

SIG 1

Table des matières

I	Opérations de base sur les signaux	3
1	Signal numérique de synthèse	3
1.1	Génération du signal	3
1.2	Énergie et puissance	3
1.3	Quantification	4
2	Signal audio	5
2.1	Enregistrement	5
2.2	Restitution	5
2.3	Quantification	6
2.4	Extraction de mots	6
II	Classification des signaux	8
3	Exemple de calcul théorique	8
4	Programmation	8
5	Application à la classification de quelques signaux simples	8
6	Classification de signaux de parole voisés ou non voisés	10
III	Aspects fréquentiels	14
7	Échantillonnage	14
8	La Transformée de Fourier discrète (TFD)	14
8.1	Densité spectrale d'énergie	14
8.2	Zéro-padding	15
8.3	Réduction/élévation de cadence	16
9	Analyse Spectrale	17
9.1	Analyse d'une tranche de signal par TFD	17
9.2	Effets de quelques fenêtres	18
IV	Application : détection de pitch	21
10	Détection de signal voisé	21
11	Mesure du pitch	21
12	Détection de la note	21
13	Détection de la partie de musique	22
14	Test de l'algorithme	22

Table des figures

1	Signal sinusoïdal généré.	3
2	Quantification du signal sinusoïdal généré.	5
3	Quantification du signal sinusoïdal généré - zoom.	5
4	Quantification du signal vocal enregistré.	6
5	Quantification du signal vocal enregistré - comparaison en zoom.	7
6	Comparatif de la fonction xcorr et de notre implémentation de l'autocorrélation.	9
7	Autocorrélation de plusieurs signaux.	9
8	Autocorrélation de divers échantillons du signal vocal.	10
9	Autocorrélation de divers échantillons du signal vocal.	11
10	Autocorrélation de divers échantillons du signal vocal.	12
11	Autocorrélation de divers échantillons du signal vocal.	13
12	Signal échantillonné à $2f + 1$	14
13	Signal échantillonné à $40f$	14
14	Signal échantillonné à $\frac{1}{2}f$	14
15	Signal voisé et DSE	15
16	Signal non-voisé et DSE	15
17	Comparaison de la DSE avec ou sans 0-padding	16
18	Signal échantillonné à 16kHz	16
19	Elevation de cadence	17
20	Réduction de cadence	17
21	TFD d'une fenêtre de Hanning	18
22	Analyse d'une somme de cosinus	18
23	Signal analysé.	19
24	Effet du fenêtrage rectangulaire.	20
25	Effet du fenêtrage de Hanning	20
26	Effet du fenêtrage de Hamming	20
27	Effet du fenêtrage de Blackman	20
28	Enveloppe de l'autocorrélation	21
29	Partition exportée.	22
30	Partition reconstituée.	23

Première partie

Opérations de base sur les signaux

Dans cette première partie, notre objectif est de définir quelques fonctions de calcul de base sur les signaux, afin de découvrir les outils que propose Matlab.

1 Signal numérique de synthèse

1.1 Génération du signal

Tout d'abord, nous avons cherché à générer un signal sinusoïdal et à l'échantillonner avant de tracer son graphe. La figure 1 montre le résultat obtenu pour une fréquence d'échantillonnage de 50 Hz, une fréquence de 4 Hz et 25 points de tracé.

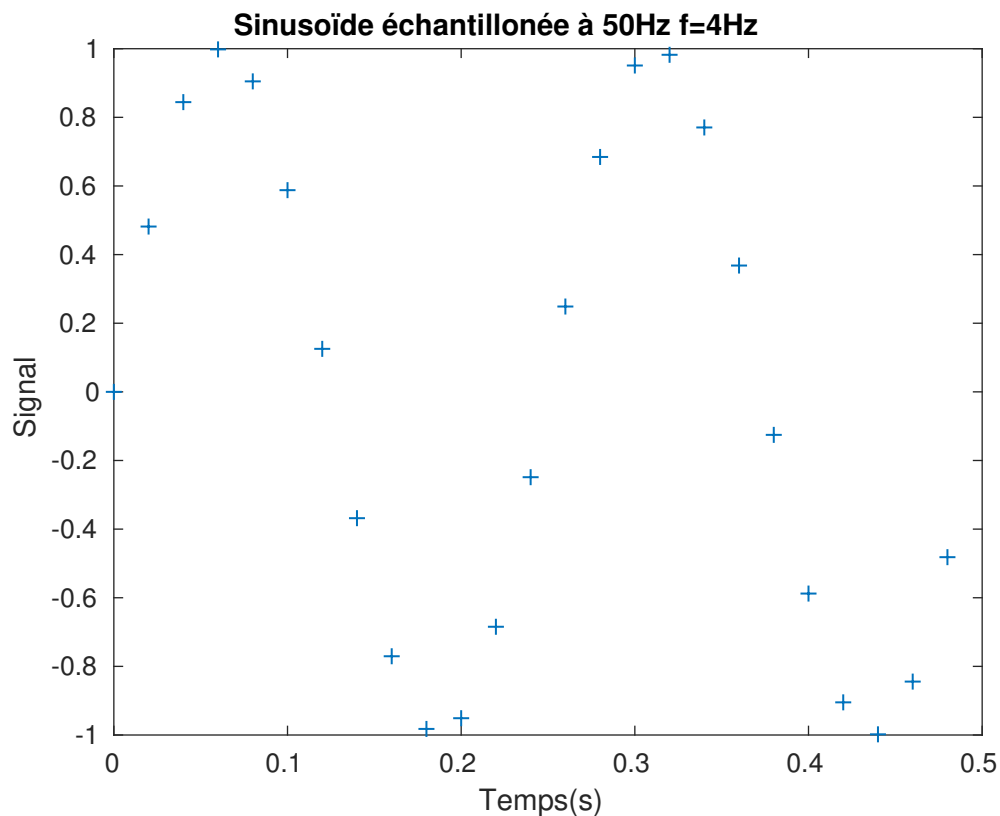


FIGURE 1 – Signal sinusoïdal généré.

1.2 Énergie et puissance

Dans le but de calculer l'énergie d'un signal sans utiliser `for`, nous avons effectué un produit terme à terme et utilisé l'outil `sum`. Une manière encore plus efficace de procéder aurait été de réaliser le produit entre le vecteur donné en entrée et sa transposée. Nous avons ensuite calculé la puissance moyenne de manière théorique, en utilisant l'identité :

$$\sin^2(x) = \frac{1 - \cos(2x)}{2} \quad (1)$$

On obtient

$$\frac{1}{2\pi} \int_0^{2\pi} \sin^2(x) dx = \frac{1}{2} \quad (2)$$

Par ailleurs, nous avons estimé la puissance de notre signal en utilisant la fonction puissance qui effectue la somme du produit terme à terme du signal et divise par la longueur de ce signal.

```
>> s = sig1_sinus(4, 50, 2*50/4);
>> puissance(s)

ans =
```

```
0.5000
```

On obtient un résultat conforme au calcul réalisé.

1.3 Quantification

Pour quantifier un signal sur N bits, on commence par le centrer en 0 et le borner entre -0.5 et 0.5 à l'aide d'une première homothétie. Soit s_1 le signal ainsi obtenu. On réalise ensuite la quantification. Si $N > 0$ est le nombre de bit de quantification, on pose $q = \frac{1}{2^N}$. Dans un premier temps, on pose s_2 tel que :

$$s_2[k] = (\lfloor \frac{s_1[k]}{q} \rfloor + \frac{1}{2}) \times q \quad (3)$$

Cependant cette formule est problématique pour la valeur maximale atteinte par la fonction. En effet, on a alors :

$$s_2[k] = \left(\lfloor \frac{1}{2q} \rfloor + \frac{1}{2} \right) \times q \quad (4)$$

$$= \left(2^{N-1} + \frac{1}{2} \right) \times \frac{1}{2^N} \quad (5)$$

$$> \frac{1}{2} \quad (6)$$

Ce qui est incohérent avec le signal original. Une solution est de poser :

$$s_2[k] = \min \left(\left(\lfloor \frac{s_1[k]}{q} \rfloor + \frac{1}{2} \right) \times q, \frac{1}{2} - \frac{q}{2} \right) \quad (7)$$

Il ne reste plus qu'à remettre la fonction à l'échelle. On obtient le code suivant :

```
function sig_quant = quantifie(X, n_bit)
    m = min(X);
    M = max(X);
    c = (m+M)/2;
    d = M-m;
    q = 1 / (2 ^ n_bit);
    s1 = (X - c)/d;
    sig_quant = min((floor(s1/q) + 1/2) * q, 1/2-q/2)*d + c;
end
```

En appliquant la fonction `quantifie` pour $N = 3$ et $N = 8$ à un signal généré grâce à la fonction écrite dans la partie 1, on obtient la figure 2. On peut noter qu'il est difficile de différencier le signal d'entrée du signal quantifié sur 8 bits. La figure 3 offre un zoom qui permet de différencier le signal original de celui quantifié sur 8 bits.

Pour déterminer le bruit de quantification, on calcule la différence entre le signal d'origine et le signal quantifié, et on en détermine la puissance moyenne. On calcule également la valeur théorique du bruit de quantification. On obtient

Nombre de bits	Puissance moy. théorique du bruit	Puissance moy. mesurée du bruit	RSB en dB
3	0.0013	0.0063004	plop
8	1.272×10^{-6}	4.6464×10^{-6}	plop

On remarque tout d'abord que les valeurs mesurées de la puissance moyenne du bruit de quantification sont proches de celles prédites par la théorie. On remarque que l'échantillonnage à 8 bits est bien plus efficace, bien que la valeur théorique soit plus faible que celle calculée.

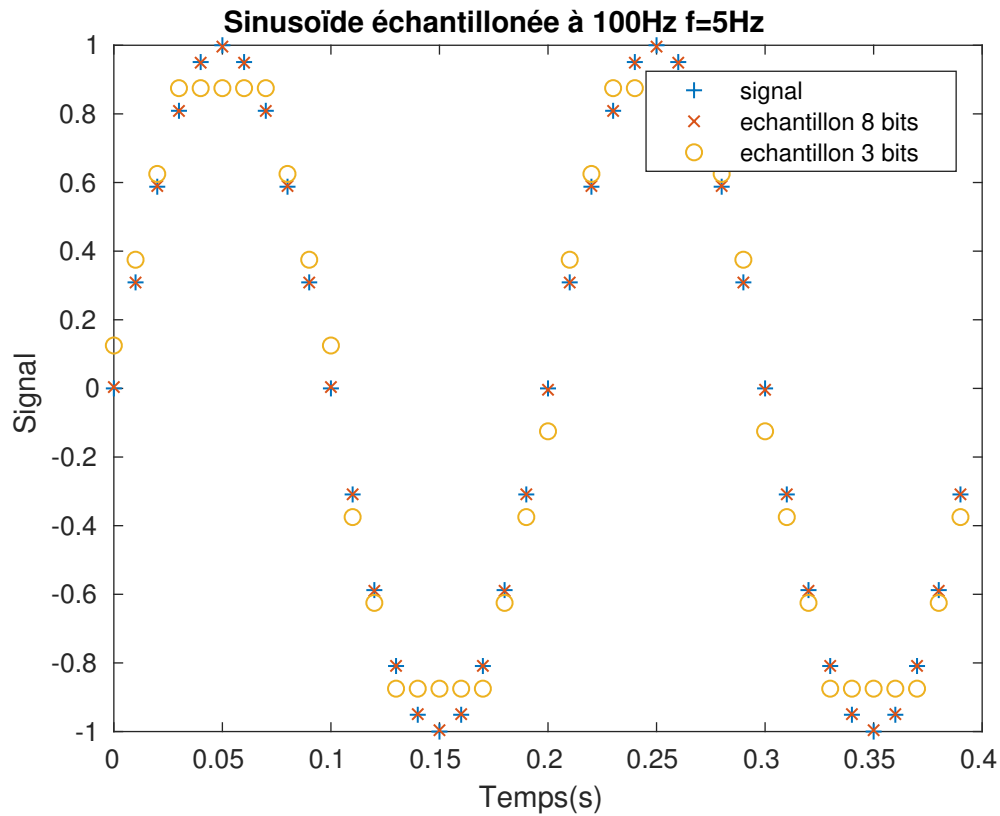


FIGURE 2 – Quantification du signal sinusoïdal généré.

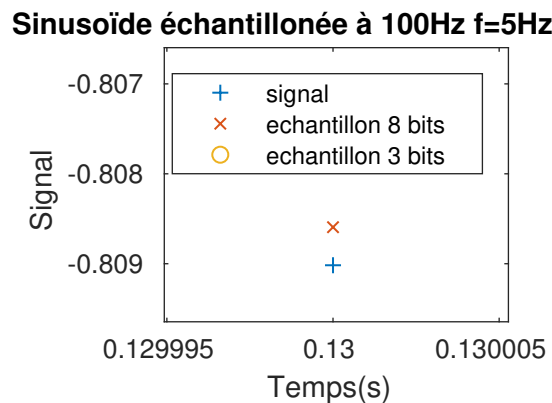


FIGURE 3 – Quantification du signal sinusoïdal généré - zoom.

2 Signal audio

2.1 Enregistrement

On enregistre un signal audio en utilisant le logiciel Audacity. Il est important de veiller à ne pas saturer le micro lors de l'enregistrement. La fréquence d'échantillonnage utilisée est de 11 kHz. On le retrouvera dans la figure 4. Les deux mots prononcés sont "Hervé, Biausser".

2.2 Restitution

En écoutant le signal enregistré à différentes fréquences de restitutions, on remarque qu'augmenter cette fréquence diminue la durée de la restitution et décale les fréquences de l'enregistrement vers des fréquences plus hautes. Inversement, diminuer la fréquence de restitution augmente la durée de l'enregistrement tout en déplaçant les fréquences des sons de l'enregistrement vers les graves.

2.3 Quantification

A l'aide de la fonction `quantifie` réalisée dans la partie précédente, on réalise la quantification du signal enregistré. On observe que plus le nombre de bits de quantification est faible, plus le bruit (à l'écoute) est important.

Les figures 4 et 5 donnent une représentation graphiques de différentes quantification du signal vocal.

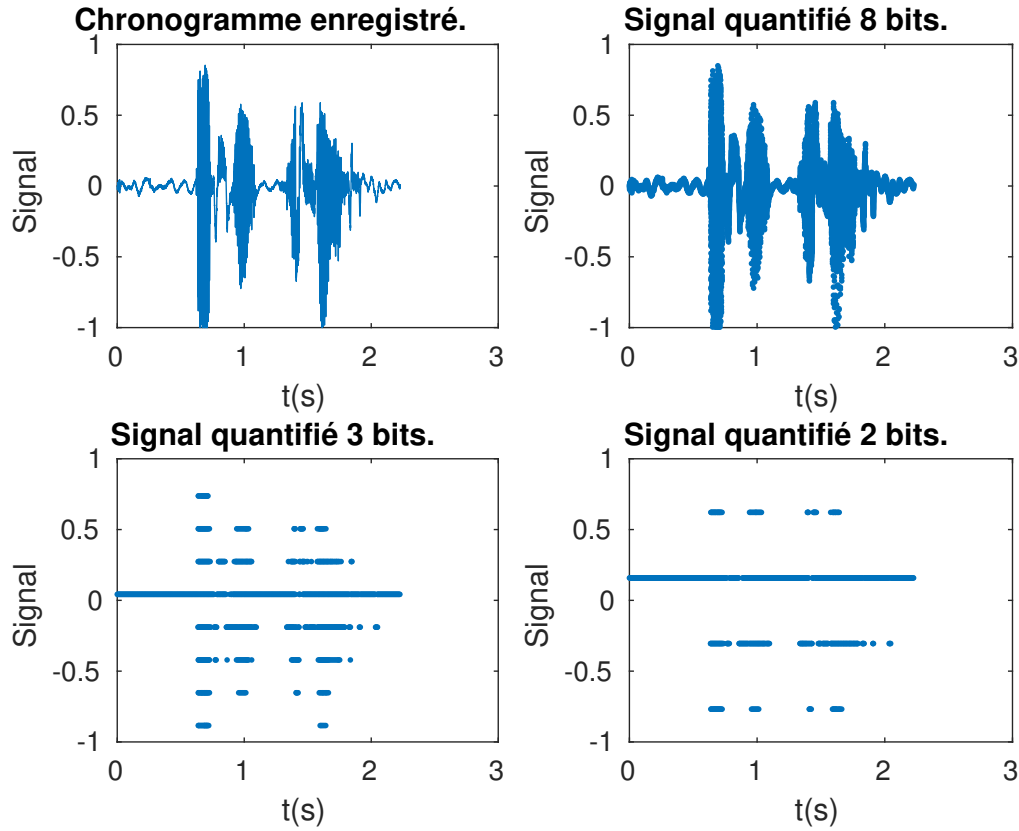


FIGURE 4 – Quantification du signal vocal enregistré.

2.4 Extraction de mots

En observant le chronogramme du signal, on distingue les deux mots, ce qui permet de repérer l'instant t où couper le signal pour séparer ces deux mots. L'indice correspondant du vecteur dans lequel est stocké le signal est $k = t * f_e$ où f_e est la fréquence d'échantillonnage.

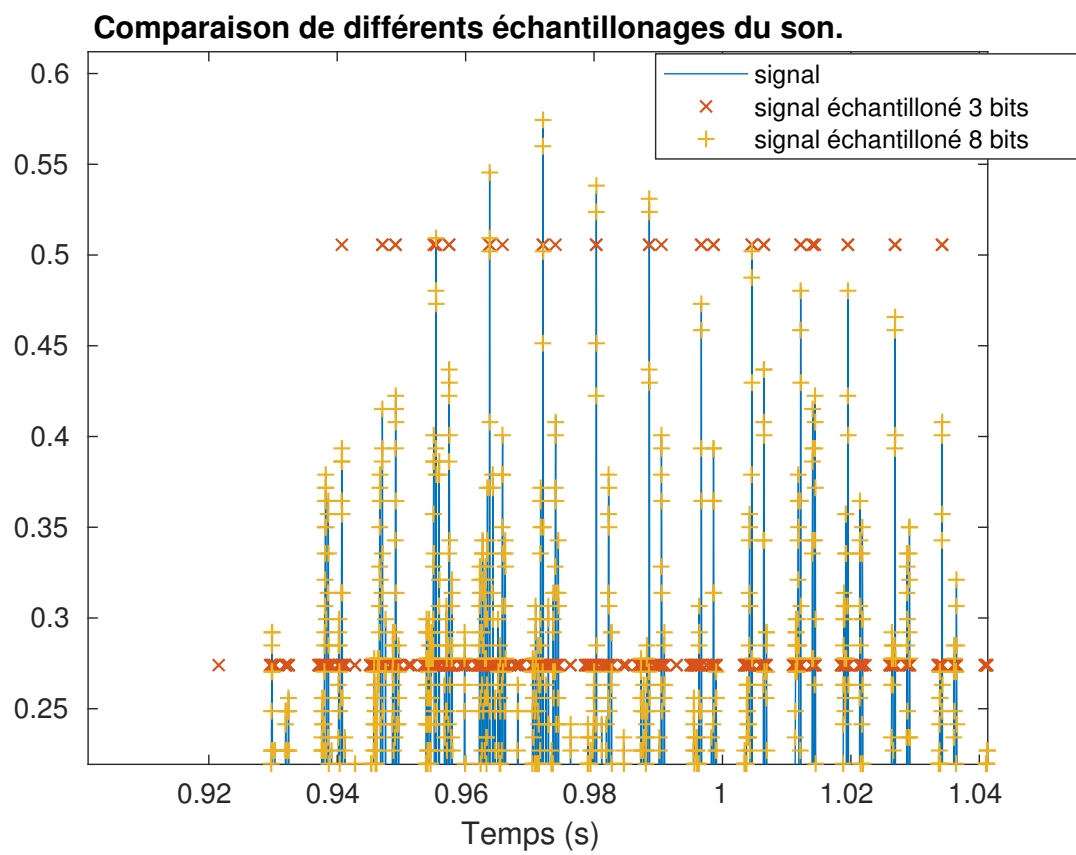


FIGURE 5 – Quantification du signal vocal enregistré - comparaison en zoom.

Deuxième partie

Classification des signaux

Le but de cette partie est de classer les signaux, notamment en utilisant l'autocorrélation.

3 Exemple de calcul théorique

Pour le signal $y(t) = A \sin(2\pi ft)$, calculons l'autocorrélation :

$$\gamma_y(\tau) = \int_0^{\frac{1}{f}} A^2 \sin(2\pi ft) \sin(2\pi f(t + \tau)) dt \quad (8)$$

$$= A^2 \int_0^{\frac{1}{f}} \frac{1}{2} \times (\cos(2\pi ft - 2\pi f(t + \tau)) + \cos(2\pi ft + 2\pi f(t + \tau))) dt \quad (9)$$

$$= \frac{A^2}{2} \cos 2\pi f\tau - \underbrace{\frac{A^2}{2} \int_0^{\frac{1}{f}} \cos(2\pi f(2t + \tau)) dt}_{=0} \quad (10)$$

On obtient la fonction d'autocorrélation suivante :

$$\gamma_y(t) = \frac{A^2}{2} \cos(2\pi ft) \quad (11)$$

4 Programmation

La formule de l'autocorrélation pour un signal discret x est :

$$\forall n \in \mathbb{N}, \gamma_x[n] = \sum_{k \in \mathbb{Z}} x[k] \times x[k + n] \quad (12)$$

Puisqu'on travaille sur des signaux à support fini, stockés sur des indices compris entre 1 et N (avec $N = \text{length}(x)$), on implémente l'autocorrélation avec la formule suivante :

$$\forall n \in [-N, N], \gamma_x[n + N] = \sum_{k, (k+n>0) \wedge (k+n \leq N)} x[k] x[k + n] \quad (13)$$

Matlab propose une fonction `xcorr` qui permet de calculer l'autocorrélation. Afin de comparer notre implémentation et cette fonction, on trace les résultats de l'appel aux deux fonctions sur un même signal. On prend soin d'afficher le résultat sur une échelle cohérente pour τ .

La figure 6 montre que le résultat de `xcorr` et de notre implémentation de l'autocorrélation sont identiques.

Remarque Nous n'avons d'abord tracé que la partie positive, avant de nous rendre compte de la nécessité de tracer également la partie négative. On obtient un résultat différent du résultat théorique car le support de notre signal est fini, ce qui donne un cosinus amorti de manière symétrique et non un simple cosinus.

5 Application à la classification de quelques signaux simples

Lorsque le signal se "répète" au cours du temps son autocorrélation est semblable sur de courtes périodes (reproduction du même motif au cours des périodes), on dit alors que le signal est stationnaire. Par contre si son autocorrélation est presque nulle, cela veut dire qu'il n'y a pas reproduction des motifs au cours des périodes et donc que le signal est non stationnaire.

La figure 7 montre que les signaux sinusoïdaux et triangulaires sont stationnaires, contrairement au signal du bruit.

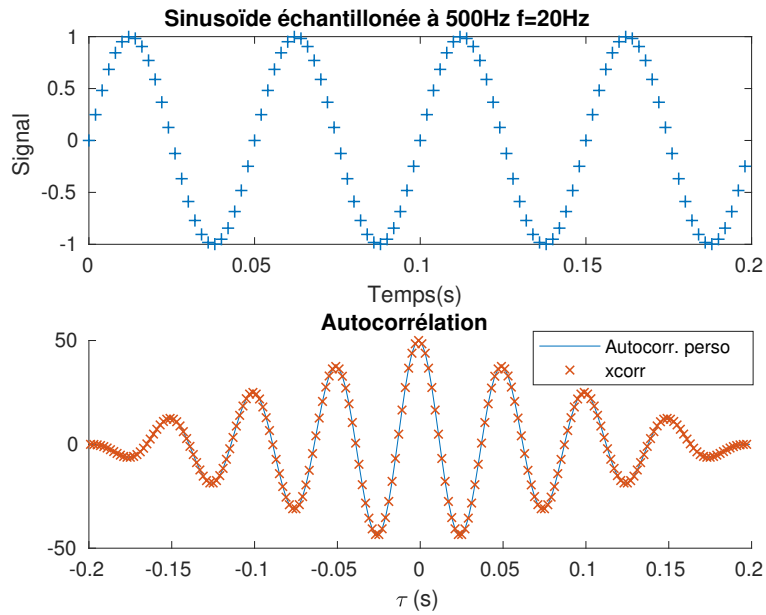


FIGURE 6 – Comparatif de de la fonction xcorr et de notre implémentation de l'autocorrélation.

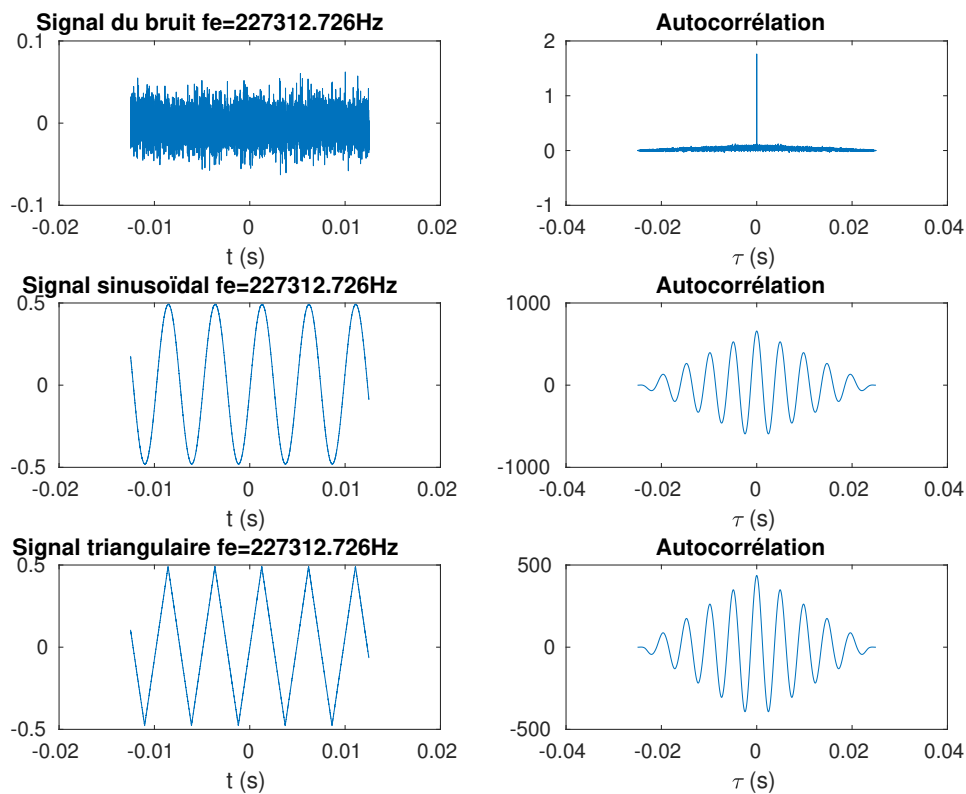


FIGURE 7 – Autocorrélation de plusieurs signaux.

La figure 8 montre que le signal vocal est quasi-stationnaire. En effet, sur de courtes périodes, son autocorrélation varie peu. ces résultats vont nous permettre dans la section suivante de séparer les différentes composantes d'un signal vocal. En effet, celles constituées essentiellement de bruit présenteront une autocorrélation très faible devant celles des parties voisées.

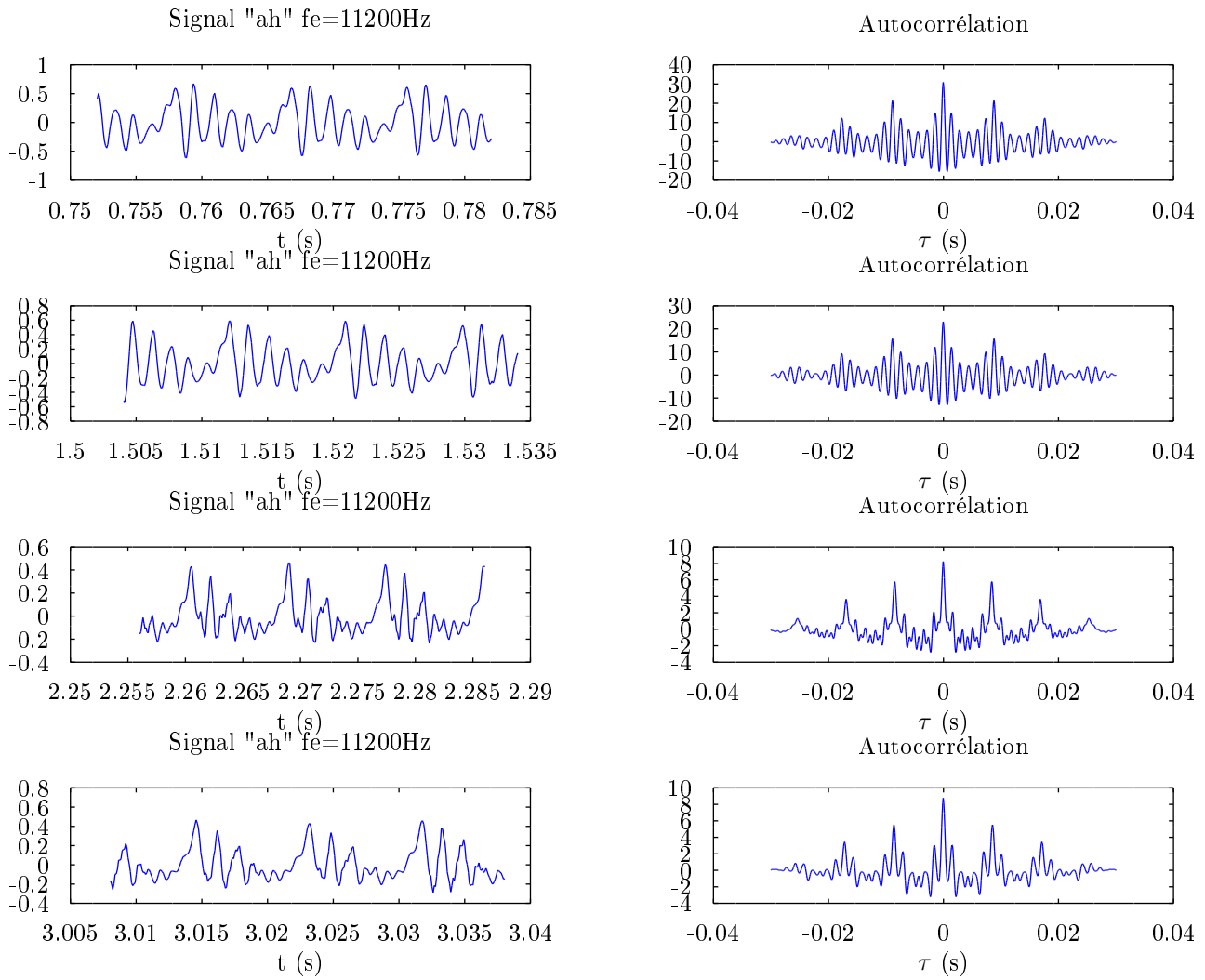


FIGURE 8 – Autocorrélation de divers échantillons du signal vocal.

6 Classification de signaux de parole voisés ou non voisés

D'après les résultats de la section précédente, on peut différencier un signal de parole voisé d'un non voisé en visualisant l'autocorrélation. En effet un signal non voisé est constitué de bruit et devrait présenter une autocorrélation faible, et non périodique. Se pose alors la question de la longueur des échantillons sur lesquels on calcule l'autocorrélation. Nous avons essayé les durées 0.01, 0.06, 0.16 et 0.64 secondes.

La figure 9 montre que des échantillons de durée 0.01 secondes sont trop courts pour différencier aisément le signal voisé du non voisé.

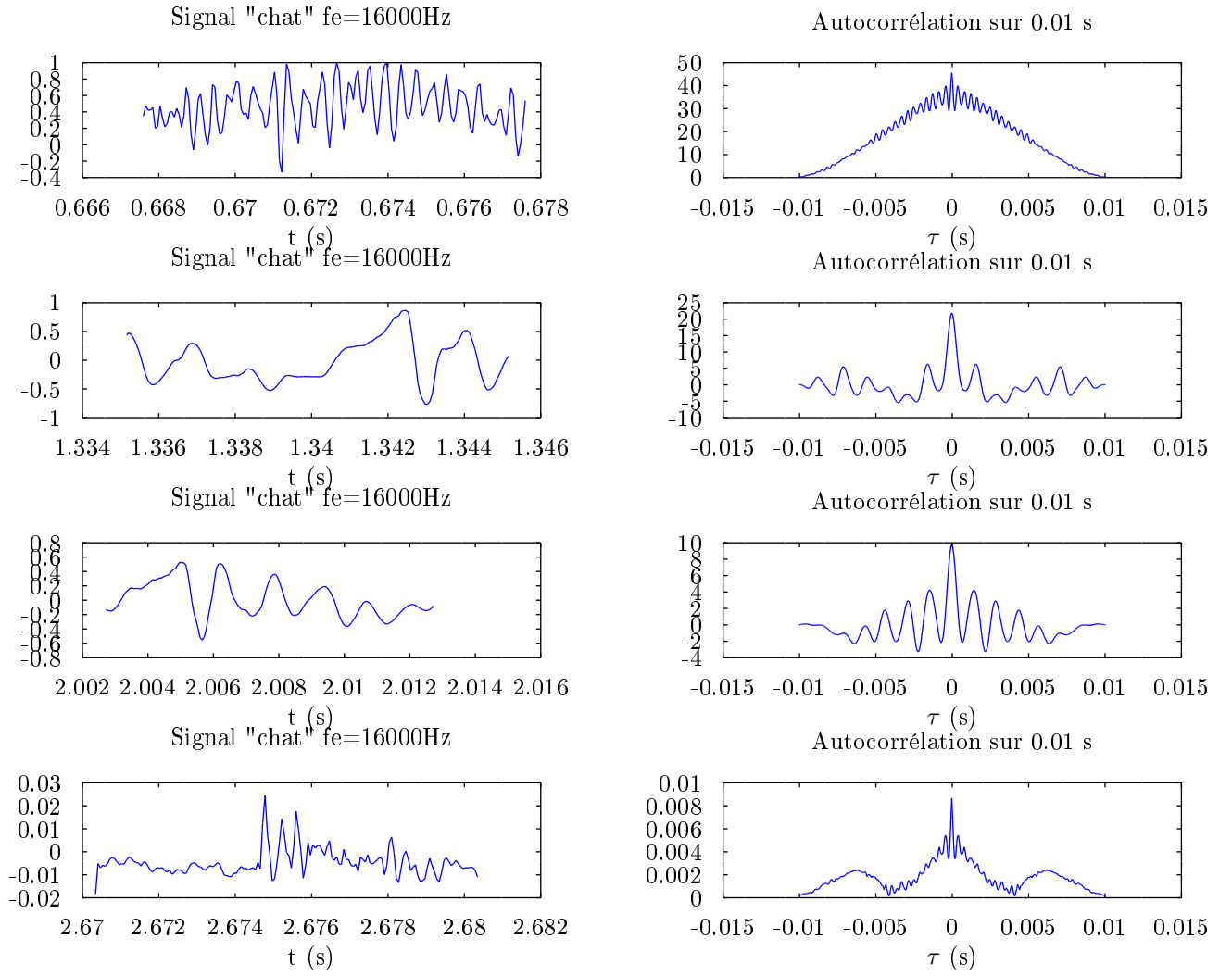


FIGURE 9 – Autocorrélation de divers échantillons du signal vocal.

La figure 10 montre que des échantillons de durée 0.64 secondes sont trop longs. En effet, on se rapproche alors de la période de variation du signal liée au sens de la parole (visible sur l'échantillon pris entre 2 et 2.8 secondes par exemple).

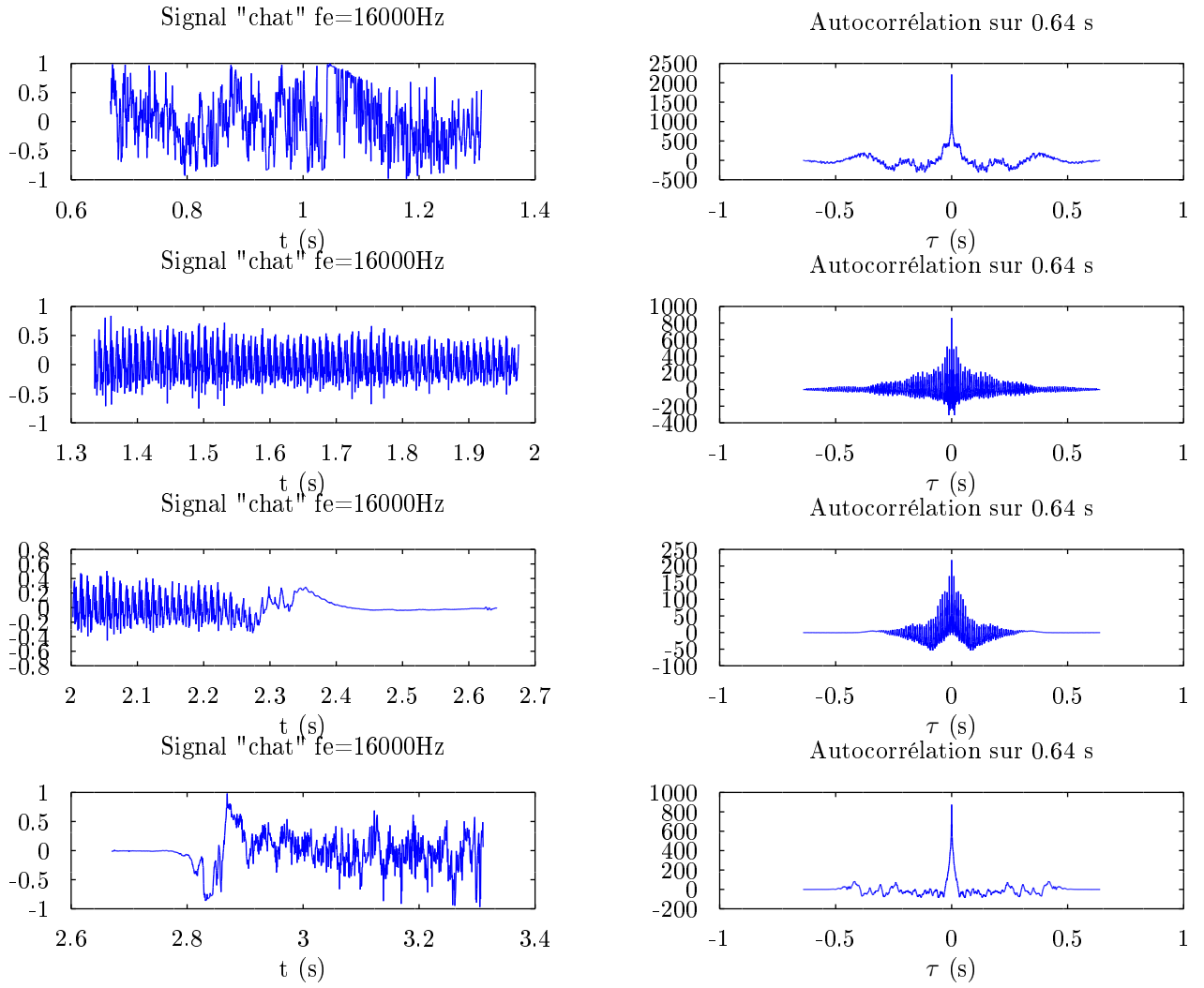


FIGURE 10 – Autocorrélation de divers échantillons du signal vocal.

Finalement, on choisit une durée d'échantillon de 0.16 secondes (figure 11), car bien que celle de 0.06s convienne également pour différencier le signal voisé du non voisé, l'autocorrélation sur une plus grande période fait apparaître plus de maximums locaux, ce qui permet d'estimer plus précisément la fréquence fondamentale du signal voisé. On trouve ainsi une fréquence du fondamental de 100 Hz environ.

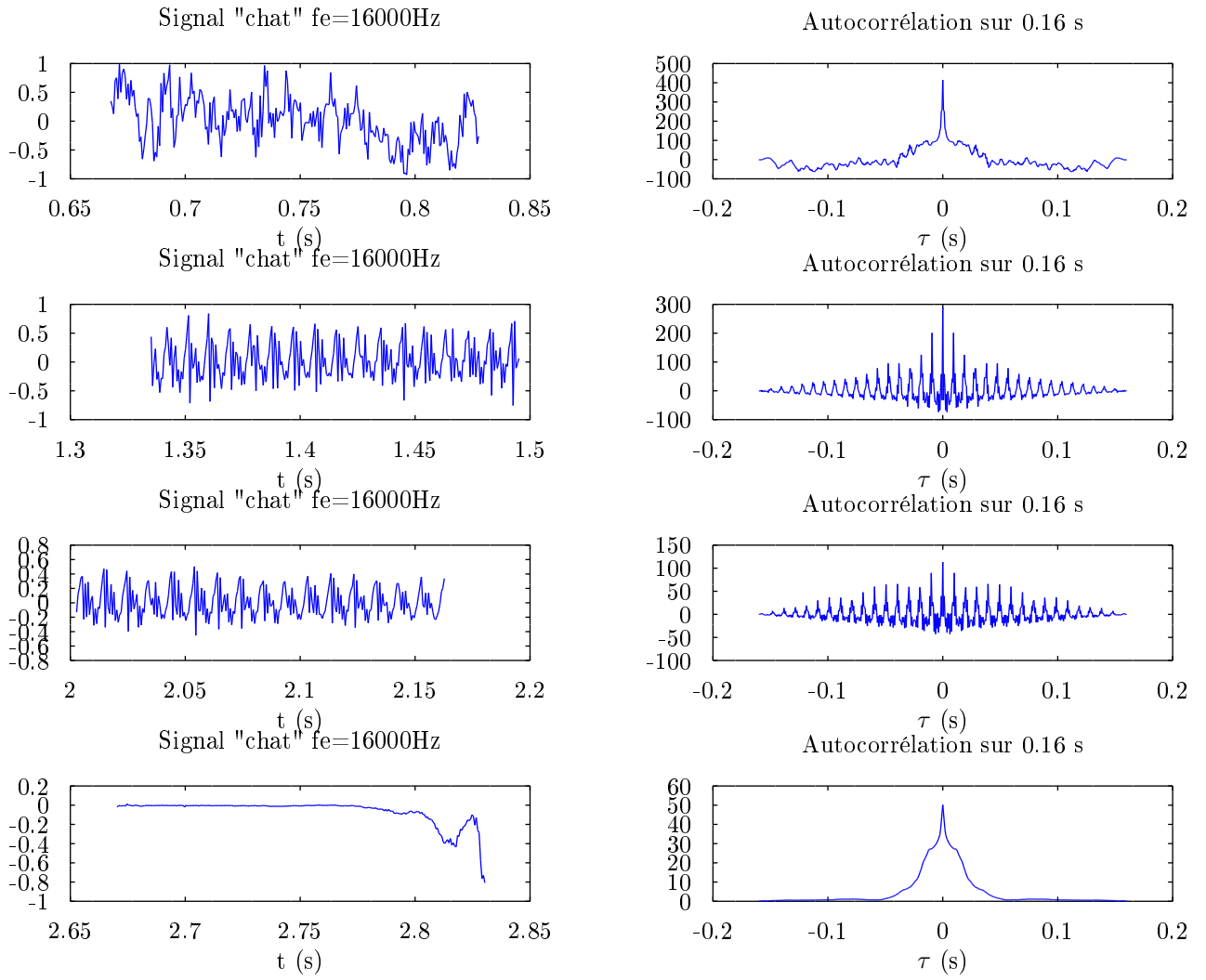


FIGURE 11 – Autocorrélation de divers échantillons du signal vocal.

Troisième partie

Aspects fréquentiels

7 Échantillonnage

On réalise l'échantillonnage d'un même signal de fréquence $f = 500 \text{ Hz}$ échantillonné à plusieurs fréquences différentes, le résultat obtenu est représenté dans les figures 12 (on se place à la limite de ce qu'impose le critère de Shannon), 13 (on échantillonne avec une très grande fréquence d'échantillonnage) et 14 (on échantillonne avec une très faible fréquence d'échantillonnage).

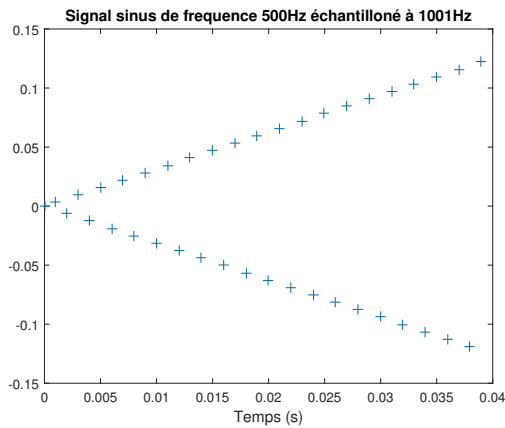


FIGURE 12 – Signal échantillonné à $2f + 1$.

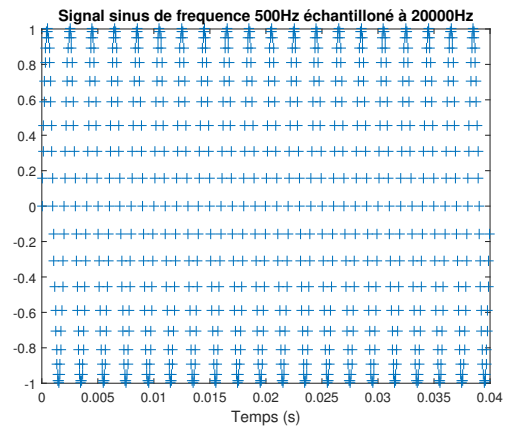


FIGURE 13 – Signal échantillonné à $40f$.

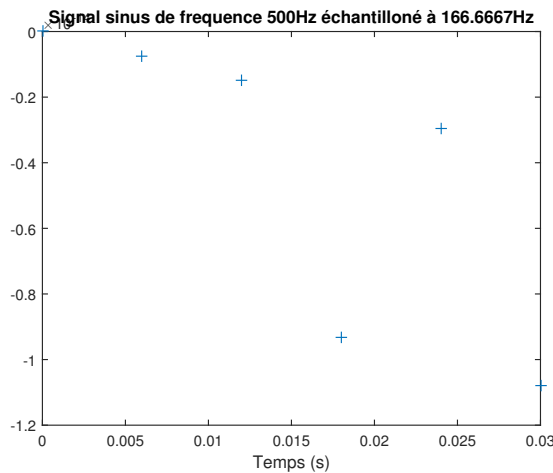


FIGURE 14 – Signal échantillonné à $\frac{1}{2}f$.

Visuellement, on constate que pour $f_e \gg f_{max}$, le signal est bien reconstitué, contrairement au cas $f_e \ll f_{min}$ (critère de Shannon).

8 La Transformée de Fourier discrète (TFD)

8.1 Densité spectrale d'énergie

On trace la densité spectrale d'énergie d'un signal voisé (figure 15) et d'un signal non-voisé (figure 16). On remarque que le signal non voisé ("ch") comporte de l'énergie dans toutes les fréquences de manière identique, tandis que le signal voisé ("a") présente de l'énergie principalement sur certaines fréquences. En effet, la figure 15 présente des pics d'énergie espacés régulièrement, particulièrement visibles entre 0 et 1000 Hz. On remarque que ces pics sont espacés de 100Hz : on a un résultat cohérent avec ce qui est obtenu par la méthode temporelle.

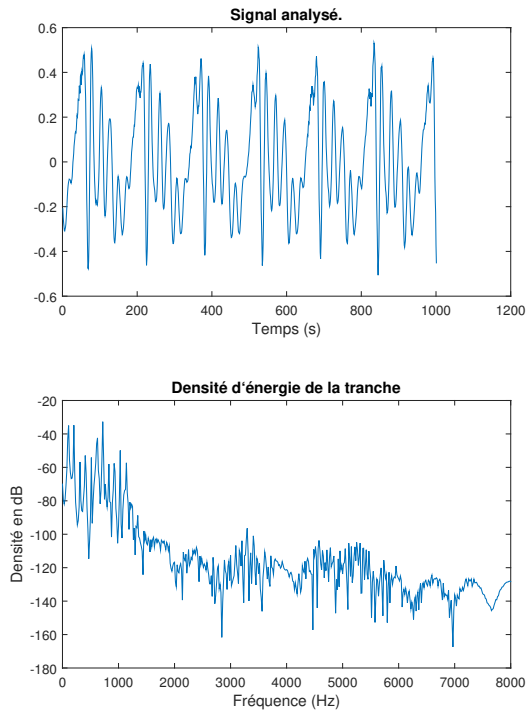


FIGURE 15 – Signal voisé et DSE

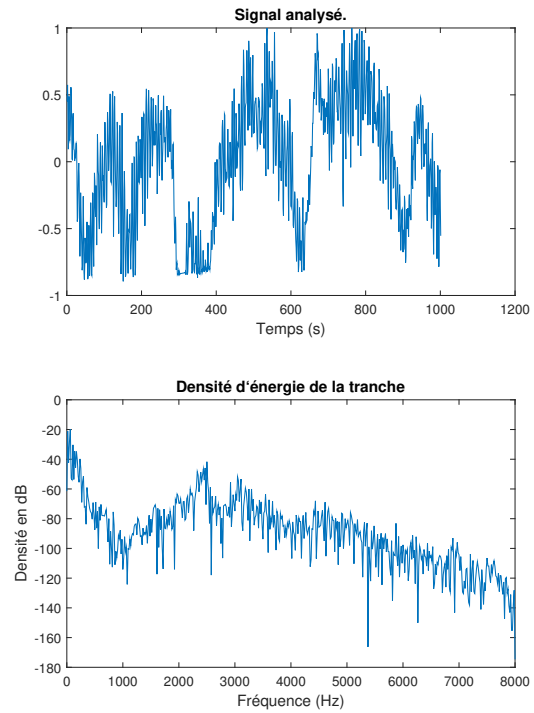


FIGURE 16 – Signal non-voisé et DSE

8.2 Zéro-padding

On réalise ensuite une fonction de 0-padding. On compare la DSE pour un signal voisé, la figure 17 montre une amélioration de la précision spectrale sans modifier la résolution (l'ajout de zéro n'ajoute pas d'informations dans le spectre).

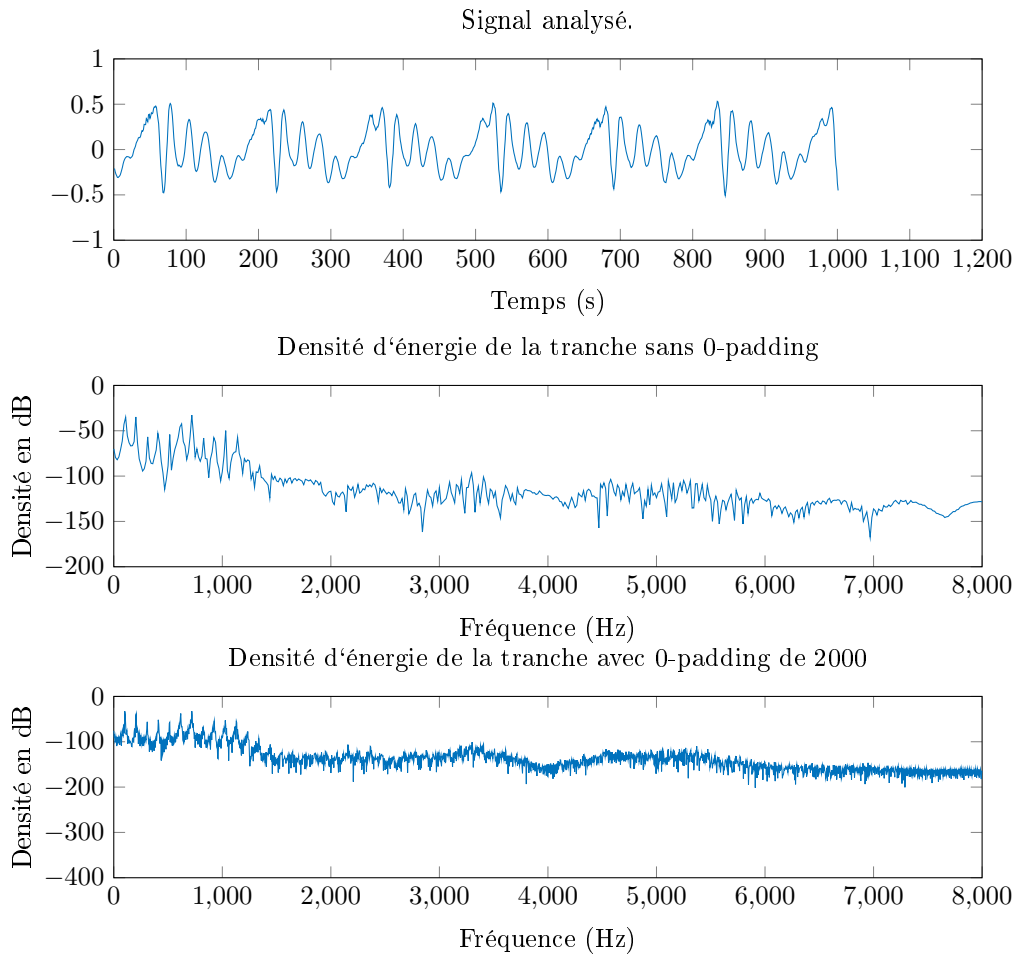


FIGURE 17 – Comparaison de la DSE avec ou sans 0-padding

8.3 Réduction/élévation de cadence

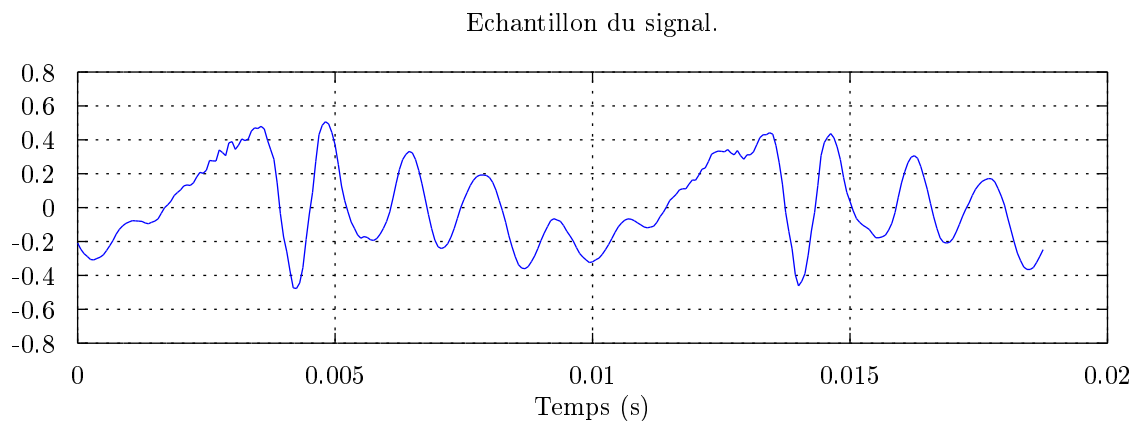
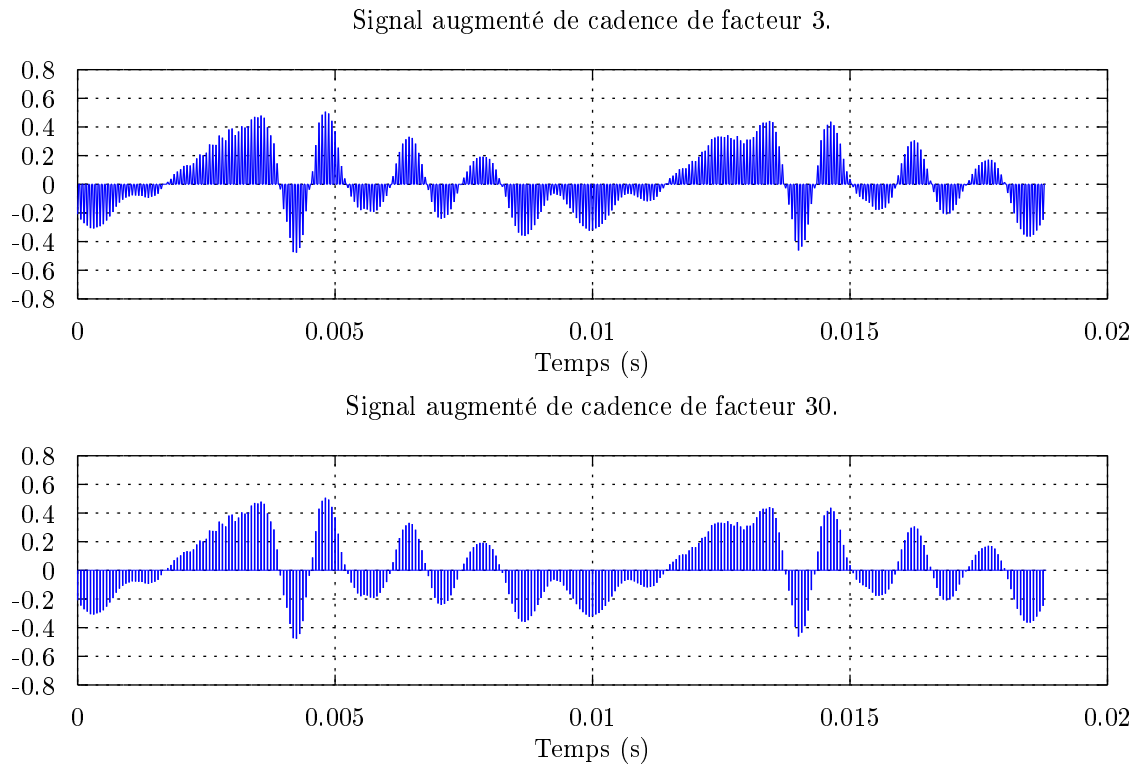
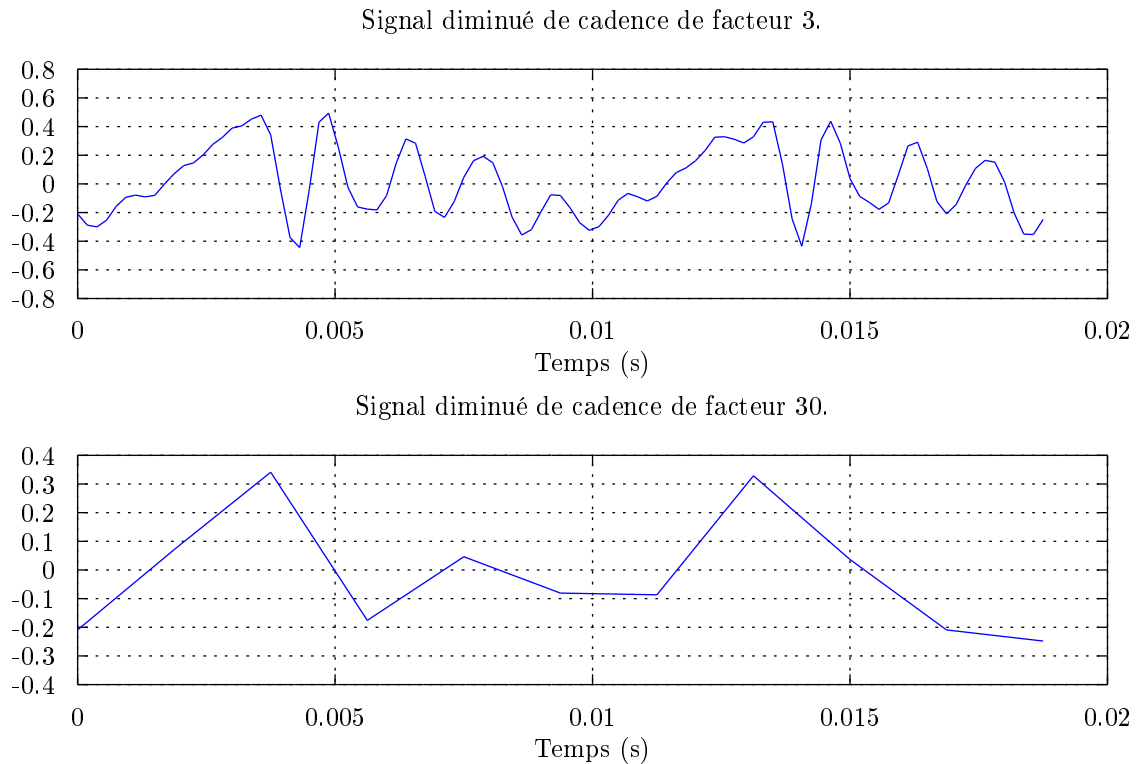


FIGURE 18 – Signal échantillonné à 16kHz

On reprend notre signal ("a") (figure 18) pour évaluer l'influence de l'élévation et de la réduction de cadence. L'élévation de cadence (figure 19) amène à une fréquence d'échantillonnage plus élevée en insérant des zéros entre les intervalles temporels, ce qui ajoute des composantes plus aigües à notre signal. La réduction de cadence (figure 20) quant à elle supprime des échantillons et il est possible de ne plus respecter le critère de Shanon, impliquant du repliement spectral perceptible au travers d'un son en général plus grave.

En écoutant le résultat de la fonction `resample`, on se rend compte que le résultat obtenu est de meilleur qualité. En effet, la fonction `resample` applique un filtre passe bas de manière à supprimer les composantes parasites (le bruit qu'on a en augmentant ou diminuant la cadence), ce qui explique pourquoi on a un résultat moins bruité.

**FIGURE 19** – Elevation de cadence**FIGURE 20** – Réduction de cadence

9 Analyse Spectrale

9.1 Analyse d'une tranche de signal par TFD

Le module de la transformée de Fourier d'un signal réel étant paire, et puisque la figure 21 montre que l'on a environ un triangle dont un sommet est en $(0,13 \approx \frac{4}{2 \times 15}, 0)$ et un autre en $(0,14 \approx \frac{2 \times 15}{4})$. On vérifie donc que

le spectre de la fenêtre de Hanning peut s'approximer par un triangle de surface unité et de largeur $\frac{4}{N}$.

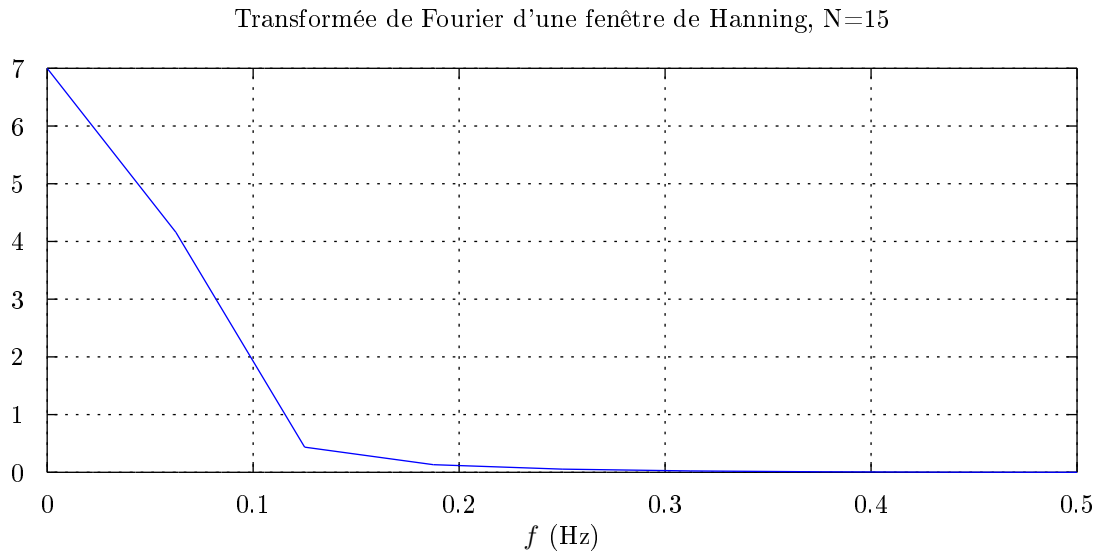


FIGURE 21 – TFD d'une fenêtre de Hanning

On remarque grâce à la figure 22 que, du fait de l'échantillonnage de la transformée de Fourier, le second pic du graphe est tronqué. On retrouve ainsi le résultat du TD 4.

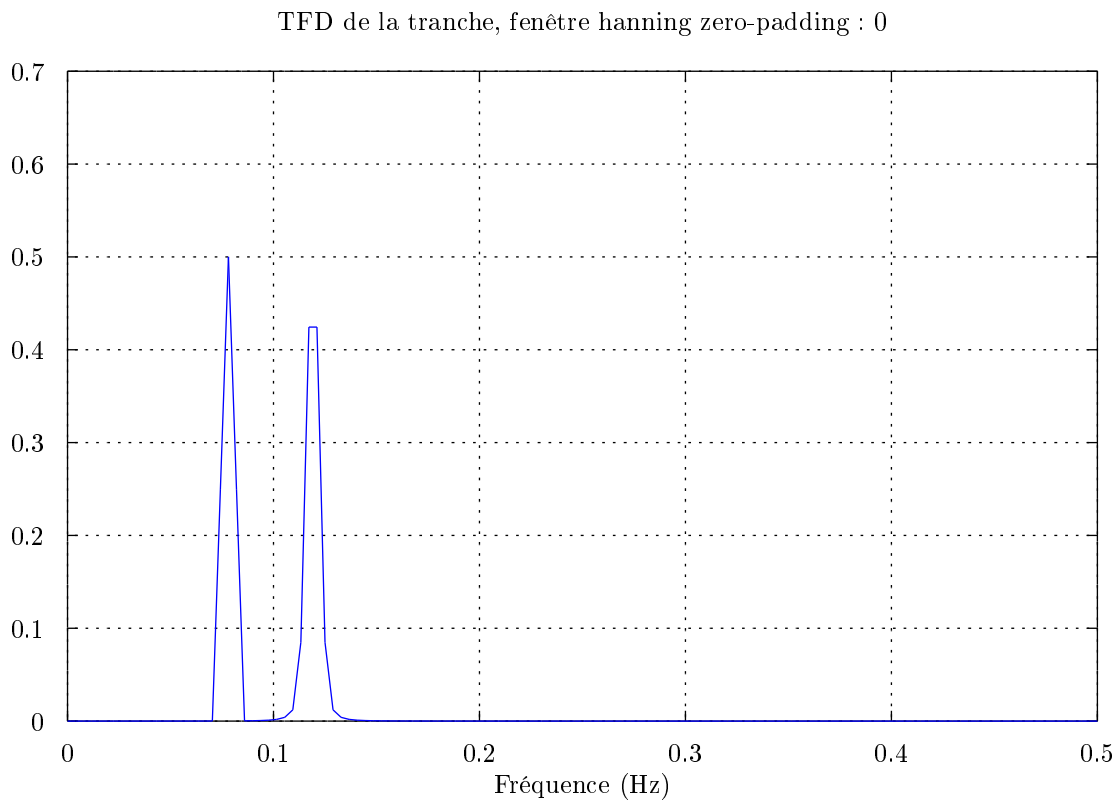


FIGURE 22 – Analyse d'une somme de cosinus

9.2 Effets de quelques fenêtres

Après modification de la fonction d'analyse pour permettre à l'utilisateur de choisir la fenêtre de pondération. On remarque tout d'abord que le 0-padding permet d'augmenter la résolution de la transformée de Fourier discrète. C'est pourquoi on comparera les effets des différentes fenêtres avec un 0-padding de 2048 points.

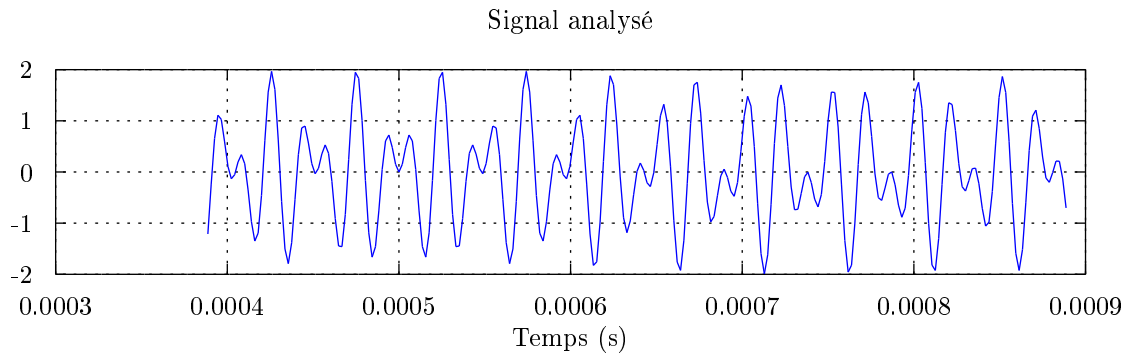
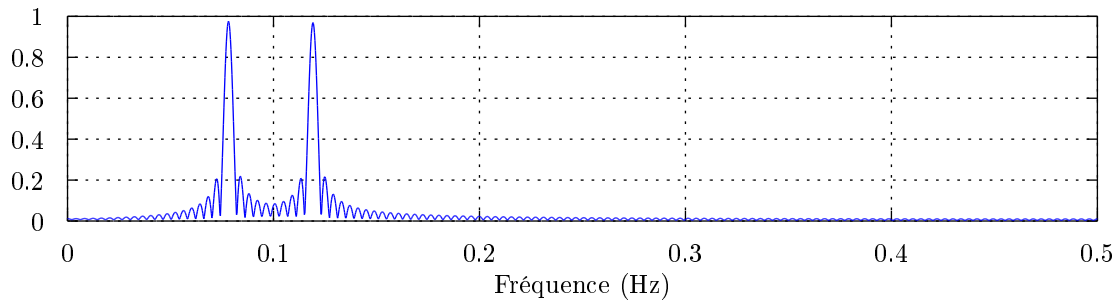


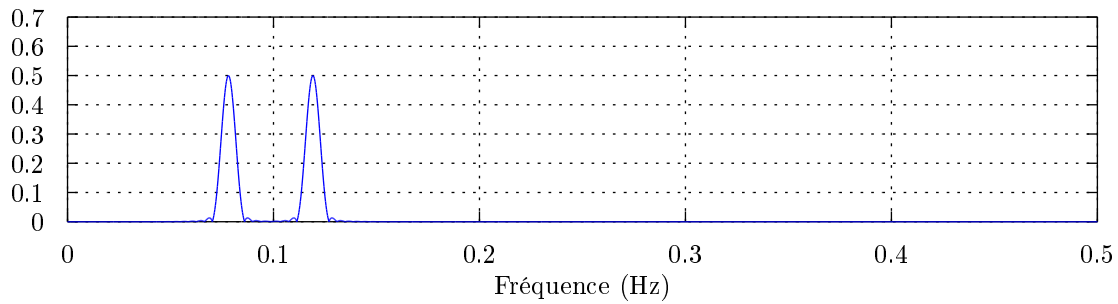
FIGURE 23 – Signal analysé.

On remarque que le fenêtrage rectangulaire (figure 24), s'il produit des pics de faible largeur devant ceux des autres fenêtres produit des lobes secondaires de hauteur importante. Les fenêtres de Hanning (figure 25), puis de Hamming (figure 26) et de Blackman (figure 27) produisent des lobes secondaires de hauteur de plus en plus faibles, mais au détriment de la largeur du lobe principal.

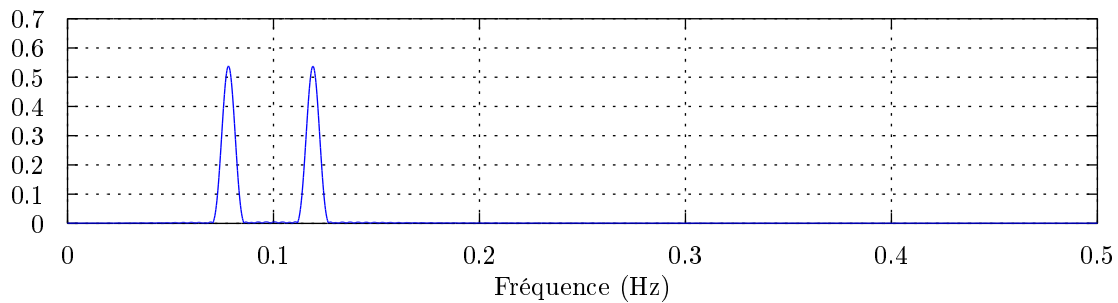
TFD de la tranche, fenêtre rectangle zero-padding : 2048

**FIGURE 24** – Effet du fenêtrage rectangulaire.

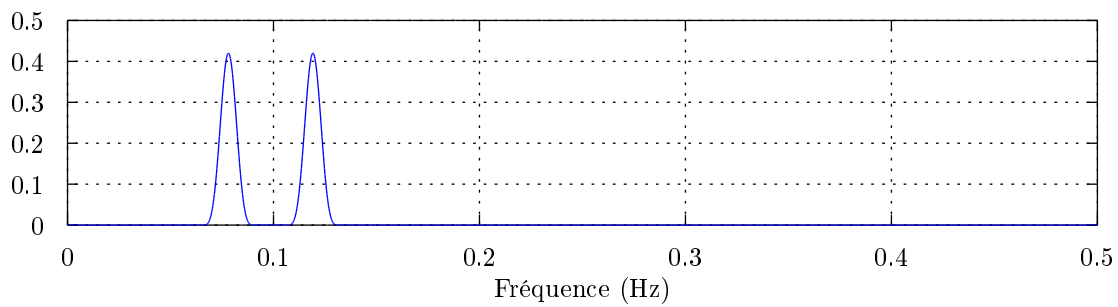
TFD de la tranche, fenêtre hanning zero-padding : 2048

**FIGURE 25** – Effet du fenêtrage de Hanning

TFD de la tranche, fenêtre hamming zero-padding : 2048

**FIGURE 26** – Effet du fenêtrage de Hamming

TFD de la tranche, fenêtre blackman zero-padding : 2048

**FIGURE 27** – Effet du fenêtrage de Blackman

Quatrième partie

Application : détection de pitch

Le but de cette partie est de réaliser une fonction capable de détecter la note jouée lors d'un enregistrement sonore. Pour cela on va découper le signal en tranches réparties régulièrement sur toute la durée de l'enregistrement.

10 Détection de signal voisé

Dans un premier temps il est nécessaire de pouvoir différencier une tranche de signal voisé d'une tranche non voisée. Nous avons vu dans la section 6 que nous pouvons classifier le signal en utilisant l'autocorrélation. On remarque qu'un signal périodique que l'on pondère par une fenêtre rectangulaire de largeur T possède une autocorrélation comprise dans une enveloppe triangulaire telle que décrite par la figure 28.

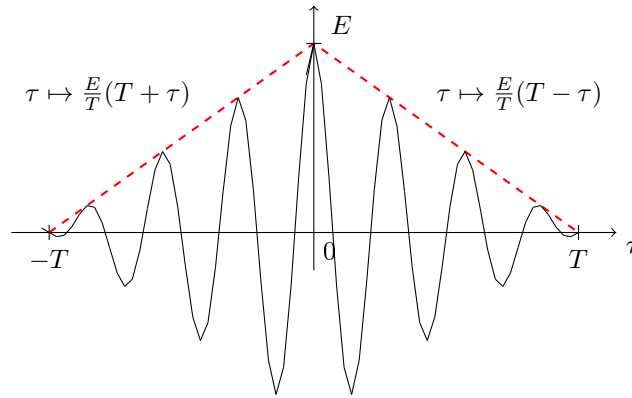


FIGURE 28 – Enveloppe du signal de l'autocorrélation pour un signal périodique pondéré par une fenêtre rectangulaire de largeur T

Afin de détecter la pseudo-périodicité dans l'autocorrélation, et donc la nature voisée du signal, on va chercher des maximums locaux dont la distance au triangle est faible. De plus ces maximums devront satisfaire un critère de périodicité, c'est à dire que si k est l'indice d'un tel maximum local, l'ensemble des $\{nk, nk \leq T\}$ doivent également correspondre à des points de l'autocorrélation proches du triangle. Enfin, l'autocorrélation étant paire, on ne regardera que les k positifs.

11 Mesure du pitch

La mesure du pitch est nécessaire à la détermination de la note. Cela se fait également sur l'autocorrélation, en mesurant la période des maximums locaux qui déterminent si le signal est voisé. On modifie donc la fonction écrite dans la section précédente afin de prendre en argument la plage de recherche du pitch (ce qui va également accélérer l'exécution de la fonction).

On renvoie un booléen, qui indique si le signal est voisé, et l'indice du maximum qui a permis de déterminer ce résultat. Si on appelle k l'indice ainsi obtenu, l'utilisateur peut déterminer la fréquence du fondamental f avec $f = \frac{f_e}{k}$ où f_e est la fréquence d'échantillonnage du signal.

12 Détection de la note

La formule donnée par le sujet du TL relie le numéro de la note à la fréquence du fondamental associé.

$$f = 440 \times 2^{\frac{n-69}{12}} \quad (14)$$

Il suffit donc d'inverser l'équation 14 pour obtenir le numéro de la note en fonction de la fréquence mesurée sur le signal. La note obtenue n'étant pas forcément juste, il faut adapter un peu pour obtenir un nombre entier à coup sûr.

$$n = \lfloor 12 \times \log_2 \left(\frac{f}{440} \right) + 69 \rfloor \quad (15)$$

Remarque : L'utilisation de la partie entière n'est ici pas forcément judicieuse. En effet si l'enregistrement joue une note de fréquence $f_0 - \epsilon$ au lieu de f_0 , alors la note détectée sera la note $n - 1$ au lieu de n . Pour corriger cela on peut modifier la manière de passer à l'entier en utilisant la fonction `round` de MATLAB.

13 Détection de la partion de musique

En utilisant les fonctions écrites précédemment, on peut écrire une fonction qui détecte la partition jouée dans un enregistrement. Pour cela on mesure la fréquence du fondamental sur chacune des tranches comportant un signal voisé, puis on fait correspondre la note MIDI associée. Pour lisser le résultat obtenu, on impose une durée minimale aux notes jouées, c'est à dire qu'une note de longueur inférieure ne sera pas détectée. En sortie on donne une matrice à 3 colonnes avec la date de début de la note, la fréquence mesurée et le numéro de la note. On remarque que la durée de la note n'est pas spécifiée. En effet on choisit de ne stocker que des notes de durée T_{min} , où T_{min} est la durée minimale d'une note. Ainsi si l'enregistrement comporte une note de durée $2.5 \times T_{min}$ de fréquence f , alors on stocke 2 notes successives de fréquence f .

14 Test de l'algorithme

On commence à tester notre algorithme sur le signal de parole enregistré 'chat', en recherchant le pitch entre 80 et 2000 Hz, pour des tranches de durée 0.1 s et des notes de durée minimale de 0.1 s. On obtient :

```
>> test_detection_pitch
```

```
notes =
```

1.3001	103.8961	44.0000
1.5001	102.5641	44.0000
1.6001	102.5641	44.0000
1.7001	101.9108	44.0000
1.8001	101.9108	44.0000
1.9001	101.2658	44.0000
2.0001	100.6289	43.0000

On trouve a un résultat cohérent avec ce qui a été trouvé dans les parties précédentes de l'étude de laboratoire. On peut donc tester notre fonction sur des signaux plus complexes.

Pour vérifier la justesse de la reconstitution, on créer un signal audio en utilisant le logiciel MuseScore et on l'exporte en fichier WAV. La partition enregistrée est celle de la figure 29.

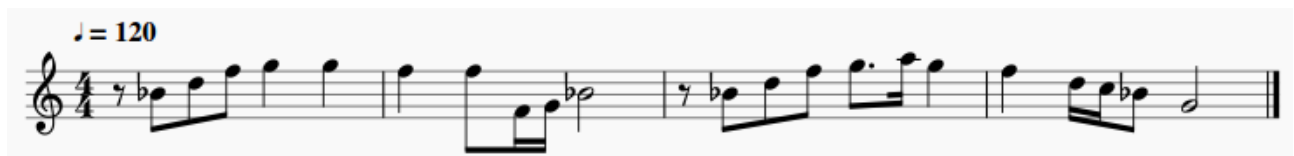


FIGURE 29 – Partition exportée.

On lance ensuite notre fonction avec une plage de recherche comprise entre 80Hz et 2500Hz, une durée minimale de la note de 0.0625s et une taille de fenêtre de 0.05s. On obtient un résultat de la forme :

```
notes =
```

0.2500	464.2105	70.0000
0.3750	464.2105	70.0000
0.4375	464.2105	70.0000
0.5625	580.2632	74.0000
0.6875	580.2632	74.0000
1.1250	773.6842	79.0000
1.5000	773.6842	79.0000
1.6250	773.6842	79.0000
1.6875	773.6842	79.0000
1.7500	773.6842	79.0000
1.8125	773.6842	79.0000
1.8750	773.6842	79.0000

1.9375773.684279.0000

2.0000700.000077.0000

2.0625700.000077.0000

...

On vérifie que les valeurs calculées des notes sont celles attendues (par exemple la note 70 est un Sib, ce qui correspond à la première note de la partition).

Afin de mieux interpréter le résultat, on symbolise les barres de mesures par des lignes verticales. La figure 30 nous permet d’apprécier la similitude entre la reconstitution et la partition originale.

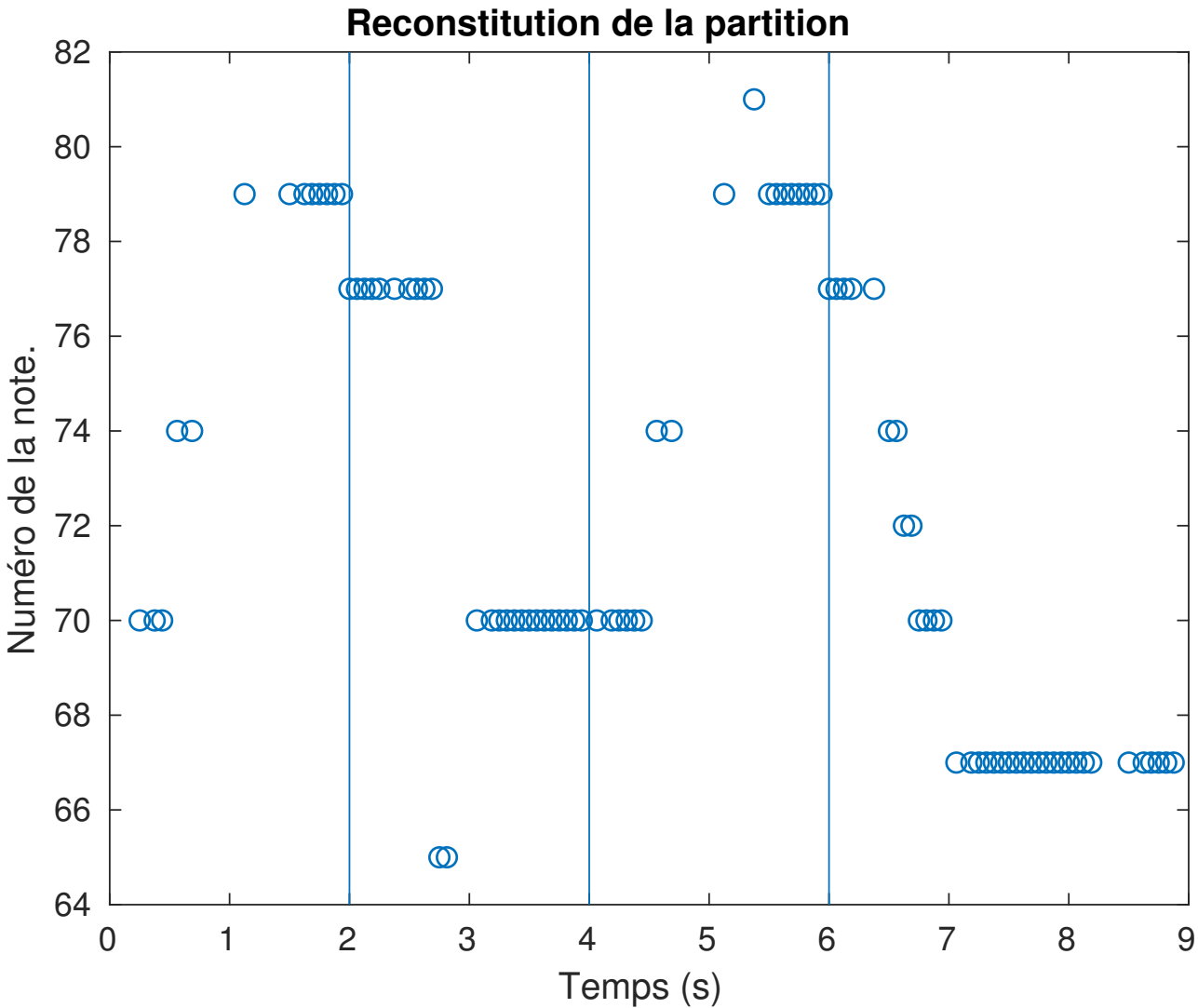


FIGURE 30 – Partition reconstituée.