

# TP de Filtrage Numérique

## SEATECH parcours IRIS

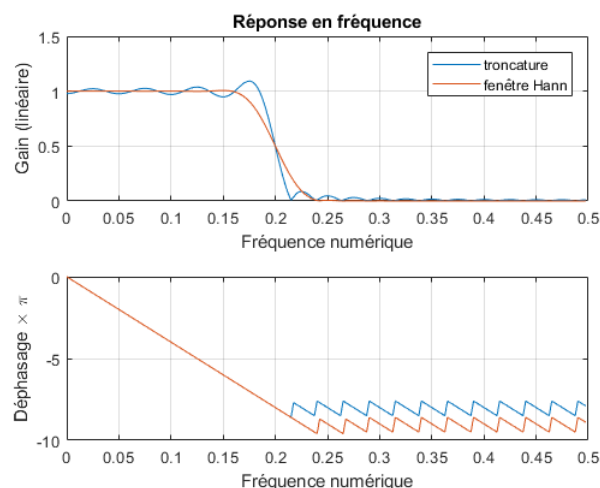
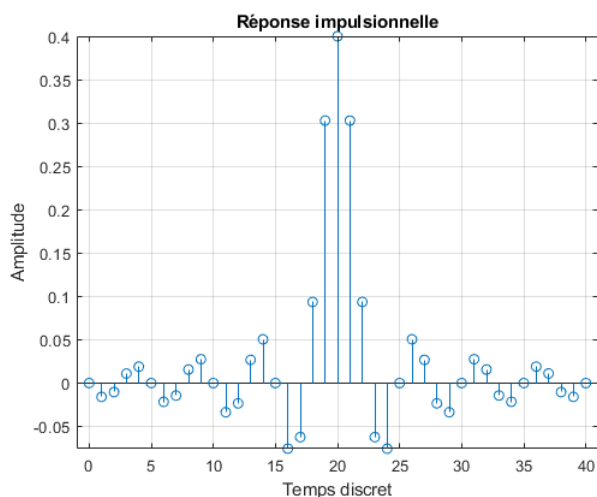
### TP 3 : Synthèse de filtres RIF et AR

## 1 Troncature de la RI et méthode de la fenêtre

On souhaite synthétiser un filtre RIF passe-bas de type 1. On commence par appliquer la méthode de la troncature simple de la RI.

A partir des résultats du cours, générer une RI tronquée d'ordre  $N = 40$ . Afficher la RI et la RF du filtre (gain en échelle linéaire). Mesurer l'amplitude maximale des oscillations dans la bande passante.

Modifier le filtre en appliquant une fenêtre de Hann (fonction `hann()`). Afficher la nouvelle RF, en la superposant avec la RF précédente.



Sous Matlab, il existe une fonction intégrée pour générer des RIF standards par la méthode de la fenêtre, avec la syntaxe suivante :

```
h = fir1(N,vc,'type',w)
```

$N$  est l'ordre du filtre,  $vc$  est le vecteur des fréquences de coupure,  $w$  est le vecteur qui définit la fenêtre et 'type' décrit la forme générale de la courbe de gain. Les valeurs usuelles sont :

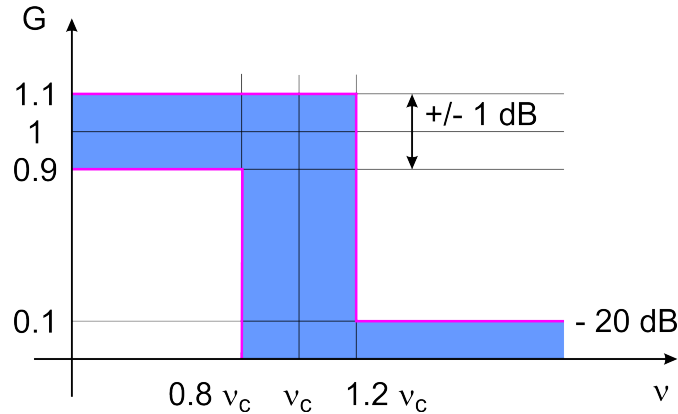
- 'low' : passe-bas (1 fréquence de coupure)
- 'high' : passe-haut (1 fréquence de coupure)
- 'bandpass' : passe-bande (2 fréquences de coupure)
- 'stop' : coupe-bande (2 fréquences de coupure)

**Attention :** Matlab représente la fréquence de coupure en la normalisant par la demi-fréquence d'échantillonnage. La demi-fréquence d'échantillonnage est donc représentée par la valeur 1 et non-pas par la valeur 0.5. Avec cette fonction, il faut donc penser à multiplier les fréquences de coupure par 2.

Générer à nouveau un filtre selon les spécifications précédentes avec la fonction `fir1` et vérifier qu'on obtient bien le même résultat.

## 2 Synthèse RIF sous contrainte

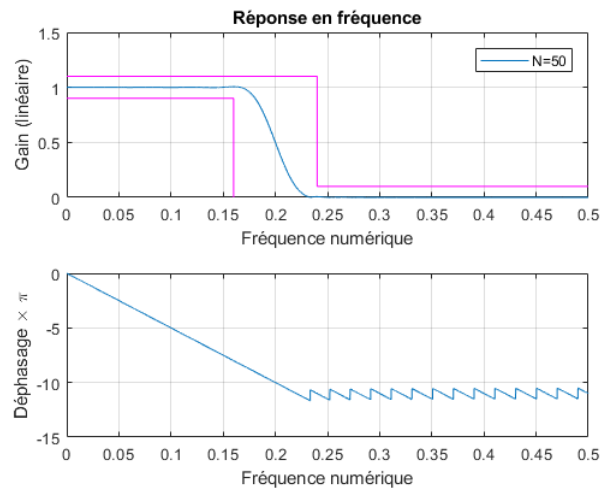
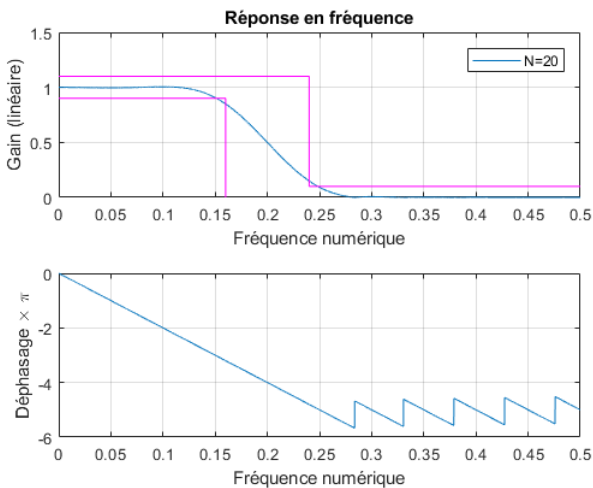
On souhaite synthétiser un filtre RIF passe-bas de type 1 par la méthode de la fenêtre en respectant des contraintes sur la forme de la courbe de gain. Ces contraintes (aussi appelées gabarit) sont résumées par la figure suivante (l'espace possible pour la courbe de gain est représenté en bleu) :



Reprendre la synthèse RIF par fenêtre de Hann précédente, en superposant à la courbe de gain des lignes matérialisant les limites de la contrainte. On pourra utiliser la syntaxe suivante :

```
plot([x1 x2], [y1 y2], 'couleur').
```

En choisissant une fréquence de coupure de 0.2 (soit 40% de la bande de fréquence totale), chercher l'ordre minimum qui vérifie le gabarit. Recommencer avec les fréquences de coupures 0.1 (20% de la bande), 0.05 (10% de la bande), 0.01 (2% de la bande) et 0.005 (1% de la bande). Conclure sur la relation existant entre la fréquence de coupure et l'ordre du filtre pour ce type de gabarit.

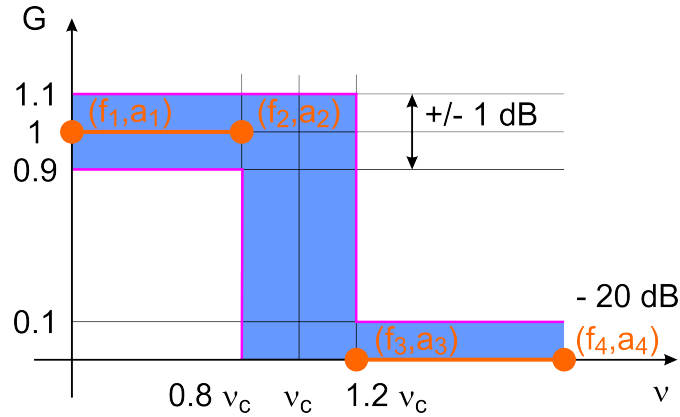


## 3 Synthèse RIF par optimisation numérique

On souhaite synthétiser un filtre RIF passe-bas de type 1 par optimisation MLS, tout en respectant les mêmes contraintes sur la forme de la courbe de gain. La méthode d'optimisation permet de spécifier les limites exactes de la bande passante et de la bande coupée, qu'on fixera donc respectivement à  $[0, 0.8\nu_c]$  et à  $[1.2\nu_c, 0.5]$ . La syntaxe de la fonction intégrée de Matlab est la suivante :

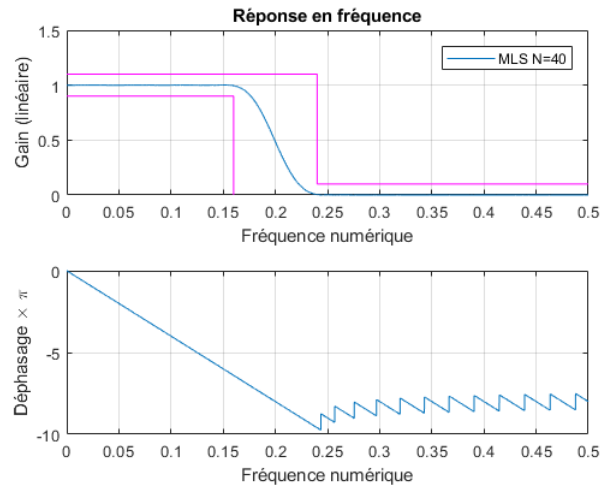
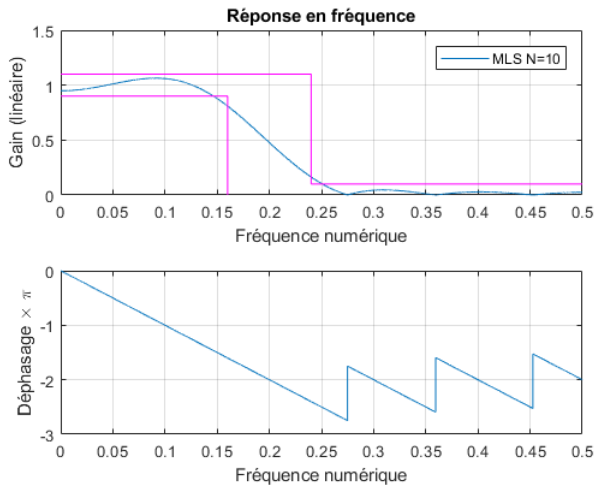
```
h = firls(N,[f1 f2 f3 f4],[a1 a2 a3 a4])
```

Les paramètres  $f_i$  (fréquences limites) et  $a_i$  (valeurs du gain) sont définis comme illustré ci-dessous :



On prendra garde au fait que, comme pour la fonction `fir1`, **il faut multiplier les fréquences par 2**.

Reprendre le même protocole expérimental que dans la question précédente pour chercher, à chaque fréquence de coupure, le plus petit ordre qui vérifie le gabarit. Comparer avec les filtres obtenus par la méthode de la fenêtre et conclure sur l'intérêt de l'optimisation MLS.



## 4 Synthèse RII avec critère MSE

On souhaite synthétiser un filtre passe-bande de fréquences de coupures  $\nu_1 = \nu_0 - D_\nu$  et  $\nu_2 = \nu_0 + D_\nu$ , avec  $\nu_0 = 0.25$  et  $D_\nu = 0.05$ .

A titre de référence, on synthétise des RIF avec une fenêtre de Hann. Superposer sur une même figure les RF des filtres obtenus avec des ordres 10, 20, 50 et 100.

On va maintenant synthétiser des filtres AR au sens du critère MSE. Pour cela, on utilise la fonction Matlab `lpc()` qui synthétise le dénominateur du filtre AR à partir de la réponse impulsionnelle idéale. La réponse impulsionnelle du passe-bande idéal (RII) s'écrit :

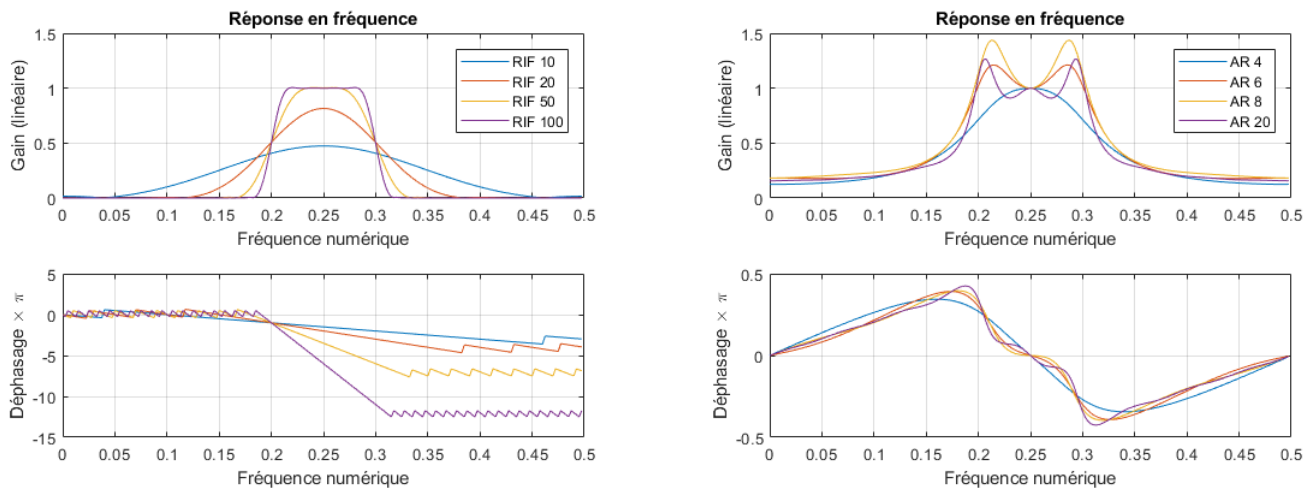
$$h[n] = 2\nu_1 \text{sinc}(2\pi\nu_1 n) - 2\nu_2 \text{sinc}(2\pi\nu_2 n)$$

Pour obtenir une bonne estimation du filtre AR, il faut partir d'une version suffisamment longue de la RI idéale. Générer la RI idéale pour  $n \in [0, 1000]$ .

Pour calculer le dénominateur du filtre AR optimal d'ordre  $M$ , on utilise la syntaxe : `A = lpc(h,M)`

Le filtre AR normalisé correspond à  $B = 1$ , mais cela ne garantit pas un gain égal à 1 dans la bande passante. En utilisant la fonction `freqz()`, calculer le numérateur de la fonction de transfert  $B$  qui assure un gain 1 à la fréquence  $\nu_0$ .

Superposer sur une même figure les RF des filtres AR obtenus avec des ordres 4, 6, 8 et 20. Comparer les courbes de gain des filtres RIF et AR.



## 5 Filtre de prédiction AR

On va maintenant synthétiser le filtre AR associé à un prédicteur à long terme pour un signal de parole. Charger le signal contenu dans le fichier `meteo1.wav` et extraire la portion de signal entre les échantillons 1901 et 2600. Afficher la forme d'onde de la portion signal et son spectre (énergie en dB, sur 500 points en fréquence).

Avec la fonction `lpc`, effectuer une optimisation AR d'ordre 2. Superposer la RF du filtre AR au spectre du signal de parole.

Appliquer le filtre d'analyse (filtre RIF inverse du filtre AR) à la portion du signal de parole pour obtenir le résiduel de prédiction. Superposer la forme d'onde du résiduel à celle du signal de parole. Superposer le spectre du résiduel aux autres spectres. Recommencer pour des ordres 2, 4, 6, 8 et 10. Conclure sur l'effet blanchissant du filtre d'analyse.

