"EXTRACTION OF ALPHA AND BETA RHYTHMS OF THE PROVIDED SAMPLE EEG"

Step 1: Band pass Filter Design

- The required specifications of the Filter :

  Passband: 8 Hz to 30 Hz;
  - Transition bands: 1 Hz each;
  - Passband ripple: 0.01;
  - Stopband attenuation: -41 dB.

- The ML code to design the Band pass filter :

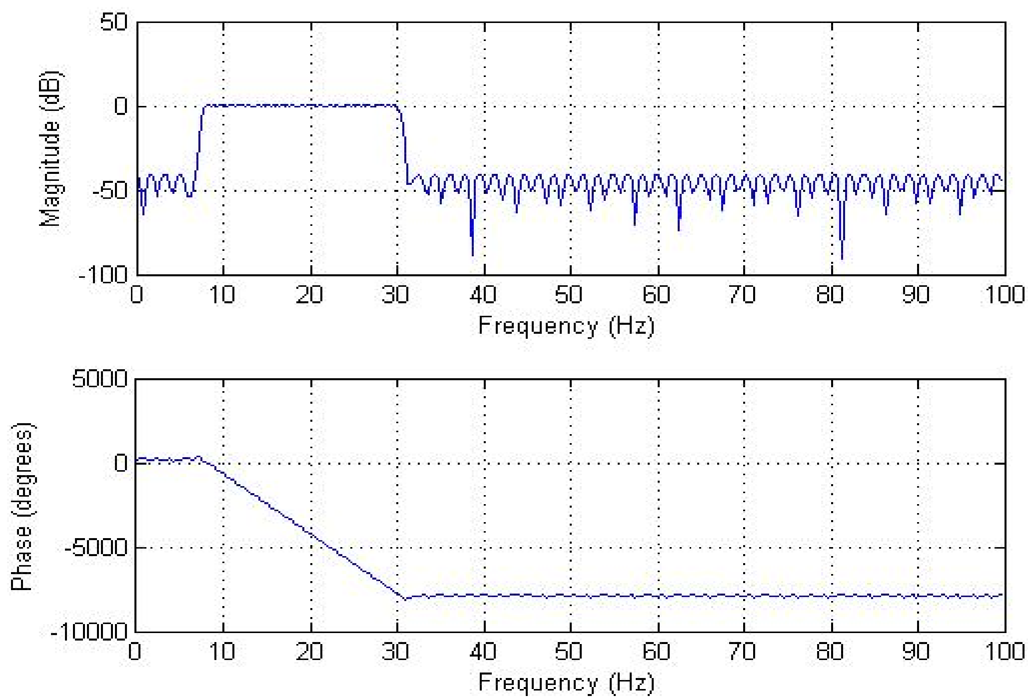
```
function [b]= bpf()

%- Passband: 8 Hz to 30 Hz;
%- Transition bands: 1 Hz each;
%- Passband ripple: 0.01;
%- Stopband attenuation: -41 dB.
%load('test.mat')
rp=-20*log10(1-0.01);
rs=48;
rplin=0.01;
rslin=10^(-2.05)
fs=200;
%fnorm=freq/fs;
%f2=[7 8 30 31];
%a2=[0 1 1 0];
[n,fo,mo,w] = firpmord( [ 7 8 30 31], [ 0 1 0 ], [rslin  rplin  rslin], fs );
b=firpm(n,fo,mo,w);
%n
%figure(1)
%freqz(b,1,256,fs);
%fvtool(b,1);


%c3=test.data(15,:);
%figure(2)
%plot(c3);

%y=filter(b,1,c3);
%figure(3);
%plot(y)
End
```

The Major Functions used are:

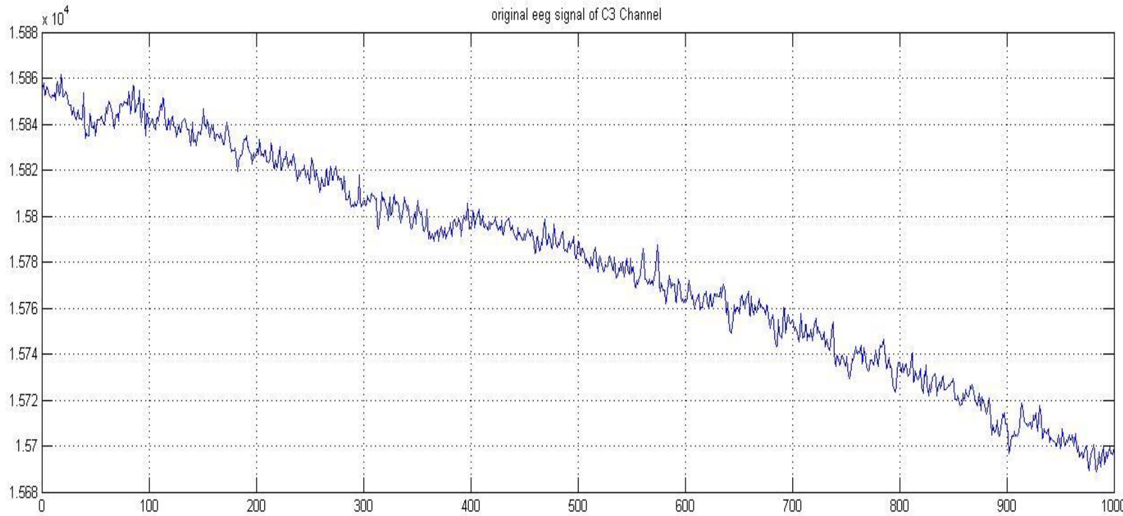
1. Firpmord(f,a,dev,fs):
   This function is used to estimate the optimal order of the Parks-McClellan optimal FIR filter design.

2. Firpm(n,fo,mo,w]:
   This function is used design Parks-McClellan optimal FIR filter design using the specifications produced by the firpmord function.The function outputs a vector of fir filter coefficients.

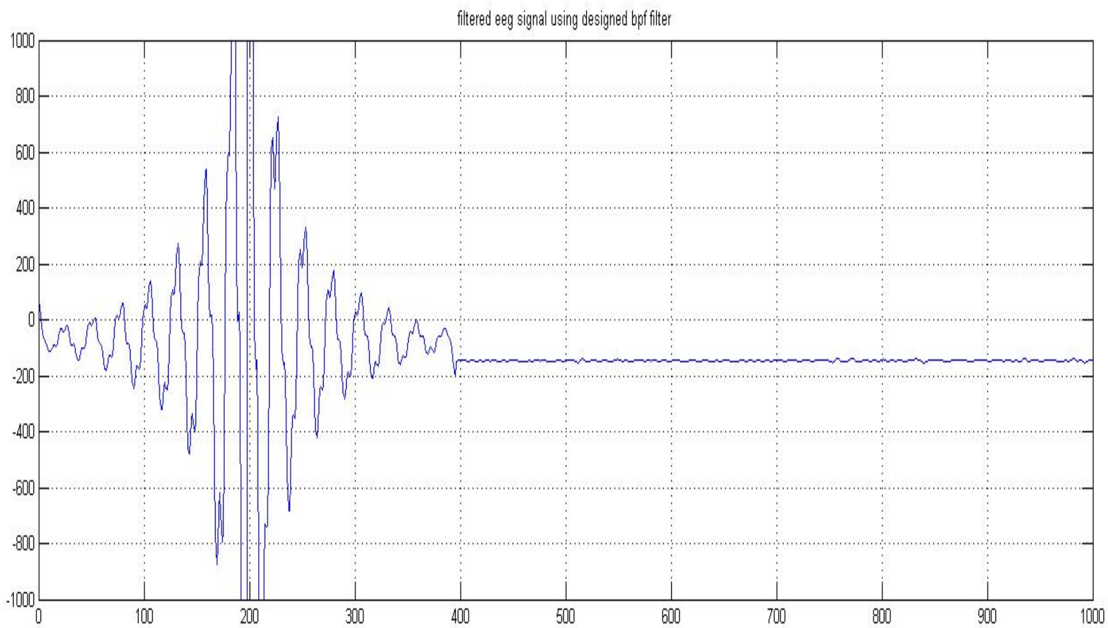
- Output Analysis:
  The frequency response of the designed fir filter is given by figure(1).The time plot of the EEG signal from channel C3 ,which is located in the 15^{th} row of the test.data matrix is given by figure(2).As an experiment the time plot of the EEG data from channel C3 after filtered through the given filter is also plotted and shown in figure(3).



Figure(1):Frequency and Phase response of designed FIR filter



Figure(2):Original EEG signal from channel C3



Figure(3):Filtered EEG signal from C3 channel

- ✓ From figure(1) we can see from the magnitude response that the passband of the filter is from 8 Hz to 30 Hz with a transition band of 1 Hz only . Here the passband ripple is very less that is about 0.01 linear scale and we can see the stopband attenuation is -48dB.

  From the test i.e. figure(3) we can see that the frequency components that do not fall between 7 Hz and 30 Hz are suppressed and only the components falling on the pass bands are passed.

  As the designed filter meets all the required specifications as shown from the figures above and has successfully tested on the C3 Channel , the ML code is a very Good Laboratory product and can be used to test and analyze the signals with varying specifications.

Step 2: common reference spatial filter and DC removal

- ➢ Algorithm used
  The implementation of the common reference spatial filter with two classes is based on the following basic algorithm.
  > Input: a set of EEG trials Xi with known class Yi ∈ {1, 2}.
  > Output: W projection matrix
  > 1: for i = 1 to I do
  > 2: PXi = Cov (Xi) {Compute Sample Covariance matrix}
  > 3: end for
  > 4: P1 = A (PXi , Yi = 1) {Arithmetic mean of covariance matrices for class 1}
  > 5: P2 = A (PXi , Yi = 2) {Arithmetic mean of covariance matrices for class 2}
  > 6: (P1 + P2) −1P1 = WD1WT {Eigenvalue decomposition}
  > 7: return W

- ➢ The ML code is:

```
function [z1ac z2ac]=crsf(eegdata)
%%% common refrene spacial filter implementation and DC removal

N=1200;
load ('test.mat');
eegdata=(test.data);
%number of channels
K=32;
fs=200;
R1=eegdata([1:16],:);%first class
R2=eegdata([17:32],:);%second class
r1=(R1*R1')/trace(R1*R1');
r2=(R2*R2')/trace(R2*R2');
n=length(r1);
r1=mean(r1,8);
r2=mean(r2,8);
r=r1+r2;
[U,Lambda] = eig(r);%%eigen vector generation
[Lambda,ind] = sort(diag(Lambda),'descend');
U=U(:,ind);
```

```
P=sqrt(inv(diag(Lambda)))*U';%%penalty calculation
S{1}=P*r1*P';
S{2}=P*r2*P';
[B,G] = eig(S{1},S{2});
[G,ind] = sort(diag(G));
B = B(:,ind);
W=(B'*P);%%common spatial filter cofficients
for i=1:length(ind), W(i,:)=W(i,:)./norm(W(i,:)); end

%C3=test.data(15,:);
z1=W*eegdata([1:16],:);
z2=W*eegdata([17:32],:);
%std(eegdata(15,:));variance of original data
%std(z(15,:));variance of filtered data
figure(4);
subplot(2,1,1);
plot(z1(15,:));
title('C3 data after filtered with common spatial filter');


%%%%%%DC removal %%%%%%%%%
for i=1:16
    for j=1:1200
        z1ac(i,:)=detrend(z1(i,:));
        j=j+1;
    end
end
for i=1:16
    for j=1:1200
        z2ac(i,:)=detrend(z2(i,:));
        j=j+1;
    end
end
subplot(2,1,2)
plot(z1ac(15,:));
title('C3 data after application of common spatial filter and DC removal')
end
```

- Major Functions used:

  1. eig():
     Eigenvalues and eigenvectors.
     E = eig(A) produces a column vector E containing the eigenvalues of a square matrix A.
  2. sort():
     Sort in ascending or descending order.
     For vectors, sort(X) sorts the elements of X in ascending order.
     For matrices, sort(X) sorts each column of X in ascending order.

For N-D arrays, sort(X) sorts the along the first non-singleton dimension of X

3. detrend():

   Remove a linear trend from a vector, usually for FFT processing.

   Y = detrend(X) removes the best straight-line fit linear trend from the data in vector X and returns the residual in vector Y. If X is amatrix, detrend removes the trend from each column of the matrix.

4. mean():

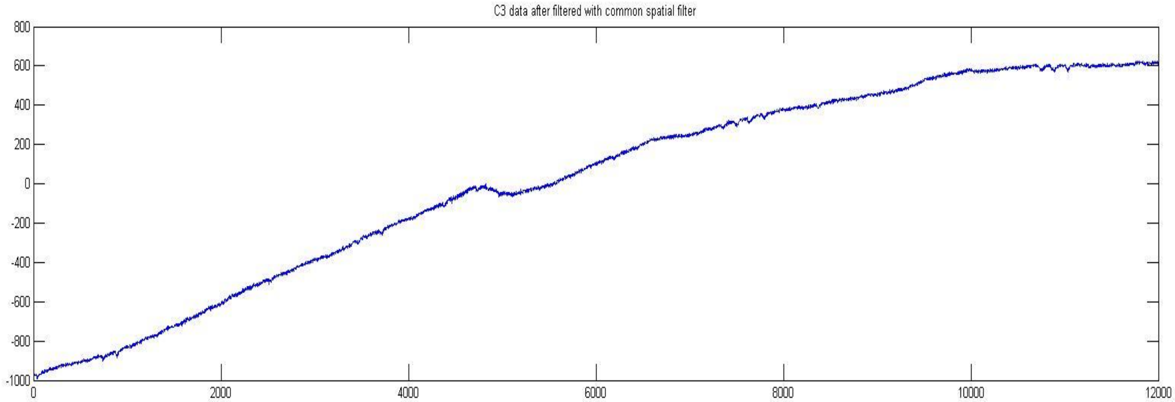
   Average or mean value.

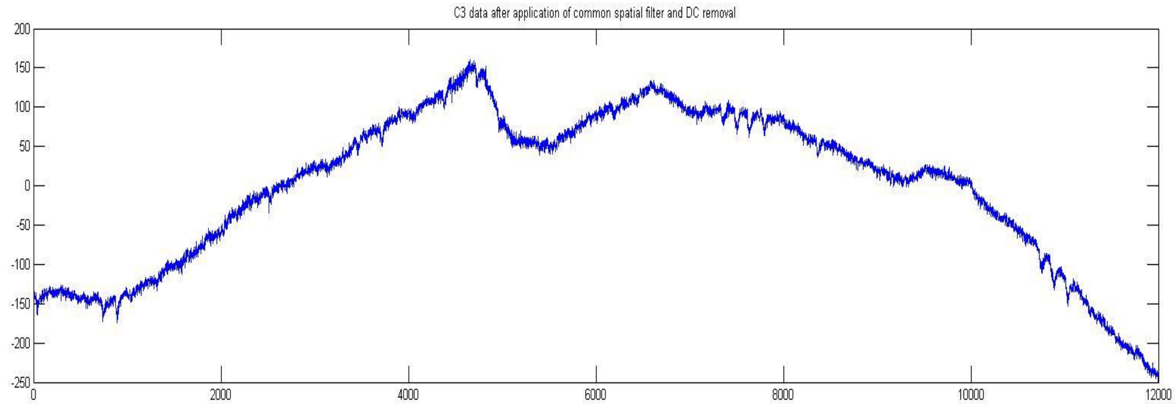
   S = mean(X) is the mean value of the elements in X if X is a vector. For matrices, S is a row vector containing the mean value of each column.

   For N-D arrays, S is the mean value of the elements along the first array dimension whose size does not equal 1.

- Results and analysis:

  Figure(4) below shows the EEG signal from channel C3 after application of common spatial filter given by the coefficient matrix W and also shows the data after DC removal.





Figure(4):EEG signal from channel C3 after common spatial filtration and DC removal

Step 3:Spectrum Estimation

- The ML code is :

```
clc
clear all
load('test.mat')
fs=200;%sampling frequency 200 HZ
```

```
dat=test.data;%data initializtion
%%step 2 : common spatial filterring and DC removal
[z1,z2]=crsf(dat)%step 2 o/p in z
ch15=dat(15,:);
psch15=z1(15,:);%load ch5 with processed chanel C3 data
%%figure;
%%freqz(ch15,1,1024,fs);%unprocessed channel C3 frequency plot
%%figure;
%%freqz(ch15,1,1024,fs);
%%filter application
b=bpf();
fildat=filter(b,1,z1);%filteration of processed data
fltch15=fildat(15,:);
figure;
%freqz(fltch15,1,1024,fs);%%Filtered channel C3 frequency plot
plot(fltch15);
title('preprocessed C3 data after filtered by bpf')

%%%fragment extraction
tt=12000/200;%total time of a channel in sec
n=randi(tt-5);%random time selection
t1=n/tt;
t2=t1+0.5;
n1=fix(t1*12000);
n2=fix(t2*12000);
nfframe=dat(15,[n1:n2])
frame=fildat(15,[n1:n2]);
figure;
%pxx=pburg(frame,10,[],200)
pburg(nfframe,10,[],200);%%Auto regressive power spectrum Analysis of chanel C3
title('C3 spectrum for random 0.5 sec fragment for unfiltered data')

figure;
%pxx=pburg(frame,10,[],200)
pburg(frame,10,[],200);%%Auto regressive power spectrum Analysis of chanel C3
title('C3 spectrum for random 0.5 sec fragment')
```

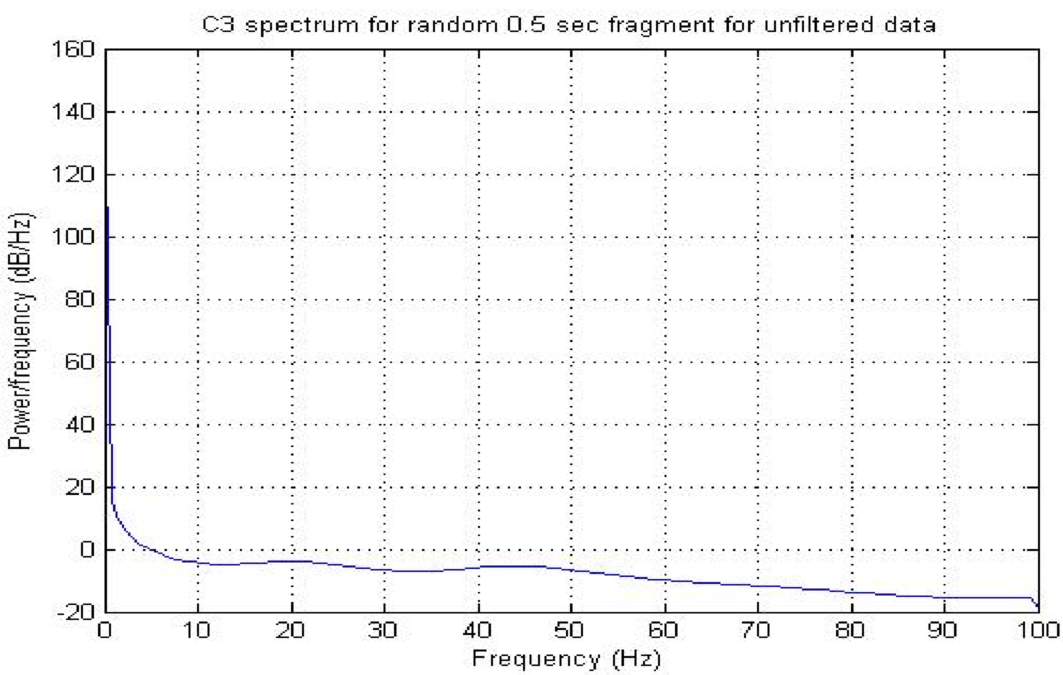
- Major Functions used:

1. filter():
   One-dimensional digital filter.
   Y = filter(B,A,X) filters the data in vector X with the filter described by vectors A and B to create the filtered data Y

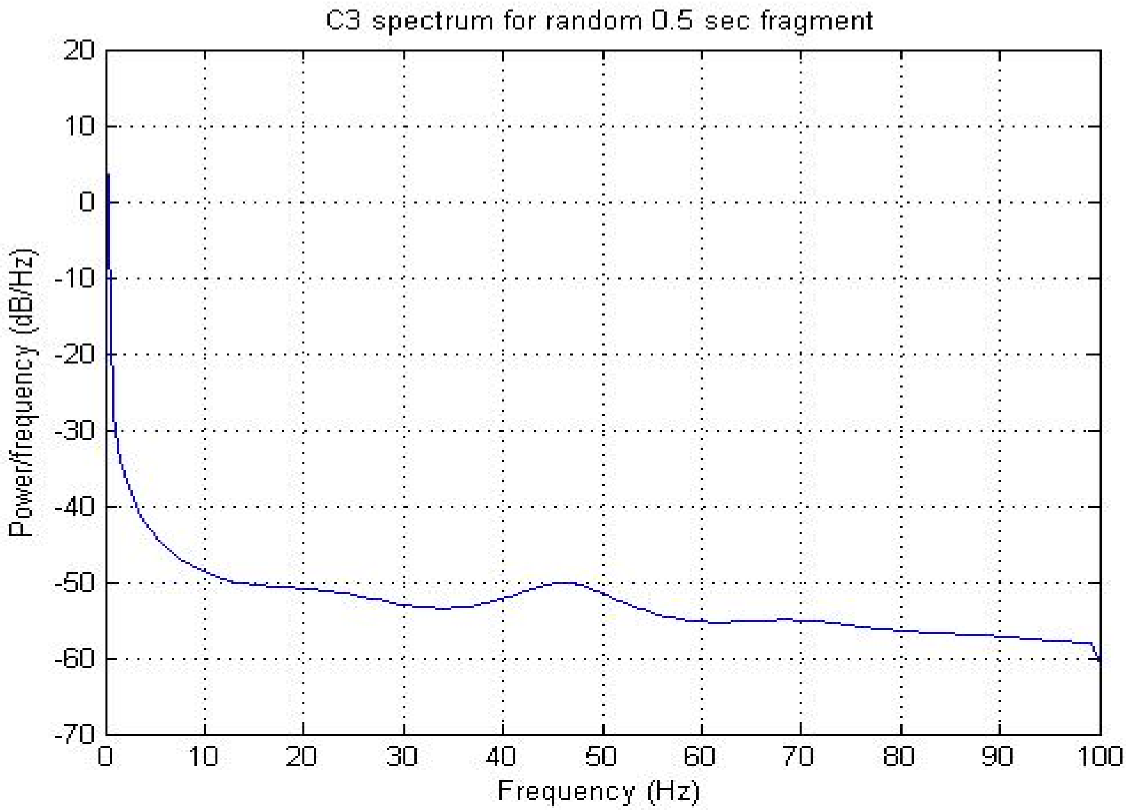
2. randi():
Pseudorandom integers from a uniform discrete distribution.
R = randi(IMAX,N) returns an N-by-N matrix containing pseudorandom

3. pburg():
Power Spectral Density (PSD) estimate via Burg's method.
Pxx = pburg(X,ORDER) returns the PSD of a discrete-time signal vector X in the vector Pxx. Pxx is the distribution of power per unit frequency. The frequency is expressed in units of radians/sample. ORDER is the order of the autoregressive (AR) model used to produce the PSD. pburg uses a default FFT length of 256 which determines the length of Pxx.

- Result and Analysis
Figure(5) is the power spectrum of the unprocessed EEG data from C3 Channel and figure(6) shows the power spectrum of filtered C3 data



Figure(5): Spectrum of unprocessed C3 channel data



Figure(6):Spectrum of Filtered C3 channel Data

- For the spectrum estimation of the processed signal, Autoregressive method using burg algorithm is used as most of the studies and experiments have verified that the AR method is the most valuable spectrum estimation method for the spectrum estimation and feature extraction of the EEG signal. Also as we are dealing with short fragments of data Autoregressive (AR) spectral estimation techniques are known to provide better resolution than classical periodogram methods when short segments of data are selected for analysis.