

Calcul des coefficients d'un filtre R.I.F.

Le programme matlab "**filtreEX.m**" fournit ci-dessous:

- calcule les coefficients d'un filtre passe bas RIF,
- affiche graphiquement les coefficients et les diagrammes de Bode du filtre,
- ajoute un signal de fréquence élevée à un fichier "wav" pour tester le filtre passe-bas réalisé,
- et sauvegarde les coefficients calculés en virgule fixe (au format Q15) en hexadécimal dans un fichier (Utile pour faire un copier-coller dans un programme pour DSP dans le TP suivant).

a) Observer et commenter les résultats

- lorsqu'on diminue le nombre de coefficients (M...)
- lorsqu'on change la fenêtre de pondération (tester entre autre Kaiser...)

b) Modifier le fichier pour obtenir les coefficients d'un passe-haut. Vérifier et tester l'efficacité du filtre sur un enregistrement sonore .

c) Ecrire le fichier pour calculer les coefficients d'un filtre réjecteur destiné à éliminer un signal parasite de fréquence 750Hz (coupe-bande centré sur 750Hz de largeur à choisir).

- Tester le filtre sur les fichiers .wav parasités *Gershwin_750Hz.wav* et *Bizet_750Hz.wav*
- Choisir M, les fréquences de coupures f1, f2 et une fenêtre pour que le 200Hz ne soit plus audible.

Remarque : Choisir M le plus petit possible pour réduire les temps de calculs (entre 1 et 500)

```

%-----
%                               PASSE BAS RIF
%                               Calcul des M+1 coefficients a0 a1 a2 ...aM d'un filtres RIF passe-bas
%                               Master SET
%-----
% Formules de quelques fenêtres classiques intéressantes à tester
%(0.54*ones(1,M+1)+(1-0.54)*cos(2*pi*(k-M/2)/(M+1)));           %Hamming
%(0.5*ones(1,M+1)+0.5*cos(2*pi*(k-M/2)/(M+1)));                 %Hanning
%(0.42*ones(1,M+1)-0.5*cos(2*pi*k/(M+1))+0.08*cos(4*pi*k/(M+1))); %blackman
%(0.42323*ones(1,M+1)-0.49755*cos(2*pi*k/(M+1))+0.07922*cos(4*pi*k/(M+1))); %blackman Harris
%(7938/18608*ones(1,M+1)-9240/18608*cos(2*pi*k/(M+1))+1430/18608*cos(4*pi*k/(M+1))); %blackman exacte
%(0.2810639*ones(1,M+1)-0.5208972*cos(2*pi*k/(M+1))+0.1980399*cos(4*pi*k/(M+1))); %toit plat
%(besseli(0,9*sqrt(1-(4*(k-M/2).*(k-M/2))/((M)^2)))/besseli(0,9)); %kaiser
% exp(k/(M+1));                                                  %exponentielle
% exp(-1*(k/(M+1)).^2);                                          %gauss
%-----

close all;
clear all;
hold off;

fc=4000; % f coupure du filtre
fe=11025; % f echantillonnage
M=100; % M+1 coefficients avec M pair

%Calcul des coefficients a -----
k=[0:M];
a=2*fc/fe*sinc(2*fc/fe*(k-M/2)); % sinus cardinal: sinc(x) = sin(pi*x)/pi*x
a=a.*(0.54*ones(1,M+1)+(1-0.54)*cos(2*pi*(k-M/2)/(M+1))); %Hamming

%Contrôle des performances -----
[H,freq]=freqz(a,1,1000,11025); % Calcul de la transmittance H avec les coeff a ci-dessus

%Affichage graphique -----
figure(1)
subplot(2,2,1);
plot(k,a,'+-'); % Affichage graphique des coefficients ak

```

```

subplot(2,2,2);
plot(freq,abs(H)); % module de |H|
subplot(2,2,3);
plot(freq,20*log10(abs(H))); % Bode 20log(|H|)
subplot(2,2,4);
plot(freq,unwrap(angle(H))); % Bode Phase

%Test sur un fichier sonore-----
[y1,fe,bits] = wavread('guitare.wav'); % y1 reçoit les échantillons sonore
y1=y1';
siz=length(y1);
Te=1/fe;
n=[0:1:(siz-1)];

y2=sin((2*pi*5000/fe)*n); %fabrication d'un parasite 5000Hz (sifflement)

y2=y1+y2; %y2 (musique+sifflement)

sys=filt(a,1); % synthèse du filtre par une fonction matlab
y3=lsim(sys,y2,n); %y3 = signal y2 filtré par une fonction matlab

%Affichage des signaux y1(original), y2(parasité) et y3(après filtrage) en fonction du temps-----
figure(2);
subplot(3,1,1);
plot(n,y1);
subplot(3,1,2);
plot(n,y2);
subplot(3,1,3);
plot(n,y3);

%Pour écouter les sigaux y1, y2, et y3 appuyer sur une touche après chaque morceau-----
soundsc(y1,fe);
pause;
soundsc(y2,fe);
pause;
soundsc(y3,fe);

%MISE au format Q15: les coeff sont multipliés par 2^15-----
%Sauvegarde en hexadécimal dans le fichier ASCII PasseBasbxx_xxxx.txt pour programme DSP

coeff= hex16(round(a*32768)); % hex16.m est dans le fichier filtre.zip
nom=sprintf('PasseBas%d_%d.txt',length(a),fc); % fabrication du nom de fichier
fid = fopen(nom,'w'); %ouverture du fichier en écriture
for i=1:length(a)
    fprintf(fid,'0x%s, ',coeff(i,:)); %écriture du fichier
end
fclose(fid); %fermeture du fichier

```