TP3 Traitement de signal ECG :

clear all

close all

clc

%Declaration des variable

load('ecg.mat');

fe = 500;

N = length(ecg);

t = (0:N-1)\*1/fe;

f=(0:N-1)\*(fe/N);

fc = 0.5;

fc0 = 50;

fc1=40;

%[0.5;1.5] la taille d une periode

%%

%tarnsformer de fourier rapide

y = fft(ecg);

fshift = (-N/2:N/2-1)\*(fe/N);

%filtrage

%creation du filtre pass haut

filtre\_pass\_Haut = ones(size(ecg));

index\_fc = ceil((fc\*N)/fe);

filtre\_pass\_Haut(1:index\_fc) = 0;

filtre\_pass\_Haut(N-index\_fc+1:N) = 0;

%filtrage

ecg\_filtre\_freq = filtre\_pass\_Haut .\*y;

%restitution du signal filtrer

ecg\_filtre\_temp = ifft(ecg\_filtre\_freq,"symmetric");

%le bruit de bass frequence

bruit= ecg-ecg\_filtre\_temp;

%%

%filtrage du bruit d'interferance

%creation du filtre

filtre\_interferance = ones(size(ecg));

index\_fc0 = ceil((fc0\*N)/fe)+1;

filtre\_interferance(index\_fc0)=0;

filtre\_interferance(N-index\_fc0+1)=0;

%application du filtre

ecg\_filtre\_int\_freq = filtre\_interferance .\*fft(ecg\_filtre\_temp);

%restitution du signal filtrer

ecg\_filtre\_int\_temp = ifft(ecg\_filtre\_int\_freq,"symmetric");

bruit\_inter = ecg\_filtre\_temp-ecg\_filtre\_int\_temp;

%%

%tarnsformer de fourier rapide

y = fft(ecg);

fshift = (-N/2:N/2-1)\*(fe/N);

%filtrage

%creation du filtre pass haut

filtre\_pass\_Haut = ones(size(ecg));

index\_fc = ceil((fc\*N)/fe);

filtre\_pass\_Haut(1:index\_fc) = 0;

filtre\_pass\_Haut(N-index\_fc+1:N) = 0;

%filtrage

ecg\_filtre\_freq = filtre\_pass\_Haut .\*y;

%restitution du signal filtrer

ecg\_filtre\_temp = ifft(ecg\_filtre\_freq,"symmetric");

%le bruit de bass frequence

bruit= ecg-ecg\_filtre\_temp;

%%

%filtrage du bruit haute frequence

%creation du filtre pass bas

filtre\_pass\_bas = zeros(size(ecg));

index\_fc1 = ceil((fc1\*N)/fe);

filtre\_pass\_bas(1:index\_fc1)=1;

filtre\_pass\_bas(N-index\_fc1+1:N)=1;

%application du filtre

ecg\_filtre\_bas\_freq = filtre\_pass\_bas .\*fft(ecg\_filtre\_temp);

%restitution du signal filtrer

ecg\_filtre\_bas\_temp = ifft(ecg\_filtre\_bas\_freq,"symmetric");

%bruit haute frequence

bruit\_haut = ecg\_filtre\_int\_temp-ecg\_filtre\_bas\_temp;

%%

% representation

subplot(4,3,1)

plot(t,ecg,'linewidth',1)

xlim([0.5 1.5])

legend(" signal d'origine")

subplot(4,3,2)

plot(fshift,fftshift(abs(y)));

legend("spectre du signal d'origine")

subplot(4,3,3)

plot(t,bruit,'linewidth',1);

legend("bruit")

subplot(4,3,4)

plot(t,ecg\_filtre\_temp,'linewidth',1);

xlim([0.5 1.5])

legend("signal filtre")

subplot(4,3,5)

plot(fshift,fftshift(abs(fft(ecg\_filtre\_temp))));

legend("spectre du signal filtrer")

subplot(4,3,6)

plot(t,bruit\_inter,"Linewidth",1)

legend("bruit interferance")

subplot(4,3,7)

plot(t,ecg\_filtre\_int\_temp,'linewidth',1);

legend(" signal filtrer interferance")

xlim([0.5 1.5])

subplot(4,3,8)

plot(fshift,fftshift(abs(fft(ecg\_filtre\_int\_temp))));

legend("spectre du signal filtrer")

subplot(4,3,9)

plot(t,bruit\_haut,'linewidth',1);

legend("bruit")

subplot(4,3,10)

plot(t,ecg\_filtre\_bas\_temp,'linewidth',1);

xlim([0.5 1.5])

legend("signal filtre")

subplot(4,3,11)

plot(fshift,fftshift(abs(fft(ecg\_filtre\_bas\_temp))));

legend("spectre du signal filtrer")

%%

% subplot(1,3,1)

% plot(f,filtre\_pass\_Haut,"Linewidth",1.5)

% legend("Filtre pass haut")

% subplot(1,3,3)

% plot(f,filtre\_interferance,"Linewidth",1.5)

% legend("Filtre interferance")

% subplot(4,3,12)

%

% plot(t,bruit,'linewidth',1);

% legend("bruit")