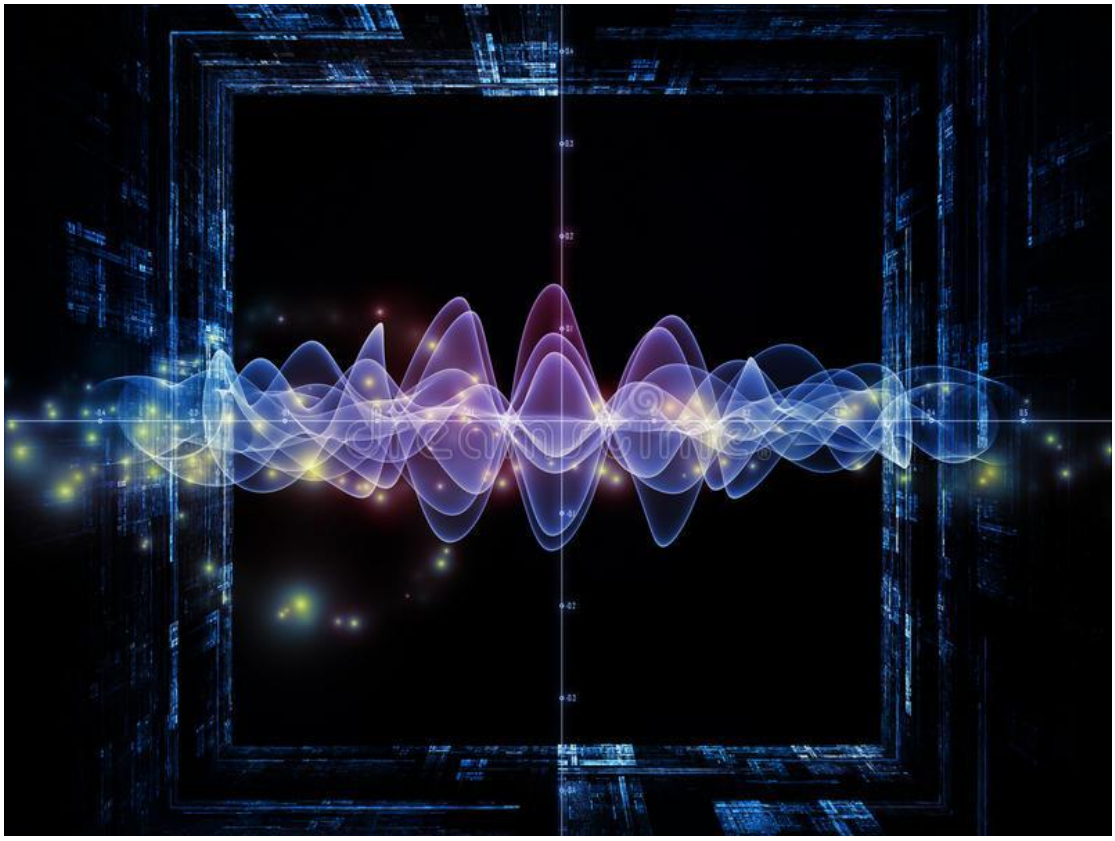
TP3 : Traitement d'un signal ECG:



Réaliser par : Youssef laamari

Module : Traitement de signal

Prof : AMMOUR Alae

Objectifs :

* **Suppression du bruit autour du signal produit par un électrocardiographe. Recherche de la fréquence cardiaque. Commentaires : Il est à remarquer que ce TP traite en principe des signaux continus. Or, l'utilisation de Matlab suppose l'échantillonnage du signal. Il faudra donc être vigilant par rapport aux différences de traitement entre le temps continu et le temps discret. Tracé des figures : toutes les figures devront être tracées avec les axes et les légendes des axes appropriés. Travail demandé : un script Matlab commenté contenant le travail réalisé et des commentaires sur ce que vous avez compris et pas compris, ou sur ce qui vous a semblé intéressant ou pas, bref tout commentaire pertinent sur le TP.**

1-Sauvegarder le signal ECG sur votre répertoire de travail, puis charger-le dans Matlab à l'aide la commande load.

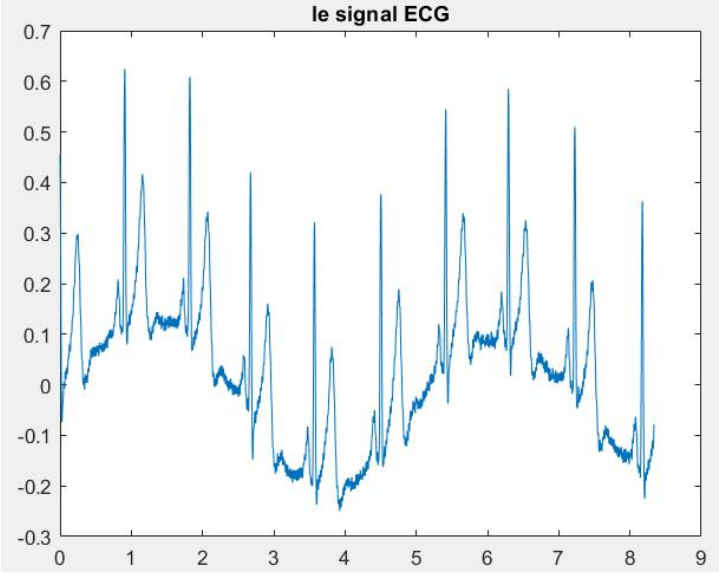
Code:

```
% Charger le fichier .mat contenant le signal ECG
load('ecg.mat');
```

2- Ce signal a été échantillonné avec une fréquence de 500Hz. Tracer-le en fonction du temps, puis faire un zoom sur une période du signal.

Code:

```
clear all
close all
clc
load("ecg.mat");
% Tracer le signal en fonction du temps
fs = 500; % Fréquence d'échantillonnage
t = (0:length(ecg)-1)/fs; % Temps
plot(t, ecg)
xlabel('Temps (s)')
ylabel('Amplitude')
```



3- 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.

Code:

```
clear all
close all
clc
load("ecg.mat");
% Tracer le signal en fonction du temps
fs = 500; % Fréquence d'échantillonnage
t = (0:length(ecg)-1)/fs; % Temps
plot(t, ecg)
xlabel('Temps (s)')
ylabel('Amplitude')

% Tracer le signal en fonction du temps
fs = 500; % Fréquence d'échantillonnage
t = (0:length(ecg_signal)-1)/fs; % Temps
plot(t, ecg_signal)
xlabel('Temps (s)')
ylabel('Amplitude')

% Faire un zoom sur une période du signal
xlim([t_start t_end])
```

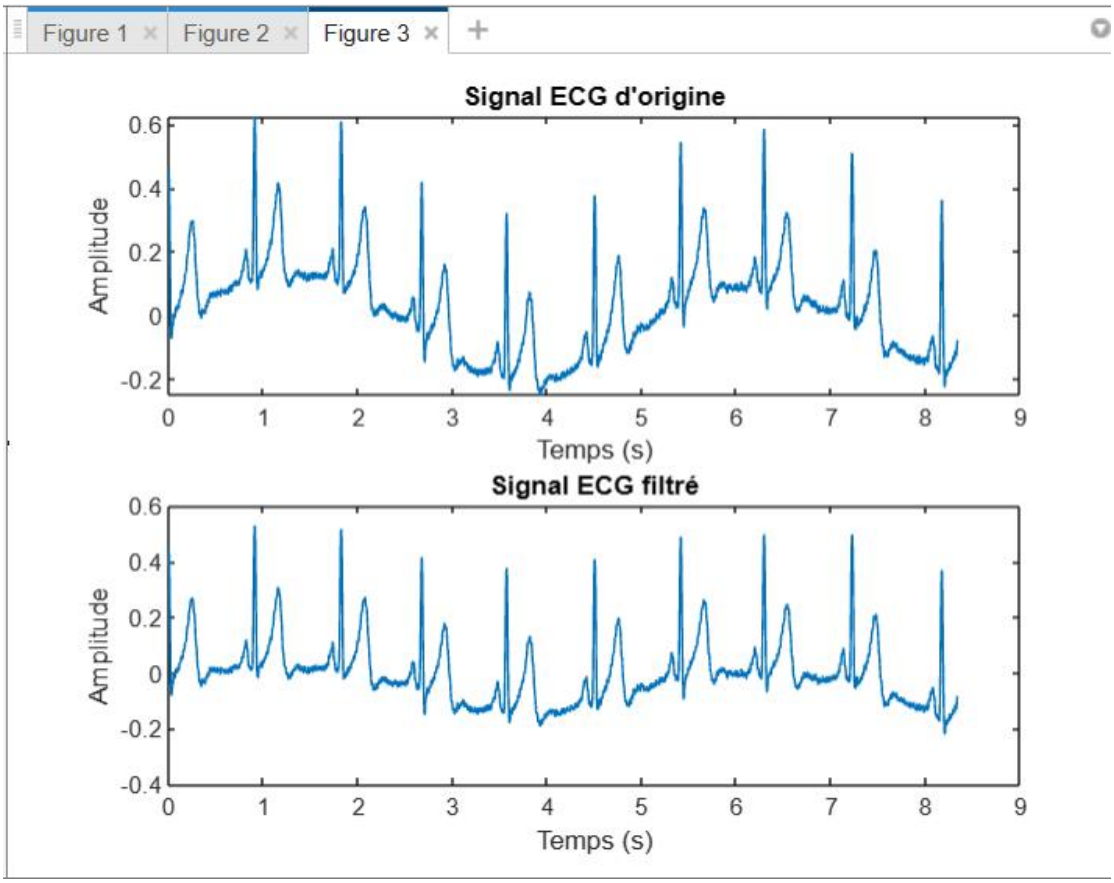
4- Tracer le nouveau signal ecg1, et noter les différences par rapport au signal d'origine.

Code:

```
% Tracer le signal filtré
figure
plot(t, ecg_filtered)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG filtré')

% Tracer les deux signaux côte à côte pour faciliter
la comparaison
figure
subplot(2,1,1)
plot(t, ecg_signal)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG d''origine')

subplot(2,1,2)
plot(t, ecg_filtered)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG filtré')
```



5 + 6)

Code:

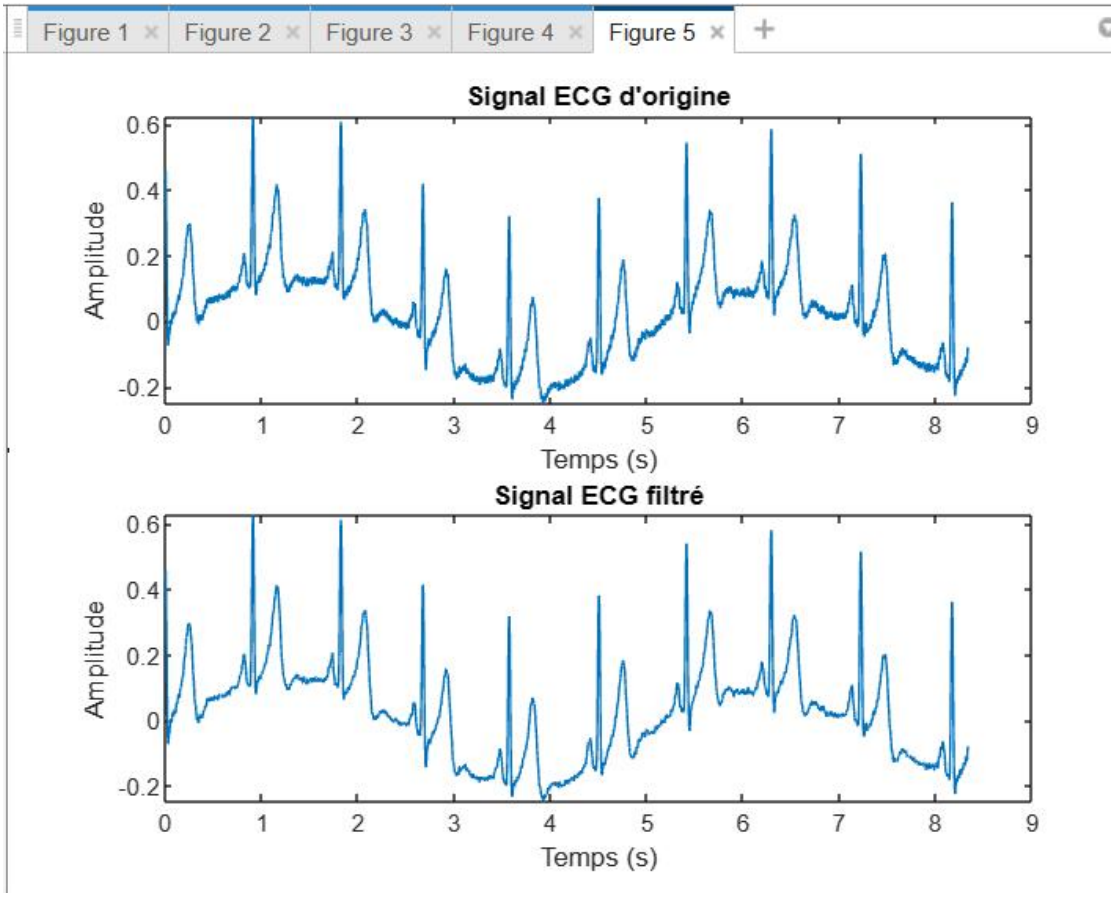
```
% Calculer la TFD du signal ECG
ecg_tfd = fft(ecg_signal);

% Régler la fréquence de bruit du secteur
50Hz à zéro
notch_freq = 50; % Fréquence de bruit du
secteur en Hz
notch_index =
round(notch_freq*length(ecg_signal)/fs);
ecg_tfd(notch_index-5:notch_index+5) = 0;

% Effectuer une TFDI pour restituer le signal filtré
ecg_filtered = ifft(ecg_tfd);
```

```
% Tracer les deux signaux côte à côte pour faciliter
la comparaison
figure
subplot(2,1,1)
plot(t, ecg_signal)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG d''origine')

subplot(2,1,2)
plot(t, ecg_filtered)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG filtré')
```



7-

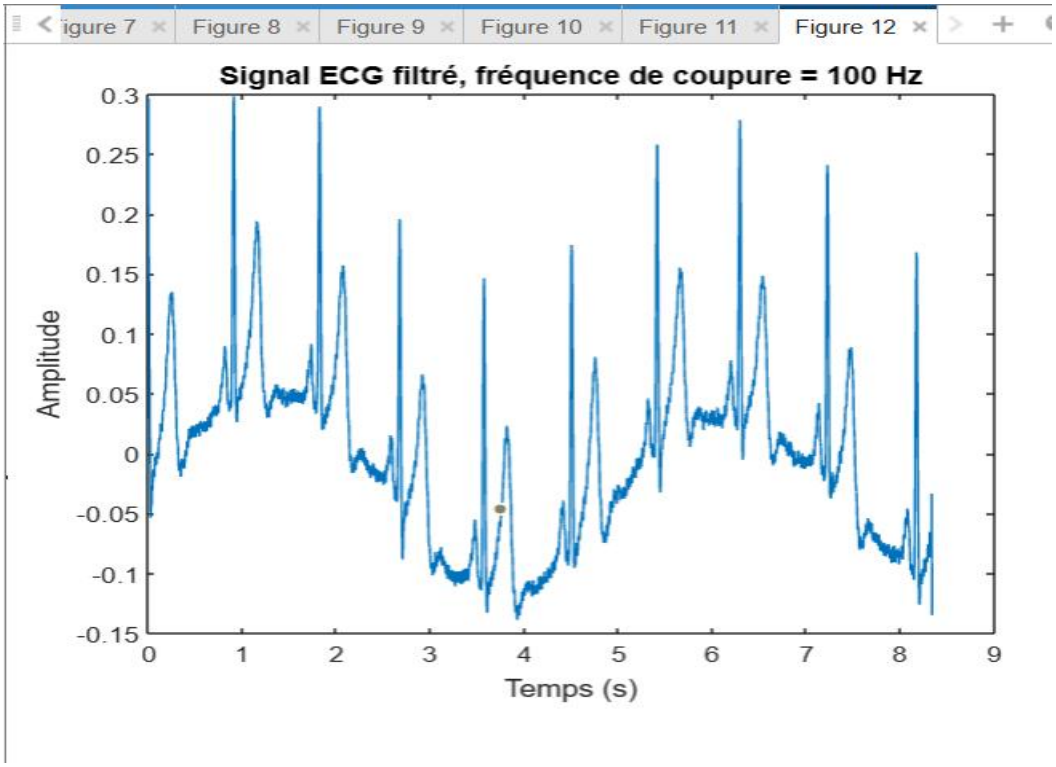
Code:

```
% Essayer différentes fréquences de coupure
cutoff_freq_list = [5, 10, 20, 30, 40, 50, 60, 70, 80,
90, 100];

for i = 1:length(cutoff_freq_list)
    cutoff_freq = cutoff_freq_list(i);
    cutoff_index =
round(cutoff_freq*length(ecg_signal)/fs);
    ecg_tfd = fft(ecg_signal);
    ecg_tfd(1:cutoff_index) = 0;
    ecg_filtered = ifft(ecg_tfd);

    % Tracer le signal filtré pour chaque fréquence
de coupure
    figure
    plot(t, ecg_filtered)
    xlabel('Temps (s)')
    ylabel('Amplitude')
    title(sprintf('Signal ECG filtré, fréquence de
coupure = %d Hz', cutoff_freq))
End
```

8)



9)

```
[acf,lags] = xcorr(ecg3,ecg3);
stem(lags/fe,acf)
```