

TD Synthèse de filtres numériques RIF

I. Introduction

L'objectif de cette manipulation est de synthétiser et de mettre en œuvre un filtre numérique dans le but d'extraire un signal noyé dans un bruit.

Le signal à traiter est un signal basse-fréquence, issu d'un capteur dont le faible rendement ne permet pas d'obtenir un bon rapport signal sur bruit. La fréquence d'échantillonnage est $F_e=8$ kHz. On sait que l'acquisition est fortement perturbée par un bruit de type haute-fréquence.

Tout naturellement, votre travail se décompose en trois parties :

- l'analyse du signal à filtrer,
- la synthèse du filtre numérique,
- le filtrage du signal

II. Analyse du signal

Le résultat de l'acquisition est stocké dans un fichier appelé *fichsig.mat* (la commande *load fichsig.mat* permet de récupérer le signal bruité sous la forme d'un vecteur nommé *sig*).

1. Tracer la représentation temporelle du signal (il suffit d'enlever les commentaires et d'exécuter la partie II.1 du programme *RIF.m*). Combien d'échantillons ont été prélevés?
2. Compléter la partie II.2 du programme pour calculer la transformée de Fourier discrète du signal (avec la commande Matlab *FFT*) puis tracer son module.
3. Délimiter le domaine fréquentiel correspondant au signal utile ainsi que celui du bruit.

III. Synthèse de filtre RIF

Afin d'éliminer le bruit, on va filtrer le signal bruité. On requiert que les caractéristiques fréquentielles du filtre soient les suivantes :

- la bande de transition entre la bande passante et la bande atténuée doit être de l'ordre de 1 kHz,
- l'atténuation en bande passante doit être inférieure à 1dB,
- l'atténuation en bande atténuée doit être supérieure à 72dB.

1. Compléter la partie III.1 du programme pour tracer le gabarit du filtre à synthétiser.

Pour synthétiser des filtres RIF, on utilisera la méthode vue en cours consistant à tronquer la réponse impulsionnelle d'un filtre idéal. On rappelle que la réponse impulsionnelle d'un filtre passe-bas idéal de pulsation de coupure ω_c est

$$h_{id}[n] = \frac{\omega_c}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi}n\right)$$
, ou si on veut l'exprimer en fonction de la fréquence du signal analogique échantillonné,
$$h_{id}[n] = \frac{2f_c}{F_e} \operatorname{sinc}\left(\frac{2f_c}{F_e}n\right)$$
.

2. Considérant le gabarit de la partie III.1, choisir la fréquence de coupure du filtre idéal (f_c). En utilisant la fonction Matlab *sinc*, compléter la partie III.2 du programme *RIF.m* pour calculer la réponse impulsionnelle tronquée sur l'intervalle $[-P,P]$ et tracer le résultat.

L'ordre du filtre ($M=2P$) n'étant pas connu à l'avance, on procédera par essais successifs pour déterminer l'ordre du filtre répondant au gabarit.

3. La partie III.3 du programme permet de tracer la réponse en fréquence du filtre ($20 \log_{10}|H(\omega)|$) et de la comparer avec le gabarit. Par différents essais successifs, correspondant aux différentes valeurs de P , déterminer l'ordre du filtre nécessaire pour satisfaire le gabarit. Etant donné le nombre d'échantillons du signal, est-il possible de filtrer le signal avec un filtre de tel ordre ?
4. La partie III.4 du programme permet de pondérer la réponse impulsionnelle calculée dans la partie précédente par une fenêtre de Hanning. Par différents essais successifs, correspondant aux différentes valeurs de P , déterminer l'ordre du nouveau filtre nécessaire pour satisfaire le gabarit. Conclusion ?

IV. Filtrage

Le résultat idéal du filtrage est le signal non bruité. Afin de rendre possible les comparaisons, celui-ci est accessible dans le vecteur *sig_ideal* qui a été chargé lors de l'exécution de la commande *load fichsig.mat*.

1. La partie IV.1 du programme permet de filtrer le signal bruité à l'aide du filtre choisi et de tracer le signal de référence et le signal filtré. Expliquer les différences entre les deux signaux.

Annexe : Programme RIF.m

```
% TD Synthèse de filtres numériques
% L3 IUP AISEM/ICM
% 2006-2007

clear all
close all
nom='?????????';

% II. Analyse du signal
%%%%%%%%%%
PARTIE II.1
%%%%%%%%%%
% load fichsig;
% L=length(sig); %Nombre d'échantillons
% Fe=8000; %Fréquence d'échantillonnage
% Te=1/Fe;
```

```

% t=[0:L-1]*Te; %vecteur de temps
% f=0:Fe/L:Fe-Fe/L; %vecteur de fréquence
% figure(1);
% plot(t,sig);
% title(['PARTIE II.1', nom]);
% xlabel('t');
% ylabel('sig');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PARTIE II.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TFD= ?;
% module_TFD= ?;
% figure(2);
% plot(f,module_TFD);
% title(['PARTIE II.2', nom]);
% xlabel('f');
% ylabel('TFD');

% III. Synthèse de filtre RIF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PARTIE III.1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fp=?; % fin de la bande de passante
% fs=?; % début de la bande atténuée
% Rp=?; % atténuation maximale dans la bande passante
% Rs=?; % atténuation minimale dans la bande atténuée
% Gabf=[0 fp fp; fs fs Fe/2];
% GabH=[-Rp -Rp -Rs; -Rp -Rs -Rs];
% figure(3)
% plot(Gabf,GabH');
% title(['PARTIE III.1', nom]);
% xlabel('f');
% ylabel('Gabarit du filtre a implanter (db)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PARTIE III.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fc=?; % fréquence de coupure du filtre idéal
% P=?;
% n=[-P:P];
% h=?; % réponse impulsionnelle du filtre tronqué
% figure(4);
% title(['PARTIE III.2', nom]);
% ylabel('réponse impulsionnelle tronquée');
% plot(h);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PARTIE III.3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [Hrect,fdirect]=freqz(h,1,4000,Fe);
% figure(5);
% plot(fdirect,20*log10(abs(Hrect)));
% title(['PARTIE III.3', nom]);
% xlabel('f');
% ylabel('Réponse en fréquence du filtre tronqué');
% hold on
% plot(Gabf,GabH','r');

```

```

%%%%%%%%%%
% Partie III.4
%%%%%%%%%%
% h_hanning=h.*hanning(2*P+1);
% [H_hanning,fd_hanning]=freqz(h_hanning,1,4000,Fe);
% figure(6);
% plot(fd_hanning,20*log10(abs(H_hanning)));
% title(['PARTIE III.4', nom]);
% xlabel('f');
% ylabel('Réponse en fréquence du filtre tronqué Hanning');
% hold on
% plot(Gabf,GabH,'r');

%%%%%%%%%%
% Partie IV.1
%%%%%%%%%%
% sig_filtre=filter(h_hanning,1,sig);
% figure(7);
% plot(t,sig_ideal,'b',t,sig_filtre,'r--');
% title(['PARTIE IV.1', nom]);
% xlabel('t');
% ylabel('signux filtre (rouge) et ideal (blue)');

```