

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

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 récurrente. Autrement dits leur fonction de transfert possède que des zéros et par conséquent ils sont toujours stables. Les filtres RIF peuvent être également à phase linéaire, qui est une caractéristique très sollicitée, si leur réponse impulsionnelle $h(n)$ (composé de N échantillons).

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 :

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + b_1 z^{-1} + \dots + b_M z^{-M}}$$

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 indicelle.
- Déterminer la réponse fréquentielle et le retard de groupe (dérivée de la phase), etc

I. Introduction

Un filtre passe-bas est un dispositif électronique ou un circuit qui laisse passer les signaux dont la fréquence est inférieure à une certaine fréquence de coupure, tout en atténuant les signaux dont la fréquence est supérieure à cette fréquence de coupure. En d'autres termes, il permet aux basses fréquences de passer tout en bloquant les hautes fréquences.

Ces filtres sont largement utilisés dans divers domaines, comme l'audio, le traitement du signal, et les communications. Par exemple, dans un système audio, un filtre passe-bas peut être utilisé pour éliminer les bruits indésirables de haute fréquence et ne conserver que les sons graves.

Il existe différents types de filtres passe-bas, y compris les filtres actifs (qui utilisent des amplificateurs) et passifs (qui utilisent uniquement des résistances, des condensateurs et des inductances).

Pour synthétiser et appliquer un filtre RIF (Réponse Impulsionnelle Finie) passe-bas à l'aide de la méthode des fenêtres, il faut suivre plusieurs étapes. Voici une synthèse de la démarche, ainsi qu'une explication des différentes fenêtres que vous pouvez utiliser.

II. Étapes de Synthèse d'un Filtre RIF Passe-bas:

1. Définir les spécifications du filtre:

- Fréquence de coupure (F_c).
- Fréquence d'échantillonnage (F_e).

- Ordre du filtre (N).
- Type de réponse (passe-bas).

2. **Calculer la réponse impulsionnelle idéale:**

- La réponse impulsionnelle d'un filtre passe-bas idéal peut être décrite par:

$$h_d(n) = \frac{\sin(2\pi f_c n)}{\pi n} \quad (1)$$

où f_c est la fréquence de coupure normalisée.

3. **Choisir une fenêtre:**

- Les fenêtres modifient la réponse impulsionnelle pour réduire les effets de lobes secondaires.
- Voici quelques fenêtres populaires :

Fenêtre de Hanning:

$$\omega(n) = 0.5(1 - \cos(\frac{2\pi n}{N-1})) \quad (2)$$

Fenêtre de Hamming:

$$\omega(n) = 0.42 - 0.5 \cos(\frac{2\pi n}{N-1}) \quad (3)$$

Fenêtre de Blackman:

$$\omega(n) = 0.42 - 0.5 \cos(\frac{2\pi n}{N-1}) + 0.8 \cos(\frac{4\pi n}{N-1}) \quad (4)$$

Fenêtre de Bessel:

- Moins courante, elle est souvent utilisée pour des applications spécifiques.

Ou méthode des équations : Approximations telles que Butterworth, Chebyshev, etc.

4. **Appliquer la fenêtre à la réponse impulsionnelle:**

- Multiplier la réponse impulsionnelle idéale par la fenêtre choisie:

$$h(n) = h_d(n) \cdot \omega(n) \quad (5)$$

5. **Normaliser le filtre:**

- Assurez-vous que le filtre soit de type passe-bas en normalisant la somme des coefficients soit égale à 1 (pour un filtre à gain unitaire).

$$h(n) = \frac{h(n)}{\sum_{k=0}^N h(k)} \quad (6)$$

6. Application du Filtre:

a. Convolution:

- Appliquez le filtre à un signal d'entrée en utilisant la convolution:

$$y(n) = x(n) * h(n) \quad (7)$$

où $x(n)$ est le signal d'entrée et $y(n)$ est le signal filtré.

b. Analyse des performances:

- Évaluez la réponse en fréquence du filtre et l'effet de la fenêtre sur la qualité du filtrage.
- Comparez les performances avec différentes fenêtres (par exemple, analysez les lobes secondaires et la bande passante).

Donner un exemple de code MATLAB pour synthétiser et appliquer un filtre RIF passe-bas en utilisant la méthode des fenêtres (Hanning, Hamming, Blackman). Ce code inclut la création du filtre, son application à un signal et l'affichage des résultats.

III.1. Étapes du Code MATLAB à faire:

1. Définir les paramètres du filtre.
2. Créer la réponse impulsionnelle idéale.
3. Appliquer la fenêtre choisie.
4. Normaliser le filtre.
5. Filtrer un signal d'entrée:

$$\text{Signal}_1 = \sin(2\pi ft) + 0.5\text{randn}(\text{size}(t))$$

Avec : randn est un bruit blanc gaussien, $f = 50 \text{ Hz}$.

6. Afficher les résultats.

Exemple de Code MATLAB

III.2. Explications du Code

1. **Paramètres du filtre** : on définit la fréquence d'échantillonnage, la fréquence de coupure, et l'ordre du filtre.
2. **Signal d'entrée** : un signal composé d'une sinusoïde et de bruit est généré.
3. **Réponse impulsionnelle idéale** : calculée à l'aide de la fonction sinc.
4. **Fenêtre choisie** : une fenêtre de Hanning est appliquée, mais vous pouvez facilement passer à Hamming ou Blackman en décommentant les lignes correspondantes.
5. **Filtrage** : la convolution du signal d'entrée avec le filtre est effectuée.
6. **Affichage** : les signaux d'entrée, de sortie et la réponse en fréquence du filtre sont affichés.

Dans cette expérimentation, on va utiliser aussi la commande Matlab « **FIR1** » pour réaliser les différents types de filtres RIF utilisant l'approche de fenêtrage.

✚ 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**. 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 :

$$h = \text{fir1}(N, f_c, \text{type}, \text{window}) \quad (8)$$

- Les fréquences f_c sont normalisées par rapport à la fréquence de Nyquist ($f_c = F_c/(F_e/2)$, $0 \leq f_c \leq 1$), f_c indique la fréquence de coupure pour le passe-bas et passe-haut, et les fréquences de coupure de passe bande et coupe-bande.
- La chaîne de caractère type précise le type de filtre '**low**' pour passe-bas, '**high**' pour passe-haut, '**stop**' pour coup-bande, type omis pour le passe bande.
- Le vecteur window de longueur $n+1$, correspond à la fenêtre prise en compte (par défaut fenêtre de Hammig).

Exemple MATLAB d'un filtre FIR passe-bas

III.3. Étapes du Code MATLAB à faire:

1. Définir les paramètres du filtre: fréquence de coupure, fréquence d'échantillonnage, ordre du filtre.
2. Normaliser la fréquence de coupure.
3. Créer le filtre.
4. Filtrer le signal.
5. Visualiser la réponse en fréquence.
6. Prendre $N = 100$ puis 500 puis 1000 mesurer les 3 fréquences. Commenter.
7. Comment réduire les oscillations ?
8. Choisir d'autres fenêtrage et l'appliquer puis commenter ?
9. Déterminer les pôles et les zéros du filtre (et étudier sa stabilité).
10. Déterminer la réponse impulsionnelle ou indicielle.
11. Déterminer la réponse fréquentielle et le retard de groupe (dérivée de la phase), etc
12. Ré-exécuter le travail avec un signal audio.

III.4. Explications du code:

1. **Paramètres:** on définit la fréquence d'échantillonnage, la fréquence de coupure et l'ordre du filtre.
2. **Normalisation:** la fréquence de coupure est normalisée pour le filtre.
3. **Création du filtre:** la fonction **fir1** est utilisée avec la fenêtre de Hamming pour générer les coefficients du filtre.
4. **Filtrage d'un signal:** on crée un signal de test composé de deux sinusoides (50 Hz et 200 Hz) et on applique le filtre.
5. **Visualisation:** **fvtool/freqz** est utilisé pour visualiser la réponse en fréquence du filtre, et des graphiques montrent le signal original et le signal filtré.
6. **$N = 100, 500$, puis 1000** : voir l'impact du nombre d'échantillons.
7. **Fenêtrage** : utiliser les autres fenêtres Hanning et Blackman, comparer les résultats.
8. **Pole-Zero** : **zplane** est utilisé pour illustrer les pôles et les zéros,
9. **La réponse impulsionnelle** : **impz** est utilisé pour visualiser la réponse impulsionnelle du filtre.
10. **La réponse fréquentielle et le retard de groupe** : par la commande **grpdelay**.

✚ Quelques fonctions utiles

$S = \text{filter}(b, a, e)$: 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 .

Pour déterminer la réponse impulsionnelle sera un Dirac, pour la réponse indicielle sera un échelon.

$[H, F] = \text{freqz}(b, a, N, f_e)$: 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 $[0, f_e/2]$.
(H : représente l'amplitude et F représente la phase).

$[b, a] = \text{invfreqz}(H, F, n_a, n_b)$: 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 .

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

$[\tau, f] = \text{grpdelay}(b, a, N, f_e)$: 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 .

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

fft : (Fast fourier transform) : fournit la transformé de fourier rapide.

fftshift : Permutation des hautes fréquences vers les fréquence négatives.

ifft : fournit la transformé de fourier discrètes inverse.

Fenêtres

```
hanning_window = hanning(N + 1);
hamming_window = hamming(N + 1);
blackman_window = blackman(N + 1);
bessel_window = ones(N + 1, 1); % Bessel est une approximation
```

Application des fenêtres

```
b_hanning = fir1(N, Wn, 'low', hanning_window);
b_hamming = fir1(N, Wn, 'low', hamming_window);
b_blackman = fir1(N, Wn, 'low', blackman_window);
b_bessel = fir1(N, Wn, 'low', bessel_window);
```