

# LABO 5 - 6 - 7

## PROJET :

# SIMULATION DU PRINCIPE DE CODAGE/DECODAGE MP3 SOUS MATLAB

Karlheinz Brandenburg, mathématicien, ingénieur électricien et docteur de l'Université d'Erlangen (Nuremberg). Après un post-doctorat chez AT&T Bell Laboratories Murray Hill, USA, et un poste de professeur à l'Université, il devient chef de projet au Fraunhofer-Institut (Erlangen), où il travaille sur le codage audio perceptuel sur un projet Européen, en collaboration avec l'Université d'Erlangen (Prof. Dieter Seitzer), et co-invente le principe de codage aujourd'hui normalisé sous le nom de MPEG1-Layer3 (MP3).



Leonardo Chiariglione, Ingénieur Electricien de l'Ecole Polytechnique de Turin (1967), PhD de l'Université de Tokyo (1973), a travaillé au centre de recherches sur les télécommunications de Telecom-Italia (CSELT) jusqu'en 2001, où il occupa la fonction de vice président pour la branche multimédia. L. Chiariglione est le fondateur du groupe de standardisation ISO MPEG. Il est également l'initiateur du Digital Media Project, pour le développement d'une plateforme de gestion des droits numériques standardisée.



## 5.1 Introduction

Au cours de séances précédentes, nous avons appris à *utiliser* un certain nombre d'outils fondamentaux en traitement du signal : générateurs (et échantillonneurs), analyseurs, et filtres. Ces outils vont maintenant nous permettre de *modéliser* et *modifier* des signaux audio-numériques.

Plus précisément, le but de ce projet sera de simuler sous MATLAB le principe général du codage-décodage MPEG1-Layer3, aussi appelé MP3.

Le projet sera mené en 3 phases :

1. Nous analyserons le principe du codeur-décodeur, et poserons quelques simplifications afin de rendre le problème traitable en 3 séances.
2. Nous concevrons un système de base sans compression, organisé autour d'un banc de filtres en sous-bandes, pour vérifier que les pertes

engendrées par les calculs sont négligeables. Cette étape nous permettra également d'estimer la charge de calcul minimale avant dégradation du signal.

3. Nous réaliserons la même opération par FFT/IFFT, et comparerons les résultats obtenus avec le système précédent, ainsi que la charge de calcul nécessaire.

## 5.2 Codage-Décodage MPEG1

### 5.2.1 Audition - Masquage auditif

L'oreille humaine (c.-à-d. le *cerveau* humain) est un capteur fortement non-linéaire, à plusieurs niveaux :

- Elle présente une sensibilité variant avec la fréquence. On définit ainsi un *seuil de l'audition* comme le niveau de pression acoustique minimal (en dB SPL ; Sound Pressure Level) pour qu'un son sinusoïdal pur soit juste perçu (Fig. 1).

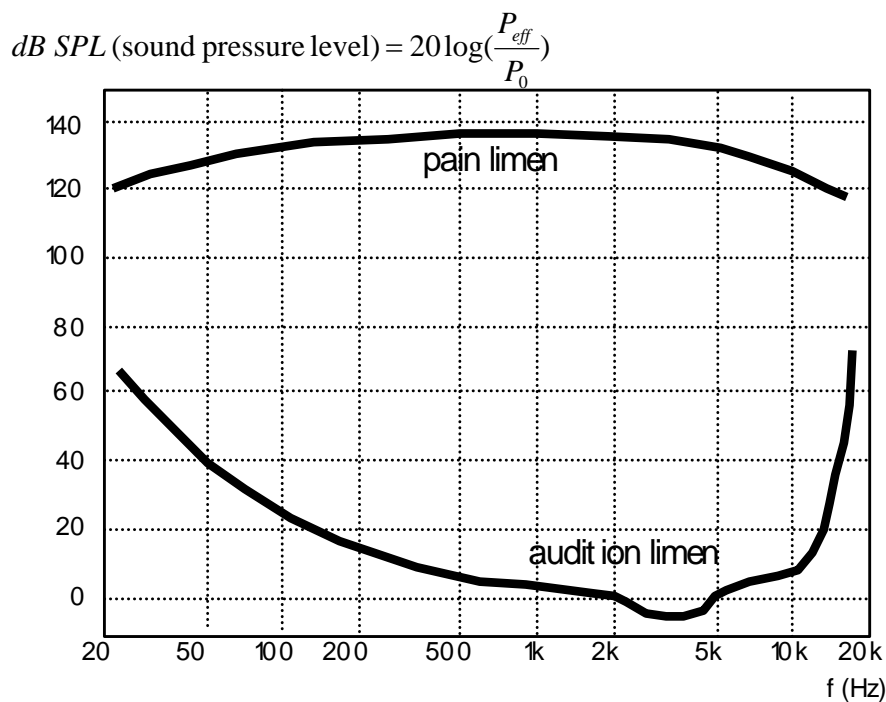


Fig. 1. Seuil d'audition; Seuil de la douleur.

- La perception auditive à une fréquence donnée n'évolue pas linéairement en fonction du niveau sonore, mais plutôt à peu près en fonction de son logarithme (dB).
- Cette évolution est limitée vers le haut par la courbe du *seuil de la douleur*, qui correspond à la pression acoustique d'une sinusoïde pure pour laquelle le cerveau perçoit une douleur, et distord d'ailleurs fortement le son perçu.

- Enfin, et c'est ici essentiel, un son peut en cacher un autre. Le phénomène de *masquage auditif* est décrit à la Fig. 2. Une sinusoïde perçue, de fréquence et d'amplitude donnée, crée une distorsion locale de la courbe du seuil de l'audition. Ainsi, par exemple, une sinusoïde à 1000 Hz d'une intensité de 80 dB SPL rendra inaudible une seconde sinusoïde à 2000 Hz et d'intensité 40 dB SPL, alors que cette sinusoïde serait parfaitement perçue sans la présence de la première<sup>1</sup>.
- Les courbes de masquage évoluent elles aussi de manière non linéaire en fonction de l'intensité du signal masquant. Elles sont également différentes selon que le son masquant est tonal (sinus)

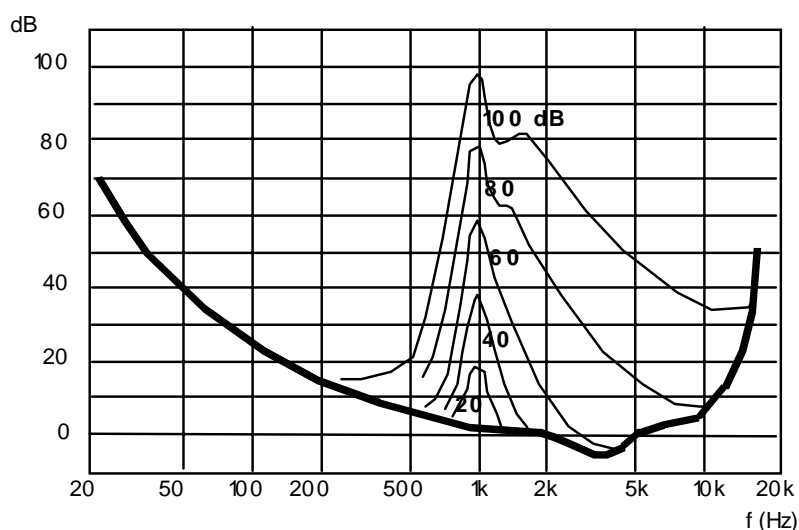


Fig. 2. Courbes de masquage auditif dû à une sinusoïde à 1000 Hz.

### 5.2.2 Codage en Sous-Bandes

Les codeurs audio en sous-bandes (SBC : Sub-Band Coders) utilisent tous une structure générale telle que celle de la Fig. 3. Le signal d'entrée est décomposé en sous-bandes (par filtrage ; cf. rappel ci-dessous, ou par FFT ; cf. plus loin), dont les échantillons sont quantifiés avec une précision imposée par un modèle psycho-acoustique. Celui-ci détermine, pour chaque sous-bande, le niveau de bruit maximum qui ne sera pas perçu, en vertu du phénomène de masquage auditif. Les échantillons quantifiés sont alors empaquetés (avec un entête, des informations liées à leur quantification, ainsi que des codes de vérification d'intégrité du paquet).

<sup>1</sup> Le masquage est également caractérisé par des enveloppes temporelles : un son peut en cacher un autre *avant* et *après* que le son masquant ne se manifeste.

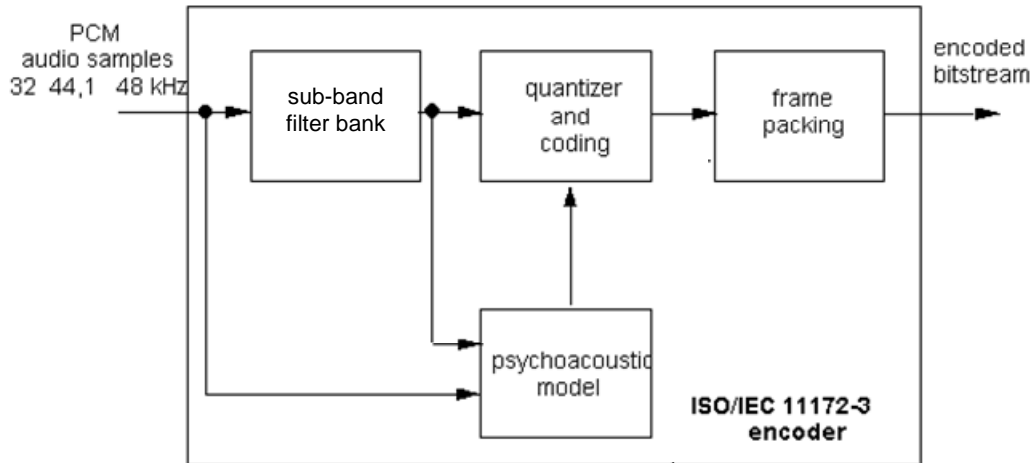


Fig. 3. Schéma de principe d'un codeur audio en sous-bandes (ici celui du standard MPEG1).

Le décodage est nettement plus simple, dans la mesure où il ne nécessite pas de modèle psycho-acoustique. Les paquets sont ouverts, et les sous-bandes sont regroupées en un signal audio de sortie.

Depuis la fin des années 80, un groupe d'experts de l'ISO (International Standardization Institute), appelé Motion Picture Experts Group (MPEG) a développé des standards de codage audio et vidéo, dont le MPEG Audio, qui est l'exemple le plus abouti de codeur audio en sous-bande.

### Rappel 1 : Décomposition d'un signal en sous-bandes

Le théorème de Shannon généralisé est mis à profit dans les systèmes dit *d'analyse en sous-bandes*, où un signal de départ est décomposé, par filtrage passe-bande, en plusieurs signaux à bande étroite, dont la somme fournit le signal de départ. Ces signaux sont alors échantillonnés séparément en respectant le théorème de Shannon généralisé. Ceci permet de transformer un flux d'échantillons de départ (large-bande, de largeur de bande  $f_M$  donnée) en  $N$  flux à bande étroite ( $f_M/N$ ) en conservant le débit total en nombre d'échantillons par seconde.

Ainsi par exemple, le signal  $x(t)$  dont le spectre  $A(f)$ , de largeur  $f_M$ , est donné à la Fig. 4 est décomposé en deux sous-bandes  $A_1(f)$  et  $A_2(f)$  de largeur  $f_M/2$ . Les deux signaux analogiques  $x_1(t)$  et  $x_2(t)$  correspondants sont tous deux échantillonnés à une fréquence d'échantillonnage  $f_e$  égale à  $f_M$ . L'échantillonnage de  $x_1(t)$  respecte donc le théorème de Shannon, et celui de  $x_2(t)$  respecte le théorème de Shannon généralisé. Les deux signaux numériques correspondants  $x_1(n)$  et  $x_2(n)$  portent bien la même information que le signal  $x(n)$  qui aurait été obtenu par échantillonnage direct de  $x(t)$  à une fréquence d'échantillonnage  $f_e$  égale à  $2f_M$  (pour respecter le théorème de Shannon sur l'échantillonnage de ce signal). Le nombre total d'échantillons utilisés pour stocker cette information dans  $x_1(n)$  et  $x_2(n)$  est bien identique à celui utilisé dans  $x(n)$ .

En pratique, la décomposition en sous-bandes est généralement réalisée directement dans le domaine numérique. Pour l'exemple ci-dessus, on échantillonne  $x(t)$  à  $f_e = 2f_M$ , puis on décompose par filtrage le spectre de  $x(n)$  en deux sous-bandes ( $[0, f_M/2]$ ,  $[f_M/2, f_M]$ ), et on décime les deux signaux numériques  $x_L(n)$  et  $x_H(n)$  ainsi obtenus par 2, pour obtenir  $x_1(n)$  et  $x_2(n)$  (Fig. 5). Le signal  $x_1(n)$  peut être vu comme une « approximation grossière », basse-fréquence, de  $x(n)$ , tandis que  $x_2(n)$  porte au contraire les détails haute-fréquence du signal. La même opération peut alors être répétée sur  $x_1(n)$  et  $x_2(n)$  séparément, ce qui conduit à une décomposition récursive en un nombre de sous-bandes de plus en plus élevé, et de

fréquence d'échantillonnage de plus en plus faible. Ce type de décomposition est appelée *décomposition en cascade* (ou pyramidale).

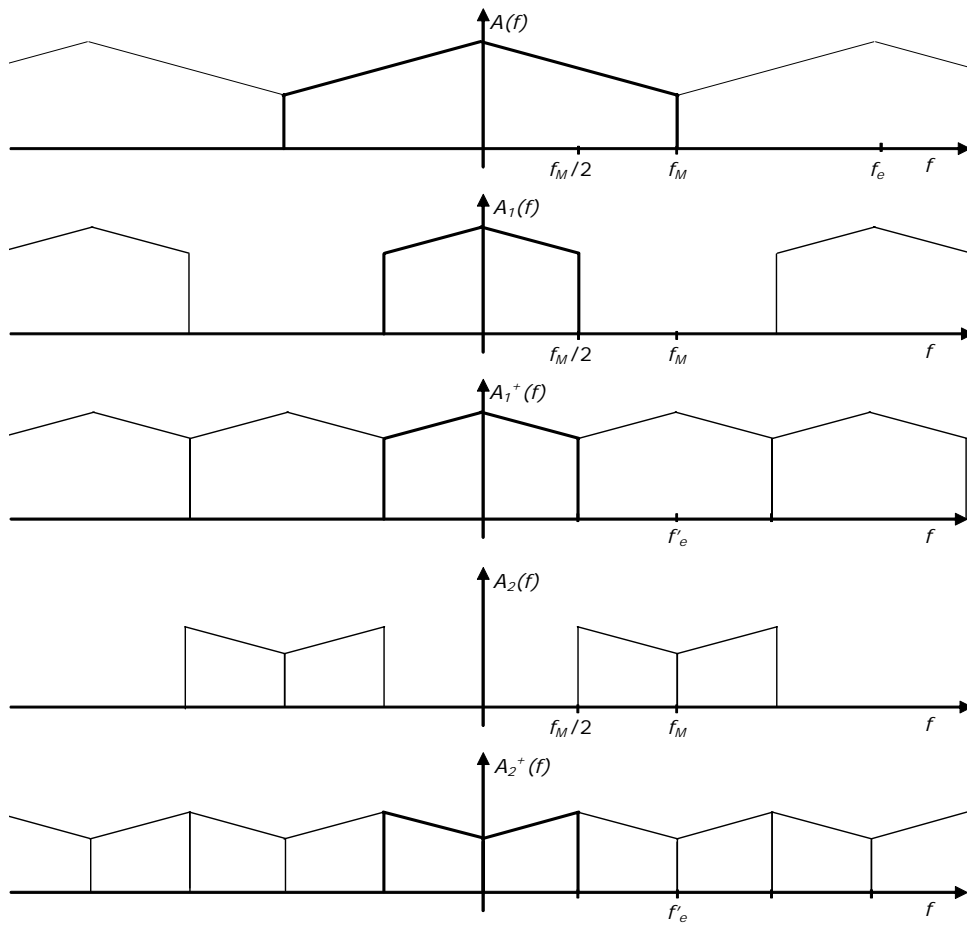


Fig. 4. Effet spectral de la décomposition d'un signal en 2 sous-bandes.

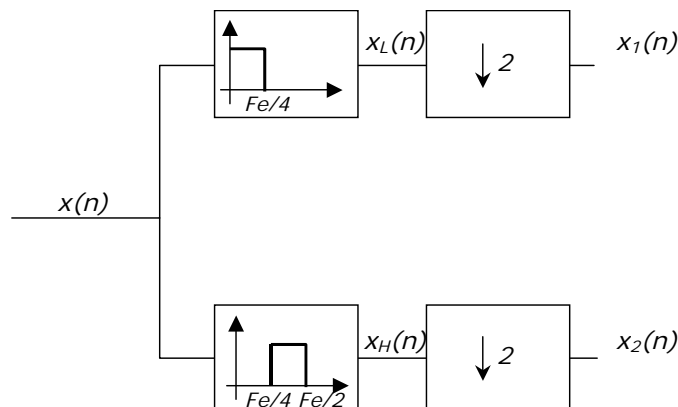


Fig. 5. Décomposition numérique en 2 sous-bandes.

## Rappel 2 : Décimation – Interpolation

Considérons un signal  $x_1(n)$  obtenu par échantillonnage d'un signal analogique  $x(t)$  à une fréquence d'échantillonnage valant  $F_e$ . En vertu de ce qui a été dit plus haut, le spectre utile du signal est limité à l'intervalle  $[0, F_e/2]$  ; un filtre de garde veille d'ailleurs normalement à ce que cet intervalle ne soit pas perturbé par du repliement spectral.

Sous-échantillonner ce signal à une fréquence  $k$  fois inférieure correspond en principe à ne retenir qu'un échantillon de  $x_1(n)$  sur  $k$  dans  $x_2(n)$ , à *décimer*  $x_1(n)$  par  $k$  (Fig. 6) :

$$x_2(n) = x_1(kn) \quad (1)$$

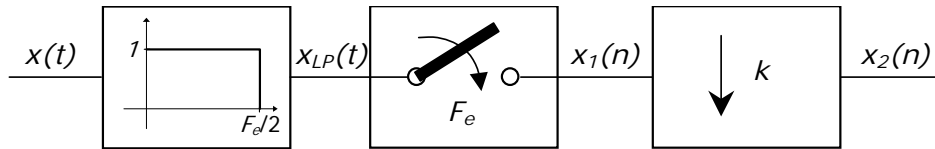


Fig. 6 Sous-échantillonnage par décimation brutale d'un signal numérique  $x_1(n)$

Ceci pose cependant un problème important : le résultat de l'échantillonnage de  $x(t)$  à  $F_e$  suivi d'un second échantillonnage à  $F_e/k$  est évidemment équivalent à un échantillonnage direct à  $F_e/k$ . Or, comme le filtre de garde a été prévu pour un échantillonnage à  $F_e$ , la décimation par  $k$  introduit, au niveau de la TFTD de  $x_2(n)$ , un repliement spectral des composantes de  $x_1(n)$  situées entre  $F_e/2k$  et  $F_e/2$ . Il est donc nécessaire de faire précéder la décimation d'un filtre *numérique* (au contraire du filtre de garde, analogique) passe-bas jouant le rôle d'un « adaptateur de filtre de garde », de fréquence de coupure  $F_e/2k$  (Fig. 7)

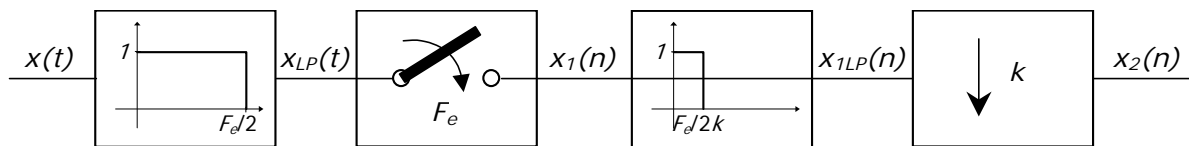


Fig. 7 Sous-échantillonnage par filtrage et décimation

Pour sur-échantillonner (au sens du mot anglais « up-sampling ») le signal  $x_1(n)$  à une fréquence  $k$  fois supérieure, il faut calculer  $k-1$  échantillons intermédiaires entre deux échantillons connus de  $x_1(n)$ . Ce calcul est possible : en vertu du théorème de Shannon, il est même possible de reconstituer complètement  $x_{LP}(t)$  en utilisant l'interpolateur idéal.

Le calcul des  $k-1$  échantillons intermédiaires de  $x_{LP}(t)$  se fait en pratique de la manière suivante : on commence par interpoler de façon brutale en insérant  $k-1$  échantillons nuls aux endroits requis :

$$\begin{aligned} x_2(n) &= x_1(n/k) \quad (n \text{ multiple de } k) \\ x_2(n) &= 0 \quad (n \text{ non multiple de } k) \end{aligned} \quad (2)$$

Cette opération ne change en rien la TFTD du signal interpolé  $x_2(n)$ , qui reste périodique de période  $F_e$ . Par contre, le spectre utile de ce signal s'étend maintenant de 0 à  $kF_e/2$ . Ce spectre fait donc apparaître des composantes de  $x_1(n)$  qui ne sont pas présentes dans  $x_{LP}(t)$  (Fig. 8, où l'on a expressément représenté les TFTD en fréquence vraie, non normalisée).

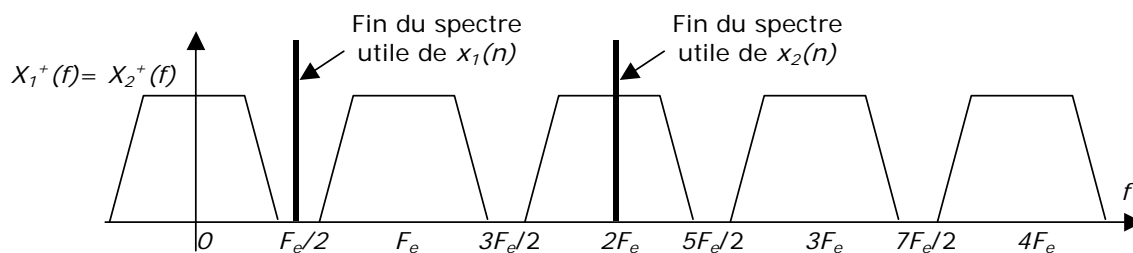


Fig. 8 Effet spectral de l'interpolation par insertion de zéros (exemple :  $k=4$ )

Ces composantes perturbatrices peuvent être éliminées en faisant suivre l'interpolation par un filtre passe-bas *numérique* de fréquence de coupure égale à  $F_e/2$ .

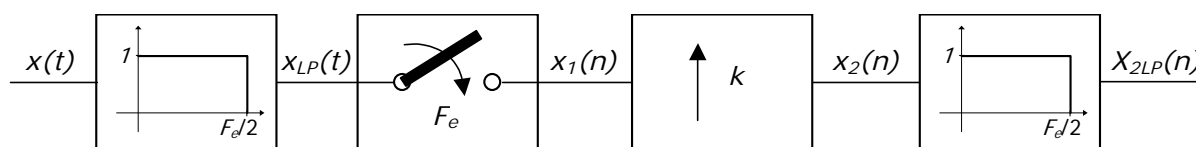


Fig. 9 Sur-échantillonnage par interpolation et filtrage

On notera que les filtres numériques utilisés pour sous- et sur-échantillonner par un facteur  $k$  sont en réalité identiques. En effet, leur fréquence de coupure *normalisée* (rapport de la fréquence de coupure à la fréquence d'échantillonnage des signaux d'entrés et de sortie) vaut dans les deux cas  $1/2k$ .

### 5.2.3 MPEG-1 Audio

La norme MPEG-1 Audio définit en réalité 3 schémas (on parle de *niveau* : layer I, layer II, et layer III), dont chacun possède son propre codeur SBC, son propre modèle psychoacoustique, et son propre quantificateur. Le niveau 3 est le plus efficace en termes de compression, mais aussi le plus complexe.

<b>1:4</b>	by <b>Layer 1</b> (corresponds to 384 kbps for a stereo signal),
<b>1:6...1:8</b>	by <b>Layer 2</b> (corresponds to 256..192 kbps for a stereo signal),
<b>1:10...1:12</b>	by <b>Layer 3</b> (corresponds to 128..112 kbps for a stereo signal),

Table 1. Taux de compression par niveau MPEG1. Les débits sont à comparer à  $16 \text{ bits} \times 44100 \times 2 = 1.4 \text{ Mbps}$  correspondant au stockage d'un signal stéréo sur un CD audio.

Le **codeur MPEG1 de niveau I** est basé sur une décomposition du signal (par tranches de 384 échantillons) en 32 en sous-bandes d'égales largeur en fréquence. Le modèle psychoacoustique est calculé sur une FFT du signal à 512 points. On y détecte les pics correspondant à des composantes tonales ou non-tonales, qui génèrent des courbes de masquage propres, et on en déduit une

courbe de masquage globale, ainsi que des valeurs seuils pour chacune des bandes de fréquences du banc de filtres. Pour chaque sous-bande, le quantificateur mesure et quantifie sur 6 bits un *facteur d'échelle* correspondant à l'amplitude maximale des échantillons (par tranche), normalise les échantillons par rapport à ce facteur d'échelle, puis détermine le nombre de bits nécessaire au échantillons de façon que la puissance du bruit de quantification soit inférieure au seuil de masquage.

Le **codeur MPEG1 de niveau II** est basé sur le même filtre en sous-bandes, mais son modèle psychoacoustique est calculé sur une FFT à 1024 points. La quantification est faite sur des tranches trois fois plus longues.

Le **codeur MPEG1 de niveau III** (mp3) est plus complexe en termes d'implémentation (mais non en termes de principe). Il complète le banc de filtres en sous-bandes par une *transformée en cosinus discrète* (variation de la FFT où l'on décompose les signaux sur des cosinusoides de fréquences harmoniques plutôt que sur des exponentielles imaginaires de fréquences harmoniques). Le codage est dynamique, allouant plus de bits aux tranches qui le nécessitent, ces bits supplémentaires étant stockés dans les tranches plus simples à quantifier (on parle de *réservoir de bits*).

Pour une meilleure compréhension du principe général, on consultera l'excellente toolbox de démonstration MATLAB réalisé par F. Petitcolas (alors thésard à l'Université de Cambridge ; il est aujourd'hui expert en watermarking électronique chez Microsoft Research) :

<http://www.petitcolas.net/fabien/software/mpeg/index.html>

On se référera également à (Pan, 1995).

### 5.3 Décomposition/Recomposition par filtrage en sous-bandes

Cette étape est décisive : elle nous permettra de mieux comprendre le principe du sous-échantillonnage (downsampling) et du sur-échantillonnage (upsampling).

Pour ne pas alourdir inutilement les calculs, nous travaillerons avec un signal échantillonné à 8 kHz sur 16 bits et en mono. Nous nous limiterons également à une découpe en 8 sous-bandes. Aucune quantification ne sera effectuée.

1. On dessinera le schéma du système de décomposition en 8 sous-bandes, en spécifiant les fréquences d'échantillonnages de tous les signaux intermédiaires.
2. On imaginera le schéma d'un bloc de recombposition de 2 signaux en sous-bande en un seul (l'opération duale de celle de la Fig. 5).
3. On en déduira le schéma du système de recombposition des 8 sous-bandes, en spécifiant les fréquences d'échantillonnages de tous les signaux intermédiaires.
4. Les spécifications du filtre passe-bas seront ajustés de façon que l'erreur de reconstruction soit inaudible. On testera tout d'abord la décomposition/recombposition sur un signal simple (un *chirp* : une sinusoïde de 4 secondes, de fréquence allant de 0 à 4000 Hz).

- a. On fera un test avec un filtre à phase linéaire
- b. On répétera ce test avec un filtre récursif, et on comparera les résultats

On en profitera systématiquement pour :

- Calculer et afficher la réponse en fréquence des filtres utilisés
- Afficher et écouter les signaux filtrés, avant et après décimation
- Afficher le spectrogramme du signal original et du signal reconstruit
- Calculer et afficher l'erreur de reconstruction, et en déduire une estimation du rapport signal à bruit (moyen sur tout le signal)
- Estimer la charge de calcul due aux bancs de filtres. Cette charge est-elle indispensable ? Comment pourrait-on en conséquence la diminuer ?

On utilisera ensuite un signal plus réaliste fourni : musique.wav (les premières mesures de la 9<sup>ème</sup> symphonie de Beethoven) et on vérifiera qu'il passe sans distorsion.

Fonctions utiles : wavread, remez, filter, sound, specgram.

NB : les fonctions decimate, interp, et resample disponibles dans MATLAB ne font qu'une partie du travail ; elles ne permettent pas d'obtenir les bandes en HF. On ne les utilisera donc pas ici (sauf éventuellement pour voir comment elles fonctionnent).

## 5.4 Décomposition/Recomposition en sous-bandes par Transformée de Fourier Discrète

Nous avons toujours considéré la transformée de Fourier comme une opération de décomposition d'un signal sur des exponentielles imaginaires (le résultat de cette transformation étant directement interprété comme les coefficients de cette décomposition). Cette interprétation conduit à considérer que les coefficients résultant de la décomposition (c.-à-d., pour un signal numérique, la TFD elle-même) sont sans dimension. Ce projet est l'occasion de voir qu'il est également possible de considérer (et donc d'utiliser !) la FFT à court-terme comme un banc de filtres dont les échantillons de sortie sont directement les coefficients de la FFT (et possèdent donc la même dimension que le signal d'entrée).

1. Considérons la *transformée de Fourier à court-terme* (STFT : short-term Fourier Transform ; Section 4.3.5 du cours), obtenue par calcul de FFT sur des tranches successives de signal :

$$X_k(i) = \sum_{n=0}^{N-1} w(n)x(iS+n)e^{-j\frac{2\pi}{N}kn} \quad (3)$$

où  $w(n)$  est une fenêtre de pondération éventuelle (Hanning, Hamming, Blackman, etc.) et  $S$  est le nombre d'échantillons de décalage entre tranches successives.

Montrer que l'équation (3) peut être interprétée comme une décomposition en  $N$  sous-bandes combinée à une décimation  $1/S$ , dont le  $k^{\text{ème}}$  signal de sortie de sortie est donné par  $\{X_k(0), X_k(1), X_k(2), \dots\}$ .

2. Quelle est la réponse en fréquence des filtres correspondant à cette décomposition ? Afficher leur réponse en fréquence sous Matlab. Ces filtres sont-ils parfaits ?
3. Réaliser une décomposition/recomposition en 8 sous-bandes (sur le modèle de la Fig. 9, mais sans modification des sous-bandes) sur les 2 signaux utilisés à la Section 5.3. Visualiser les signaux en sous-bandes (signal et spectre) et en écouter la partie réelle : il s'agit bien de signaux !

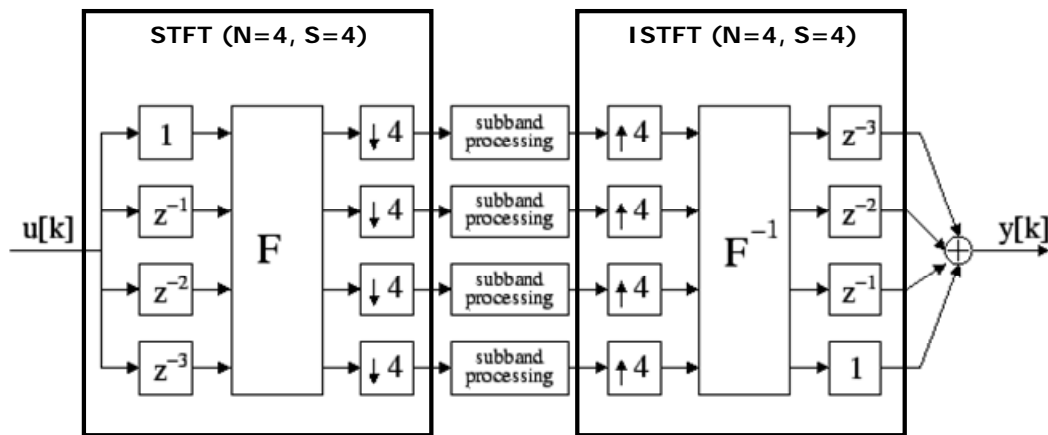


Fig. 10 Analyse en sous-bande par STFT-ISTFT (exemple pour  $N=4$  et  $S=4$ ).  
(D'après <http://www.esat.kuleuven.ac.be/~rombouts/dspII/oefenzittingen2001-2002/opgave3/>)

4. Quel est la RSB de cette décomposition/recomposition ? Peut-on en déduire que le système est parfait ?

## 5.5 Bibliographie et ressources WEB

Pan, D. (1995). "A tutorial on MPEG audio compression", *IEEE multimedia magazine*, vol. 2, no. 2, pp. 60-74.

Brandenburg, K., Popp, H. (2000). "An introduction to MPEG layer 3", *EBU Technical review*, June 2000.

<http://www.chiariglione.org/mpeg/> : Home page officielle du MPEG Group, fondé par Leonardo Chiariglione.

<http://inventors.about.com/od/mstartinventions/a/MPThree.htm> : L'histoire de l'invention du MP3

<http://www.mpeg.org> : Portail MPEG pointant vers de nombreuses ressources.

<http://www.petitcolas.net/fabien/software/mpeg/index.html> Excellente toolbox MATLAB sur le principe du codage mp3.

<http://www.otolith.com/otolith/olt/sbc.html> : Otolith, page web consacrée au traitement du signal audio et musical, maintenue par Wil Howitt.

<http://iphilgood.chez.tiscali.fr/Codage/CodageAudio.htm> : cours en ligne sur le codage audio.