Lab #3

ECE 593

TeleHealthcare Engineering)

Raed Suftah

110-44-889

❖ What is & Why use wavelet?

Learning goal: Wavelet tutorial;
Content: Use Haar or other basic mother wavelets to decompose and reconstruct for EEG signals .

[1] Write a function to build orthonormal Quadrature Mirror Filters for different wavelets: (a) Haar (b) Beylkin and (c) Coiflet.

```
 function f = MakeONFilter(Type,~)

if strcmp(Type,'Haar'),
    f = [1 1] ./ sqrt(2);
end

if strcmp(Type,'Beylkin'),
    f = [   .099305765374   .424215360813   .699825214057
...
            .449718251149   -.110927598348  -.264497231446
...
            .026900308804   .155538731877   -.017520746267
...
            -.088543630623  .019679866044   .042916387274
...
            -.017460408696  -.014365807969  .010040411845
...
            .001484234782   -.002736031626  .000640485329];
end

if strcmp(Type,'Coiflet'),
        f = [   .038580777748   -.126969125396 -
.077161555496  ...
                .607491641386   .745687558934
.226584265197];

f = f ./ norm(f);
end
```

The function *f = MakeONFilter(Type)* returns the low-pass filter as *f*. The input Type is string type parameter. It allows you to choose which filter you want to create for wavelet transform.

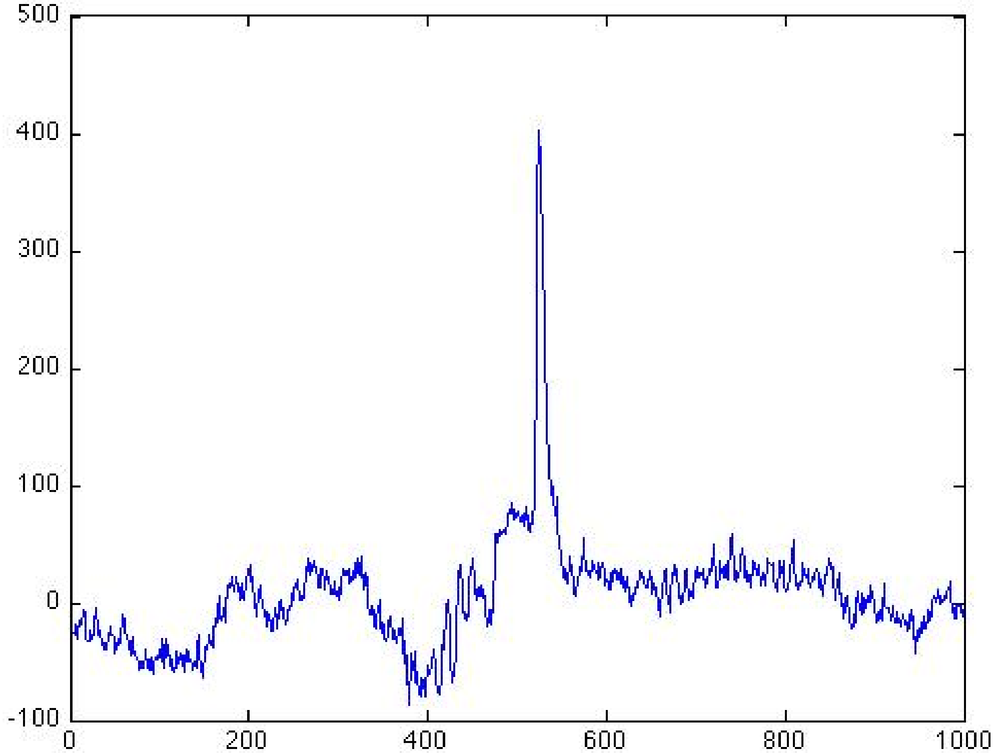
[2] Write a function to construct the wavelet transform matrix

The function *W = WavMat(h,N,k0,shift)* returns the wavelet transform matrix.

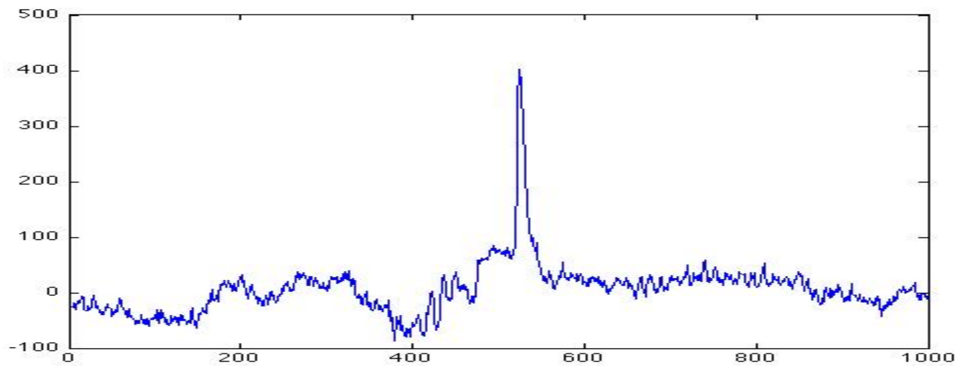
%	h	low-pass filter corresponding to orthogonal WT
%	N	size of matrix/length of data. Should be power of 2.
%	k0	depth of transformation. Ranges from 1 to J = log2(N).
%	shift	the matrix is not unique an any integer Default is 2.
%	W	N x N wavelet transformation matrix

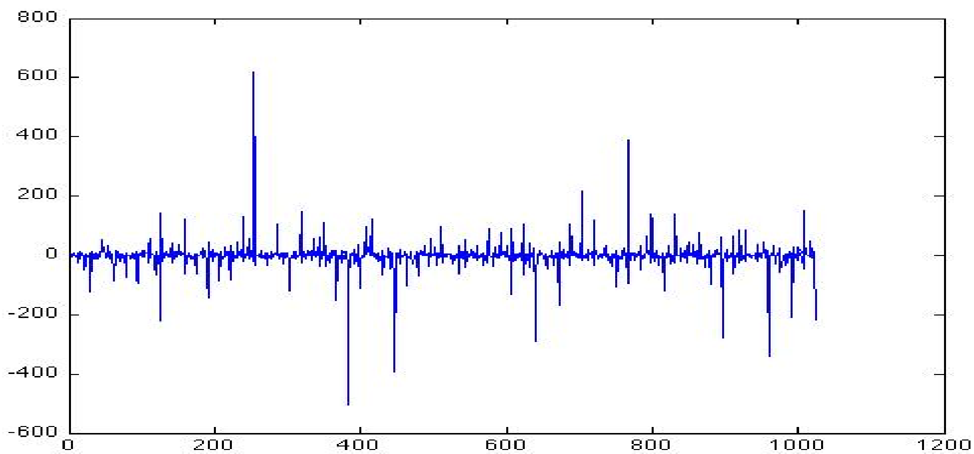
```
function W = WavMat(h, N, k0, shift)
if nargin==3
    shift = 2;
end
J = log2(N);
if nargin==2
    shift = 2;
    k0 = J;
end
%--make QM filter G
     h=h(:)';  g = fliplr(h .* (-1).^(1:length(h)));
if (J ~= floor(J) )
    error('N has to be a power of 2.')
end
h=[h,zeros(1,N)]; %extend filter H by 0's to sample by
modulus
g=[g,zeros(1,N)]; %extend filter G by 0's to sample by
modulus
oldmat = eye(2^(J-k0));
for k= k0:-1:1
    clear gmat; clear hmat;
         ubJk = 2^(J-k); ubJk1 = 2^(J-k+1);
   for  jj= 1:ubJk
       for ii=1:ubJk1
           modulus = mod(N+ii-2*jj+shift,ubJk1);
           modulus = modulus + (modulus == 0)*ubJk1;
           hmat(ii,jj) = h(modulus);
           gmat(ii,jj) = g(modulus);
       end
   end
  W = [oldmat * hmat'; gmat' ];
   oldmat = W;
end
```

[3] Load EEG signal in test_EEG.mat and display the EEG signal.

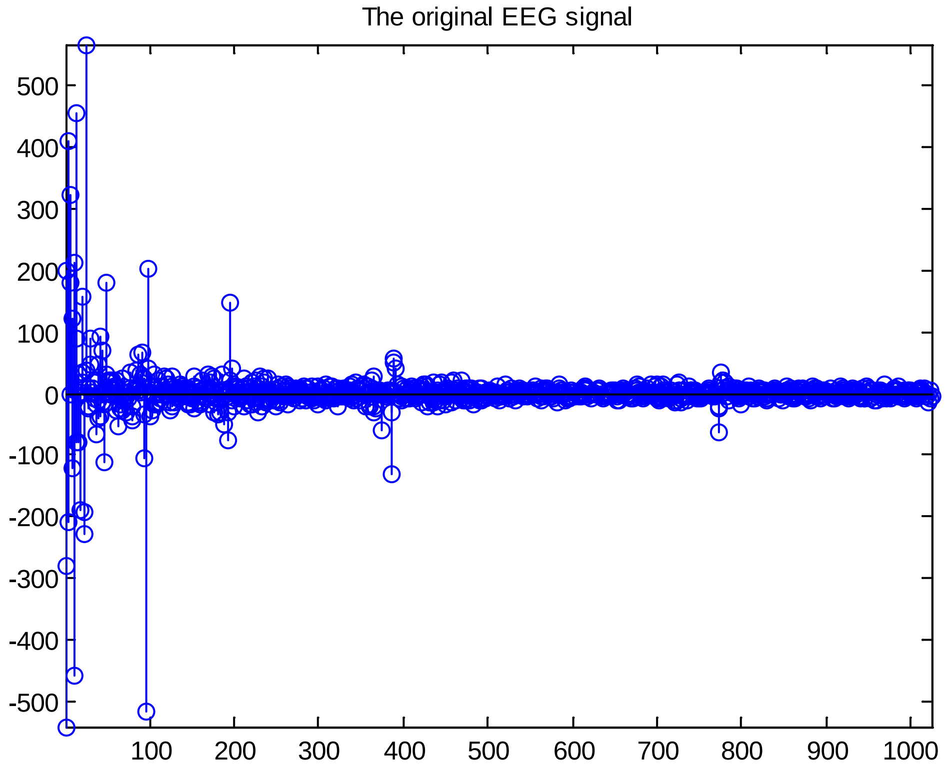


[4] Use the wavelet transform matrix to decompose and reconstruct the EEG signal. Plot the original signal and wavelet coefficients.

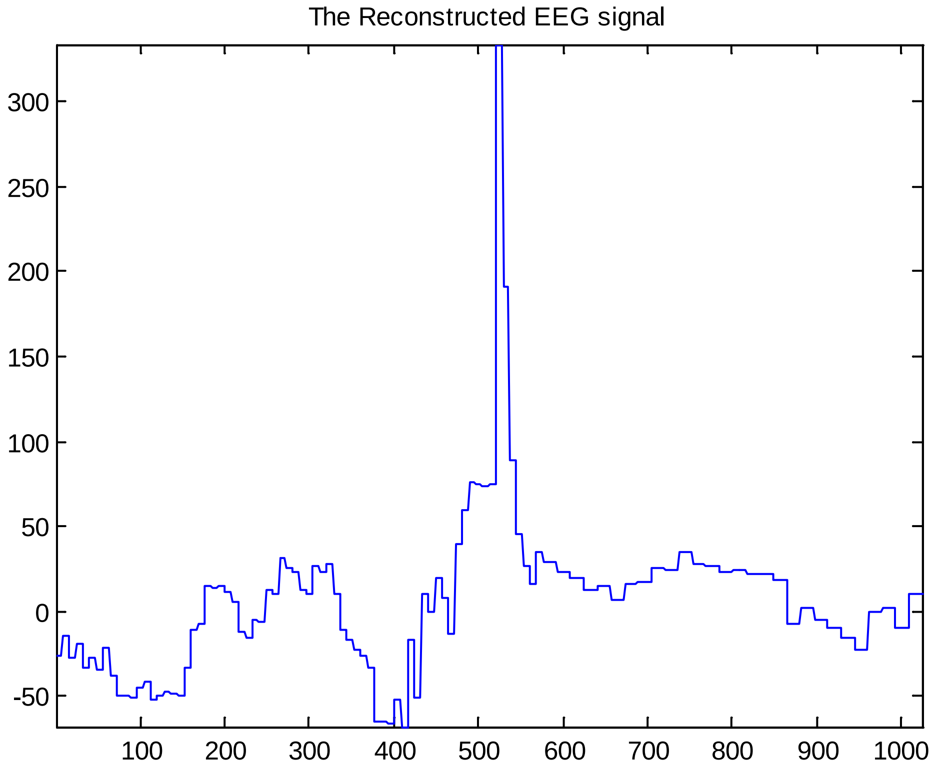




Then, I got the wavelet coefficient:

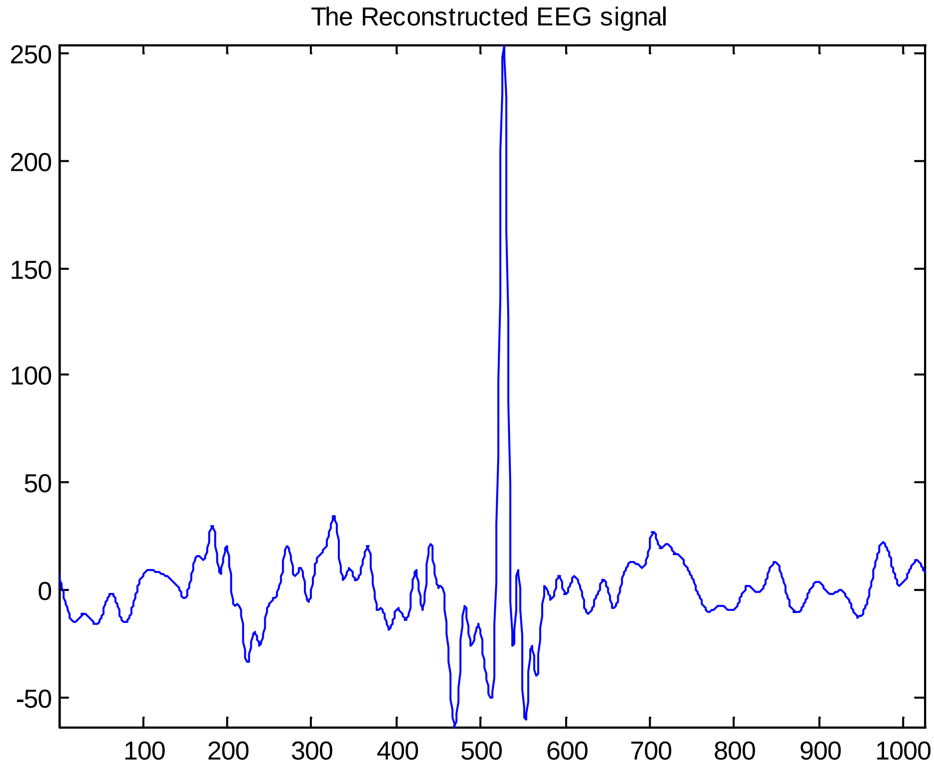


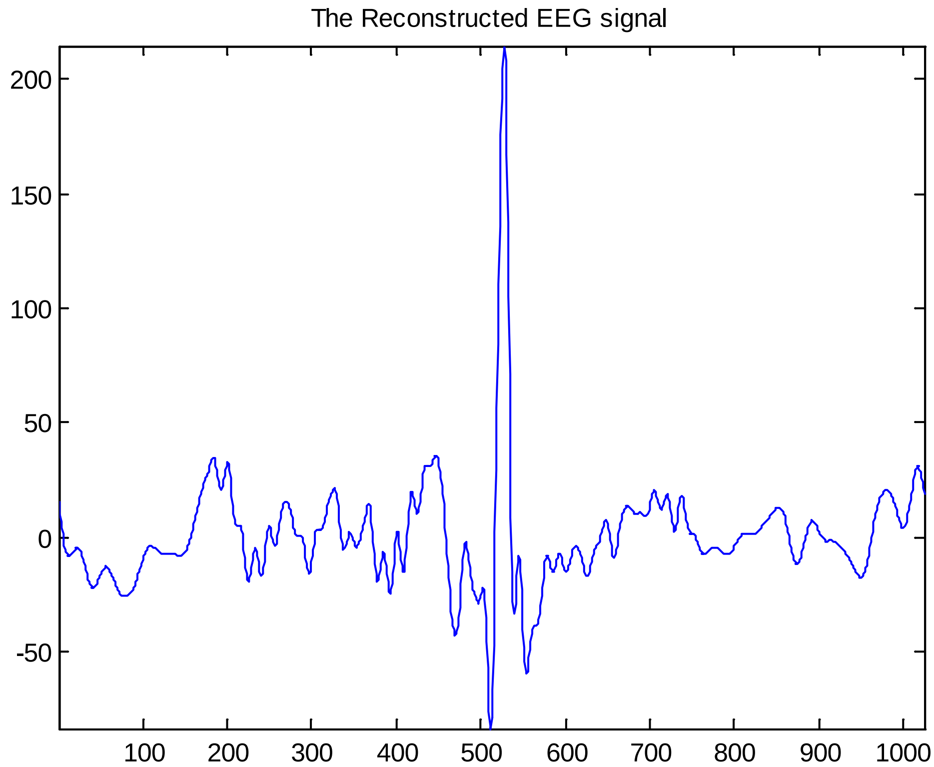
To reconstruct the signal, here I choose the first 100 coefficients and keep the remaining coefficients as zeros. That means, we keep the most significant low frequency parts and move out the high frequency parts from signal.



The Harr filter is a rectangle filter. It explains why the reconstructed signal shapes like that.

Now I use Beylkin and Coiflet filters for WT (the Coiflet has Par = 4). Still keep first 100 coefficients, the reconstructed signals will be more smooth then the Harr one. But the outlines of reconstructed signals are slightly different due to their filter's shape.


The Reconstructed EEG signal
250
200
150
100
50
0
-50
100
200
300
400
500
600
700
800
900
1000


The Reconstructed EEG signal
200
150
100
50
0
-50
100
200
300
400
500
600
700
800
900
1000