TP N 01 : Synthèse et application d'un filtre RIF par la méthode des fenêtres (TFD, Hanning, Hamming, Bessel et/ou Blackman).

But de TP: dans ce TP on teste deux méthodes différent pour synthétiser un filtre RIF : méthodes de séries de Fourier (ou méthode des fenêtres), et la méthode de la transformé de Fourrier Discrète TFD dite par échantillonnage fréquentiel.

Rappel: un filtre de réponse impulsionnelle finie (RIF) possède une fonction de transfert polynomiale, sont aussi des exemples des systèmes discrets linéaires et invariants, ils sont également appelés filtres transversaux, et ils ont le plus souvent une structure non récursifs. Autrement dits leur fonction de transfert possède que des zéros et par conséquence ils sont toujours stable. Les filtres RIF peuvent êtres également à phase linéaire, qui est une caractéristique très sollicitée, si leur réponse impulsionnelle h(n) (composé de N échantillons) appartient à 1 un des cas suivants :

- h(n) possède un axe de symétrie h(n)=h(N-1-n) et N est impair.
- h(n) possède un axe de symétrie h(n)=h(N-1-n) et N est pair.
- h(n) possède un axe d'antisymétrie h(n)=h(N-1-n) et N est impair.
- h(n) possède un axe d'antisymétrie h(n)=h(N-1-n) et N est pair.

Il ne peut pas être obtenu par transposition d'un filtre continu, comme cela est fait pour les filtres RII. Les filtres RIF présentent l'inconvénient de nécessiter un grand nombre de coefficient pour obtenir les mêmes caractéristiques fréquentielles. Mais par contre, ils sont inconditionnellement stables. On peut synthétiser des filtres RIF à phase linéaire, c'est-à-dire à temps de propagation de groupe constant.

Soit *H(z)* la transformée en z d'un filtre numérique donnée dont la décomposition sous forme fractionnelle est donnée par :

$$H(z) = \frac{b_0+b_1z^{-1}+\cdots\dots\dots\dots\dots+b_Nz^{-N}}{1+b_1z^{-1}+\cdots\dots\dots\dots\dots+b_Mz^{-M}}$$

Grace à la seule connaissance du vecteur b et du vecteur a, on peut analyser tout filtre et :

- Déterminer les pôles et les zéros du filtre (et étudier sa stabilité).
- Déterminer la réponse impulsionnelle ou indicielle.
- Déterminer la réponse fréquentielle et le retard de groupe (dérivée de la phase), etc

Quelques fonctions utiles

$\boldsymbol{S = \ filter(b, a, e)}$: filtrer numériquement les données stockées dans le vecteur e avec le filtre décrit à la fois par le vecteur b (coefficient du numérateur de H(z)) et le vecteur a (coefficients du dénominateur de H(z)) pour une entrée e. Il faut normaliser l'équation de telle sorte $a_0 = 1$.
Pour déterminer la réponse impulsionnelle sera un Dirac, pour la réponse indicielle sera un échelon.

$\boldsymbol{[H, F] = freqz(b, a, N, fe)}$**:** retourne N valeurs du gain complexe (réponse fréquentielle TFD à du filtre numérique à la fréquence f_e (Hertz), décrit par b et a. Ces valeurs sont stockées dans H et calculées pour N fréquence mise dans f. Les fréquences sont équi-espacées sur l'intervalle$[0, {}^{fe}/_{2}]$.
(H : représente l'amplitude et F représente la phase).
$\boldsymbol{[b, a] = invfreqz((H, f, nb, na)}$: retourne le numérateur b (d'ordre n_b) et le dénominateur a (d'ordre n_a) à partir de la réponse fréquentielle donnée par H et F.

$\boldsymbol{[h, n] = impz(b, a, N, fe)}$**:** retourne la réponse impulsionnelle du filtre numérique décrit par b et a. La réponse imulsionnelle est calculée en N valeurs stockée et espacés de $1/fe$, les valeurs de réponse correspondante sont stockées dans h.

$\boldsymbol{[tau, f] = grpdelay(b, a, N, fe)}$: retourne le retard de groupe (dérivée de la phase) du filtre numérique décrit par b et a. La réponse impulsionnelle est calculée en N fréquences mises dans f.

$\boldsymbol{zplane(b, a)}$: permet de tracer les pôles et les zéros dans le plan complexe.

$\boldsymbol{fft}$: (Fast fourier transform) : fournit la transformé de fourrier rapide.

$\boldsymbol{fftshift}$: Permutation des hautes fréquences vers les fréquence négatives.

$\boldsymbol{ifft}$: fournit la transformé de fourrier discrètes inverse.

Gabarit d'un filtre :

Le gabarit d'un filtre n'est autre que l'ensemble des caractéristiques du filtre à savoir :
- Le gain du filtre dans la bande passante BP.
- L'atténuation du filtre en bande coupée f_a .
- La fréquence de coupure f_c , on l'exprime souvent sous forme normalisée par rapport à la fréquence d'échantillonnage.
- La largeur de bande de transition Δf souhaitée qui doit être la plus petite possible.
- Les éventuelles oscillations en bande passante et/ou atténuée.

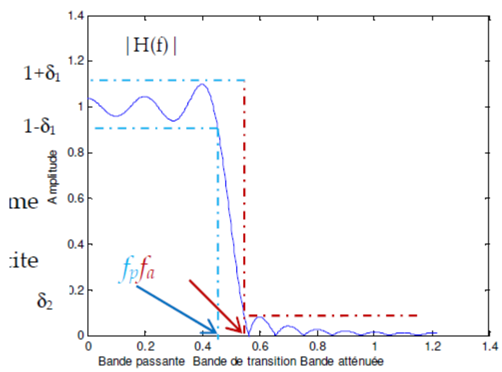

Figure 1 : gabarit d'un filtre numérique.

I. Mise en œuvre de la méthode de Gibbs

Tester le programme suivant :

- Te = 0.2 ;
- fe =1/Te ;
- N = 50 ;
- t=Te*(-N/2:N/2) ;
- N=length(t) ; h = Te*sinc(t);
- figure (1) ;
- subplot(2,1,1) ; stem(t,h) ;grid ; xlabel('Ampl') ; title('Sinc tronqué')
- NF =2048 ; H=fft(h,NF) ;
- H= fftshift (H) ;
- axe_f = fe*(-1/2:1/NF:1/2-1/NF) ;
- subplot(2,1,2) ; plot(axe_f,abs(H)); grid ; xlabel('f') ; ylabel('Amp') ; title('TF sinc tronq')

1- Identifier ce filtre.
2- Expliquer la provenance des oscillations observées.
3- Utiliser le zoom pour mesurer le dépassement en bande passante (en %)
4- Déterminer f_a et f_b la largeur de la bande de transition puis f_c (fréquence de coupure).
5- Prendre N = 100 puis 500 puis 1000 et mesurer à nouveau le dépassement et les 3 fréquences. Commenter
6- Que représente N ?
7- Comment réduire les oscillations ?
8- Choisir un fenêtrage et l'appliquer puis commenter ?

Re: On constate de faibles ondulations en bande passante qui pourrait être améliorées afin d'obtenir une réponse plus plane. Dans la zone de transition, la réponse plonge convenablement, mais ensuite des ondulations importantes apparaissent dans la bande coupée, dépassant même le gabarit. Ces ondulations sont dues à l'utilisation de la fenêtre rectangulaire qui a pour effet d'accentuer ces lobes secondaires.

Pour réduire ces oscillations nous pouvons augmenter la taille de la fenêtre N (augmenter N) mais surtout utiliser d'autres fenêtres aux angles moins abruptes que celle de la fenêtre rectangulaire.

Réalisation par MATLAB du filtre :

- La détermination des coefficients d'un filtre RIF par la méthode de la fenêtre (méthode de série de fourrier) est réalisée par la fonction MATLAB **fir1**.
- Pour utiliser la technique d'échantillonnage de la réponse fréquentielle (TFD), on emploiera la fonction MATLAB **fir2**.

La fonction ***fir1*** en matlab synthétise un filtre RIF simple (défini par une seule bande passante ou coupée) par troncature et fenêtrage de la réponse impulsionnelle du filtre numérique idéal :

- $\boldsymbol{h = fir1(n, f_n, type, window)}$

-

- n est l'ordre du filtre (longueur de la réponse imulsionnelle RI moins 1).
- Les fréquences f_n sont normalisées par rapport à la fréquence de Nyquist ($f_n = \frac{f}{fe*2}, 0 \leq f_n \leq 1$), f_n indique la fréquence de coupure pour le passe-bas et passe-haut, et les fréquences de coupure de passe bande et coupe-bande.
- La chaine de caractère type précise le type de filtre 'high' pour passe-haut, 'stop' pour coup-bande, type omis pour les passe-bas et passe bande.
- Le vecteur window de longueur n+1, correspond à la fenêtre prise en compte (par défaut fenêtre de Hammig).

II. Synthèse RIF par la méthode des fenêtres :

Pour synthétiser un filtre <u>passe-bande</u> par la méthode des fenêtres, répondant aux spécifications suivantes :

```
fa = 500 Hz ;
fb = 1000 Hz ;
fe  = 4000HZ ;
fa1N = fa/(fe/2) ;
fb2N = fb/(fe/2) ;
NbCoeff = 47    ;
```

On procède comme suit:

```
fa = 500  ; % début de la bande passante
fb = 1000 ; % Fin de la bande passante
fe  = 4000 ; % Fréquece d'échantillonnage
faN = fa/(fe/2) ;
fbN = fb/(fe/2) ; % Normalisation de fa et fb
NbCoeff = 47;  % Nombrede coefiscients
N = NbCoeff-1 ; % ordre du filtre

h = fir1(N,[fa1N,fb2N],'band',rectwin(NbCoeff),'scale');
```

```
[H,F]=freqz(h,1,521,fe)  ; % calcul de H(f)
figure (2) ;
subplot(2,2,1) ; hold on ; stem(h,'b')    ; grid    ; title('Tracé de La réponse impulsionnelle')
subplot(2,2,2) ; hold on ; plot(F,abs(H)) ; grid    ;  title ('Tracé du spectre d amplitude')
subplot(2,2,3) ; hold on ; plot(F,20*log10(abs(H))) ; title ('Tracé du spectre d amplitude en db')
subplot(2,2,4) ; hold on ; zplane(h,1); grid on ; title('tracé des poles et des zeros')
```

<u>Travail à faire</u>:

1- Donner l'expression théorique de ce filtre ?
2- Calculer Δf réelle sachant que Δf normalisée soit $\frac{\Delta f}{\frac{fc}{2}} = 1/8N$ théorique puis déterminer la (pratiquement) a partir du graphe en employant les valeurs $1 - \delta_1$ et la tangente à δ_2.
3- Faites de même pour l'atténuation en db. Dépend-elle-de N ?

Reprendre les questions 2 et 3 pour un fenêtrage de Hanning :

```
h = fir1(N,[faN,f2N],'band', hann (NbCoeff), 'scale')
```

5-Superposer les nouveaux graphes pour les 4 précédents graphes (changer de couleur) et commenter.
6- Comment réduire Δf pour une même fenêtre ? Quel est l'inconvénient de cette solution ?
7- Pour NbCoeff = 23, comparer la répartition des zéros pour les 2 fenêtres en établissant la relation avec les tracés des réponses en fréquences leur correspondant.

III. Synthèse RIF par échantillonnage fréquentiel :

On veut synthétiser le même filtre passe-bande par la méthode d'échantillonnage fréquentiel, et comparer le résultat avec la méthode des fenêtres (garder l'ancien programme et rajouter) :

```
Deltaf = fe/(NbCoeff) ;
AA = zeros(1,N/2+1)  ;   % matrice des zeros compsé de 1 ligne et (N/2+1) colonnes
for i=0:N/2 ;
   if(i*Deltaf>fa &&i*Deltaf<fb)
   AA(i+1)=1 ;
   end
end

FF = (0:1/(N/2):1) ;   % Fréquenc normalisée
h1  = fir2(N,FF,AA, rectwin(NbCoeff))        ;
[H1,F1] = freqz (h1, 1, 512, fe)              ;
figure(3) ;
subplot(3,1,1) ;  stem(h1,'r')       ; grid  ;  title('Tracé de La réponse impulsionnelle')
subplot(3,1,2) ; stem(F1,abs(H1),'k:'); grid ;  title ('Tracé du spectre d amplitude')
subplot(3,1,3) ; ; stem(F1,20*log10(abs(H1)),'r'); grid ; title ('Tracé du spectre d amplitude en db')
```

<u>Travail à faire</u>:

- Donner l'expression théorique de ce filtre.

- Expliquer le rôle de la boucle.
- Tester la fenêtre de Hanning également et comparer avec la fenêtre triangulaire (Δf et le dépassement).

Rappel : le fonctions Matlab, disponibles pour créer des fenêtres sont :
Barlett, blackman, boxcar (rectangulaire), chebwin (chebychev), Hamming, hanning, kaiser, triang (triangulaire).

Fenètre	**Observation de la réponse en fréquence**
Rectangulaire	Elle fait apparaitre le plus d'ondulation, mais donne une bonne raideur de coupure malgré les quelques dépassements en bande coupée.
Hamming	Elle fait apparaitre le moins d'ondulation, mais sa raideur de coupure est très mauvaise puisque la réponse en fréquence n'atteint pas l'atténuation minimum requise en bande coupée.
Blackman	Elle fait apparaitre une légère diminution continue du gain en fonction de la fréquence en bande passante, ce qui peut être gênant. D'autre part, la raideur de coupure obtenue est déplorable puisque c'est avec cette fenêtre qu'on obtient l'atténuation minimum en bande coupée.
Hanning	Elle a les mêmes qualités et défaut que la fenêtre de Hamming mais fait apparaitre des ondulations en bande coupée plus rapidement que cette dernière.