TD Analyse Spectrale - Transformée de Fourier

On cherche souvent à détecter et analyser des phénomènes périodiques à partir de données expérimentales. Par exemple, on peut vouloir calculer des cycles d'activités solaires à partir des rayons cosmiques qui arrivent du soleil afin d'en déterminer l'influence sur le climat terrestre. C'est le but de l'analyse spectrale que l'on retrouve dans tous les domaines de la physique et de l'ingénierie.

En pratique, on possède un nombre fini N d'échantillons de mesures également espacées dans le temps et l'on utilise la Transformée de Fourier Discrète (TFD) pour en obtenir une représentation spectrale.

La TFD d'un signal discret $y(n), n = 0 \dots N-1$ (de période d'échantillonnage T_e, ce qui signifie que l'échantillon $y(n)$ a été mesuré à l'instant $t(n) = (n-1)T_e$) périodique (de période $T = NT_e$) est définie par :

$$Y_m = \frac{1}{N} \sum_{n=0}^{N-1} y_n e^{-2\pi jmn/N}; \quad m = 0, 1, \dots N-1 \tag{1}$$

À chaque point Y_m de la transformée on associe une fréquence discrète :

$$f_m = \frac{m}{T_e N} \quad . \tag{2}$$

Naturellement, la fréquence non nulle la plus basse est $f_1 = 1/T_eN = 1/T$. La TFD de y est elle-même discrète (de période d'échantillonnage $\frac{1}{T}$) et périodique (de période $\frac{1}{T_e} = f_e$). De ce fait, on peut la représenter sur n'importe quel intervalle de durée $\frac{1}{T_e}$, par exemple sur $[0, f_e]$ ou $[-f_e/2, f_e/2]$.

La Transformée de Fourier Discrète inverse est donnée par :

$$y_n = \sum_{m=0}^{N-1} Y_m e^{2\pi jmn/N}. \tag{3}$$

Cette normalisation n'est pas universelle. En particulier, dans les travaux mathématiques on préfère la normalisation symétrique avec $\frac{1}{\sqrt{N}}$ pour la transformée et pour l'inverse.

Faire attention au fait que les indices des vecteurs `MATLAB` commencent avec 1 mais que m et n commencent à 0. Faire également attention au fait que la transformée est un vecteur de nombres complexes. On préfére parfois représenter le spectre d'énergie defini par :

$$S_m = |Y_m|^2 = Y_m Y_m^* \tag{4}$$

où Y_k^* est la complexe conjuguée de Y_k (On utilisera en `MATLAB` la fonction `conj`).

Pour comprendre la nature de la transformée de Fourier on considère dans un premier temps la transformée d'un signal parfaitement périodique et l'on examinera ensuite des donnés expérimentales.

a. Écrire un programme pour calculer la transformée de Fourier discrète d'un cosinus de fréquence 0,2Hz échantillonné à 1Hz et de durée 50 secondes. Naturellement on s'attend à voir une seule fréquence dans la transformée. Pour mieux comprendre le resultat on pourra représenter la TFD sur l'intervalle $[-f_e/2, f_e/2]$ et se souvenir que le cosinus est une somme d'exponentielles complexes. Faire de même pour un sinus.

b. Changer la fréquence à une valeur telle que $f \neq f_m \; \forall m$; $f < 0{,}5$. Conclusion ?

c. Analyse des données de la concentration de CO_2 mesurée à Mauna Loa, Hawaii. Les données dans le fichiers `mauna.dat` ont été mésurées avec un intervalle de temps de 14 jours à partir de 1981. Utiliser `load mauna.dat` pour lire les données dans la matrice `mauna`. Calculer la transformée et commenter. Quelle est la période principale des oscillations ?

d. Utiliser `polyfit(t,y,1)` pour éliminer la tendance linéaire du signal *detrending*. Recalculer la transformée. Conclusion ?

Lorsqu'on cherche à analyser un phénomène continu à partir de mesures recueillies à des instants réguliers, il est important d'effectuer ces mesures avec une période d'échantillonnage suffisamment petite. Le théorème de Shannon nous permet de fixer cette période d'échantillonnage afin que les mesures effectuées soient bien représentatives du phénomène continu observé.

e. Rappeler le théorème de Shannon et précisez la notion de repliement spectral (*aliasing*).

Pour illustrer ce théorème, on va simuler un signal continu que l'on va sous-échantillonner en ne conservant qu'un échantillon sur P (avec P entier) et en mettant les autres échantillons à zéro (Programme `shannon.m`). Cela va avoir pour effet de périodiser le spectre.

f. Commencez par conserver tous les échantillons ($P = 1$) et déduire du théorème de Shannon le sous-échantillonnage maximal que l'on va pouvoir effectuer pour le signal donné.

g. Vérifier votre résultat en prenant différentes valeurs pour P.

Enfin, en guise de conclusion, pour illustrer l'importance de la phase de la transformée de Fourier, executer le programme `phase.m`.

Programme

```
%  Programme tfd.m
%
%  Calcul de la transformee de Fourier discrete
% -----------------------------------------------------------

clear all; clg

% Initialisation  (a completer)
% -----------------------------

initiales='?????????'

T    = ?  ;                % duree totale du signal (T)
freq =  ? ;                % Frequence du cosinus/sinus
Te   = ?  ;                % periode d'echantillonnage


fe = 1/Te;                         % Frequence d'echantillonnage
N = 2*ceil(T/Te/2);                % Nombre pair de points

% calcul de la fonction
```

```
% ----------------------

for n = 0:N-1;
    t(n+1) = n*Te ;
    y(n+1) = ? ;
end

%%%%  Exercice 5 %%%%
% Enlever les commentaires et completer
%Te = ?;

%load mauna.dat;
%N = length(mauna);
%y=mauna;
%ftau=1/tau
%t=Te*[1:N]';
%p = polyfit(t,y,1)
%for n = 1:N
%lin(n) = p(1)*t(n) + p(2);
%y(n) = y(n) - lin(n);
%end
%%%%%            %%%%%

% Calcul de la tranformee de Fourier
% ------------------------------------

for m=0:N-1

   somme = 0;

   for n=0:N-1
    somme = somme + ? ;
   end

   Y(m+1)  = ? ;                % Transformee de Fourier Discrete
   Se(m+1) = ? ;                % Spectre d'energie

end

% Construction du vecteur des frequences (a completer)
% ---------------------------------------------------

  fmin= 0;

  for m=1:N
      fk(m) = ?;
  end

  f=fmin+fk;

% Lors du passage de [0,fe] \'a [-fe/2,fe/2] enlever les commentaires
%  Y  = [Y(N/2+1:N) Y(1:N/2)];
%  Se = [Se(N/2+1:N) Se(1:N/2)];


% Partie graphique
% ----------------

figure(1)
subplot(211)
  plot(t,y);
  title(['Signal- ', initiales]);
  ylabel('Amplitude');
  xlabel('Temps')
```

```
subplot(212)
  plot(f,Se);grid
  title('Spectre d''energie')
  xlabel('Frequence')

% ----------------------------

figure(2)
subplot(211)

  plot(f,real(Y));grid
  title(['Partie reelle ', initiales]);
  ylabel('T.F.D.')
  xlabel('Frequence')

subplot(212)
  plot(f,imag(Y));grid
  title('Partie Imaginaire')
  ylabel('T.F.D.')
  xlabel('Frequence')


%-----------------------------------------------------------------------


% Programme Shannon.m
%
% Transformee de Fourier Discrete, echantillonnage, repliement
%
%------------------------------------------------------------
nom = '???';

%---------------------------------------------------------------------
% On dispose d'un signal x sur un tres grand nombre de points.
% On peut assimiler ce signal a un signal continu.
% On va illustrer la construction de la TFD, et les effets de
% l'echantillonnage.
%---------------------------------------------------------------------

%---------------------------------------------------------------------
% 1 - Etude du signal
%---------------------------------------------------------------------

% Construction du signal
 lx = 128;                                      % Longueur du signal
 t  = 0:lx-1;                                   % Temps
 x = sin(2*pi*t/20).*exp(-1/2/300*(t-60).^2);   % Signal

% Visualisation du signal
  figure(1)
  clg
  subplot(211);
  plot(x); grid;
  title([nom,' Signal original']);
  xlabel('temps')

% Calcul visualisation du module de la TF
  X = fft(x);
% Echelle des frequences arbitraire allant de -1/2 a 1/2
  f = ( [1:lx] - (lx/2+1) )/(lx);
  subplot(212);
  plot(f,fftshift(abs(X))); grid;
```

```
  title(sprintf('Module de la TF du signal discret (%dpoints)',lx));
  xlabel('Frequence')

%----------------------------------------------------------------------
% 2 - Echantillonage du signal continu
%----------------------------------------------------------------------
% On va verifier que l'echantillonnage d'un signal continu provoque
% une periodisation de sa TF.
%
% On va simuler l'echantillonnage d'un signal continu en ne retenant
% dans y qu'un point sur n de x et en mettant les autres a zero
%----------------------------------------------------------------------
  n = input('Facteur de sous echantillonnage P: ');
  fprintf('On ne conserve qu''un point sur %d du signal original\n',n);
  y = zeros(1,lx);
  for k = 1:lx/n,
      y(n*k) = x(n*k);
  end

% On superpose les signaux continus et echantillonne
  figure(2)
  clg
  subplot(211)
  plot(x,'-');
  hold on;
  plot(y,'g--');
  hold off;
  title([nom, 'Signaux continu et << echantillonne >>'])

% On calcule et affiche le module de la TF de xx et de yy
  Y = fft(y);
% Pour conserver la meme amplitude, on multiplie cette TF n
% correspondant a la perte moyenne en energie du signal lors de
% l'echantillonnage.
  Y = n*Y;
  subplot(212)
  plot(f,fftshift(abs(X)),'-')
  hold on
  plot(f,fftshift(abs(Y)),'g--')
  hold off
  title('Module de la TF des signaux <<continu>> et <<echantillonne>>')
  xlabel('Frequence'); grid on

%------------------------------------------------------------------------


%  Programme phase.m
%
%  Illustration de l'importance de la phase de la TFD
%
% ------------------------------------

clear all
%%%% CHARGEMENT DES IMAGES
% Chargement de l'image x a partir du fichier clown.mat
  load clown;
  x = X;

% Chargement de l'image x a partir du fichier gatlin2.mat
  load gatlin;
%  load clown;
  y = X;

% mise a la meme taille de x et y
```

```
  l = min(size(x,1),size(y,1));
  c = min(size(x,2),size(y,2));
  x = x(1:l,1:c);
  y = y(1:l,1:c);

%%%%% CALCUL DES NOUVELLES IMAGES
% Calcul des TF de x et y
  X=fft2(x);
  Y = fft2(y);

% Construction de z1 avec le module de X et la phase de Y
  Z1 = abs(X).*exp(i*angle(Y));

% Construction de z2 avec le module de Y et la phase de x
  Z2 = abs(Y).*exp(i*angle(X));

% Calcul des TF inverses de z1 et z2
% On en prend la partie reelle car la partie imaginaire est theoriquement
% nulle mais elle vaut numeriquement 10^-14
  z1 = real(ifft2(Z1));
  z2 = real(ifft2(Z2));

%%%%%% AFFICHAGE
% Affichage des TF de x et y : Module et Phase
  figure(1)
  fh = ([1:l]-l/2)/l;
  fv = ([1:c]-c/2)/c;
  subplot(3,2,1); image(x); axis('image'); title('image x');
       axis off; colormap(gray)
  subplot(3,2,2); image(y); axis('image'); title('image y'); axis off
  subplot(3,2,3); imagesc(fh,fv,log(abs(fftshift(X.^2))));
       axis('image'); title('Module de X');
  xlabel('frequences horizontales');ylabel('frequences verticales')
  subplot(3,2,4); imagesc(fh,fv,log(abs(fftshift(Y.^2))));
       axis('image'); title('Module de Y');
  xlabel('frequences horizontales');ylabel('frequences verticales')
  subplot(3,2,5); imagesc(fh,fv,angle(fftshift(X)));
       axis('image'); title('Phase de X')
  xlabel('frequences horizontales');ylabel('frequences verticales')
  subplot(3,2,6); imagesc(fh,fv,angle(fftshift(Y)));
       axis('image'); title('Phase de Y')
  xlabel('frequences horizontales');ylabel('frequences verticales')

% Affichage des images construites z1 et z2
  figure(2)
  subplot(2,2,1); image(x); axis('image'); title('image x'); axis off
  subplot(2,2,2); image(y); axis('image'); title('image y'); axis off
  subplot(2,2,3); image(z1); axis('image'); axis off
  title('image module de x phase de y')
  subplot(2,2,4); image(z2); axis('image'); axis off
  title('image module de y phase de x')
  colormap('gray')
```