



RÉALISATION ET ÉVALUATION DE CODAGES
NUMÉRIQUES DU SON DE HAUTE QUALITÉ
POUR LA RADIODIFFUSION
PHASE I

CENTRE DE RECHERCHES SUR LES COMMUNICATIONS

FACULTÉ DES SCIENCES APPLIQUÉES

UNIVERSITÉ DE SHERBROOKE

TÉL.: 819-821-7141

TÉLEX 05-836149

SHERBROOKE, QUÉBEC, CANADA, J1K 2R1

IC

LKC
QA
268
.R43
1988
c.2

QA
268
R43
1988
S-Gen

CENTRE DE RECHERCHES SUR LES COMMUNICATIONS

Faculté des sciences appliquées

Département de génie électrique

Université de Sherbrooke

RÉALISATION ET ÉVALUATION DE CODAGES

NUMÉRIQUES DU SON DE HAUTE QUALITÉ

POUR LA RADIODIFFUSION

PHASE I

Rapport final

**Ministère des Communications du Canada
Ottawa - Canada**

dans le cadre du

programme de Centres d'excellence

Contrat no. 36000-7-0219/01-SS

Industry Canada
Library - Queen

OCT 15 2013

Industrie Canada
Bibliothèque - Queen

TABLE DES MATIERES

INTRODUCTION.....	1
A TECHNIQUE DE QUANTIFICATION VECTORIELLE PROPOSÉE.....	3
1. Segmentation variable du signal.....	3
1.1 Algorithme numérique de segmentation variable.....	5
1.2 Compression-expansion des sous-blocs.....	13
1.2.1 Algorithme de compression de temps.....	14
1.2.2 Algorithme d'expansion de temps.....	16
1.2.3 Résultats de compression.....	17
1.2.4 Extension de l'analyse par sous-blocs normalisés.....	25
1.3 Estimation des résultats de la compression expansion - taux de réduction.....	28
2. Quantification vectorielle.....	30
2.1 Principe de la k-moyenne.....	30
2.2 Quantification des vecteurs.....	33
2.3 Résultats préliminaires.....	39
2.3.1 Résultats expérimentaux.....	40
2.3.2 Résultats expérimentaux (notes synthétiques).....	48
2.3.4 Comparaison des dictionnaires, musique naturelle notes synthétiques.....	55
3. Conclusion.....	58
B CODAGE DIFFÉRENTIEL "QUASI-INSTANTANÉ".....	59
1. Fonctionnement du codeur - musique naturelle.....	60
2. Fonctionnement du codeur - notes synthétiques.....	65
2.1 Facteur de prédiction "a".....	66
2.2 Rapports Signal sur Bruit notes et musique.....	72
ANNEXE.....	76
FILTRE DE SUR-ÉCHANTILLONNAGE.....	76

INTRODUCTION

Pour les besoins de réduction de débit numérique des transmissions, on développe plusieurs techniques de codage numérique. La référence actuelle de qualité peut être considérée comme étant celle de l'enregistrement/lecture du disque compact numérique.

On lui associe une simple "quantification" linéaire à 16 bits. On ne définit pas, à proprement parler, de codage dans ce cas et le débit élevé (44 kHz de fréquence d'échantillonnage) n'est pas admissible pour les transmissions sur réseau numérique ou par satellite (SRS: Service de Radiodiffusion par Satellite - DBS en anglais "Digital Broadcast ^{Direct} ~~System~~ ^{Satellite} ~~System~~) ainsi que pour le "son" de haute qualité requis dans les nouveaux standards de télévision.

Une première étape de "codage" est franchie dans les techniques de "compression" logarithmique instantanée, ou de "compression" par bloc (quasi-instantanée) actuellement en ^{opérations} cours de développement pour les applications visées.

Les techniques Loi A, et NICAM 3 ont été intégralement simulées. Elles réduisent le débit, par codage, à seulement 11 bits (Loi A) ou 10,1 bits en moyenne par échantillon (Loi quasi-instantanée NICAM 3). On ne compte pas ici les bits nécessaires à la transmission et à la ^{en la Pr} correction d'erreurs qui contribuent au débit de référence de 12 bits par échantillon avec une ^{bits par échantillon} fréquence d'échantillonnage de 32 kHz. Il faut cependant réduire encore plus ce débit.

Le codage Delta (ADM) présente une alternative intéressante de codage en raison surtout d'une extrême simplicité de conception du récepteur. Les Laboratoires DOLBY ont présenté une réalisation d'un modulateur, de performances comparables aux lois de quantification précédentes, pour ^{des} un débit ^{de transmission} similaire (330 kbits/s).

Le codage en sous-bandes amène une réduction par 2 (192 kbit/s au lieu de 384 kbit/s) avec très peu de dégradation sensible. On peut espérer atteindre un rapport 3 (valeur visée 128 kbit/s).

réponse

Le codage "par transformée", considérablement plus complexe, donne déjà de très bons résultats à des débits inférieurs à 100 kbit/s (valeur visée 64 kbit/s).

Def.

finières
 Les deux techniques concernent un traitement "spectral" du signal et tentent d'utiliser au mieux les caractéristiques de perception acoustique de l'oreille. *Intégrés les récepteurs*
avant tendance à être complexes. Dans les cas précédents, la complexité était *complexe*
concentrée sur encodage seul
 La quantification vectorielle, déjà largement utilisée dans le domaine du traitement de la parole, pourrait contribuer à l'obtention d'un débit réduit, *(cette quantification vectorielle est faite en*
 en groupant par bloc ou "vecteur" un certain nombre d'échantillons consécutifs. *avec une complexité de calcul relativement faible*

Chaque vecteur ainsi formé est comparé à un ensemble de "vecteur de base" appelé aussi "vecteurs centres". Ce nombre de vecteurs "centres" étant très réduit, par choix initial, seul un numéro d'identification, sous forme de quelques bits seulement, est transmis (au lieu d'un bloc d'échantillons).

Exemple:

- Dictionnaire de vecteurs "centres" de 256 représentants,
- Vecteurs (ou blocs) de 32 échantillons consécutifs,
- Transmettre le plus proche représentant nécessite 8 bits (comptage de 1 à 256) au lieu des 32 fois 16 bits (ou 11 bits si compression logarithmique).

Cet exemple est évidemment hypothétique. Une compression $R = (32 \times 11)/8 = 44!$ est fantaisiste pour les cas pratiques.

A: TECHNIQUE DE QUANTIFICATION VECTORIELLE PROPOSEE

La segmentation du signal en blocs (ou vecteurs) de durée (longueur) fixe pose un dilemme:

- Si le vecteur contient beaucoup d'échantillons à coder (plus de 8...): les formes d'ondes obtenues par blocs de 8 ou plus sont extrêmement variées; il faudra donc un dictionnaire de base extrêmement étendu, contenant un "ensemble" de toutes les formes d'ondes possibles. D'où bien sûr aussi un problème de recherche assez lourd et long en temps de calcul.
- Si le vecteur contient très peu d'échantillons, le dictionnaire sera plus restreint, mais le nombre d'informations à transmettre dans le même temps sera bien plus élevé...

Fixe est plus simple *Résultats de P. Mabilleson* *FIXE*
mais limitée en termes de compression

On peut envisager un compromis en recherchant, sur le signal de base (musique ou même parole) une forme d'onde "prototype", qui est la plus fréquemment possible identifiable et telle que, d'un bloc de signal à un autre, la dispersion obtenue soit faible en prenant pour référence cette forme prototype. ~~Il est donc exclu d'effectuer une~~ ^{La} segmentation par intervalles fixes sur le signal d'origine. *et sans mise de côté pour se concentrer sur la segmentation variable*

1. Segmentation variable du signal

On propose de découper le signal en blocs de durée variable par une construction géométrique simple. Soit, le dernier échantillon supposé connu au récepteur (donc

éventuellement codé et décodé par un récepteur aussi localisé au point d'émission donc de codage):

- A partir de ce point, on recherche, sur la forme d'onde du signal le "point de tangence" en menant la droite de tangence. La figure 1 montre ce principe sur 2 points consécutifs, avec une courbe continue. L'algorithme numérique qui s'appliquera sur le signal échantillonné est alors extrêmement simple à élaborer. Il est évident, sur la figure 1, que l'écart E , entre le point de départ (quelconque à priori) et le point de tangence est de signe strictement constant dans tout l'intervalle.

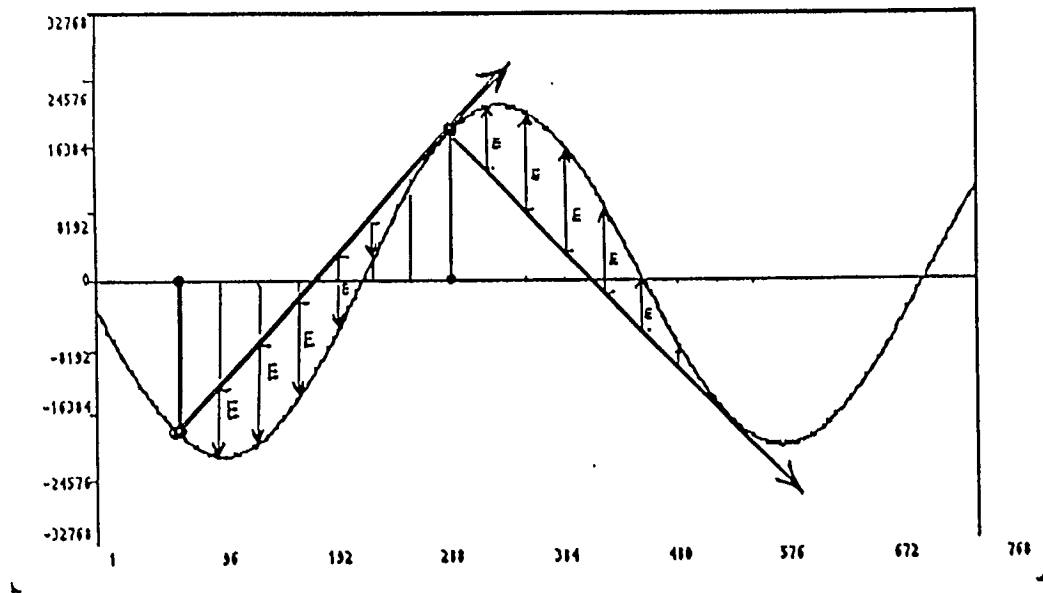


Figure1: Principe de segmentation.

Bien sûr, chaque courbe ainsi segmentée est définie par:

- son amplitude maximale,
- son signe;
- sa durée (longueur).

On choisit de subdiviser le signal en deux:

- un ensemble de sous-blocs constitué des courbes formées par les "Ecart" entre le signal original et la "tangente" ou droite d'interpolation linéaire;
- un ensemble d'échantillons isolés constitué des échantillons "de départ" de chaque droite de tangence. On les appellera les "Supports" du signal.

Ainsi, et toujours à l'avenir, à partir du signal échantillonné à 32 kHz, on construira les deux ensembles de signaux.

1.1 Algorithme numérique de segmentation variable

L'algorithme de segmentation est basé sur la simple détection du changement de signe du signal d'écart, entre l'original et l'interpolation linéaire.

~~Le de la pente relative (par rapport au segment d'interpolation)~~

Cette dernière peut être faite, au choix dans les simulations, avec un calcul en "réel" ou un calcul en "entier" Dans ce dernier cas, en raison des arrondis de calcul, une erreur est inévitable. Elle est cependant réinjectée, au codeur, sur le signal des sous-blocs pour les calculs suivants:

Handwritten notes and signatures:
 P. Gagnon
 J. Gagnon
 J. Gagnon
 J. Gagnon
 J. Gagnon

Le récepteur reçoit directement les échantillons de base du signal (ou "supports") correspondant d'ailleurs à un sous-échantillonnage du signal. Il détermine l'interpolation linéaire entre ceux-ci, soit en "entier" soit en "réel". A ce signal interpolé, le récepteur ajoutera ensuite le signal de "sous-bloc" affecté de l'amplitude quantifiée et de la longueur adéquate: ces informations sont fournies par le codeur et sont transmises en ligne.

- L'algorithme de segmentation variable est strictement appliqué aux seuls échantillons d'origine du signal échantillonné à 32 kHz. En simulation, utilisant des fichiers sur-échantillonnés, cette opération est réalisée par une instruction (modulo...).

On a fait l'hypothèse que le signal d'origine était toujours fourni à 32 kHz et que le signal codé/transmis/décodé serait aussi à 32 kHz.

Le fait de limiter, strictement aux échantillons du signal, l'analyse de segmentation pose quelques problèmes dans le cas des hautes fréquences, où le sous-bloc concerné risque de ne contenir qu'un seul échantillon, et deux périodes d'échantillonnage.

Les figures suivantes illustrent ces faits: la figure 2 montre un extrait de harpe et la présence de hautes fréquences. La figure 3 correspondant à la même séquence montre que la segmentation ne donne qu'une information très limitée (1 échantillon ou 2 par sous-bloc). Il serait alors difficile d'analyser ces formes d'onde et d'envisager de les comparer à des vecteurs de base contenus dans un dictionnaire.

Par contre l'idée de départ présentée sur la figure 1 incite à envisager:

- Un sur-échantillonnage du signal, de façon à définir avec plus de précision la forme d'onde (idéalement analogique), c'est-à-dire une introduction d'information complémentaire sur la

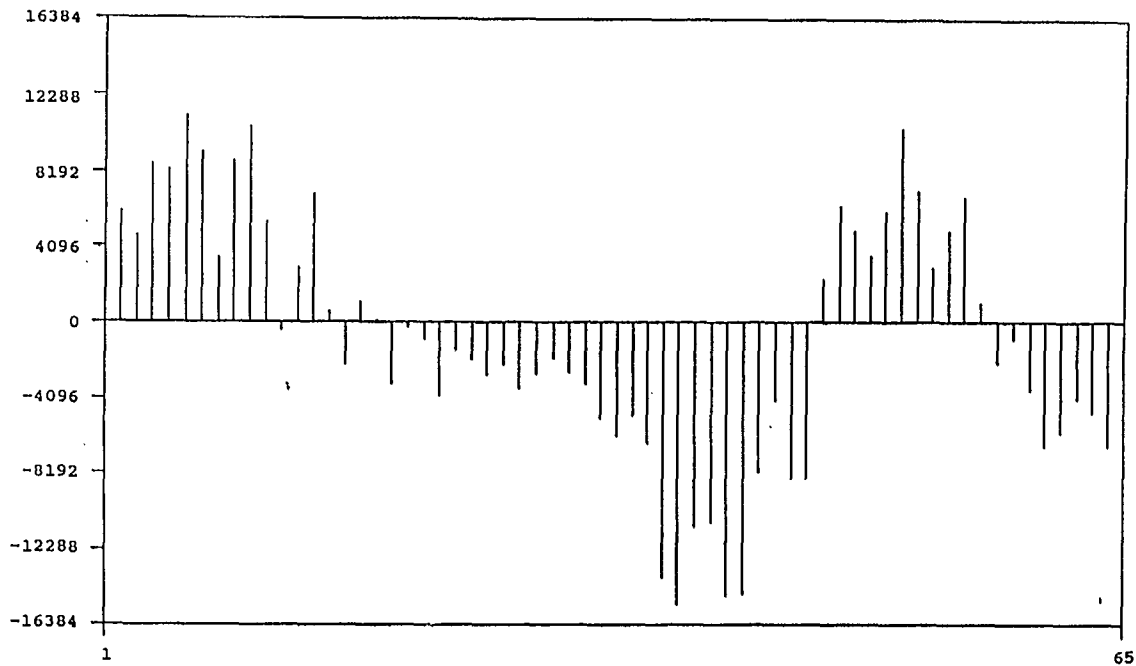


Figure 2: Extrait de harpe à 32 kHz.

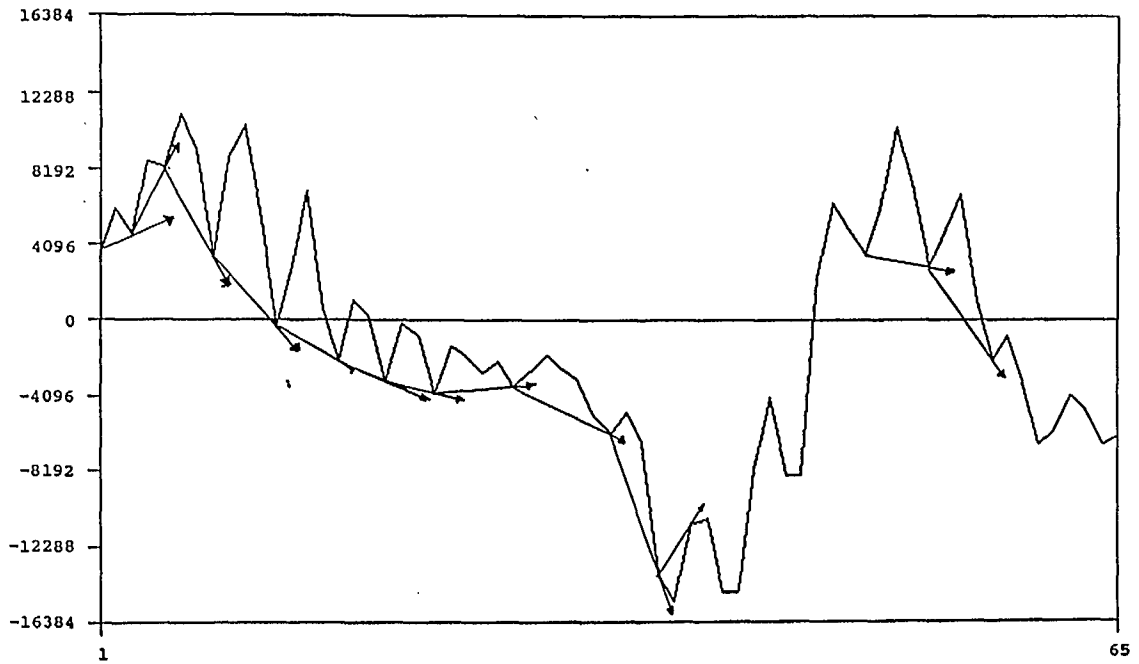


Figure 3: Principe de segmentation appliqué à cet extrait de harpe sur les échantillons à 32 kHz.

configuration géométrique de l'onde. Dès lors, il devient envisageable de comparer entre elles ces formes d'ondes (segmentées) ainsi obtenues.

La figure 4 montre la même séquence sur-échantillonnée par un facteur $R=32$. Les simulations prévoient tous les cas possibles $R \leq 32$. On se limitera aux puissances de 2 (4,8,16 et 32).

La figure 5 illustre un résultat de segmentation obtenu sur le signal sur-échantillonné mais toujours en limitant les frontières de segmentation aux seuls échantillons d'origine à 32 kHz . On notera que dans ce cas, les points dits de tangence sur le signal à 32 kHz ne correspondent plus exactement à des "points de tangence" sur le signal sur-échantillonné.

- Les sous-blocs n'en sont pas moins construits à partir de ce signal sur-échantillonné;
- Les "supports", rappelons-le, ne sont constitués qu'à partir d'échantillons d'origine (32 kHz) eux-mêmes sous-échantillonnés, selon bien sûr la longueur du "sous-bloc" correspondant, ce qui définit un rythme variable de sous-échantillonnage.

Les figures 6 et 7 montrent, sur quelques échantillons (8 blocs de 256 à la fréquence de sur-échantillonnage soit 32 kHz x 32, correspondant à seulement 64 échantillons d'origine à 32 kHz):

- les courbes (sur-échantillonnées) de "sous-blocs" (figure 6);
- les "supports" (sous-échantillonnés) du signal, (figure 7).

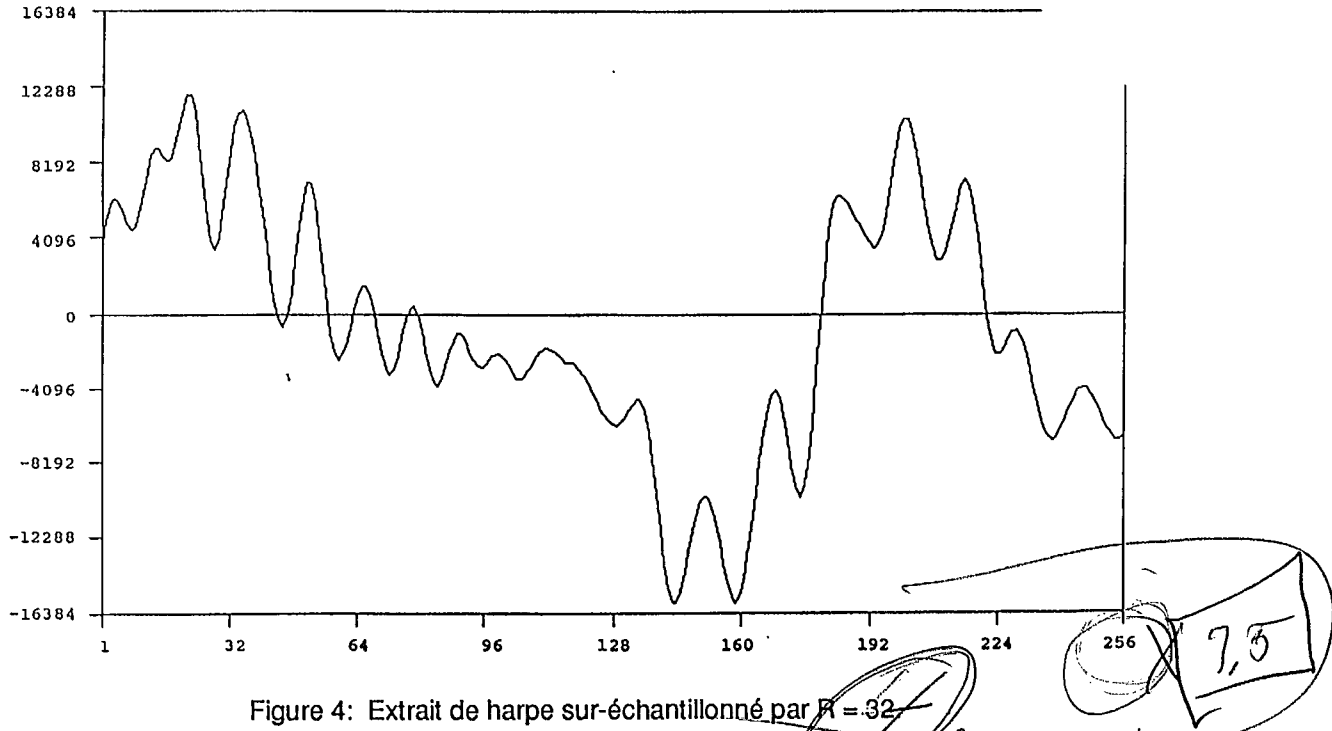


Figure 4: Extrait de harpe sur-échantillonné par $N = 32$

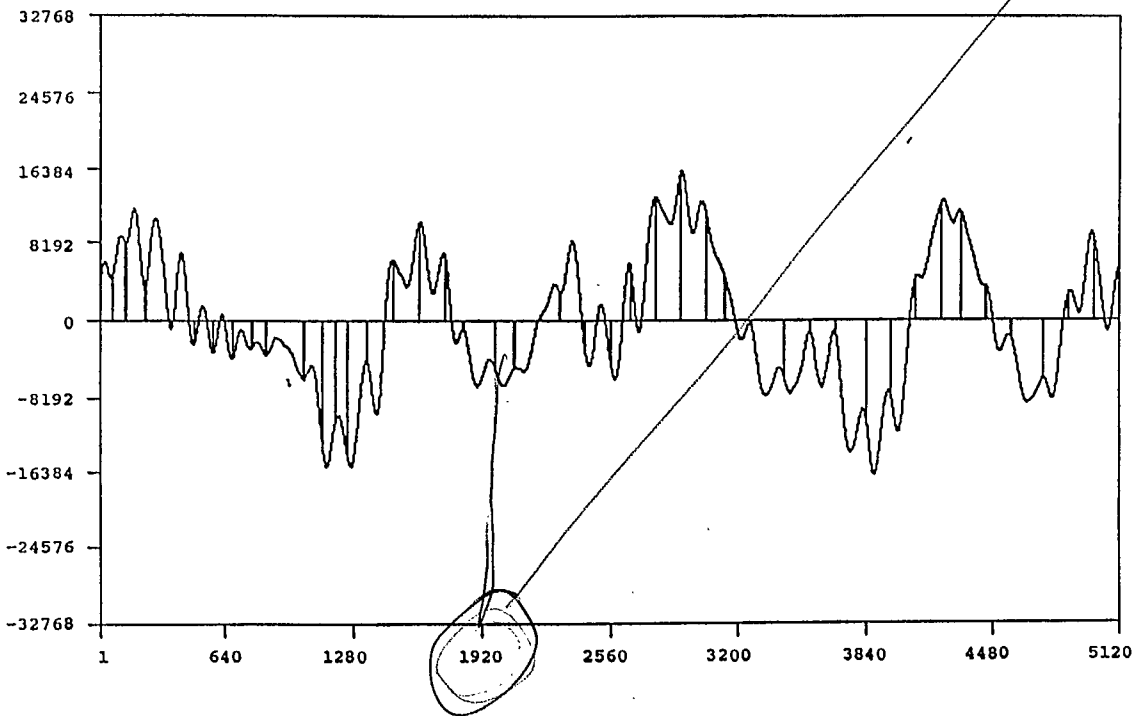


Figure 5: Principe de segmentation appliqué à cet extrait de harpe.

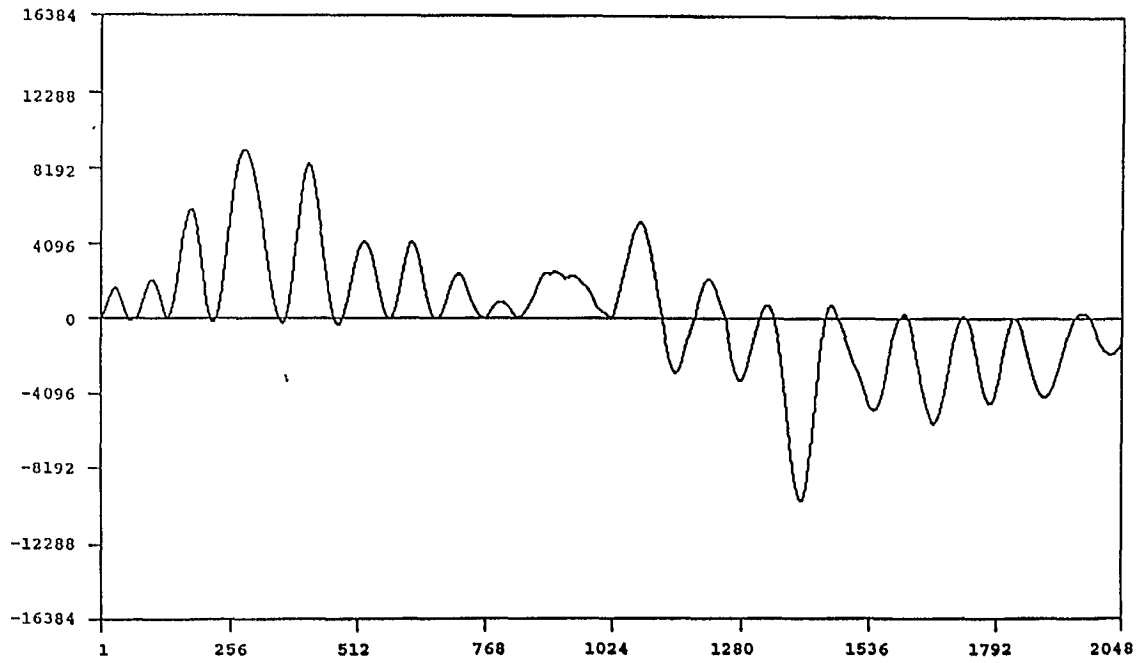


Figure 6: Signaux de "sous-blocs" après segmentation à rythme variable (et formés par le signal sur-échantillonné ...)

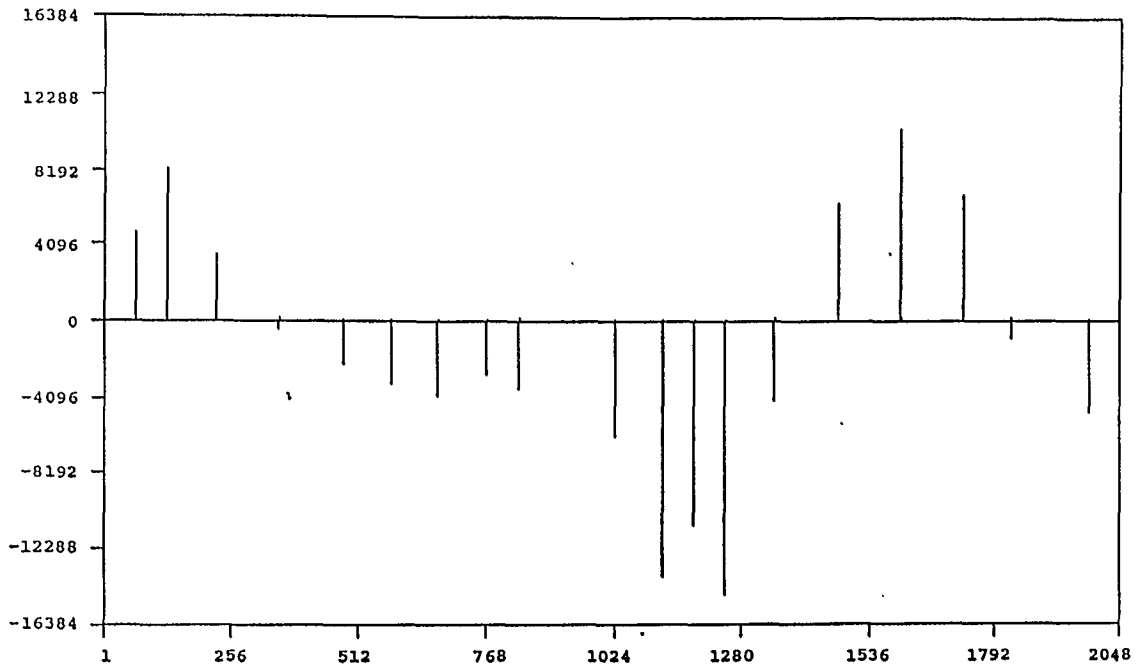
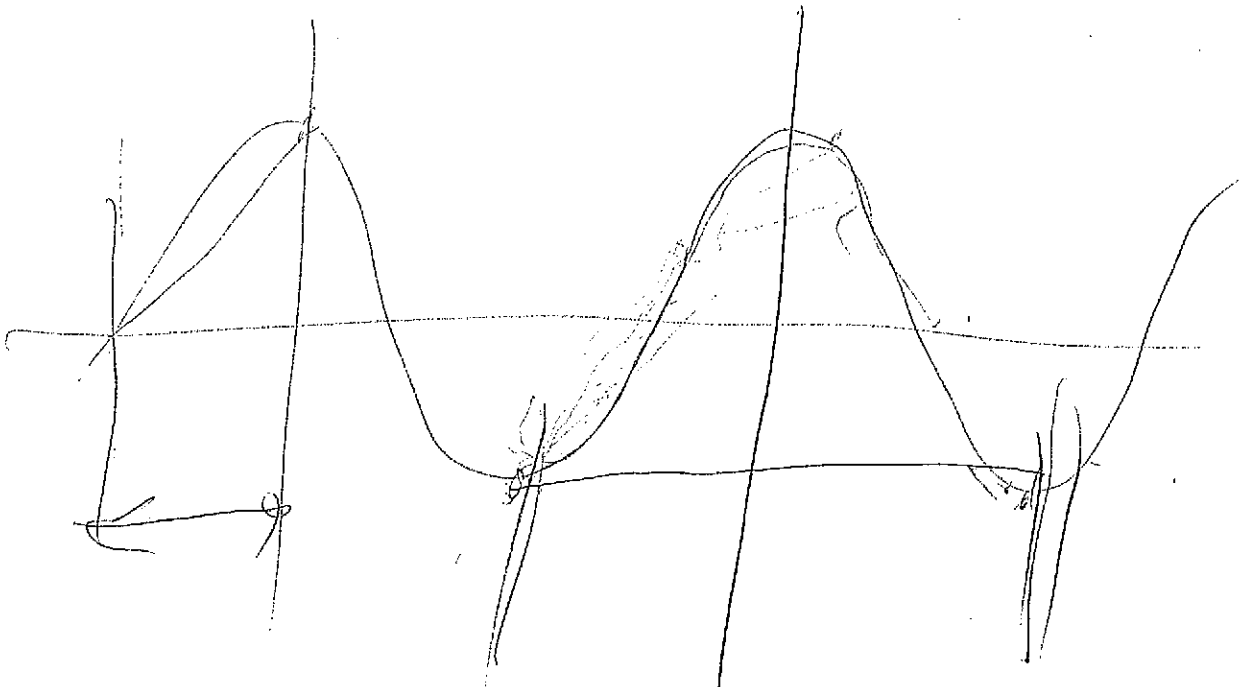


Figure 7: "Supports" du signal extraits par segmentation (et formés seulement d'échantillons d'origine à 32 kHz ...)

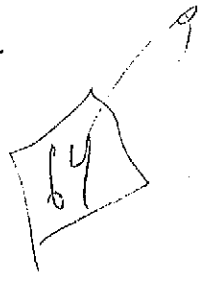


$3\pi/2$



$1/1$

$3\pi/2$



On note immédiatement la "relative uniformité" de formes d'ondes obtenues dans les sous-blocs.

Il conviendra d'extraire une ou des formes d'ondes prototypes, de cet ensemble.

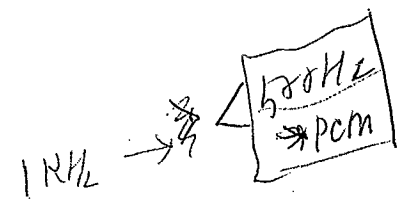
On vérifie d'ailleurs, qu'avec n'importe quel autre fichier musical, l'ensemble obtenu conserve des formes générales très voisines. Des résultats "moyens" sont présentés dans la suite du texte.

Complément à l'algorithme

En cours de simulation, l'algorithme a dû être sensiblement modifié et complété par un autre test d'analyse et en imposant des limites:

- En premier lieu, il a fallu définir une fenêtre de largeurs minimale et maximale. Toujours à partir du signal à 32 kHz, on doit limiter l'analyse à:

- Fenêtre minimale: 2 périodes d'échantillonnage, 15 kHz
- Fenêtre maximale: 32 périodes soit une milliseconde, notée F_{max} .



Il n'y a pas intérêt, et il est d'ailleurs impossible, compte tenu de la suite des traitements effectués, d'utiliser une fenêtre maximale supérieure au taux de sur-échantillonnage: R.

32 échantillons de base

$F_{max} \leq R$ en milliseconde 1 kHz

Le taux de sur échantillonnage ne peut pas être supérieur à 32

On choisira toujours des puissances de 2 pour R.

Expliquer

Block long

demande

meilleures précision

de bits + ...

- En second lieu, compte tenu de l'échantillonnage à 32 kHz "limitant l'information sur la forme d'onde analogique d'un point de vue géométrique", la recherche du point de tangence conduit parfois au cas de figure 8:

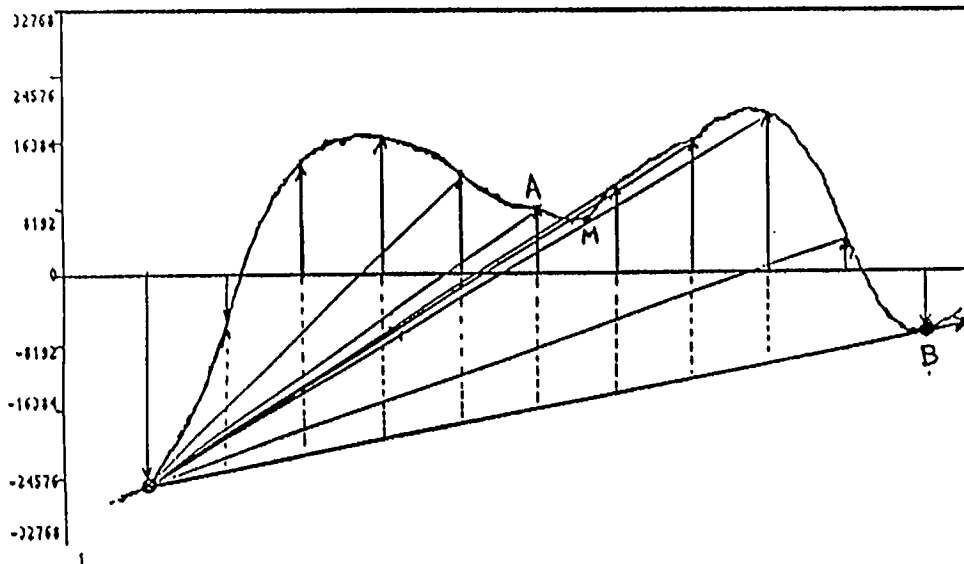


Figure 8: Cas particulier de segmentation.

à l'instant relative
ne change pas nécessairement de signe,
voir fig 6

Le premier changement de signe de l'écart n'est détecté qu'au point B et non en A pour des raisons de construction géométrique et de valeurs de "pentes" des droites. Le sous-bloc qui en résulte (et qui lui est sur-échantillonné - voir courbe en trait continu -) est une courbe "à 2 bosses" qui s'éloigne donc de la forme prototype recherchée.

L'algorithme a donc été complété pour une recherche de la variation de pente des droites d'interpolation. A partir du point A, la pente de la droite qui a d'abord augmenté puis diminué réaugmente à nouveau. Le programme revient donc au point A comme point limite du sous-bloc, et y prélève l'échantillon du signal comme étant le "support" correspondant.

La droite d'interpolation est recalculée point à point, au récepteur, directement à partir des valeurs reçues pour les points de support. Le récepteur présente aussi l'option de sous-échantillonnage (à la demande au terminal). De la sorte, le fichier "final" de sortie est sous-échantillonné par R, ce qui le ramène à 32 kHz. Il est directement comparable au fichier original.

En formant en séquence les 3 opérations mais sans aucun codage ni quantification intermédiaire i.e.

- **Sur-échantillonnage**

qui est l'opération schématisée sur la figure 4;

- **Décomposition**

en "sous-blocs" et en "supports" (avec interpolation linéaire) qui constitue l'ensemble présenté sur les figures 6 et 7;

- **Recomposition du fichier avec sous-échantillonnage**

que l'on retrouve, successivement, sur les figures 5 puis 3;

on mesure un RSB d'au moins 75 dB dans cette séquence d'exécution de 3 programmes quelle que soit la sélection musicale. L'erreur minime est strictement égale à ± 1 lorsqu'elle n'est pas nulle. En effet les calculs effectués (dans l'interpolation linéaire) sont arrondis "au plus proche entier" (nearest integer: instruction NINT), c'est ce qui explique ces légères différences.

1.2 Compression-expansion des sous-blocs

Les formes d'ondes extraites de la figure 6 bien que "ressemblantes" ne sont pas directement comparables. Il faut envisager une opération d'analyse destinée à les normaliser selon un format amplitude-longueur (ou temps). La normalisation en amplitude est aisée. Toutes ces formes n'ont qu'un seul maximum. On les normalisera donc par un facteur de gain (mémorisé

- et transmis au récepteur en information complémentaire -). On a choisi une valeur décimale 30000 pour les représentations graphiques entre autres.

1.2.1 Algorithme de compression de temps

Un même programme réalise simultanément:

- la normalisation d'amplitude,
- la normalisation de durée.

En effet le signal a d'abord été sur-échantillonné (par exemple $R=32$). Chaque "sous-bloc" contient N ($N \geq 2$ et $N \leq 32$) échantillons du signal d'origine (à 32 kHz). Chaque sous-bloc ainsi sur-échantillonné contient au total $N \times R$ échantillons (en toute rigueur il s'agit de périodes d'échantillonnage).

Il est donc réalisé une opération de sous-échantillonnage sur ces $N \times R$ échantillons du sous-bloc pour se ramener à un sous-bloc normalisé de R échantillons.

L'ensemble des opérations permet la séquence:

- N échantillons à 32 kHz puis:
- Sur-échantillonnage par R
- .. d'où $N \times R$ échantillons

ensuite

- Sous-échantillonnage par N
- .., d'où R échantillons.

On peut donc ramener, de cette façon toutes les courbes à un format unique de "courbe prototype", à R échantillons (alors que l'on est parti de N échantillons).

Chaque courbe "normalisée" est caractérisée par:

- amplitude maximale: 30000 (valeur arbitraire),
- longueur (durée R périodes d'échantillonnage).

La figure 9 présente des sous-blocs ainsi normalisés par blocs de 32 échantillons. Il est clair que la "ressemblance" des courbes permet d'envisager un traitement par quantification vectorielle.

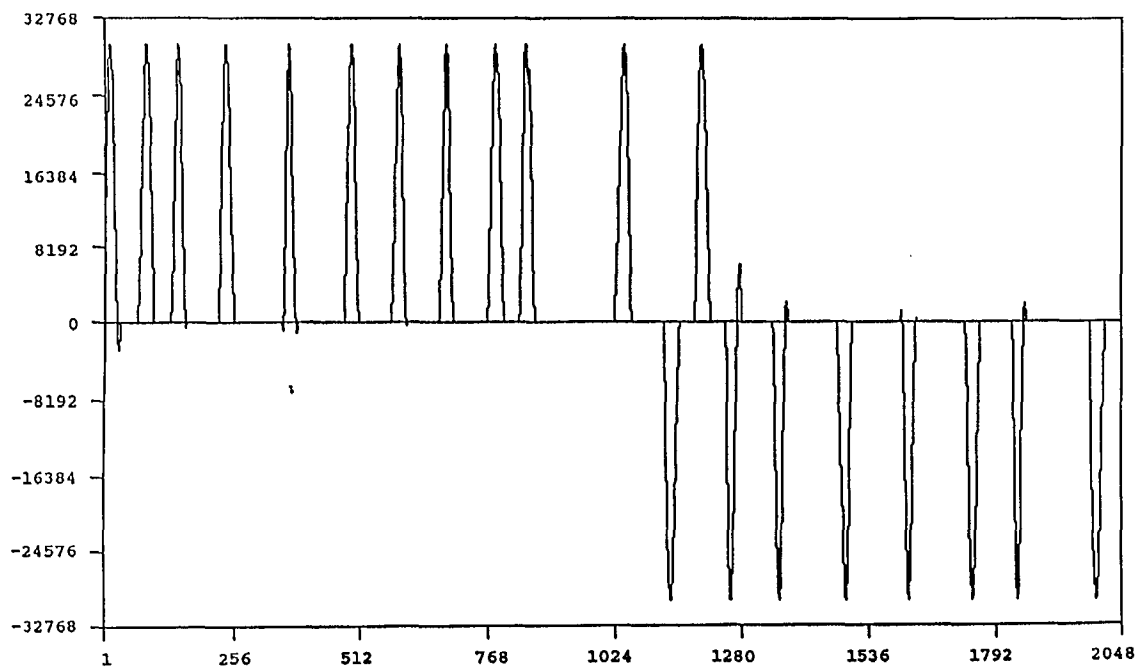


Figure 9: Exemple de "sous-blocs" normalisés en amplitude (30000) et en durée (32 échantillons chacun).

1.2.2 Algorithme d'expansion de temps

Un autre programme "dit de dénormalisation" ramène les sous-blocs normalisés (de R échantillons) au format d'origine c'est-à-dire rajoutent les échantillons manquants pour retrouver les fichiers de sous-blocs sur-échantillonnés. Il réajuste aussi l'amplitude convenable de tous les échantillons.

Cette opération est faite de la manière la plus simple qui soit. On effectue un "blocage" d'ordre zéro en ajoutant N-1 fois la valeur d'un même échantillon entre deux valeurs consécutives.



Ce programme donne donc en sortie un fichier de "sous-blocs", au rythme du sur-échantillonnage initial par R (soit R fois 32 kHz). Ce fichier sera directement utilisé par le décodeur pour régénérer le signal original. Dans ce décodeur l'option de sous-échantillonnage par R est évidemment incluse soit un retour à 32 kHz.

On complète alors la séquence des trois opérations déjà citées (page 13) en y ajoutant après la *décomposition* en "sous-blocs" et "supports"

- **Normalisation**

en amplitude (30000) et en durée (ou temps: R) du fichier des "sous-blocs", ce que représente la figure 9.

- **Dénormalisation**

de l'amplitude et du temps (par blocage d'ordre 0) pour revenir au "format" d'origine des sous-blocs".

Le signal, alors *recomposé et sous-échantillonné*, est restitué à 32 kHz.

Le niveau de qualité obtenu en terme de RSB (> 75 dB) est identique au précédent, et les erreurs (0,±1) sont de même nature. On notera qu'à la séquence précédente (paragraphe 1.1), on a ajouté les deux opérations de compression et d'expansion, sans dégradation supplémentaire.

1.2.3 Résultats de compression

Un programme permet de tracer les "sous-blocs normalisés" d'un même fichier (figure 9), superposés les uns aux autres. On obtient ainsi des diagrammes en forme d'oeil présentés sur les figures suivantes 10, 11, 12 et 13. Ces diagrammes sont construits sur des portions restreintes du signal sur-échantillonné par 8, 16 ou 32 puis sous-échantillonné par N pour obtenir des sous-blocs de 8, 16 ou 32 échantillons.

Sur ces figures, aucune précaution particulière, quant au choix des blocs de signal d'origine, n'a été prise. On relève cependant une prédominance manifeste de forme d'onde "prototype" quelle que soit la partition musicale utilisée (harpe, piano, chant, opéra ou orchestre).

Un autre programme détermine la courbe moyenne (moyenne arithmétique simple) de cet ensemble de courbes du diagramme en oeil.

Les figures suivantes 14 à 24 montrent ces courbes moyennes pour tous les fichiers de musique disponibles (sur 8 ou 32 échantillons). On remarque une identité remarquable de ces courbes qui rappelons-le sont construites à partir des diagrammes de l'oeil formé de toutes les courbes de sous-blocs normalisés des figures 10 à 13 et d'autres fichiers...

On constate que, par exemple sur la figure 11, certaines courbes "s'écartaient" de la forme prototype. Du fait que leur influence n'apparaît pas dans la moyenne c'est:

- qu'elles ne sont pas très nombreuses,
- qu'elles pourront, à l'avenir, être supprimées de l'analyse proprement dite.

ne normalisées à très faibles amplitudes et donc n'ont quasiment d'influence sur le RSB. Elles pourraient être écartées de l'analyse.

*OK
p 95*

Il s'agit de courbes qui ont des amplitudes très faibles (ex bruit), et en réalité ces courbes ont été

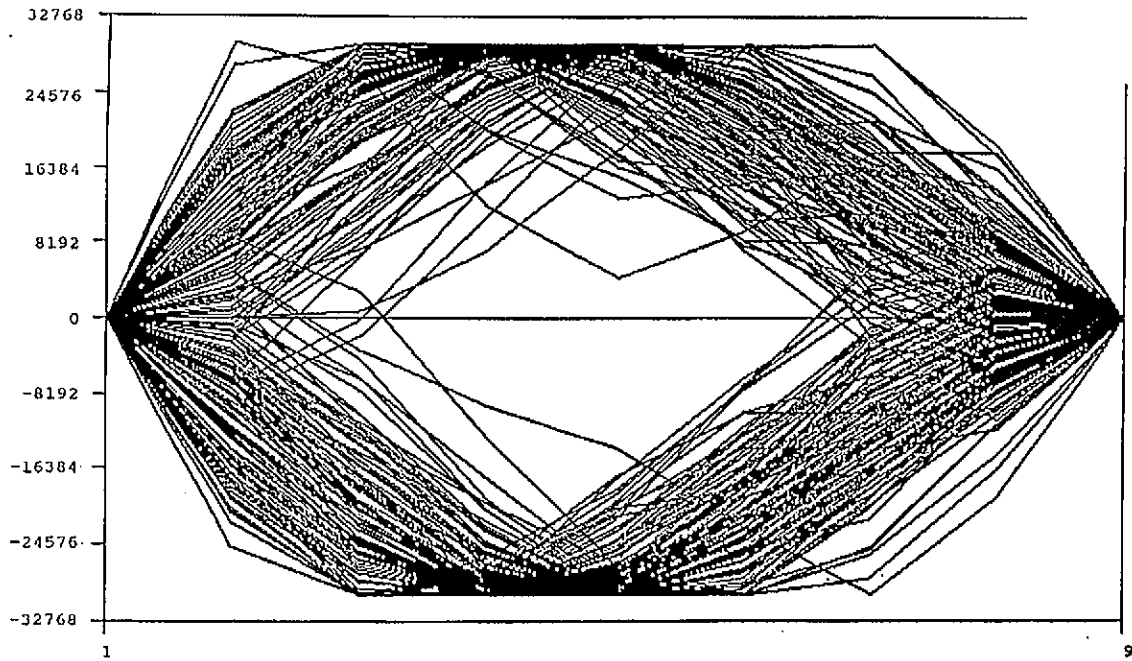


Figure 10: Superposition de "sous-blocs" de 8 échantillons (R=8) pour un extrait de harpe.

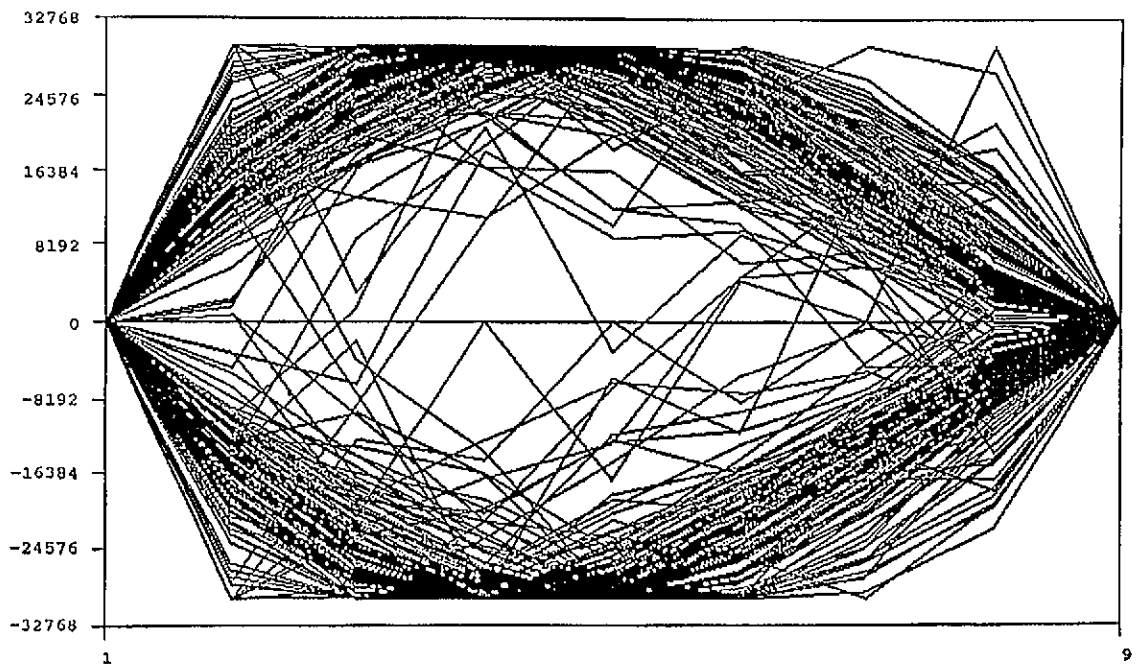


Figure 11: Superposition de "sous-blocs de 8 échantillons (R=8) pour un extrait de piano.

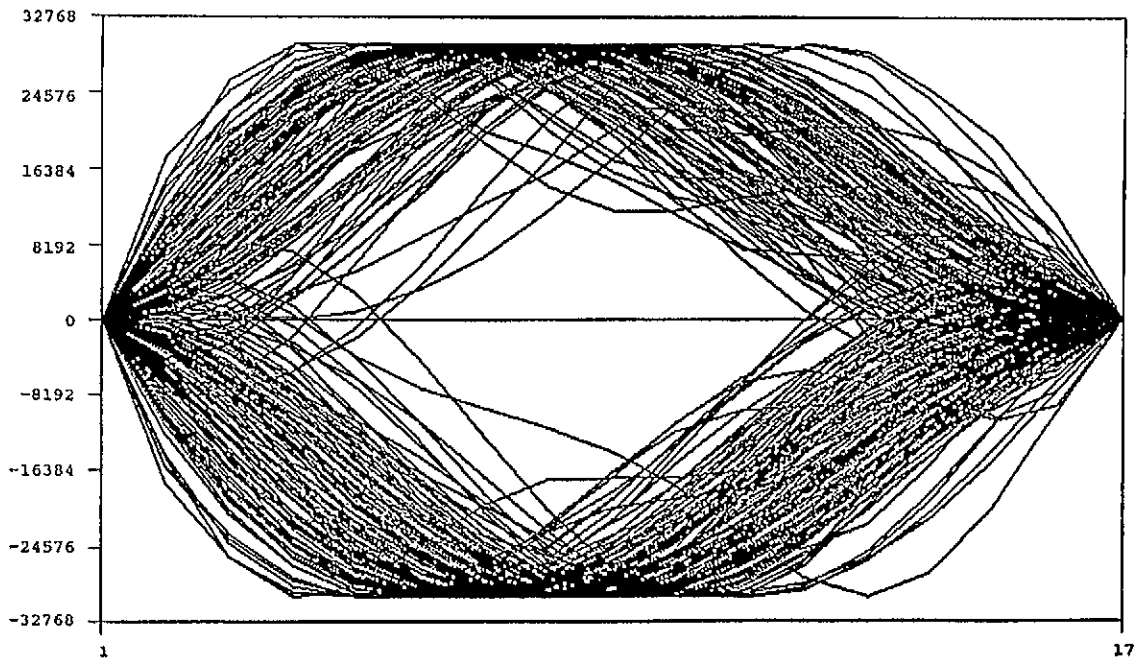


Figure 12: Superposition de "sous-blocs" de 16 échantillons (R=16) pour un extrait de harpe.

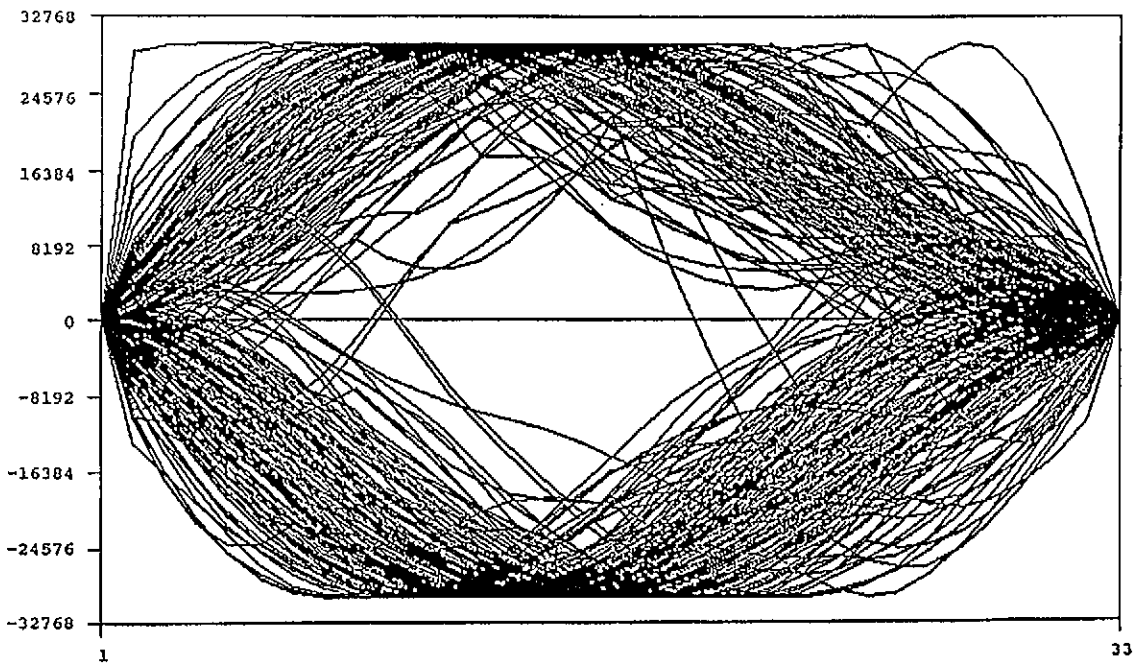


Figure 13: Superposition de "sous-blocs" de 32 échantillons (R=32) pour un extrait de harpe.

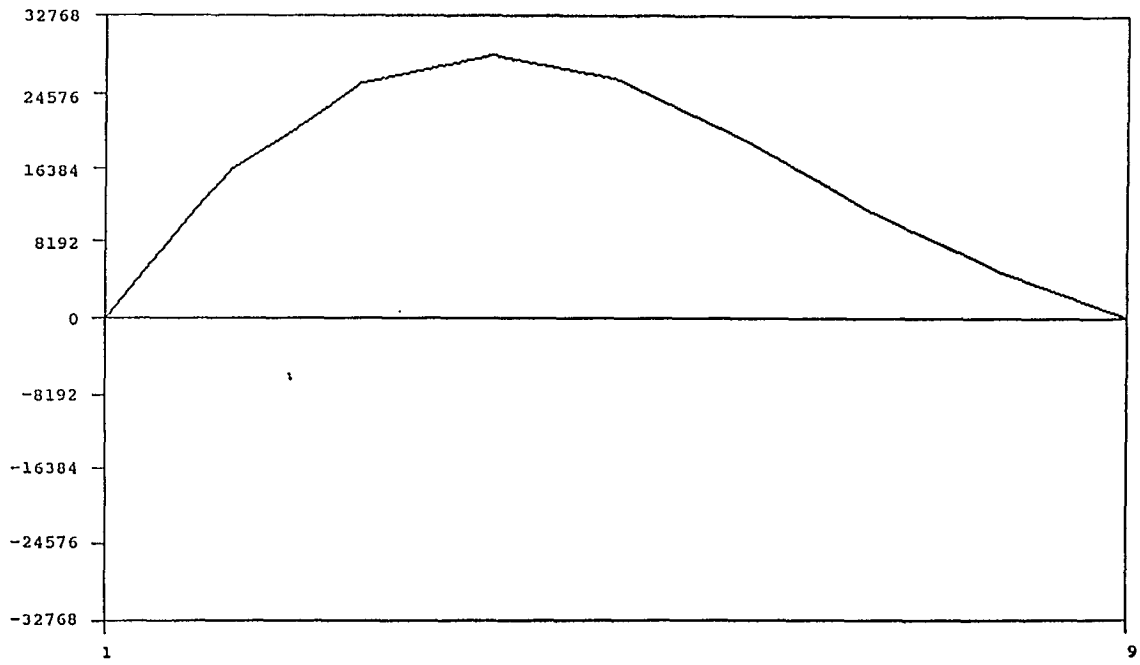


Figure 14: Courbe "moyenne" de tous les sous-blocs pour un extrait d'opéra 1 (R=8).

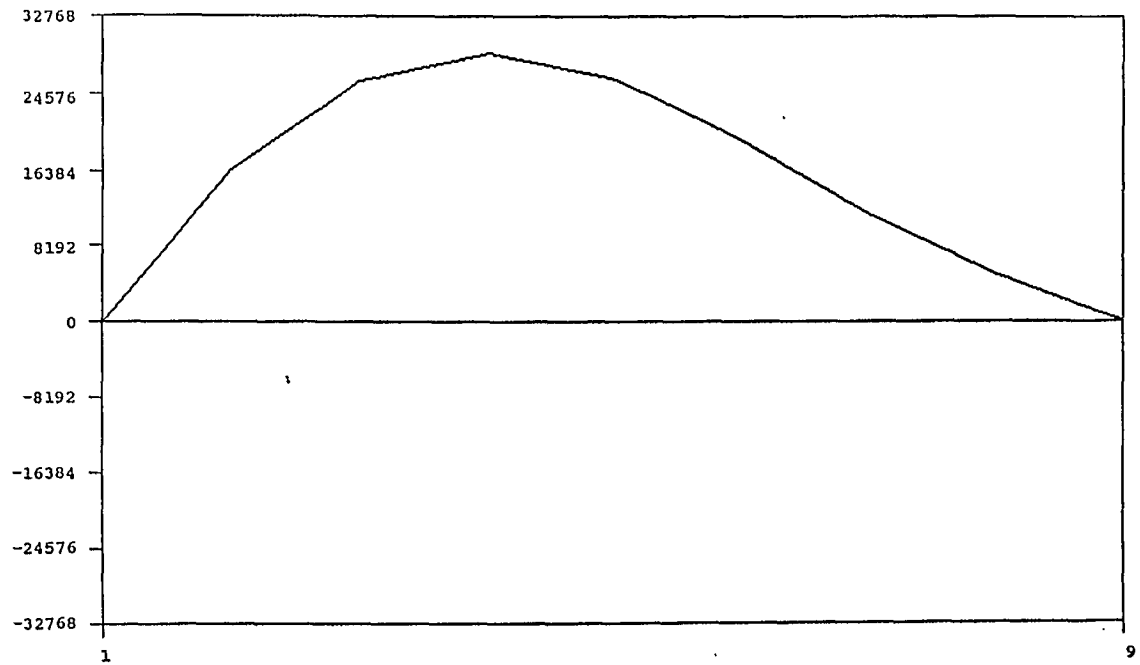


Figure 15: Courbe "moyenne" de tous les sous-blocs pour un extrait de piano 1 (R=8).

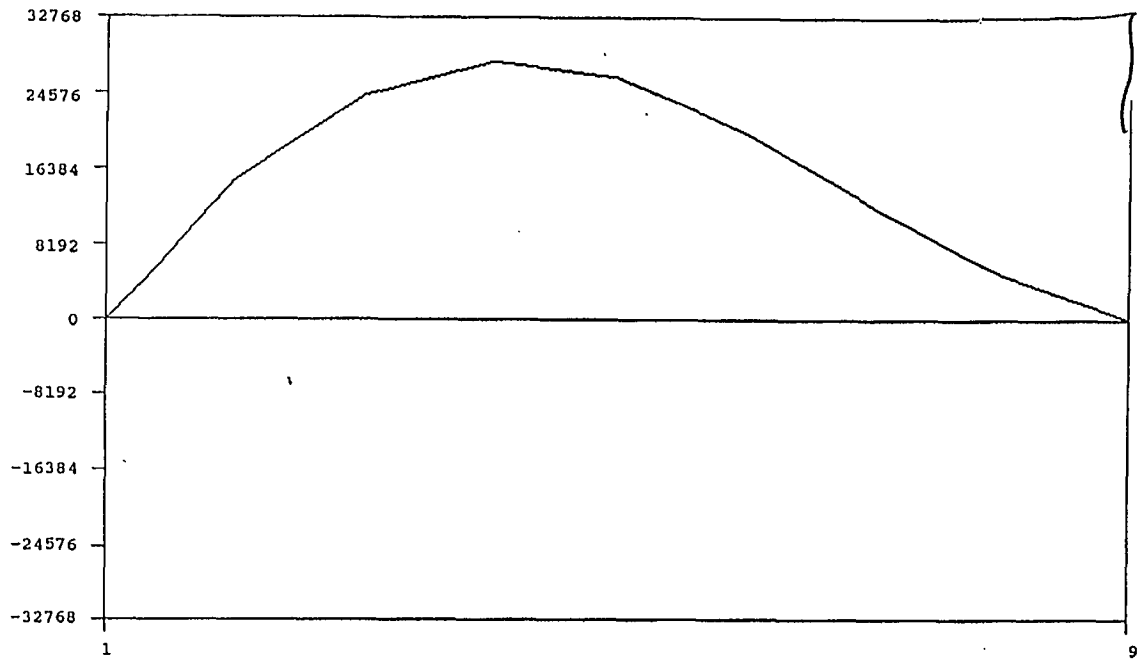


Figure 16: Courbe "moyenne" de tous les sous-blocs pour un extrait de violon 1 (R=8).

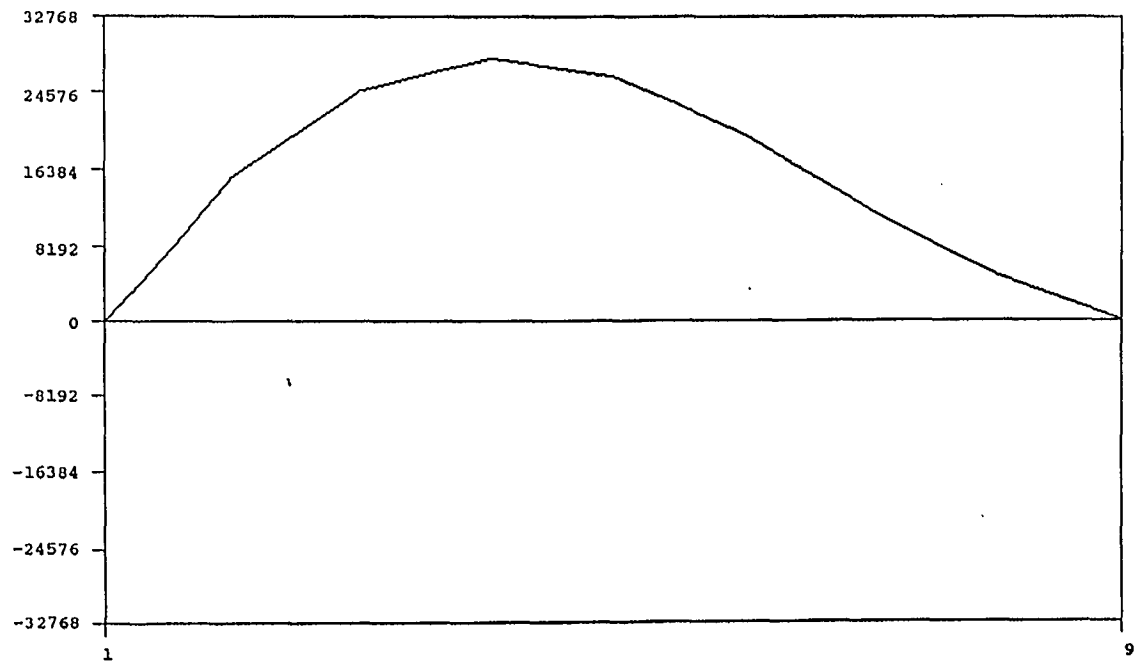


Figure 17: Courbe "moyenne" de tous les sous-blocs pour un extrait de chant (R=8).

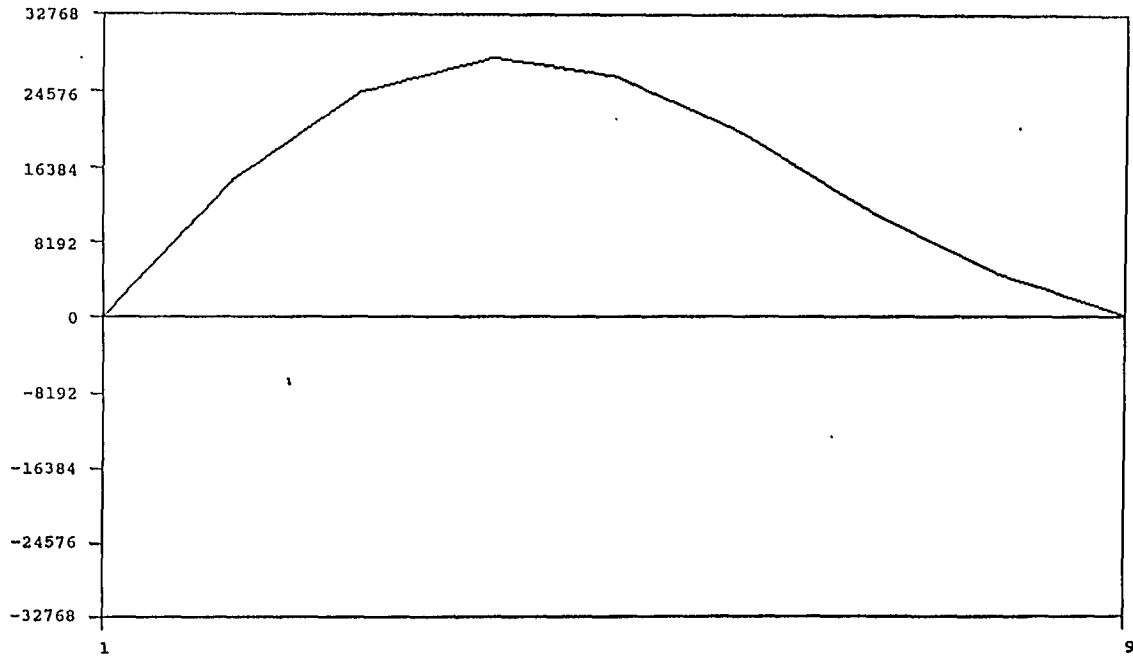


Figure 18: Courbe "moyenne" de tous les sous-blocs pour un extrait de violon 2 (R=8).

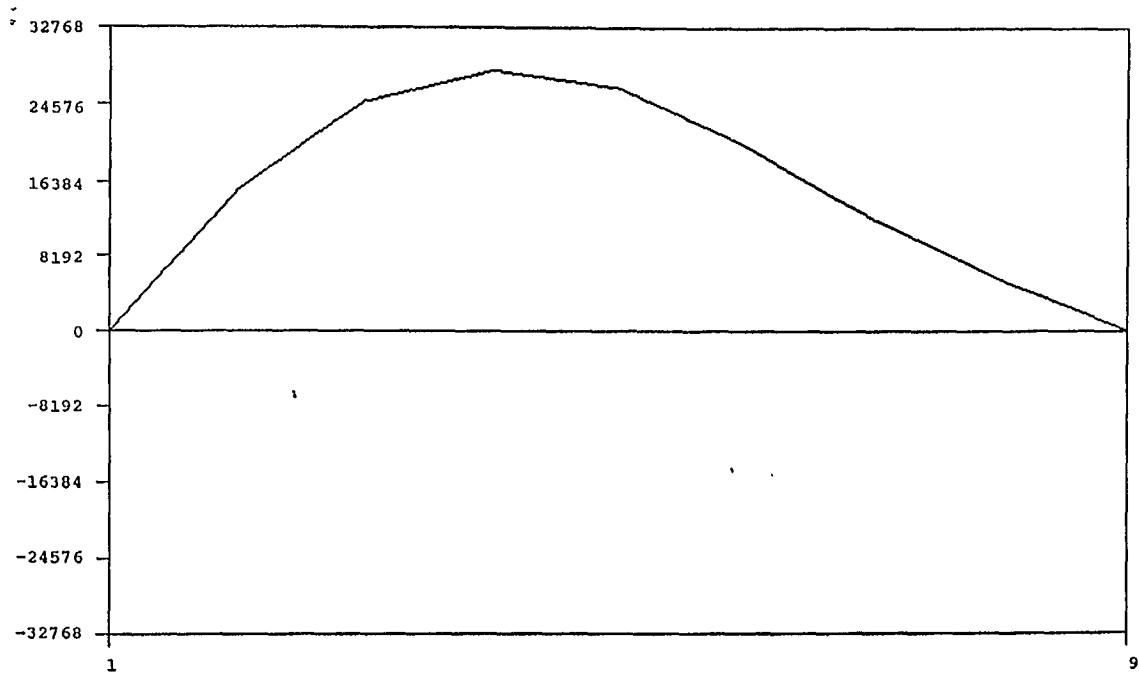


Figure 19: Courbe "moyenne" de tous les sous-blocs pour un extrait d'opéra 2 (R=8).

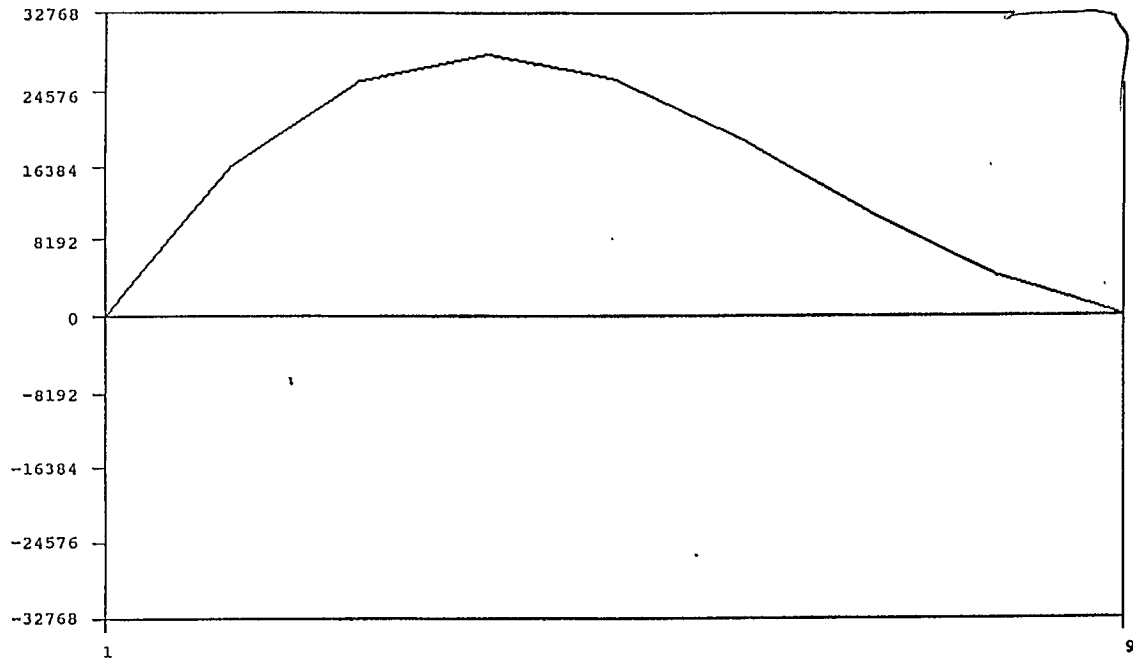


Figure 20: Courbe "moyenne" de tous les sous-blocs pour un extrait de flûte de Pan (R=8).

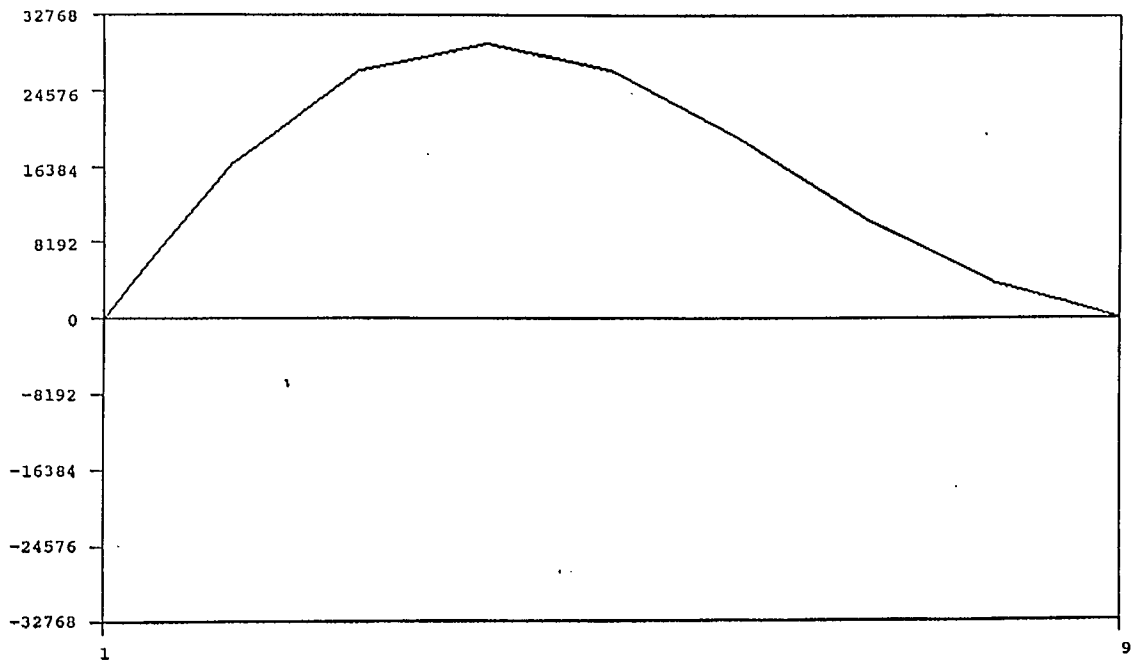


Figure 21: Courbe "moyenne" de tous les sous-blocs pour un extrait d'orgue (R=8).

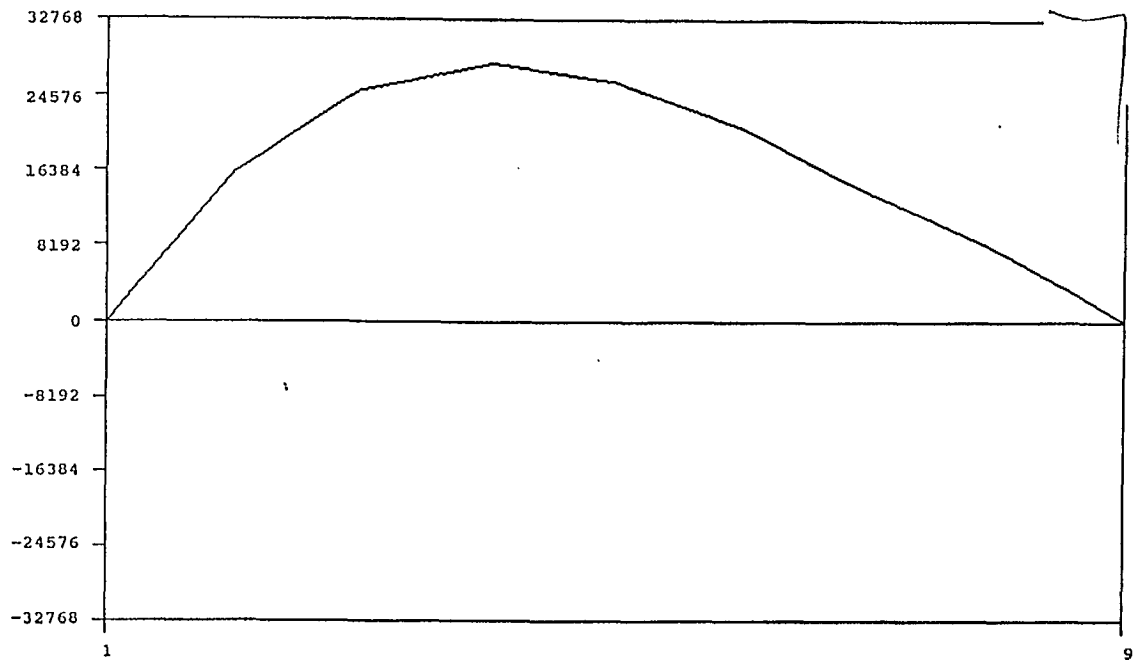


Figure 22: Courbe "moyenne" de tous les sous-blocs pour un extrait de piano 2 (R=8).

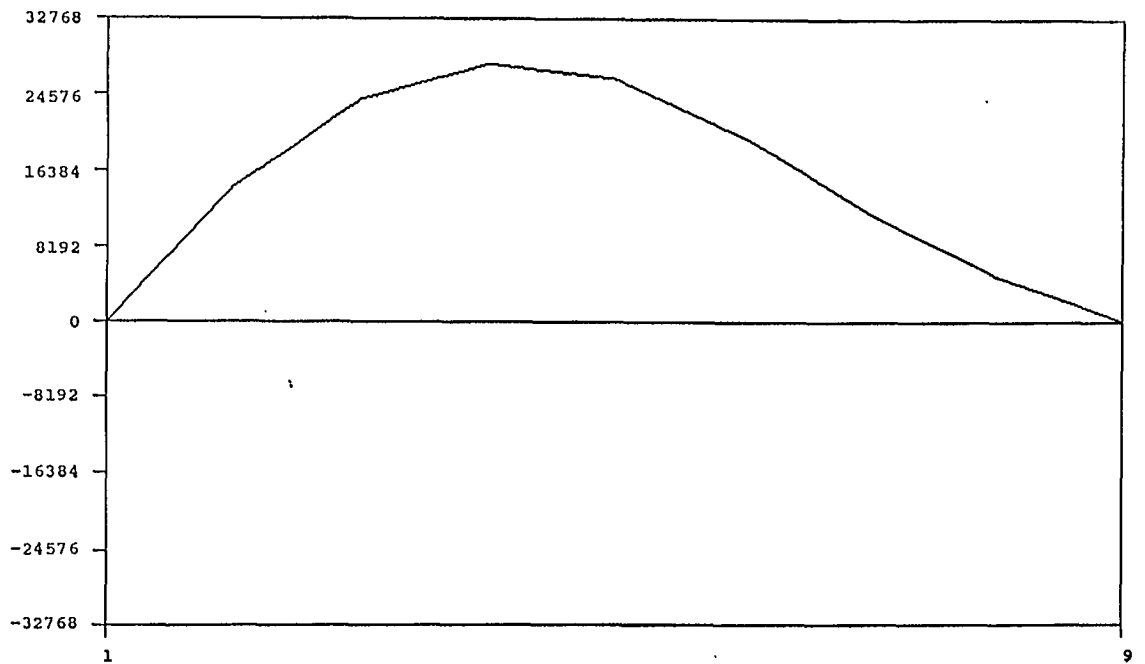


Figure 23: Courbe "moyenne" de tous les sous-blocs pour un extrait d'orchestre (R=8).

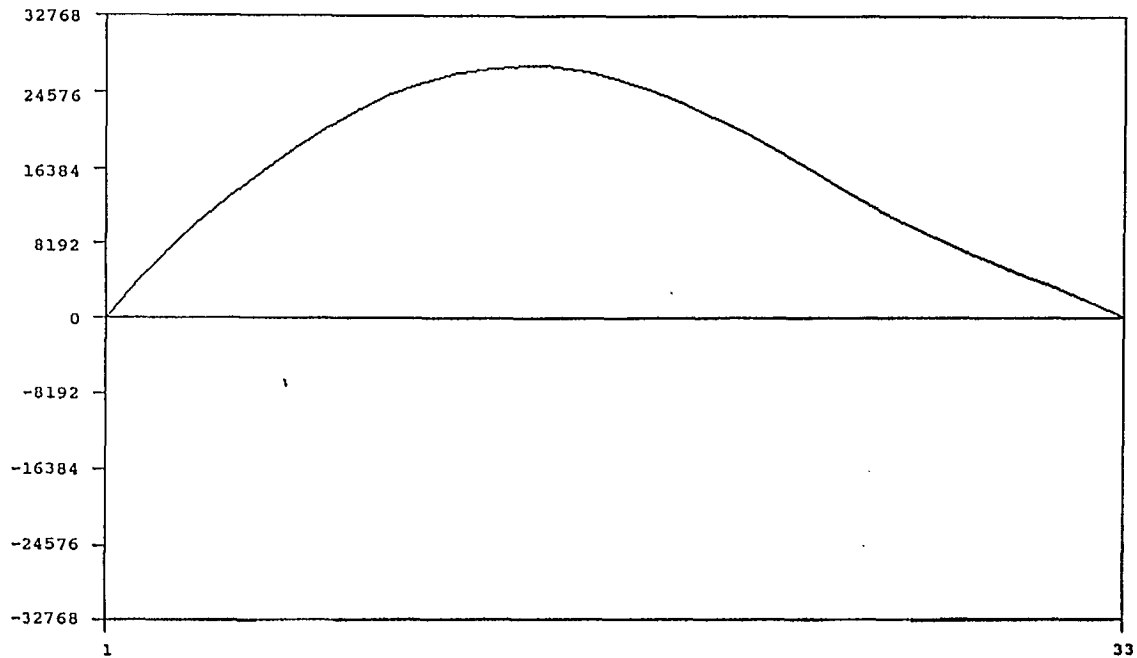


Figure 24: Courbe "moyenne" de tous les sous-blocs pour un extrait de harpe (R32).

1.2.4 Extension de l'analyse par sous-blocs normalisés

La figure 11 montrait la présence de certaines courbes de sous-blocs dites "marginales". On a pu mettre en évidence la raison principale de la présence de la plupart de ces courbes et donc ainsi de les éliminer dès l'analyse.

En effet, comme le montre la figure 25 ci-dessous, ces configurations sont dues essentiellement à des sous-blocs construits à partir de signaux de très basse fréquence, de plus largement sur-échantillonné (par $R=8$, 16 ou 32...). En raison du "bruit de calcul" du filtre numérique de sur-échantillonnage, le signal "d'écart" avec la droite d'interpolation linéaire oscille alors autour du zéro. En amplitude, il est extrêmement faible vis-à-vis des échantillons du signal

musical, mais la normalisation (en amplitude à 30000) du sous-bloc génère alors ce type de courbe marginale.

Un seuil de détection d'amplitude (noté $1/Q$) permet d'éliminer ces courbes: le programme ne conserve que les sous-blocs originaux dont l'échantillon maximal vaut au moins $1/Q$ fois l'amplitude moyenne du signal musical (il les met à zéro dans le cas contraire, ce qui, à la reconstitution donne seulement l'interpolation linéaire de la figure 25).

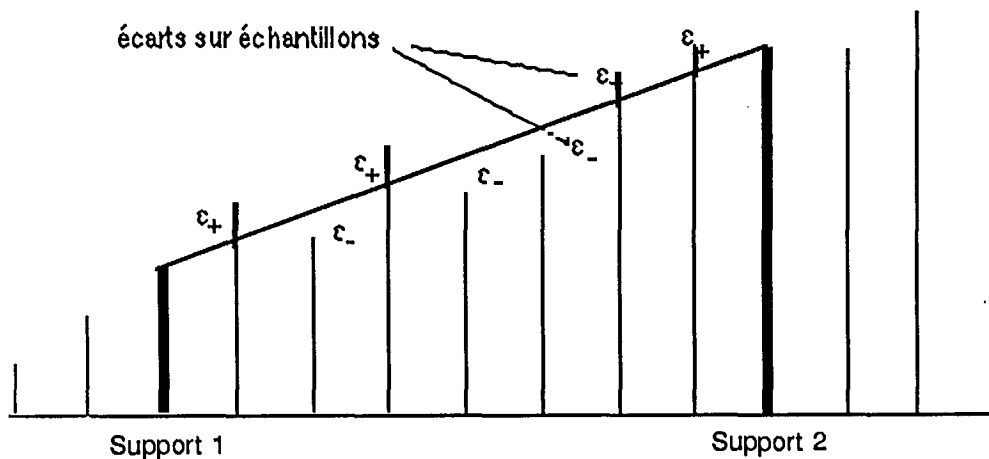


Figure 25 : Cas particulier: exemple de formation des " courbes marginales" à supprimer de l'analyse

Ce seuil n'est utilisé que pour la construction du dictionnaire qui servira ensuite au codage par quantification vectorielle.

Dans l'ensemble des opérations de codage d'un fichier de musique quelconque, la décomposition en sous-blocs peut être faite sans option de seuil. Certes le codage de ce type de sous-blocs ne sera sans doute pas efficace, mais après dénormalisation d'amplitude, l'influence du bruit de codage sera négligeable. Les figures 26 et 27 montrent la présence et l'élimination de l'une des courbes "marginales" obtenues par cette technique de seuil à l'analyse.

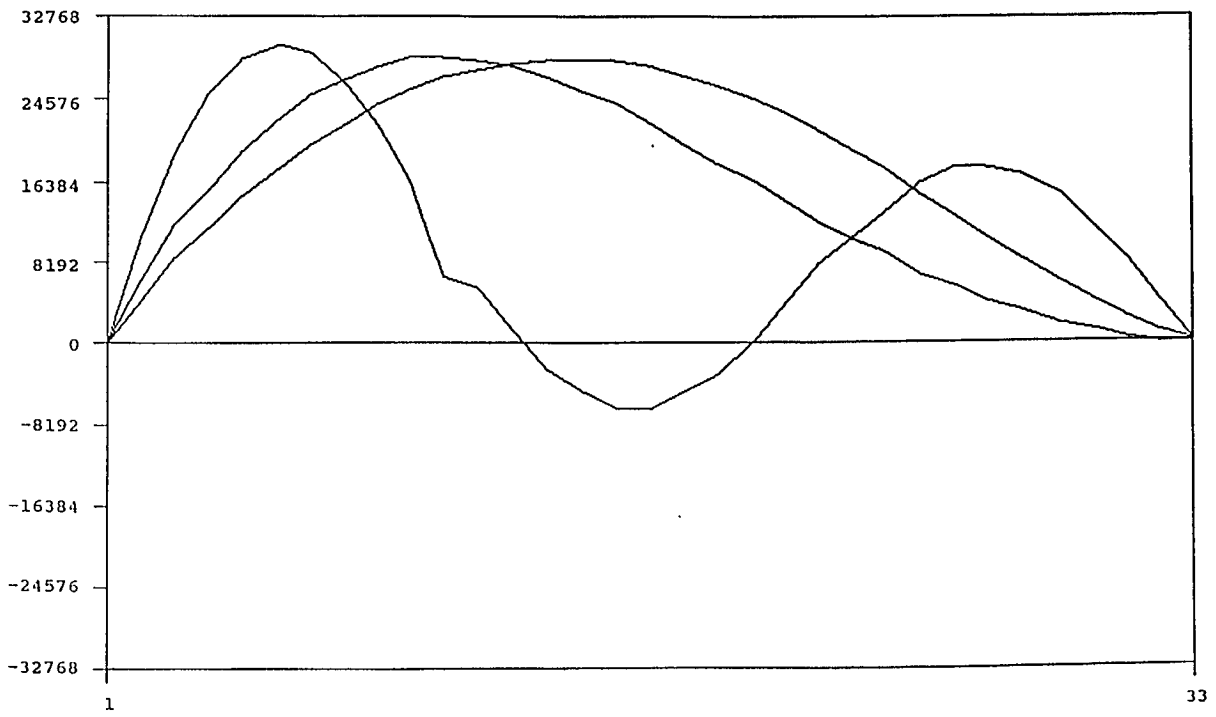


Figure 26: Exemple de sous-blocs avec une courbe marginale.

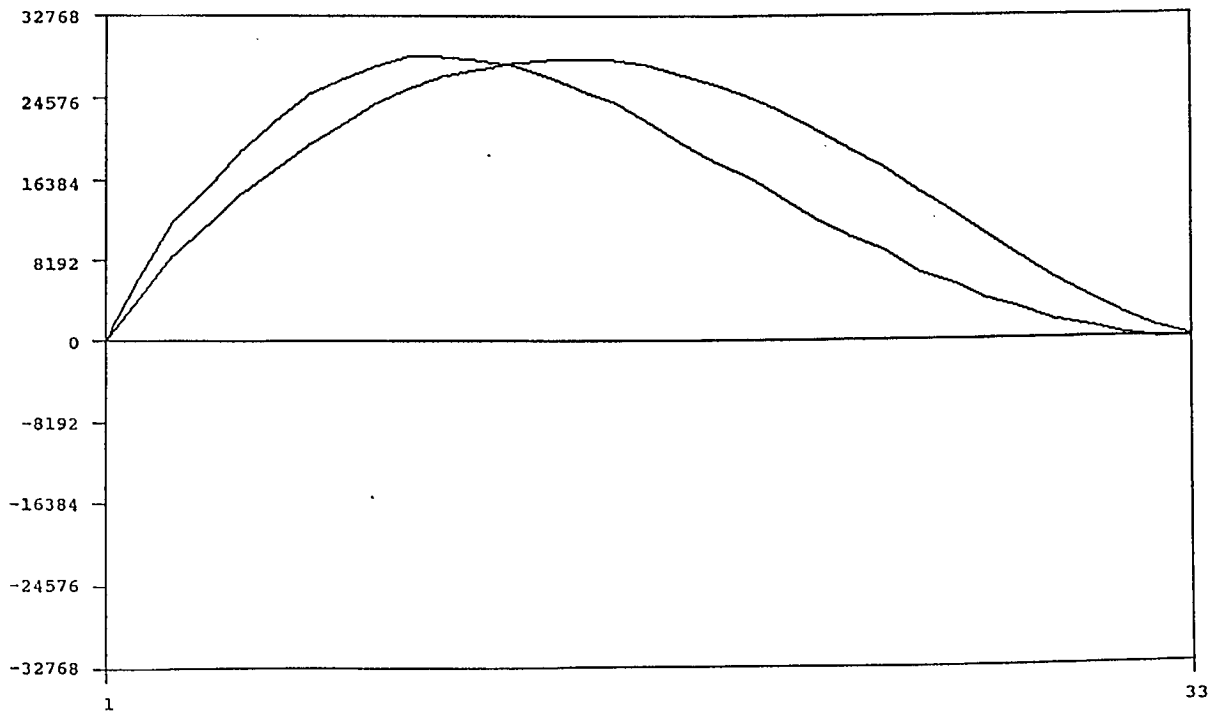


Figure 27: Elimination de la courbe marginale par un seuil ($Q=8$).

1.3 Estimation des résultats de la compression expansion - taux de réduction

Le programme de codage donne en fin d'exécution le taux (τ) de réduction obtenu. Il s'agit d'une moyenne de la longueur des "sous-blocs" obtenus par application de l'algorithme complet. Tous les "sous-blocs" sont pris en compte, y compris ceux obtenus pendant les intervalles dits de "silence" où le bruit de fond est le seul signal présent.

Pour des essais préliminaires, on a relevé, avec R=32 pour divers fichiers musicaux (de 10 à 30 secondes) les valeurs arrondies moyennes suivantes:

harpe	$\tau=7$
violon	$\tau=6$
orgue	$\tau=8$
flûte de Pan	$\tau=7$
piano 1	$\tau=9$
piano 2	$\tau=11$
chant	$\tau=6$
opéra 1	$\tau=5$
opéra 2	$\tau=6$
orchestre	$\tau=6$

Sur des portions brèves, en cas de hautes fréquences importantes (amplitude et durée) ces taux peuvent être temporairement réduits autour de 3 voire 4. On n'a pas, pour le moment, étudié la taille de la mémoire tampon nécessaire pour assurer le maintien d'une transmission à débit fixe avec une prise de décision à débit variable. Cependant on peut penser qu'une mémoire de 1 kmots (1024) serait suffisante.

$$32K/3 \approx 10KHz$$

$$10 \text{ Kbyte} = 10 \text{ sec} = 10 \text{ ms}$$

$$\text{Extrapolation} = 40 \text{ msec}$$

A partir de cela, une estimation de performance en réduction de débit peut être envisagée en prenant $\tau = 5$ comme taux moyen et la loi A - CCITT - 11 bits comme référence.

encore plus long pour
 τ plus grand!

- Loi A : codage de 5 échantillons, donc 55 bits

- Codage vectoriel variable, sur-échantillonnage par R=8

1 échantillon (support) 11 bits

1 numéro de courbe (512 max) 9 bits (voire 8)

1 facteur d'amplitude* 4 bits (ou 5)

1 facteur de durée (R =8)

Total

27 à 28 bits

38
5 bits

origine
semble faible! → 16 bits

* L'amplitude, variant peu d'un bloc à l'autre pourrait être codée en "différentiel adaptatif".

Soit en gros une réduction par 2 par rapport aux propositions actuelles.

→ image!
↓
hauts moins
beaucoup!

2. Quantification vectorielle

Bien connu dans le domaine du traitement de la parole, la quantification vectorielle devrait être un outil très intéressant pour quantifier des formes d'ondes aussi "ressemblantes" et ce avec un débit assez modeste.

2.1 Principe de la k-moyenne

La quantification vectorielle nécessite plusieurs étapes, dont une d'apprentissage:

a) Il faut choisir un nombre N de vecteurs d'apprentissage qui soient assez caractéristiques du signal à quantifier par la suite.

- On utilisera donc le programme d'analyse pour générer en premier lieu un fichier contenant les sous-blocs normalisés sur R échantillons. Le seuil d'élimination des courbes "marginales" est fixé à Q=8.

- Un autre programme utilise ensuite ce fichier pour former d'une part:

- Un fichier "d'apprentissage" pouvant contenir tout ou partie de ces sous-blocs (N sous-blocs).

- Un dictionnaire initial est construit à partir de ces sous-blocs par un tirage arbitraire (modulo n) soit k vecteurs de base ou "centres des k classes".

= $\frac{1}{8}$ →
rapport entre Δ et
vecteurs horaires
et amplitude de la
forme d'onde de référence
est logarithmique!
 $\frac{1}{8}$ minimum!

Note: Tous ces vecteurs sont composés de R (R=32, ou 16, ou 8) périodes d'échantillonnage donc de (R+1) points. Le premier et le dernier élément des sous-blocs normalisés sont toujours nuls (voir courbes des figures 16 à 26).

b) Il faut ensuite comparer chacun des N vecteurs d'apprentissage (de M composantes) à chacun des k vecteurs choisis comme vecteurs du dictionnaire initial.

- Chaque vecteur d'apprentissage est ainsi "plus proche" voisin d'un seul vecteur de base. Il appartient à la classe associée à ce vecteur de base. La distance utilisée est une distance quadratique euclidienne. AMS

Note:- Un index "pointeur", inscrit pour chaque vecteur d'apprentissage, détermine le numéro du vecteur suivant appartenant à la même classe (figure 2.8). - Un tableau pointeur indique le numéro de chaque premier vecteur de la classe de rang 1 puis 2 puis 3, etc.. Ce tableau finit toujours par "0" de même que le dernier vecteur d'apprentissage d'une classe est toujours suivi d'un "0". Cela sert de test de bon fonctionnement.

La figure 28 illustre le concept de simulation de la k-moyenne

En fin d'exécution de k-moyenne le fichier d'origine, contenant k vecteurs de base a été remplacé par un fichier contenant les "vecteurs centres" moyens de tous les vecteurs appartenant à la même classe - (plus proches voisins d'un centre particulier au sens d'une distance moyenne euclidienne quadratique) .

Le programme de k-moyenne fournit aussi les résultats de toutes les itérations (nombre de vecteurs par classe, erreur moyenne globale sur toutes les classes, erreur moyenne par

Procédure de K-moyenne

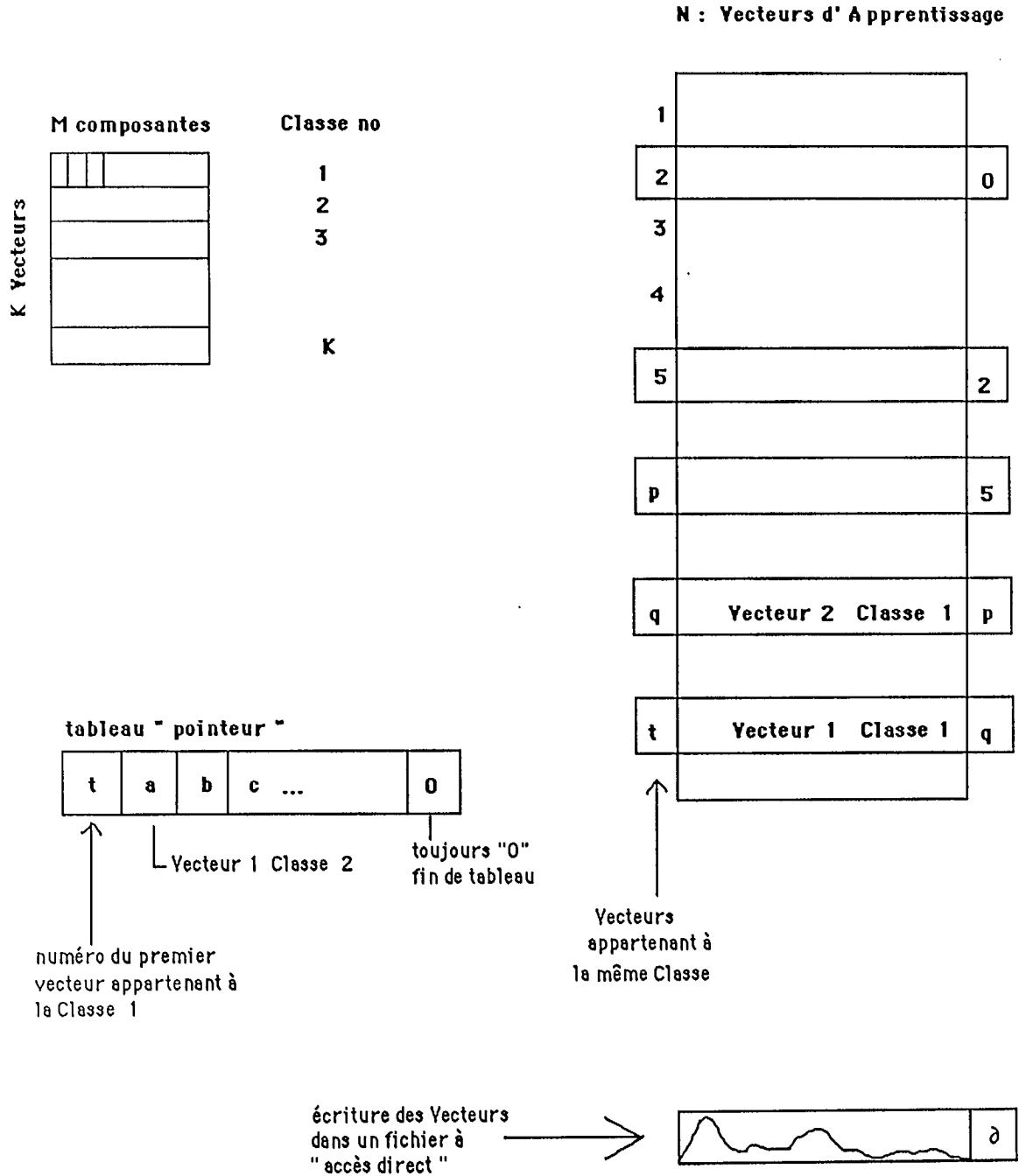


Figure 28 : Algorithme de k- moyenne

composante de chaque vecteur...). Le nombre d'itération est limité par un choix (au terminal) ou par un seuil: si l'erreur globale moyenne varie de moins de "x pour mille" (x choisi au terminal) les itérations sont terminées à la première limite atteinte.

Le programme de k- moyenne peut ensuite être utilisé plusieurs fois consécutives en prenant comme origine le dictionnaire obtenu après convergence et en lui appliquant une nouvelle séquence d'apprentissage issue par exemple d'un autre fichier musical.

2.2 Quantification des vecteurs

Pour la construction du dictionnaire, on prendra soin à n'utiliser que des parties de fichiers où le signal correspond effectivement à de la musique et non à du bruit de fond. Une fois le dictionnaire final obtenu après les "itérations" de k-moyenne sur une partie, ou la totalité de un ou plusieurs fichiers de musique, on effectue une recherche (exhaustive) du centre "le plus proche voisin" (au sens de la distance euclidienne) de chacune des courbes de sous-blocs normalisées. Le numéro d'ordre de ce centre est la *seule information transmise* avec comme information latérale, l'*amplitude* (en fait, il s'agit d'un facteur de normalisation ou gain) et la *durée* pour dénormalisation.

a) Dictionnaire de 256 vecteurs de 8 composantes chacun

La figure 29 montre un extrait d'une courte séquence d'apprentissage (284 courbes) sur la harpe, construite avec un seuil défini par $Q=8$. On y remarque l'absence des courbes "marginales" signalées.

La figure 30 montre un dictionnaire initial extrait de 256 premières courbes de la séquence (bien sûr la partie négative symétrique existe mais n'est pas représentée...).

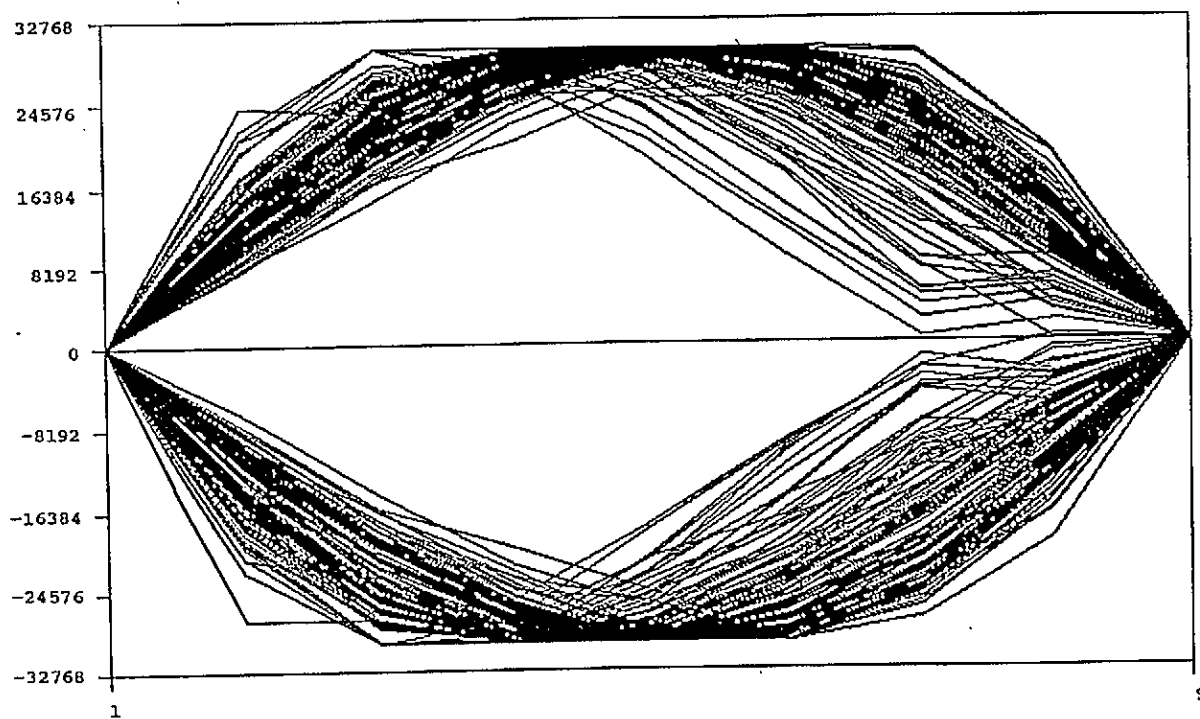


Fig. 29 : Extrait de la séquence d'apprentissage (Harpe)

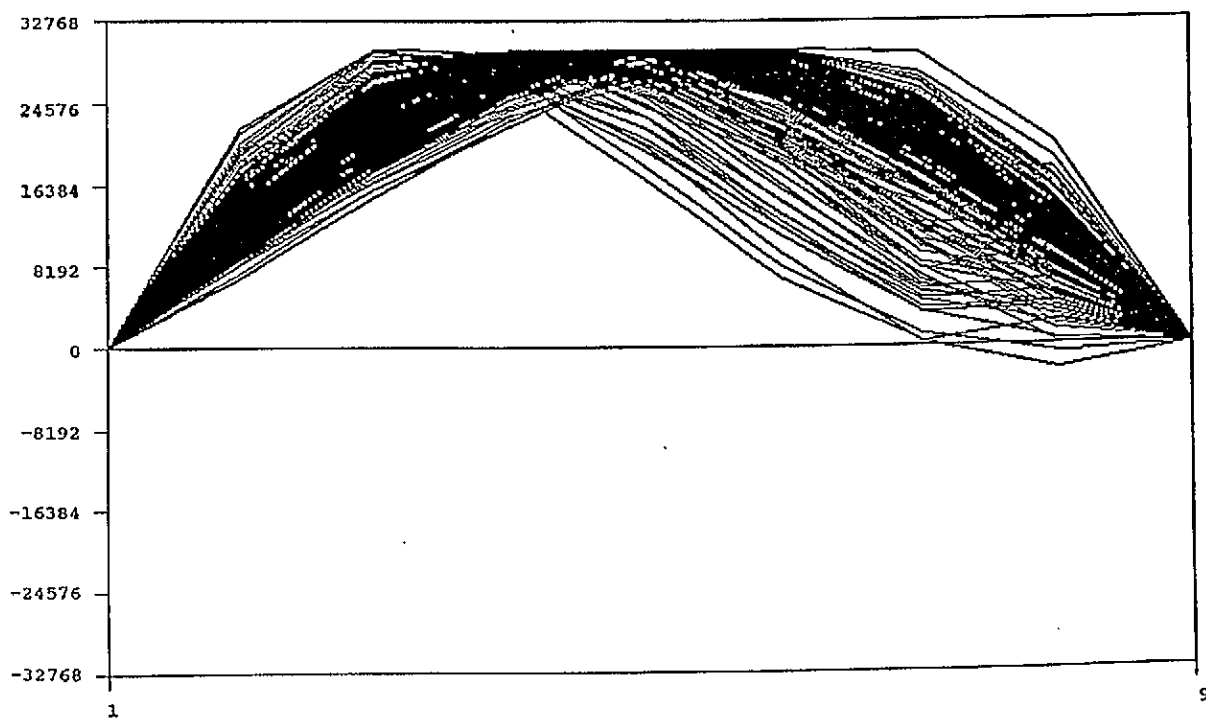


Fig.30 : Dict. initial 256 vecteurs 8 composantes sur séq. app. (8 x 200 blocs) Q=8

La figure 31 montre un essai préliminaire de k-moyenne. On constate un léger "resserrement" de la dispersion des courbes. On a d'ailleurs vérifié que pour $k=1$, le programme converge bien, au 1er tour vers la courbe moyenne des figures 14 à 24.

La figure 32 montre un résultat étendu sur 5000 blocs contenant plus de 10000 courbes d'apprentissage. La dispersion s'explique par la *variété des courbes d'apprentissage rencontrées*.

b) Dictionnaire de 256 vecteurs de 32 composantes chacun

Les figures suivantes illustrent quelques éléments de comportement des résultats, en prenant un facteur $R=32$ pour le sur-échantillonnage.

La figure 33 est un dictionnaire obtenu sans précaution (donc sans seuil Q) et incluant par conséquence toutes les courbes "marginales" intervenues dans la suite des 256 premiers sous-blocs prélevés pour construire le dictionnaire Initial.

La figure 34 montre la "convergence après k-moyenne". Il reste malheureusement encore beaucoup de courbes marginales. On vérifie sur le tableau présenté au terminal seulement que ces vecteurs de base ne correspondent qu'à quelques cas seulement. Autant les supprimer de l'histogramme et autant le faire avant, c'est-à-dire au cours de l'analyse par utilisation du seuil $Q=8$.

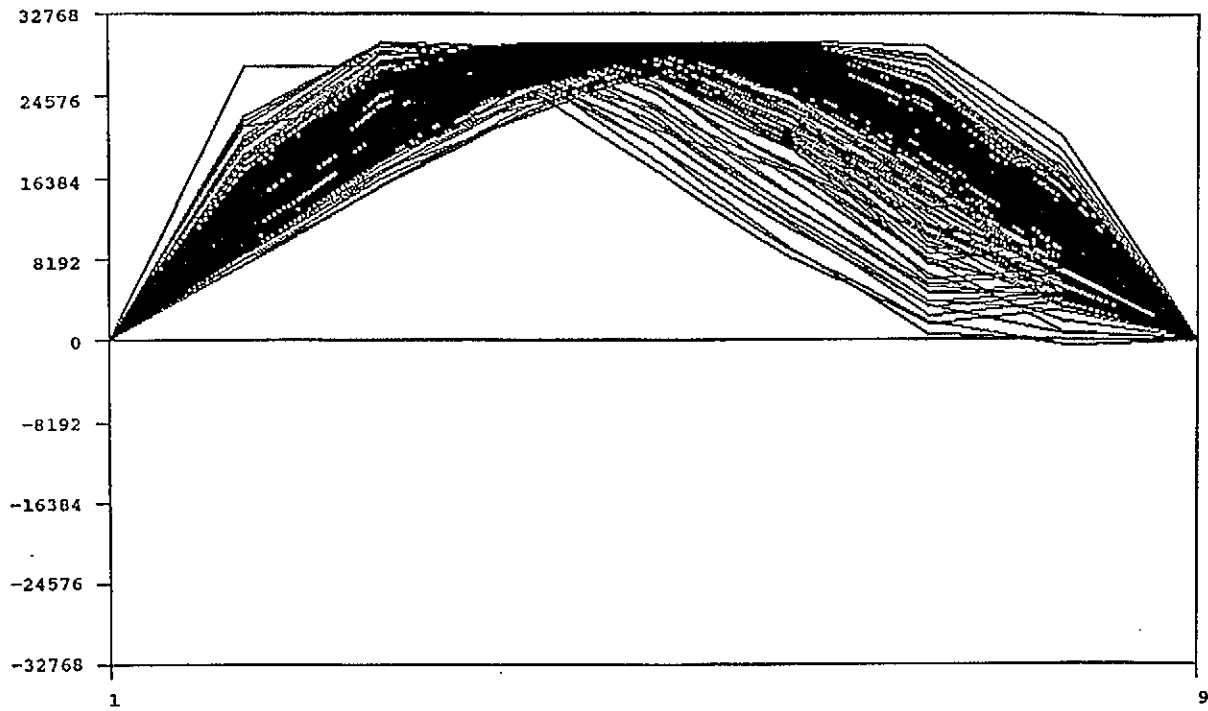


Fig. 31 : Essai de k-moyenne sur seulement 500 blocs.Nouveau dictionnaire

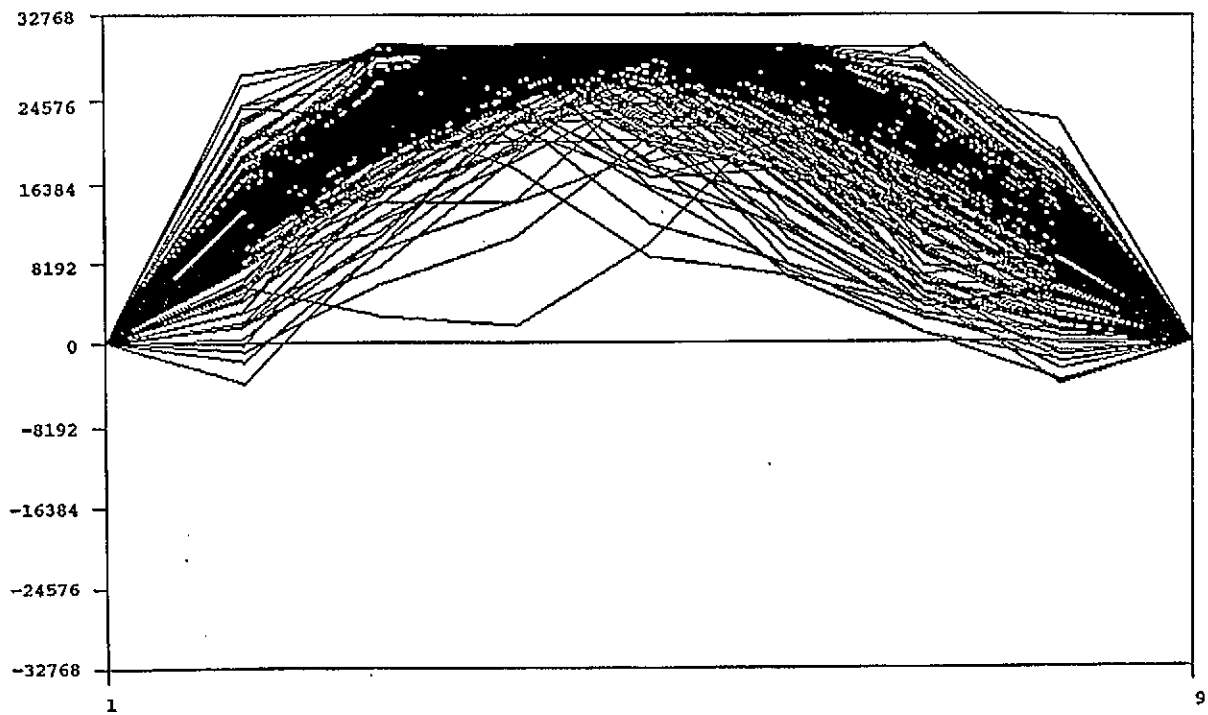


Fig. 32 :Dict. final après k-moy. sur plus de 10000 vect.d'apprentissage (Harpe)

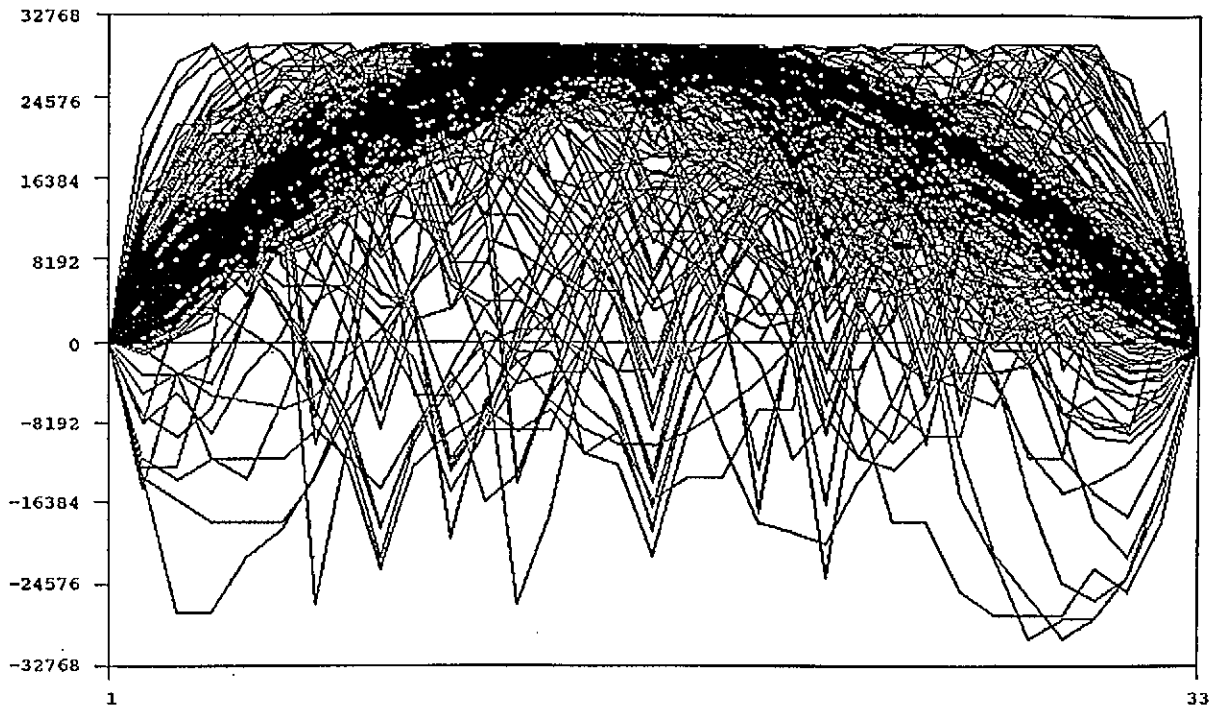


Figure 33 : Dictionnaire 256 vect. obtenu sur segment d'apprentissage initial (14336)

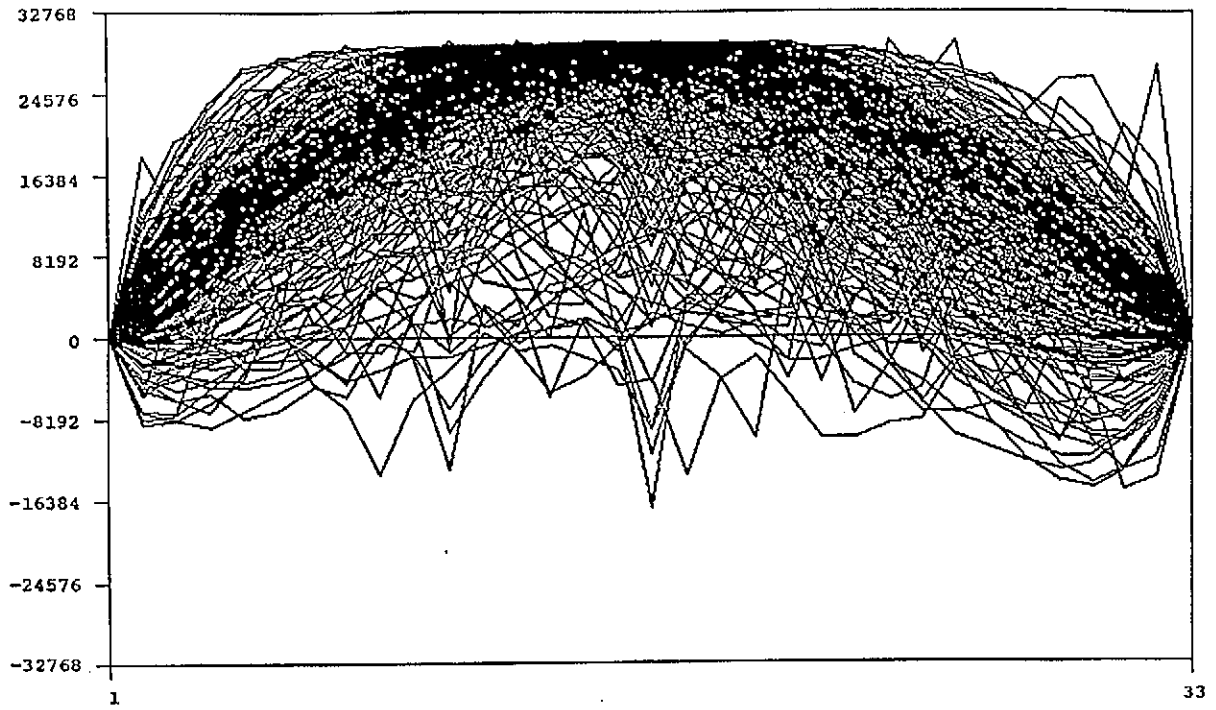


Figure 34 : Dictionnaire obtenu après k-moyenne(reste 241 vecteurs)

La figure 35 quant à elle, utilise bien le facteur $Q=8$ à l'analyse et peut être comparée à la figure 33. Pourtant au moins 3 courbes "marginales" sont présentées de manière évidente. Et bien, parce que le programme élimine tous les sous-blocs au-dessous du seuil [$1/Q$ fois la moyenne des échantillons de support du signal (sous-échantillonné)] et que ici, on a une séquence d'apprentissage qui commence sur des très faibles signaux, du même ordre de grandeur que les sous-blocs, le seuil n'a pas agi sur les 3 premiers sous-blocs concernés. Il faut donc veiller à prendre certaines précautions AVANT de créer arbitrairement des séquences d'apprentissage.

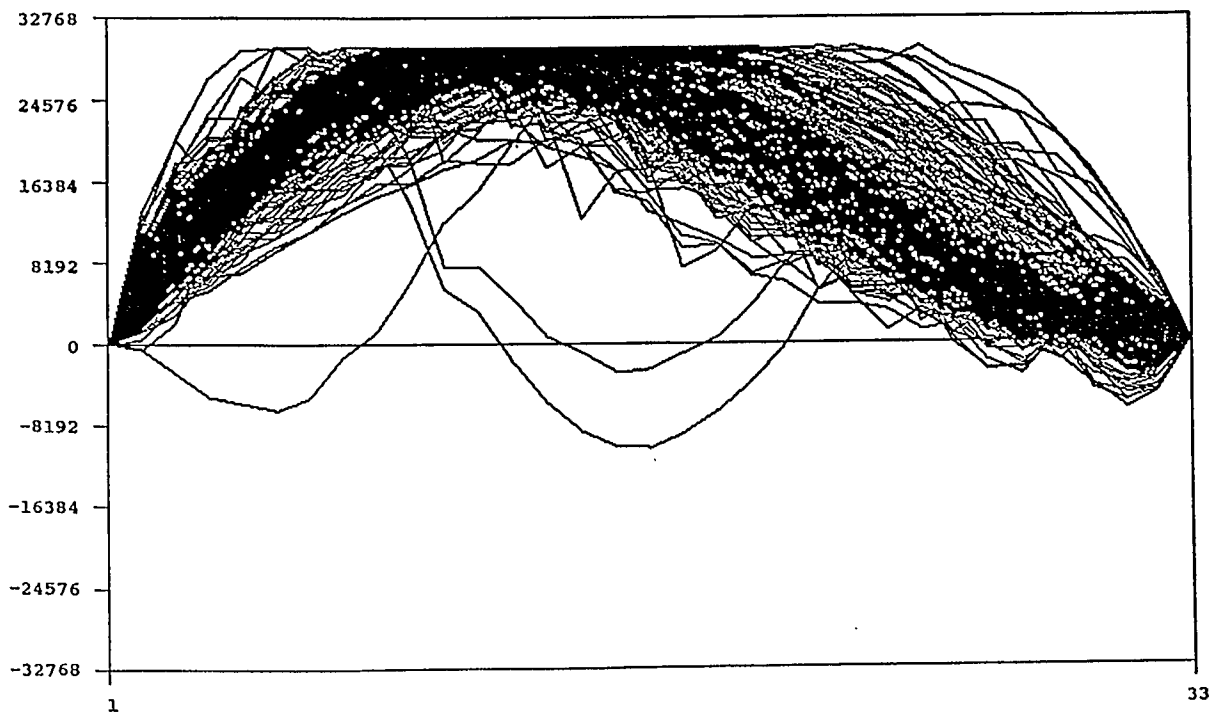


Figure 35 : Dict. initial 256 vect. 32 composantes sur seq.d'apprent. (32x200 blocs) $Q=8$

c) Décodage

Le décodage, au niveau du récepteur est quant à lui, extrêmement rapide. On lit le "vecteur centre" identifié par le numéro reçu, on le dénormalise

- une multiplication par composante pour la partie amplitude
- un blocage (N-1) d'ordre 0 par N-1 périodes
- puis un sous-échantillonnage par R pour la partie durée ou temps), et
- enfin on ajoute ce sous-bloc à l'interpolation linéaire effectuée entre les "supports" du signal .

2.3 Résultats préliminaires

En raison du temps limité disponible les simulations "efficaces" n'ont permis que de fournir des résultats préliminaires.

Une simulation a été faite avec un dictionnaire construit, en apprentissage, sur les blocs 850 à 1050 du fichier de harpe. Ces blocs présentent l'intérêt de comprendre seulement du signal (pas de bruit de fond) formé soit d'assez basses fréquences (400 - 500 Hz), soit de fréquences plus élevées avec surtout des harmoniques jusqu'à 7-8 kHz.

Ce dictionnaire a été limité à 512 représentants de base, soit 9 bits de codage. Le signe est inclus. Chaque forme d'onde est composée de 8 échantillons.

On obtient les chiffres suivants:

- Taux de réduction $\tau = 5,16$
- Débit par sous-bloc codé 9 bits
- Amplitude (hypothèse) 4 bits
- Durée ~~3 bits~~

2 15
5

Soit une transmission de 27 bits (en codant chaque support avec 11 bits) en lieu et place de $11 \times 5,16 = 57$ bits. On serait juste dans un ordre de grandeur de 2 pour la compression de débit. A la limite, le dictionnaire peut être doublé en taille et ainsi avec 10 bits, on peut maintenir le rapport 2 de compression.

Ces chiffres ne sont qu'indicatifs car la qualité subjective d'écoute montre un bruit de fond de quantification nettement perçu. On remarquera cependant que pour diminuer le bruit de fond, on devrait normalement augmenter la taille du dictionnaire. Doubler cette taille ne représente qu'un seul bit supplémentaire.

Il est intéressant de noter que si le taux de réduction (ou longueur moyenne de sous-blocs) reste élevé (5 ou plus), il contribue à une réduction théorique du débit très considérable¹ (5 ou plus); alors que doubler voire quadrupler ou octupler la taille du dictionnaire ne rajoute que 1, 2 ou 3 bits par "sous-bloc codé". Il est sans doute possible d'arriver à un compromis intéressant, autour d'un facteur 2 de compression.

2.3.1 Résultats expérimentaux

La figure 36 montre tout d'abord des tableaux de RSB calculés tant dans le domaine temporel que dans le domaine fréquentiel.

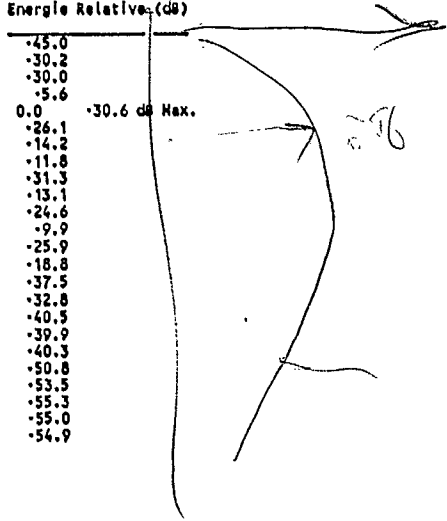
Les calculs par bande de fréquence sont effectués sur les spectres du signal et du signal bruité. Les 24 bandes sont ici les "bandes critiques" de Zwicker.

¹ Chaque échantillon affecté à un sous-bloc "enlève" 11 bits de la trame de transmission.

RSB Classique = 49.18 dB
 RSB Segmentaire = 49.18 dB
 Niveau moyen du Signal par rapport au maximum (saturation) : -28.60 dB Max.
 RSB Fréquentiel = 52.25 dB

0 Seg. a bruit nul
 0 Seg. a signal nul (bruit moyen = 0)
 pour tout le spectre : bande unique
 RSB freq./bande Energie Relative (dB)

Bande de Freq.	()	⇒		
Bande de Freq. 1	(20	100)	⇒	22.98
Bande de Freq. 2	(100	200)	⇒	36.33
Bande de Freq. 3	(200	300)	⇒	38.68
Bande de Freq. 4	(300	400)	⇒	63.14
Bande de Freq. 5	(400	510)	⇒	68.14
Bande de Freq. 6	(510	630)	⇒	42.79
Bande de Freq. 7	(630	770)	⇒	55.78
Bande de Freq. 8	(770	920)	⇒	56.87
Bande de Freq. 9	(920	1080)	⇒	38.04
Bande de Freq. 10	(1080	1270)	⇒	50.12
Bande de Freq. 11	(1270	1480)	⇒	42.30
Bande de Freq. 12	(1480	1720)	⇒	54.69
Bande de Freq. 13	(1720	2000)	⇒	39.21
Bande de Freq. 14	(2000	2320)	⇒	46.62
Bande de Freq. 15	(2320	2700)	⇒	24.92
Bande de Freq. 16	(2700	3150)	⇒	32.52
Bande de Freq. 17	(3150	3700)	⇒	22.14
Bande de Freq. 18	(3700	4400)	⇒	22.28
Bande de Freq. 19	(4400	5300)	⇒	22.55
Bande de Freq. 20	(5300	6400)	⇒	10.51
Bande de Freq. 21	(6400	7700)	⇒	7.77
Bande de Freq. 22	(7700	9500)	⇒	5.37
Bande de Freq. 23	(9500	12000)	⇒	6.67
Bande de Freq. 24	(12000	15500)	⇒	5.97

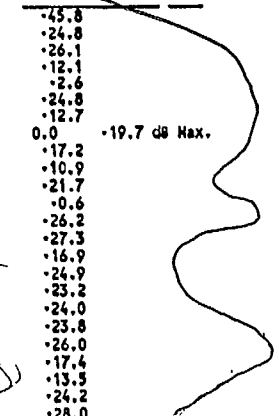


Basse fréquence

RSB Classique = 31.77 dB
 RSB Segmentaire = 31.16 dB
 Niveau moyen du Signal par rapport au maximum (saturation) : -15.44 dB Max.
 RSB Fréquentiel = 35.95 dB

0 Seg. a bruit nul
 0 Seg. a signal nul (bruit moyen = 0)
 pour tout le spectre : bande unique
 RSB freq./bande Energie Relative (dB)

Bande de Freq.	()	⇒		
Bande de Freq. 1	(20	100)	⇒	6.32
Bande de Freq. 2	(100	200)	⇒	27.19
Bande de Freq. 3	(200	300)	⇒	29.53
Bande de Freq. 4	(300	400)	⇒	40.47
Bande de Freq. 5	(400	510)	⇒	50.89
Bande de Freq. 6	(510	630)	⇒	26.47
Bande de Freq. 7	(630	770)	⇒	43.42
Bande de Freq. 8	(770	920)	⇒	55.83
Bande de Freq. 9	(920	1080)	⇒	32.61
Bande de Freq. 10	(1080	1270)	⇒	41.06
Bande de Freq. 11	(1270	1480)	⇒	31.02
Bande de Freq. 12	(1480	1720)	⇒	49.83
Bande de Freq. 13	(1720	2000)	⇒	22.73
Bande de Freq. 14	(2000	2320)	⇒	22.48
Bande de Freq. 15	(2320	2700)	⇒	34.89
Bande de Freq. 16	(2700	3150)	⇒	24.20
Bande de Freq. 17	(3150	3700)	⇒	23.49
Bande de Freq. 18	(3700	4400)	⇒	22.33
Bande de Freq. 19	(4400	5300)	⇒	19.88
Bande de Freq. 20	(5300	6400)	⇒	16.49
Bande de Freq. 21	(6400	7700)	⇒	24.86
Bande de Freq. 22	(7700	9500)	⇒	27.25
Bande de Freq. 23	(9500	12000)	⇒	14.63
Bande de Freq. 24	(12000	15500)	⇒	9.54



Haute fréquence

Fig.36 : Calcul de RSB par bande pour une partie de signal(harpe) basse fréquence et une partie de haute fréquence

On constate le relativement bon comportement dans les bandes de haute énergie avec un RSB de l'ordre de 56 dB à 68 dB, mais aussi les faibles performances en très haute fréquence et avec des énergies faibles.

De la page 36
Le tableau supérieur concerne un extrait de basse fréquence de notes de harpe, et le tableau inférieur concerne un extrait de haute fréquence.

Cet extrait est présenté sur la figure 37 (en continu: original, en "bâtons verticaux", le signal codé/décodé).

La figure 38 montre le seul bruit de codage sur le même bloc de signal. La figure 39 montre les répartitions spectrales du signal (en continu) du bruit (en pointillé). Une autre courbe (en tirets) identifie les pentes de masquage selon Zwicker.

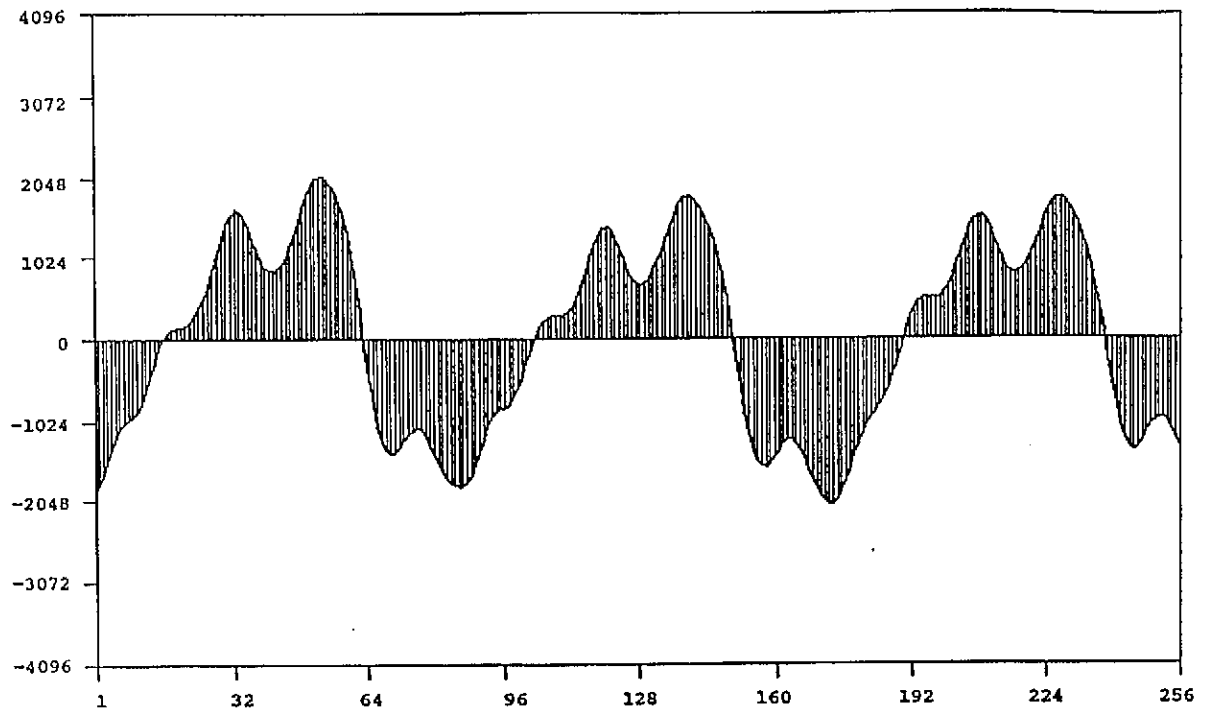


Figure 37 : Harpe et Zhfinal Bloc 850,1

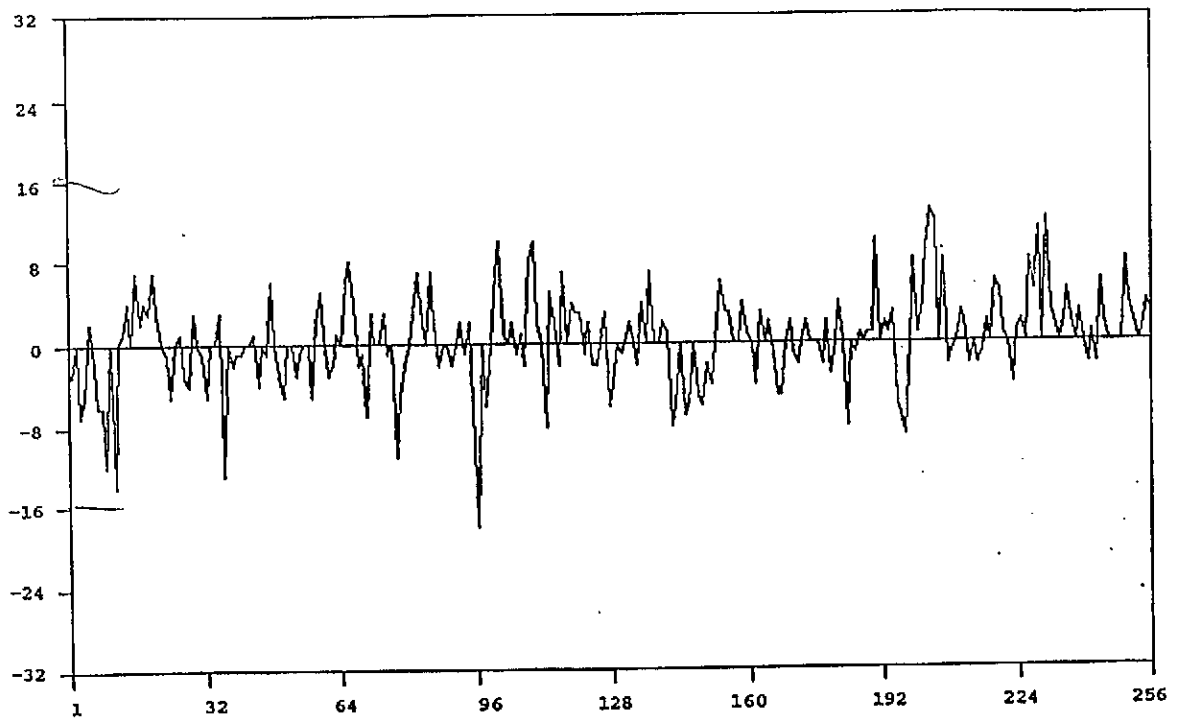


Figure 38 : Harpe bruit de codage par "quantification vectorielle"

On constate la "nature de bruit blanc" et le "passage" du bruit au-dessus des zones de masquage en hautes fréquences (Note: les tableaux présentent des résultats par bande qui intègrent plusieurs raies; il en résulte que le tableau ne présente pas de RSB négatifs même si la figure 39 montre parfois un bruit plus "fort" que le signal...)

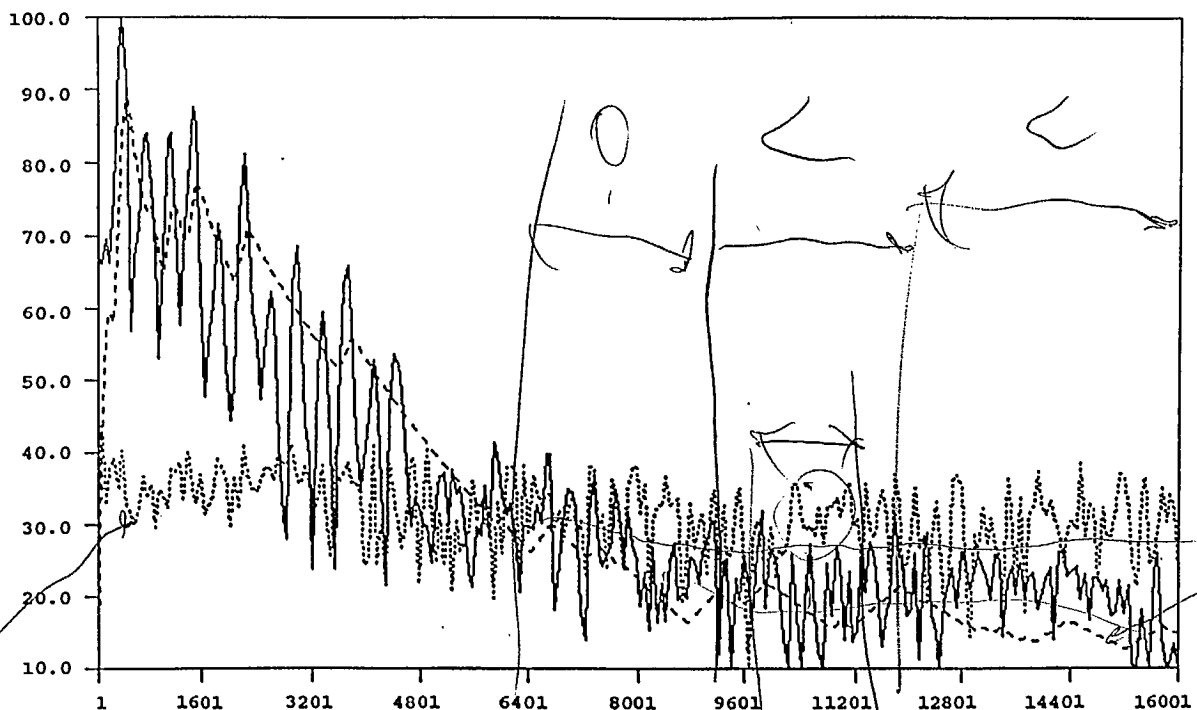


Figure 39 : Spectre du signal et du bruit (harpe avec basse fréquence)

Un autre essai, directement sur un bloc de hautes fréquences a donné les résultats du tableau inférieur de la figure 36.

On y relève que le RSB total est passé ici à seulement 36 dB (domaine fréquentiel) ou 32 dB (domaine temporel) à la place des 52 dB (ou 49 db) obtenus en basse fréquence.

Les figures 40, 41 et 42 montrent les mêmes types de résultat. On constate encore la présence de bruit blanc et des caractéristiques générales déjà citées.

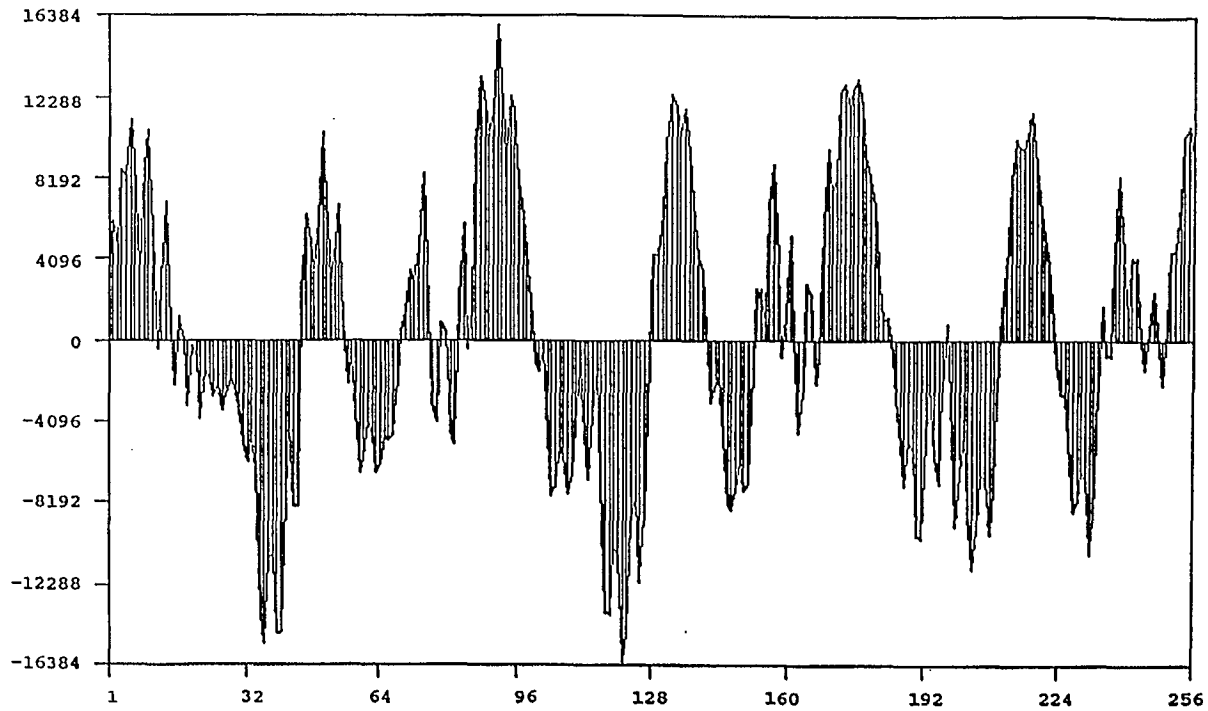


Figure 40 :Harpe et zhfinal bloc 987,1

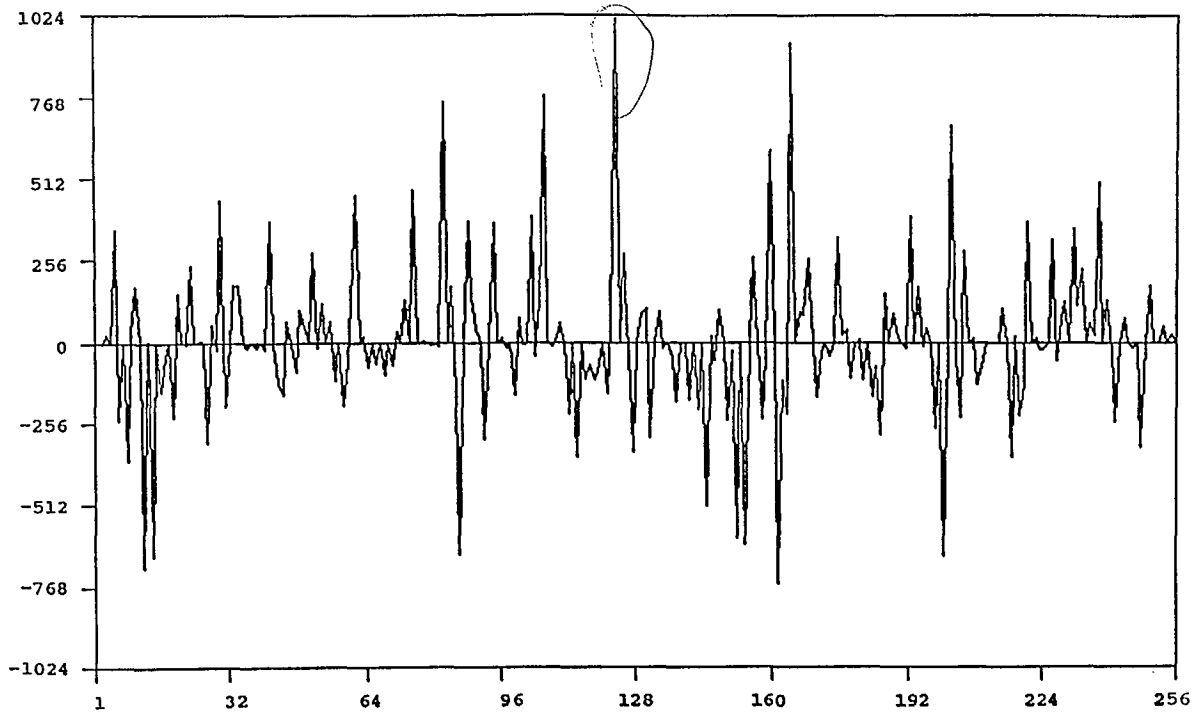


Figure 41 Harpe bruit de codage par "quantification vectorielle"

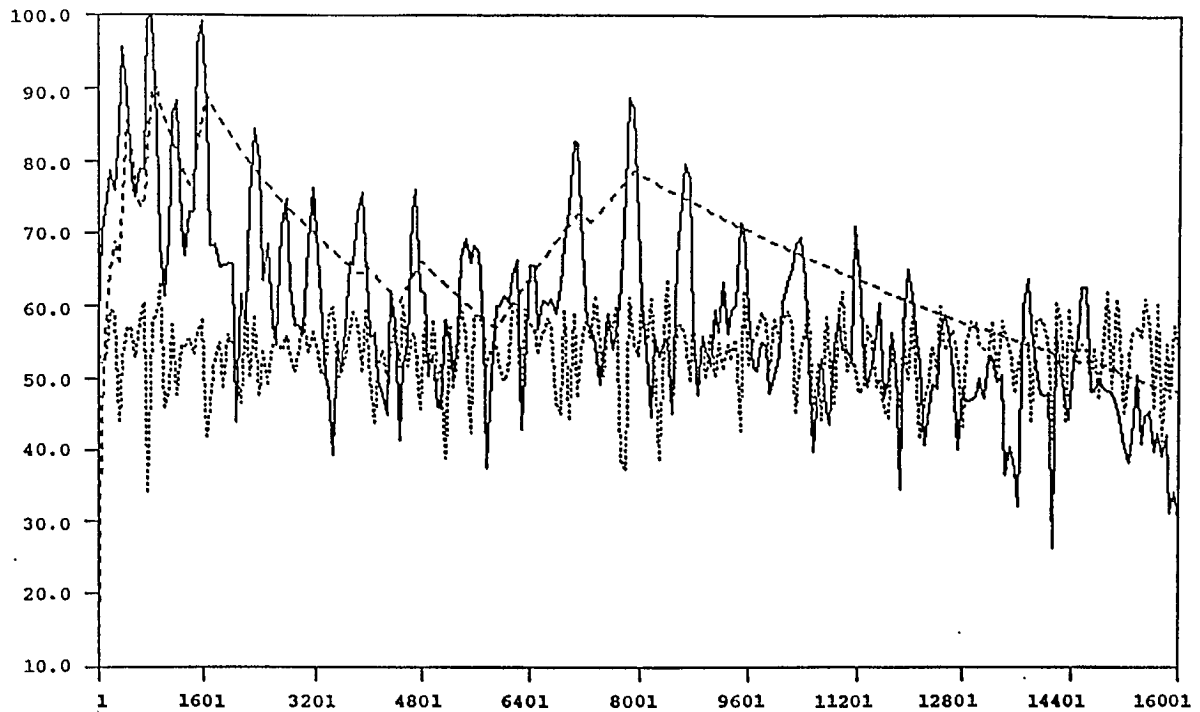


Figure 42 : Spectre du signal et de bruit (harpe avec haute fréquence)

Les figures 43, 44 et 45 présentent des résultats obtenus avec un codage d'un fichier de piano par le quantificateur vectoriel extrait d'une séquence (200 blocs) des fichiers de harpe (SHAR2).

On constate la similarité des performances (52 dB) en moyenne sur des signaux constitués surtout de basse fréquence. Les hautes fréquences présentent le même problème de qualité de codage.

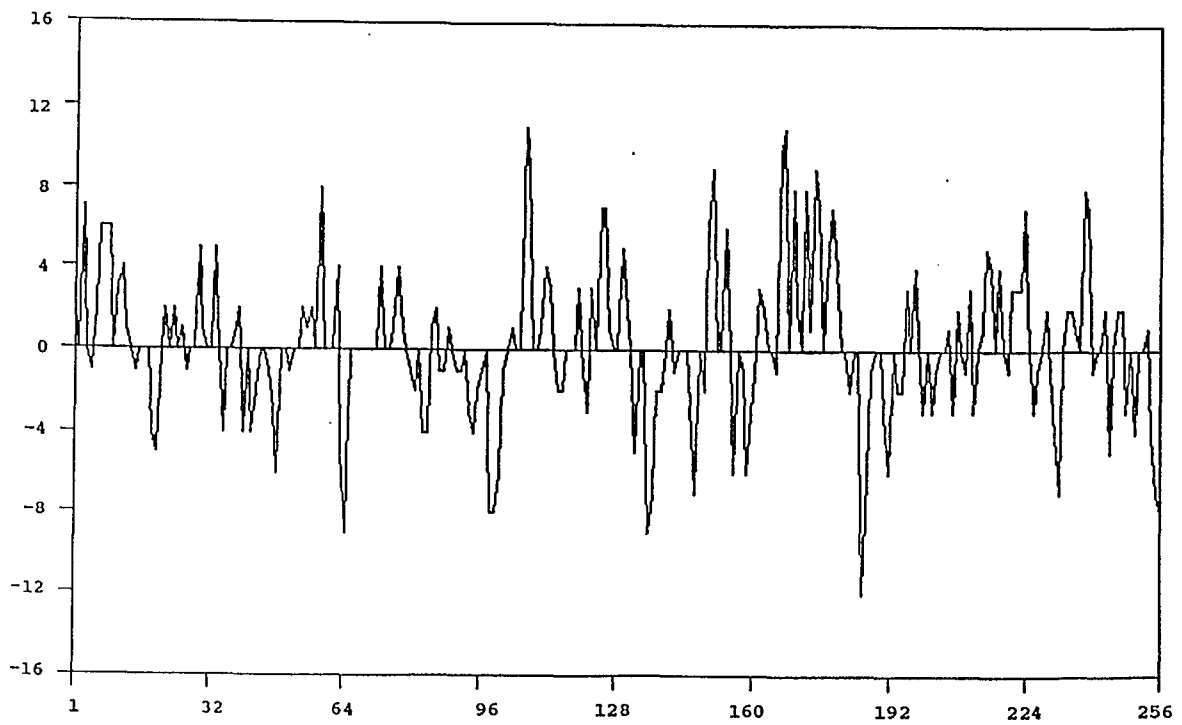


Figure 45 : Bruit de codage, piano, dict. de la harpe.

2.3.2 Résultats expérimentaux (notes synthétiques)

Une recherche de dictionnaire a été expérimentée à partir de notes synthétiques de basse fréquence, dans un premier temps. La note DO de fréquence 65,406 Hz a été choisie arbitrairement.

La figure 46 montre un dictionnaire construit avec 32 vecteurs de base de 8 échantillons chacun (en fait 8 périodes ...). La courbe "moyenne" (moyenne arithmétique) est aussi présentée sur la même figure.

Note: Seule la partie positive est représentée: donc la moitié des courbes.

La figure 47 montre un dictionnaire construit avec 64 courbes ainsi que la "moyenne".

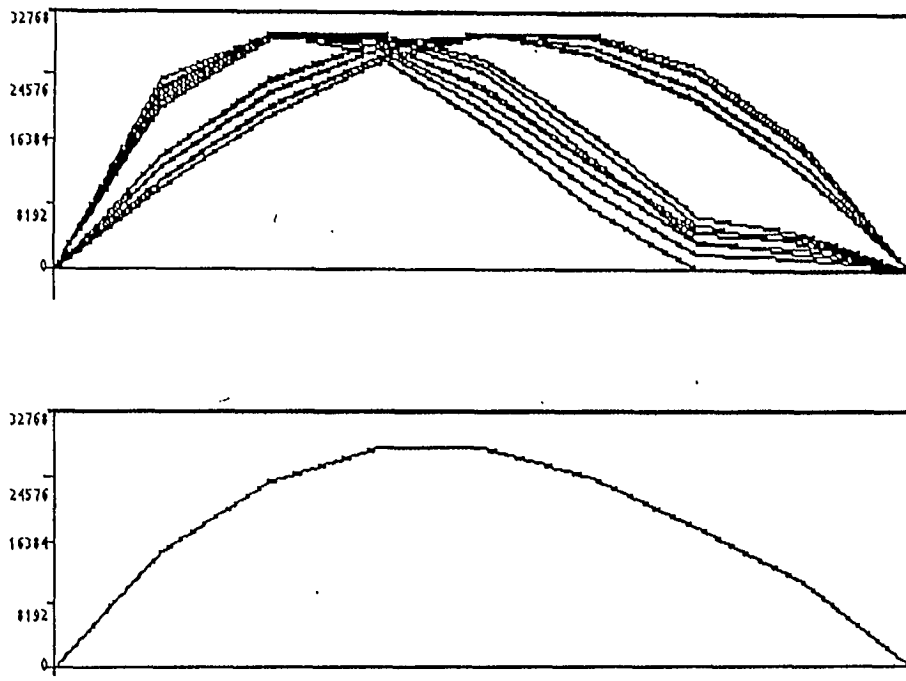


Figure 46: Note DO. Dictionnaire construit avec 32 vecteurs de base de 8 échantillons (a) et courbe moyenne (b).

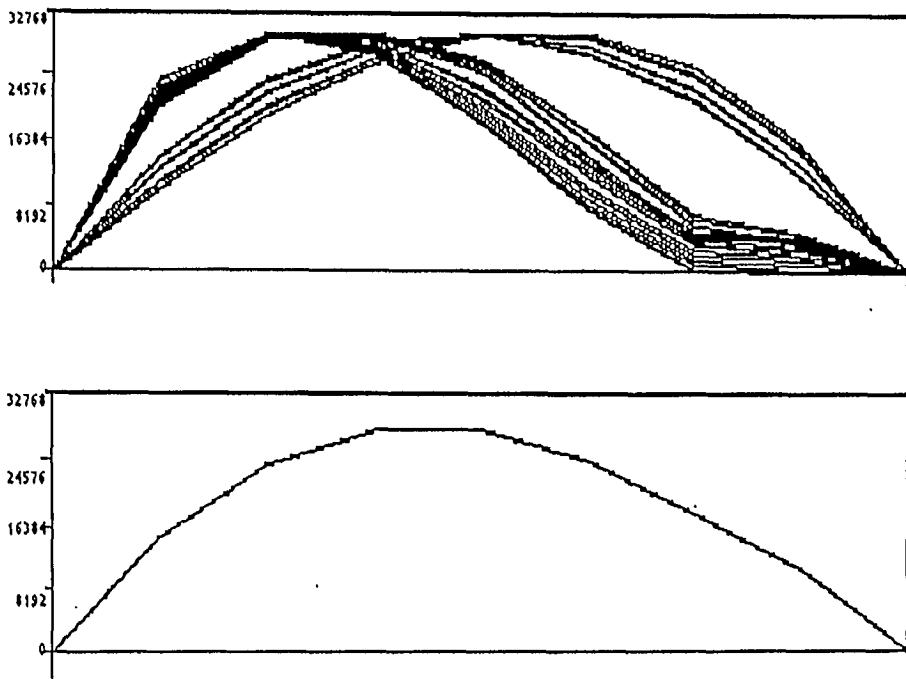


Figure 47: Note DO. Dictionnaire construit avec 64 courbes ainsi que la moyenne.

La figure 48 montre un dictionnaire de 256 courbes et leur "moyenne".

On constate sur ces figures l'uniformité des vecteurs (courbes) de base qui se limitent à deux formes générales, réparties par moitié de l'ensemble (plusieurs courbes sont pratiquement superposées,...).

On prévoit donc que la quantification vectorielle par ces dictionnaires de 32, 64, 256 et même plus ne doit pas donner d'amélioration sensible du codage étant donné la faible "dispersion" relative des courbes disponibles pour le codage. A la limite, deux courbes représentant chacune l'un des deux sous-ensembles, devraient pouvoir suffire au codage.

Un essai de codage, avec chacun de ces dictionnaires a fourni des résultats très similaires en terme de RSB (≈ 40 dB). La bande contenant la raie spectrale est, quant à elle, codée très efficacement avec un RSB de 62,4 dB comme le montre le tableau de la figure 49.

On constate aussi dans ce tableau de la figure 49 que les "bandes" latérales ne contiennent pas d'énergie (-60 à -70 dB). Cependant l'"étalement" du spectre du bruit, sur toutes les bandes, conduit à des RSB par bande très négatifs (-10 à -15 dB). Ainsi le bilan global sur tout le spectre ne donne qu'un RSB de 40 dB.

Les figures suivantes 50, 51 et 52 montrent successivement:

- la note - (dynamique ± 20000);
- l'erreur de codage - (dynamique ± 600);
- l'évolution du rapport signal sur bruit RSB

Les figures 53 et 54, puis 55 et 56 montrent le signal et l'erreur de codage sur des parties de la note d'origine. En particulier, la figure 56 montre le détail d'une erreur de codage "périodique" par blocs de 8 échantillons (entre deux zéros qui correspondent aux "supports" du signal d'origine et qui sont transmis (codés) séparément.

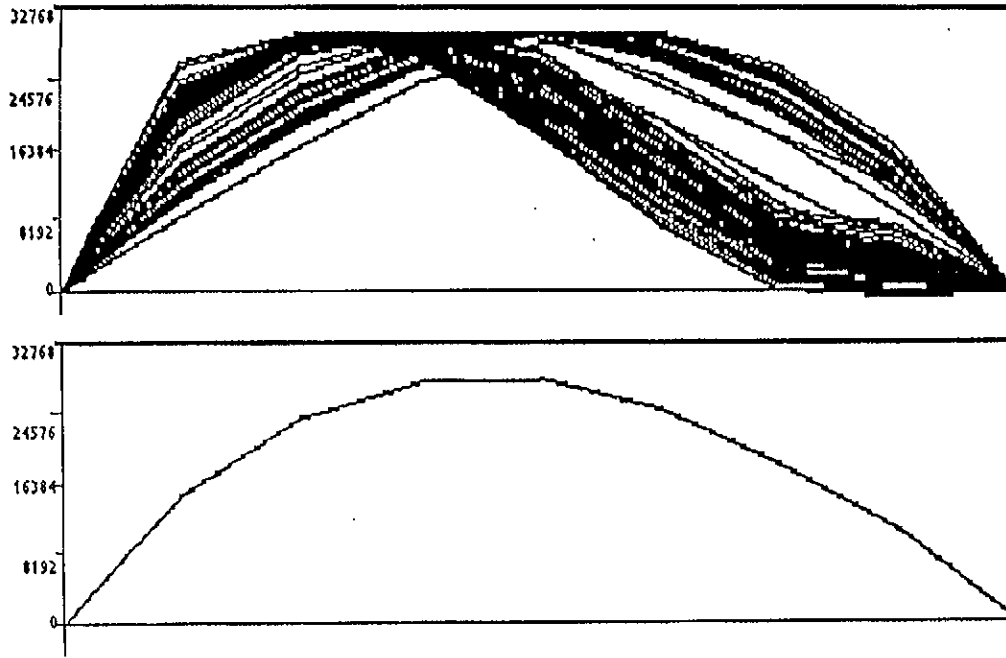


Figure 48: Note DO. Dictionnaire de 256 courbes ainsi que la moyenne.

```

FE = 32000.00 Hz Nbre de points 512 par 16 ms
Dynamique Max. du fichier "bruite": 32767 sous 16 bits
*****
RSB Classique = 38.04 dB 0 Seg. a bruit nul
RSB Segmentaire = 38.50 dB 0 Seg. a signal nul ( bruit moyen = 0 )
*****
Niveau moyen du Signal par rapport au maximum (saturation) : -17.17 dB Max.
*****
RSB Fréquentiel = 39.86 dB pour tout le spectre : bande unique
*****

```

	RSB freq./bande	Energie Relative (dB)
Bande de Freq. 1 (20 - 100) =>	53.84	-5.0
Bande de Freq. 2 (100 - 200) =>	62.40	0.0 -17.7 dB Max.
Bande de Freq. 3 (200 - 300) =>	24.75	-48.2
Bande de Freq. 4 (300 - 400) =>	9.19	-56.4
Bande de Freq. 5 (400 - 510) =>	-11.34	-56.6
Bande de Freq. 6 (510 - 630) =>	-11.99	-58.3
Bande de Freq. 7 (630 - 770) =>	-5.79	-59.9
Bande de Freq. 8 (770 - 920) =>	7.43	-61.3
Bande de Freq. 9 (920 - 1080) =>	-11.71	-61.0
Bande de Freq. 10 (1080 - 1270) =>	-6.73	-62.6
Bande de Freq. 11 (1270 - 1480) =>	-0.93	-63.9
Bande de Freq. 12 (1480 - 1720) =>	-12.20	-63.9
Bande de Freq. 13 (1720 - 2000) =>	-6.65	-64.4
Bande de Freq. 14 (2000 - 2320) =>	-11.20	-65.8
Bande de Freq. 15 (2320 - 2700) =>	-13.02	-66.2
Bande de Freq. 16 (2700 - 3150) =>	-14.57	-66.9
Bande de Freq. 17 (3150 - 3700) =>	-19.16	-67.1
Bande de Freq. 18 (3700 - 4400) =>	-9.30	-67.6
Bande de Freq. 19 (4400 - 5300) =>	-16.55	-68.0
Bande de Freq. 20 (5300 - 6400) =>	-7.96	-68.4
Bande de Freq. 21 (6400 - 7700) =>	-16.08	-69.2
Bande de Freq. 22 (7700 - 9500) =>	-14.32	-69.2
Bande de Freq. 23 (9500 - 12000) =>	-13.65	-69.1
Bande de Freq. 24 (12000 - 15500) =>	-13.41	-68.6

Figure 49: Note DO. Calcul du RSB par bande de fréquences.

La figure 48 montre un dictionnaire de 256 courbes et leur "moyenne".

On constate sur ces figures l'uniformité des vecteurs (courbes) de base qui se limitent à deux formes générales, réparties par moitié de l'ensemble (plusieurs courbes sont pratiquement superposées,...) On prévoit donc que la quantification vectorielle par ces dictionnaires de 32, 64, 256 et même plus ne doit pas donner d'amélioration sensible du codage étant donné la faible "dispersion" relative des courbes disponibles pour le codage. A la limite deux courbes représentant chacune l'un des deux sous-ensembles, devraient pouvoir suffire au codage. Un essai de codage, avec chacun de ces dictionnaires a fourni des résultats très similaires en terme de RSB (≈ 60 dB). La bande contenant la raie spectrale est, quant à elle, codée très efficacement avec un RSB de 66,6 dB comme le montre le tableau de la figure 49.

On constate aussi dans ce tableau de la figure 49 que les "bandes" latérales ne contiennent pas d'énergie (-60 à -70 dB). Cependant l'"étalement" du spectre du bruit, sur toutes les bandes, conduit à des RSB par bande de l'ordre de 10 à 20 db pour la plupart. Ainsi le bilan global sur tout le spectre ne donne qu'un RSB calculé sur les raies spectrales de près de 62 db..

Les figures suivantes 50, 51 et 52 montrent successivement:

- la note - (dynamique ± 20000);
- l'erreur de codage - (dynamique $\pm 16^*$);
- l'évolution du rapport signal sur bruit RSB

*Note: Des travaux ultérieurs ont montré que de fait ce bruit n'était dû qu'aux diverses opérations de filtrage. Rappelons que le fichier est suréchantillonné par $R = 8$, donc filtré une première fois. Ensuite le codage vectoriel étant effectué sur un fichier échantillonné à 32 kHz et le fichier définitif étant produit à 44 kHz, il y a un deuxième filtrage. C'est le "bruit de calcul" de ces filtrages qui est prépondérant. L'erreur spécifique de quantification vectorielle sur cette note Basse Fréquence (65 kHz) ne dépasse pas ± 1 .

Les figures 53 et 54, puis 55 et 56 montrent le signal et l'erreur de codage sur des parties de la note d'origine. En particulier, la figure 56 montre le détail d'une erreur de codage "périodique" par blocs de 8 échantillons (entre deux zéros qui correspondent aux "supports" du signal d'origine et qui sont transmis (codés) séparément.

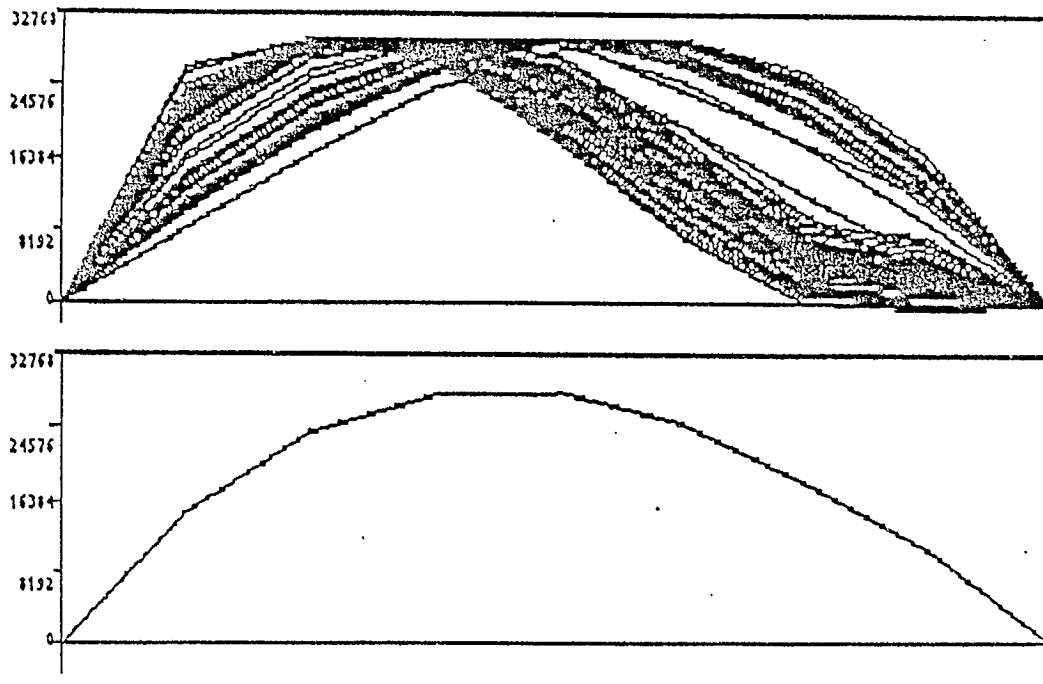


Figure 48: Note DO. Dictionnaire de 256 courbes ainsi que la moyenne.

```

-----
FE = 32000.00 Hz Nbre de points 512 par 16 ms
Dynamique Max. du fichier "bruite": 32767 sous 16 bits
*****

RSB Classique = 60.36 dB 0 Seg. a bruit nul
RSB Segmentaire = 38.65 dB 0 Seg. a signal nul ( bruit moyen = 0 )
*****
Niveau moyen du Signal par rapport au maximum (saturation) : -18.19 dB Max.
*****
RSB Frequentiel = 61.73 dB pour tout le spectre : bande unique
*****

```

	RSB freq./bande	Energie Relative (dB)
Bande de Freq. 1 (20 - 100) =>	59.60	-6.0
Bande de Freq. 2 (100 - 200) =>	66.56	0.0 -18.7 dB Max.
Bande de Freq. 3 (200 - 300) =>	29.42	-47.5
Bande de Freq. 4 (300 - 400) =>	22.22	-55.7
Bande de Freq. 5 (400 - 510) =>	23.33	-55.8
Bande de Freq. 6 (510 - 630) =>	22.49	-57.5
Bande de Freq. 7 (630 - 770) =>	21.24	-59.1
Bande de Freq. 8 (770 - 920) =>	20.57	-60.5
Bande de Freq. 9 (920 - 1080) =>	19.28	-60.2
Bande de Freq. 10 (1080 - 1270) =>	18.23	-61.7
Bande de Freq. 11 (1270 - 1480) =>	17.26	-63.0
Bande de Freq. 12 (1480 - 1720) =>	15.97	-63.1
Bande de Freq. 13 (1720 - 2000) =>	15.25	-63.6
Bande de Freq. 14 (2000 - 2320) =>	15.21	-64.9
Bande de Freq. 15 (2320 - 2700) =>	15.14	-65.4
Bande de Freq. 16 (2700 - 3150) =>	15.08	-66.0
Bande de Freq. 17 (3150 - 3700) =>	15.06	-66.3
Bande de Freq. 18 (3700 - 4400) =>	13.49	-66.8
Bande de Freq. 19 (4400 - 5300) =>	10.65	-67.2
Bande de Freq. 20 (5300 - 6400) =>	9.15	-67.6
Bande de Freq. 21 (6400 - 7700) =>	9.03	-68.4
Bande de Freq. 22 (7700 - 9500) =>	6.94	-68.3
Bande de Freq. 23 (9500 - 12000) =>	6.08	-68.3
Bande de Freq. 24 (12000 - 15500) =>	5.49	-67.8

Taper <RETURN> pour continuer

Figure 49: Note DO. Calcul du RSB par bande de fréquences.

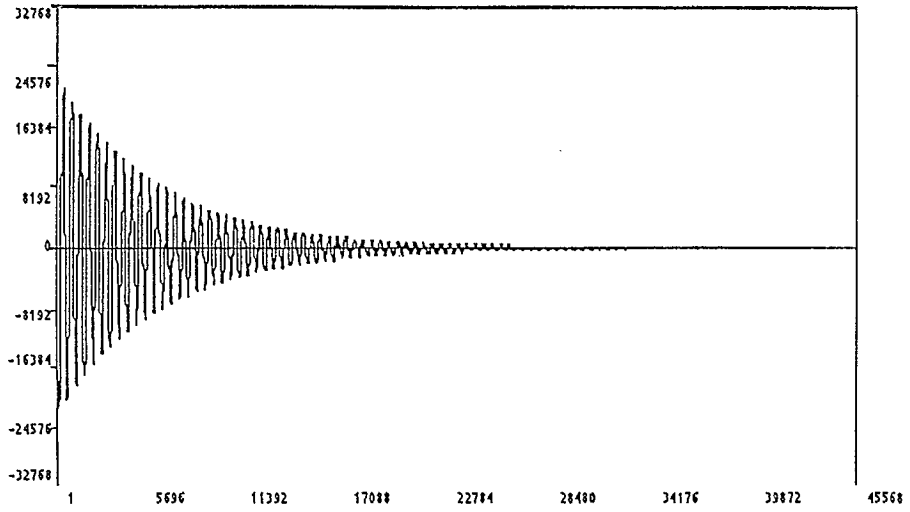


Figure 50: Note DO. (Dictionnaire de 32 vecteurs); dynamique $\pm 20,000$.

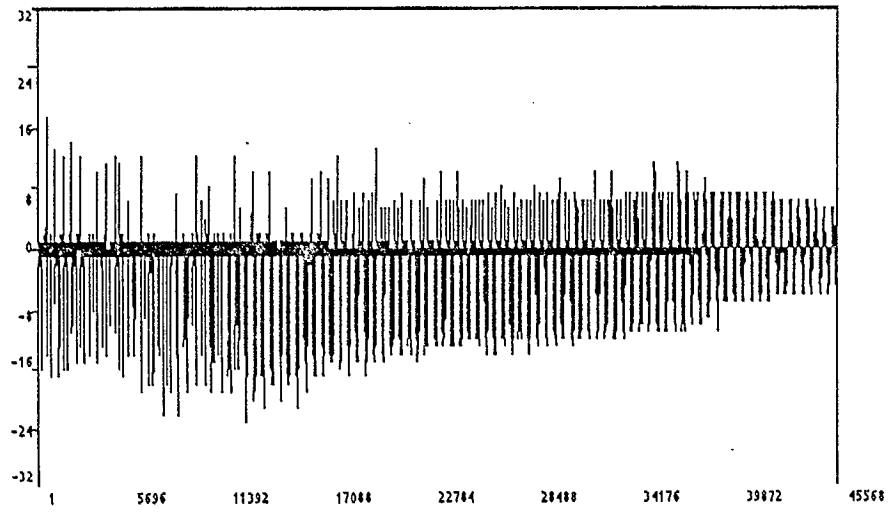


Figure 51: Erreur de codage: dynamique ± 16 .

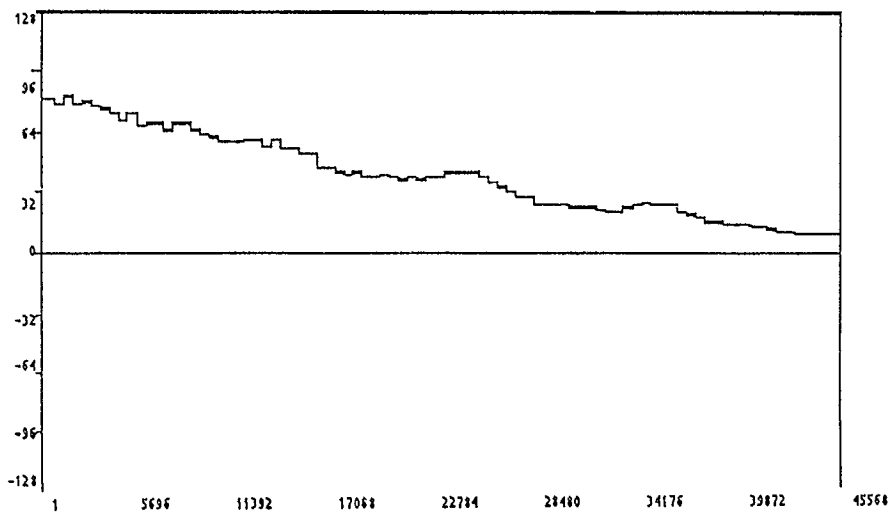


Figure 52: Evolution du rapport signal sur bruit RSB.

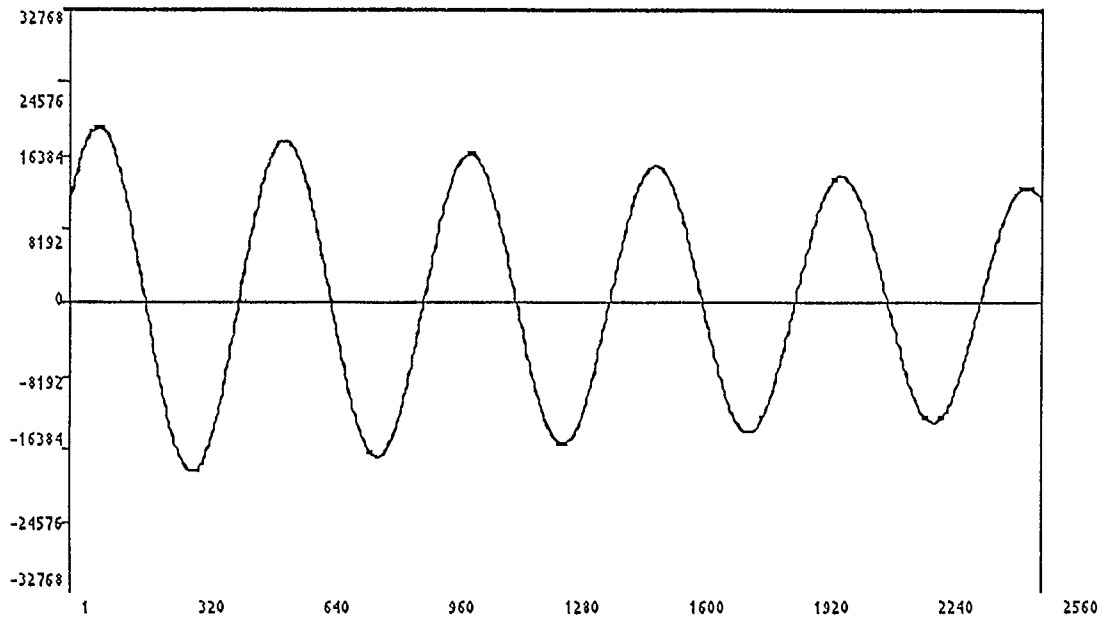


Figure 53: Note DO. Parties du signal d'origine.

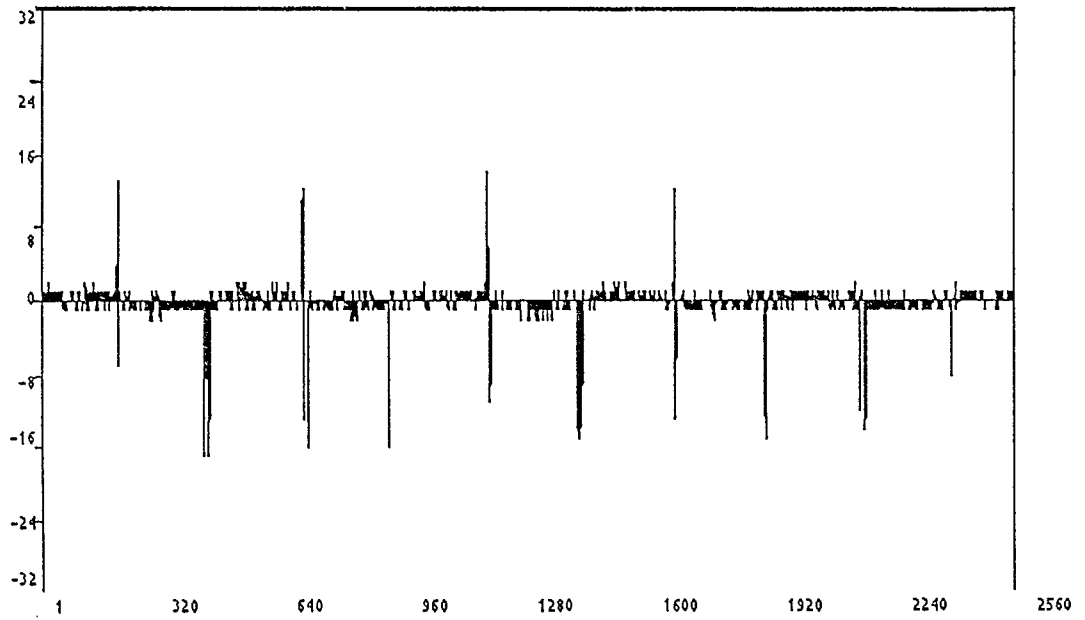


Figure 54: Erreur de codage.

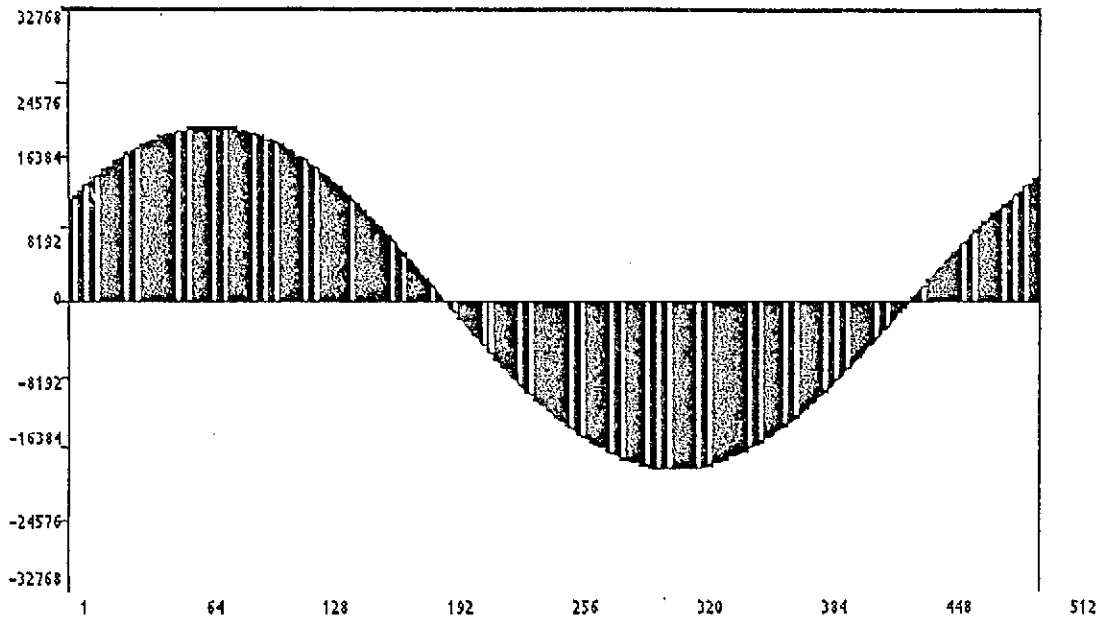


Figure 55: Note DO. Signal original et codé.

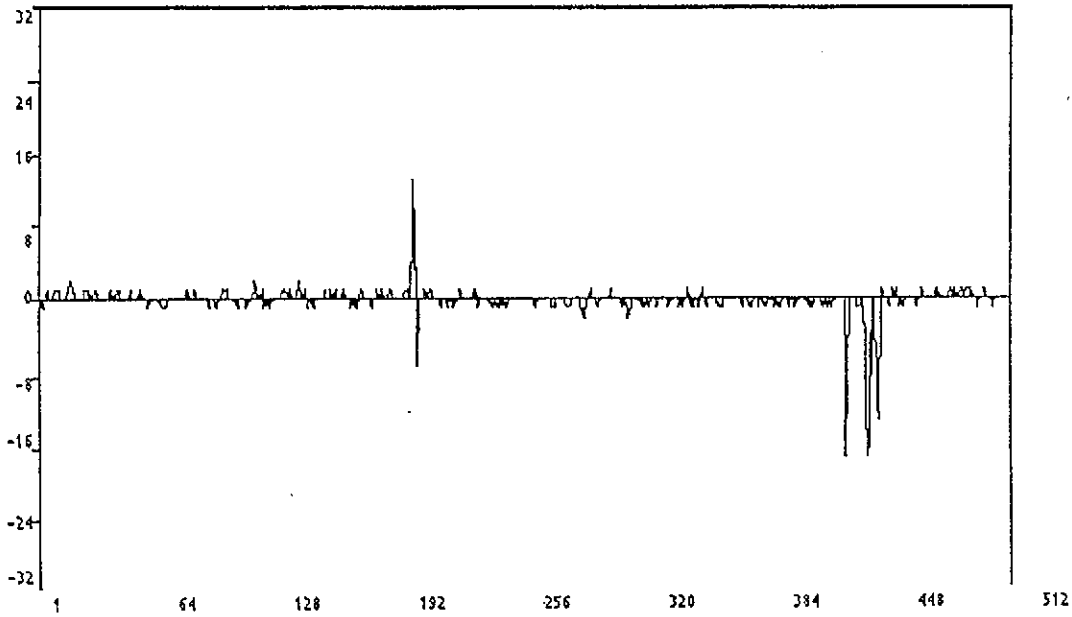


Figure 56: Détail d'une erreur de codage

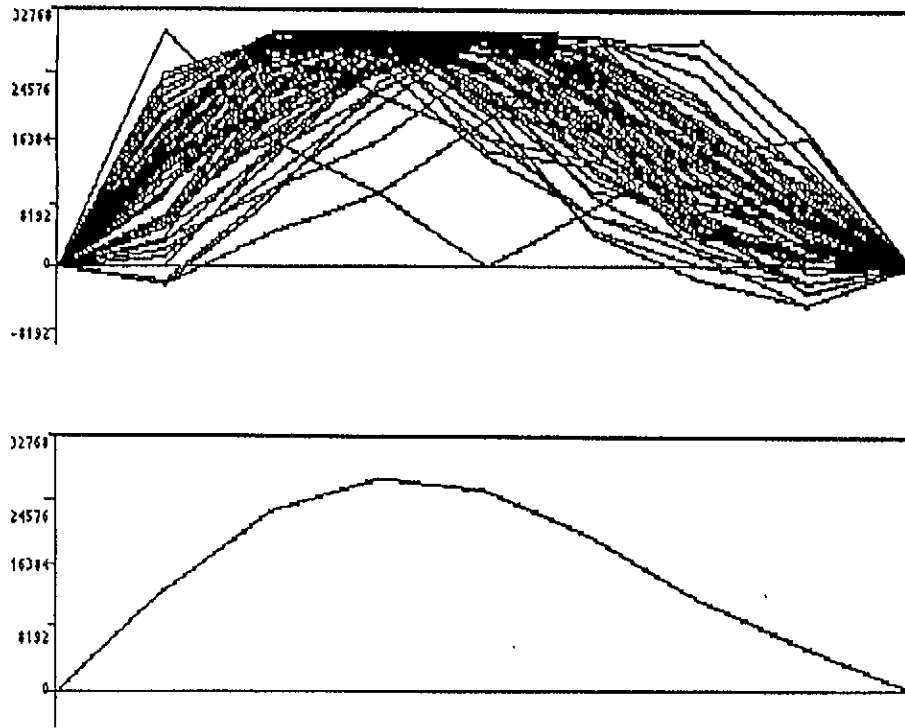


Figure 59: Vecteurs de 8 périodes d'échantillonnage et moyenne (dict. de 64 vecteurs).

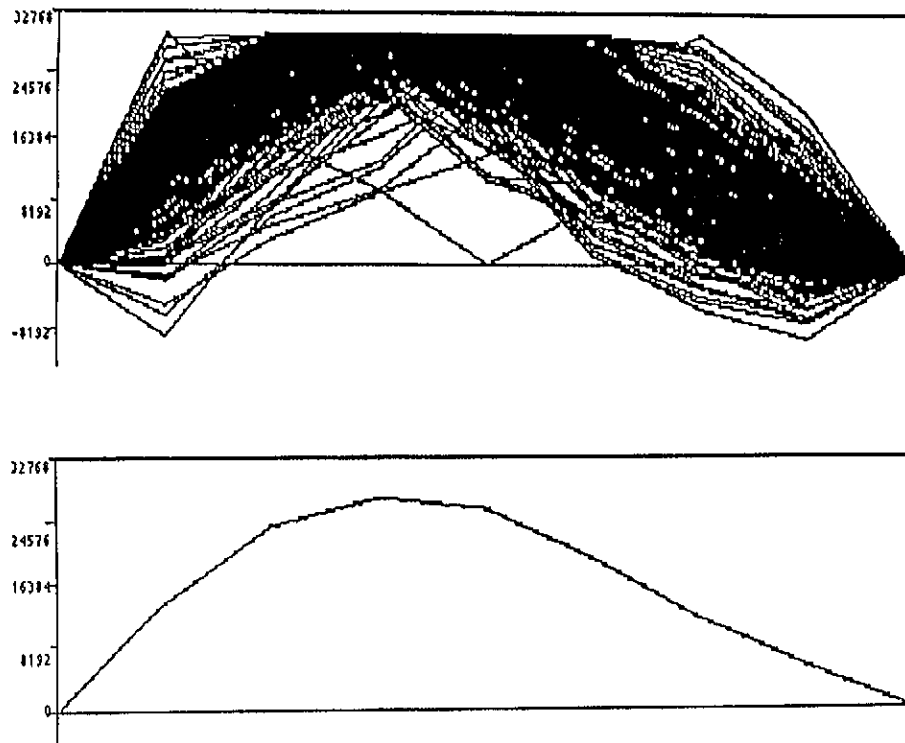


Figure 60: Dictionnaire de 256 vecteurs.

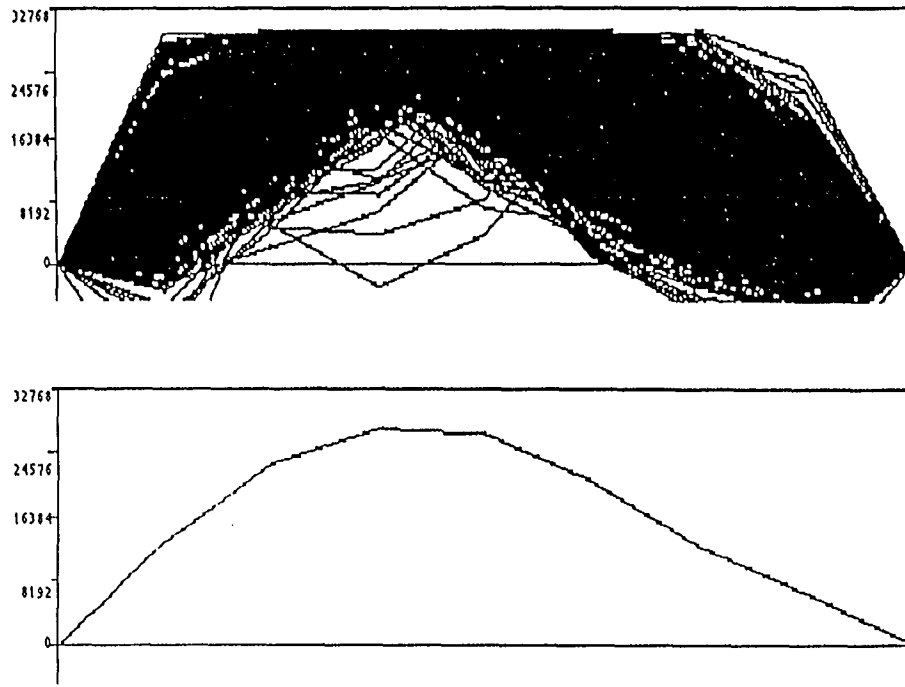


Figure 61: Dictionnaire de 1024 vecteurs.

3. Conclusion

Bien que prématuré de conclure, il apparaît utile de proposer des voies d'études pour améliorer le comportement général de cette technique:

- Le choix du dictionnaire initial et de la séquence d'apprentissage pour application de l'algorithme de k-moyenne paraît prépondérant. De nombreuses "déconvenues" sont dues à de mauvais choix. Il faut approfondir la connaissance de ces facteurs.

- Les hautes fréquences semblent lourdement pénalisées dans cette technique. Il y a lieu d'envisager de les traiter différemment. Par exemple avec un dictionnaire spécifique approprié, construit sur des vecteurs plus courts ou plus longs!

B: CODAGE DIFFÉRENTIEL "QUASI-INSTANTANÉ"

Proposé par le Japon (CCIR - document 10/52-E - 17 juillet 1987), cette technique de codage reprend des idées déjà connues.

- a) Le signal de "différence filtré" présente une dynamique fortement réduite (surtout en basse fréquence);

De la sorte, il nécessite un débit binaire plus faible à qualité de codage égale;

- b) Le codage quasi-instantané, connu par la proposition NICAM 3 de la B.B.C., présente une qualité de codage accrue pour les signaux de haute dynamique.

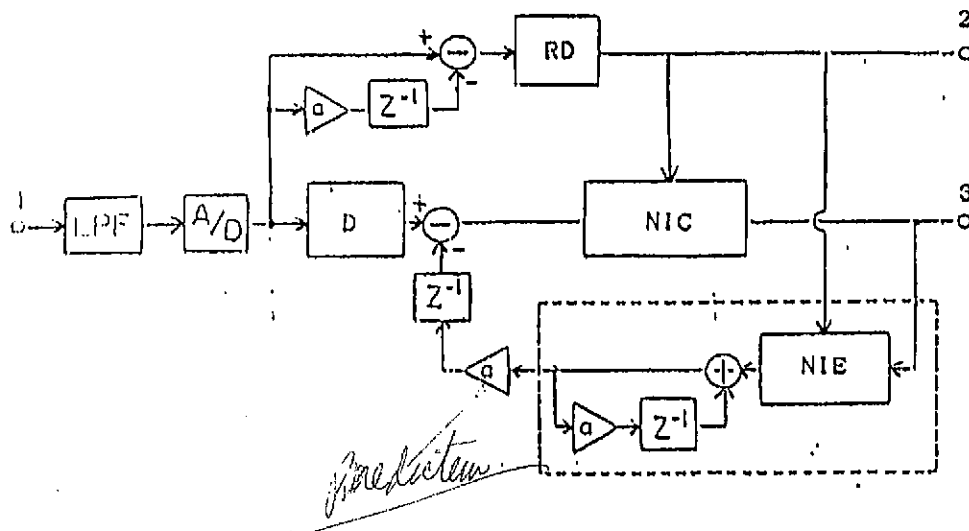
Une combinaison de deux techniques devrait fournir une qualité au moins égale tout en autorisant une réduction du débit numérique.

Un programme de simulation a été complètement conçu pour cette version particulière. En se référant à des rapports antérieurs, on construit un codeur à 8 "gammes" ou segments de codage.

- Chaque bloc de 32 échantillons se voit affecter un numéro (de 1 à 8) sous 3 bits, caractérisant la dynamique (ou gain du quantificateur) du plus grand échantillon du bloc.
- Chaque échantillon, formé en fait de la différence entre le signal d'entrée et l'échantillon "précédent reconstitué et filtré", est alors quantifié sur une échelle linéaire de 8 bits.
- Le filtrage précité assure en fait un facteur de pertes qui garantit l'évanouissement des effets des erreurs de transmission éventuelles.

On notera que la sélection du numéro de gamme de codage est faite à partir de la différence (filtrée elle aussi) des vrais échantillons d'entrée consécutifs; et non pas à partir d'un quelconque échantillon reconstitué. Ce fait est imposé par la conception d'ensemble et ne pose aucun problème étant donné le niveau de qualité obtenu par un tel codeur (50 à 60 dB).

La figure 62 présente le concept d'ensemble.



- | | |
|------------------------------------|------------------------------------|
| 1: audio signal input | D: 1 block length delay |
| 2: compression range signal | NIC: near instantaneous compressor |
| 3: compressed differential signal | NIE: near instantaneous expander |
| LPF: low pass filter | LD: local decoder |
| A/D: analogue to digital converter | a: leak factor |
| RD: compression range detector | z ⁻¹ : delay unit |

Figure 62: Bloc diagramme du codeur différentiel quasi-instantané (extrait du document CCIR 10/52-E)

1 Fontionnement du codeur: musique naturelle

On a choisi de présenter un extrait de harpe et pour l'illustration graphique, on a sélectionné une "attaque de note". Cette attaque présente à la fois la dynamique maximale de

tout le fichier (l'amplitude maximale atteint 16800) et à la fois une variation d'amplitude entre échantillons consécutifs très importante (de l'ordre de 10,000 au moment de l'attaque).

La figure 63 présente la note de harpe. La figure 64 présente les échantillons codés et transmis en ligne (par blocs de 32). Rappelons que ces échantillons sont formés de la différence entre échantillons consécutifs (échantillons original d'entrée et échantillons reconstitués et filtrés). Pour le quantificateur utilisé, avec 8 bits, la plage dynamique de sortie s'étend donc de 128 à +127 (le zéro étant un mot de code inclus).

La figure 65 montre l'évolution du numéro de "gamme (ou segment; *range* en anglais). Chaque bloc de 32 échantillons est associé à une "gamme" de quantification J . Le quantificateur correspondant permet alors une dynamique maximale pour le signal d'entrée de 2^{7+J}

La présentation des 3 figures montre l'évolution dans le temps de ces paramètres.

Les figures 66, 67 et 68 montrent les mêmes caractéristiques mais avec une échelle dilatée.

L'échelle encore plus dilatée des figures 69, 70 et 71 montrent avec plus de précision ces divers signaux. On y remarque la "nature bruitée" (mais encore influencée par la périodicité de la note de musique) du signal codé et transmis. La figure 70, plus particulièrement, met ce résultat en évidence.

Il est utile de rappeler que le "filtre" de différence constitue aussi, pour le codeur, un "prédicteur" fixe d'ordre 1. En théorie un prédicteur optimal donne un signal bruité de spectre "blanchi" en sortie mais il devrait bien évidemment être d'ordre beaucoup plus élevé [1]¹. Il peut être intéressant de rechercher un coefficient de filtrage (et donc de prédiction) adapté aux signaux. Une première analyse simple a été faite avec des notes synthétiques.

¹ [1] Jayant, N.S., Noll, P., "Digital Coding of Waveforms", Ed. Prentice Hall, 1984.

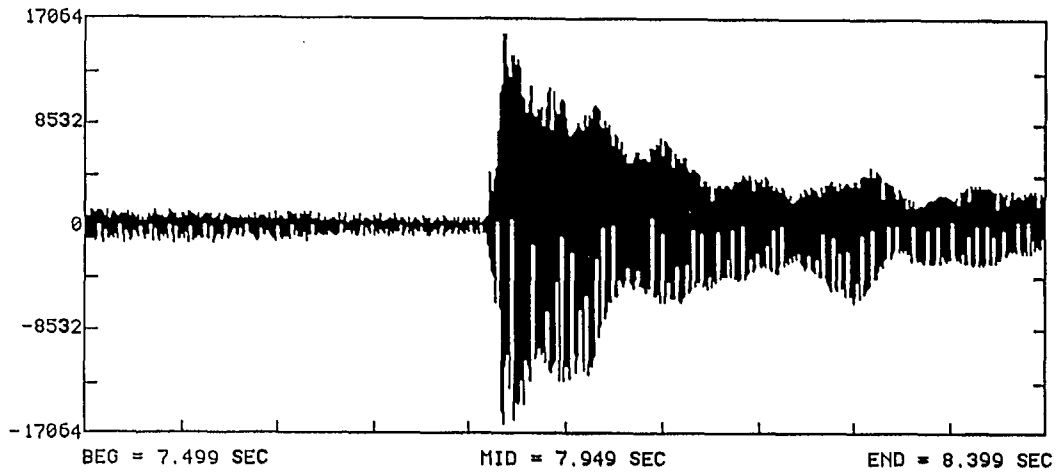


Figure 63: "Attaque" d'une note de harpe: signal de musique.

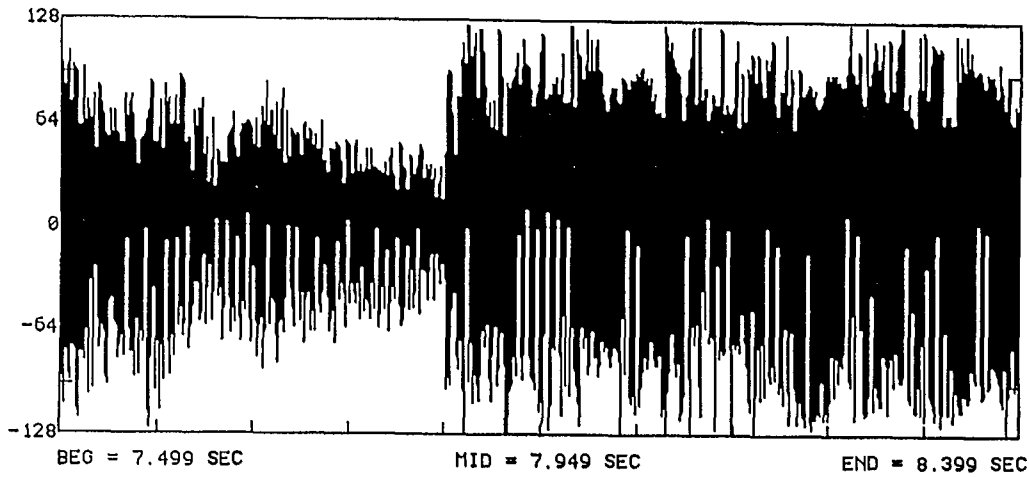


Figure 64: "Attaque" d'une note de harpe: signal de différence (filtrée) codé et transmis.

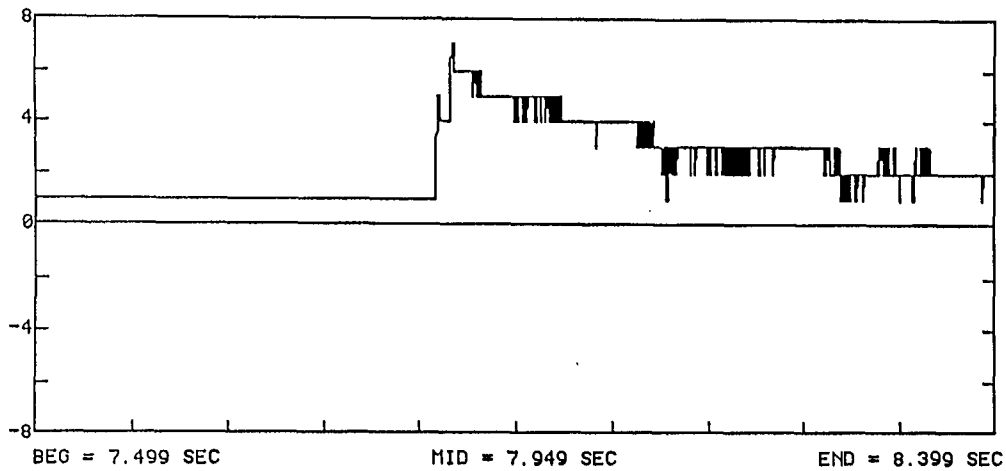


Figure 65: "Attaque" d'une note de harpe: numéro de gamme (gain du quantificateur linéaire) du codage par bloc.

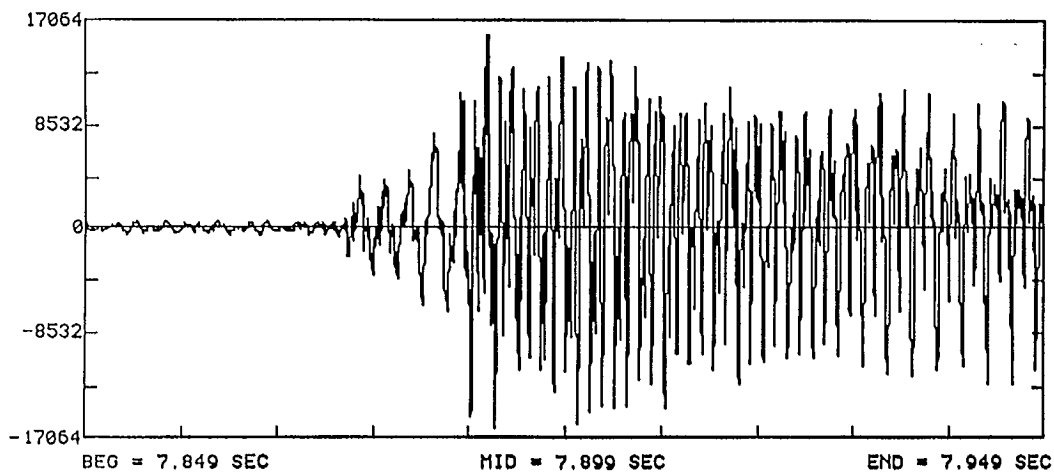


Figure 66: "Attaque" d'une note de harpe: signal de musique.

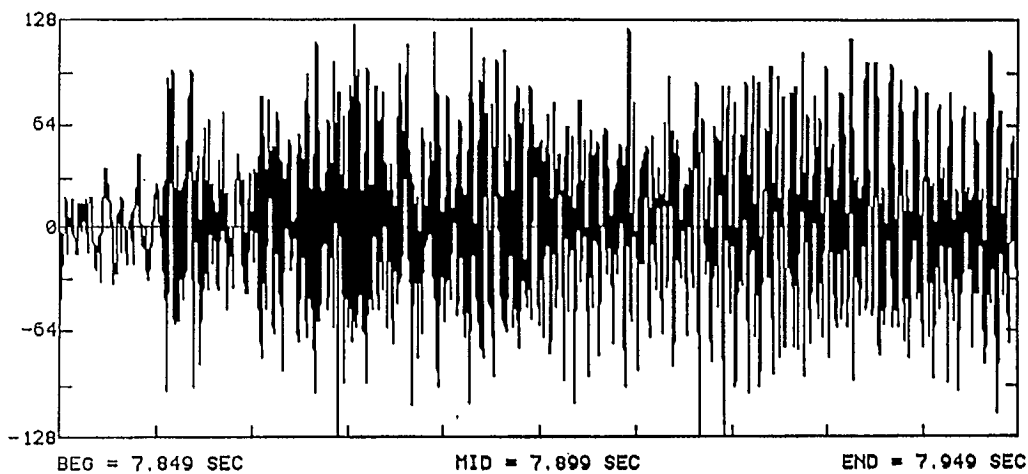


Figure 67: "Attaque" d'une note de harpe: signal de différence (filtrée) codé et transmis.

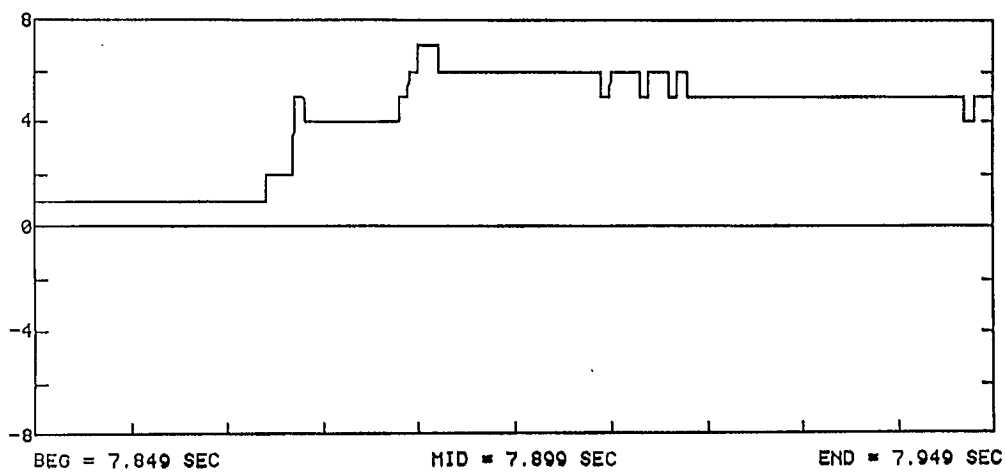


Figure68: "Attaque" d'une note de harpe: numéro de gamme (gain du quantificateur linéaire) du codage par bloc.

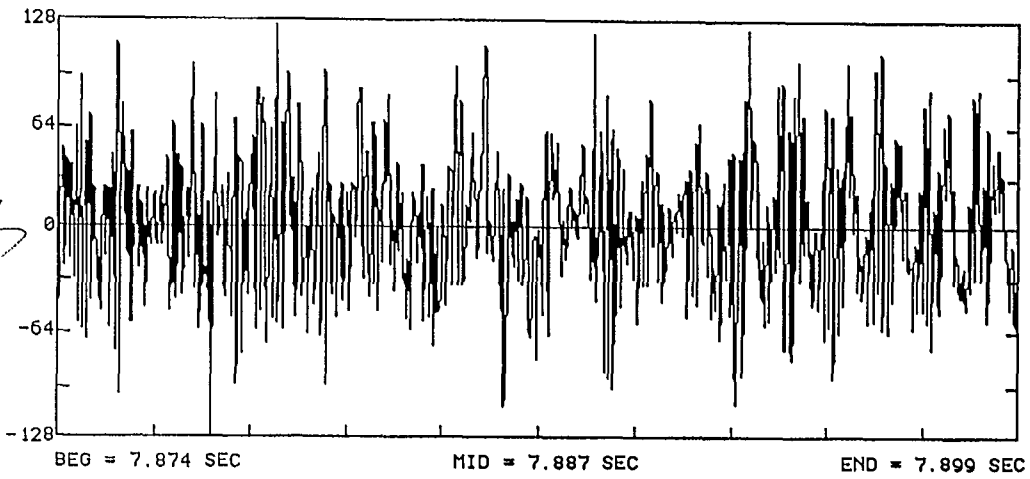


Figure 69: "Attaque" d'une note de harpe: signal de musique.

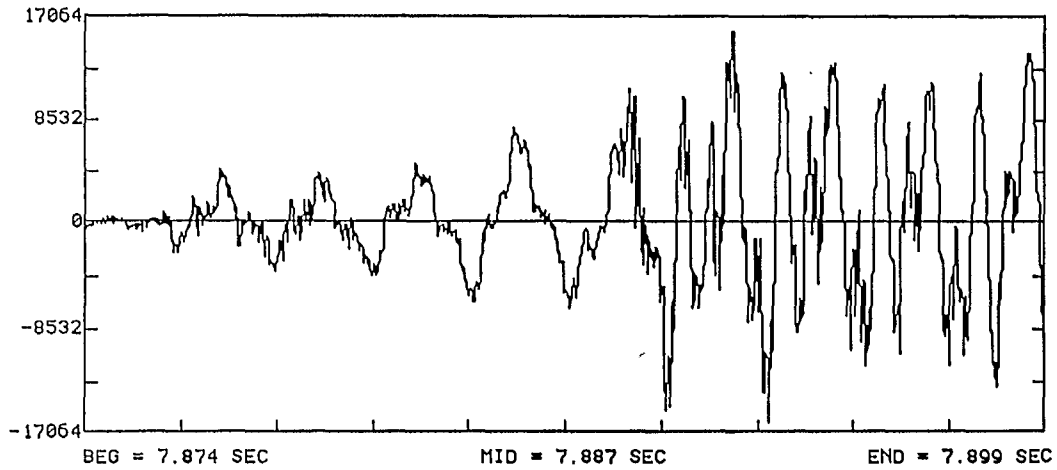


Figure 70: "Attaque" d'une note de harpe: signal de différence (filtrée) codé et transmis.

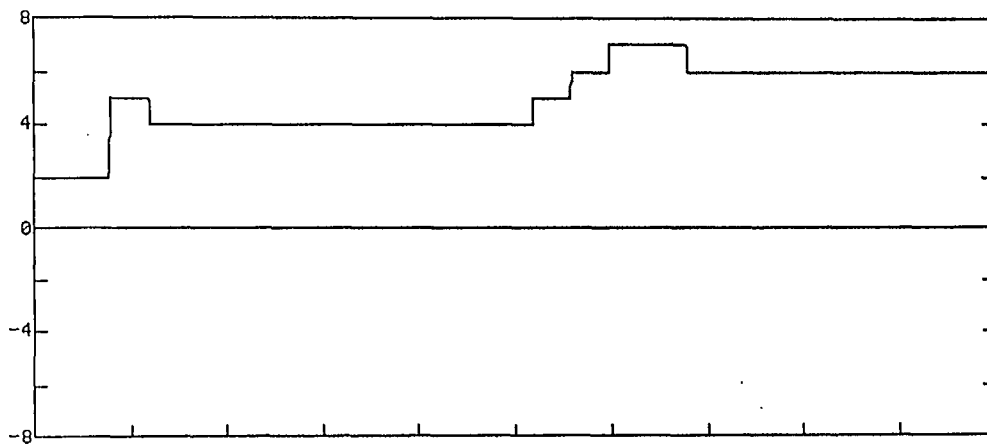


Figure 71: "Attaque" d'une note de harpe: numéro de gamme (gain du quantificateur linéaire) du codage par bloc.

2 Fonctionnement du codeur - notes synthétiques

On a choisi les notes synthétiques (DO - RE - MI - DO) "Frère Jacques..."
 en basse (60 - 80 Hz) et haute fréquence (4 kHz - 6 kHz).

La figure 72 montre une séquence de notes et la dynamique choisie.

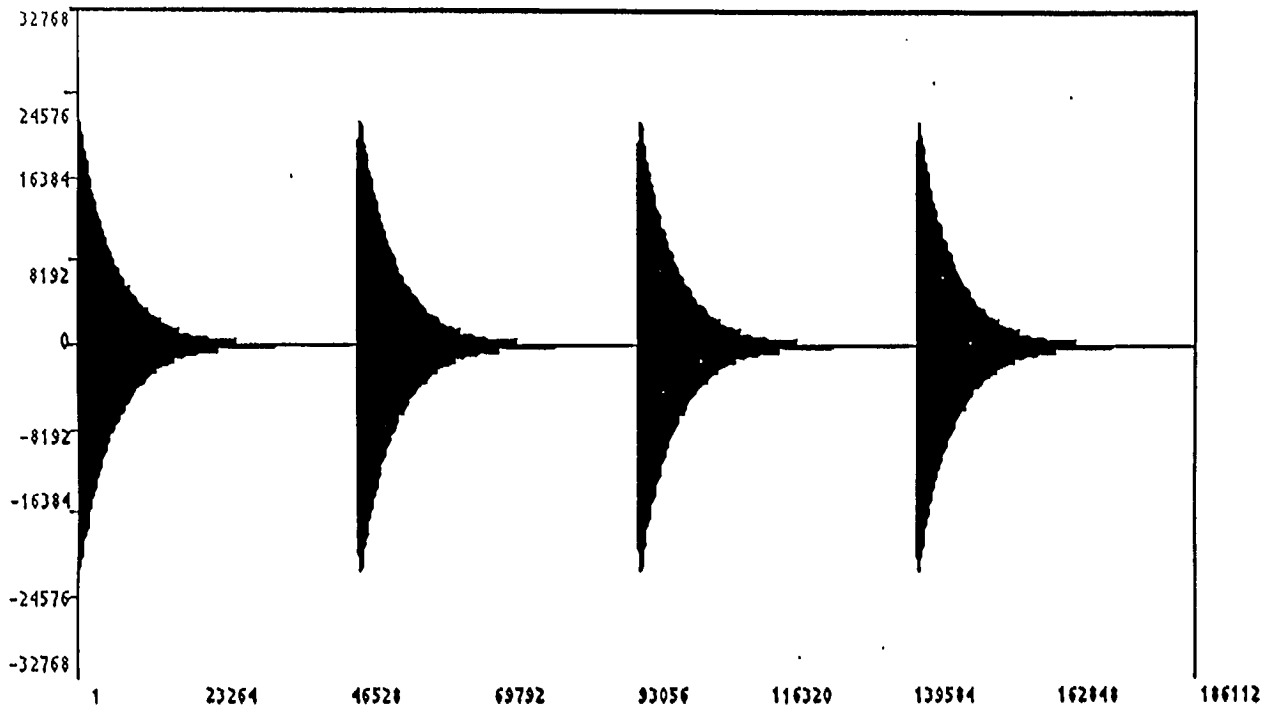


Figure 72: Notes synthétiques DO - RE - MI - DO.

2.1 Facteur de prédiction "a"

Le filtre numérique, présent au décodeur local sur la figure 1, de fonction de transfert:

$$H(z) = \frac{1}{1 - az^{-1}}$$

a été étudié avec un coefficient "a" variable, strictement pour des signaux sinusoïdaux.

La figure 73 montre un fichier de dix séquences de sinusoïdes de fréquence variable dans la bande audio.

Avec un codeur de type "Différentiel par Bloc" à 8 bits de codage (et 8 gammes), on a obtenu l'ensemble des résultats des figures suivantes.

La figure 74 (a, b et c), 75 (a, b, c et d) montrent l'évolution de la qualité de codage déterminée par le Rapport Signal sur Bruit (RSB) en fonction du paramètre "a". Ce coefficient variable a montré une très nette dégradation des performances "Basse-Fréquence" pour des valeurs inférieures à 0,900. Les "Hautes-Fréquences" sont, quant à elles, peu sensibles à la variation de "a".

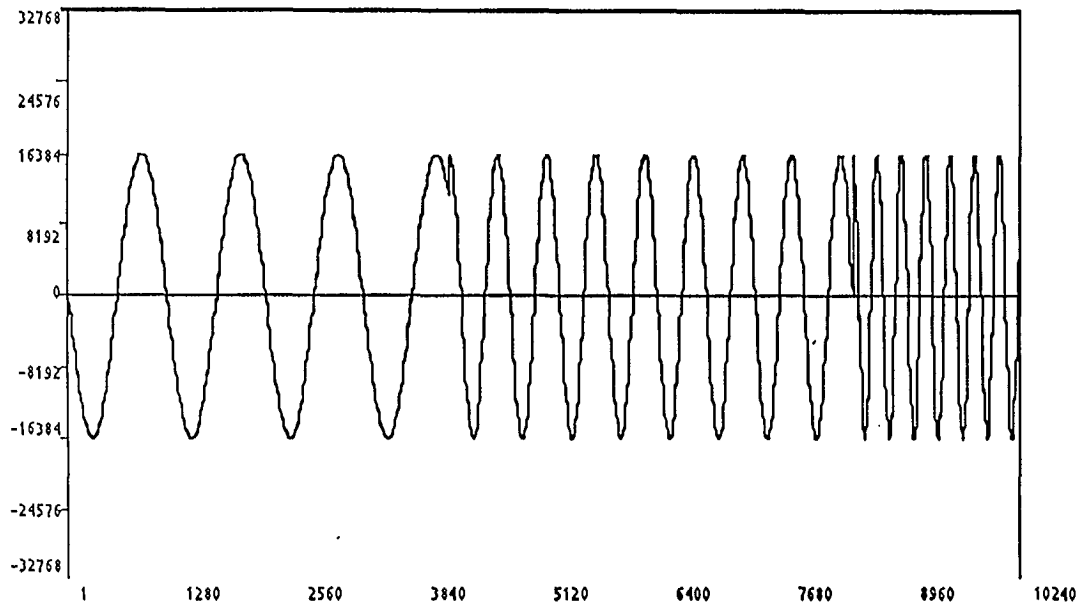


Figure 73a: Extrait d'un fichier de sinusoïdes de fréquence variées. 30 Hz, 60 Hz et 120 Hz (échantillonnées à 32 kHz).

```

0
2.* Pi = 6.283185
fréquence de la sinusoïde : 30.31835
fréquence de la sinusoïde : 60.63670
fréquence de la sinusoïde : 121.2734
fréquence de la sinusoïde : 242.5468
fréquence de la sinusoïde : 485.0936
fréquence de la sinusoïde : 970.1872
fréquence de la sinusoïde : 1940.374
fréquence de la sinusoïde : 3880.749
fréquence de la sinusoïde : 7761.498
fréquence de la sinusoïde : 15523.00
$
    
```

Figure 73b: Tableau des fréquences choisies.

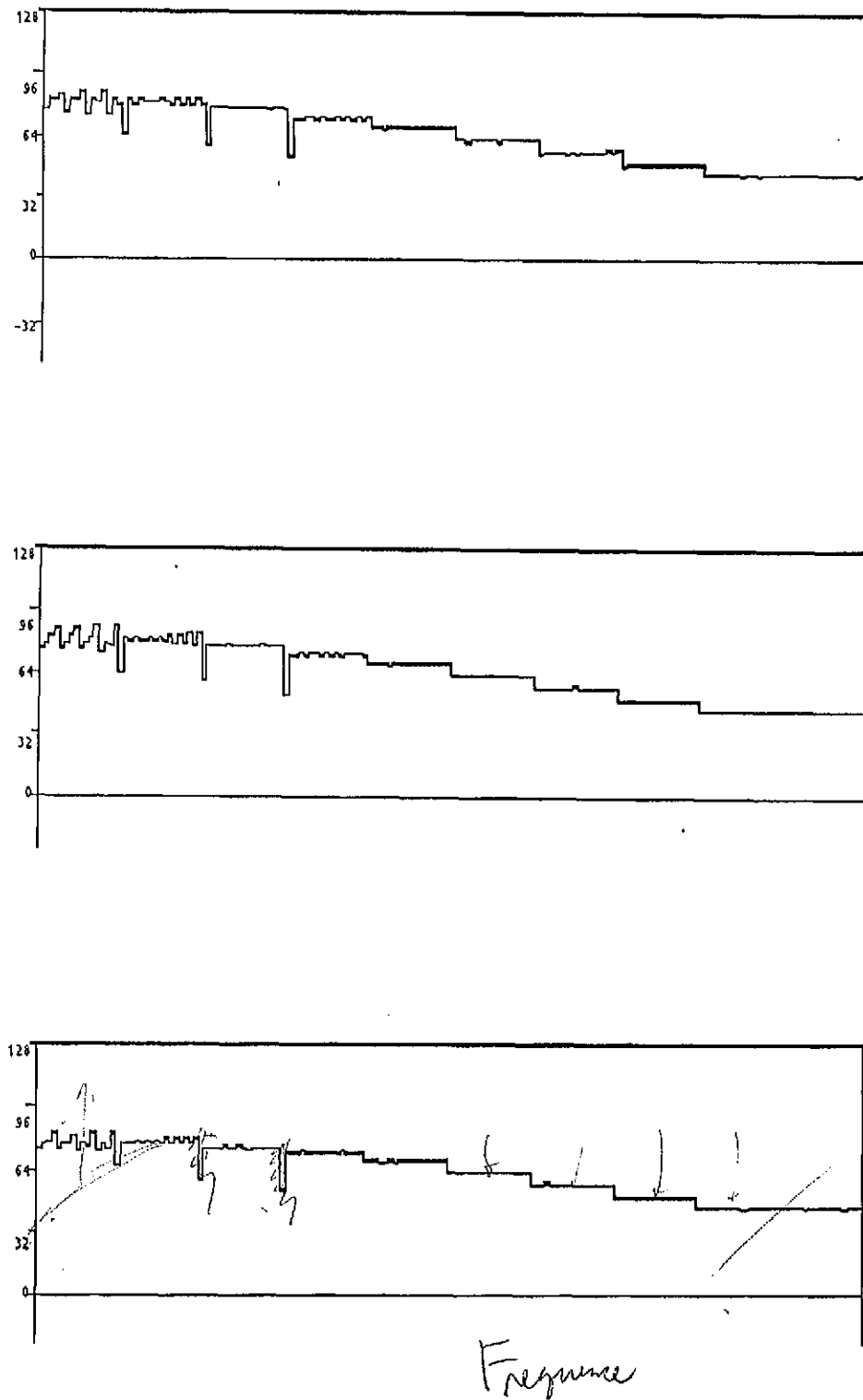


Figure 74: Codeur différentiel (quasi-instantané) par bloc avec 8 bits de codage. RSB en fonction de la fréquence (30 Hz à 15 kHz) pour un facteur de prédiction "a".
 a) "a" = 0,999; b) "a" = 0,990; c) "a" = 0,975.

meilleure prediction sur BF
 mais propagation des erreurs

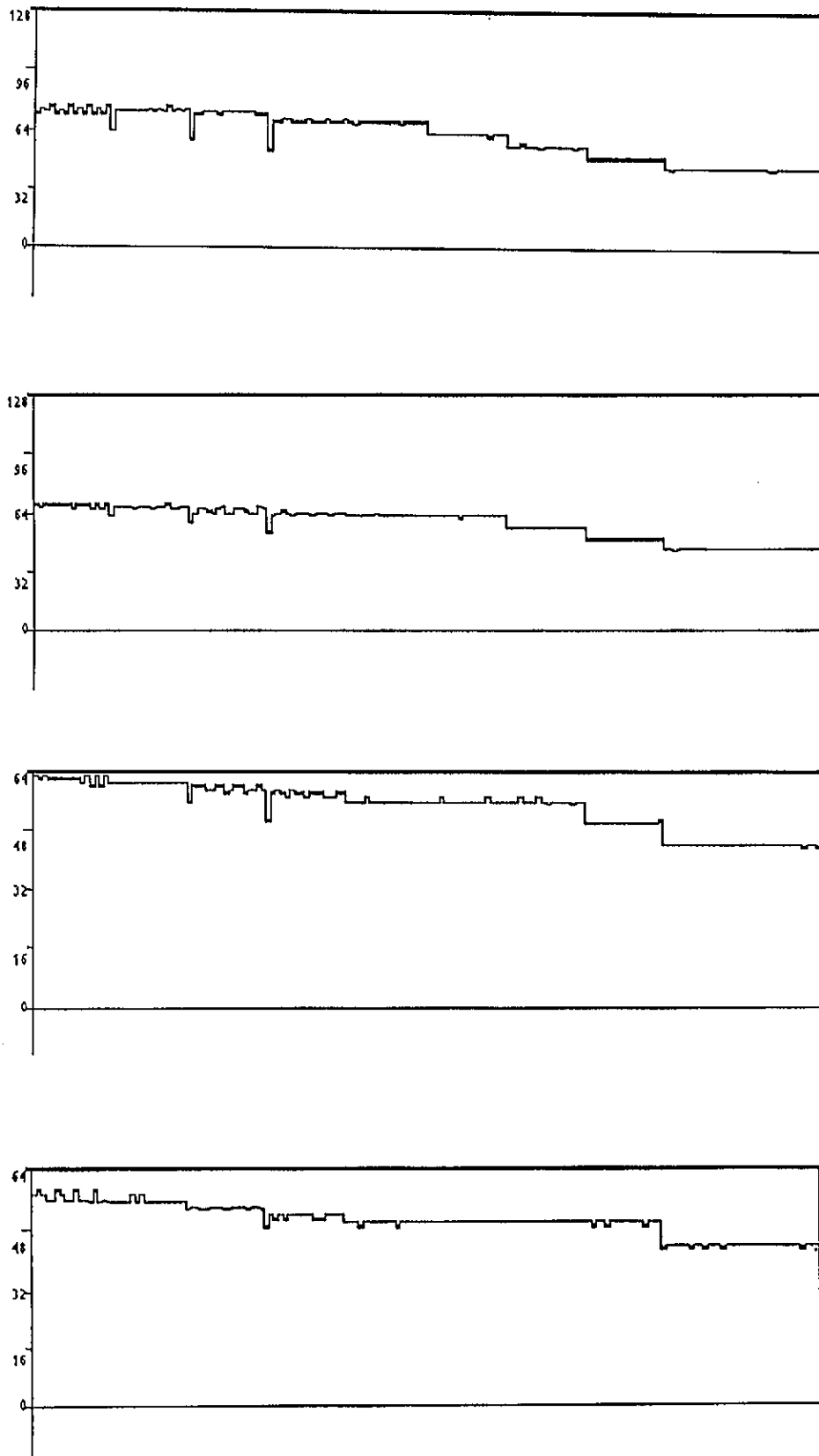


Figure 75: Codeur différentiel (quasi-instantané) par bloc avec 8 bits de codage. RSB en fonction de la fréquence (30 Hz à 15 kHz) pour un facteur de prédiction "a".
 a) "a" = 0,950; b) "a" = 0,875; c) "a" = 0,750; d) "a" = 0,500.

On choisit pour les simulations à venir:

$$a = 0,975$$

- pour assurer un bon niveau de performance en Basse-Fréquence (70 dB) environ;
- pour assurer *simultanément* un facteur de pertes suffisant pour le filtre $H(z)$ placé au décodeur distant. Rappelons que ce filtre atténue la propagation des erreurs de transmission par un effet de filtrage passe-bas.

Enfin, à titre de comparaison, le même fichier de 10 sinusoïdes (figure 73) a été testé avec un codeur de type **NICAM 3 à 10 bits de codage**.

La figure 76 confirme que la quantification instantanée est totalement indépendante de la fréquence du signal (de 30 Hz à 15 kHz). Seul le niveau d'amplitude affecte le RSB, et seulement pour les petits niveaux.

On retiendra, de la comparaison des figures 74, 75 avec la figure 76, un comportement du Codeur Différentiel (avec prédiction d'ordre 1) nettement supérieur en fréquence basse (jusque vers 1 kHz) mais une dégradation sensible au-delà. Toutefois en termes subjectifs, il n'est pas assuré que cette dégradation soit audible.

On conservera, pour la musique naturelle un facteur de prédiction

$$"a" = 0,975$$

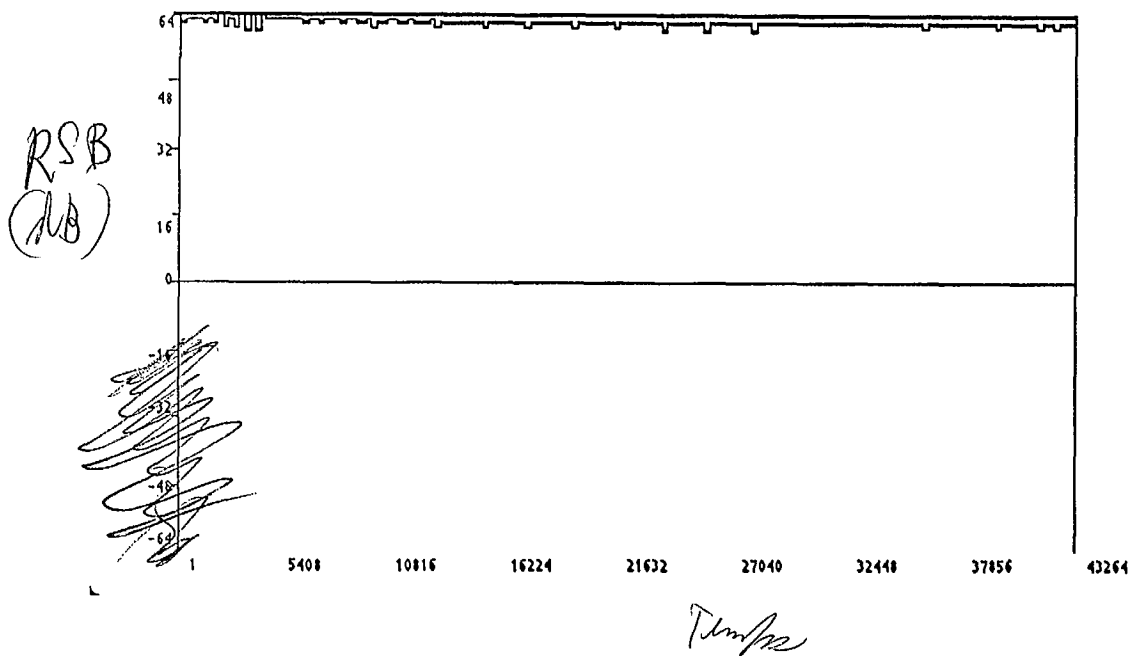


Figure 76: Codeur par bloc (quasi instantané) NICAM 3 avec 10 bits de codage: RSB en fonction de la fréquence (30 Kz à 15 kHz). (Il n'y a pas de facteur de prédiction "a").

et l'on rappelle un résultat connu de la théorie de la prédiction [1]¹ en ce qui concerne le RSB global de codage:

$$RSB_{global} = RSB_{\text{quantificateur}} + G_p$$

8 bits
 $\frac{a}{\Delta}$

où

- $RSB_{\text{quantificateur}}$ est la qualité de codage du quantificateur à 8 bits .
- G_p est le gain de prédiction, en dB, mesurant la compression de dynamique entre le signal original et le signal de différence (filtré).

pour expliquer les variations en fonction de la fréquence relevées sur les courbes des figures 74 et 75.

¹ [1] Jayant, N.S., Noll, P., "Digital Coding of Waveforms", Ed. Prentice Hall, 1984.

2.2 Rapports Signal sur Bruit: notes et musique

a) *Note synthétique basse fréquence*

La figure 77 montre le RSB calculé avec un codeur différentiel sur des notes de basse fréquence, alors que la figure 78 montre un codage direct (type NICAM).

On notera, de plus, que sur la figure 77, on n'utilisait que 8 bits de codage (codage différentiel, par blocs) et sur la figure 78, on en utilisait 10 (codage direct NICAM 3, par blocs).

On relève la très nette supériorité du codeur différentiel, (malgré 2 bits de moins).

b) *Note synthétique haute fréquence*

Par contre, en haute fréquence, on obtient les résultats des figures 79 et 80.

On rappellera que le système NICAM 3 donne un RSB indépendant de la fréquence (figures 76, 78 et 80).

c) *Musique naturelle : harpe*

Enfin, pour un morceau de harpe (figure 81), les résultats chiffrés sont assez comparables.

Les figures 82 et 83 montrent, en comparaison, l'évolution du Rapport Signal sur Bruit (calculé par segments de 256 points consécutifs - soient 4 blocs de 32 points). On y constate des courbes de RSB très sensiblement voisines avec toutefois un léger avantage pour le codeur différentiel (à 8 bits), notamment au moment des pointes de dynamique du signal. Il est cependant impossible d'affirmer qu'un tel avantage objectif puisse être significatif au niveau subjectif. Il est impératif de former une séquence de tests subjectifs.

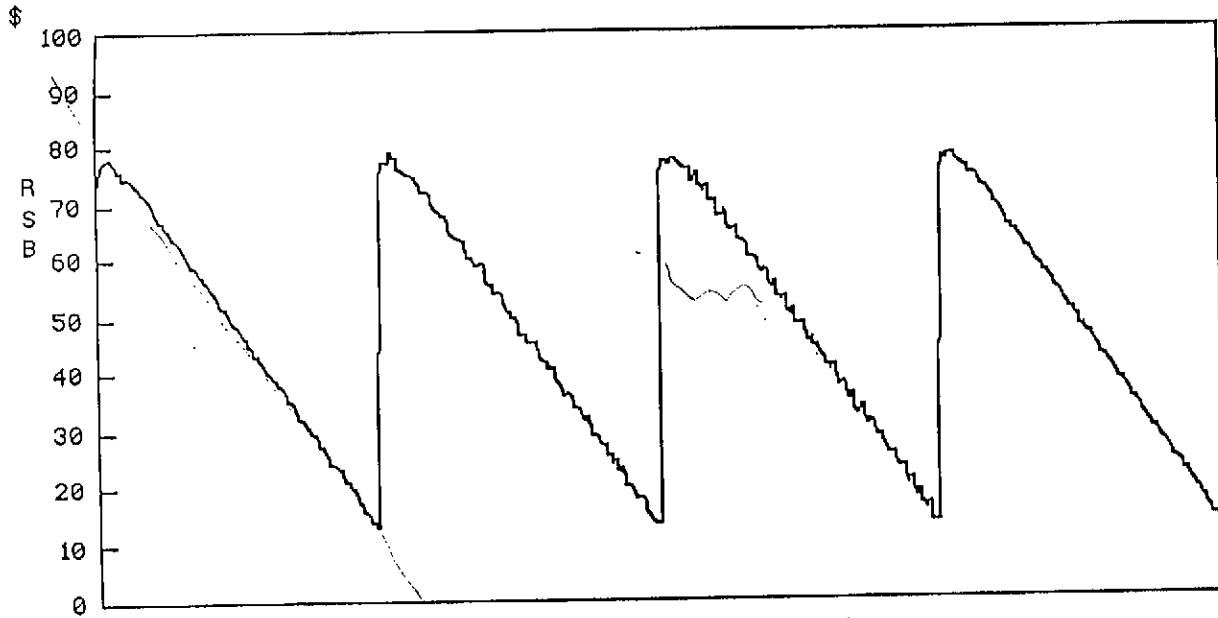


Figure 77: Notes basse fréquence; code différentiel à 8 bits

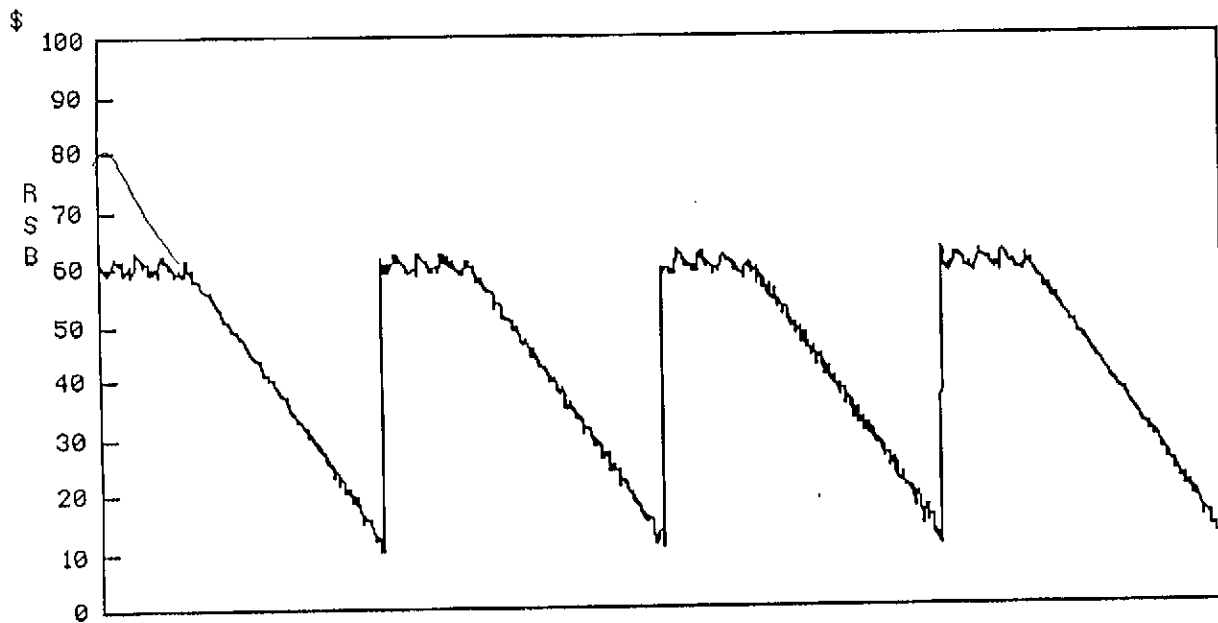


Figure 78: Notes basse fréquence; codeur NICAM 3 à 10 bits.

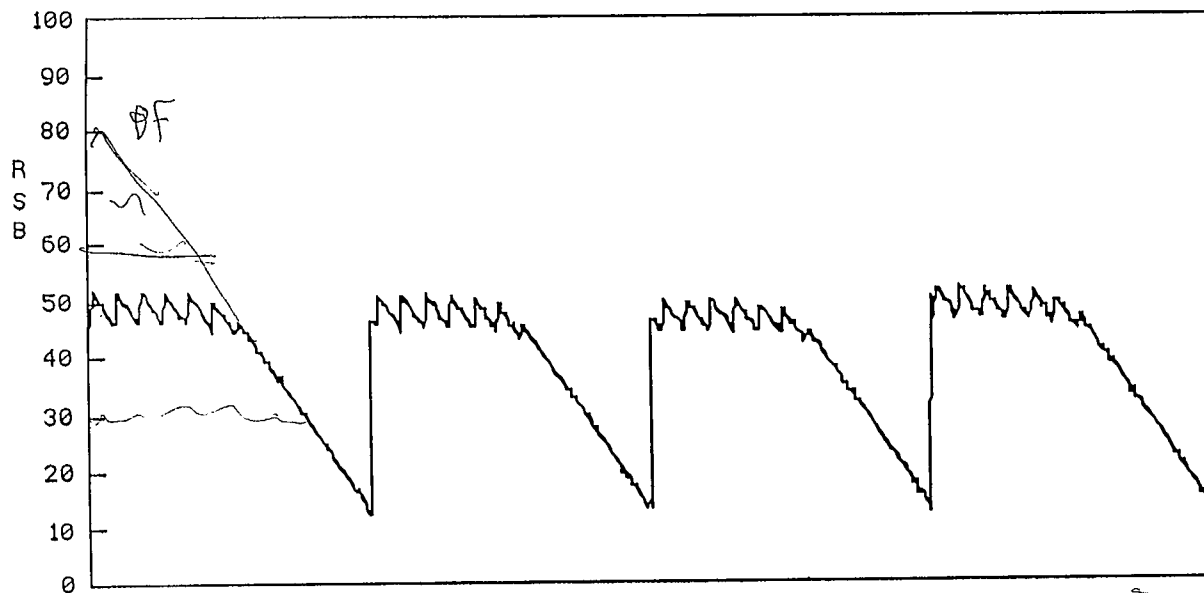


Figure 79: Notes haute fréquence; code différentiel à 8 bits.

YKH₂

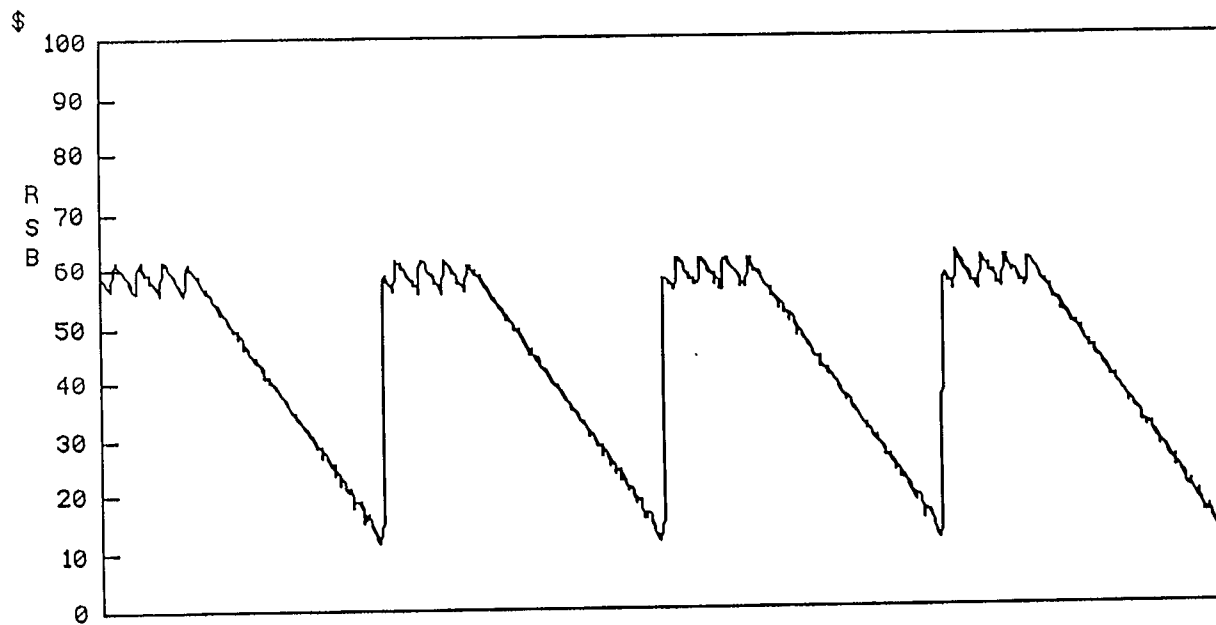
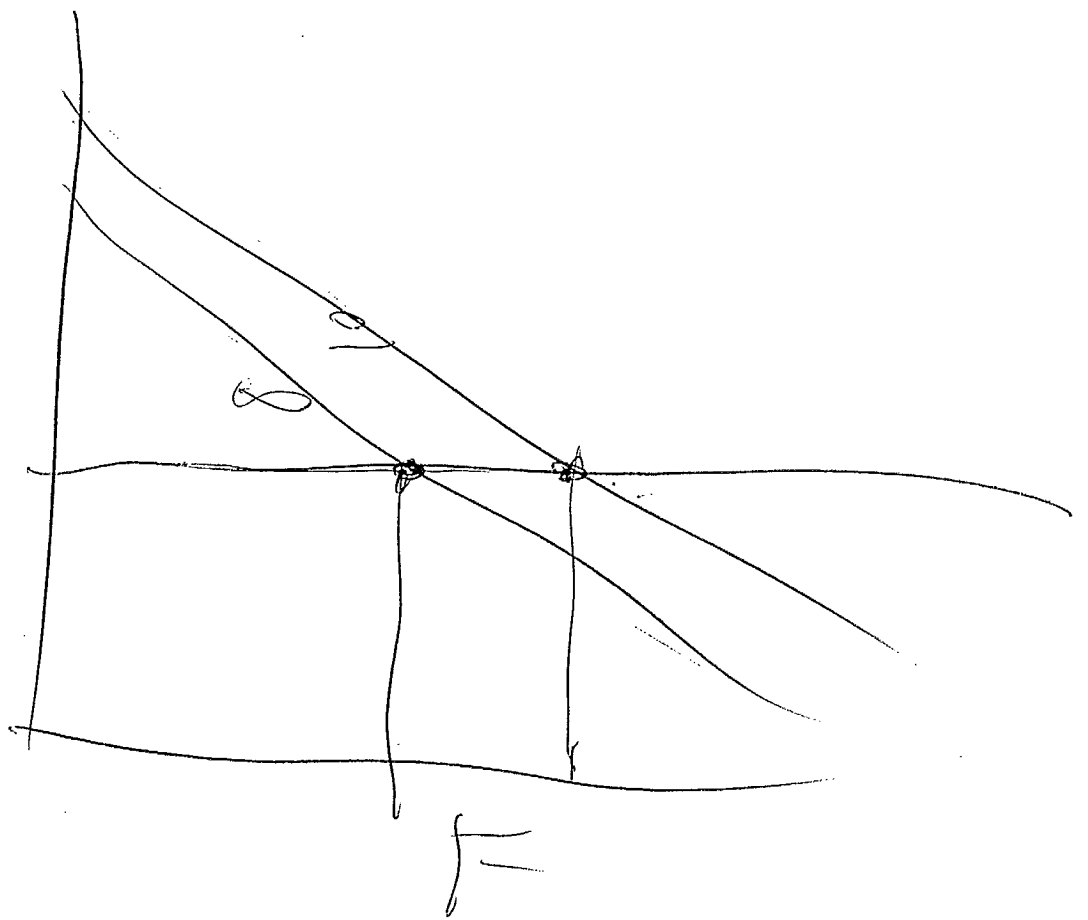


Figure 80: Notes haute fréquence; codeur NICAM 3 à 10 bits.



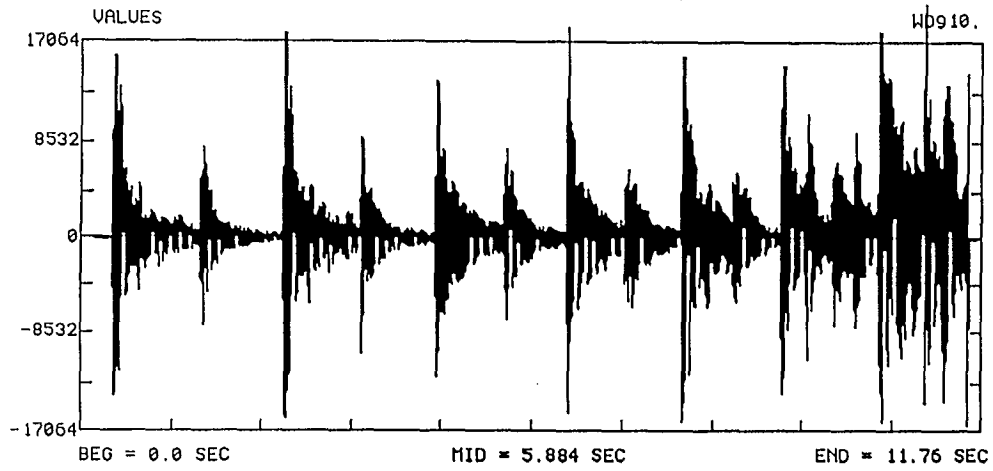


Figure 81: Extrait de harpe (environ 10 secondes).

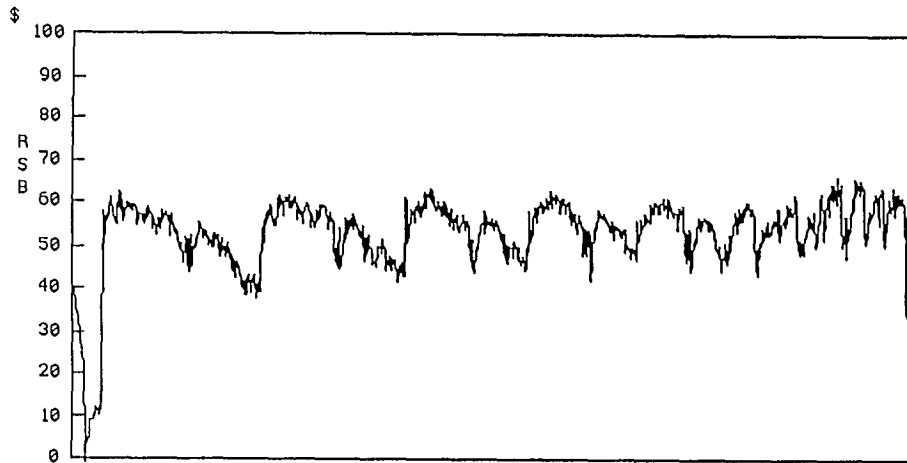


Figure 82: Codage différentiel quasi-instantané à 8 bits.

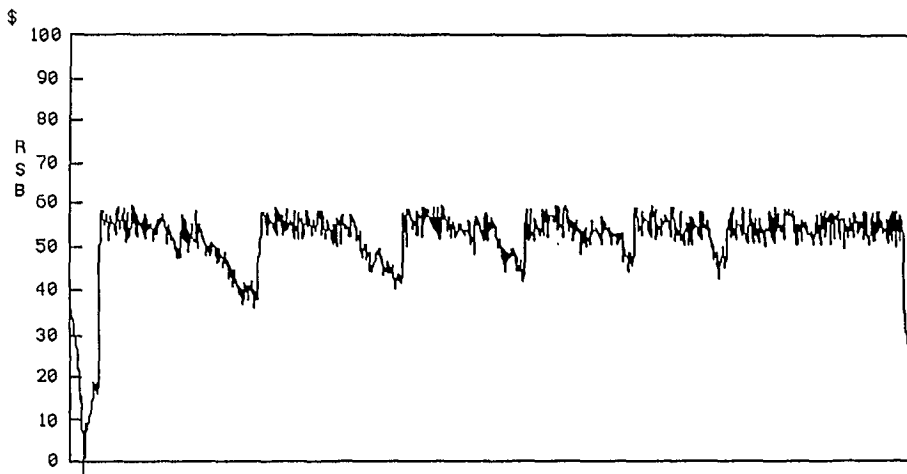


Figure 83: Codage NICAM 3 à 10 bits.

ANNEXE

FILTRE DE SUR-ÉCHANTILLONNAGE

Le filtre "optimal" de sur-échantillonnage a été construit en simulation par calcul des coefficients de sa réponse impulsionnelle. Il s'agit d'un filtre présenté dans un article:

"Interpolation and Decimation of Digital Signals - A Tutorial Review"

R.E. Crochiere, L.R. Rabiner

Proceedings of the IEEE, Vol. 69, #3, March 1981

et publié en détail dans:

"A Computer Program for Digital Interpolator Design",

G. Oetken, T.W. Parks, H.W. Schüssler,

IEEE, Section 8.1



La courbe de réponse en fréquence a été tracée pour un facteur de sur-échantillonnage par 8 (figure A1) et par 32 (figure A2).

On y constate un "affaiblissement" hors-bande de 70 à 80 dB, ce qui est peut être insuffisant pour les applications envisagées.

Il faudra peut être rechercher (et donc recalculer au complet) des filtres de performances plus élevées.

Note: Le filtre de la figure A1 a une réponse impulsionnelle de 257 coefficients et celle de la figure A2 en comporte 513.

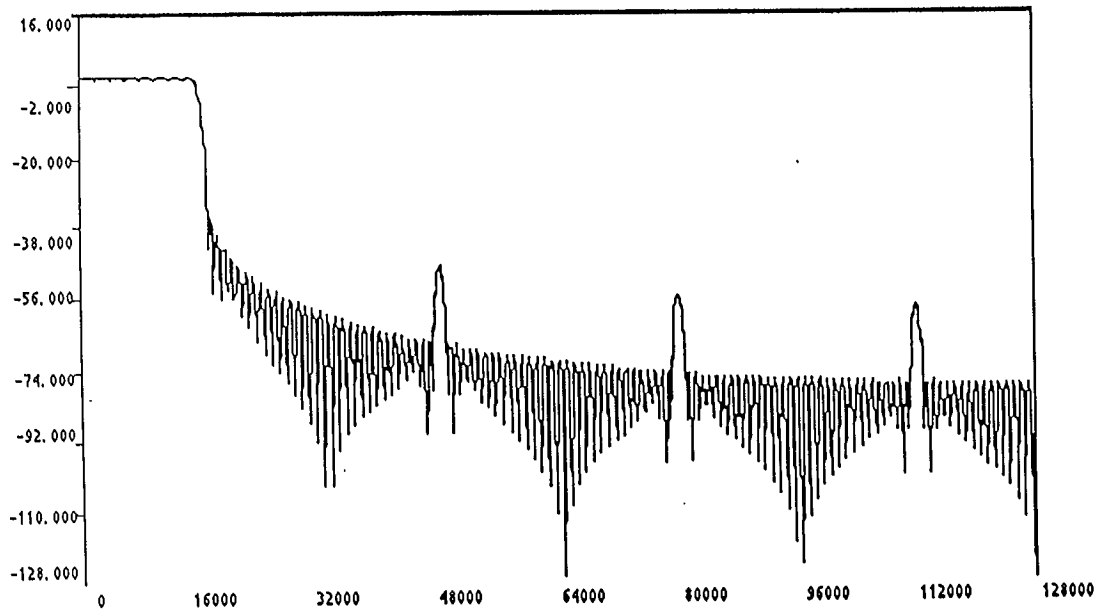


Figure A1: Filtre optimal de sur-échantillonnage: réponse en fréquence pour $R=8$.

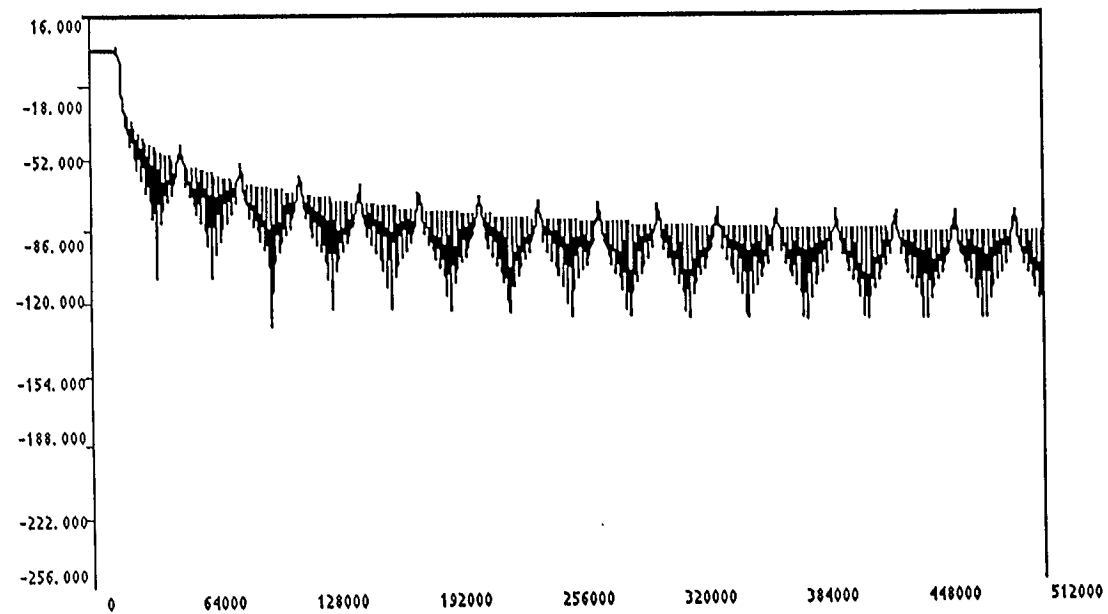


Figure A2: Filtre optimal de sur-échantillonnage: réponse en fréquence pour $R=32$.

