

Cours d'analyse informatique des données

Séance de laboratoire 3

Commandes utiles

plot : graphe d'un signal (points reliés).

subplot : permet d'avoir plusieurs graphes sur la même figure.

Commandes supplémentaires

filtrage_RII : filtrage d'un signal avec un filtre RII

banc_de_filtres : filtrage d'un signal cardiovasculaire avec 3 filtres correspondant aux bandes VLF, LF, et HF.

reponse_amplitude : réponse en amplitude d'un filtre

Conseils

- Pour générer une sinusoïde de longueur $L=100$, (commençant à l'indice zéro) et de fréquence normalisée $f=0.2$, vous pouvez faire :

```
>> L = 100 ; f = 0.2 ;
```

```
>> n = 0 :L-1 ;
```

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

- Pour filtrer un signal x avec un filtre RII de paramètres a et b et avoir la sortie dans y :

```
>> y = filter(b,a,x) ;
```

- Et pour représenter entrée et sortie ensemble :

```
>> plot(n,x,n,y,'r')
```

Expérience 1 : pas le même décalage à toutes les fréquences

Les commandes suivantes vont vous permettre de réaliser un filtre RII de type Chebyshev de fréquence de coupure normalisée 0.23 :

```
>> [N,Wn] = cheb1ord(0.44,0.48,0.5,20);
```

```
>> [b,a] = cheby1(N,0.5,Wn) ;
```

Vous pouvez visualiser la réponse en amplitude du filtre avec `reponse_amplitude`.

Générez une sinusoïde de longueur $L = 100$ et de fréquence 0.05, et faites la passer dans le filtre. Représentez entrée et sortie ensemble, et estimez le nombre d'échantillons de décalage entre entrée et sortie.

Faites la même opération pour une sinusoïde de fréquence 0.2 (attention, il y a plus d'une période de décalage). Conclusion ?

Expérience 2: filtrage d'un électrocardiogramme

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 RII de type Butterworth et de fréquence de coupure 40 Hz (donc 0.08 en fréquence normalisée) :

```
>> [N,Wn] = buttord(0.14,0.18,0.5,20) ;
```

```
>> [b,a] = butter(N,Wn) ;
```

3. Filtrez `ecg` avec `filtrage_Rll`. Vérifiez que l'interférence disparaît bien, mais que la sortie est retardée. **De plus observez la déformation des complexes QRS à leur fin en sortie.** La compensation de déphasage fait-elle bien son boulot ? La déformation est-elle encore présente ?

Expérience 3: extraction des composantes VLF, LF, et HF d'un signal cardiovasculaire

Le fichier `heart_1.dat` contient 3 colonnes, correspondant à 3 signaux acquis simultanément à une fréquence d'échantillonnage de 4 Hz. La première colonne correspond aux intervalles RR (intervalles entre battements cardiaques, en millisecondes), la seconde est la pression artérielle (en mm de mercure) et la troisième le volume pulmonaire instantané (respiration, unités arbitraire, en fait même signal que dans le labo 1).

Une fois le fichier chargé avec `load`, récupérez les intervalles RR avec :

```
>> RR = heart_1(:,1);
```

et enlevez la valeur moyenne :

```
>> RR = RR - mean(RR);
```

Pour cette expérience, il faut aussi sous-échantillonner le signal d'un facteur 2 (la fréquence d'échantillonnage devient 2 Hz) avec :

```
>> RR = resample(RR,1,2);
```

1. Filtrez avec `banc_de_filtres`. Observez les trois sorties de filtre pour repérer dans le signal initial ce qui leur correspond (surtout les bandes LF et HF).

2. Vous pouvez refaire l'opération avec les données de `heart_4.dat` (mêmes colonnes, même fréquence d'échantillonnage). Le sujet est à 4000 mètres d'altitude et vous pouvez vérifier que la respiration, présente un aspect bizarre. Il s'agit de ce qu'on appelle un syndrome de Cheyne-Stoke dû à l'altitude. Quelle est la différence marquante dans la bande LF ?