Cours de traitement des signaux biomédicaux
1ère séance Matlab

Commandes utiles

plot : graphe d'un signal (points reliés).
subplot : permet d'avoir plusieurs graphes sur la même figure.
stem : graphe d'un signal (style « numérique »).
freqz : réponse en fréquence d'un filtre.
filter : calcul de la sortie d'un filtre.
filtfilt : calcul de la sortie d'un filtre avec application dans les deux sens du temps.
fir1 : conception d'un filtre à réponse impulsionnelle finie.
buttord, cheb1ord, cheb2ord, ellipord: determination ordre des filtres
butter, cheby1, cheby2, ellip: synthèse des filtres numériques.
resample : ré-échantillonnage d'un signal.
hilbert : calcul du signal analytique.
unwrap : déballage de la phase d'un signal complexe.

Commandes supplémentaires

filtre_hilbert : génération d'un filtre de Hilbert RIF (TF inverse et fenêtre de Hamming).
diffe : génération d'un filtre différentiateur RIF (TF inverse et fenêtre de Hamming).

Remarques importantes

- La convention utilisée par Matlab pour la conception de filtres n'est pas d'aller de 0 à 0.5 en fréquences normalisée, mais d'aller de 0 à 1 en portion de π en radians pour la pulsation (fréquence multipliée par 2π). C'est l'un des seuls défauts de Matlab. Si l'on veut par exemple avoir une fréquence de coupure normalisée de 0.2 pour un filtre passe-bas, il faut donc rentrer un paramètre w = 2*0.2 = 0.4
- La routine freqz demande le nombre de valeurs de fréquence pour lesquelles la réponse en fréquence sera calculée. Mettez-en au moins 1000, ça ne coûte pas cher.
- La routine freqz utilise les vecteurs de coefficients du numérateur b et du dénominateur a de la fonction de transfert du filtre. Si l'on a un filtre à réponse impulsionnelle finie, il faut prendre pour b la réponse impulsionnelle du filtre, et prendre a = 1. Même remarque pour filter.
- Si avec freqz on ne met pas de variables de sortie, un graphe de la réponse en amplitude (en dB) et de la réponse en phase du filtre est automatiquement généré, bien sûre avec la convention Matlab mentionnée précédemment.
- Si l'on veut visualiser par exemple la réponse en amplitude d'un filtre pour les fréquences normalisées, le plus simple est d'utiliser les commandes :

```
>> [h,w]=freqz(b,a,1000) ;
>> plot(linspace(0,0.5,1000,abs(h))
```

- Pour génerer un signal sinusoïdal de fréquence f et de longueur n :

```
>> x=sin(2*pi*f*(1:n)) ;
```

- Dans fir1, si l'on donne un ordre n, la longueur du filtre est n+1.

Expérience 1 : du transitoire associé aux filtres RIF, mais de l'importance d'avoir un filtre à phase linéaire

Le signal dans **ecg.dat** contient plusieurs secondes d'un ECG échantillonné à 500 Hz. Les petites oscillations visibles principalement entre les complexes PQRST sont dues à la présence d'une interférence à 50 Hz, qu'on peut supprimer avec un filtrage passe-bas, mais en essayant de ne pas déformer les complexes.

Les commandes suivantes vont vous permettre de réaliser un filtre RIF à phase linéaire de longueur 101 et de fréquence de coupure 40 Hz (donc 0.08 en fréquence normalisée) et de visualiser sa réponse (2000 valeurs de fréquence). Vérifiez pourquoi.

```
>> g = fir1(100,0.16) ;
>> freqz(g,1,2000) ;
```

Les commandes suivantes vont vous permettre de réaliser un filtre RII de Butterworth de fréquence de coupure 40 Hz (donc 0.08 en fréquence normalisée), avec une bande de transition entre 35 Hz et 45 Hz, un taux d'oscillation en bande passante de 0.5 dB, et une atténuation de 20 dB, et de visualiser sa réponse (2000 valeurs de fréquence). Vérifiez pourquoi.

```
>> [N,Wn] = buttord(0.14,0.18,0.5,20) ;
>> [b,a] = butter(N,Wn) ;
>> freqz(b,a,2000) ;
```

Vous pouvez vérifier que les deux filtres ont une réponse en amplitude très semblable en les représentant sur le même graphe. Mais appliquez les tous à un signal sinusoïdal de longueur 150 et de fréquence normalisée 0.05, et représentez les sorties simultanément. Quel filtre a le transitoire le plus court ?

Vous pouvez alors filtrer le signal d'ECG avec les deux filtres :

```
>> ecg1 = filter(g,1,ecg) ;
>> ecg2 = filter(b,a,ecg) ;
```

et opérer un filtrage avant-arrière :

```
>> ecg3 = filtfilt(b,a,ecg) ;
```

Représentez avec subplot le signal initial et au dessous les différentes sorties.

1.1 Voit-on bien la suppression de la perturbation à 50 Hz ?

1.2 Lequel des deux filtres déforme-t-il le moins les complexes QRS ?

1.3 Y a-t-il encore distorsion des complexes avec le filtrage avant-arrière ?

Expérience 2 : conception de filtre de calcul de signal dérivé par transformation de Fourier inverse – utilisation pour l'extraction d'une fréquence instantanée

Le fichier **mersurepoutre.dat** contient un signal, échantillonné à 1000 Hz, correspondant à l'oscillation générée sur une « micro-poutre » par une culture de cellules cardiaques se synchronisant et s'excitant donc en même temps de manière périodique. Réalisez les opérations suivantes :

1. Sous-échantillonnage du signal d'un facteur 100 avec resample (les fréquences intéressantes sont beaucoup plus petites que 1000 Hz, pas la peine d'avoir autant d'échantillons).

2. Filtrage passe-bas du signal avec filtfilt et un filtre RII généré par :

```
>> [N,wn]=cheb1ord(0.06,0.1,0.5,20);
>> [b,a] = cheby1(N,0.5,wn);
```
Vous pouvez vérifier en représentant entrée et sortie sur le même graphe que filtfilt n'introduit pas de déphasage, alors qu'un filtrage classique par filter le fait.

3. Calcul du signal analytique du signal filtré en utilisant la routine Matlab hilbert. Vous pouvez vérifier en représentant signal et module du signal analytique sur le même graphe qu'on obtient bien l'enveloppe du signal.

4. Calcul de la phase instantanée « déballée » du signal. Si le signal analytique s'appelle xa, la phase déballée phi s'obtient avec :
```
>> phi  = unwrap(angle(xa));
```
Finalement, il ne reste plus qu'à extraire la fréquence instantanée en dérivant la phase instantanée avec le filtre généré par diffe, et en divisant le résultat par 2π. Utilisez un filtre de longueur 11, et compensez le retard induit par le filtre causal. Vous pouvez vérifier que la réponse impulsionnelle est anti-symétrique, du fait que la réponse en fréquence est imaginaire pure.

Expérience 3 : conception de filtre de Hilbert par transformation de Fourier inverse

Le fichier **accel.dat** contient un enregistrement d'un signal (fréquence d'échantillonnage 40 Hz) provenant d'un accéléromètre placé sur le poignet d'un soignant hospitalier. Le but est de repérer les épisodes de désinfection (produit sur les mains et frottement). On sait que dans l'enregistrement une désinfection a lieu entre les indices 900 et 1100 environ.

1. Pour détecter l'épisode de désinfection concevons un filtre passe-bande avec:

```
>> [N,Wn]=buttord([0.14,0.16],[0.13 0.17],0.5,20);
>> [b1,a1]=butter(N,Wn);
```

Comment l'inspection de la représentation graphique du signal justifie-t-elle le type du filtre, et le choix de sa fréquence centrale? Pourquoi filtfilt doit-il être utilisé pour filtrer le signal?

2. La puissance instantanée de la sortie du filtre peut ensuite être estimée en utilisant un filtre passe-bas conçu avec:

```
>> [N,Wn]=buttord(0.04,0.02,0.5,20);
>> [b2,a2]=butter(N,Wn);
```

et en filtrant la sortie du filtre au carré (pourquoi?), en utilisant bien sûr filtfilt de nouveau.

3. Répétez les mêmes opérations avec un filtre passe-bande de bande passante trois fois plus large, avec une région de transition deux fois plus large. Commentez les mérites relatifs des deux approches.