

Rapport de Traitement du signal

Kévin Fardel et Rick Ghanem

5 janvier 2011

Résumé

Table des matières

I	Exercice 1	2
II	Exercice 2	2
III	Exercice 3	2
IV	Exercice 4	2
V	Exercice 5	3
VI	Exercice 6	4

Table des codes sources

1	Code source pour l'exercice 1	2
2	Code source pour l'exercice 4	2
3	Code source pour l'exercice 6	4

Première partie

Exercice 1

```
1 clear;
  clf;
3
  f0 = 1;% Unite de frequence
5 T0 = 1/f0;% Unite de temps

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

17 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
19 my_title('Signal porte de largeur 16 et d'amplitude 1');

21 input('Figure suivante ? ');

23 [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
25 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

Troisième partie

Exercice 3

Quatrième partie

Exercice 4

```
1 clear;
  clf;
3
  fe=8000;%frequence d'echantillonnage
5 fcut = 1000;%frequence de coupure
  largeur = 200;%largeur de transition
7 N=1024;%Nombre de point qu'on veut calculer
  n0=N/2;
9
  Wp=(2*fcut)/fe;% borne inferieur de la bande passante
```

```

11 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
13 [B A]= butter(n, Wn);%Genere le filtre butterworth
   x = zeros(1,N) ;%on initialise les 1024 point de la courbe x a zeros
15 x(1) = 1 ;
   y=filter(B,A,x);%on applique le filtre genere precedemment a la courbe
17
   %Trace de la reponse impulsionnel
19 stem (y) ;
   xlim([0,150]);
21 my_title ('Reponse impulsionnelle') ;

23 %Pole/zero
   input ("Figure suivante ? ") ;
25 zplane(B, A) ;%on trace les poles et zeros
   my_title("Zeros (o) et poles (x)");
27
   % Fonctions de transfert avec freqz
29 input ('Figure suivante ? ') ;
   [H f] = freqz (B,A) ;
31 plot (f,20*abs(H), 'b');
   xlim([0,1]);
33 my_title ('Fonction de transfert') ;

35 % Somme de deux sinusoides
   input ('Figure suivante ? ') ;
37 Te = 1/fe;%Periode d'echantionnage
   fe1=800;%Frequence de la premiere sinusoide
39 fe2 = 1400;%Frequence de la seconde sinusoide
   t=(0:N-1)*Te;
41 x1=sin(2*pi*fe1*t);% Definition de la premiere sinusoide
   x2=sin(2*pi*fe2*t);% Definition de la seconde sinusoide
43 X=x1.+x2;%On ajoute chaque valeur de chaque sinusoide une a une
   plot(X);% On trace la somme des deux sinusoide
45 xlim([0,1000]);
   my_title ('Somme de deux sinusoides') ;
47
   %Spectre du signal
49 input ('Figure suivante ? ') ;
   fX=fft(X);%Calcul du spectre du signal via la tranforme rapide du signal
51 plot(abs(fX));
   xlim([0,1100]);
53 my_title ('Spectre du signal') ;

55 %Spectre du signal filtre
   input ('Figure suivante ? ') ;
57 FX=filter(B,A,X);%On applique le filtre au signal
   fFX = fft(FX);%on recupere son spectre
59 plot(abs(fFX));
   xlim([0,1100]);
61 my_title ('Spectre du signal filtre') ;

```

Listing 2– Code source pour l'exercice 4

Cinquième partie

Exercice 5

Sixième partie

Exercice 6

```
1 clear
2 clf
4 load "TD_ESIL.mat"; % Dans le fichier matLab on recupere la var fe, et les tableaux A,B,x,
   y
6 %Recuperation de la longueur des tableau x,y
x_max = length(x); %taille de x
8 y_max = length(y);
10 %Allure temporelle de X
h = stem (1:1:x_max, x); %affichage de l'allure temporelle de x
12 xlim([0 x_max]); %intervalle de visualisation
set_ylim(x) ;
14 my_title ('Allure temporelle de x') ;
16 input("Figure suivante ?");
%Spectre de X
18 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
20 xlim([0 x_max]); %intervalle de visualisation
set_ylim(abs(spectreX)) ;
22 my_title ('Spectre de x') ;
24 input("Figure suivante ?");
%Allure temporelle de Y // on fait pareil qu'avec y
26 h = stem (1:1:y_max, y); %affichage de l'allure temporelle de y
xlim([0 y_max]);
28 set_ylim(y) ;
my_title ('Allure temporelle de y') ;
30
32 input("Figure suivante ?");
%Spectre de Y // On fait pareil qu'avec x
spectreY = fft(y) ;
34 h = stem (1:1:y_max, abs(spectreY)); % fft de y
xlim([0 y_max]);
36 set_ylim(abs(spectreY)) ;
my_title ('Spectre de y') ;
38
39 input("Afficher caracteristique de x ?");
40 % Caracteristique du signal x
moye = mean(x) ; %calcul de la moyenne du signal
42 ecarT = std(x) ; %calcul de l'ecart type
vari = var(x) ; %calcul de la variance
44 printf ('Moyenne = %.2f\n',moye) ;
printf ('Ecart type = %.2f\n',ecarT) ;
46 printf ('Variance = %.2f\n',vari) ;
48 input ('Figure suivante ? ') ;
% Densite spectrale de puissance
```

```

50 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 fichier matlab charge
52 plot(fX, abs(psdX)) ;
    hold on
54 plot([min(fX) max(fX)], [1/fe 1/fe]) ;
    my_title ('Densite spectrale du signal x') ;
56 set_ylim(psdX) ;
    xlim([min(fX) max(fX)]);
58 hold off

60 input ('Figure suivante ? ');
    % Densite spectrale de puissance
62 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 fichier matlab charge
64 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 harmonique 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 3– Code source pour l'exercice 6