TP Filtres numériques

Octobre 2006
G. Baudoin, P. Bildstein

EXERCICES SUR LES CELLULES ÉLÉMENTAIRES DE FILTRAGE

Pour les filtres suivants, on étudiera la réponse impulsionnelle, le module et la phase de la fonction de transfert en fréquence, ainsi que le temps de propagation de groupe et le pôles et les zéros.

La réponse impulsionnelle peut être obtenue en filtrant une impulsion par la commande ***filter*** ou bien directement à l'aide de la commande ***impz***. On calculera seulement quelques dizaines de points de la réponse impulsionnelle des filtres IIR.

On normalisera la fréquence d'échantillonnage fe à 1.

On se limitera à Npts points pour le calcul de la fonction de transfert en fréquence.

Etude de la cellule d'ordre 2 purement récursive

$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$ avec a_1 = -1.0690 et a_2 = 0.5831

Le corrigé du début de cet exercice est donné en fin d 'énoncé de façon à vous permettre de prendre en main les principales commandes matlab pour l'étude des filtres.

Mesurer dans ce cas la fréquence de résonance, l'amplitude et la largeur de bande (à 3 dB) à la résonance. Comparer les résultats obtenus avec les valeurs théoriques approchées pour r proche de 1.
Constater le lien entre l'argument des pôles et la valeur de la fréquence de résonance.

Augmenter le module des pôles (en les gardant inférieurs à 1, on prendra par exemple r=0.99), tout en conservant la valeur de l'argument. Calculer les coefficients a_i correspondant à ces nouveaux pôles et observer comment évolue la résonance.

Changer le signe de a_1. Observer la transformation du filtre. Faites le lien avec une transformation de z en -z et ses conséquences dans le domaine fréquentiel.

Filtres FIR à phase linéaire

Tracer le module et la phase de la fonction de transfert en fréquence ainsi que le temps de propagation de groupe des 4 filtres FIR suivants.

$H_{il}(z) = 1 + 4z^{-1} + z^{-2}$

$H_{pl}(z) = 1 + 4z^{-1} + 4z^{-2} + z^{-3}$

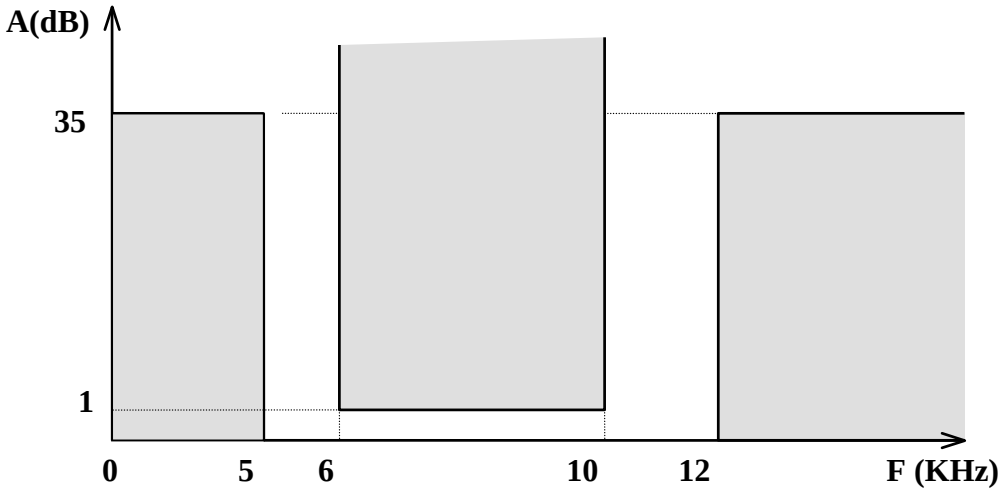
$H_{ia}(z) = 1 - z^{-2}$

$H_{pa}(z) = 1 + 4z^{-1} - 4z^{-2} - z^{-3}$

Commenter la valeur des temps de groupe.
Commenter la forme des fonctions de transfert (vérifier les propriétés vues en cours sur les 4 types de FIR à temps de propagation de groupe constant).

EXERCICE : CALCUL D'UN FILTRE NUMERIQUE IIR DERIVE D'UN FILTRE ANALOGIQUE

Calculer la fonction de transfert d'un filtre numérique IIR satisfaisant les contraintes de réponse en amplitude définies ci dessous.



Fonction de transfert avec une fréquence d'échantillonnage de 40KHz

- Trouver l'ordre minimal requis pour les différentes approximations:
  - ⇒ Butterworth: ***buttord***
  - ⇒ Tchebyshev type 1 et 2***:*** ***cheb1ord*** et ***cheb2ord***
  - ⇒ Filtre elliptique: ***ellipord***
- Calculer les quatre fonctions de transfert numérique (en z) par la méthode de la transformation bilinéaire à l'aide des fonctions ***cheby1, cheby2, ellip*** et ***butter*** convenablement paramétrées (Les programmes MATLAB permettent non seulement le calcul direct, mais aussi ils effectuent la prédistorsion du gabarit automatiquement).
- Visualiser sur le même système d'axes les quatre réponses en amplitude puis les quatre réponses en temps de groupe (à l'aide de la fonction ***grpdelay***).

.

Pour l'une de ces fonctions de transfert:

- Calculer et représenter la réponse impulsionnelle (fonction ***impz***) et repérer les pôles et zéros dans le plan complexe en z (fonction ***zplane***)

EXERCICES SUR LE CALCUL DE FILTRES NUMERIQUES FIR PAR LES METHODES DE LA FENETRE OU D'OPTIMISATION

Le but de ces exercices est de calculer plusieurs filtres numériques FIR, par la méthode de la fenêtre et par des méthodes d'optimisation.

MATLAB dispose de deux routines pour les méthodes empiriques : méthode de la fenêtre : **FIR1** et échantillonnage en fréquence **FIR2**.

MATLAB dispose de deux fonctions de calcul numérique direct de filtres FIR avec des algorithmes similaires mais des critères d'optimisation différents :
Fonction ***FIRLS***: minimisation des carrés (éventuellement pondérés)
Fonction ***REMEZ***: optimisation minimax (éventuellement pondérés)

Utilisation de la fonction *FIR1*: Filtres standards

1. Exercice préliminaire: tracer plusieurs fenêtres standards à l'écran, leur longueur étant de 50 points, avec la fonction *stem :*
   - ***Boxcar***
   - ***Hamming***
   - ***Kaiser*** avec β=1 et β=9

2. Calcul d'un simple filtre passe bas satisfaisant les contraintes suivantes:
   Bande passante de 0 à 6 Khz avec une ondulation maximale de 1dB
   Bande atténuée au delà de 10KHz avec une atténuation minimale de 35dB
   Fréquence d'échantillonnage Fs = 40khz.
   - Quelle est la valeur de l'argument d'entrée ***Wn***?
   - Calculer plusieurs filtres avec ***n***=10; ***n***=20; ***n***=50; et ***n***=100 et une fenêtre rectangulaire
   - Essayer la fenêtre par défaut (Hamming) au lieu de la fenêtre rectangulaire
   - Puis une fenêtre de Kaiser avec beta=21 et beta = 9
   - Vérifier sur quelques filtres que le temps de groupe est constant.

3. Calcul du filtre passe bande ayant les mêmes spécifications que celles définies dans l'exercice sur le calcul du filtre IIR.
   - Déterminer l'argument d'entrée ***Wn***
   - Essayer de calculer plusieurs filtres avec des valeurs croissantes de ***n*** et trouver la valeur minimale satisfaisant les spécifications.

Utilisation de la fonction *REMEZ*: optimisation minimax (éventuellement pondérés)

1. Filtre passe bas simple: calculer le filtre passe bas défini dans la question 2 de l'exercice avec la fonction FIR1 :
   - Déterminer les vecteurs d'entrée ***f*** et ***a***
   - Calculer plusieurs filtres d'ordre 20, 50 et 100 avec la fonction ***remez***
   - Tracer les réponses en amplitude de ces filtres dans la même fenêtre et comparer les résultats

2. Filtre passe bande classique: calculer le même filtre que celui de la question 3 ::
    - Déterminer les vecteurs d'entrée ***f*** et ***a***
    - Calculer plusieurs filtres d'ordre 20, 50 et 100 avec la fonction ***remez***
    - Tracer les réponses en amplitude de ces filtres dans la même fenêtre et comparer les résultats
    - Essayer d'améliorer le résultat du filtre d'ordre 50 en utilisant un vecteur de pondération
    - Utiliser la fonction r***emezord*** pour déterminer l'ordre nécessaire (consulter le ***help*** pour obtenir les informations sur cette fonction)

EXERCICE SUR L'IMPLANTATION D'UN FILTRE IIR EN PRECISION FINIE FORMAT FIXE

Le but de cet exercice est d'analyser pour un filtre IIR l'influence de la quantification des coefficients et des données en format fixe.

On reprend le filtre elliptique calculé dans l'exercice sur le calcul des IIR.

Etude de la quantification des coefficients

Cas de la structure directe

Etudiez la fonction de transfert obtenue en quantifiant les coefficients sur 12 et 16 bits en format fixe, et ceci pour la structure directe.

Pour quantifier un ensemble de coefficients en format fixe sur B bits, il faut :

- déterminer le plus grand coefficient en valeur absolue, pour en déduire le nombre de bits devant la virgule Be pour la partie entière,
- puis en déduire le nombre de bits derrière la virgule pour la partie fractionnaire Bf = B-Be.

Vous disposez pour ce travail de la commande ***round*** qui arrondit à l'entier le plus proche.
Ainsi on peut calculer la valeur xr, correspondant à x=1,2345 quantifiée sur 10 bits avec 2 bits devant la virgule, par la commande:

```
xr=round(1,2345*2^8)/2^8
ans =
    1.2343750000000
```

Pour chaque nombre de bits, vous calculerez les coefficients quantifiés puis vous tracerez la fonction de transfert correspondante et vous vérifierez si elle vérifie encore le gabarit.

Vous observerez par ailleurs la position des pôles et des zéros.

Remarque :
Pour les tracés en fréquence on prépare un vecteur gabarit, appelé gaba :
gaba=[35*ones(1,f1),NaN*ones(1,f2-f1-1),1*ones(1,f3-f2+1),NaN*ones(1,f4-f3-1),35*ones(1,f5-f4+1)];
où 1 et 35 correspondent aux 2 atténuations caractéristiques et où f1, f2, f3, f4, f5 sont les indices des fréquences caractéristiques sur la grille en fréquence (à Npts points) :
f1=round(5/20*Npts) ; f2=round(6/20*Npts); f3=round(10/20*Npts); f4=round(12/20*Npts); f5=Npts

Cas de la structure cascade

On pourra utiliser la fonction ***tf2sos*** (tf= transfer function, sos = second order sections) pour obtenir les différentes cellules de la structure cascade.

Chaque ligne de la matrice SOS donnée par la fonction matlab tf2sos comprend à 6 valeurs, correspondant aux coefficients du numérateur et du dénominateur d'une cellule.

SOS = [b01 b11 b21 1 a11 a21
b02 b12 b22 1 a12 a22
...
b0L b1L b2L 1 a1L a2L]

Chaque ligne correspond à la cellule d'ordre 2 de fonction de transfert Hk(z) pour la ligne k :

$$Hk(z) = \frac{b0k + b1k\, z^{-1} + b2k\, z^{-2}}{1 + a1k\, z^{-1} + a2k\, z^{-2}}$$

La fonction fournit de plus un gain G. $H(z) = G*H1(z)*H2(z)*...*HL(z)$

Etudiez les pôles et les zéros de chaque cellule.

Vérifier que l'appariement fait par Matlab correspond à associer les pôles avec les zéros les plus proches en commençant par le pôle le plus proche du cercle unité.

Vérifier que les cellules sont rangées en allant du pôle le plus près de 0 (ligne 1) vers le pôle le plus près du cercle unité (dernière ligne), c'est-à-dire par ordre de surtension croissante.

Etudier l'influence de la quantification des coefficients pour une structure cascade.

Le numérateur et le dénominateur de la fonction H(z) sont décomposés en un produit de facteurs d'ordre 1 ou 2. Et on quantifie les coefficients de ces facteurs.

Etudiez la fonction de transfert obtenue en quantifiant les coefficients d'une structure cascade sur 12 et 16 bits, en format fixe. Comparer ces résultats avec ceux obtenus pour la structure directe

Observer les pôles et les zéros de la fonction de transfert quantifiée pour la structure cascade.

Etude de la quantification des données, bruit de calcul en structure cascade

Dans cette partie du travail on étudie l'influence de la quantification des données en format fixe et on choisit de travailler avec la structure cascade (forme canonique DN).

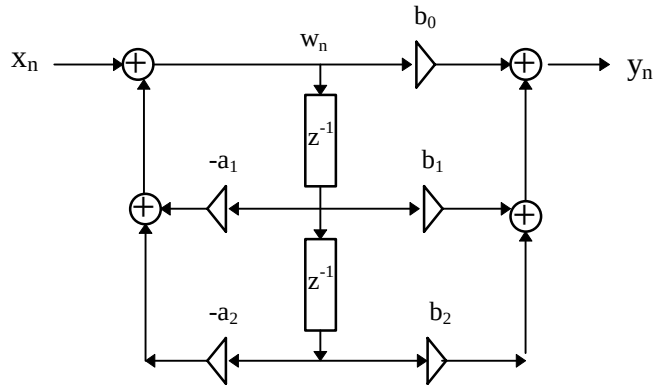
Pour une réalisation du filtre en structure cascade, il faut :

- Calculer les pôles et les zéros,
- Apparier les pôles et les zéros pour obtenir des cellules (sections) d'ordre 1 ou 2,
- Ordonner les cellules obtenues,
- Calculer les facteurs d'échelle.

Pour un filtre d'ordre 8, il y a 4 facteurs au dénominateur et au dénominateur. Il y a donc 4 ! = 24 façons d'apparier les facteurs du dénominateur avec ceux du dénominateur, et pour chaque appariement il y a 4 ! (ordonnancements possibles. On obtient donc 24x24 = 576 structures cascade possibles.

Le problème est de trouver la structure qui permet de minimiser la puissance du bruit de calcul en sortie du filtre dans une implantation cascade en format fixe avec des facteurs d'échelle calculés pour la norme $L\infty$ (pas de saturation pour les fréquences pures).

On supposera que chaque cellule est implantée par une structure DN et que la quantification est effectuée une seule fois avant la mise en mémoire de wn (voir figure).



La routine filt2 donnée ans ce TP effectue un filtrage d'ordre 1 ou 2 en structure DN avec une quantification de wn sur B bits. La routine suppose que les données (x, y, w) sont comprises entre +1 et –1.
Les arguments de cette routine sont :
[y,Cfinal]=filt2(b,a,x,B,Cini) ;
Cini est le vecteur ligne de conditions initiales [w(n-1), w(n-2)] et Cfinal est le vecteur ligne de conditions finales pour wn.

Structure cascade avec coefficients de surtension en ordre croissant
On étudiera la structure cascade en ordonnant les cellules par ordre de surtension croissante.
Calculer les facteurs d'échelle sc1, sc2, sc3, sc4 de façon à ce qu'il n'y ait pas de saturation pour une entrée purement sinusoïdale.
Calculer le dernier facteur d'échelle sc5 pour que G = sc1 x sc2 x sc3 x sc4 x sc5.
et en calculant les facteurs d'échelle correspondant. La routine ***tf2sos*** peut faire ce calcul.

Fabriquer une entrée test x à l'aide de la fonction ***rand*** :
x=(rand(1,5000)-0.5)*2 ;

Calculer la sortie y correspondant à cette entrée sans bruit de calcul avec la fonction Matlab ***filter***.

Calculer la sortie yq correspondant à cette entrée avec quantification des données sur 16 bits en utilisant la fonction ***filt2*** et en prenant en compte les facteurs d'échelle précédents.

Calculer le bruit de calcul en sortie en effectuant la différence entre y et yq.
Estimer la puissance moyenne de ce bruit de calcul.
Estimer la puissance moyenne de y.
Calculer le rapport signal sur bruit (en dB) obtenu en sortie.

Calculer la puissance du bruit en utilisant la formule vue en cours :

Structure cascade avec coefficients de surtension en ordre décroissant
Ordonner les 4 cellules par ordre de surtension décroissante.
Refaire les mêmes calcul que précédemment.

Comparer les résultats pour les 2 ordonnancements.

Corrigé du début de l'exercice sur la cellule IIR d'ordre 2 purement récursive

Etude de la cellule d'ordre 2 purement récursive

$$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$ avec $a_1 = -1.0690$ et $a_2 = 0.5831$

- Initialisation des vecteurs de coefficients du numérateur et du dénominateur:

b=1 ;

a=[1 -1.0690 0.5831] ;

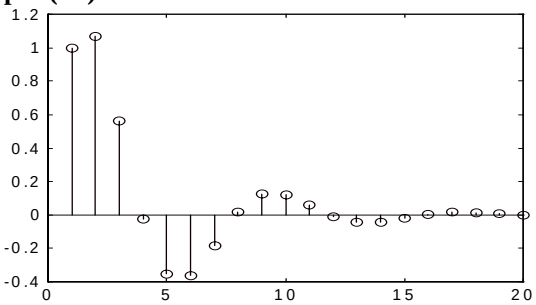
- Calcul de la réponse impulsionnelle

calcul de la réponse impulsionnelle à l'aide de la fonction ***impz*** :

hn=impz(b,a,20) ;

Le nombre 20 représente le nombre de points demandés pour la réponse impulsionnelle.

plot(hn)

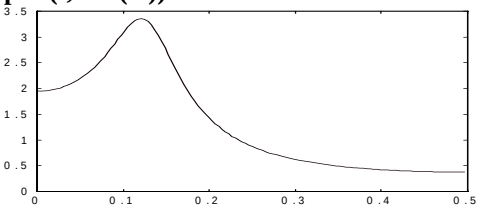


- Calcul de la fonction de transfert en fréquence : calcul de H(f) pour Npts valeurs de f entre 0 et fe/2 en normalisant fe à 1.

[H,f]=freqz(b,a,Npts,1) ;

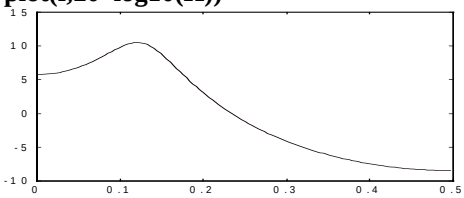
Tracé du module de H en linéaire

plot(f, abs(H))



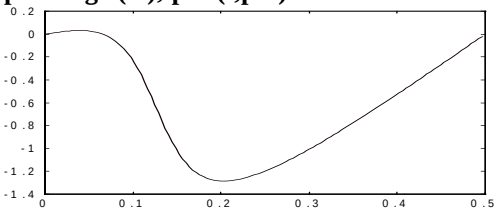
Tracé du module de H en dB

plot(f,20*log10(H))



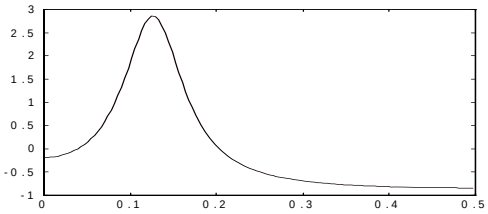
Calcul et tracé de la phase

phi=angle(H); plot(f,phi)



Calcul et tracé du temps de propagation de groupe

[tau,f]=grpdelay(b,a,Npts,1); plot(f,tau)



- Calcul et tracé des pôles et des zéros

poles=roots(a)

poles =

0.5345 + 0.5454i

0.5345 - 0.5454i

r=abs(poles)

r =

0.7636

0.7636

theta=angle(poles)

theta =

0.7954

-0.7954

zplane(0,poles)