Travaux Pratique
TR-C1 : Traitement du signal Avancée

TP 3 : Filtrage

Objectifs :

Le but de ce TP est:

- De synthétiser un filtre numérique sous Matlab
- De filtrer une séquence audio.

Manipulations sous SPTools : Utilitaire sous Matlab pour la génération de filtre

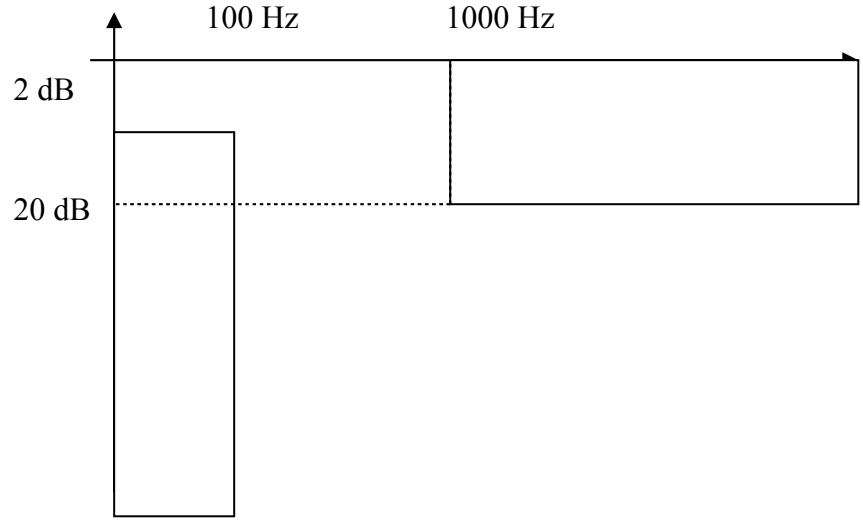
I) Synthèse du filtre

Sous la fenêtre de commande matlab, taper *sptool*. Il s'agit d'une commande matlab qui permet d'exécuter un utilitaire spécialisé dans le filtrage de signaux.
De la fenêtre de dialogue Sptool vous allez pouvoir importer et ou exporter des données depuis/vers des fichiers ou la fenêtre de commande matlab. Il est possible de visualiser immédiatement les effets du filtre, visualiser le gabarit et la réponse du filtre synthétisé.

1. Dans la fenêtre SPTool, cliquez sur new design. Une fenêtre Filter Designer s'ouvre.
   Saisissez le type de filtre, puis le gabarit (Fp, Rp, Fs, Rs, Sampling Frequency) et cliquez sur apply.

   - Fp bande passante.
   - Fs bande coupée.
   - Rp atténuation maximale (en dB) dans la bande passante.
   - Rs atténuation minimale (en dB) dans la bande coupée.

On souhaite synthétiser un filtre dont le gabarit est le suivant :



Nous choisirons d'abord un filtre « elliptic IIR » avec une fréquence d'échantillonnage de 8192 Hz.

Sélectionnez les bons paramètres, et cliquez sur apply puis sur File/close. Le filtre synthétisé apparaît sous le nom filt1, donnez lui un nom par Edit/Name.

2. Construisez un nouveau filtre de Butterworth de fréquence de coupure 300 Hz et d'ordre 4.

II) Visualisation du filtre

3. Dans la fenêtre SPTool, choisissez un filtre puis cliquer sur 'View'. Une fenêtre Filter Viewer s'ouvre.
4. Dans la fenêtre Filter Viewer, sélectionnez les réponses que vous voulez observez. A titre d'exemple, nous choisirons la réponse impulsionnelle. Que pouvez vous dire de la réponse du filtre.
5. Trouver les paramètres du filtre.

III) Visualisation des signaux.

6. Synthétiser un signal sin et sincard sur une durée de 2 ms et de fréquence 500 Hz que vous sauvegarderez sous les noms sinus et sinus_card
7. Récupérer l'extrait 'Toons.wav' et ré-echantillonner le à 8192 Hz à partir du logiciel GoldWave.
8. Dans la fenêtre SPTool, importez ces différents signaux sous File/Import. Choisissez en un ou plusieurs, puis cliquez sur view. Vous devez retrouver l'enveloppe du signal. Vous pouvez aussi jouer la séquence sonore

IV) Filtrage

9. Avec Filter Designer générer maintenant un filtre passe bande correspondant à la ligne téléphonique (gain de -3 dB entre 300 Hz – 3400 Hz et gain de -40 dB à 250 Hz et 3450 Hz).
10. Indiquez le type de filtre utilisé et son ordre.

Dans la fenêtre SPTool, sélectionnez un signal et un filtre puis cliquez sur apply.
Avec le signal Browser, observer le signal avant et après filtrage.
Réduisez la bande du filtre (300 – 2000 Hz) et recommencez.

V) Analyse spectrale

Dans la fenêtre SPTool, sélectionnez un signal, puis dans la colonne Spectre, cliquez sur 'Create'. Dans la fenêtre Spectrum Viewer, choisissez une méthode d'analyse spectrale puis cliquer sur apply.

Manipulations sous Matlab

Le filtrage du vecteur x par le filtre numérique défini par a et b est effectué par :
>> y = filter(b,a,x);

où b représente le polynôme en Z au numérateur et a, le polynôme en Z au dénominateur.

Soit le filtre suivant $H(z)=(1+z^{-1})/(1+z^{-1}+z^{-2})$. Lorsqu'on écrit le filtre de cette façon, nous supposons avoir déjà réaliser la synthèse du filtre.

1. Tracez la réponse fréquentielle du filtre par la commande
   >>H=freqz(b,a,f,fe)
   f sera pris égal à 1000 Hz, et fe à 8192 Hz

2. Ouvrez un fichier en .wav sous GoldWave, modifiez la fréquence d'échantillonnage à 8192 Hz
3. Récupérez le signal .wav sous Matlab et filtrer le.
4. Ecouter le signal, visualisez son spectre

Nous allons maintenant définir le gabarit d'un filtre et à partir de ce dernier, sans en connaître sa fonction de transfert, générer la Transformée en Z du filtre numérique. Il existe plusieurs méthodes, non exactes sous Matlab, que nous définissons ci-après.

Synthèse des filters RIF

IL existe différentes méthodes de synthèse de fitres RIF, approchant un filtre idéal, la première consiste à tronquer la réponse impulsionnelle d'un filtre :

h=fir1(n,fn,type,windows)
où n est l'ordre du filtre, les fréquences fn sont normalisées par rapport aux fréquences de Nyquist, (fn=f/(fe/2))), la chaîne de caractère précise le type de filtre (haut, bas) et window le fenêtrage utilisé (pour la troncature).

H=firls(n,fn,m) est un filtre obtenue par la méthode des moindres carrées.

Synthèse des filters RII

La méthode utilisée est obtenue par transposition des fréquences (méthodes d'équivalence).

A titre d'exemple,
>> [bd,ad]=bilinear(b,a,fe)

Attention, n'oubliez pas que la transformation bilinéaire provoque une deformation des fréquences.. Il est donc nécessaire de pré-déformer le gabarit du filtre analogique.

Sous Matlab, on peut aussi générer des filtres de type Butterworth, Tchebyscheff par les commandes :

[n, fn] = buttord(fp,fs,Rp,Rs) ;
[n, fn] = cheb1ord(fp,fs,Rp,Rs) ;
[n, fn] = ellipord(fp,fs,Rp,Rs) ;

où n représente l'ordre du filtre.

fn fréquence propre du filtre numérique. Pour un filtre passe-bas fp et fs sont les fréquences hautes de la bande passante et basse de la bande coupée.
Pour un filtre passe-bande, fp contient les fréquences basse et haute de la bande passante et fs les fréquences haute et basse de la bande coupée. Attention, les fréquences sont normalisées par rapport à la fréquence de Nyquist fe/2.

Application : Conception d'un filtre FIR passe bas par la méthode des moindres carrés

```
% Génération du signal
Fe = 8e3;
N = 512;
t = (0:N-1)/Fe;
x = square(2*pi*Fe*t/50);
% TFD sur [0, Fe]
X = fft(x);
f = (0:N-1)/N*Fe;
% Affichage
subplot(1,2,1); plot(t,x);
xlabel('temps t'), ylabel('x(t)');
subplot(1,2,2); plot(f(1:N/2),abs(X(1:N/2)));
xlabel('fréquence f'), ylabel('X(f)');


% Synthese du filtre passe bas
% (RIF moindres carrées)
% Bande passante [0, 200 Hz]
% Bande coupée [400Hz, 4000Hz]
% Réponse impulsionnelle
h = firls(39,[0 500 750 Fe/2]/Fe*2,[1 1 0 0]);
% Réponse en fréquence
[H, freq] = freqz(h,1,512,Fe);
% Affichage
```

```
subplot(1,2,1); plot(h);
xlabel('échantillon'), ylabel('h[n]');
subplot(1,2,2); plot(freq,20*log10(abs(H)));
xlabel('fréquence f'), ylabel('H(f)');



% Filtrage du signal
y = filter(h,1,x);
Y = fft(y);
% Affichage
subplot(1,2,1); plot(t,y);
xlabel('temps t'), ylabel('y(t)');
subplot(1,2,2); plot(f(1:N/2),abs(Y(1:N/2)));
xlabel('fréquence f'), ylabel('Y(f)');
```

Rmq : La fonction firls synthétise un filtre RIF approchant au mieux, au sens des moindres carrées (norme L2), la réponse en fréquence du filtre analogique idéal.

Récupérer une séquence audio que vous sur-échantillonnerez en conséquence (avec GoldWave), appliquer un filtrage du signal et écouter le résultat pour différents types de filtres.

Recommencez avec un filtre FII.en utilisant la méthode bilinéaire.