

Rapport de Traitement du signal

Kévin Fardel et Rick Ghanem

6 janvier 2011

Résumé

Table des matières

I	Exercice 1	3
II	Exercice 2	3
III	Exercice 3	4
1	Code source	4
1.1	Fonction $\frac{1-z^{-1}}{2}$	4
1.2	Fonction $\frac{1+z^{-1}}{2}$	5
1.3	Fonction $\frac{1-z^{-2}}{2}$	6
1.4	Fonction $\frac{2z^{-1}}{2-z^{-1}}$	7
1.5	Fonction $\frac{2z^{-1}-z^{-5}}{2-z^{-1}}$	8
2	Résultats obtenus	9
2.1	Fonction $\frac{1-z^{-1}}{2}$	9
IV	Exercice 4	10
V	Exercice 5	11
VI	Exercice 6	11

Table des codes sources

1	Code source pour l'exercice 1	3
2	Code source pour l'exercice 2	3
3	Fonction pour créer un echelon	4
4	Code source pour l'exercice 3 fonction $\frac{1-z^{-1}}{2}$	4
5	Code source pour l'exercice 3 fonction $\frac{1+z^{-1}}{2}$	5

6	Code source pour l'exercice 3 fonction $\frac{1-z^{-2}}{2}$	6
7	Code source pour l'exercice 3 fonction $\frac{2z^{-1}}{2-z^{-1}}$	7
8	Code source pour l'exercice 3 fonction $\frac{2z^{-1}-z^{-5}}{2-z^{-1}}$	8
9	Code source pour l'exercice 4	10
10	Code source pour l'exercice 6	11

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',25);
  %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx1/signalPorte.jpg"
21
  input('Figure suivante ? ');
23
  [x, f] = TFD(s, 1, 512);%on calcul la transforme de fourier de pour les 512 points
25 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
27 my_title('Transformee de fourier de la porte sur l''intervalle [0,0.5]', 25) ;
  %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx1/tfd.jpg"

```

Listing 1– Code source pour l'exercice 1

Deuxième partie

Exercice 2

```

1  clf;
2  clear;

4  %Definition de l'intervalle de visualisation
  n_min=-4;%borne min de l'intervalle de visualisation
6  n_max=9;%borne max de l'intervalle de visualisation
  n=n_min:1:n_max;
8  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);
10 %on trace la reponse impulsionnelle sur l'intervalle de visualisation
  stem(n, filtre);
12 my_title('Fonction de transfert du filtre',25);
  axis([n_min,n_max]);
14 %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx2/fctTransfert.jpg"

16 input('Figure suivante ? ');
  %Le signal d'entre est la soustraction de deux echelons * n/2
18 signal = (n./2).*(echelon(n,0).-echelon(n,-6));
  %on trace le signal

```

```

20 stem(n, signal);
   my_title('Signal discret d''entre',25);
22 axis([n_min,n_max]);
   %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx2/signalEntree.jpg"
24
   input('Figure suivante ?');
26 %Reponse impulsionnelle
   %Pour afficher la reponse impulsionnelle du filtre nous faisons un produit de convolution
     entre un dirac et le filtre
28 dirac=zeros(1,length(n));
   dirac(abs(n_min)+1)=1;
30 y=conv(dirac, filtre);
   stem(y);
32 my_title('Reponse impulsionnelle du filtre',25);
   axis([n_min,n_max]);
34 %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx2/repImpulsionnelle.jpg"
36
   input('Figure suivante ?');
   %Signal en sortie de filtre. On utilise le produit de convolution.
38 y=conv(signal, filtre);
   stem(y);
40 my_title('Sortie du filtre',25);
   axis([n_min,n_max]);
42 %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx2/sortieFiltre.jpg"

```

Listing 2– Code source pour l'exercice 2

```

% Fonction qui cree un echelon
2 % =====
% n = ensemble des points sur lesquels on veut tracer l'echelon
4 % dec = decalage qu'on souhaite appliquer a l'echelon unite
% E(n) = u(n+dec)
6 % E(-dec) = u(0) = 1
% E(n) = 0 ssi n < -dec
8 % E(n) = 1 ssi n >= dec
function [E] = echelon(n,dec)
10 N = length(n);
   E = zeros(1,N);
12 for i=(1-dec)+abs(min(n)):N
   E(i)=1;
14 endfor
endfunction

```

Listing 3– Fonction pour créer un echelon

Troisième partie

Exercice 3

1 Code source

1.1 Fonction $\frac{1-z^{-1}}{2}$

```

1 clear
   clf
3 %Coefficient du filtre etudie
   A = [2 0] ;
5 B = [1 -1] ;

```

```

7 fe = 1 ; % Frequence d'echantillonnage
[H, w] = freqz(B,A) ; % w : pulsation entre 0 et pi
9 nu = w/(2*pi) ; % Frequence reduite
f = nu*fe ; % Frequence

11 %On trace le diagramme de gain pour determiner la nature du filtre
13 plot(nu,20*log10(abs(H))) ;
my_title('Diagramme de gain en frequence reduite en db',25) ;
15 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f1Gain.jpg"

17 input('Figure suivante ?') ;
%Diagramme de phase
19 plot(nu,angle(H)) ;
my_title('Diagramme de phase en frequence reduite',25) ;
21 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f1Phase.jpg"

23 input('Figure suivante ?') ;
%Zeros et poles
25 zplane(B, A) ;
my_title('Zeros et poles',25) ;
27 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f1ZP.jpg"

29 %Reponse impulsionnelle
%Definition de l'intervalle de visualisation
31 N = 64 ;
n0 = N/2 ;
33 n_min = 1-n0 ;
n_max = N-n0 ;
35 n = n_min:1:n_max ;

37 %Definition du dirac
dirac = zeros(1,N) ;
39 dirac(n0) = 1 ;
y = filter(B,A,dirac) ;
41 input('Figure suivante ?') ;
stem(n,y) ;
43 my_title('Reponse impulsionnelle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f1Impulsion.jpg"

45 input('Figure suivante ?') ;
47 %Reponse indicielle
%Definition de l'echelon
49 u = zeros(1,N) ;
for i=n0:N
51 u(i)=1;
endfor
53 i = filter(B,A,u) ;
h_obj = stem(n,i) ;
55 my_title('Reponse indicielle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f1Indice.jpg"

```

Listing 4– Code source pour l'exercice 3 fonction $\frac{1-z^{-1}}{2}$

1.2 Fonction $\frac{1+z^{-1}}{2}$

```

1 clear
clf
3 %Coefficient du filtre etudie
A = [2 0] ;
5 B = [1 1] ;

```

```

7 fe = 1 ; % Frequence d'echantillonnage
[H, w] = freqz(B,A) ; % w : pulsation entre 0 et pi
9 nu = w/(2*pi) ; % Frequence reduite
f = nu*fe ; % Frequence

11 %On trace le diagramme de gain pour determiner la nature du filtre
13 plot(nu,20*log10(abs(H))) ;
my_title('Diagramme de gain en frequence reduite en db',25) ;
15 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f2Gain.jpg"

17 input('Figure suivante ?') ;
19 %Diagramme de phase
plot(nu,angle(H)) ;
21 my_title('Diagramme de phase en frequence reduite',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f2Phase.jpg"

23 input('Figure suivante ?') ;
25 %Zeros et poles
zplane(B, A) ;
27 my_title('Zeros et poles',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f2ZP.jpg"

29 %Reponse impulsionnelle
31 %Definition de l'intervalle de visualisation
N = 64 ;
33 n0 = N/2 ;
n_min = 1-n0 ;
35 n_max = N-n0 ;
n = n_min:1:n_max ;

37 %Definition du dirac
39 dirac = zeros(1,N);
dirac(n0) = 1;
41 y = filter(B,A,dirac);
input('Figure suivante ?') ;
43 stem(n,y) ;
my_title('Reponse impulsionnelle',25) ;
45 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f2Impulsion.jpg"

47 input('Figure suivante ?') ;
%Reponse indicielle
49 %Definition de l'echelon
u = zeros(1,N) ;
51 for i=n0:N
u(i)=1;
53 endfor
i = filter(B,A,u) ;
55 h_obj = stem(n,i) ;
my_title('Reponse indicielle',25) ;
57 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f2Indice.jpg"

```

Listing 5– Code source pour l'exercice 3 fonction $\frac{1+z^{-1}}{2}$

1.3 Fonction $\frac{1-z^{-2}}{2}$

```

clear
2 clf
%Coefficient du filtre etudie
4 A = [2 0 0] ;
B = [1 0 -1] ;

```

```

6
fe = 1 ; % Frequence d'echantillonnage
8 [H, w] = freqz(B,A) ; % w : pulsation entre 0 et pi
nu = w/(2*pi) ; % Frequence reduite
10 f = nu*fe ; % Frequence

12 %On trace le diagramme de gain pour determiner la nature du filtre
plot(nu,20*log10(abs(H))) ;
14 my_title('Diagramme de gain en frequence reduite en db',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f3Gain.jpg"

16
input('Figure suivante ?') ;
18 %Diagramme de phase
plot(nu,angle(H)) ;
20 my_title('Diagramme de phase en frequence reduite',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f3Phase.jpg"

22
input('Figure suivante ? ') ;
24 %Zeros et poles
zplane(B, A) ;
26 my_title('Zeros et poles',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f3ZP.jpg"

28
%Reponse impulsionnelle
30 %Definition de l'intervalle de visualisation
N = 64 ;
32 n0 = N/2 ;
n_min = 1-n0 ;
34 n_max = N-n0 ;
n = n_min:1:n_max ;

36
%Definition du dirac
38 dirac = zeros(1,N);
dirac(n0) = 1;
40 y = filter(B,A,dirac);
input('Figure suivante ?') ;
42 stem(n,y) ;
my_title('Reponse impulsionnelle',25) ;
44 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f3Impulsion.jpg"

46
input('Figure suivante ?') ;
%Reponse indicielle
48 %Definition de l'echelon
u = zeros(1,N) ;
50 for i=n0:N
    u(i)=1;
52 endfor
i = filter(B,A,u) ;
54 h_obj = stem(n,i) ;
my_title('Reponse indicielle',25) ;
56 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f3Indice.jpg"

```

Listing 6– Code source pour l'exercice 3 fonction $\frac{1-z^{-2}}{2}$

1.4 Fonction $\frac{2z^{-1}}{2-z^{-1}}$

```

1 clear
clf
3 %Coefficient du filtre etudie
A = [2 -1] ;
5 B = [0 2] ;

```

```

7 fe = 1 ; % Frequence d'echantillonnage
[H, w] = freqz(B,A) ; % w : pulsation entre 0 et pi
9 nu = w/(2*pi) ; % Frequence reduite
f = nu*fe ; % Frequence

11 %On trace le diagramme de gain pour determiner la nature du filtre
13 plot(nu,20*log10(abs(H))) ;
my_title('Diagramme de gain en frequence reduite en db',25) ;
15 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f4Gain.jpg"

17 input ('Figure suivante ?') ;
%Diagramme de phase
19 plot(nu,angle(H)) ;
my_title('Diagramme de phase en frequence reduite',25) ;
21 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f4Phase.jpg"

23 input ('Figure suivante ? ') ;
%Zeros et poles
25 zplane(B, A) ;
my_title('Zeros et poles',25) ;
27 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f4ZP.jpg"

29 %Reponse impulsionnelle
%Definition de l'intervalle de visualisation
31 N = 64 ;
n0 = N/2 ;
33 n_min = 1-n0 ;
n_max = N-n0 ;
35 n = n_min:1:n_max ;

37 %Definition du dirac
dirac = zeros(1,N);
39 dirac(n0) = 1;
y = filter(B,A,dirac);
41 input ('Figure suivante ?') ;
stem (n,y) ;
43 my_title('Reponse impulsionnelle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f4Impulsion.jpg"

45 input ('Figure suivante ?') ;
47 %Reponse indicielle
%Definition de l'echelon
49 u = zeros(1,N) ;
for i=n0:N
51 u(i)=1;
endfor
53 i = filter(B,A,u) ;
h_obj = stem (n,i) ;
55 my_title('Reponse indicielle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f4Indice.jpg"

```

Listing 7– Code source pour l'exercice 3 fonction $\frac{2z^{-1}}{2-z^{-1}}$

1.5 Fonction $\frac{2z^{-1}-z^{-5}}{2-z^{-1}}$

```

1 clear
clf
3 %Coefficient du filtre etudie
A = [2 -1 0 0 0 0] ;
5 B = [0 2 0 0 0 -1] ;

```



```

7 fe = 1 ; % Frequence d'echantillonnage
[H, w] = freqz(B,A) ; % w : pulsation entre 0 et pi
9 nu = w/(2*pi) ; % Frequence reduite
f = nu*fe ; % Frequence

11 %On trace le diagramme de gain pour determiner la nature du filtre
13 plot(nu,20*log10(abs(H))) ;
my_title('Diagramme de gain en frequence reduite en db',25) ;
15 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f5Gain.jpg"

17 input ('Figure suivante ?') ;
%Diagramme de phase
19 plot(nu,angle(H)) ;
my_title('Diagramme de phase en frequence reduite',25) ;
21 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f5Phase.jpg"

23 input ('Figure suivante ? ') ;
%Zeros et poles
25 zplane(B, A) ;
my_title('Zeros et poles',25) ;
27 %print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f5ZP.jpg"

29 %Reponse impulsionnelle
%Definition de l'intervalle de visualisation
31 N = 64 ;
n0 = N/2 ;
33 n_min = 1-n0 ;
n_max = N-n0 ;
35 n = n_min:1:n_max ;

37 %Definition du dirac
dirac = zeros(1,N);
39 dirac(n0) = 1;
y = filter(B,A,dirac);
41 input ('Figure suivante ?') ;
stem (n,y) ;
43 my_title('Reponse impulsionnelle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f5Impulsion.jpg"

45 input ('Figure suivante ?') ;
%Reponse indicielle
%Definition de l'echelon
49 u = zeros(1,N) ;
for i=n0:N
51 u(i)=1;
endfor
53 i = filter(B,A,u) ;
h_obj = stem (n,i) ;
55 my_title('Reponse indicielle',25) ;
%print -deps "/home/kewin/octave/td_matLab/rapport/src/resEx3/f5Indice.jpg"

```

Listing 8– Code source pour l'exercice 3 fonction $\frac{2z^{-1}-z^{-5}}{2-z^{-1}}$

2 Résultats obtenus

2.1 Fonction $\frac{1-z^{-1}}{2}$

Nous commençons par afficher le diagramme de gain en décibel ()ainsi que le diagramme de phase en radians () pour déterminer la nature du filtre représenté par cette fonction de transfert. salut salut

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',25) ;
   %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx4/repImpulsion.jpg"
23
   %Pole/zero
25  input ("Figure suivante ? ") ;
   zplane(B, A) ;%on trace les poles et zeros
27  my_title('Zeros (o) et poles (x)',25);
   %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx1/tfd.jpg"
29
   % Fonctions de transfert avec freqz
31  input ('Figure suivante ? ') ;
   [H f] = freqz (B,A) ;
33  plot (f,20*abs(H), 'b');
   xlim([0,1]);
35  my_title ('Fonction de transfert', 25) ;
   %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx4/fctTransfert.jpg"
37
   % Signal constitue par la somme de deux sinusoides
39  input ('Figure suivante ? ') ;
   Te = 1/fe;%Periode d'echantionnelle
41  fe1=800;%Frequence de la premiere sinusoide
   fe2 = 1400;%Frequence de la seconde sinusoide
43  t=(0:N-1)*Te;
   x1=sin(2*pi*fe1*t);% Definition de la premiere sinusoide
45  x2=sin(2*pi*fe2*t);% Definition de la seconde sinusoide
   X=x1.+x2;%On ajoute chaque valeur de chaque sinusoide une a une
47  plot(X);% On trace la somme des deux sinusoide
   xlim([0,1000]);
49  my_title ('Somme de deux sinusoides') ;
   %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx4/2sin.jpg"
51
   %Spectre du signal
53  input ('Figure suivante ? ') ;
   fX=fft(X);%Calcul du spectre du signal via la tranforme rapide du signal
55  plot(abs(fX));
   xlim([0,1100]);
57  my_title ('Spectre du signal',25) ;

```

```

59 %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx4/spectreEntre.jpg"
%Spectre du signal filtre
61 input ( 'Figure suivante ? ' ) ;
FX=filter(B,A,X);%On applique le filtre au signal
63 fFX = fft(FX);%on recupere son spectre
plot(abs(fFX));
65 xlim([0,1100]);
my_title ( 'Spectre du signal filtre',25) ;
67 %print -deps "/home/kewin/octave/td_matLab/rapport/fig/resEx4/spectreSortie.jpg"

```

Listing 9– Code source pour l'exercice 4

Cinquième partie

Exercice 5

Sixième partie

Exercice 6

```

1 clear
  clf
3 load "TD_ESIL.mat";% Dans le fichier matLab on recupere la var fe, et les tableaux A,B,x,
  y
5 %Recuperation de la longueur des tableau x,y
7 x_max = length(x);%taille de x
  y_max = length(y);
9 %Allure temporelle de X
11 h = stem (1:1:x_max, x);% affichage de l'allure temporelle de x
  xlim([0 x_max]);%intervalle de visualisation
13 set_ylim(x) ;
  my_title ( 'Allure temporelle de x' ) ;
15 input("Figure suivante ?");
17 %Spectre de X
  spectreX = fft(x) ;%on recupere le spectre de l'echantillon x par la transformee rapide
    de fourrier
19 h = stem (1:1:x_max, abs(spectreX));%on trace
  xlim([0 x_max]);%intervalle de visualisation
21 set_ylim(abs(spectreX)) ;
  my_title ( 'Spectre de x' ) ;
23 input("Figure suivante ?");
25 %Allure temporelle de Y // on fait pareil qu'avec y
  h = stem (1:1:y_max, y);% affichage de l'allure temporelle de y
27 xlim([0 y_max]);
  set_ylim(y) ;
29 my_title ( 'Allure temporelle de y' ) ;

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

```

```

37 set_ylim(abs(spectreY)) ;
my_title ('Spectre de y') ;

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

47 input ('Figure suivante ? ') ;
49 % 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
51 [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
plot(fX, abs(psdX)) ;
53 hold on
plot([min(fX) max(fX)], [1/fe 1/fe]) ;
55 my_title ('Densite spectrale du signal x') ;
set_ylim(psdX) ;
57 xlim([min(fX) max(fX)]);
hold off

59 input ('Figure suivante ? ');
61 % Densite spectrale de puissance
psd = spectral_xdf (y, "rectangle", 1/sqrt(y_max)) ;%on recupere la densite spectrale de
    y dans une fenetre rectangle
63 [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
plot(fY, abs(psdY)) ;
65 hold on
plot([min(fY) max(fY)], [1/fe 1/fe]) ;
67 my_title ('Densite spectrale du signal y') ;
set_ylim(abs(psdY)) ;
69 xlim([min(fY) max(fY)]);
hold off

71 input ('Figure suivante ? ');
73 %Allure de fonction de transfert en harmonique compare a la fonction de tranfert definie
    par A et B
clf
75 tXY = sqrt(psdY ./ psdX);%fonction de transfert en harmonique du filtre : densite spectral
    en sortie / densite spectrale en entree
[tAB f]= freqz(B, A);% fonction de transfert definie par A et B
77 hold on
plot(fX, tXY, C="g");
79 plot((f/(2*pi))*fe, abs(tAB));
legend('Fonction de transfert harmonique du filtre', 'Fonction de transfert definie par A
    et B', 'location', 'west');
81 xlim([0 500]);
set_ylim(tXY) ;
83 my_title ('Fonctions de transfert') ;
hold off

```

Listing 10– Code source pour l'exercice 6