TD Traitement du signal

Ces travaux dirigés nécessitent le logiciel libre Scilab (version 5.3) [1].

1 Introduction rapide à Scilab

Les instructions doivent être tapées après le symbole $-->$ (le *prompt*). La commande `cd` permet de se rendre dans le répertoire de travail. Lorsque le nombre d'instructions devient important, il est utile d'écrire un script et de l'exécuter. Pour cela, il faut ouvrir Scipad et enregistrer un fichier au format `.sce`. Par convention, l'extension `.sci` est réservée aux fonctions. L'exécution peut être effectuée sous Scipad en cliquant dans le menu (en haut à droite) sur *Exécuter>...fichier sans écho*. La combinaison de touches *Ctrl+C* suivi de `abort` permet d'interrompre un calcul en cours. De l'aide peut être obtenue avec les commandes `help` et `apropos`. Les commandes commençant par `//` sont considérées comme des commentaires. Suivant que la ligne de commande se termine ou non par un point-virgule, le résultat du calcul s'affiche ou non. La fonction `plot` émule la fonction du même nom de Matlab (elle remplace `plot2d`). Lors du débogage d'un programme, il peut être utile d'afficher le contenu d'une variable avec `disp`.

1. Le script *sript.sce* :
```
clear // efface toutes les variables encore en mémoire
exec('\Func_square.sci'); // déclaration de la fonction Func_square
x=0:0.01:10; // génère un vecteur x allant de 0 à 10 par pas de 0.01
y=Func_square(x);
figure(1) // impose la fenêtre 1
plot(x,y,'-b'); // tracé d'une courbe continue en bleu
xtitle('sound.wav','t','y(t)') // légende la figure : titre, abscisses, ordonnées
a=gca();
a.data_bounds=[min(x),min(y);max(x),max(y)]; // axes
xstring(min(x),min(y),'N ='+string(length(y))); // écrire sur le plot
```

2. La fonction *Func_square.sci* :
```
function [y]=Func_square(x)
  y=x^2;
endfunction
```

2 Chronogramme et échantillonnage

Le **chronogramme** est la représentation temporelle d'un signal $s = f(t)$.

1. Générez et visualisez un signal sinusoïdal s d'une durée de 1 s, de fréquence F_0=100 Hz. Quelle est la longueur du vecteur s ? Le pas temporel doit être faible pour représenter un signal à temps continu.
2. Quelle est la fréquence minimale d'échantillonnage F_e ? Échantillonner s au-dessus de cette fréquence.

1. http ://www.scilab.org/

3. Reconstruire le signal s en utilisant la formule (1) d'interpolation idéale de Shannon, où $T_e = 1/F_e$. Échantillonner s en-dessous de F_e et comparer avec le signal d'origine.

$$s(t) = \sum_{k=-\infty}^{+\infty} s(kT_e)\,\mathrm{sinc}\,(\pi F_e(t - kT_e)) \tag{1}$$

3 Visualiser le spectre

La transformée de fourier sur des valeurs discrètes est appelée TFD (transformée de fourier discrète). Divers algorithmes permettent d'accélérer les calculs de la TFD (on les appelle FFT pour *fast Fourier transform*). La fonction `fft` de Scilab emploie l'algorithme de Cooley et Tuckey [2] qui nécessite un nombre d'échantillons en puissance de 2.

```
N=length(y);      // nombre de points
dnu=Fs/(N-1);
F=dnu*(0:1:N-1); // vecteur fréquences
X=abs(fft(y))/N; // module des coefficients de Fourier complexes
X=X/N;           // normalisation à l'amplitude
```

1. **Efficacité de la FFT :** On rappelle que la TFD d'un signal f échantillonné à une période T_e est
$$S(kF_e/N) = \sum_{n=0}^{N-1} f\,(nT_e) \exp\left(-2\pi i n k/N\right), k \in \{0, 1, ..., N-1\} \tag{2}$$
où N est le nombre d'échantillons, T_e la période et F_e la fréquence d'échantillonnage. Implémentez le calcul de la TFD dans une fonction. Assurez vous que les deux fonctions donnent le même résultat, puis comparez les temps de calcul entre la TFD et la FFT (grâce à la fonction `timer()`). Tester sur des signaux de taille 2^p croissantes.
2. **Retour à l'échantillonnage :** copiez le son *sound.wav* dans votre dossier de travail.
```
[y,Fs]=wavread('sound.wav'); // le signal est chargé dans le vecteur y
t=(0:1:length(y)-1)/Fs;      //reconstruction du vecteur temps
```
• Visualisez le signal temporel de ce son $s(t)$ et son spectre $S(f)$. On voit apparaître un spectre symétrique entre $F_e/2$ et F_e, lié à l'échantillonnage : le spectre initial réapparaît aux fréquences multiples de F_e. La commande `fftshift` permet de symétriser le spectre autour de zéro (les fréquences de l'intervalle $[F_e/2, F_e]$ sont envoyées dans l'intervalle $[-F_e/2, 0]$).
• Quelle est la plus basse fréquence f_{min} présente dans ce signal ? Et sa plus haute fréquence f_{max} ? Que signifie la valeur du spectre en $f = 0$?
3. **Transformée de Fourier inverse** : à partir du spectre, retrouvez le signal original grâce à la fonction `ifft`.
4. **Sur-échantillonnage** : on veut augmenter la fréquence d'échantillonnage d'un facteur M. Pour trouver la valeur des M-1 points à ajouter entre les points initiaux de s, on peut songer à utiliser une interpolation polynômiale. Une autre solution consiste à insérer des échantillons nuls. Sous Scilab, on peut utiliser la fonction `matrix` appliquée à une matrice dont la première ligne est le vecteur s et les M-1 autres lignes sont remplies de 0 grâce à `zeros`. Comparez les spectres Sf et SMf. Quelle est la fréquence maximale du signal sur-échantillonné SMf ? Combien observe-t-on de répliques du spectre initial Sf ? Pour obtenir le même signal que l'original mais avec M fois plus d'échantillons, il faut appliquer un filtre passe-bas. Quelle est la fréquence de coupure de ce filtre ?

2. James W. Cooley and John W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. **19** 297 (1965).

5. **Sous-échantillonnage** : l'opération consistant à ne retenir qu'un échantillon sur M (décimation) n'est pas correcte. Pourquoi ? Quelle étape est nécessaire avant la décimation ? Le sous-échantillonnage est une forme de compression. Quel est le gain de cette compression ?
6. Visualisez le spectre d'une fonction porte (comparez avec la fonction théorique). Comment varie la largeur du pic fréquentiel lorsqu'on change la largeur de la porte ?
7. **Technique du zéro padding** (ou remplissage par des zéros) : elle consiste à ajouter aux N points du signal, une séquence de M valeurs nulles afin d'obtenir un plus grand nombre de points de fréquence. Quelle est la conséquence sur le pas en fréquence ?
8. **Précision en fréquence** : On génère un signal de fréquence variable f_0 et on y prélève un segment de longueur N=32. Que constatez vous sur le spectre lorsque f_0 est multiple de F_e/N ? Dans le cas contraire ? Pour comprendre ce qui se passe, superposez le spectre d'un segment de longueur N=1024. Au final, quelle est la précision pour une raie monochromatique ?
9. **Effet du fenêtrage et résolution** : un signal numérique est par nature de durée limitée. Cette troncature est équivalente à multiplier le signal par une fenêtre rectangulaire (ou naturelle). L'étude de la fonction porte nous a permis d'observer la déformation du spectre associée. Il existe de nombreuses fenêtres d'apodisation : Bartlett, Hamming, Hanning, Blackman, etc. Par exemple, la fênêtre de Hamming est implémentée dans Scilab par la fonction `window('hm',n)` où n détermine la longueur du vecteur.

   • Il y a un compromis à trouver entre la largeur du lobe principal, qui définit la résolution, et la hauteur des lobes secondaires. Malheureusement, la réduction en hauteur des lobes secondaires s'accompagne toujours de l'élargissement du lobe principal. Observez cet effet pour les fenêtres rectangulaire et de Hamming en comparant les spectres d'amplitude en dB. La conversion en dB de l'amplitude se fait par $20\log_{10}(|FFT(\text{signal})|)$.

   • Quelle est la fenêtre la plus adaptée pour séparer deux raies lorsque leurs amplitudes sont similaires ? Dans le cas contraire ?

4 Convolution et filtrage

La convolution est l'opération mathématique suivante

$$y(t) = \int_{\infty}^{-\infty} h(\tau)x(t-\tau)d\tau.$$

Cette équation s'écrit symboliquement $y = h * x$. Une propriété essentielle de la convolution est que la transformée de Fourier d'un produit de convolution est un produit simple (parfois appelé théorème de Plancherel) :

$$Y(\nu) = TF\{y(t)\} = TF\{h(t) * x(t)\} = TF\{h(t)\} \cdot TF\{x(t)\} = H(\nu) \cdot X(\nu)$$

L'opération de filtrage dans le domaine fréquentiel est donc une opération de convolution dans le domaine temporel.

1. La commande Scilab est `convol`. Exemple de syntaxe : `y=convol(f1,f2)`. Les vecteurs f1 et f2 doivent avoir le même pas temporel. Quelle est la dimension du vecteur y par rapport aux dimensions de f1 ou f2 ?
2. Quelle est la convolution d'un sinus et du vecteur `h=[0 0 1 0 0]` ? De deux signaux rectangulaires ? De deux signaux triangulaires ? Utilisez la fonction `window('tr',...)` pour générer facilement les signaux triangulaires.

3. Dans l'espace des fréquences : décaler un filtre passe-bas idéal (modélisé par une fonction porte) d'une fréquence f_{mod} pour créer un filtre passe-bandes. Par TF inverse, observer que la translation en fréquence d'un filtre se traduit dans le domaine temporel par une modulation par une fonction $\cos(2\pi f_{mod} t)$.
4. Ecrire une fonction qui filtre le signal dans le domaine temporel par un filtre passe-bas idéal $H(f) = 1$ si $|f| \leq F_0$ et 0 sinon. On rappelle que la fonction duale est $h(t) = \frac{\sin 2\pi F_0 t}{\pi t}$.

5 Transformée de Fourier à court terme (ou glissante)

Cette méthode permet l'étude des signaux non stationnaires, c'est-à-dire des signaux qui évoluent dans le temps. L'analyse de Fourier à fenêtre glissante (FFG, ou court terme) permet de visualiser le signal dans un plan *temps-fréquence*. Plus récemment s'est développée la transformée en ondelettes, apparue dans les années 1980 avec les travaux de J. Morlet sur les signaux sismiques. Dans cette transformée, on remplace les sinusoïdes par des ondelettes adaptées au signal à étudier.

Le principe est le suivant : le signal x est découpé en petits intervalles temporels (de taille $2p$, fenêtrés) dont on fait la TF (les intervalles peuvent se chevaucher). Chaque TF est rangée dans une colonne de la matrice finale S. Finalement, on dessine un graphe 2D, c'est-à-dire une surface (une valeur étant indexée par une couleur) appelée *spectrogramme* :

```
// Coeur du code de FFG
N=length(x);T=2*p/fe;F=[1:p]/T;t=[];S=[];
for k=p+1:p+n:N-p-1
    t=[t k/fe];
    xw=x(k-p:k+p-1).*window('hn',2*p);
    yw=fft(xw);
    S=[S; abs(yw(1:p))];
end
// Représentation temps-fréquence
grayplot(t,F,S);
g=gcf();g.color_map=jetcolormap(64);  // change le code couleur
colorbar(min(min(S)),max((S))); // affiche une échelle colorée verticale
```

1. Générez un signal s composé d'une sinusoïde de fréquence f_1 pendant une durée T_1 suivie d'une sinusoïde de fréquence f_2 pendant une durée T_2. Le nombre total d'échantillons sera noté $N = T_1/T_e + T_2/T_e = N_1 + N_2$. Implémentez la FFG dans Scilab et appliquez-la à s.
2. Visualisez un signal dont la fréquence varie linéairement avec le temps ($s = \cos(At^2)$) en variant la pente A. Que se passe-t-il lorsque la pente est trop forte ? A quoi est dû ce phénomène ?
3. **Principe d'incertitude :** générez un son dont la fréquence varie sinusoïdalement avec le temps (on fait une modulation de fréquence $s = \cos[\Omega t + A\sin(\omega t)]$) où $A = |\Omega - \omega|/\omega$, puis variez la taille de la fenêtre. Montrez que quand on améliore la résolution en temps on détériore la résolution en fréquence.
4. **Densité spectrale de puissance :** au lieu de présenter le spectre du signal original, on représente souvent la densité spectrale de puissance. Pour cela, on calcule le module au carré de la transformée de Fourier de chaque segment. La densité spectrale de puissance est alors donnée par la moyenne de l'énergie spectrale sur L segments consécutifs.