les états lors du calcul des variables *forward* et *backward* (récursions (D.6) et (D.7)) ainsi que dans le calcul des probabilités *a posteriori*

(D.2) et (D.3). La simplification à la base du Max-Log-MAP est l'approximation dite de la « *séquence dominante* ». Cette approximation est valable tant que le rapport signal à bruit en entrée du décodeur n'est pas trop bas. Elle est mise en œuvre successivement à deux niveaux. Plus précisément :

On suppose qu'il existe des *séquences d'états dominantes*, c'est-à-dire des chemins beaucoup plus probables que les autres, de telle sorte que :

$$\begin{aligned} p(q_k = q|\mathbf{Y}) &= \sum_{\mathbf{q}\in Q(q,k)} p(\mathbf{q}|\mathbf{Y}) \\ &\approx \max_{\mathbf{q}\in Q(q,k)} p(\mathbf{q}|\mathbf{Y}) \end{aligned} \tag{D.11}$$

où $Q(q,k)$ désigne l'ensemble des chemins du treillis passant par l'état q à l'étape k (nœud $q_k = q$).

Autrement dit, on remplace la sommation par une maximisation. Ceci permet de se ramener à la structure d'un algorithme de Viterbi, cherchant le meilleur *chemin* passant par le nœud $q_k = q$. On notera $\delta_k(q)$ la probabilité de ce chemin :

$$\delta_k(q) \triangleq \max_{\mathbf{q}\in Q(q,k)} p(\mathbf{q}|\mathbf{Y}) \tag{D.12}$$

Pour maintenir l'analogie avec l'algorithme MAP, introduisons la variable « *forward* » modifiée suivante :

$$\tilde{\alpha}_k(q) = \max_{q_1,\ldots,q_{k-1}} p\big(\mathbf{y}_1,\ldots,\mathbf{y}_k,q_1,q_2,\ldots,q_k = q\big) \tag{D.13}$$

qui s'interprète comme la probabilité du meilleur chemin *partiel* jusqu'au nœud $q_k = q$ (ou chemin survivant). On a d'après la règle de Bayes :

$$\tilde{\alpha}_k(q) = C \max_{q'} \big\{ \tilde{\alpha}_{k-1}(q')\, p\big(\mathbf{y}_k \big| q_{k-1} = q', q_k = q\big)\, p\big(q_k = q\big) \big\} \tag{D.14}$$

où C est une constante de normalisation.

Comme on considère ici que les états sont *a priori* équiprobables, le calcul de la probabilité *a posteriori* se réduit à un calcul de vraisemblance, soit :

$$\tilde{\alpha}_k(q) = C \max_{q'} \big\{ \tilde{\alpha}_{k-1}(q')\, p\big(\mathbf{y}_k \big| q_{k-1} = q', q_k = q\big) \big\} \tag{D.15}$$

ce qui, dans le domaine logarithmique correspond au calcul de métrique de l'algorithme de Viterbi :

$$M_k(q) = \max_{q'} \big\{ M_{k-1}(q') + \log\big[\, p\big(\mathbf{y}_k \big| q_{k-1} = q', q_k = q\big)\big] \big\} \tag{D.16}$$

où $M_k(q)$ est la métrique du chemin survivant au nœud $q_k = q$.

On remarquera que la récursion (D.14) dérive de celle du MAP (D.6) en remplaçant l'intégration sur tous les états précédents q_{k-1} par la sélection du meilleur état. De cette constatation, il apparaît qu'on peut aussi calculer la métrique de Viterbi dans le sens des k décroissants (dérivation de la récursion *backward*) :

$$M_k(q) = \max_{q'} \left\{ M_{k+1}(q') + \log\left[p\left(\mathbf{y}_{k+1} \middle| q_k = q, q_{k+1} = q' \right) \right] \right\} \tag{D.17}$$

Ce calcul des métriques de Viterbi dans les deux sens [Tortelier, 1995] permet d'exprimer, au prix d'une complexité réduite (absence de *traceback*), la métrique totale $\log\{\delta_k(q)\}$ du meilleur chemin *complet* passant par le nœud $q_k = q$.

Une fois calculée la probabilité $\delta_k(q)$, on peut exprimer d'après (D.11), (D.2) et (D.3), les probabilités *a posteriori* d'obtenir un bit décodé u_k égal à 0 ou 1 :

$$p\left(u_k = 0 | \mathbf{Y}\right) = \sum_{q \text{ pair}} \delta_k(q) \tag{D.18}$$

$$p\left(u_k = 1 | \mathbf{Y}\right) = \sum_{q \text{ impair}} \delta_k(q) \tag{D.19}$$

L'approximation de la séquence dominante est appliquée ici une seconde fois, c'est-à-dire qu'on ne considère dans chacune de ces deux sommes que les chemins de probabilité maximale. Ainsi, l'algorithme Max-Log-MAP renvoie simplement *la différence entre les métriques des meilleurs chemins décodant respectivement* $u_k = 0$ *et* $u_k = 1$.

$$L\left(u_k\right) = \log \frac{p\left(\hat{u}_k = 0 | \mathbf{Y}\right)}{p\left(\hat{u}_k = 1 | \mathbf{Y}\right)} = M_0 - M_1 \tag{D.20}$$

avec : $M_0 = \log\left[\max_{q \text{ pair}} \{\delta_k(q)\} \right]$ et $M_1 = \log\left[\max_{q \text{ impair}} \{\delta_k(q)\} \right]$.

D.1.3 L'algorithme SOVA

L'algorithme du Max-Log-MAP est plutôt adapté à un fonctionnement par trames, dans le cas contraire (décodage à flot continu), sa complexité demeure relativement importante. L'algorithme SOVA introduit par [Hagenauer et al., 1989] répond au souci de modifier le moins possible l'algorithme de Viterbi en lui adjoignant une étape de décision souple de complexité limitée. C'est aussi l'algorithme le moins optimal parmi les trois présentés ici.

Considérons un algorithme de Viterbi de profondeur de décodage ou *délai de décision* δ . Pour chaque chemin survivant $\mathbf{q}_k^{(\ell)} = \left[q_{k-\delta}^{(\ell)}, \ldots, q_{k-1}^{(\ell)}, q_k^{(\ell)} \right]$ *à l'étape* k, le SOVA stocke la décision ferme $u_j^{(\ell)}$ prise

pour les bits d'information aux étapes $(k-\delta \leq j < k)$ (équivalent à stocker les états formant le chemin) ainsi qu'une information de fiabilité de cette décision :

$$L_j^{(\ell)}(k) = \log \frac{1 - p_j^{(\ell)}(k)}{p_j^{(\ell)}(k)} \tag{D.21}$$

où $p_j^{(\ell)}(k)$ est la probabilité d'erreur associée à la décision $u_j^{(\ell)}$ sachant $\mathbf{Y}_1^k = \left[\mathbf{y}_1,...,\mathbf{y}_j,...,\mathbf{y}_k\right]$.

Pour expliciter le calcul de la fiabilité (D.21), considérons le chemin survivant $\mathbf{q}_k^{(1)}$ au nœud $q_k = q$ et $\mathbf{q}_k^{(2)}$ le chemin éliminé par l'algorithme de Viterbi en ce nœud (Figure D.1).

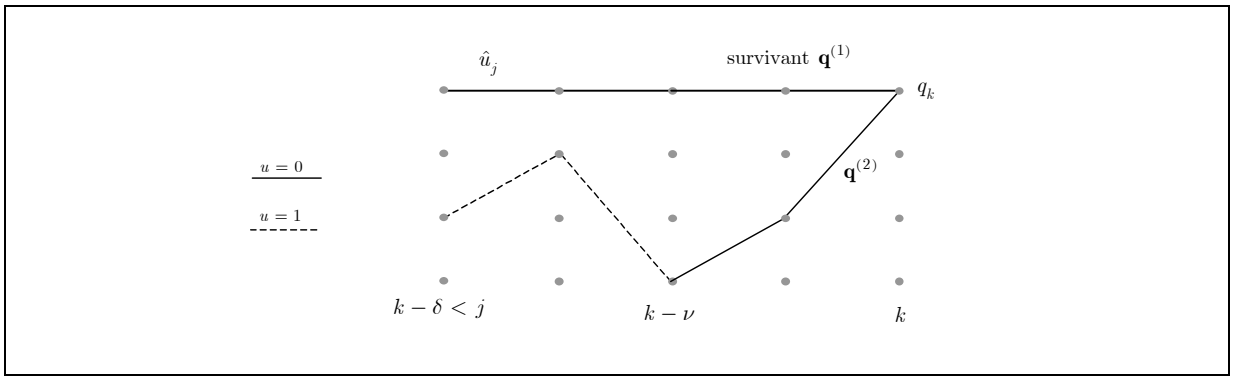

Figure D.1 : Séléction du chemin survivant

En supposant les états équiprobables, les probabilités *a posteriori* $p\left(\mathbf{q}_k^{(\ell)} \middle| \mathbf{Y}_1^k\right)$ se réduisent aux vraisemblances $p\left(\mathbf{Y}_1^k \middle| \mathbf{q}_k^{(\ell)}\right)$ et l'on a :

$$p\left(\mathbf{q}_k^{(\ell)} \middle| \mathbf{Y}_1^k\right) \propto p\left(\mathbf{Y}_1^k \middle| \mathbf{q}_k^{(\ell)}\right) \propto e^{M_k^{(\ell)}} \tag{D.22}$$

où $M_k^{(\ell)}$ est la métrique cumulée du chemin $\mathbf{q}_k^{(\ell)}$.

On peut dès lors exprimer la probabilité d'erreur p_{q_k} associée à la décision de l'algorithme de Viterbi au nœud $q_k = q$:

$$p_{q_k} = \frac{p\left(\mathbf{q}_k^{(2)} \middle| \mathbf{Y}_1^k\right)}{p\left(\mathbf{q}_k^{(1)} \middle| \mathbf{Y}_1^k\right) + p\left(\mathbf{q}_k^{(2)} \middle| \mathbf{Y}_1^k\right)} = \frac{e^{M_k^{(2)}}}{e^{M_k^{(1)}} + e^{M_k^{(2)}}} = \frac{1}{1 + e^{\Delta}} \tag{D.23}$$

avec $\Delta = M_1 - M_2 \geq 0$.

L'erreur de décision au nœud $q_k = q$ se répercute de la façon suivante sur les décisions $u_j^{(1)}$:

- si $k - \nu < j \leq k$, l'erreur n'a aucune conséquence puisque $u_j^{(1)} = u_j^{(2)}$.

- pour $j = k - \nu$, on a forcément $u_j^{(1)} \neq u_j^{(2)}$ donc la probabilité d'erreur associée à la décision $\hat{u}_j = u_j^{(1)}$ prise à l'instant t est : $p_j^{(1)}(k) = p_{q_k}$.

- pour $k - \delta \leq j < k - \nu$, on dispose déjà des probabilités d'erreurs $p_j^{(1)}(k-1)$ et $p_j^{(2)}(k-1)$ associées respectivement aux décisions $\hat{u}_j = u_j^{(1)}$ et $\hat{u}_j = u_j^{(2)}$ sachant $\mathbf{Y}_1^{k-1}$. La mise à jour de la probabilité d'erreur pour le chemin survivant doit se faire selon les deux cas :

$$\text{si } u_j^{(1)} = u_j^{(2)} \text{ alors } p_j^{(1)}(k) = \left(1 - p_{q_k}\right) p_j^{(1)}(k-1) + p_{q_k} p_j^{(2)}(k-1) \tag{D.24}$$

$$\text{si } u_j^{(1)} \neq u_j^{(2)} \text{ alors } p_j^{(1)}(k) = \left(1 - p_{q_k}\right) p_j^{(1)}(k-1) + p_{q_k}\left(1 - p_j^{(2)}(k-1)\right) \tag{D.25}$$

Afin de simplifier la mise à jour des fiabilités (D.21), on fait *l'approximation* suivante :

$$p_j^{(2)} = p_j^{(1)} \tag{D.26}$$

ainsi, $p_j^{(1)}$ est invariante dans le cas $u_j^{(1)} = u_j^{(2)}$ et dans le cas $u_j^{(1)} \neq u_j^{(2)}$, la mise à jour devient :

$$\text{si } u_j^{(1)} \neq u_j^{(2)} \text{ alors } p_j^{(1)}(k) = \left(1 - p_{q_k}\right) p_j^{(1)}(k-1) + p_{q_k}\left(1 - p_j^{(1)}(k-1)\right) \tag{D.27}$$

La récurrence (D.27) peut s'effectuer directement dans le domaine des fiabilités (domaine logarithmique). D'après (D.23), il vient :

$$L_j(k) = \ln\left\{\frac{1 + \exp(L_j(k-1))\exp(\Delta)}{\exp(L_j(k-1)) + \exp(\Delta)}\right\} \tag{D.28}$$

où on a omis les indices de chemins ℓ pour plus de lisibilité.

Néanmoins, la formule précédente demeure complexe, c'est pourquoi on préfère l'approximation suivante [Hagenauer et al., 1989] qui, bien que sous optimale, donne de bons résultats :

$$L_j(k) = \min(L_j(k-1), \Delta) \tag{D.29}$$

Finalement, au bout du délai δ, le SOVA relâche la sortie souple :

$$L\left(u_j\right) = \hat{u}_j.\hat{L}_j(j+\delta) = \hat{u}_j \log\frac{1 - p\left(\hat{u}_j \neq u_j \middle| \mathbf{Y}_1^{j+\delta}\right)}{p\left(\hat{u}_j \neq u_j \middle| \mathbf{Y}_1^{j+\delta}\right)} \tag{D.30}$$

D.1.4 Comparaison du MAP, Max-Log-MAP et du SOVA

Pour résumer cette présentation, nous mettons ici en exergue les différences d'approche entre les 3 algorithmes présentés. Ces différences sont illustrées sur la Figure D.2.

- Le MAP considère *tous les chemins du treillis* mais les divise en deux ensembles : ceux qui décodent la valeur 0 à l'étape k et ceux qui décodent la valeur 1. Il retourne le rapport entre les probabilités *a posteriori* de ces 2 ensembles.

- Le Max-Log-MAP considère à chaque étape k uniquement deux chemins : le meilleur qui décode un 0 et le meilleur qui décode un 1. Il délivre le rapport des probabilités de ces 2 chemins. Ces chemins peuvent changer d'un instant à l'autre même si l'un d'entre eux demeure toujours le chemin de vraisemblance maximum.

- Le SOVA compare, à chaque étape k, le chemin de vraisemblance maximum à un chemin décodant une valeur opposée à l'étape k mais ce dernier n'est pas forcément le meilleur chemin décodant une valeur opposée à cette étape.

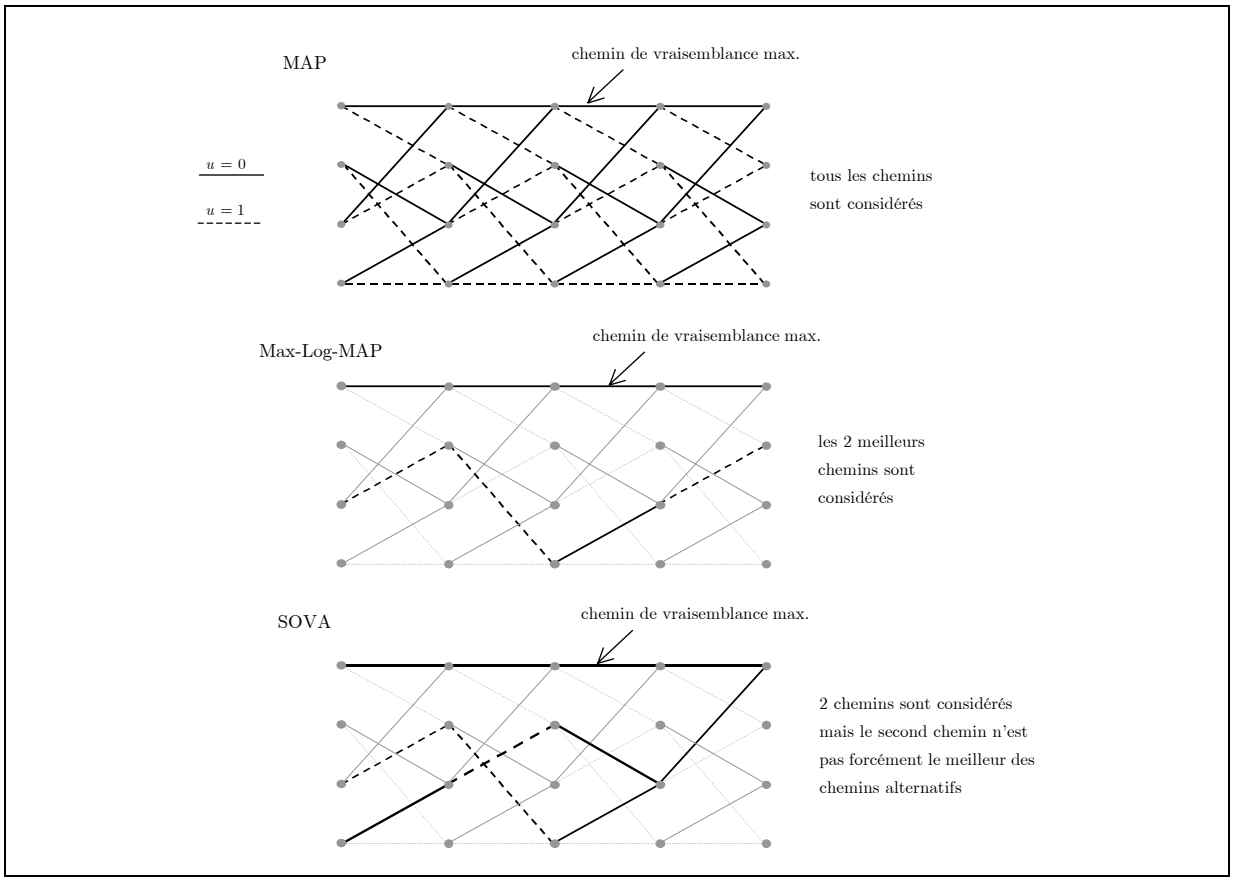

Figure D.2 : Comparaison des chemins exploités par le MAP, le Max-Log-MAP et le SOVA afin d'estimer une information de fiabilité

Bibliographie

[Adrat et al., 2000] M. Adrat, J. Spittka, S. Heinen, et P. Vary, "Error Concealment by Near Optimum MMSE Estimation of Source Codec Parameters," in Proc. IEEE Speech Coding Workshop, pp. 84-86, 2000.

[Alajaji et al., 1996] F. I. Alajaji, N. C. Phamdo, et T. E. Fuja, "Channel Codes that Exploits the Residual Redundancy in CELP Encoded Speech," IEEE Trans. Speech Audio Process., vol. 4, pp. 325-336, 1996.

[Atungsiri et al., 1990] S. A. Atungsiri, A. M. Kondoz, et B. G. Evans, "Error Detection and Control for the Parametric Information in CELP Coders," in Proc. ICASSP, pp. 229-232, 1990.

[Bahl et al., 1974] L. R. Bahl, J. Cocke, F. Jelinek, et J. Raviv, "Optimal Decoding of Linear Codes for minimizing Symbol Error Rate," IEEE Trans. Inform. Theory, vol. 20, pp. 284-287, 1974.

[Bayya et al., 1996] A. Bayya et M. Vis, "Objective Measures for Speech Quality Assessment in Wireless Communications," in Proc. ICASSP, vol. 1, pp. 495-498, 1996.

[Beaugeant, 1999] C. Beaugeant, Réduction de Bruit et Contrôle de l'Echo pour les Applications Radiomobiles, Thèse de Doctorat, Université de Rennes I, 1999.

[Boite et al., 1987] R. Boite et M. Kunt, Traitement de la parole: Presses Polytechniques Romandes, 1987.

[Cox et al., 1989] R. V. Cox, W. B. Kleijn, et P. Kroon, "Robust CELP Coders for Noisy Backgrounds and Noisy Channels," in Proc. ICASSP, vol. 1, pp. 739-742, 1989.

[Cruchant et al., 1998] L. Cruchant et P. Dupuy, "La qualité de parole dans les systèmes GSM," in La Revue des Télécommunications d'Alcatel, vol. 4, 1998, pp. 281-285.

[De Martin et al., 2000] J. C. De Martin, T. Unno, et V. Viswanathan, "Improved Frame Erasure Concealment for CELP-Based Coders," in Proc. ICASSP, vol. 3, pp. 1483-1486, 2000.

[Dogan, 1992] M. C. Dogan, "Real-Time robust Pitch Detector," in Proc. ICASSP, vol. I, pp. 129-132, 1992.

[Duhamel et al., 1997] P. Duhamel et O. Rioul, "Codage Conjoint Source/Canal : Enjeux et Approches," in Proc. Colloque Gretsi, pp. 699-704, Grenoble, 1997.

[Erkelens et al., 1995] J. S. Erkelens et P. M. T. Broersen, "On statistical properties of line spectrum pairs," in Proc. ICASSP, pp. 768-771, 1995.

[Fingscheidt et al., 2000] T. Fingscheidt, T. Hindelang, R. V. Cox, et N. Seshadri, "Combined Source/Channel Decoding: When Minimizing Bit Error Rate is Suboptimal," in Proc. 3rd ITG Conf. Source Channel Coding, pp. 273-277, Munich, Germany, 2000.

[Fingscheidt et al., 2000] T. Fingscheidt, T. Hindelang, R. V. Cox, et N. Seshadri, "On Quantizer Dimensions in Joint Speech/Channel Coding," in Proc. IEEE Speech Coding Workshop, pp. 81-83, 2000.

[Fingscheidt et al., 1997] T. Fingscheidt et O. Scheufen, "Robust GSM Speech Decoding Using the Channel Decoder's Soft Output," in Proc. Eurospeech, pp. 1315-1318, 1997.

[Fingscheidt et al., 1997] T. Fingscheidt et P. Vary, "Robust Speech Decoding : A Universal Approach to Bit Error Concealment," in Proc. ICASSP, pp. 1667-1670, 1997.

[Fingscheidt et al., 2001] T. Fingscheidt et P. Vary, "Softbit Speech Decoding : A New Approach to Error Concealment," IEEE Trans. Speech Audio Process., vol. 9, pp. 240-251, 2001.

[Forney, 1973] J. D. Forney, "The Viterbi Algorithm," in Proc. Proc. IEEE, vol. 61, pp. 268-278, 1973.

[Gerlach, 1993] C. G. Gerlach, "A Probalbilistic Frame work for Optimum Speech Extrapolation in Digital Mobile Radio," in Proc. ICASSP, vol. 2, pp. 419-422, 1993.

[Görtz, 1997] N. Görtz, "Zero-Redundancy Error Protection For CELP Speech Codecs," in Proc. Eurospeech, vol. 3, pp. 1283-1286, 1997.

[Görtz, 1998] N. Görtz, "On The Combination Of Redundant And Zero-Redundant Channel Error Detection In CELP Speech-Coding," in Proc. ICASSP, pp. 721-724, 1998.

[Gray et al., 2000] P. Gray, M. P. Hollier, et R. E. Massara, "Non-intrusive speech quality assessment using vocal-tract models," IEE Proc. -Vis. Image Signal Process., vol. 147, pp. 493-501, 2000.

[GSM, 05.03] GSM, "Digital cellular telecommunication system (Phase 2+); Channel coding," GSM, Recommandation 05.03.

[GSM, 06.60] GSM, "Digital cellular telecommunication system; Enhanced Full Rate speech transcoding," GSM, Recommandation 06.60.

[GSM, 06.10] GSM, "Digital cellular telecommunication system; Full Rate speech transcoding," GSM, Recommandation 06.10.

[GSM, 06.61] GSM, "Substitution and muting of lost frames for Enhanced Full Rate speech traffic channels," GSM, Recommandation 06.61.

[Hagenauer, 1995] J. Hagenauer, "Source-Controlled Channel Decoding," IEEE Trans. on Communications, vol. 43, pp. 2449-2457, 1995.

[Hagenauer et al., 2003] J. Hagenauer et N. Gortz, "The Turbo Principle in Joint Source-Channel Coding," in Proc. ITW2003, pp. pp. 275–278, 2003.

[Hagenauer et al., 1989] J. Hagenauer et P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," in Proc. GLOBECOM'89, pp. 1680-1686, 1989.

[Hashimoto, 1987] T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," IEEE Trans. on Information Theory, vol. 33, pp. 866-876, 1987.

[Hedelin et al., 1995] P. Hedelin, P. Knagenhjelm, et M. Skoglund, "Theory for Transmission of Vector Quantization Data," in Speech Coding and Synthesis, W. B. Kleijn et K. K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995.

[Hedelin et al., 1995] P. Hedelin, P. Knagenhjelm, et M. Skoglund, "Vector Quantization for Speech Transmission," in Speech Coding and Synthesis, W. B. Kleijn et K. K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995, pp. 311-345.

[Hedelin et al., 2000] P. Hedelin et J. Skoglund, "Vector Quantization Based on Gaussian Mixture Models," IEEE Trans. Speech Audio Process., vol. 8, pp. 385-401, 2000.

[Heinen et al., 1997] S. Heinen, A. Geiler, et P. Vary, "MAP Channel Decoding by Exploiting Multilevel Source A Priori Knowledge," in Proc. European Personal Mobile Communications Conference (EPMCC), pp. 467-473, Bonn, Germany, 1997.

[Heinen et al., 2000] S. Heinen et P. Vary, "Joint Source-Channel MMSE-Decoding of Speech Parameters," in Proc. ICASSP, vol. 3, pp. 1507-1510, 2000.

[Hess, 1983] W. Hess, Algorithms and Devices for Pitch Determination of Speech-Signals. Berlin, 1983.

[Hindelang, 2000] T. Hindelang, "Combined Source/Channel (De-)Coding : Can A Priori Information Be Used Twice?," IEEE Proc. ICC, vol. 1, pp. 1208-1212, 2000.

[Hindelang et al., 1997] T. Hindelang, W. Xu, et C. Erben, "Quality Enhancement of Coded and Corrupted Speeches in GSM Mobile Systems Using Residual Redundancy," in Proc. ICASSP, vol. 1, pp. 259-262, 1997.

[ITU-T, G.729] ITU-T, "Coding of speech at 8kbit/s using conjugate-structure algebraic-code-excited linear prediction," ITU-T, Recommendation G.729.

[Järvinen et al., 1997] K. Järvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, R. Salami, C. Laflamme, et J.-P. Adoul, "GSM Enhanced Full Rate Speech Codec," in Proc. ICASSP, vol. 1, pp. 771-774, 1997.

[Kay, 1988] S. M. Kay, Modern spectral estimation, 1988.

[Kleijn et al., 1995] B. Kleijn et K. K. Paliwal, Speech Coding and Synthesis: Elsevier Science, 1995.

[Koch et al., 1990] W. Koch et A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in Proc. GLOBECOM, pp. 1679-1684, 1990.

[Kohler et al., 2000] M. A. Kohler et R. K. Yarlagadda, "Markov Chain Prediction for Missing Speech Frame Compensation," in Proc. IEEE Speech Coding Workshop, pp. 75-77, 2000.

[Kondoz, 1994] A. M. Kondoz, Digital Speech: Wiley, J., 1994.

[Lahouti, 2003] F. Lahouti, "Reconstruction of Predictively Encoded Signals Over Noisy Channels Using A Sequence MMSE Decoder," IEEE Trans. on Communications, accepted for publication, 2003.

[Lahouti, 2003, Report] F. Lahouti, "Soft Reconstruction of Speech in the Presence of Noise and Packet Loss," University of Waterloo Report 2003.

[Lahouti et al., 2001] F. Lahouti et A. K. Khandani, "Approximating and Exploiting the Residual Redundancies - Applications to Efficient Reconstruction of Speech over Noisy Channels," in Proc. ICASSP, 2001.

[Laroche, 1995] J. Laroche, "Traitement des Signaux Audio-Fréquences," ENST, Cours de 3ième année 1995.

[Ligdas et al., 1997] P. Ligdas, W. Turin, et N. Seshadri, "Statistical Methods for Speech Transmission Using Hidden Markov Models," in Proc. 31st Conf. Information Sciences Systems, pp. 546-551, 1997.

[Lindblom et al., 2000] J. Lindblom, J. Samuelsson, et P. Hedelin, "Model Based Spectrum Prediction," in Proc. IEEE Speech Coding Workshop, pp. 117-119, 2000.

[Markel et al., 1976] J. D. Markel et A. H. Gray, Linear Prediction of Speech, 1976.

[Martin et al., 2001] R. Martin, C. Hoelper, et I. Wittke, "Estimation of Missing LSF Parameters Using Gaussian Mixture Models," in Proc. ICASSP, vol. 2, pp. 729-732, 2001.

[Miller et al., 1998] D. J. Miller et M. Park, "A Sequence-Based Approximate MMSE Decoder for Source Coding Over Noisy Channels Using Discrete Hidden Markov Models," IEEE Trans. on Communications, vol. 46, pp. 222-231, 1998.

[Moreau, 1995] N. Moreau, Techniques de compression des signaux: Masson, 1995.

[Paping et al., 1997] M. Paping et T. Fähnle, "Automatic Detection of disturbing Robot Voice and Ping Pong Effects in GSM Transmitted Speech," in Proc. Eurospeech, pp. 1631-1634, 1997.

[Pascal et al., 1999] D. Pascal et D. Etourneaud, "Performances comparées des codeurs EFR et FR en environnement non bruité," CNET, Note technique 1999.

[Phamdo et al., 1994] N. Phamdo et N. Farvardin, "Optimal Detection of Discrete Markov Sources Over Discrete Memoryless Channels -- Applications to Combined Source-Channel Coding," IEEE Trans. Information Theory, vol. 40, pp. 187-193, 1994.

[Picinbono, 1989] B. Picinbono, "Théorie des signaux et des systèmes," . Paris: Dunod, 1989.

[Proakis, 1989] J. G. Proakis, Digital Communications, 2 ed: McGraw-Hill, 1989.

[Rabiner, 1989] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

[Rissanen, 1978] J. Rissanen, "Modeling by shortest data description," Automatica, vol. 14, pp. 465-471, 1978.

[Ruscitto et al., 1997] A. Ruscitto et T. Hindelang, "Channel decoding using residual intra-frame correlation in a GSM system," Electronics Letters, vol. 33, pp. 1754-1755, 1997.

[Salami et al., 1996] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, et Y. Shoham, "Design and Description of CS-ACELP: a Toll Quality 8kb/s Speech Coder," IEEE Trans. on Speech and Audio Processing, vol. 6, pp. 116-130, 1996.

[Sayood et al., 1991] K. Sayood et J. C. Borkenhagen, "Use of Residual Redundancy in the Design of Joint Source/Channels Coders," IEEE Trans. on Communications, vol. 39, pp. 839-846, 1991.

[Scalart, 1997] P. Scalart, "Radiocommunications et Mobilité," , Cours de 3ième année de l'ENSSAT 1997.

[Schroeder et al., 1985] M. Schroeder et B. Atal, "Code-excited linear prediction (CELP) : high quality speech at very low," in Proc. ICASSP, pp. 937-940, 1985.

[Sereno, 1991] D. Sereno, "Frame Substitution and Adaptive Post-Filtering in speech Coding," in Proc. ICASSP, vol. 1, pp. 595-598, 1991.

[Shannon, 1948] C. E. Shannon, "A mathematical theory of communications," Bell Syst. Tech. J., vol. 27, pp. 379-423, 1948.

[Skoglund et al., 1997] J. Skoglund et J. Lindén, "Predictive VQ for Noisy Channel Spectrum Coding: AR or MA?," in Proc. ICASSP, pp. 1351-1354, 1997.

[Skoglund, 1999] M. Skoglund, "Soft Decoding for Vector Quantization Over Noisy Channels with Memory," IEEE Trans. Information Theory, vol. 45, pp. 1293-1307, 1999.

[Strauch et al., 1998] P. Strauch, C. Luschi, M. Sandell, et R. Yan, "Improved Source Controlled Channel Decoding in a GSM System," in Proc. ISPACS'98, Melbourne, Australia, 1998.

[Tortelier, 1995] P. Tortelier, "Procédé de décodage à sortie pondérée de codes convolutifs de rendement 1/N en fonctionnement par blocs.," mémoire technique CNET 1995.

[UIT-T, P.862] UIT-T, " Perceptual Evaluation of Speech Quality (PESQ), an Objective Method for end-to-end Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs.," , Recommendation P.862.

[Veaux, 1998] C. Veaux, "Analyse et caractérisation des dégradations de la parole dans le réseau GSM plein débit," CNET, Rapport intermédiaire 1998.

[Veaux et al., 1999] C. Veaux, P. Scalart, et A. Gilloire, "Analysis and on-line detection of audible distortions in GSM telephony," in Proc. Eurospeech, vol. 6, pp. 2579-2582, Budapest, 1999.

[Veaux et al., 2000] C. Veaux, P. Scalart, et A. Gilloire, "Channel Decoding using Adaptive Interframe and Intraframe Bit Prediction in GSM System," in Proc. ICASSP, vol. 5, pp. 2589-2592, Istanbul, 2000.

[Veaux et al., 2000] C. Veaux, P. Scalart, et A. Gilloire, "Channel Decoding Using Inter- and Intra-correlation of Source Encoded Frames," in Proc. Data Compression Conference, pp. 103-112, Snowbird, Utah, 2000.

[Wellekens, 1987] C. J. Wellekens, "Explicit Time Correlation in Hidden Markov Models for Speech Recognition," IEEE, pp. 384-386, 1987.

[Zwicker et al., 1981] E. Zwicker et R. Feldtkeller, Psychoacoustique - L'Oreille Récepteur d'Information: Masson, 1981.