POLYTECH SORBONNE Année 2017-2018
EISE3

TD4 sur machine : synthèse de filtres numériques

Objectifs pédagogiques

À la fin de cette séance, vous serez capable de synthétiser un filtre sous Matlab. Plus précisément, vous serez capable, sous Matlab, de :

- Synthétiser un filtre RIF passe-bas, passe-haut, passe-bande ou coupe-bande, étant données ses caractéristiques, par la méthode de la fenêtre ou par la méthode de Parks-McClellan/Remez
- Synthétiser un filtre RII passe-bas, passe-haut, passe-bande ou coupe-bande, étant données ses caractéristiques, par transformation homographique de filtres de Butterworth ou de filtres elliptiques
- Synthétiser un filtre dérivateur par méthode de filtre optimal RIF ou RII
- Vérifier un filtre (i.e. vérifier que le filtre synthétisé a bien les caractéristiques attendues)

De plus, vous remobiliserez votre capacité à :

- Synthétiser analytiquement un filtre RIF, étant données ses caractéristiques, par la méthode de la fenêtre
- Synthétiser analytiquement un filtre RII, étant données ses caractéristiques, par la méthode de la transformation homographique d'un filtre de Butterworth d'ordre 1, 2 ou 3

Propos

Le but de ce TP est de synthétiser des filtres numériques par quelques méthodes parmi les plus utilisées, à partir de spécifications spectrales. Un point important est de vérifier systématiquement vos filtres, car certaines méthodes sont itératives et peuvent ne pas converger vers une solution acceptable.

Reportez sur votre compte rendu toutes les remarques qui vous paraissent pertinentes sur le choix d'une méthode ou une autre, le choix de l'ordre du filtre, le comportement des méthodes et leur implémentation.

Le travail est à effectuer en binôme. Il est cependant souhaitable que chaque étudiant dispose de son propre compte rendu. Si vous le souhaitez, vous pouvez rendre votre compte rendu en fin de séance pour validation.

Conseils

Pour bien organiser votre séance, suivez les conseils ci-dessous :

- Lisez ce texte en entier
- Partagez-vous le travail au sein du binôme
- Dans Matlab, ouvrez un éditeur et commencez votre programme dedans. Écrivez un fichier de commande (par exemple à votre nom).
- En parallèle, commencez tout de suite la rédaction de votre compte rendu.
- Utilisez les pages de manuel des fonctions Matlab : tapez `help` ou `doc` suivi du nom de la fonction. Voici une liste de fonctions pouvant servir dans ce travail : `fdatool`, `fir1`, `firpm`, `firpmord`, `butter`, `buttord`, `ellip`, `ellipord`, `hann`, `hamming`, `fvtool`, `impz`, `stepz`, `freqz`, `phasez`, `zplane`, `grpdelay`, `fft`, `filter`, `filtfilt`, `cos`, `sin`, `log10`, `abs`, `sum`, `max`, `round`, `ceil`, `length`, `size`, `zeros`, `ones`, `figure`, `plot`, `stem`, `pause`, `audioread`, `soundsc`, `load`.

Travail à réaliser

Synthèse d'un passe-bas

Synthétiser un filtre consiste à générer les coefficients du filtre qui répondent à des spécifications données le plus souvent en terme de gabarit fréquenciel. Le type de gabarit (gabarit cible ou gabarit par contrainte) dépend de la méthode de synthèse. Cette partie vise à comparer les différentes méthodes de synthèse. Pour cela, dans toute cette partie, le filtre à synthétiser sera un filtre discret passe-bas de fréquence de coupure $f_d = 2000Hz$ pour une fréquence d'échantillonnage de $F_e = 16000Hz$, avec une atténuation d'au moins 50dB en bande atténuée, bande qui commence en $2500Hz$. Pour chaque méthode vous devrez systématiquement vérifier les caractéristiques de votre filtre avec les outils d'analyse, par exemple avec `freqz`. Il est aussi conseillé de vérifier l'effet du filtrage sur un signal audio.

Transformation homographique d'un Butterworth

L'un des filtres les plus couramment employés est le passe-bas obtenu par transformation homographique d'un filtre de Butterworth. Pour une fréquence de coupure du filtre discret f_d donnée en Hertz, la méthode consiste à déterminer la constante K de transformation homographique puis à appliquer cette transformation au filtre normalisé de Butterworth.

En pratique, Matlab dispose de la fonction `butter` qui fournit directement les coefficients du filtre discret en fonction de la fréquence de coupure f_d désirée, exprimée en fréquence relative par rapport à $F_e/2$ (Rappel : les fréquences réduites dans Matlab sont des pulsations réduites divisées par π, et valent donc $2f/F_e$ où f est la fréquence en Hz).

Q1. : Synthétisez le filtre à partir d'un filtre de Butterworth d'ordre 3. Vérifiez votre filtre avec `freqz`. Pour quelle fréquence l'atténuation est-elle de 50dB ?

Visiblement l'ordre du filtre n'est pas suffisant pour satisfaire les spécifications de largeur de bande de transition. En pratique, dans Matlab, la fonction `buttord` fournit l'ordre minimum vérifiant des spécifications données.

Q2. : Utilisez `buttord` pour déterminer l'ordre du filtre de Butterworth tel que l'atténuation à 2500Hz soit de 50dB. Vérifiez ces caractéristiques avec `freqz`.

Filtre elliptique

Si on libère la contrainte de monotonie de la réponse en fréquence, l'ordre (ou la largeur de bande) peut considérablement être réduit par l'usage d'un filtre elliptique.

Q3. : Synthétisez le filtre à partir d'un filtre elliptique, en choisissant l'ordre tel que l'atténuation à 2500Hz soit de 50dB et tel que les oscillations en bande passante ne dépassent pas 0,1dB (utilisez `ellip` et `ellipord`). Vérifiez ces caractéristiques avec `freqz`. Comment le filtre elliptique se compare-t-il au précédent ? Observez notamment l'atténuation à f_d.

Filtre RIF par méthode de fenêtrage

Lorsque le filtre causal doit être à phase linéaire, il faut utiliser un filtre RIF. La méthode de fenêtrage est la plus simple à mettre en oeuvre. Pour l'implémenter, soit vous pouvez programmer le calcul du sinus cardinal et le multiplier par la fenêtre choisie, soit vous pouvez utiliser la fonction `fir1` qui est prévue pour cela dans Matlab.

Obtenir la largeur de bande désirée peut se réaliser en 2 temps : faites d'abord un premier essai avec une valeur arbitraire (par exemple une trentaine de points), déterminez la fréquence de début de

bande atténuée (par exemple avec `freqz`) ; ensuite, déduisez-en la taille correcte sachant que la largeur de bande est inversement proportionnelle à la taille, puis synthétisez le filtre désiré.

Q4. : Synthétisez le filtre par la méthode de fenêtrage, en utilisant une fenêtre de Hamming. Choisissez la taille de la RI pour que l'atténuation de la bande atténuée soit atteinte en 2500Hz. Vérifiez votre filtre. Quelle est la valeur de l'atténuation à f_d ?

Filtre RIF optimal

L'utilisation d'un RIF optimal permet de réduire l'ordre du filtre. La fonction Matlab correspondante est `firpm`, dans laquelle il faut spécifier le gabarit par contrainte. La détermination de l'ordre du filtre peut se faire par `firpmord`.

Q5. : Synthétiser le filtre par la méthode du RIF optimal ayant moins de 0,1dB d'oscillations en bande passante et 50dB d'atténuation au dessus de 2500 Hz. Vérifiez votre filtre. Quelle est l'ordre du filtre ?

Autres types de filtre

Passe-haut

Q6. : Reprenez chaque méthode une par une pour synthétiser un passe-haut de même fréquence de coupure $f_d = 2000Hz$ avec une bande atténuée allant de 0Hz à $1500Hz$ et ayant les mêmes caractéristiques d'atténuation et d'oscillation en bande passante.

Passe-bande

Q7. : Choisissez une méthode pour synthétiser un filtre passe-bande téléphonique, donc de fréquences de coupure 300Hz et 3400Hz pour une fréquence d'échantillonnage de 8000Hz. Vous ferez en sorte que l'oscillation en bande passante soit inférieure à 1dB et l'atténuation en bande atténuée supérieure à 50dB.

Dérivateur

La dérivation numérique peut se réaliser très simplement par différence finie, comme par exemple par l'approximation symétrique : $\frac{dx}{dt}(kTe) \simeq \frac{1}{2T_e}(x(k+1) - x(k-1))$ Cette approximation peut être vue comme un filtre discret. Pour réaliser un filtre dérivateur ayant une réponse en fréquence plus proche de la réponse en fréquence théorique $(j2\pi f)$, Matlab prévoit par exemple la fonction `firpm` avec l'option `'d'`.

Q8. : Synthétisez un filtre dérivateur en utilisant `firpm` avec un ordre d'une cinquantaine de points. Que vaut-il mieux, un ordre pair ou impair ? Justifiez. Fixez le gabarit de la façon suivante : `b = G*firpm(N,[0 0.9],[0 0.9],'d');` où `G` est la pente de la réponse en fréquence, à déterminer. Filtrez un signal test pour vérifier que ce filtre dérive bien en fonction du temps (par exemple : $(1-t^2)sin(2\pi 100t)$ avec $F_e = 16000$).

Utilisation de fdatool

Matlab prévoit un outil de synthèse de filtres par GUI obtenu en tapant : `fdatool()`. Il suffit ensuite de spécifier les caractéristiques des filtres puis de lancer la synthèse.

Q9. : Synthétisez le filtre de la question Q7 par la méthode de votre choix en utilisant `fdatool()`. Exportez votre filtre pour pouvoir l'appliquer sur un signal audio.