

TRAITEMENT NUMERIQUE DU SIGNAL ET SES APPLICATION TDAO3

Les filtres RII en Scilab

- L'avantage principal des filtres RII par rapport aux filtres RIF est qu'ils répondent généralement à un ensemble de spécifications données avec un ordre de filtrage beaucoup plus faible qu'un filtre RIF correspondant.
- Afin de réaliser des filtres avec Signal Processing Toolbox de Scilab, on doit normaliser la fréquence de coupure, de sorte qu'elle ne requière pas la fréquence d'échantillonnage du système comme argument d'entrée supplémentaire. La fréquence normalisée est toujours dans l'intervalle $0 \leq f \leq 0.5$ (en respectant le théorème de Shannon). Par exemple, pour une fréquence d'échantillonnage de 1000 Hz, la fréquence de coupure de 300 Hz devient $f_N = \frac{300}{1000} = 0,3$.
- Le toolbox: Signal Processing fournit des fonctions permettant de créer tout type de filtres RII classiques dans les domaines analogiques et numériques : Butterworth, Chebyshev Types I et II, elliptiques (à l'exception de Bessel).

Filtre Butterworth :

Le filtre **butterworth** est un type de filtre linéaire (analogique et numérique) qui possède une fonction de transfert dont les dénominateurs sont des polynômes d'approximation. C'est le filtre le plus utilisé et celui que l'on rencontre dans les appareils audio et vidéo.

- La fonction *iir* conçoit un filtre numérique en utilisant des conceptions de filtres analogiques. La syntaxe est :

nomdufiltre=iir (n, ftype, fdesign, frq, delta)

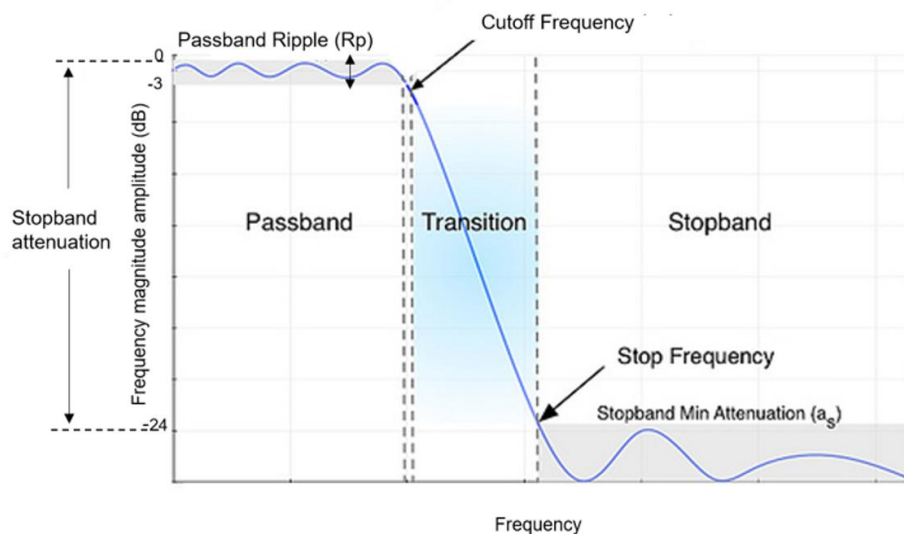
où:

- *n*: l'ordre du filtre
- *ftype*: le type de filtre (passe-bas, passe-haut, passe-bande ...)

- *fdesign*: la technique de conception de filtre analogique à utiliser pour concevoir le filtre, exemple : butterworth, Chebyshev type 1 ou Chebyshev type 2.
 - *frq*: la fréquence de coupure normalisée : Fréquence de coupure/Fréquence d'échantillonnage.
 - *delta* : dans le cas du filtre Butterworth, cet argument n'est pas utilisable. Pour les filtres Chebyshev du premier type, seul le premier élément de ce vecteur est utilisé et il sert comme la valeur de l'ondulation dans la bande passante.
 - *hz* : la fonction de transfert du filtre.
- La fonction *flts* permet de filtrer un signal à l'aide du filtre réalisé comme argument à la fonction. En effet, elle réalise le passage du domaine temporel au domaine fréquentiel, le filtrage fréquentiel puis le passage du domaine fréquentiel au domaine temporel. Elle retourne donc le signal filtré en domaine temporel.

$$\text{Signalde sortie} = \text{flts}(\text{signal à filtrer}, \text{nom du filtre})$$

Utilisez le *help* de Scilab pour plus de détails sur leur utilisation



Exercice 1 : filtrage d'un signal sinusoïdal

- a) Utiliser l'aide de Scilab pour la fonction *iir* pour créer un filtre Butterworth passe-bas d'ordre 4 avec une fréquence de coupure = 25Hz. La fréquence d'échantillonnage = 100 Hz. Utiliser la fonction *frmag* pour dessiner la réponse du filtre en amplitude sur N=512 points. Commenter.

$$[hz, fr] = \text{frmag}(\text{nom du filtre}, N);$$

$$\text{Plot}(fr, hz)$$

- b) Créer un signal sinusoïdal x construit à partir de la somme de 3 signaux sinusoïdaux de fréquences : $f_1=10$ Hz, $f_2=30$ Hz et $f_3=40$ Hz. La fréquence d'échantillonnage est $f_s=100$ Hz et le nombre de points du signal est 512.
- c) Dessiner le spectre signal x (voir TDAO2).
- d) Appliquer le filtre passe-bas sur le signal x et tracer le spectre du signal résultant. On utilisera la fonction *flts* comme expliqué ci-dessus. Commenter votre résultat.
- e) Changer maintenant l'ordre du filtre passe-bas à 10 et appliquer de nouveau le filtre sur le signal, tracer le spectre du signal filtré et la réponse du filtre. Comparer les résultats et conclure.

Exercice 2 : filtrage d'un signal sonore bruité

L'objectif de cette partie est de mettre en application les connaissances acquises dans un cas concret : le filtrage d'un signal sonore numérique entaché d'un bruit de souffle comme cela pourrait se passer dans le cas d'un microphone enregistreur. L'intérêt du filtrage numérique est donc de supprimer ou au moins d'atténuer les parasites sonores induits par le bruit.

- a) Récupérer les données échantillonnées et la fréquence d'échantillonnage du fichier «affreux.wav» en utilisant la fonction : *wavread*. Vous pouvez au besoin lire le fichier (entendre le son) en utilisant la fonction *playsnd*.

$$[y,fe,bits]=wavread('sound_pb.wav')$$

- b) Calculer la transformée de Fourier discrète de ce signal et tracer son spectre centré (voir TDAO2).
- c) Créer un bruit de moyenne=0 et de variance égale à celle du signal. Le bruit doit être de la même longueur que le signal sonore.

$$b=stdev(y)*rand(1,length(y),"n")$$

- d) Ajouter le bruit au signal sonore. Enregistrer le signal bruité en utilisant la fonction *wavwrite*. Attention : utiliser la même fréquence d'échantillonnage pour l'enregistrement :

$$wavwrite(yb, fe, 'affreuxB')$$

- e) Calculer la TFD du signal sonore bruité et tracer son spectre centré. Déterminer où se situe principalement le bruit (dans les basses ou les hautes fréquences) ?
- f) Proposer un filtre numérique de type RII (soit Tchebychev soit Butterworth) pour atténuer le bruit.
- g) Tracer la réponse fréquentielle du filtre proposé.
- h) Appliquer le filtre au signal sonore bruité et tracer le spectre du signal filtré. Vous pouvez écouter les deux sons et les comparer.

Exercice 3

Soit $H(s) = \frac{2}{s+1}$ (s est la variable complexe de la transformée de Laplace $s=p$)

- Créer le numérateur nommé NN et le dénominateur nommé DD de $H(s)$ en utilisant la fonction *poly*.
- Créer la fonction de transfert H en utilisant la fonction *syslin* qui sert à définir un système dynamique linéaire.
- Dessiner le diagramme de Bode en utilisant la fonction *bode(H)*. Quel est le type de ce filtre?
- Calculer théoriquement la valeur de H pour $s=0$ (f très faible). Comparer avec la figure.
- Déterminer la fréquence de coupure.