Synthèse de filtres numériques

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.

Avant d'arriver en séance, vous devez impérativement avoir assimilé les notions théoriques concernant :
- la représentation spectrale des signaux discrets et en particulier la Transformée de Fourier Discrète ;
- la synthèse des filtres à réponse impulsionnelle finie (RIF) et à réponse impulsionnelle infinie (RII) ;
- le filtrage des signaux et la représentation spectrale de cette opération.

De même, vous devez avoir :
- étudié l'annexe concernant l'utilisation de MATLAB et de sa boîte à outils *traitement du signal* ;
- étudié ce sujet de TP, tant du point de vue méthodologique que des fonctions MATLAB utilisées.

Le signal à traiter est issu d'un capteur dont le faible rendement ne permet pas d'obtenir un bon rapport signal sur bruit. Le schéma d'acquisition du signal est donné Figure 1.

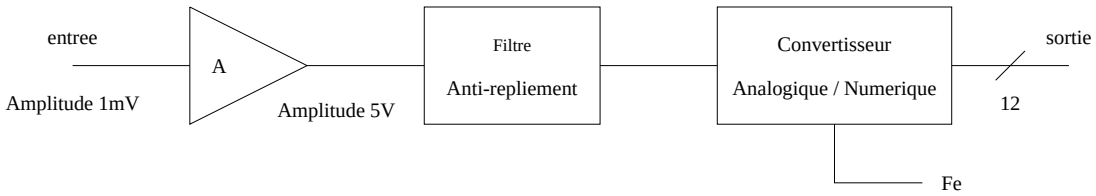

FIG. 1 – Manipulation

La fréquence d'échantillonnage est $F_e = 8$ kHz et l'amplitude du signal est quantifiée sur 12 bits. Le résultat de l'acquisition est stocké dans un fichier appelé `fichsig.mat`.

Tout naturellement, votre travail se décompose en trois étapes :
- l'analyse du signal à filtrer ;
- la synthèse du filtre numérique ;
- le filtrage du signal.

Pour ces trois étapes vous utiliserez le logiciel MATLAB. En particulier, vous utiliserez la boîte à outils *traitement du signal* (*signal processing toolbox*) de MATLAB. Une introduc-

tion à MATLAB est proposée en annexe.

II ANALYSE DU SIGNAL

A l'aide de la commande MATLAB `load fichsig.mat` vous allez récupérer le signal sous la forme d'un vecteur nommé `sig`.

A l'aide du logiciel MATLAB :

1. Tracez la représentation temporelle du signal.
2. Tracez la représentation spectrale du signal.

A partir de la représentation en échelle logarithmique de la représentation spectrale du signal, sachant que le bruit est de type passe-haut, délimitez le domaine fréquentiel correspondant au signal utile.

3. Calculez grossièrement le rapport signal sur bruit.
4. En fonction de la chaine d'acquisition, calculez le rapport signal sur bruit de quantification maximum ; quelle atténuation maximale doit on prendre en compte pour atténuer le bruit ?

III SYNTHÈSE DE FILTRES

Afin d'éliminer le bruit on va effectuer le filtrage du signal. On requiert que les caractéristiques fréquentielles du filtre soient les suivantes :
- la bande de transition entre la bande passante et bande atténuée doit être de l'ordre de 1 kHz ;
- l'atténuation en bande atténuée doit utiliser au mieux la dynamique permise par la chaine d'acquisition.

1. Déterminez la réponse en fréquence du filtre analogique ideal ainsi que le gabarit du filtre numérique à synthétiser.

III.1 Filtres à réponse impulsionnelle finie (RIF)

Pour synthétiser des filtres à réponse impulsionnelle finie, on utilisera la méthode consistant à tronquer la réponse impulsionnelle du filtre numérique idéal. Les ordres des filtres n'étant pas connus à l'avance, on procédera par essais successifs pour déterminer l'ordre du filtre répondant au gabarit.

2. Calculez la réponse impulsionnelle du filtre analogique idéal.
3. Discrétisez et tronquez (sur un intervalle $[-P, P]$) cette réponse impulsionnelle.
4. Ponderez cette réponse impulsionnelle par les fenêtres rectangulaire et de Blackman.

5. Pour ces deux fenêtres et pour les ordres choisis, tracez les réponses en fréquence correspondantes (cf. la fonction Matlab **freqz**).
6. Calculez le temps de propagation de phase de ces deux filtres.
7. Conclusions...

III.2 Filtres à réponse impulsionnelle infinie (RII)

Pour synthétiser des filtres à réponse impulsionnelle infinie, on utilisera la méthode de discrétisation d'un filtre analogique par la transformation bilinéaire. Là encore, les ordres des filtres n'étant pas connus à l'avance, on procédera par essais successifs pour déterminer l'ordre du filtre répondant au gabarit.

8. Déterminez le gabarit du filtre analogique à synthétiser.
9. Synthétisez le filtres passe-bas analogique normalisé de type Butterworth.
10. Transformez ce filtre afin que ses fréquences caractéristiques correspondent au gabarit.
11. Discrétisez ce filtre par transformation bilinéaire.
12. Pour l'ordre choisi, tracez la réponse en fréquence et le temps de propagation de phase correspondant.
13. Conclusions...

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 la commande `load fichsig.mat`.

Pour les deux filtres synthétisés précédemment :

1. Tracez une représentation temporelle du signal de référence et du signal filtré.
2. Expliquez les différences, en particulier pour l'origine des temps.
3. Tracez les représentations temporelles des signaux de référence et filtrés entre les instants $t = 400T_e$ à $t = 450T_e$ et $t = 750T_e$ à $t = 800T_e$. Quel est le retard observé ? Comment retrouve-t-on ce résultat à partir du temps de propagation de phase ?

V CHOIX DES FILTRES

Vous avez construit des filtres de structures différentes répondant à un objectif commun. A partir des résultats obtenus, quel type de filtres choisiriez vous dans chacun des cas suivants :

1. Temps de traitement minimal.
2. Déphasage entre les composantes fréquentielles inexistant.
3. Stabilité inconditionnelle du filtre.

Donnez vos conclusions sur la synthèse et le choix des filtres.

VI ANNEXE : FONCTION MATLAB freqz

FREQZ Digital filter frequency response.

[H,W] = FREQZ(B,A,N) returns the N-point complex frequency response vector H and the N-point frequency vector W in radians/sample of the filter :

$$H(e^{jw}) = \frac{B(e^{jw})}{A(e^{jw})} = \frac{b(1) + b(2)e^{-jw} + \cdots + b(m+1)e^{-jmw}}{a(1) + a(2)e^{-jw} + \cdots + a(n+1)e^{-jnw}}$$

given numerator and denominator coefficients in vectors B and A. The frequency response is evaluated at N points equally spaced around the upper half of the unit circle. If N isn't specified, it defaults to 512.

[H,W] = FREQZ(B,A,N,'whole') uses N points around the whole unit circle.

H = FREQZ(B,A,W) returns the frequency response at frequencies designated in vector W, in radians/sample (normally between 0 and pi).

[H,F] = FREQZ(B,A,N,Fs) and [H,F] = FREQZ(B,A,N,'whole',Fs) return frequency vector F (in Hz), where Fs is the sampling frequency (in Hz).

H = FREQZ(B,A,F,Fs) returns the complex frequency response at the frequencies designated in vector F (in Hz), where Fs is the sampling frequency (in Hz).

FREQZ(B,A,...) with no output arguments plots the magnitude and unwrapped phase of the filter in the current figure window.

See also FILTER, FFT, INVFREQZ, FVTOOL, and FREQS.