

# TP3

Signal ECG



---

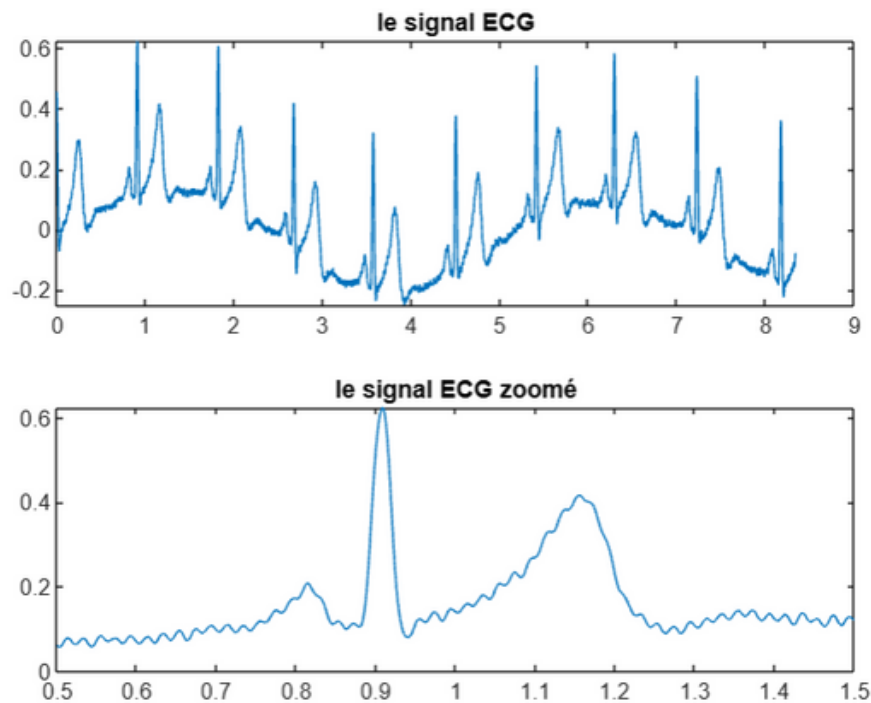
**BENACHOUR Ayoub**

## qst 1

```
load("ecg.mat");  
x=ecg;
```

## qst 2

```
fs=500;  
N=length(x);  
ts=1/fs;  
  
%tracer ECG en fonction de temps  
t=(0:N-1)*ts;  
subplot(2,1,1)  
plot(t,x);  
title("le signal ECG");  
  
% %tracer ECG zoomé  
subplot(2,1,2)  
plot(t,x);  
xlim([0.5, 1.5])  
| title("le signal ECG zoomé");
```



## qst 3

```
%le spectre Amplitude
```

```
y = fft(x);  
f = (0:N-1)*(fs/N);  
fshift = (-N/2:N/2-1)*(fs/N);
```

```
%spectre Amplitude centré
```

```
plot(fshift,fftshift(abs(y)))  
title("spectre Amplitude")
```

```
%suppression du bruit à très basse fréquence dues aux mouvements du corps
```

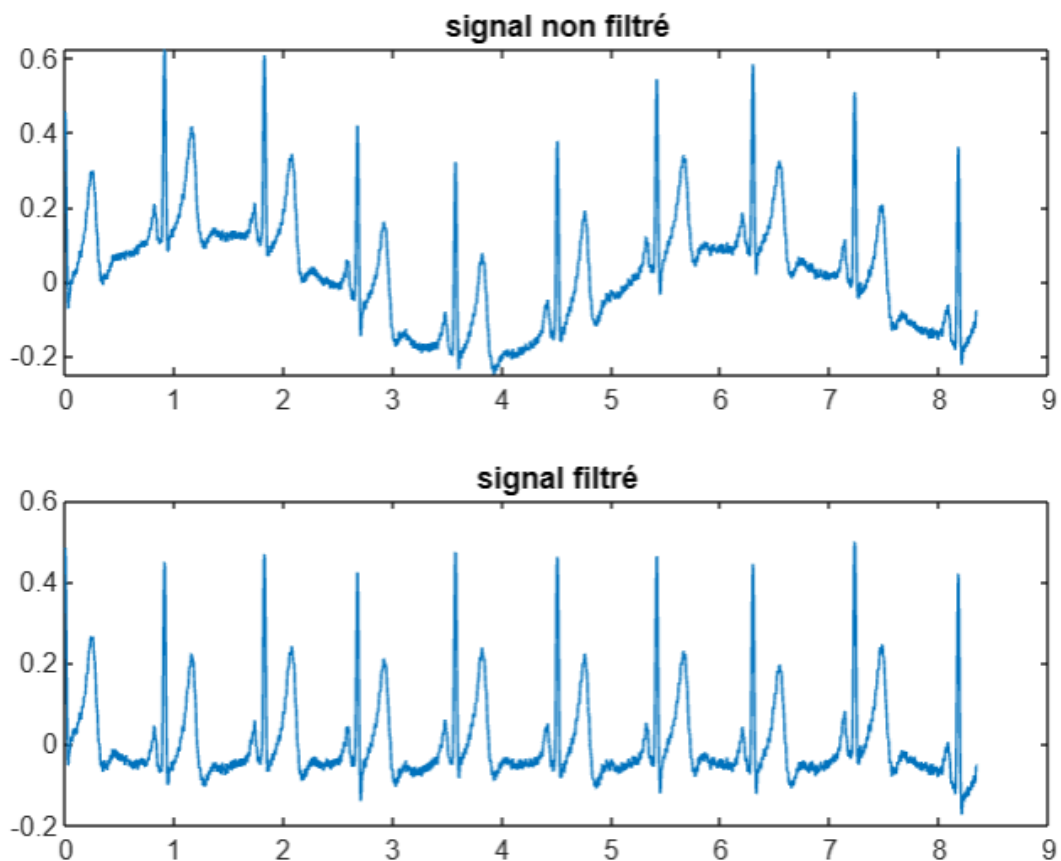
```
h = ones(size(x));  
fh = 0.5;  
index_h = ceil(fh*N/fs);  
h(1:index_h)=0;  
h(N-index_h+1:N)=0;  
ecg1_freq = h.*y;  
ecg1 =ifft(ecg1_freq,"symmetric");
```

Explication :

Pour supprimer les bruits à très basse fréquence dues aux mouvements du corps, on a utilisé la fonction `fft` pour convertir le signal de domaine temporel au domaine fréquentiel. Puis, on a créé un filtre passe-haut en réglant les fréquences inférieures à 0.5Hz à zéro. Enfin, on a utilisé la fonction `ifft` pour convertir le signal filtré de nouveau au domaine temporel.

## qst 4

```
subplot(2,1,1)
plot(t,ecg);
title("signal non filtré")
subplot(2,1,2)
plot(t,ecg1);
title("signal filtré")
```



Explication :

lorsqu on a utilisé un filtre passe-haut pour supprimer le bruit à très basse fréquence dans un signal ECG,nous remarquons une amélioration de la qualité du signal en enlevant les vibrations indésirables ou les bruits de fond.

## qst 5

```
% Elimination interference 50Hz

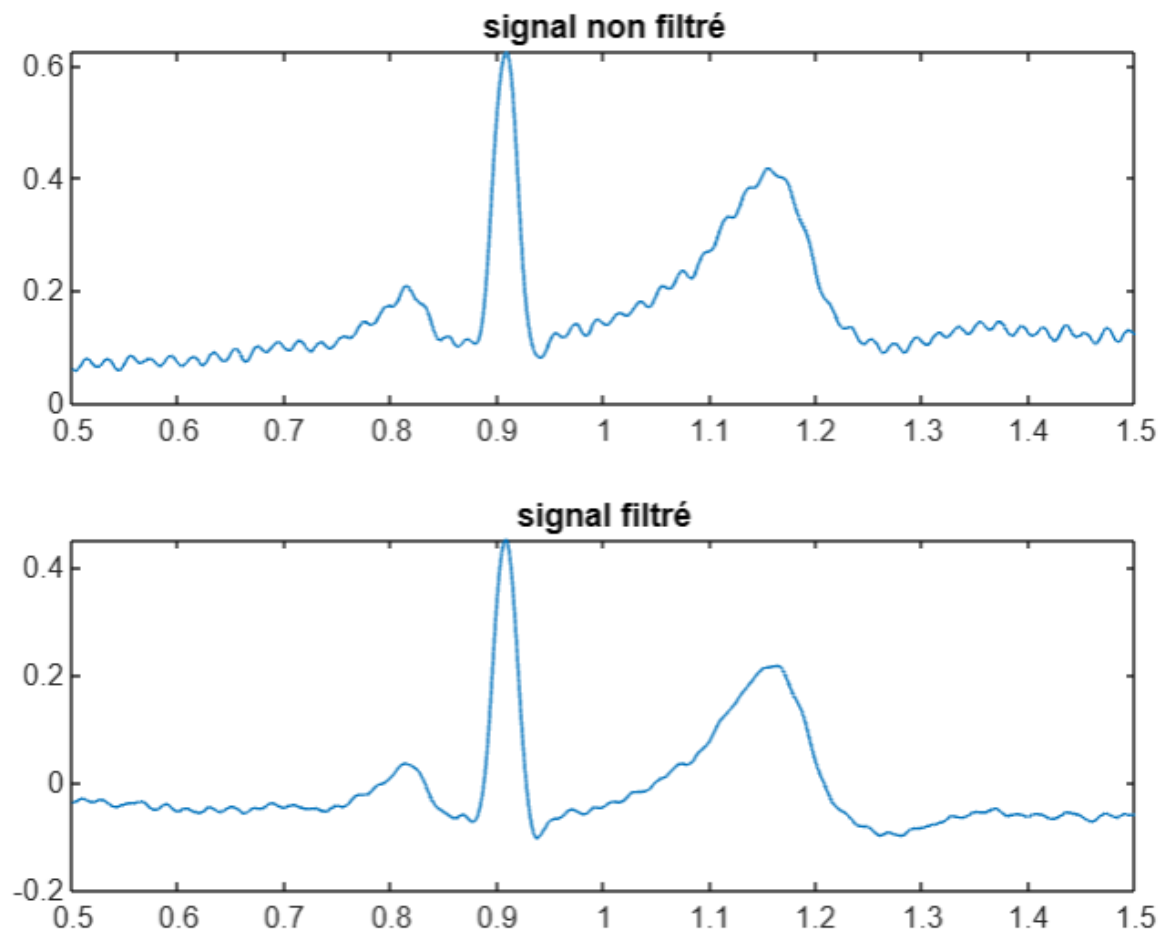
Notch = ones(size(x));
fcu = 50;
index_hcu = ceil(fcu*N/fs)+1;
Notch(index_hcu)=0;
Notch(index_hcu+2)=0;
ecg2_freq = Notch.*fft(ecg1);
ecg2 =ifft(ecg2_freq,"symmetric");
```

Explication :

utilise un filtre notch pour supprimer une bande de fréquence spécifique 50 hz dans un signal ECG. Le filtre notch est créé en utilisant un vecteur "Notch" de la même taille que le signal d'entrée "x", puis en remplaçant les valeurs correspondant à la fréquence cible (fcu) et aux deux fréquences adjacentes par zéro. Le signal d'entrée est ensuite converti en domaine fréquentiel en utilisant la FFT (Fast Fourier Transform), le filtre notch est appliqué en multipliant le signal fréquentiel par le vecteur "Notch", puis le signal filtré est converti de nouveau en domaine temporel en utilisant la IFFT (Inverse Fast Fourier Transform) avec l'option "symmetric" pour conserver la symétrie du signal d'origine.

## qst 6

```
subplot(2,1,1)
plot(t,ecg);
xlim([0.5 1.5])
title("signal non filtré")
subplot(2,1,2)|
plot(t,ecg2);
title("signal filtré")
xlim([0.5 1.5])
```



## Explication :

Lorsqu'on a utilisé un filtre notch pour supprimer les interférences des lignes électriques de fréquence 50Hz dans un signal ECG, on a remarqué une amélioration de la qualité du signal en enlevant les vibrations indésirables ou les bruits de fond causés par les lignes électriques. Cela permet de mieux visualiser les variations de fréquence cardiaque normales et les anomalies éventuelles. Cela permet également d'améliorer la précision des mesures de la fréquence cardiaque. On peut également remarquer que les oscillations de fréquence 50Hz qui étaient présentes dans le signal d'origine ont été supprimées ou fortement atténuées.

## qst 7

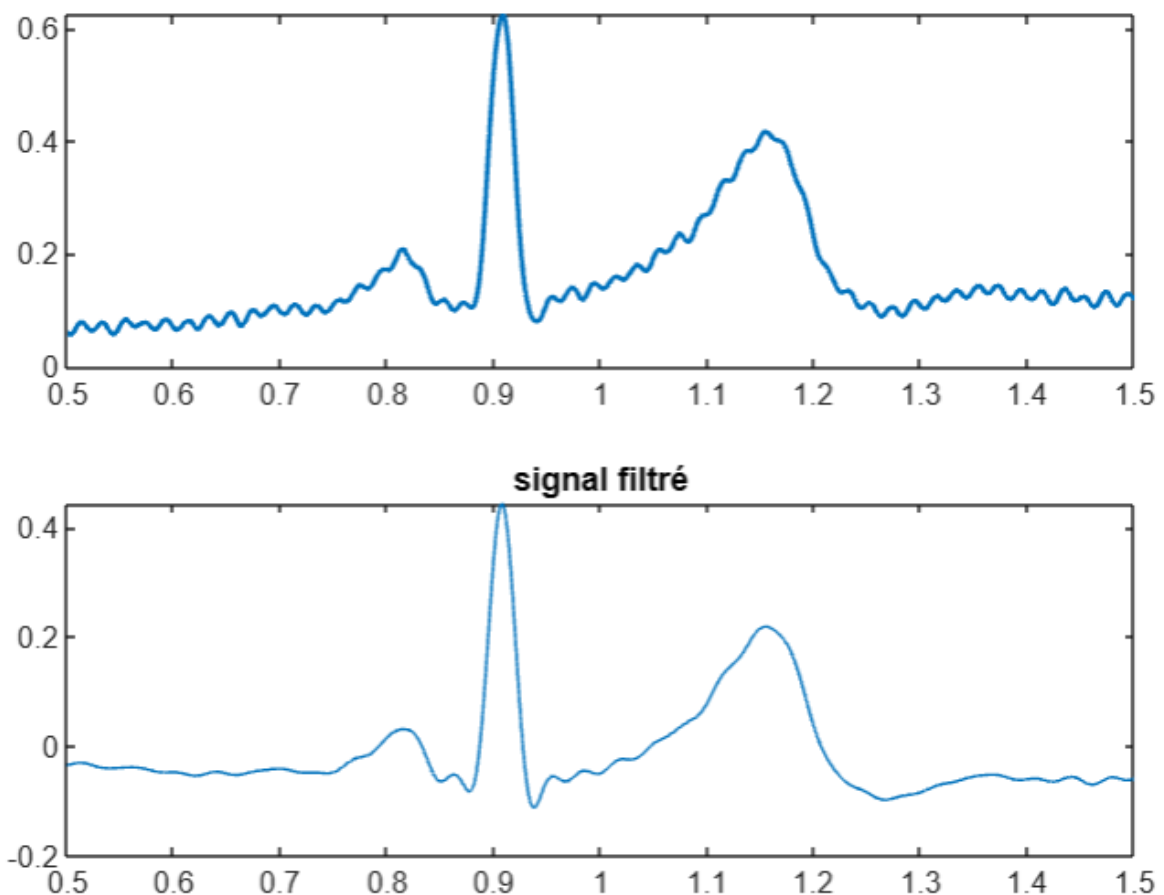
```
pass_bas = zeros(size(x));  
fcb = 30;  
index_hcb = ceil(fcb*N/fs);  
pass_bas(1:index_hcb)=1;  
pass_bas(N-index_hcb+1:N)=1;  
ecg3_freq = pass_bas.*fft(ecg2);  
ecg3 = ifft(ecg3_freq,"symmetric");
```

### Explication :

Après plusieurs tests, on a choisi 30Hz comme fréquence de coupure, puis, on a utilisé un filtre passe-bas pour laisser passer les fréquences inférieures à une fréquence de coupure spécifique dans un signal ECG. Le filtre passe-bas est créé en utilisant un vecteur "pass\_bas" de la même taille que le signal d'entrée "x", puis en remplaçant les valeurs correspondant aux fréquences inférieures à la fréquence de coupure (fcb) par un. Le signal d'entrée est ensuite converti en domaine fréquentiel en utilisant la FFT (Fast Fourier Transform), le filtre passe-bas est appliqué en multipliant le signal fréquentiel par le vecteur "pass\_bas", puis le signal filtré est converti de nouveau en domaine temporel en utilisant la IFFT (Inverse Fast Fourier Transform) avec l'option "symmetric" pour conserver la symétrie du signal d'origine.

## qst 8

```
subplot(2,1,1)
plot(t,ecg,"linewidth",1.5);
xlim([0.5 1.5])
subplot(2,1,2)
plot(t,ecg3);
title("signal filtré")
xlim([0.5 1.5])
```



Explication :

nous observons un changement dans l'apparence du signal sur le graphe. Le signal filtré aura une forme plus lisse et moins de détails dans les hautes fréquences. Les oscillations à haute fréquence qui étaient présentes dans le signal d'origine auront été atténuées ou supprimées. Cela peut rendre plus facile de visualiser les variations de fréquence cardiaque normales et les anomalies éventuelles.



## qst 9

### Identification de la fréquence cardiaque avec la fonction d'autocorrélation :

```
%autocorrélation de signal ECG
[c,lags] = xcorr(ecg3,ecg3);

%stem(lags/fs,c)
E = length(c); %la longueur de la fonction d'autocorrélation
Vector = [0]; %initialisation d'un vecteur
for n = 1:E
    if c(n) > 10
        Vector(end+1) = c(n); %l'ajout de valeur au vecteur
    end
    %pour éliminer les valeurs qui égaient au 0
    M = max(Vector);
    in = find(c == M);
    s = lags(in);
    if s < 12
        Vector(Vector == M) = [];
    end
end

%extraction de la valeur max de vecteur
frequence = (s/fs)*60; %calculer la fréquence
frequence_min=30;
frequence_max=160;
if frequence > frequence_min & frequence < frequence_max
    fprintf('la fréquence cardiaque de ce patient est :%f .',frequence);
end
```

#### Explication :

on a utilisé la technique d'autocorrélation pour estimer la fréquence cardiaque d'un signal ECG filtré par la fonction "xcorr" pour calculer la fonction d'autocorrélation du signal filtré. ensuite on a utilisé une boucle pour parcourir les valeurs de la fonction d'autocorrélation, et enregistre les valeurs supérieures à 10 dans un vecteur, et on a calculé ensuite la valeur maximale du vecteur et trouve son indice dans la fonction d'autocorrélation pour calculer la fréquence cardiaque en utilisant l'échantillonnage du signal, et vérifie que la fréquence cardiaque est dans une plage acceptable (30 à 160 battements par minute) avant de l'afficher .

## qst 10

non , le programme ne trouve pas de bon pouls , une fréquence cardiaque normale se situe entre 60 et 100 battements par minute , cependant , notre programme trouve une frequence de 54,84 battements par minute , ce qui signifie que le cœur bat à une fréquence anormalement faible.