Lab 1 - Wavelet-based biomedical signal processing

- Lab Task 1.1 (need 1 week) – **What & Why is wavelet?**

  Learning goal: Wavelet tutorial;
  Content: Use Haar or other basic mother wavelets to study the Time-Frequency domain features of a downloaded EEG signals.

- Lab Task 1.2 (need 1 week) – **How to use WT (wavelet transform)?**

  Learning goal: Wavelet-based compression.
  Content: check the compression ratio by wavelets. Reconstruct the compressed EEG stream.

- Lab Task 1.3 (need 1 week) – **What do we need in the signal**?

  Learning goal: Seek singularity points via LHE algorithm.
  Content: Use WTMM theory to find the LHE of the wavelet "dominant" features.

What & Why is wavelet?

1. Wavelet Transform Tutorial

Mathematical transforms are applied to signals to obtain information that is not readily available in the raw signals. Most popular transforms include Fourier transform and wavelet transform. Both two transforms are reversible linear transforms.

a) Fourier Transform

The Fourier Transform is a mathematical operation to transfer the signal from time to the frequency domain. We can obtain frequency information for signal analysis.

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{j2\pi ft} dt$$

Inverse Fourier Transform

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

Discrete Fourier Transform:

$$F(m) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi mn/N}$$

Inverse Discrete Fourier Transform:

$$f(n) = \frac{1}{N}\sum_{m=0}^{N-1} F(m) e^{j2\pi mn/N}$$

The matrix implementation of Discrete Fourier Transform:

$$\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(m) \\ \vdots \\ F(N) \end{bmatrix} = \begin{bmatrix} W_N^{0\times 0} & W_N^{0\times 1} & \cdots & W_N^{0\times n} & \cdots & W_N^{0\times N} \\ W_N^{0\times 1} & & & & & \\ & & & & & \\ & & & W_N^{m\times n} & & \\ & & & & & \\ W_N^{N\times 0} & & \cdots & & \cdots & W_N^{N\times N} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(n) \\ \vdots \\ f(N) \end{bmatrix}$$

where $W_N^{m\times n} = e^{-j2\pi mn/N}$.

The matrix implementation of Discrete inverse Fourier Transform:

$$\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(m) \\ \vdots \\ f(N) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} V_N^{0\times 0} & V_N^{0\times 1} & \cdots & V_N^{0\times n} & \cdots & V_N^{0\times N} \\ V_N^{0\times 1} & & & & & \\ & & & & & \\ & & & V_N^{m\times n} & & \\ & & & & & \\ V_N^{N\times 0} & & \cdots & & \cdots & V_N^{N\times N} \end{bmatrix} \begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(n) \\ \vdots \\ F(N) \end{bmatrix}$$

where $V_N^{m\times n} = e^{j2\pi mn/N}$.

Example:

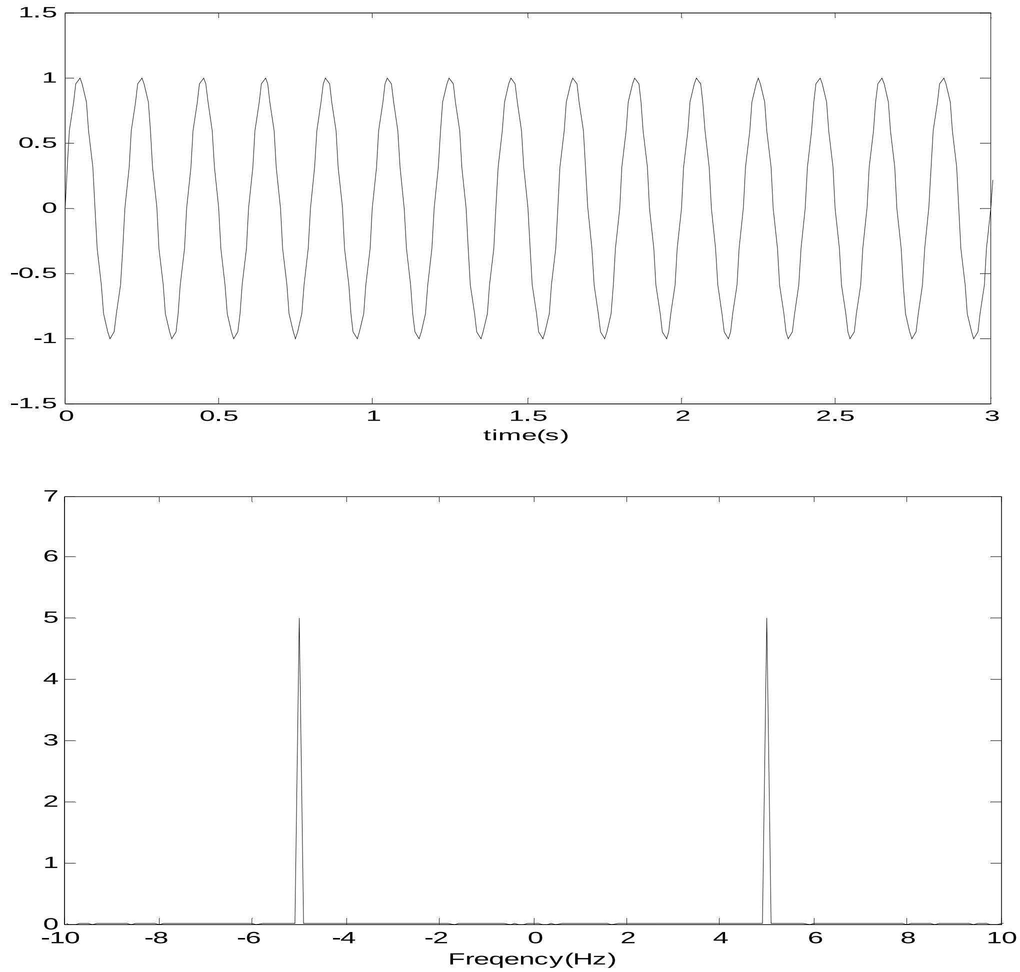


Figure 1

One limitation is that the Fourier transform can't localize the frequency features on time domain and deal ineffectively with non-stationary signals.

b) Short-time Fourier transform (STFT)

STFT is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.

$$X(t, f) = \int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{-j2\pi f\tau}\,d\tau$$

where $w(t)$ is a mask window function.

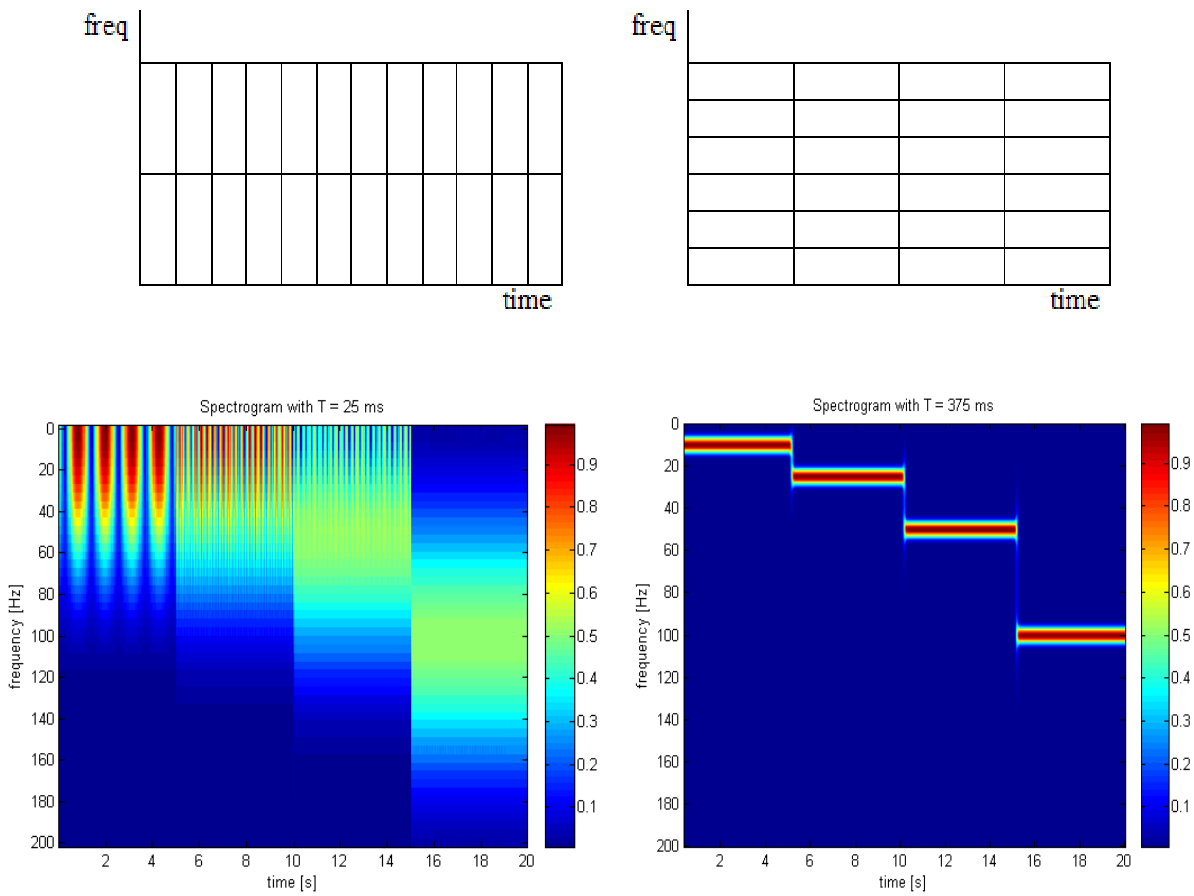


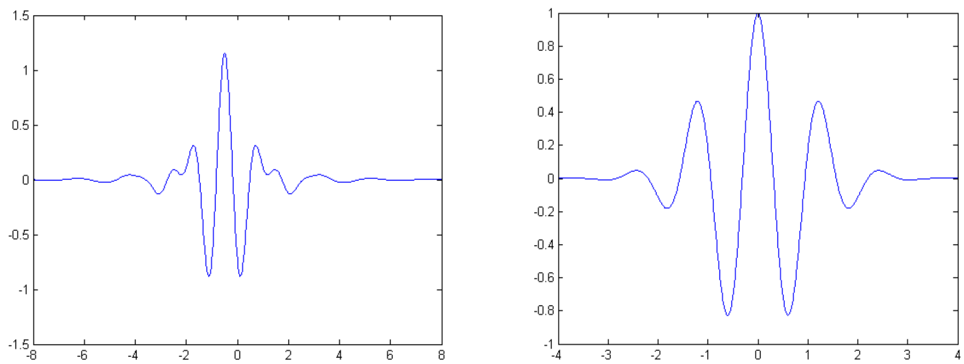
Figure 2

Separate signal in frequency-time domain for fixed resolution. (figure 2)

c) Wavelet Transform

Wavelet transform is to decompose and represent signals by different wavelet functions to extract useful information we need. It has flexible resolution in both time and frequency domains and can localize the information easily.

Wavelet transforms are broadly divided into three classes: continuous, discrete and multiresolution-based.



Meyer

Morlet

Figure 3

- Mother wavelet $\psi(t)$. (figure 3)

- Scaling $\frac{1}{\sqrt{a}}\psi(\frac{t}{a})$ and shifting $\psi(t \pm b)$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$$

i. Continuous Wavelet Transform (CWT)

For time-frequency signal analysis we use CWT.

$$w_{a,b} = \langle \psi_{a,b}, x(t) \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi_{a,b}(\frac{t-b}{a}) dt$$

Inverse CWT:

$$x(t) = \frac{1}{C_{\psi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w_{a,b} \psi_{a,b}(t) \frac{dadb}{a^2}$$

$$where \quad C_{\psi} = \int_{0}^{\infty} \frac{|\varphi(w)|}{w} dw \quad and \quad \int_{-\infty}^{\infty} |\varphi(w)| dw < \infty$$

ii. Discrete Wavelet Transform (DWT)

For implement wavelet transform computation we use DWT.

DWT:

$$w_{m,n} = \langle x(t), \psi_{m,n} \rangle = a_0^{m/2} \int f(t) \psi(a_0^m (t) - nb_0) dt$$

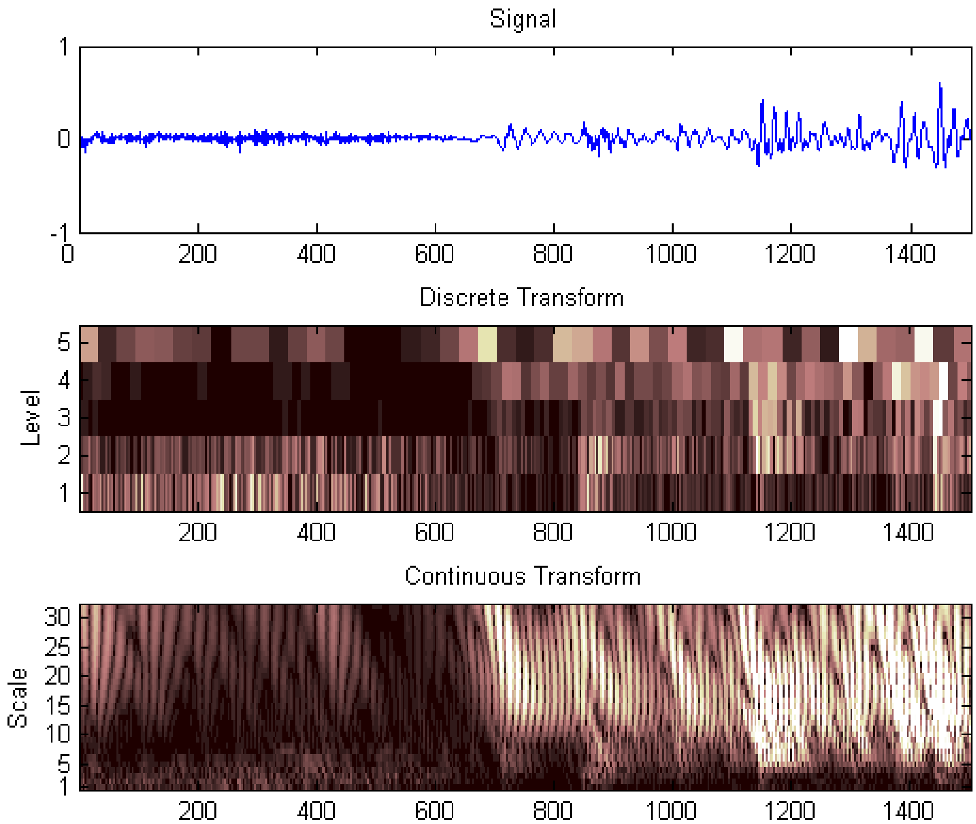
$$\psi_{m,n}(t) = a^{-m/2} \psi(a^{-m} t - nb).$$

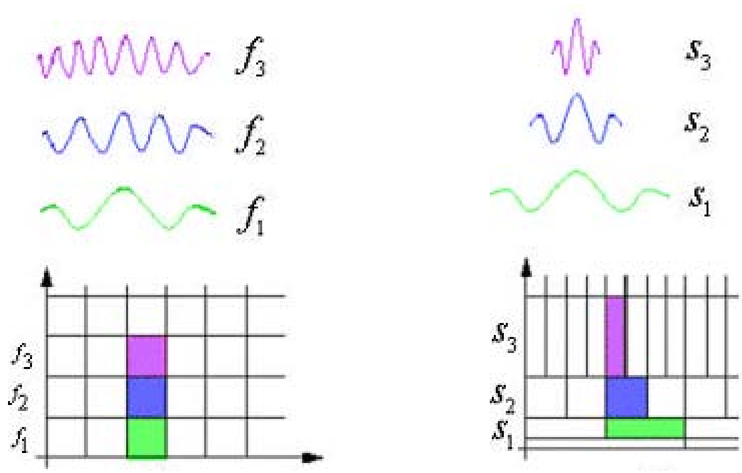
IDWT:

$$x(t) = \sum_m \sum_n w_{m,n} \psi_{m,n}(t)$$

iii. Multiresolution

To reduce the numerical complexity, we use one auxiliary function (farther function) to represent the DWT. And meanwhile we get a multiresolution expression of the signal based on wavelet transform. (figure 4)





(a) STFT (b) DWT

Figure 4

Wavelet Transform:

$$V_{\varphi}(m_0,n)=\sum_{x=0}^{N-1}f(x)\varphi_{m_0,n}(x)$$

$$W_{\psi}(m,n)=\sum_{x=0}^{N-1}f(x)\psi_{m,n}(x)$$

Inverse Wavelet Transform:

$$f(x)=\sum_{n}V_{\varphi}(m_0,n)\varphi_{m_0,n}(x)+\sum_{m=m_0}\sum_{n}W_{\psi}(m,n)\psi_{m,n}(x)$$

Where for DYADIC wavelet transform:

$\varphi_{m,n}(x)=2^{\frac{m}{2}}\varphi(2^m x-n)$ is the scaling function (father function); and

$\psi_{m,n}(x)=2^{\frac{m}{2}}\psi(2^m x-n)$ is the wavelet function (mother function).

And W_m is the orthogonal complement of V_m to the V_{m+1}.

$$V_{m+1}=V_m\oplus W_m$$

The matrix implementation of Discrete Wavelet Transform (Haar):

$$\begin{bmatrix} W(0)\\ W(1)\\ \vdots\\ W(m)\\ \vdots\\ W(N)\end{bmatrix}=\begin{bmatrix}\varphi_{0,0}(x)\\ \hline \psi_{0,0}(x)\\ \hline \psi_{1,0}(x)\\ \psi_{1,1}(x)\\ \hline \psi_{2,0}(x)\\ \psi_{2,1}(x)\\ \psi_{2,2}(x)\\ \psi_{2,3}(x)\\ \hline \cdots\\ \hline \psi_{L-1,0}(x)\\ \psi_{L-1,1}(x)\\ \cdots\\ \psi_{L-1,K-1}(x)\end{bmatrix}\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ f(n)\\ \vdots\\ f(N)\end{bmatrix}=\begin{bmatrix}\multicolumn{4}{c}{\varphi_{0,0}(x)}\\ \hline \multicolumn{4}{c}{\psi_{0,0}(x)}\\ \hline & \psi_{1,0}(x) & 0 & \\ & 0 & \psi_{1,0}(x) & \\ \hline \psi_{2,0}(x) & 0 & 0 & 0\\ 0 & \psi_{2,0}(x) & 0 & 0\\ 0 & 0 & \psi_{2,0}(x) & 0\\ 0 & 0 & 0 & \psi_{2,0}(x)\\ \hline \multicolumn{4}{c}{\cdots}\\ \hline \psi_{L-1,0}(x) & 0 & 0 & 0\\ 0 & \psi_{L-1,0}(x) & 0 & 0\\ 0 & 0 & \cdots & 0\\ 0 & 0 & 0 & \psi_{L-1,0}(x)\end{bmatrix}\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ f(n)\\ \vdots\\ f(N)\end{bmatrix}$$

where $L=\log_2 N,\quad K=N/2$, that is $W=HF$.

The matrix implementation of inverse Discrete Wavelet Transform:

$$F = H^T W$$

An alternative matrix implementation of Discrete Wavelet Transform:

$$W = H_L H_{L-1} \cdots H_2 H_1 F$$

where

$$H_L = \left[\begin{array}{c|c} \varphi_{K-1,0} & \\ \cdots & \\ \varphi_{K-1,M-1} & \\ \hline \psi_{K-1,0} & O \\ \cdots & \\ \psi_{K-1,M-1} & \\ \hline O & I \end{array}\right] \quad \text{and } K = N/2, M = \log_2 N - L, L \in [1, \log_2 N]$$

The alternative matrix implementation of Inverse Discrete Wavelet Transform:

$$F = H_1^T H_2^T \cdots H_{L-1}^T H_L^T W$$

For example, take the Haar wavelet as the base for a signal of 8 samples,

$$\varphi_{0,0} = \frac{1}{\sqrt{8}}[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$$

$$\psi_{0,0} = \frac{1}{\sqrt{8}}[1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad -1 \quad -1]$$

$$\psi_{1,0} = \frac{1}{2}[1 \quad 1 \quad -1 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\psi_{1,1} = \frac{1}{2}[0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad -1 \quad -1]$$

$$\psi_{2,0} = \frac{1}{\sqrt{2}}[1 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\psi_{2,1} = \frac{1}{\sqrt{2}}[0 \quad 0 \quad 1 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\psi_{2,2} = \frac{1}{\sqrt{2}}[0 \quad 0 \quad 0 \quad 0 \quad 1 \quad -1 \quad 0 \quad 0]$$

$$\psi_{2,3} = \frac{1}{\sqrt{2}}[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad -1]$$

The matrix implementation of Discrete Wavelet Transform is

$$
\begin{bmatrix} W(0) \\ W(1) \\ W(2) \\ W(3) \\ W(4) \\ W(5) \\ W(6) \\ W(7) \end{bmatrix} = \begin{bmatrix}
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} \\
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} \\
\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}
\end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}
$$

The matrix implementation of Inverse Discrete Wavelet Transform is

$$
\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix} = \begin{bmatrix}
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & -\frac{1}{2} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & -\frac{1}{2} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\
\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\
\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & 0 & \frac{1}{2} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 \\
\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & 0 & -\frac{1}{2} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{8}} & -\frac{1}{\sqrt{8}} & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{\sqrt{2}}
\end{bmatrix} \begin{bmatrix} W(0) \\ W(1) \\ W(2) \\ W(3) \\ W(4) \\ W(5) \\ W(6) \\ W(7) \end{bmatrix}
$$

For the alternative matrix implementation of DWT and IDWT, we have

$$H_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$H_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to prove that $H = H_3 H_2 H_1$

iv. Filter Bank

In practical filter bank are mostly used for wavelet transform implementation.

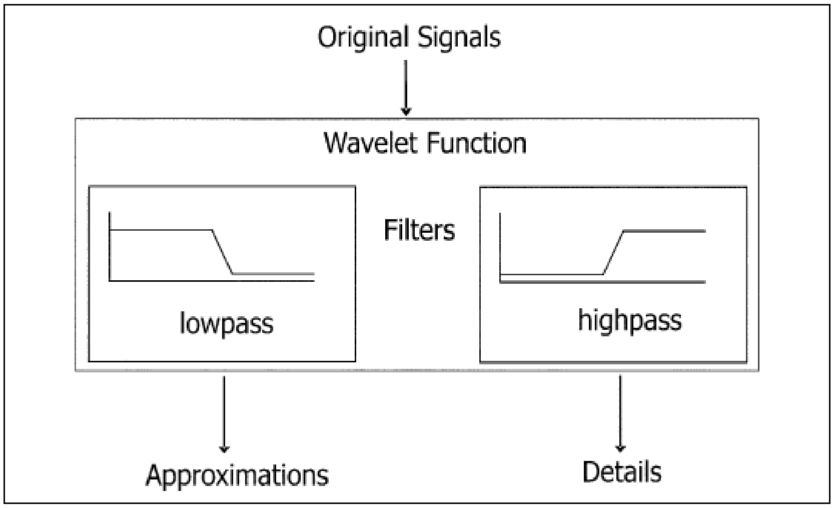
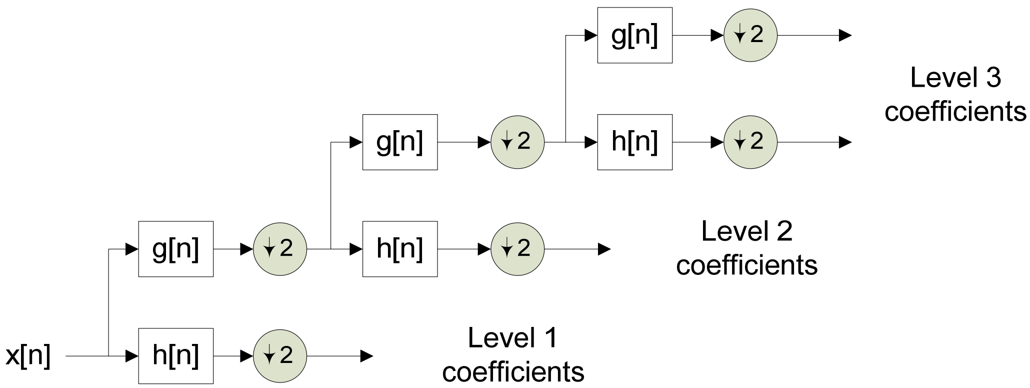


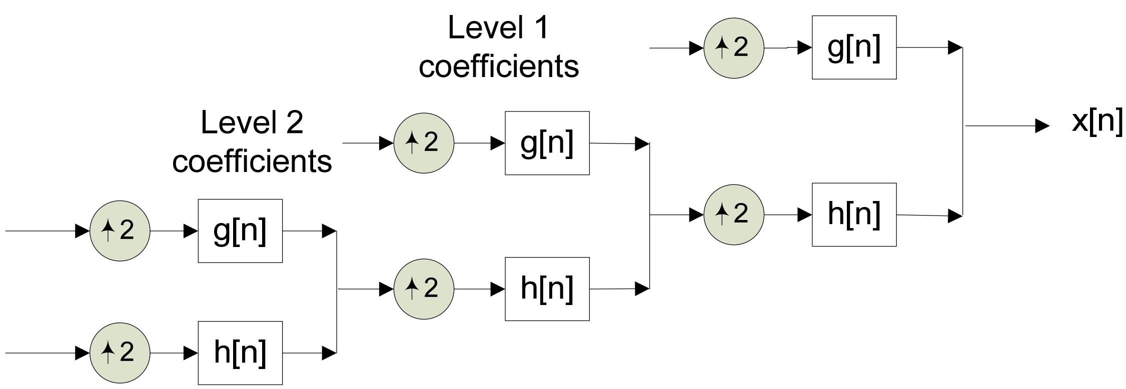
Figure 5

As shown in figure 5, signal Is filtered to low frequency approximations and the high frequency details.

For analysis with orthogonal wavelets the high pass filter is calculated as the quadrature mirror filter (QMF) of the low pass, and reconstruction filters are the time reverse of the decomposition filters. We implement the WT matrix by this.(figure 6). And the reconstruction is as the inverse direction and use the up-sampling operation instead of the down-sampling in the decomposition.



Decomposition



Reconstruction

Figure 6 Implementation of wavelet decomposition and reconstruction with filter bank.

2. EEG Signal de-noising and compression applications with WT.

In addition to signal analysis, based on this features the wavelet transform (WT) could also be used for signal compression and noise removal.

a) De-noising

Base on the wavelet transformation, we can cancel the small coefficients of signal to reduce the noise interferences.

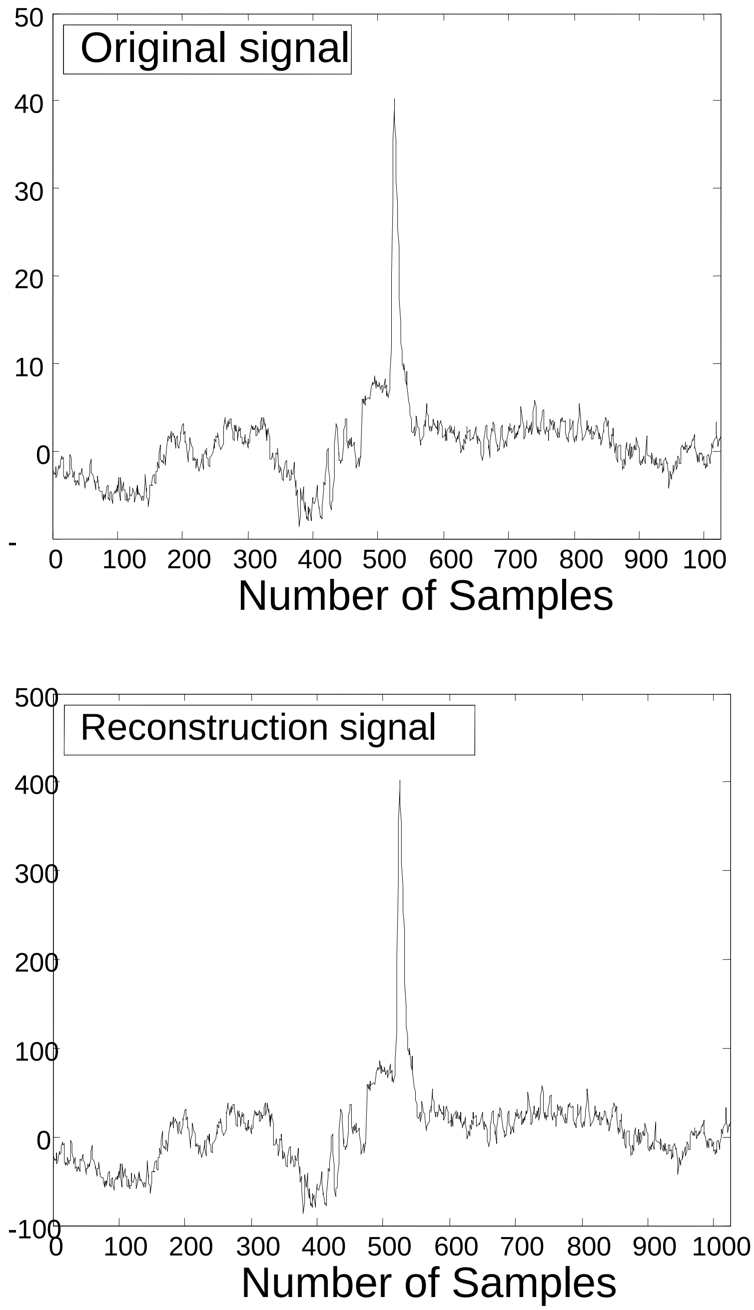


Figure 7 Original and reconstructed EEG signals using wavelets

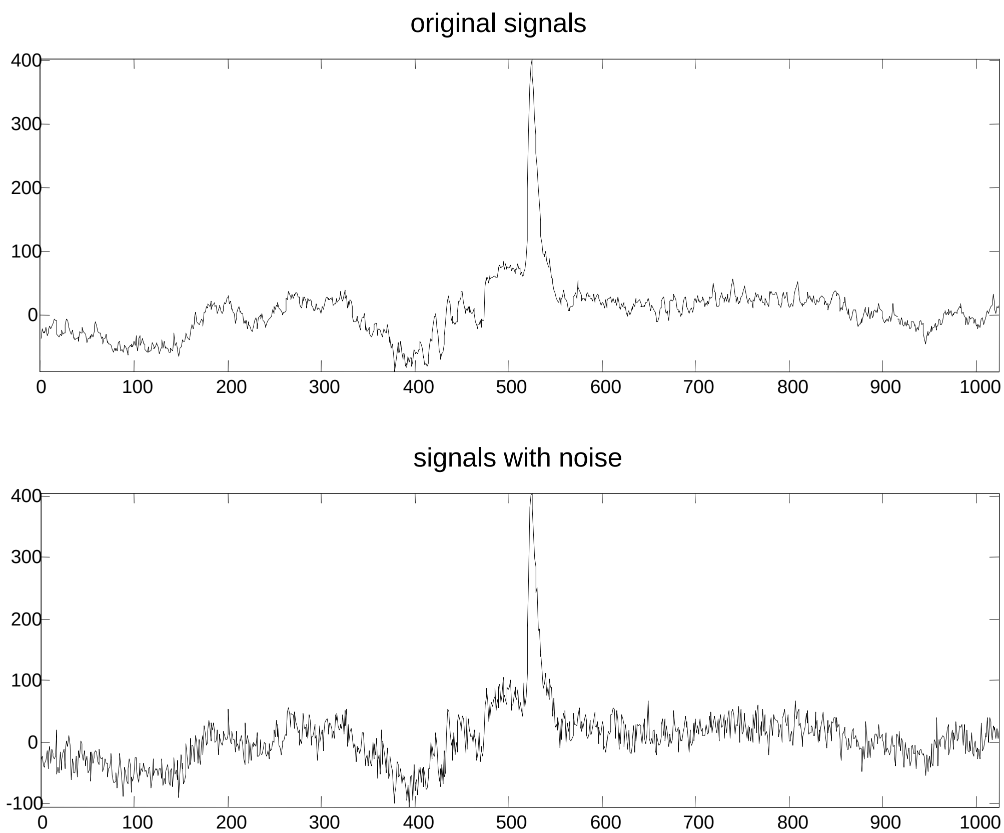


Figure 8 EEG signals: (a) original one. (b) noisy one

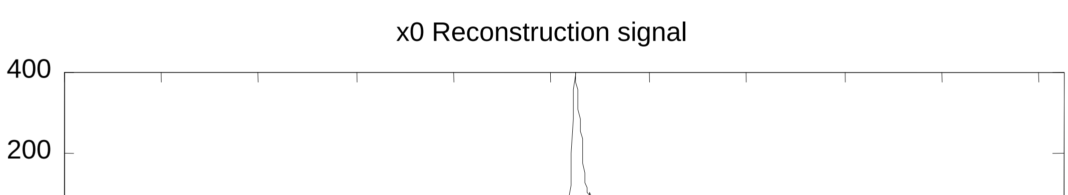


Figure 9 EEG signals after WT de-noising

b) Compression

When signals transferred into wavelets domain they would be sparse or compressible. Therefore we could keep part of the large coefficients to reconstruct the original signals meanwhile without much quality loss. (Figure 10)

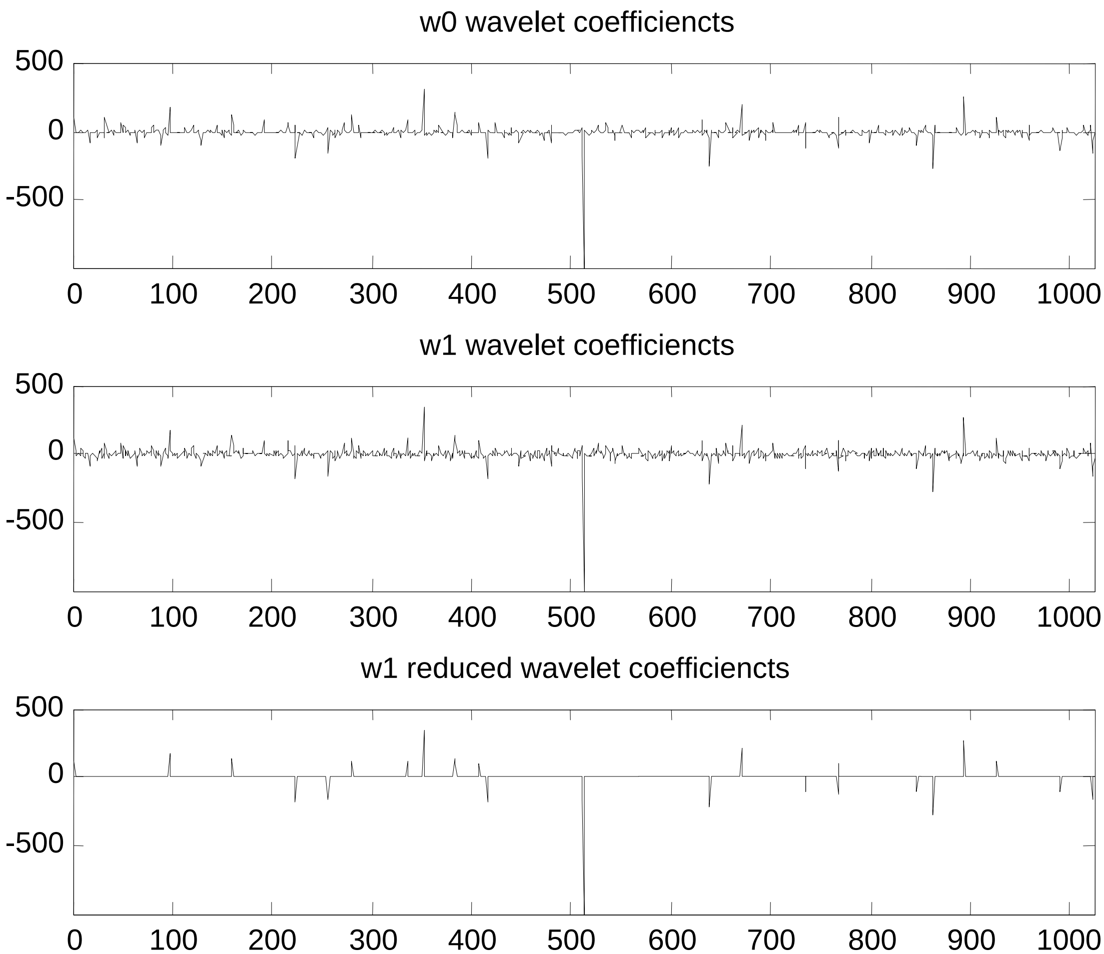


Figure 10 Wavelet coefficients (a) (b) (c)

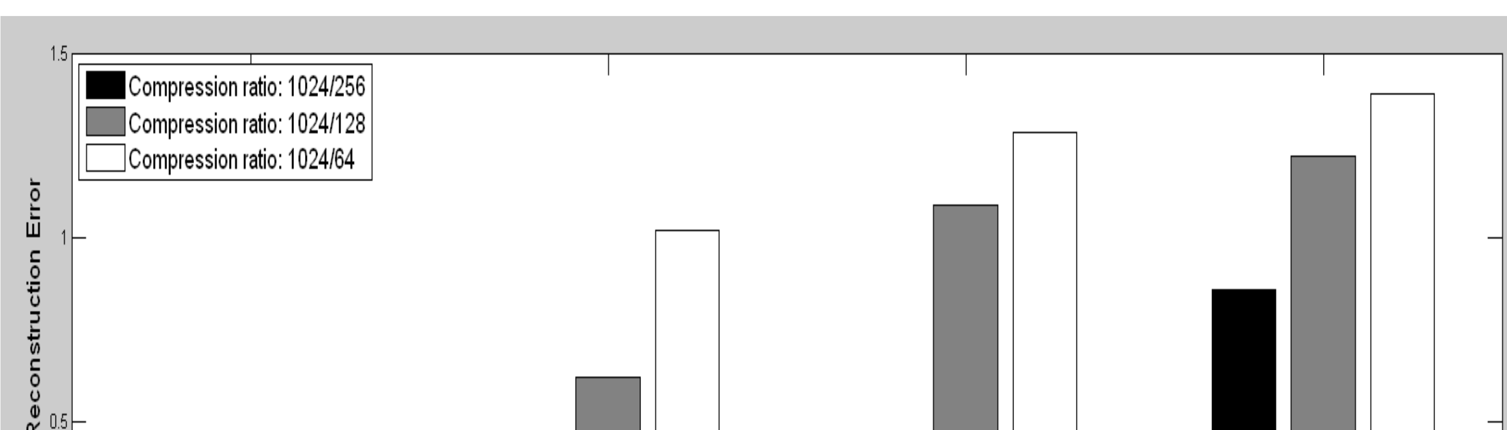


Figure 11 Wavelet reconstruction based on different compression ratio

What do we need in the signal?

3. EEG Signal structure feature extraction

a) Smoothing

Different from other wavelet transform, in this lab we use uniform sampling other than down sampling to see the derivation detail in the whole ECG signal. Then we get the max value for WTMM.

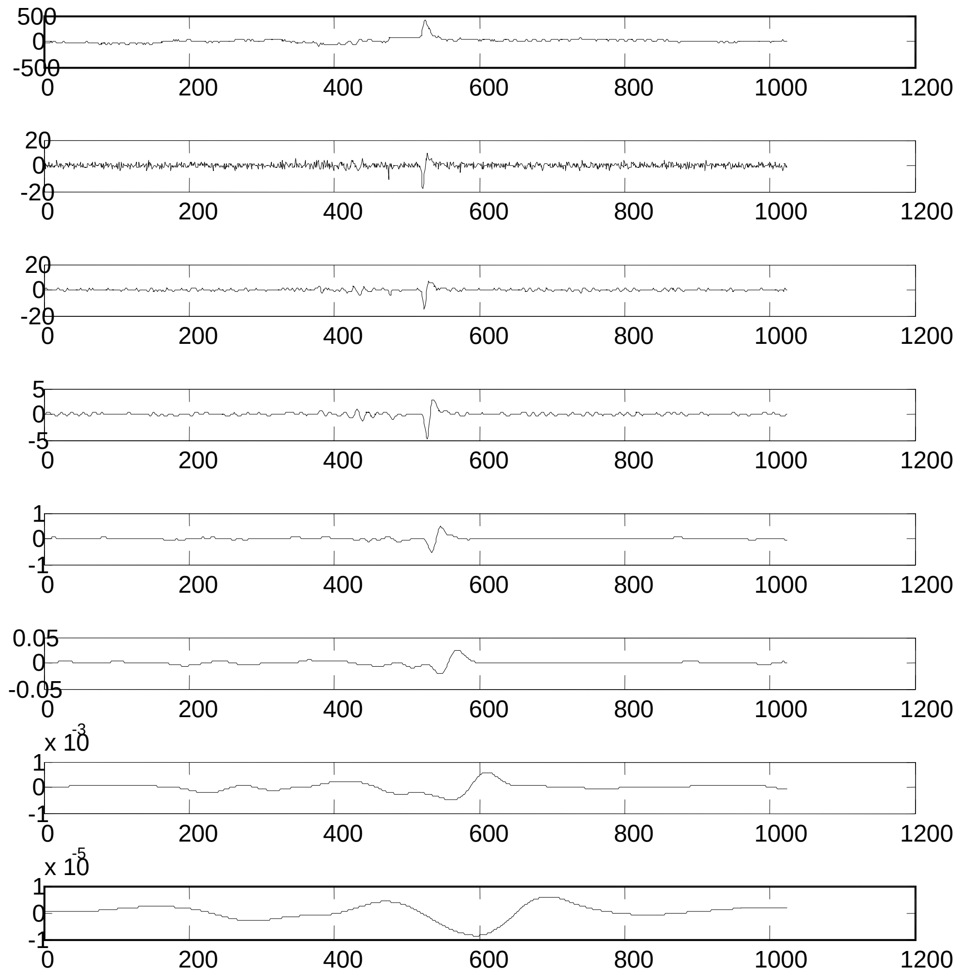


Figure 12 smoothed signal by different scale WT

b) Wavelet Transform Modulus Maxima.

When a mother wavelet is a gradient of a smoothing function, multi-scale gradients can be computed as wavelet transform.

Two wavelet functions are the 1-order and 2-order gradients of a smooth, finite support function □□(x),

$$\psi^a(x) = \frac{d\theta(x)}{dx}, \qquad \psi^b(x) = \frac{d^2\theta(x)}{dx^2}$$

A function with dilation s is denoted by

$$\psi_s(x) = \frac{1}{s}\psi\left(\frac{x}{s}\right).$$

The wavelet transform of a function $f(x)$ is given by

$$W_s^a f(x) = \frac{1}{s} f(x) * \psi^a\left(\frac{x}{s}\right), \quad W_s^b f(x) = \frac{1}{s} f(x) * \psi^b\left(\frac{x}{s}\right),$$

where ‘*’ represents convolution.

Therefore, by detecting the modulus maxima of the wavelet transform of a signal, $f(x)$, the structure information of the signal can be captured.

Use WTMM for the sparse and simple signals, we can detect it and get the information we need meanwhile deduce the size and save the resources.

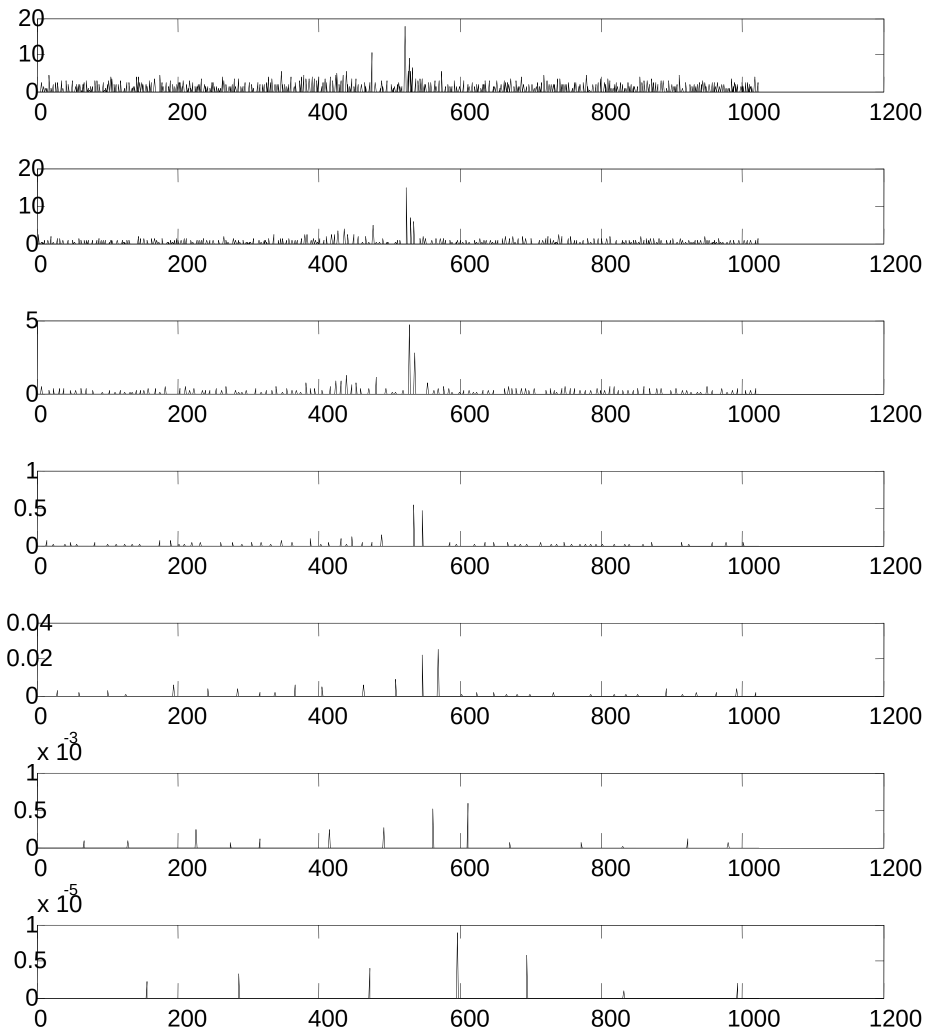


Figure 13 Signal structure captured by WTMM

Figure 13 showed the structure feature extracted by WTMM after multi-scale smoothing.

Matlab Experiments:

Lab 1.1: Write a piece of code that can develop the matrix implementation (in size of 256x256) of DWT and IDWT using the Haar base.

Lab 1.2: Download a piece of EEG signal and perform DWT and IDWT. Compare the wavelet coefficients of different level wavelet transform. Replace some coefficients of small magnitudes with zeros; plot the reconstruction errors with respect the compression ratio.

Lab 1.3: Perform the wavelet transform of a piece of EEG signal and detect the modulus maxima to capture the structure of the EEG signal.

Matlab Code Samples:

1. Build filter function for different wavelets

2. Build wavelet transform matrix

3. Load EEG signals

4. Perform wavelet decomposition and reconstruction

5. WTMM

1. Build filter function for different wavelets

```
function f = MakeONFilter(Type,Par)
% Outputs
% qmf quadrature mirror filter

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'),
   if Par==1,
      f = [   .038580777748   -.126969125396  -.077161555496  ...
            .607491641386   .745687558934   .226584265197   ];
   end
```

…

(Different filter has different coefficients to make the QMF)

2. Build wavelet transform matrix

```
function W = WavMat(h, N, k0, shift)

…

%--make QM filter G

    h=h(:)';  g = fliplr(h .* (-1).^(1:length(h)));

…

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 signals

4. Perform wavelet decomposition and reconstruction

```
load  test_eeg;
n       = 1024;
filter  = MakeONFilter('Daubechies',6)
W1      = WavMat(filter, n, 1);
W       = W1^9;

xx0     = aa(1,1:n)';
w0      = W*xx0;
%w0(find(abs(w0)<100))=0;
x0      = W'*w0;

xx1     = awgn(xx0,10,'measured');
w1      = W*xx1;
w2      = w1;
w2(abs(w1)<100)=0;
x1      = W'*w1;
x2      = W'*w2;
```

5. WTMM.

```
sz=size(wff);
maxmap = zeros(sz);
wff    = abs(wff);
figure

for t=1:m
for k=2:n-1
    kplus=k+1;
    kminus=k-1;
    if wff(t,k) > wff(t,kplus) && wff(t,k) > wff(t,kminus)
    maxmap(t,k) = wff(t,k);
    end
end
subplot(m,1,t);
plot(maxmap(t,:));
end
```

Reference

[1] Stephane Mallat and Sifen Zhong, “Characterization of Signals from Multiscale Edges”, IEEE TRANSACTIONS ON PATIERN ANALYSIS AND MACHINE INTELLIGENCE

[2] Stephane Mallat, A Wavelet Tour of Signal Processing

[3] Ingrid Daubechies, “10 couses on wavelet”

[4] Mallat, S. G.， A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Trans. PAMI, vol. 11, no. 7, July 1989, pp. 674-693.

[5] http://en.wikipedia.org/wiki/Wavelet

[6] http://en.wikipedia.org/wiki/Fourier_transform

[7] Qi Hao, Fei Hu, “ A Compressive Eletroencephalography (EEG) Sensor Design”, 2010 IEEE Sensors