

M1 info : Bases du Traitement du Signal

TP : Synthèse de filtres RIF

A la fin de ce TP, vous devez rendre un compte-rendu sur une copie identifiée par votre login de manière visible (pas sous le rabat). N'oubliez pas de sauvegarder les fichiers *.sce* modifiés, qui seront récupérés de manière automatique à la déconnexion, de même que les figures indiquées.

Il est rappelé qu'un compte-rendu de TP doit être rédigé avec soin et lisiblement. Toute observation doit faire l'objet à la fois d'une description concise et d'une interprétation.

1 Synthèse par fenêtrage

Un filtre passe-bande idéal de fréquences de coupure normalisées $f_1 = 1/8$ et $f_2 = 3/8$ a pour réponse impulsionnelle :

$$h(n) = \frac{1}{2} \cos\left(\frac{\pi n}{2}\right) \text{sinc}\left(\frac{\pi n}{4}\right)$$

Ce filtre, à réponse impulsionnelle infinie, va être approché par la méthode du fenêtrage.

1) Ouvrez *TP_1.sce* dans l'éditeur de scilab. Ce programme génère $L = 2M + 1$ échantillons de $h(n)$ (pour $n \in [-M; +M]$), calcule la réponse fréquentielle H par une FFT sur $NFFT = 512$ points et affiche le module de H .

Lancez le programme. D'après la réponse impulsionnelle, quel est le nombre minimum de multiplication-accumulation effectuées par le filtre pour chaque échantillon d'entrée ? Expliquer l'aspect de la réponse fréquentielle obtenue.

Calculez et dessinez le gabarit théorique de la réponse fréquentielle. Décommentez l'appel de la fonction `trace_gabarit_BP` et donnez-lui les bons paramètres pour vérifier votre résultat.

2) On souhaite une atténuation supérieure à 50 dB dans la bande atténuée et la même largeur de bande de transition que précédemment. Comment s'y prendre ? Testez votre proposition en sauvegardant le fichier sous un autre nom (*TP_1bis.sce*). Pour changer les extrema des axes de la figure, voir l'annexe. Que constatez-vous ? Enregistrez votre figure sous *TP_1.eps*.

Pour quelle valeur de L la réponse fréquentielle rentre-t-elle dans le gabarit ?

2 Application au multiplexage de sons numérisés

Note : pour écouter un fichier son toto.wav sous linux, taper play toto.wav dans une fenêtre de commande... tout simplement.

Ouvrez et exécutez le programme *TP_2.sce*.

Ce programme synthétise deux filtres passe-bas de fréquence de coupure normalisée 1/4 et de même largeur de bande de transition :

- h_1 , par la méthode du fenêtrage avec une fenêtre rectangulaire ;
- h_2 , par la méthode du fenêtrage avec une fenêtre de Hamming.

Le programme crée deux signaux échantillonnés à 16 kHz :

- un signal de parole x , de puissance 10, dont le spectre est de support borné par $\nu_{max} = 3500$ Hz (fichier son *fcjf0_sa1_LP.wav*) ;
- un signal harmonique y de puissance 1000, de fréquence fondamentale $\nu_0 = 150$ Hz, composé de 30 harmoniques (fichier son *harmonique.wav*)

1) Calculez le rapport, en dB, entre les puissances des deux signaux. Quelle est la fréquence de la dernière harmonique de y ?

2) Les deux signaux sont multiplexés selon la méthode décrite en annexe. Le fichier son correspondant est *multiplex.wav*. On extrait le signal de parole par un filtrage passe-bas (fonction *convol*). Le signal résultant est :

- x_1 si l'on utilise le filtre h_1 (fichier son *x1.wav*)
- x_2 si l'on utilise le filtre h_2 (fichier son *x2.wav*)

Comparez *x1.wav* et *x2.wav*. Expliquez.

3) Si le filtrage passe-bas est idéal, pourquoi peut-on *a priori* sous-échantillonner la sortie du filtre avec un facteur 1/2 ? (*i.e.* supprimer 1 échantillon sur 2 et passer à une fréquence d'échantillonnage de 8000 Hz)

Les signaux x_1 et x_2 sont ainsi sous-échantillonnés et les signaux correspondant sont enregistrés respectivement dans *x1_ssech.wav* et *x2_ssech.wav*. Écoutez, comparez au fichier *harmonique.wav* et expliquez précisément ce qui se passe, en vous aidant des représentations spectrales (faire des dessins).

3 Annexe

3.1 Multiplexage de données numériques

On cherche à transmettre simultanément deux signaux échantillonnés $x(n)$ et $y(n)$. Les spectres des signaux analogiques originaux, respectivement $X(\nu)$ et $Y(\nu)$, sont tels que

$$X(\nu) = Y(\nu) = 0 \quad \forall \nu \text{ tel que } |\nu| > \nu_e/4$$

comme le montre la figure 1

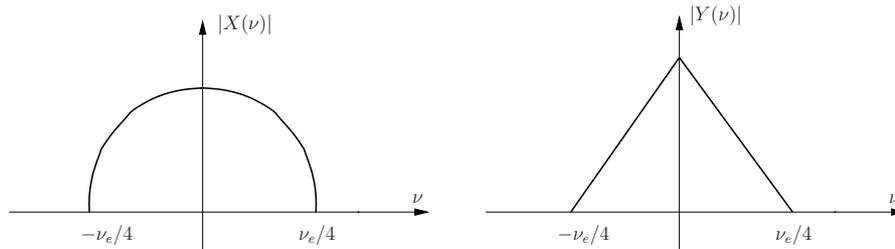


FIG. 1 – Spectres d'amplitude des signaux analogiques originaux.

On pose $z(n) = (-1)^n y(n)$ et on transmet $w(n) = x(n) + z(n)$. Comme

$$Z_e(f) = Y_e(f + 1/2)$$

les composantes fréquentielles des deux signaux sont bien séparées dans le spectre de w , comme indiqué sur la figure 2

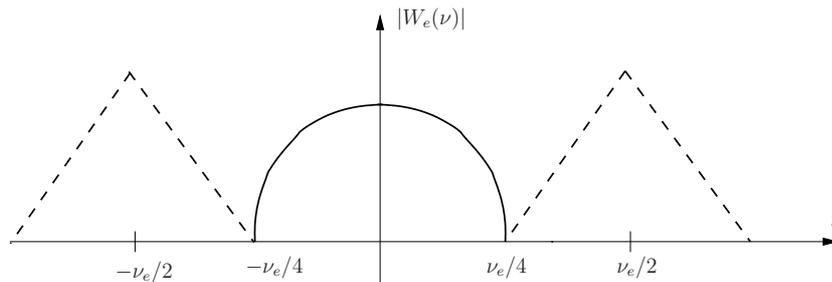


FIG. 2 – Spectre d'amplitude du multiplexe.

Lorsqu'on reçoit le multiplexe w , on peut donc récupérer $x(n)$ par filtrage passe-bas de fréquence de coupure $\nu_e/4$

3.2 Manipulation des vecteurs

Attention, en scilab les indices des vecteurs commencent à 1 au lieu de 0.

Pour prélever 1 échantillon sur K du vecteur x , de l'indice n_1 à l'indice n_2 :

```
x_ech = x(n_1:K:n_2);
```

Pour permuter la première et la deuxième moitié d'un vecteur ligne x de longueur 10 :

```
x = [x(6:10),x(1:5)];
```

Suppression des 2 premiers et des 2 derniers échantillons de x :

```
x = x(3:8);
```

Multiplication d'un vecteur ligne y de longueur N par une fenêtre de Hamming :

```
y_fen = y.*window('hm',N);
```

3.3 Courbes avec scilab

```
xset("window",0)
```

```
xbasc()
```

```
plot2d(x,y,c)
```

Sélection ou création de la figure 0

Effacement du contenu de la figure

x = vecteur des abscisses, y = vecteur des ordonnées, c = couleur

Code des couleurs :

1 : noir

-9 : 1 rond noir par échantillon

2 : bleu

3 : vert

4 : jaune

5 : rouge

...

```
plot2d([x,y,c,"011"," ",
```

```
[x_min y_min x_max y_max])
```

Définition des bornes de la figure