Rapport de Traitement du signal

Kévin Fardel et Rick Ghanem

6 janvier 2011

Résumé

Table des matières

Table des codes sources

Première partie

Exercice 1

```
clear;
clf;

f0 = 1;% Unite de frequence
T0 = 1/f0;% Unite de temps

a = 8;% 1/2 largeur de la porte
Dt = a*T0;
A = 1;%hauteur de la porte
t_min = -32*T0;%borne inferieure de l'intervalle de visualisation
t_max = 32*T0;%borne superieure
t = t_min:1:t_max;%ensemble des valeur de t que l'on va calculer
n = 512;% Nombre de points
s = zeros(1,n);% on initialise l'ensemble des points a zeros
s(512/2-a:512/2+a) = 1;%les valeurs entre -8 et 8 notre porte vaut 1

h = stem(-255:256, s);%on trace tous les points de s entre -255 et 256
xlim([t_min t_max]);%les x sont compris entre t_min et t_max
my_title('Signal porte de largeur 16 et d''amplitude 1');

input('Figure suivante ? ');

[x, f] = TFD(s, 1, 512);%on calcul la transforme de fourier de pour les 512 points
fig = stem(f, abs(x));%on trace la valeur absolu de la transforme de fourier
xlim([0 0.5]);%sur l'intervalle 0 0.5
my_title('Transformee de fourier de la porte sur l''intervalle [0,0.5]') ;
```

Listing 1– Code source pour l'exercice 1

Deuxième partie

Exercice 2

```
clf;
clear;

%Definition de l'intervalle de visualisation
n_min=-4;%borne min de l'intervalle de visualisation
n_max=9;%borne max de l'intervalle de visualisation
n=n_min:1:n_max;
% La reponse impulsionnelle du filtre est la soustraction de deux echellons * 2 * sin(n*
    pi/2)
filtre=(echelon(n,3).-echelon(n,-4)).*2.*sin(n*pi/2);
%on trace la reponse impulsionnelle sur l'intervalle de visualisation
stem(n,filtre);
title('Fonction de transfert du filtre');
axis([n_min,n_max]);
input('Figure suivante ?');
%Le signal d'entre est la soustraction de deux echelons * n/2
signal = (n./2).*(echelon(n,0).-echelon(n,-6));
%on trace le signal
stem(n,signal);
title('Signal discret d''entre');
axis([n_min,n_max]);
input('Figure suivante ?');
```

```
%Signal en sortie de filtre. On utilise le produit de convolution.
y=conv(signal,filtre);
stem(y);
title('Sortie du filtre');
axis([n_min,n_max]);
```

Listing 2– Code source pour l'exercice 2

```
% Fonction qui cree un echelon
% ============================================================
% n = ensemble des points sur lesquels on veut tracer l'echelon
% dec = decalage qu'on souhaite appliquer a l'echelon unite
% E(n) = u(n+dec)
% E(-dec) = u(0) = 1
% E(n) = 0 ssi n < -dec
% E(n) = 1 ssi n >= dec
function [E] = echelon(n,dec)
N = length(n);
E = zeros(1,N);
for i=(1-dec)+abs(min(n)):N
  E(i)=1;
endfor
endfunction
```

Listing 3– Fonction pour créer un echelon

Troisième partie

Exercice 3

Quatrième partie

Exercice 4

```
clear;
clf;

fe=8000;%frequence d'echantillonnage
fcut = 1000;%frequence de coupure
largeur = 200;%largeur de transition
N=1024;%Nombre de point qu'on veut calculer
n0=N/2;

Wp=(2*fcut)/fe;% borne inferieur de la bande passante
Ws=2*(fcut+largeur)/fe;%  borne superieur de la bande passante
[n Wn]=buttord(Wp,Ws,1,40);%calcul l'ordre du filtre Butterworth, ici nous faisons un
    filtre passe bas car Wp<Ws
[B A]= butter(n, Wn);%Genere le filtre butterworth
x = zeros(1,N) ;%on initialise les 1024 point de la courbe x a zeros
x(1) = 1 ;
y=filter(B,A,x);%on applique le filtre genere precedemment a la courbe

%Trace de la reponse impulsionnel
stem (y) ;
xlim([0,150]);
my_title ('Reponse impulsionnelle') ;
```

```
%Pole/zero
input ("Figure suivante ? ") ;
zplane(B, A) ;%on trace les poles et zeros
my_title("Zeros (o) et poles (x)");

% Fonctions de transfert avec freqz
input ('Figure suivante ? ') ;
[H f] = freqz (B,A) ;
plot (f,20*abs(H),'b');
xlim([0,1]);
my_title ('Fonction de transfert') ;

% Somme de deux sinusoides
input ('Figure suivante ? ') ;
Te = 1/fe;%Periode d'echantionnage
fe1=800;%Frequence de la premiere sinusoide
fe2 = 1400;%Frequence de la seconde sinusoide
t=(0:N-1)*Te;
x1=sin(2*pi*fe1*t);% Definition de la premiere sinusoide
x2=sin(2*pi*fe2*t);% Definition de la seconde sinusoide
X=x1.+x2;%On ajoute chaque valeur de chaque sinusoide une a une
plot(X);% On trace la somme des deux sinusoide
xlim([0,1000]);
my_title ('Somme de deux sinusoides') ;

%Spectre du signal
input ('Figure suivante ? ') ;
fX=fft(X);%Calcul du spectre du signal via la tranforme rapide du signal
plot(abs(fX));
xlim([0,1100]);
my_title ('Spectre du signal') ;

%Spectre du signal filtre
input ('Figure suivante ? ') ;
FX=filter(B,A,X);%On applique le filtre au  signal
fFX = fft(FX);%on recupere son spectre
plot(abs(fFX));
xlim([0,1100]);
my_title ('Spectre du signal filtre') ;
```

Listing 4– Code source pour l'exercice 4

Cinquième partie

Exercice 5

Sixième partie

Exercice 6

```
clear
clf

load "TD_ESIL.mat";% Dans le fichier matLab on recupere la var fe, et les tableaux A,B,x,
    y

%Recuperation de la longueure des tableau x,y
x_max = length(x);%taille de x
```

```
y_max = length(y);

%Allure temporelle de X
h = stem (1:1:x_max, x);% affichage de l'allure temporelle de x
xlim([0 x_max]);%intervalle de visualisation
set_ylim(x) ;
my_title ('Allure temporelle de x') ;

input("Figure suivante ?");
%Spectre de X
spectreX = fft(x) ;%on recupere le spectre de l'echantillon x par la transformee rapide de fourrier
h = stem (1:1:x_max, abs(spectreX));%on trace
xlim([0 x_max]);%intervalle de visualisation
set_ylim(abs(spectreX)) ;
my_title ('Spectre de x') ;

input("Figure suivante ?");
%Allure temporelle de Y // on fait pareil qu'avec y
h = stem (1:1:y_max, y);% affichage de l'allure temporelle de y
xlim([0 y_max]);
set_ylim(y) ;
my_title ('Allure temporelle de y') ;

input("Figure suivante ?");
%Spectre de Y // On fait pareil qu'avec x
spectreY = fft(y) ;
h = stem (1:1:y_max, abs(spectreY));  % fft de y
xlim([0 y_max]);
set_ylim(abs(spectreY)) ;
my_title ('Spectre de y') ;

input("Afficher caracteristique de x ?");
% Caracteristique du signal x
moye = mean(x) ;%calcul de la moyenne du signal
ecarT = std(x) ;%calcul de l'ecart type
vari = var(x) ;%calcul de la variance
printf ('Moyenne = %.2f\n',moye) ;
printf ('Ecart type = %.2f\n',ecarT) ;
printf ('Variance = %.2f\n',vari) ;

input ('Figure suivante ? ') ;
% Densite spectrale de puissance
psd = spectral_xdf (x, "rectangle", 1/sqrt(x_max)) ;%on recupere la densite spectrale par la fonction spectral_xdf du paquet signal
[psdX fX] = psd_shift (psd, fe) ;%la densite spectrale ete definie sur 0 1 on la definie sur le fe qu'on a recupere dans le fihcier matlab charge
plot(fX, abs(psdX)) ;
hold on
plot([min(fX) max(fX)], [1/fe 1/fe]) ;
my_title ('Densite spectrale du signal x') ;
set_ylim(psdX) ;
xlim([min(fX) max(fX)]);
hold off

input ('Figure suivante ? ');
% Densite spectrale de puissance
psd = spectral_xdf (y, "rectangle", 1/sqrt(y_max)) ;%on recupere la densite spectrale de y dans une fenetre rectangle
[psdY fY] = psd_shift (psd, fe) ;%la densite spectrale ete definie sur 0 1 on la definie sur le fe qu'on a recupere dans le fihcier matlab charge
plot(fY, abs(psdY)) ;
hold on
```

```
66 plot([min(fY) max(fY)], [1/fe 1/fe]) ;
   my_title ('Densite spectrale du signal y') ;
68 set_ylim(abs(psdY)) ;
   xlim([min(fY) max(fY)]);
70 hold off

72 input ('Figure suivante ? ');
   %Allure de fonction de transfert en harmonique compare a la fonction de tranfert definie
       par A et B
74 clf
   tXY = sqrt(psdY ./ psdX);%fonction de transfert en harmonque du filtre : densite spectral
       en sortie / densite spectrale en entree
76 [tAB f]= freqz(B, A);% fonction de transfert definie par A et B
   hold on
78 plot(fX, tXY, C="g");
   plot((f/(2*pi))*fe, abs(tAB));
80 legend('Fonction de transfert harmonique du filtre', 'Fonction de transfert definie par A
       et B', 'location', 'west');
   xlim([0 500]);
82 set_ylim(tXY) ;
   my_title ('Fonctions de transfert') ;
84 hold off
```

Listing 5– Code source pour l'exercice 6