

TP3 - Traitement d'un signal ECG



Fait Par

BENDIDI OUMAIMA

Introduction

Le traitement d'un signal ECG consiste à améliorer la qualité et la fiabilité des informations contenues dans le signal. Il est utilisé pour diagnostiquer les troubles cardiaques et pour surveiller l'état de santé cardiaque d'une personne. Le traitement d'un signal ECG peut inclure plusieurs étapes telles que le filtrage pour éliminer le bruit et les interférences, l'amélioration de la précision des mesures, l'extraction des caractéristiques et la segmentation du signal. Il peut également inclure des méthodes de visualisation pour rendre le signal plus compréhensible. Les méthodes de traitement d'un signal ECG sont nombreuses et varient en fonction des besoins spécifiques de chaque application. Il est important de choisir les méthodes appropriées pour obtenir des résultats précis et fiables.

Dans ce TP, on travaille sur l'amélioration du rapport signal sur bruit pour le signal ECG (Electrocardiogram). Le signal ECG est utilisé pour mesurer l'activité électrique du cœur, mais peut être affecté par des parasites provenant de l'activité musculaire extracardiaque du patient. On explorera différentes méthodes pour réduire le bruit dans les signaux, telles que la limitation de la bande passante et le filtrage du signal. Ensuite, on utilisera la fonction d'autocorrélation pour identifier la fréquence cardiaque. On écrira un programme pour effectuer automatiquement ce calcul et on vérifiera si le programme trouve le bon pouls.

Objectifs:

Trouver un compromis entre la préservation de la forme du signal ECG et la réduction du bruit.

Utiliser la fonction d'autocorrélation pour identifier la fréquence cardiaque.

Écrire un programme pour effectuer automatiquement le calcul d'autocorrélation et trouver la fréquence cardiaque.

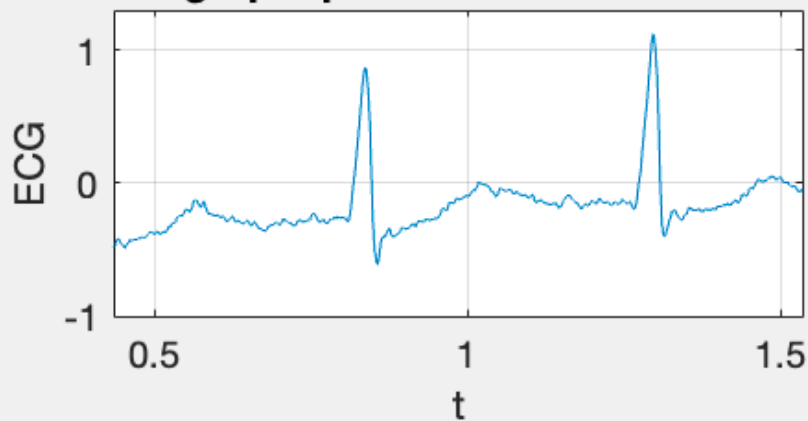
Vérifier si le programme trouve le bon pouls.

Suppression du bruit provoqué par les mouvements du corps

```
clear all
close all
clc

load('ecg.mat')
Fe=500;
Te=1/Fe;
N=length(ecg);
t = 0:Te:(N-1)*Te;
subplot(3,2,1)
plot(t,ecg)
grid on
title(" représentation graphique de l'activation électrique du cœur")
xlabel("t")
ylabel("ECG")
xlim([0.5 1.5]);
```

représentation graphique de l'activation électrique du cœur



Ce signal a été échantillonné avec une fréquence de 500Hz représenté en fonction de temps

On remarque que le signal est bruité:

- Les bruits de hautes fréquences sont provoqués par l'activité extracardiaque et les interfaces dues aux appareils électriques.
- Pour les bruits de basses fréquences sont provoqués par les mouvements du corps liés à la respiration

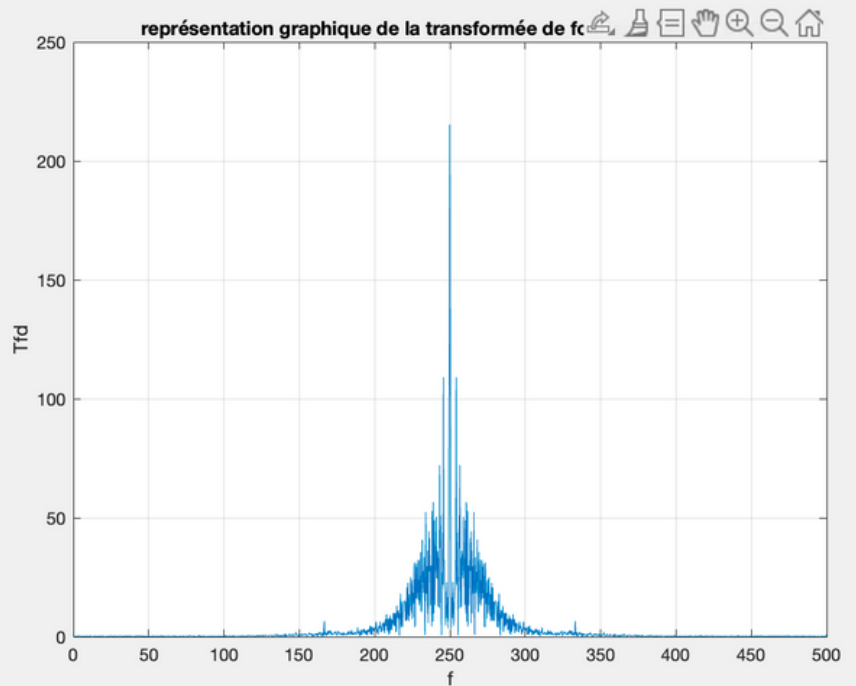
Pour supprimer les bruits à très basse fréquence dues aux mouvements du corps, on utilisera un filtre idéal passe-haut.

Pour ce faire, calculer tout d'abord la TFD du signal ECG, régler les fréquences inférieures à 0.5Hz à zéro, puis effectuer une TFDI pour restituer le signal filtré.

```

%%
f=(0:N-1)*(Fe/N);
fshift=(-N/2:N/2-1)*(Fe/N);
y = fft(ecg);
subplot(3,2,2)
plot(f,fftshift(abs(y)));
grid on
title(" représentation graphique de la transformée de fourier du signal ECG")
xlabel("f")
ylabel("Tfd")

```



```

% conception du filtre
pass_haut=ones(size(ecg));
fc=0.5;
index_fc= ceil((fc*N)/Fe);
pass_haut(1:index_fc)= 0;
pass_haut(N-index_fc+1:N) = 0;
% subplot(3,2,1)
plot(f,pass_haut,"linewidth",1.5)

%% Filtrage

ecg1_freq = pass_haut.*y;
ecg1 = ifft(ecg1_freq,"symmetric");

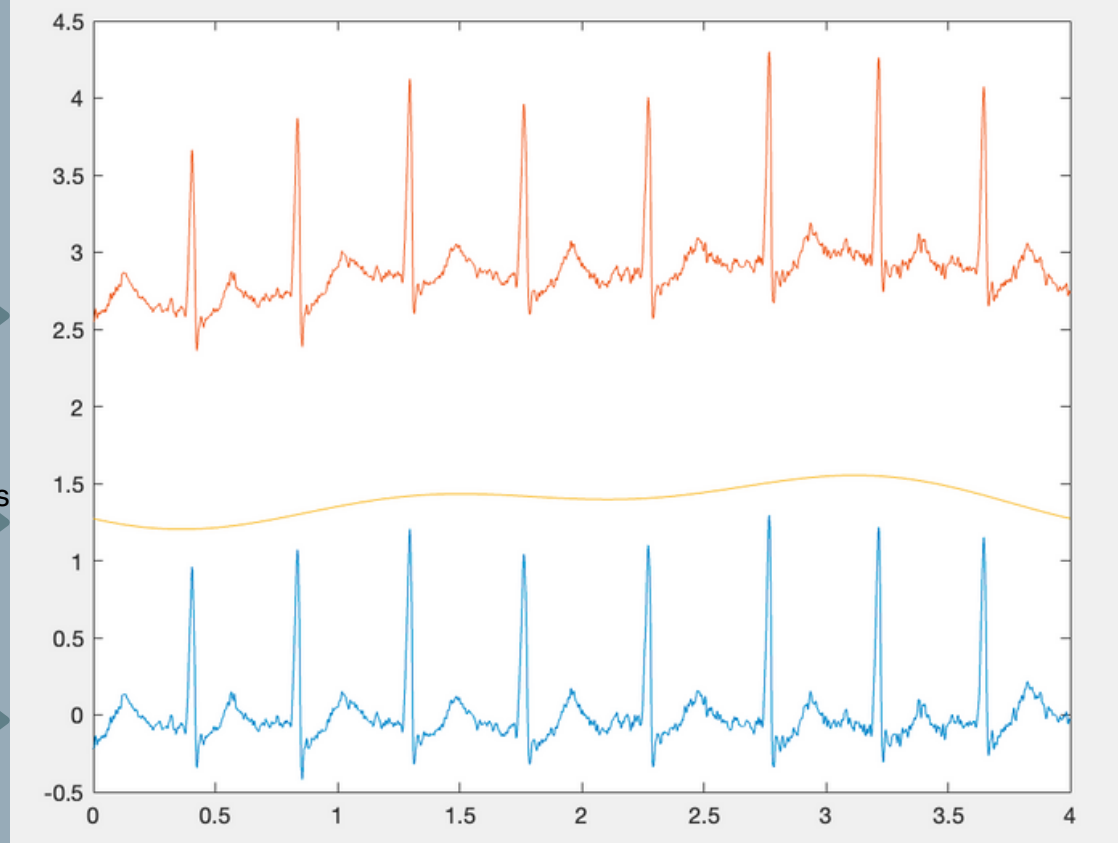
%% plot filtered signal
% subplot(3,2,2)
plot(t,ecg1)
hold on
plot(t,ecg+3)
hold on
plot(t,ecg-ecg1+1.5)

```

Signal avec bruit avant
le filtrage passe-haut

bruit de basses fréquences

Signal après le filtrage
passe-haut



Il est observé que les fréquences basses (inférieures à 0.5 Hz) ont créé un bruit qui a donné une ondulation dans le signal ECG, comme cela est visible dans la figure ci-dessus.

Suppression des interférences des lignes électriques 50Hz

```
%% Filtrage fréquence 50%

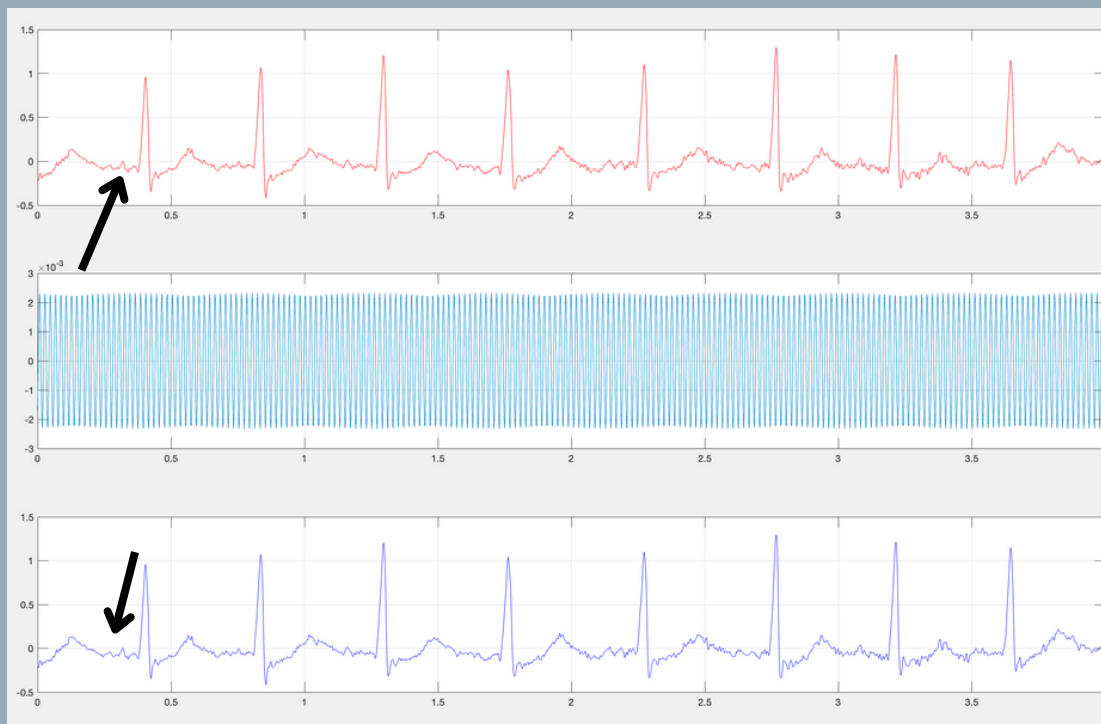
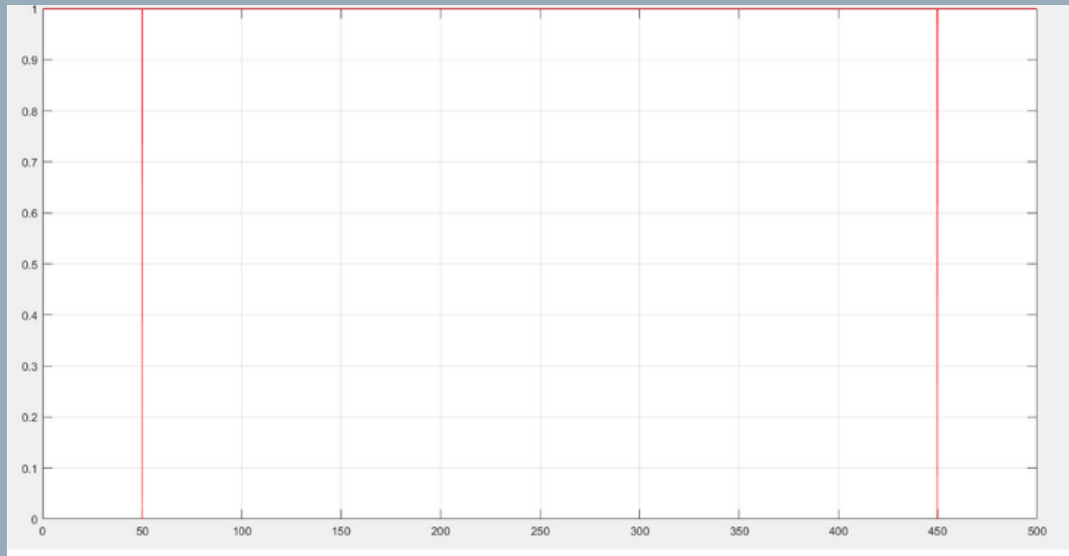
pass_notch=ones(size(ecg));
%crée un vecteur qui a la même taille que le signal ECG et est rempli de 1.
fc2=50;
index_fc2= ceil((fc2*N)/Fe)+1;
%calculé l'index de fréquence 50 Hz dans
% la représentation de domaine fréquentiel du signal ECG en utilisant la fréquence d'échantillonnage
pass_notch(index_fc2)= 0;
pass_notch(N-index_fc2+1) = 0;

%% Filtrage 2

ecg2_freq = pass_notch.*fft(ecg1);
%on recupere le signal dand le domaine temporel
ecg2 = ifft(ecg2_freq,"symmetric");

subplot(2,1,1)
plot(t,ecg)
subplot(2,1,2)
plot(t,ecg-ecg2)
```

Pour le filtre top_notch qu'on a utiliser pour éliminer des fréquences causées par des interférences des lignes électriques



Signal avec bruit avant le filtrage notch

le bruit due à la fréquence 50hz

Signal le filtrage notch

On peut voir que le bruit causé par les interférences des lignes électriques à 50Hz, entraîne de petites perturbations dans notre signal, comme cela est clairement visible dans les figures ci-dessus.

Amélioration du rapport signal sur bruit

Le signal ECG est également atteint par des parasites en provenance de l'activité musculaire extracardiaque du patient. La quantité de bruit est proportionnelle à la largeur de bande du signal ECG. Une bande passante élevée donnera plus de bruit dans les signaux, et limiter la bande passante peut enlever des détails importants du signal.

%% Filtrage pass_bas

```
pass_bas=zeros(size(ecg));  
fc3 = 15;  
index_fc3 = ceil((fc3*N)/Fe);  
pass_bas(1:index_fc3)= 1;  
pass_bas(N-index_fc3+1:N) = 1;
```

%% Filtrage 2

```
ecg3_freq = pass_bas.*fft(ecg2);  
ecg3 = ifft(ecg3_freq,"symmetric");
```

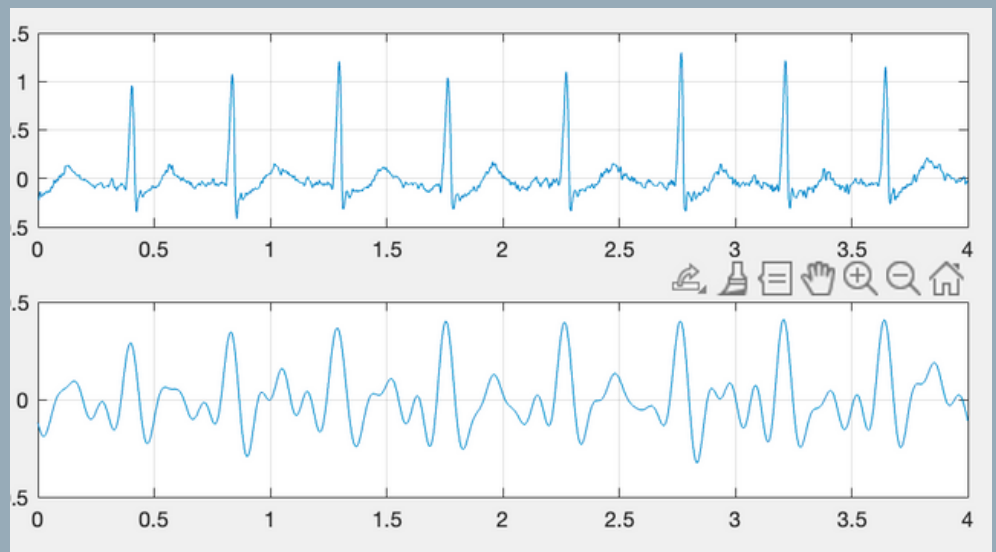
```
subplot(313)  
plot(t,ecg3)  
grid on  
subplot(311)  
plot(t,ecg)  
grid on  
subplot(312)  
plot(t,ecg2)  
grid on
```

On a essayé 3 différentes fréquences de coupure 10,15 et 20

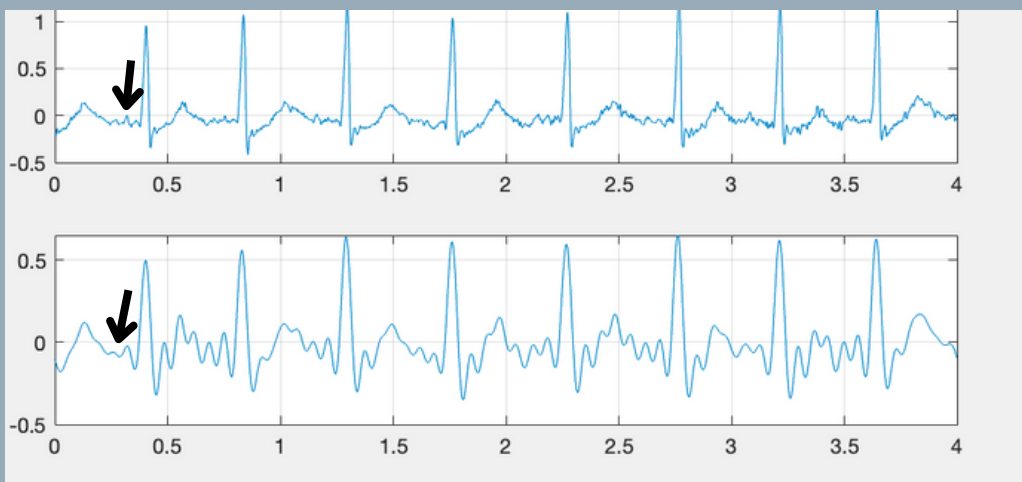
Signal après le filtrage
passe notch



Signal après le filtrage
passe-bas avec fc=10Hz



On peut bien remarquer la perte de l'information
utile avec une fréquence de coupure de 10 hz



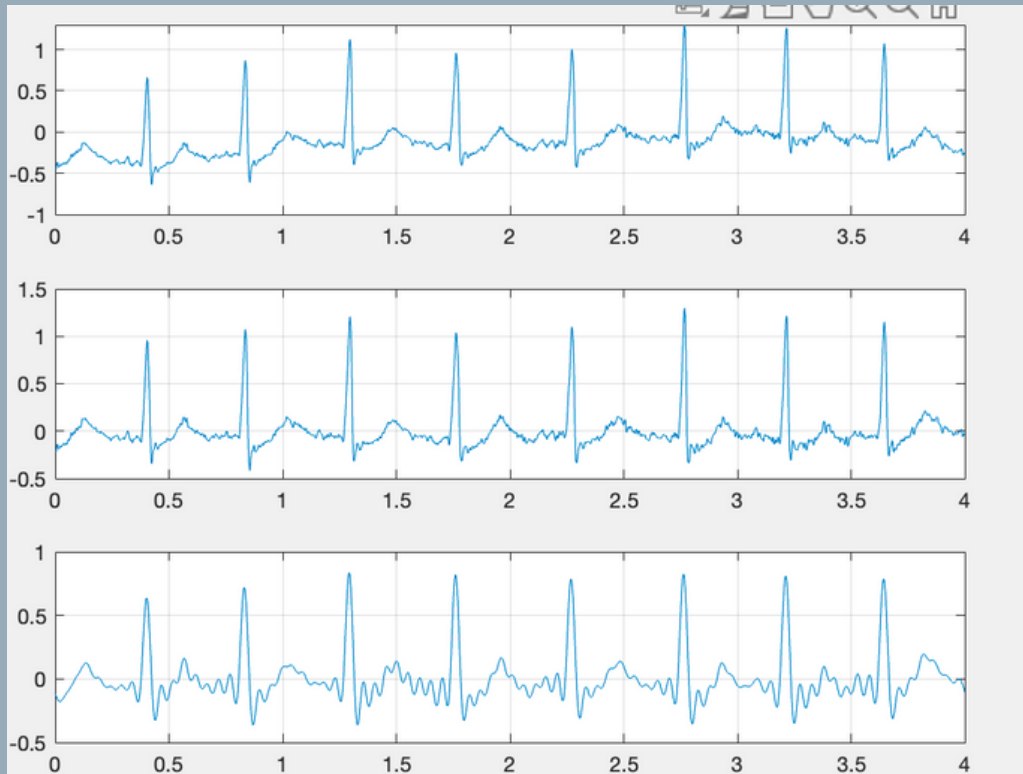
Signal après le filtrage
passe notch



Signal après le filtrage
passe-bas avec fc=15Hz



Avec une fréquence de 15 Hz on a aussi perdu de l'information
comme on peut le voir au niveau des piques

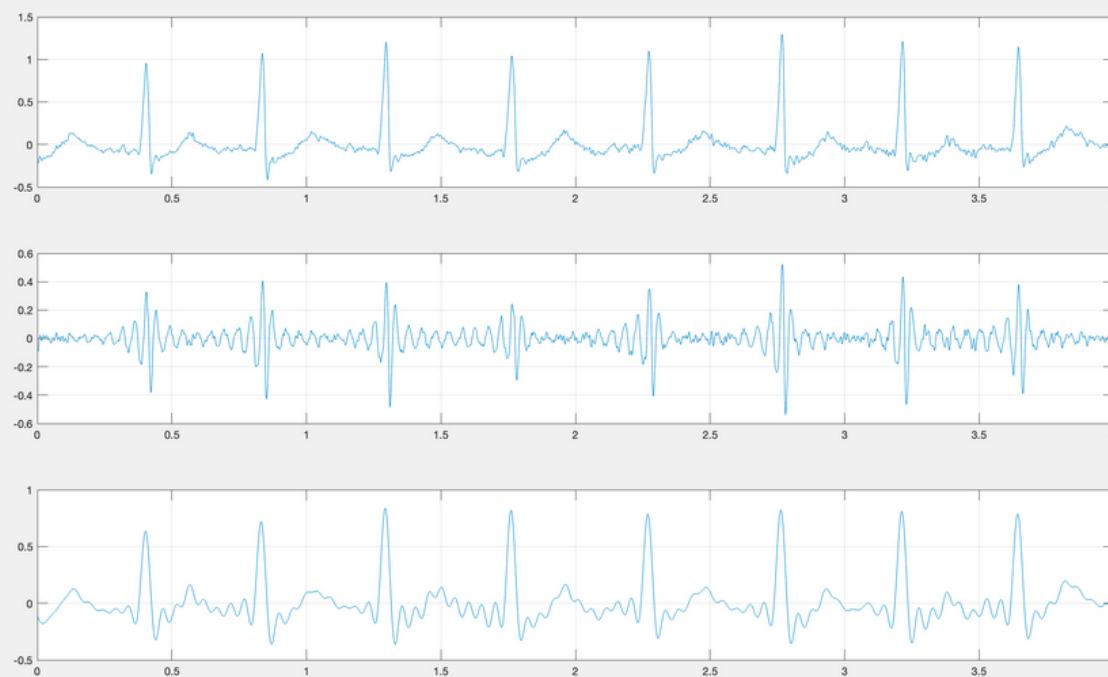


Notre premier signal
avec bruit

Signal après le filtrage
passe notch

Signal après le filtrage
passe-bas avec $f_c=20\text{Hz}$

Pour une fréquence de coupure de 20 Hz on a pu préserver l'information et de supprimer le bruit le maximum possible. On peut conclure que la fréquence 20 Hz est la meilleure qu'on a pu avoir pour supprimer du bruit sans perdre de l'information utile.



Signal après le filtrage
passe notch

le bruit due à la haute
fréquence

Signal après le filtrage
passe-bas avec $f_c=20\text{Hz}$

On a avoir ce résultat avec le code ci-dessous:

%% Filtrage 2

```
ecg3_freq = pass_bas.*fft(ecg2);  
ecg3 = ifft(ecg3_freq,"symmetric");  
  
subplot(313)  
plot(t,ecg3)  
grid on  
subplot(311)  
plot(t,ecg2)  
grid on  
subplot(312)  
plot(t,ecg2-ecg3)  
grid on
```

Identification de la fréquence cardiaque avec la fonction d'autocorrélation

L'identification de la fréquence cardiaque à partir de la fonction d'autocorrélation d'un signal ECG est une méthode couramment utilisée pour déterminer la fréquence cardiaque en utilisant des signaux ECG non invasifs. La fonction d'autocorrélation est un outil statistique qui mesure la similitude entre un signal et lui-même en décalant dans le temps. La fréquence cardiaque peut être obtenue en recherchant le premier maximum local après le maximum global (à $\tau = 0$) de la fonction d'autocorrélation. Cette méthode est simple et efficace, mais peut être affectée par le bruit et les interférences. Il est donc souvent nécessaire de prétraiter le signal avant de calculer la fonction d'autocorrélation pour améliorer la précision de la mesure de la fréquence cardiaque. Il est également important de limiter l'intervalle de recherche pour la fréquence cardiaque à la plage possible pour éviter des résultats erronés.

```

% Charger le signal ECG traité
ecg_signal = load('ecg.mat');

% Définir l'intervalle de recherche pour la fréquence cardiaque
min_bc = 40;
max_bc = 220;

% Calculer l'autocorrélation du signal ECG
[acf, lags] = xcorr(ecg_signal, ecg_signal);

% Trouver la fréquence cardiaque en se basant sur l'autocorrélation
[max_corr, max_index] = max(acf);
heart_rate = 60/(lags(max_index));

% Vérifier si la fréquence cardiaque est dans l'intervalle de recherche
if heart_rate > min_bc && heart_rate < max_bc
    disp(['Fréquence cardiaque : ', num2str(heart_rate), ' battements par minute']);
else
    disp('Fréquence cardiaque non détectée');
end

```

La première ligne utilise la fonction "max" de Matlab pour trouver le maximum de la fonction d'autocorrélation (acf) et l'assigne à la variable "max_corr", ainsi que l'index de cette valeur maximum dans le tableau "acf" à la variable "max_index".

La deuxième ligne utilise cet index pour trouver la fréquence cardiaque en utilisant le tableau des décalages (lags) et en calculant la fréquence cardiaque en utilisant la formule $60/\text{lag}(\text{max_index})$ ou $\text{lag}(\text{max_index})$ est le décalage temporel correspondant au premier maximum de l'autocorrélation.

Cette valeur est ensuite assignée à la variable "heart_rate" qui contient finalement la fréquence cardiaque en battements par minute.

Conclusion

Dans ce TP, on a exploré différentes méthodes pour améliorer le rapport signal sur bruit pour le signal ECG. On a trouvé un compromis entre la préservation de la forme du signal et la réduction du bruit en limitant la bande passante et en filtrant le signal. On a également utilisé la fonction d'autocorrélation pour identifier la fréquence cardiaque et écrit un programme pour effectuer automatiquement ce calcul. Les résultats montrent que le programme trouve efficacement la fréquence cardiaque.

En général, cet TP montre l'importance de la préparation des données pour obtenir des résultats fiables dans l'analyse de signaux ECG.