14 Analyse de la parole

14.1 Introduction

L'information portée par le signal parole peut être considéré de bien des façons. On distingue généralement plusieurs niveaux de description non exclusifs : acoustique, phonétique et phonologique.

Au niveau *acoustique*, on s'intéresse essentiellement au signal que l'on tentera de caractériser par son intensité, sa fréquence, son timbre et ses propriétés statistiques. Au plan *phonétique*, on considère la génération des sons, les phonèmes qui composent un mot et les classes auxquels ils se rattachent. Enfin, la *phonologie* s'attache à décrire le rythme, la prosodie, la mélodie d'une phrase.

Le texte qui suit traitera du signal acoustique seulement. Il n'a d'autre but que de servir d'introduction à une deuxième partie où l'on abordera le codage et décodage LPC de la parole. Une présentation plus complète peut être lue avec profit dans [1].

14.2 Analyse de la parole

Avant de vouloir traiter ou coder le signal de la parole, il est important de commencer par comprendre ce qu'est la parole, quel est son contenu spectral, quelles sont les parties qui la composent. De plus, il est primordial de réaliser que l'analyse des signaux est basée sur la stationnarité de ceux-ci alors que, par essence, un message ne peut pas être stationnaire. On sera donc constamment confronté au dilemme posé par l'analyse d'un signal transitoire considéré comme stationnaire.

Pour la suite, vous utiliserez les deux outils suivants :
- le programme CoolEdit 2000 pour l'enregistrement, l'analyse auditive et visuelle des sons et de leurs spectres ;
- le programme Matlab pour l'analyse et le traitement numériques des signaux.

14.2.1 Classification des phonèmes

Lorsqu'on recherche les composants élémentaires du langage articulé, on en trouve environ une trentaine pour la langue française. Ces éléments désignés sous le nom de *phonèmes* sont répartis en 7 classes ; ils suffisent pour représenter l'ensemble des sons. Il s'agit des :
- voyelles voisées : l**i**t, **é**té, mar**ai**s, Urs**u**le, p**eu**r, p**e**tit, j**eu**, p**a**tte, p**â**te, s**o**l, s**au**le, bij**ou** ;

– voyelles nasales : br**in**, br**un**, ch**an**t, b**on**jour ;
– semi-voyelles : pa**ill**e, l**ui**, L**oui**s ;
– consonnes fricatives : **s**au**c**i**ss**on, **z**èbre, **ch**at, **j**anvier, **f**ameux, **v**ert ;
– consones nasales : N**an**tes, m**en**the, a**gn**eau ;
– consonnes liquides : sa**l**on, bu**r**eau ;
– consonnes plosives : **p**ari, **b**ar**b**are, ba**t**eau, ba**d**aud, é**c**art, lan**g**ue.

Ces classes de phonèmes font intervenir à des degrés divers les lèvres, la cavité nasale, la langue, le palais, la glotte et les cordes vocales. Des différences subtiles entre phonèmes déterminent le sens du mot et modifient sensiblement la forme de l'onde sonore et son spectre. Ces différences ne sont pas faciles à détecter et à mettre en oeuvre.

Dans certaines applications, en téléphonie par exemple, on peut se contenter d'une approche plus grossière et de répartir les phonèmes dans deux classes seulement, les sons voisés et non voisés. Les premiers sont modélisés par un signal périodique, alors que les seconds sont représentés par un bruit. Une tâche difficile du codage de la parole consiste à déterminer si un son est voisé ou non.

14.2.2 Période des sons voisés

Considérant que les sons voisés ont un contenu périodique bien marqué, le problème à résoudre consiste à trouver la période de la composante fondamentale et à décider si le son analysé est voisé ou non. Cette période (communément appelée le pitch), est un paramètre très important pour la synthèse de la parole car l'oreille est très sensible à ses variations.

On a observé que la fréquence de la fondamentale se situe entre 40 Hz et 250 Hz pour les voix masculines alors qu'elle est comprise entre 150 Hz et 700 Hz pour les voix féminines. De manière générale, on admettra donc qu'un son est voisé si sa période ou le pitch est compris entre 2 msec et 20 msec.

14.3 Acquisition et analyse avec CoolEdit

On utilisera le logiciel CoolEdit pour acquérir des sons, sélectionner et écouter des phonèmes ou des parties de phrases et visualiser des ondes sonores à l'aide de graphes, de spectres ou de spectrogrammes.

14.3.1 Paramètres pour l'enregistrement

Considérant que l'on s'intéresse ici à des sons de la bande téléphonique, on les enregistrera en monophonie à la fréquence de 8 kHz avec un convertisseur 16 bits après un filtrage antirepliement des fréquences supérieures à 4 kHz.

Pour que les fichiers soient directement utilisables par Matlab, on les sauvegardera dans un fichier *.txt de type ASCII et on prendra garde à supprimer les 4 premières lignes qui contiennent des informations sur l'enregistrement.

14.3.2 Visualisation des signaux et de leur spectre

Dans la figure 14.1, on présente le signal correspondant au mot "bonjour". On y voit le graphe du signal complet, son spectrogramme et deux zones du son "bonjour". La première correspond au son non voisé "j", alors que la deuxième illustre le phonème voisé "ou". Pour chacune de ces deux zones, on a tracé les signaux temporels et les spectres correspondants.

Pour le son "j", on relèvera le caractère aléatoire du signal, sa faible puissance et le fort taux de passages par zéro. Pour le son "ou", on notera d'abord son caractère périodique qui conduit aux raies spectrales du domaine fréquentiel. La périodicité basse fréquence des signaux voisés conduit à un faible taux de passages par zéro. De plus, la puissance des sons voisés est sensiblement plus grande que celle des sons non voisés.

14.4 Analyse du signal acoustique avec Matlab

Le logiciel Matlab servira pour traiter les signaux par tranches successives, extraire leurs caractéristiques et mettre en évidence les résultats obtenus.

Après avoir enregistré une phrase ou un son avec CoolEdit, celui-ci doit être sauvegardé dans un fichier *.txt de type ASCII afin de pouvoir être lu par Matlab. Il ne doit contenir aucune autre information que les valeurs échantillonnées du signal.

14.4.1 Lecture du fichier de données

Le fichier *.txt comportera une colonne de N valeurs échantillonnées. Il sera lu par Matlab sous la forme d'un vecteur dont l'amplitude sera normalisée à 1 :

```
[filename,path] = uigetfile('*.txt','Choix de la phrase');
phrase = load(filename);
phrase = phrase/max(abs(phrase));
```

On désignera une tranche à l'aide de la variable `st`. La tranche désirée peut être sélectionnée en prenant une partie des composantes du vecteur `phrase` avec la commande

```
st = phrase(Ndebut :Nfin);
```

Si une analyse spectrale doit être faite, on choisira de préférence une tranche de longueur égale à une puissance de 2. Par exemple, 128 ou 256.

14.4.2 Initialisation

La visualisation du signal temporel et de son spectre débute par l'initialisation de quelques variables et la suppression de la valeur moyenne qui n'a aucun intérêt en traitement des signaux :

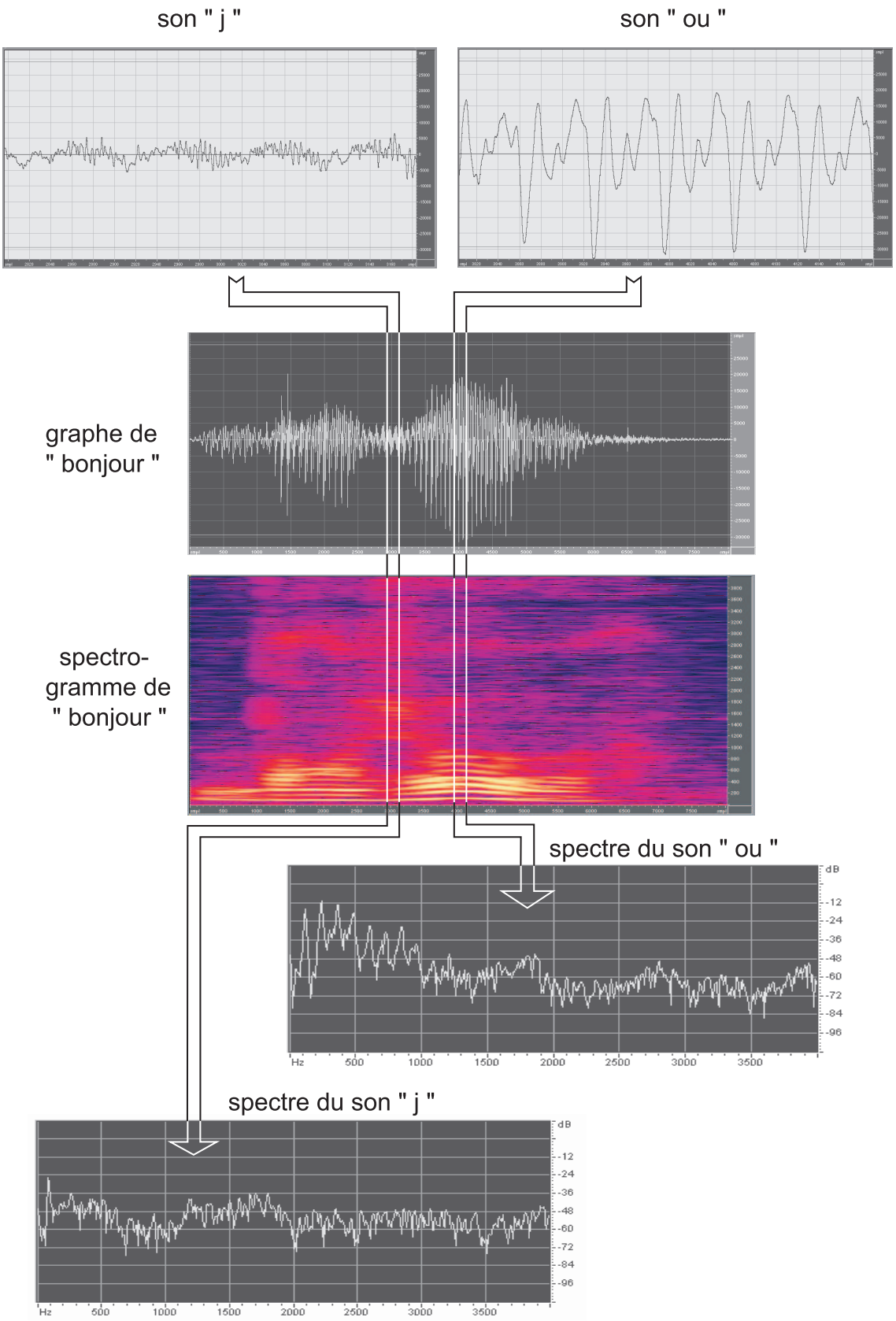

Fig. 14.1: Graphes correspondant au mot “bonjour” ; mise en évidence des sons “j” et “ou”

```
fe = 8e3 ; Te = 1/fe ;
Npoints = length(st) ;
duree = Npoints*Te ;
temps = 0 :Te :duree-Te ;
df = 1/duree ;
Ndemi = fix(Npoints/2) ;
frequence = df*(0 :Ndemi-1) ;  % 0 <= frequence < fe/2
st = st - mean(st) ;           % suppression de la valeur moyenne
```

14.4.3 Valeur efficace

Vous remarquerez par la suite que les sons voisés sont généralement plus intenses que les sons non voisés. Pour évaluer l'amplitude des signaux, on calcule la valeur efficace de la tranche considérée. Celle-ci est égale à la déviation standard du signal

```
Seff = std(st) ;               % valeur efficace du signal
```

14.4.4 Taux de passages par zéro

Le taux de passages par zéro peut également aider à la décision voisé / non voisé. Il est défini comme le rapport entre le nombre de passages par zéro et le nombre d'échantillons considérés

$$n_{xz} = \frac{N_{xz}}{N_{ech}}$$

Le nombre passages par zéro peut se calculer comme suit :

```
function [Nxz] = zcross(xt)
  xz = xt - mean(xt) ;
  xz = (1+sign(xz))/2 ; % transformation en un signal binaire 0 / 1
  xz = diff(xz) ; % derivee du signal binaire = +/- 1
  Nxz = sum(abs(xz)) ; % nombre de passages par 0
```

14.4.5 Spectre

L'analyse spectrale est faite à l'aide de la FFT. Idéalement, le nombre de points de la tranche analysée devrait être une puissance de 2. Afin d'éviter les effets de bords de la tranche qui peuvent conduire à un étalement spectral, il est nécessaire d'effectuer préalablement un fenêtrage de la tranche. Ces opérations sont réalisées à l'aide des commandes suivantes :

```
stHm = st.*Hamming(Npoints) ;
spectre = fft(stHm) ;
spectre = spectre(1 :Ndemi) % limitation à fe/2
module = abs(spectre) ; phase = angle(spectre) ;
plot(frequence,20*log10(abs(spectre)) ;
```

Une illustration de sons voisés et non voisés est donnée dans les figures 14.2 et 14.3. On notera les raies spectrales bien visibles dans le spectre du signal voisé et, en particulier, la correspondance entre la fréquence de la fondamentale et la période du signal voisé.

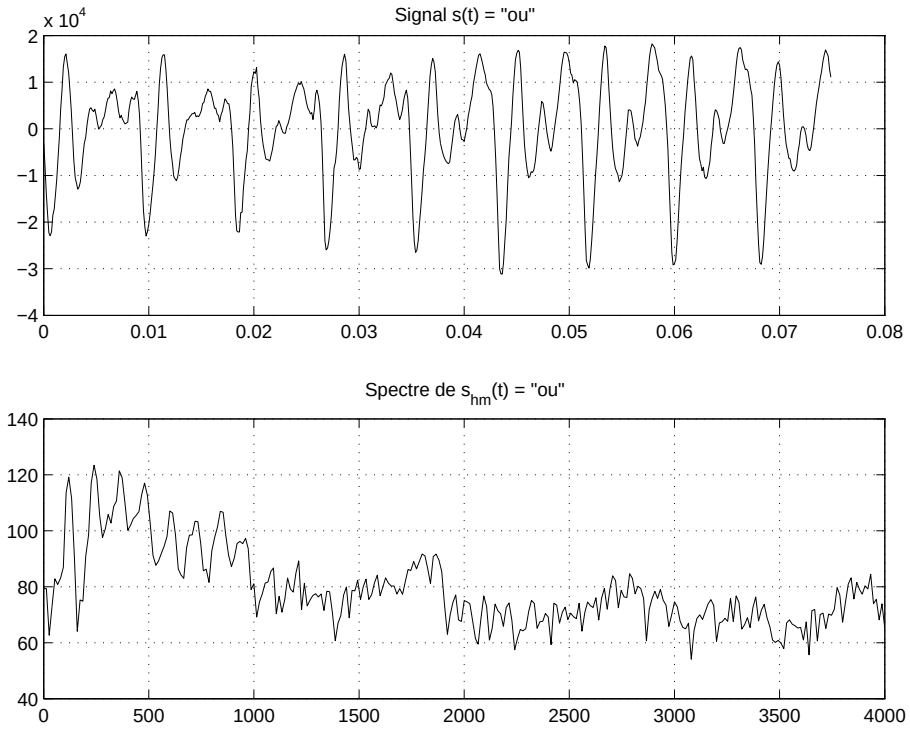

Fig. 14.2: Signal voisé et son spectre

14.5 Recherche du pitch

14.5.1 Filtrage du signal

Comme on l'a dit plus haut, on admet généralement que la période du pitch de la voix humaine est comprise entre 2 et 20 msec. Le domaine spectral qui nous préoccupe ici est donc inférieur à 500 Hz. Il est ainsi préférable, puisqu'on s'intéresse à un signal dont le spectre est limité, de commencer par éliminer les fréquences supérieures à 500 Hz. Ceci peut être fait à l'aide d'un filtre numérique passe-bas ; celui-ci est généralement du type Butterworth et d'ordre 8 :

```
fc = 500 ; fn = fe/2 ; ordre = 8 ;
[nbtw dbtw] = butter(ordre, fc/fn) ;
stf = filter(nbtw, dbtw, st) ;
```

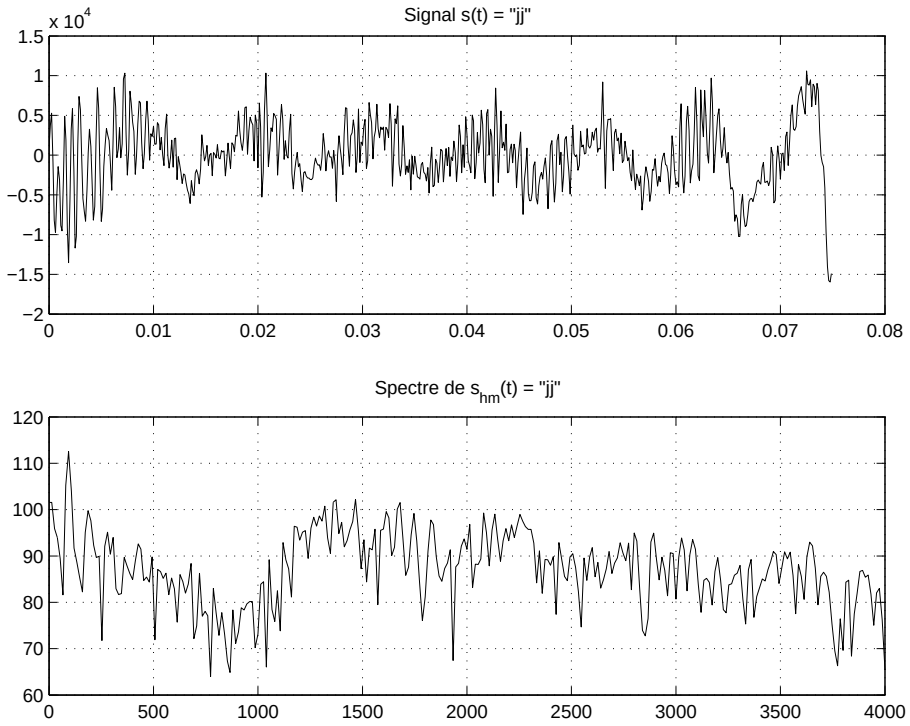

FIG. 14.3: Signal non voisé et son spectre

14.5.2 Autocorrélation

On a vu que la tranche considérée est périodique si le son est voisé et aléatoire dans le cas contraire. Afin de faciliter la recherche de la période, on travaille de préférence avec la fonction d'autocorrélation car celle-ci est généralement moins bruitée que le signal lui-même (figure 14.4).

Le résultat de l'autocorrélation est un vecteur de longueur $2N$ avec un maximum en son milieu. Si le signal est périodique, d'autres pics distants de la valeur du pitch seront présents. Pour trouver ce dernier, il suffit donc de mesurer la distance entre pics successifs.

Les commandes sont alors les suivantes :

```
% autocorrélation d'une tranche st filtrée
  rss = real(xcorr(stf))/Npoints ;
  [rssmax k0] = max(rss) ;          % k0 = position du max central
  rss = rss(k0 :length(rss)) ;      % partie droite de rss

% le 1er pic latéral doit se trouver entre Tpmin et Tpmax
  fpmax = 500 ;                     % fréquences min et max du pitch
  fpmin = 50 ;
  Tpmax = 1/fpmin ;                 % périodes min et max du pitch
  Tpmin = 1/fpmax ;
  kpmin = round(Tpmin/Te) ;         % compteurs liés à Tpmin et Tpmax
  kpmax = round(Tpmax/Te) ;
```

```
% recherche du premier pic
  rss = rss(kpmin :kpmax) ;          % domaine limité par Tpmin et Tpmax
  [ymax k1] = max(rss) ;             % k1 = position du 1er max latéral

% entier correspondant à la période du pitch
  kp = kpmin + k1 ;
  Tp = kp * Te ;
```

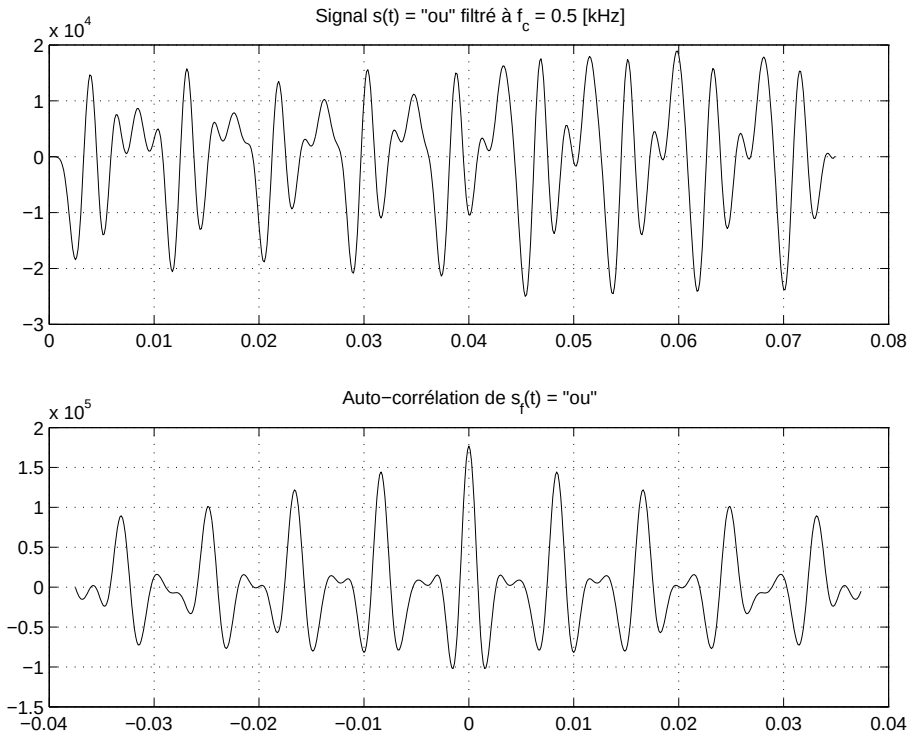

Fig. 14.4: Autocorrélation d'un son voisé

14.6 Travail pratique

Pour aborder l'analyse de la parole, je vous propose de travailler sur la phrase "Le colibri a chanté" (fichier colibri.txt) ou une phrase de votre choix.

14.6.1 Avec CoolEdit :

1. Chargez le fichier `colibri.txt` en mode `mono / 16 bits / 8 kHz`.
2. Écoutez la phrase ; observez le graphe temporel et le spectre (`Analyze/Frequency Analysis`) de diverses parties de la phrase ; notez que le temps peut être gradué en secondes ou en échantillons avec le bouton droit de la souris placé sur l'axe temporel.

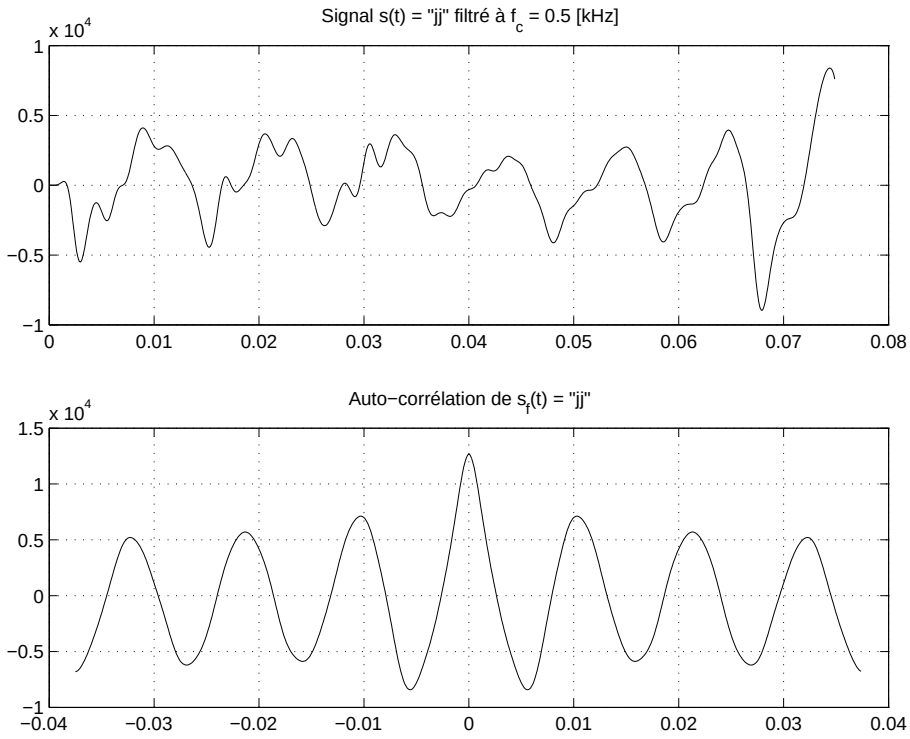

FIG. 14.5: Autocorrélation d'un son non voisé

3. Sur le spectrogramme (`View/Spectral View`), observez l'effet de la durée d'analyse N sur les résolutions temporelle et fréquentielle en prenant 128, 256 et 512 échantillons (`Options/ Settings/ Spectral`).
4. Qu'en est-il de la relation liant les résolutions temporelle et spectrale ? quelle durée d'analyse choisissez-vous ?
5. Mentionnez tous les phonèmes que l'on trouve dans cette phrase ; à quelles familles appartiennent-ils ?
6. Relevez les numéros d'échantillons correspondant au début et à la fin des phonèmes.
7. Dans l'enregistrement, choisissez librement au moins deux tranches voisées et non voisées.
8. Observez leurs spectres et spectrogramme puis analysez plus en détail leurs caractéristiques temporelles et spectrales.

14.6.2 Avec Matlab

Procédez à l'analyse détaillée de plusieurs tranches voisées / non voisées et tentez de les séparer automatiquement. Pour cela :

1. Extrayez de la phrase les zones que vous souhaitez analyser.
2. Mesurez leur valeur efficace et taux de passages par zéro ($< 100\%$!).
3. Recherchez la période du signal avec la fonction de corrélation.
4. Voyez-vous un moyen de séparer automatiquement les sons voisés / non voisés ?

Bibliographie

[1] R. Boite et al., *Traitement de la parole*, PPUR, 2000