

2023

TP3 - TRAITEMENT D'UN SIGNAL ECG

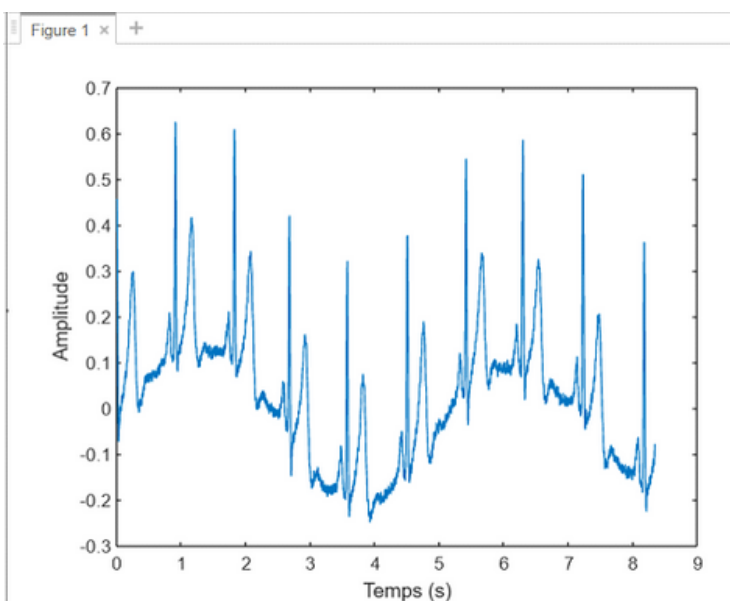
**MAROUANE AIT HAMMOU
ENCADRÉ PAR :
PR ALAE AMMOUR**

SUPPRESSION DU BRUIT PROVOQUÉ PAR LES MOUVEMENTS DU CORPS

1- SAUVEGARDER LE SIGNAL ECG SUR VOTRE RÉPERTOIRE DE TRAVAIL, PUIS CHARGER-LE DANS MATLAB À L'AIDE LA COMMANDE LOAD.

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.

```
TP3.m x +
1 clear all
2 close all
3 clc
4 load("ecg.mat");
5 % Tracer le signal en fonction du temps
6 fs = 500; % Fréquence d'échantillonnage
7 t = (0:length(ecg)-1)/fs; % Temps
8 plot(t, ecg)
9 xlabel('Temps (s)')
10 ylabel('Amplitude')
11
12 % Faire un zoom sur une période du signal
13 xlim([t_start t_end])
```

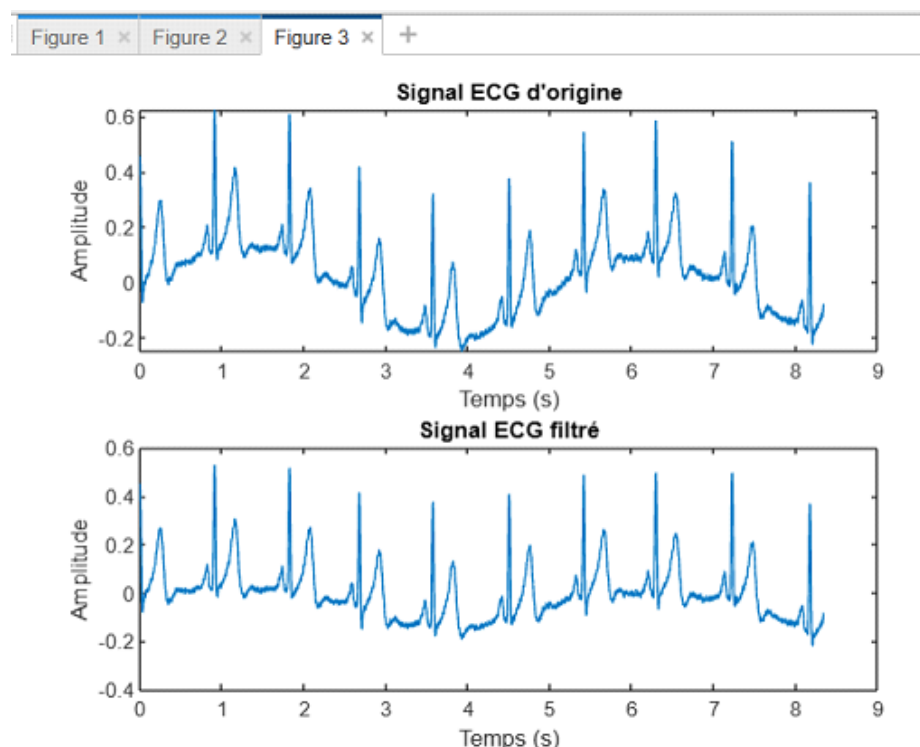


3- POUR SUPPRIMER LES BRUITS À TRÈS BASSE FRÉQUENCE DUES AUX MOUVEMENTS DU CORPS, ON UTILISERA UN FILTRE IDÉAL PASSE-HAUT. POUR CE FAIRE, CALCULER TOUT D'ABORD LA TFD DU SIGNAL ECG, RÉGLER LES FRÉQUENCES INFÉRIEURES À 0.5HZ À ZÉRO, PUIS EFFECTUER UNE TFDI POUR RESTITUER LE SIGNAL FILTRÉ.

4- TRACER LE NOUVEAU SIGNAL ECG1, ET NOTER LES DIFFÉRENCES PAR RAPPORT AU SIGNAL D'ORIGINE.

```
figure
subplot(2,1,1)
plot(t, ecg)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG d\'origine')

subplot(2,1,2)
plot(t, ecg_1)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG filtré')
```



SUPPRESSION DES INTERFÉRENCES

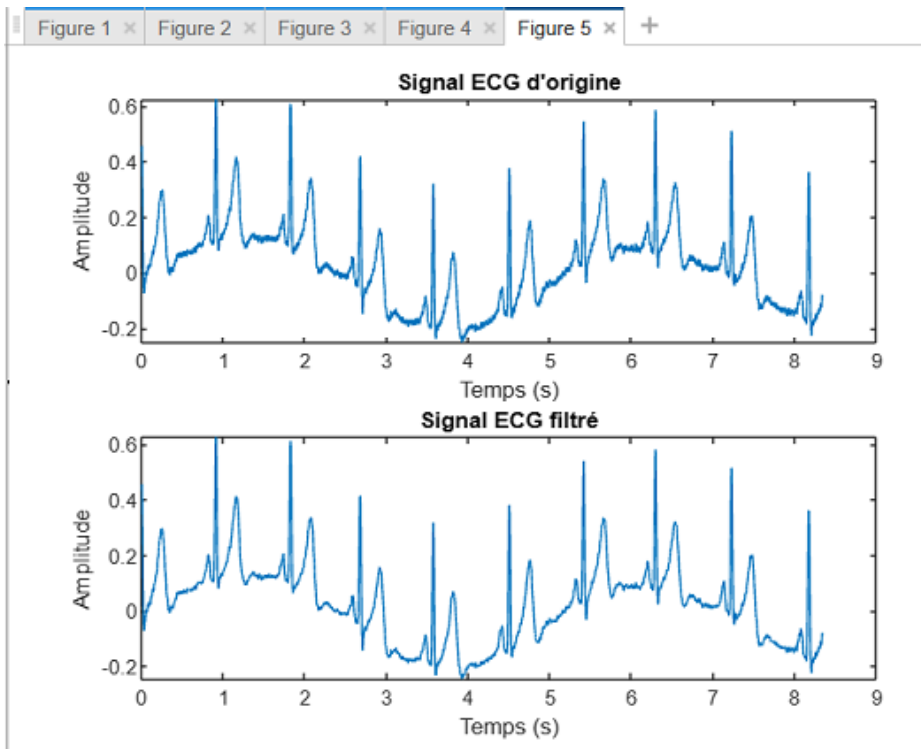
DES LIGNES ÉLECTRIQUES 50HZ

5. APPLIQUER UN FILTRE NOTCH IDÉAL POUR SUPPRIMER CETTE COMPOSANTE. LES FILTRES NOTCH SONT UTILISÉS POUR REJETER UNE SEULE FRÉQUENCE D'UNE BANDE DE FRÉQUENCE DONNÉE.

6. VISUALISER LE SIGNAL ECG2 APRÈS FILTRAGE.

```
figure
subplot(2,1,1)
plot(t, ecg)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG d''origine')
```

```
subplot(2,1,2)
plot(t, ecg_1)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal ECG filtré')
```



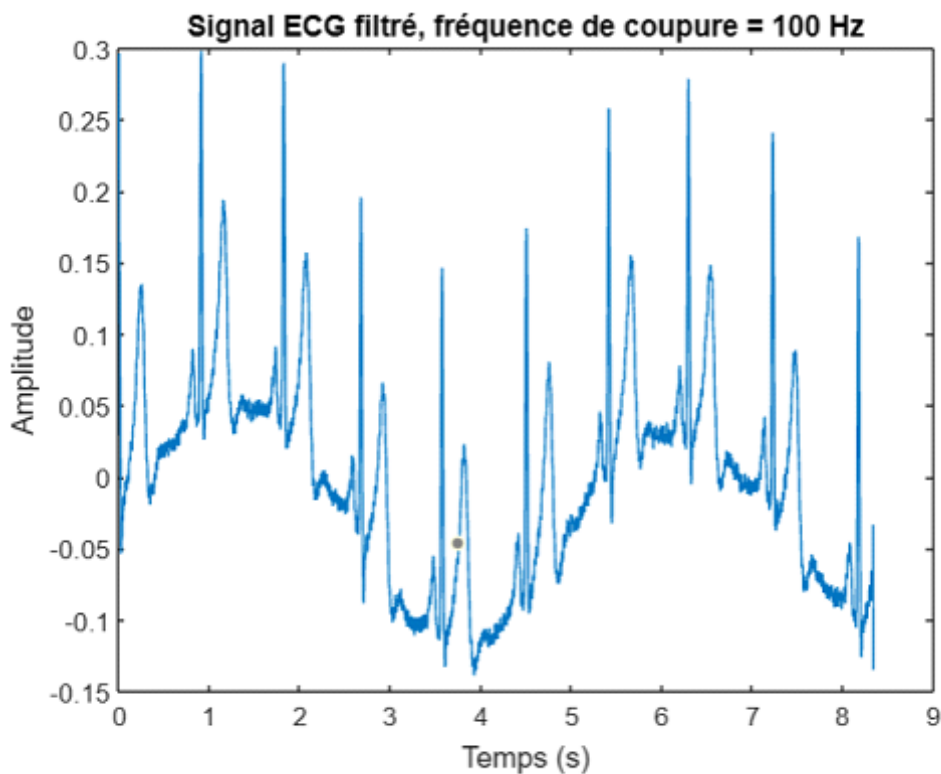
AMÉLIORATION DU RAPPORT SIGNAL SUR BRUIT

7. CHERCHER UN COMPROMIS SUR LA FRÉQUENCE DE COUPURE, QUI PERMETTRA DE PRÉSERVER LA FORME DU SIGNAL ECG ET RÉDUIRE AU MAXIMUM LE BRUIT. TESTER DIFFÉRENTS CHOIX, PUIS TRACER ET COMMENTER LES RÉSULTATS.

```
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)/fs);
    ecg_tfd = fft(ecg);
    ecg_tfd(1:cutoff_index) = 0;
    ecg_1 = ifft(ecg_tfd);

    % Tracer le signal filtré pour chaque fréquence de coupure
    figure
    plot(t, ecg_1)
    xlabel('Temps (s)')
    ylabel('Amplitude')
    title(sprintf('Signal ECG filtré, fréquence de coupure = %d Hz', cutoff_freq))
end
```



IDENTIFICATION DE LA FRÉQUENCE CARDIAQUE AVEC LA FONCTION D'AUTOCORRÉLATION

9. ECRIRE UN PROGRAMME PERMETTANT DE CALCULER L'AUTOCORRÉLATION DU SIGNAL ECG, PUIS DE CHERCHER CETTE FRÉQUENCE CARDIAQUE DE FAÇON AUTOMATIQUE. UTILISER CE PROGRAMME SUR LE SIGNAL TRAITÉ ECG3 OU ECG2 ET SUR LE SIGNAL ECG NON TRAITÉ. NB : IL FAUT LIMITER L'INTERVALLE DE RECHERCHE À LA PLAGE POSSIBLE DE LA FRÉQUENCE CARDIAQUE.

```
figure;  
[acf,lags] = xcorr(ecg_1,ecg_1);  
stem(lags/fs,acf)
```

