ECE 311 Lab7 report

Name: Yang Shi, NetID: yangshi5

Report Item 1:

Code part:

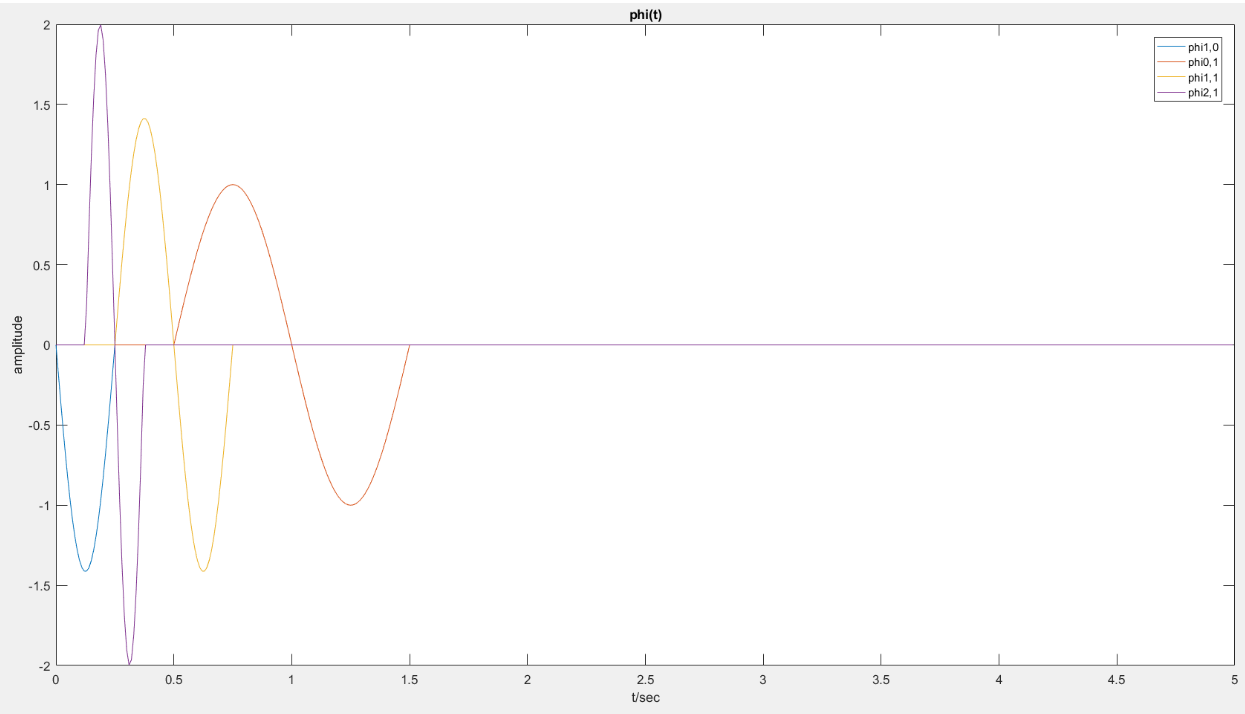
```
%report_item_1
N = 512;
deltat = 0.01;
phi10 = zeros(1, N);
phi01 = zeros(1, N);
phi11 = zeros(1, N);
phi21 = zeros(1, N);
tt = zeros(1, N);
for i = 1 : N
    %phi10
    t = (i-1)*deltat;
    tt(i) = t;
    if abs(2*t) < 0.5
        phi10(i) = power(2, 1/2)*sin(2*pi*(2*t+0.5));
    else
        phi10(i) = 0;
    end
    %phi01
    if abs(t-1) < 0.5
        phi01(i) = sin(2*pi*((t-1)+0.5));
    else
        phi01(i) = 0;
    end
    %phi11
    if abs(2*t-1) < 0.5
        phi11(i) = power(2, 1/2)*sin(2*pi*(2*t-1+0.5));
    else
        phi11(i) = 0;
    end
    %phi21
    if abs(4*t-1) < 0.5
        phi21(i) = 2*sin(2*pi*(4*t-1+0.5));
    else
        phi21(i) = 0;
    end
end

%get magnitude and phase response
```

```
w = fftshift((0:N-1)/N*2*pi);
w(1:N/2) = w(1:N/2)-2*pi;
y10 = fftshift(fft(phi10));
y01 = fftshift(fft(phi01));
y11 = fftshift(fft(phi11));
y21 = fftshift(fft(phi21));

figure(1);
plot(tt(1:500), phi10(1:500));
xlabel('t/sec');
ylabel('amplitude');
title('phi(t)');
hold on;
plot(tt(1:500), phi01(1:500));
hold on;
plot(tt(1:500), phi11(1:500));
hold on;
plot(tt(1:500), phi21(1:500));
legend('phi1,0', 'phi0,1', 'phi1,1', 'phi2,1');

%magnitude plot
figure(2);
plot(w, abs(y10));
xlabel('rad');
ylabel('magnitude');
title('magnitude response of phi1,0');
hold on;
plot(w, abs(y01));
hold on;
plot(w, abs(y11));
hold on;
plot(w, abs(y21));
legend('phi1,0', 'phi0,1', 'phi1,1', 'phi2,1');

%phase response
figure(3);
subplot(4, 1, 1);
plot(w, angle(y10));
xlabel('rad');
ylabel('phase');
title('phase response of phi1,0');
subplot(4, 1, 2);
plot(w, angle(y01));
xlabel('rad');
ylabel('phase');
title('phase response of phi0,1');
subplot(4, 1, 3);
```

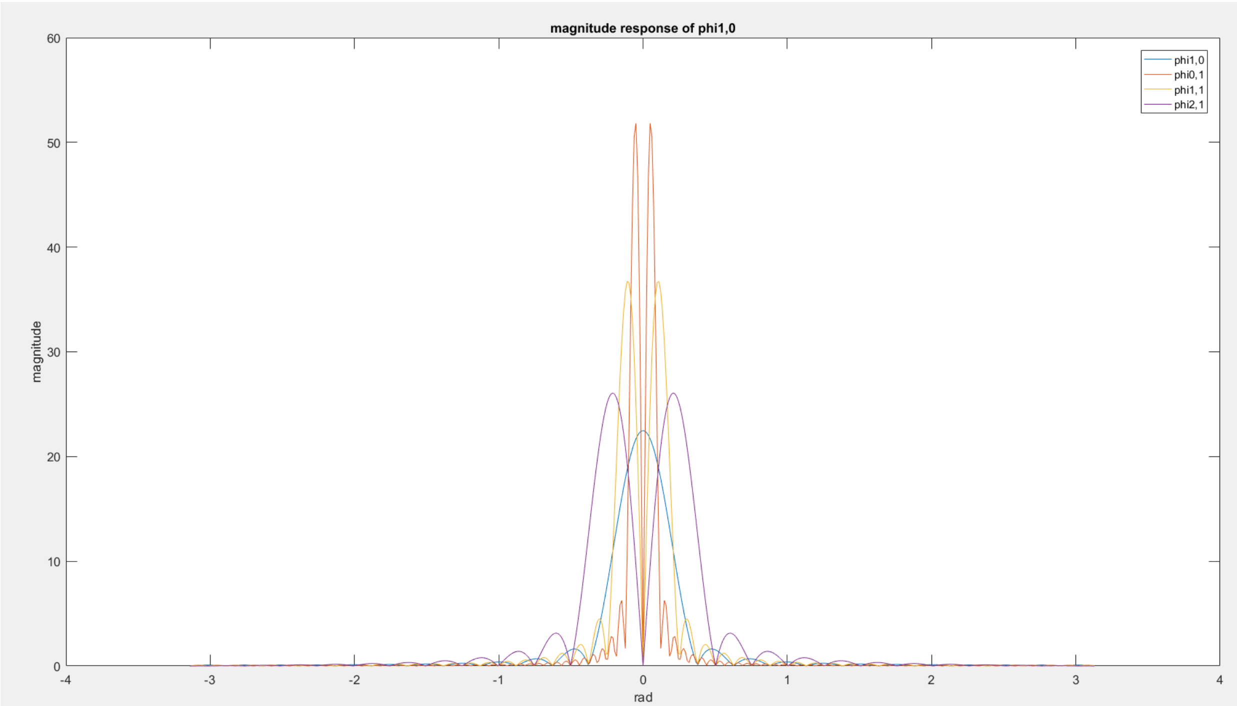
```
plot(w, angle(y11));
xlabel('rad');
ylabel('phase');
title('phase response of phi1,1');
subplot(4, 1, 4);
plot(w, angle(y21));
xlabel('rad');
ylabel('phase');
title('phase response of phi2,1');
```

Explanation part:

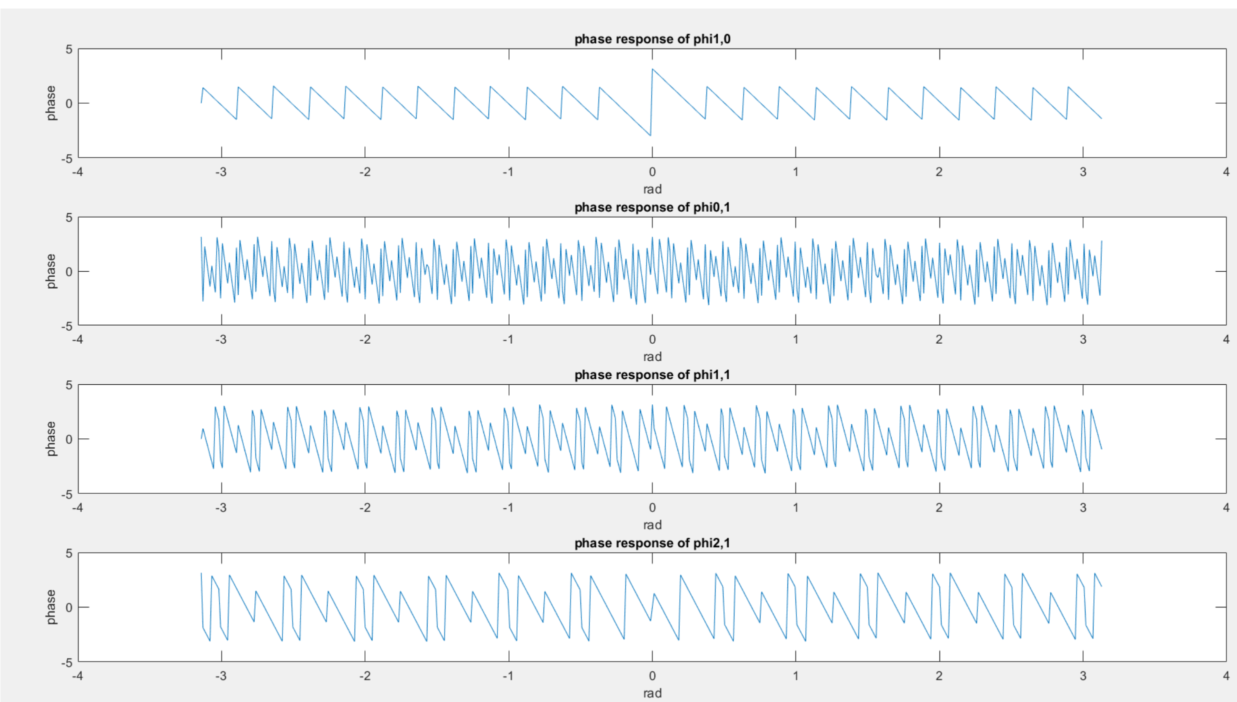
The plots of $\psi_{1,0}(t)$, $\psi_{0,1}(t)$, $\psi_{1,1}(t)$, $\psi_{2,1}(t)$ are plot below:



The plots of the magnitude response are plot below:



The plots of the phase response are plot below:



According to the magnitude response, they are orthogonal with each other because they don't have overlapping Fourier components. The DC coefficient for $\psi_{1,0}(t)$ is 22.48. And the DC coefficients for the other three wavelets are all 0.

Report Item 2:

It’s called corking function because the father wavelet can scale the function in the time domain and it handles the low-frequency components of the waveform. The difference between the wavelet coefficients and the Fourier coefficients is that Fourier coefficients describe the performance of the signal in frequency domain while the wavelet coefficients describe the performance of the signal in both time domain and frequency domain.

Report Item 3:

Code Part:

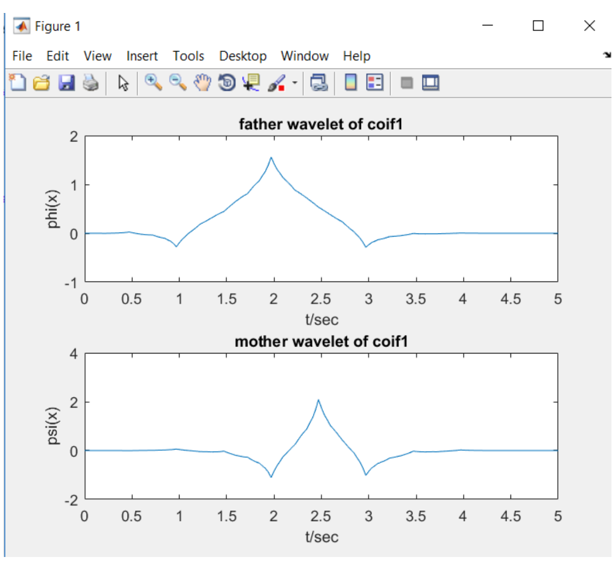
```
%report_item_3
[phi1, psi1, t1] = wavefun('coif1', 5);
[phi2, psi2, t2] = wavefun('db1', 5);
[phi3, psi3, t3] = wavefun('sym4', 5);

figure(1);
subplot(2, 1, 1);
plot(t1, phi1);
xlabel('t/sec');
ylabel('phi(x)');
title('father wavelet of coif1');
subplot(2, 1, 2);
plot(t1, psi1);
xlabel('t/sec');
ylabel('psi(x)');
title('mother wavelet of coif1');

figure(2);
subplot(2, 1, 1);
plot(t2, phi2);
xlabel('t/sec');
ylabel('phi(x)');
title('father wavelet of db1');
subplot(2, 1, 2);
plot(t2, psi2);
xlabel('t/sec');
ylabel('psi(x)');
title('mother wavelet of db1');

figure(3);
subplot(2, 1, 1);
plot(t3, phi3);
```

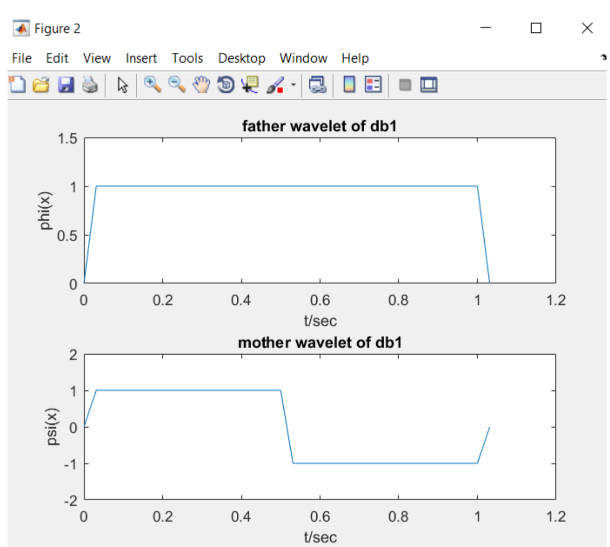
```
xlabel('t/sec');
ylabel('phi(x)');
title('father wavelet of sym4');
subplot(2, 1, 2);
plot(t3, psi3);
xlabel('t/sec');
ylabel('psi(x)');
title('mother wavelet of sym4');

N = 512;
N1 = t1(length(t1));
N2 = t2(length(t2));
N3 = t3(length(t3));

yphi1 = fftshift(fft(phi1, N));
ypsi1 = fftshift(fft(psi1, N));
yphi2 = fftshift(fft(phi2, N));
ypsi2 = fftshift(fft(psi2, N));
yphi3 = fftshift(fft(phi3, N));
ypsi3 = fftshift(fft(psi3, N));

fs1 = N/N1;
fs2 = N/N2;
fs3 = N/N3;

w = fftshift((0:N-1)/N*2*pi);
w(1:N/2) = w(1:N/2)-2*pi;
w = w/(2*pi)*fs1;

figure(4);
subplot(2, 1, 1);
plot(w, abs(yphi1));
xlim([-fs1/2 fs1/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of father wavelet of coif1');
subplot(2, 1, 2);
plot(w, abs(ypsi1));
xlim([-fs1/2 fs1/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of mother wavelet of coif1');

w = w/fs1*fs2;
figure(5);
subplot(2, 1, 1);
plot(w, abs(yphi2));
```

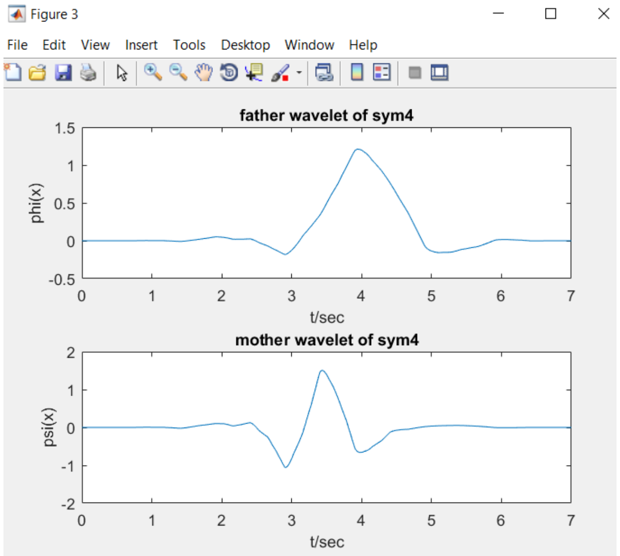
```
xlim([-fs2/2 fs2/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of father wavelet of db1');
subplot(2, 1, 2);
plot(w, abs(ypsi2));
xlim([-fs2/2 fs2/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of mother wavelet of db1');

w = w/fs2*fs3;
figure(6);
subplot(2, 1, 1);
plot(w, abs(yphi3));
xlim([-fs3/2 fs3/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of father wavelet of sym4');
subplot(2, 1, 2);
plot(w, abs(ypsi3));
xlim([-fs3/2 fs3/2]);
xlabel('frequency (Hz)');
ylabel('magnitude spectrum');
title('magnitude spectrum of mother wavelet of sym4');
```

Explanation part:

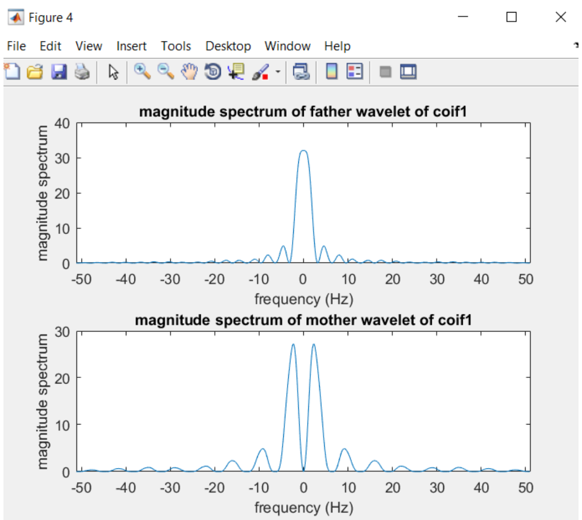
The plots of father and mother wavelet with types: coif1, db1 and sym4 are plot below.

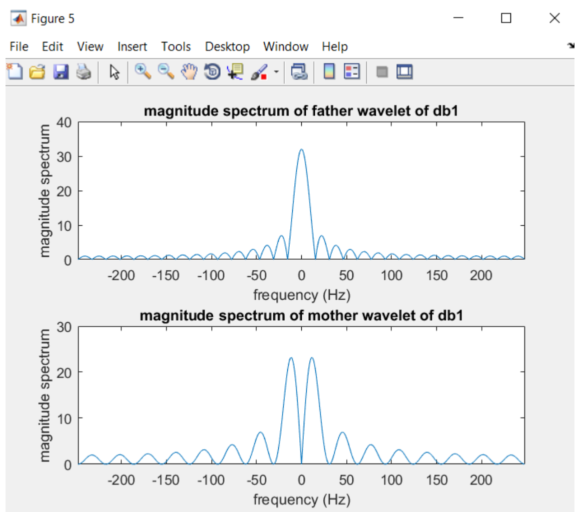


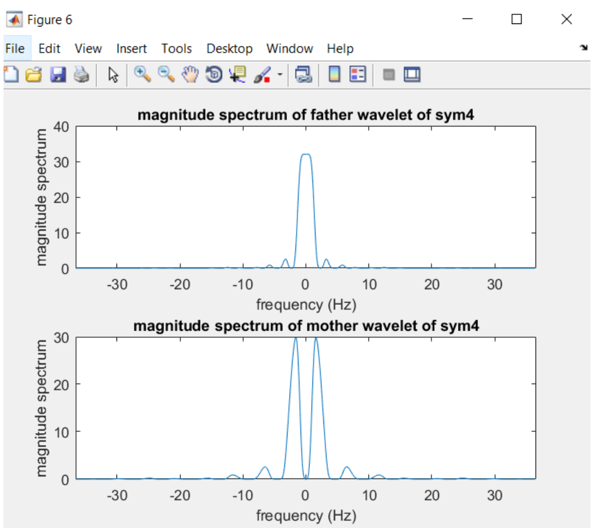




The magnitude spectrum of each wavelet is plot below:







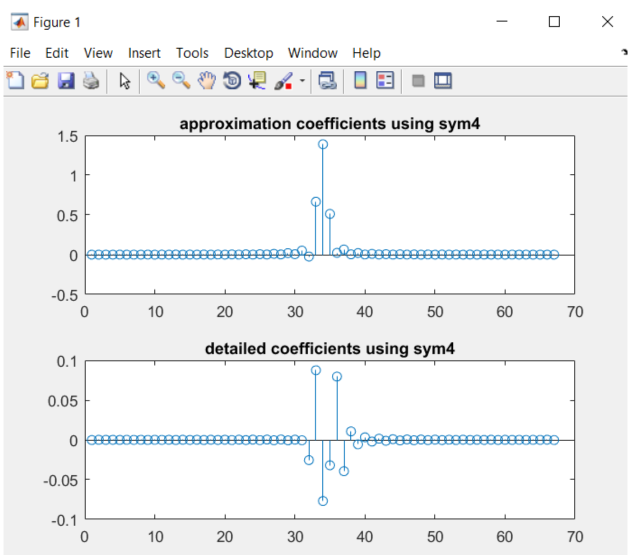
Report Item 4:

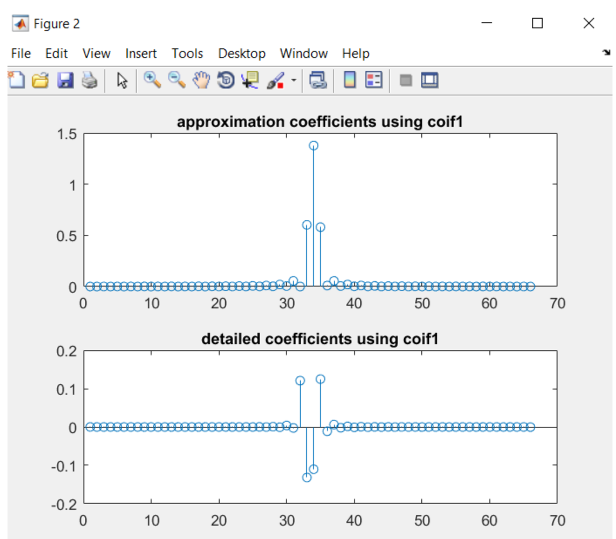
Code Part:

```
%report_item_4
load 'signal.mat';
[ca1, cd1] = dwt(x, 'sym4');
[ca2, cd2] = dwt(x, 'coif1');

figure(1);
subplot(2, 1, 1);
stem(ca1);
title('approximation coefficients using sym4');
subplot(2, 1, 2);
stem(cd1);
title('detailed coefficients using sym4');

figure(2);
subplot(2, 1, 1);
stem(ca2);
title('approximation coefficients using coif1');
subplot(2, 1, 2);
stem(cd2);
title('detailed coefficients using coif1');
```

Explanation part:

The plots of set of coefficients are shown below:





Report Item 5:

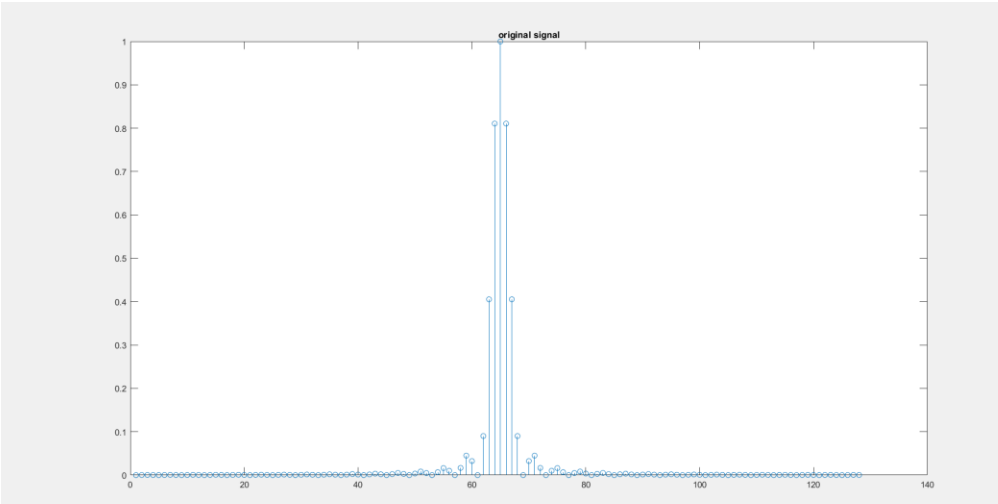
Code part:

```
%report_item_5
load 'signal.mat';
[ca1, cd1] = dwt(x, 'sym4');
[ca2, cd2] = dwt(x, 'coif1');
x1 = idwt(ca1, cd1, 'sym4');
x2 = idwt(ca2, cd2, 'coif1');

meanca1 = mean(abs(ca1));
meancd1 = mean(abs(cd1));
meanca2 = mean(abs(ca2));
meancd2 = mean(abs(cd2));

N1 = length(ca1);
N2 = length(ca2);
for i = 1 : N1
    if abs(ca1(i)) < meanca1
        ca1(i) = 0;
    end
    if abs(cd1(i)) < meancd1
        cd1(i) = 0;
    end
end
for i = 1 : N2
    if abs(ca2(i)) < meanca2
        ca2(i) = 0;
    end
    if abs(cd2(i)) < meancd2
        cd2(i) = 0;
    end
end

x1new = idwt(ca1, cd1, 'sym4');
x2new = idwt(ca2, cd2, 'coif1');

figure(1);
stem(x);
title('original signal');

figure(2);
subplot(2, 1, 1);
stem(x1);
title('idwt using sym4');
subplot(2, 1, 2);
stem(x2);
```

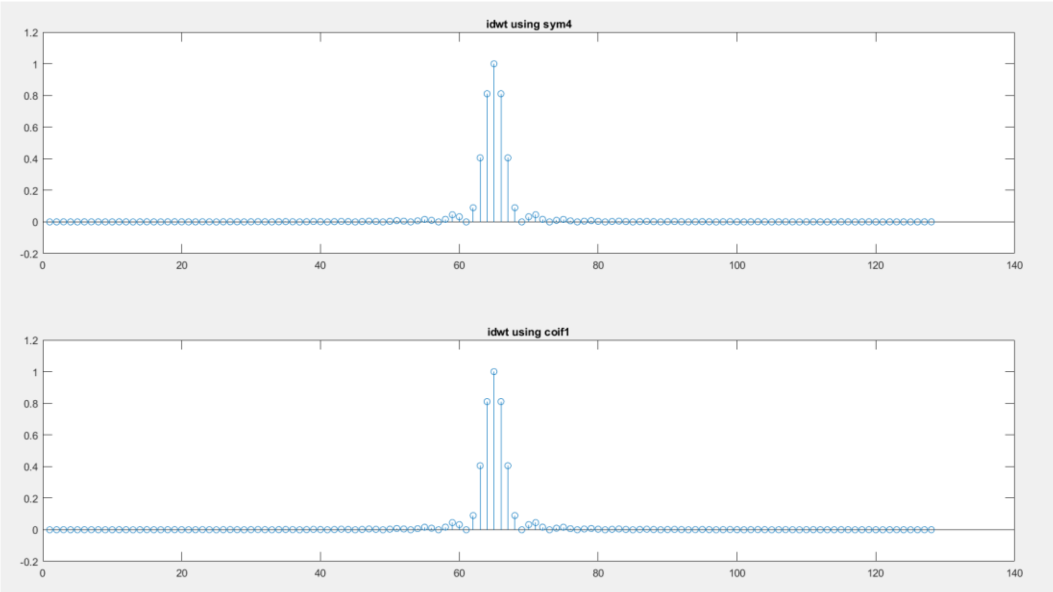
```
title('idwt using coif1');

figure(3);
subplot(2, 1, 1);
stem(x1new);
title('idwt using sym4 after deleting small coeffs');
subplot(2, 1, 2);
stem(x2new);
title('idwt using coif1 after deleting small coeffs');
```

Explanation part:

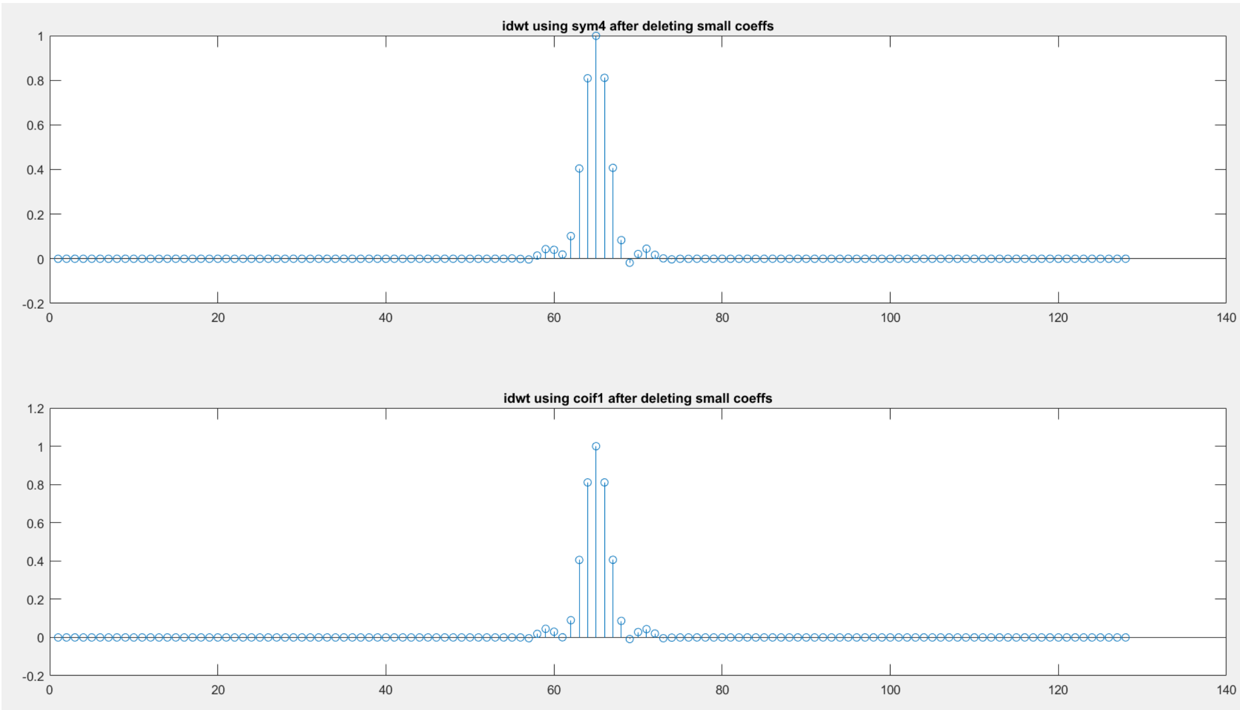
The figure 1 below is the original signal:



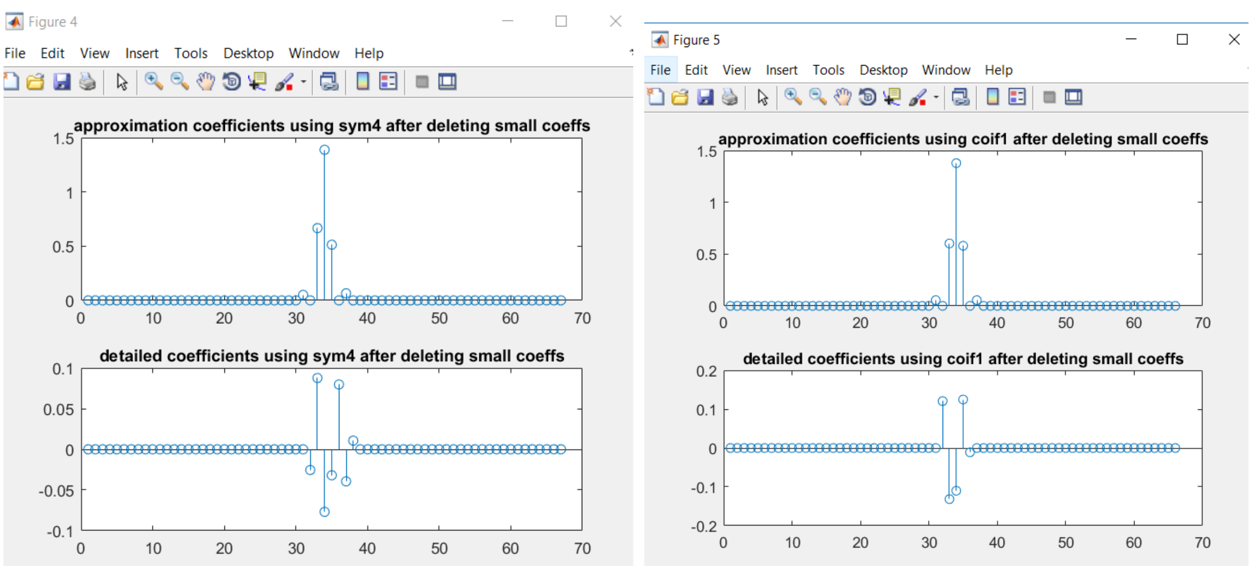
The figure 2 below is the signal reconstructed using idwt for sym4 and coif1 type:



The figure 3 below is the signal reconstructed using idwt for sym4 and coif1 type after cancelling out the small coefficients:



The figure 4 below is the set of coefficients usgin sym4 and coif1:



Report Item 6:

Code part:

```
clc, clear all, close all

[xorig,fs] = audioread('original.wav');
[xnoise,fs] = audioread('noisy.wav');

wname = 'db1';
a1thresh = 0.0; % Must be positive.
d1thresh = 0.0; % Must be positive.
a2thresh = 0.0; % Must be positive.
d2thresh = 0.0; % Must be positive.
a3thresh = 0.0; % Must be positive.
d3thresh = 0.0; % Must be positive.
a4thresh = 0.0; % Must be positive.
d4thresh = 0.0; % Must be positive.
a5thresh = 0.0; % Must be positive.
d5thresh = 0.0; % Must be positive.

% DWT
x = xnoise;
[a1,d1] = dwt(x,wname);
[a2,d2] = dwt(a1,wname);
[a3,d3] = dwt(a2,wname);
[a4,d4] = dwt(a3,wname);
[a5,d5] = dwt(a4,wname);

% Inverse DWT
N = 500;
number = 0;
min = 1000000;
for i = 1 : N
    a1thresh = (i-1)/N;%change this line through a1 to d5
independently
    a5_ = (abs(a5)>a5thresh).*a5;
    d5_ = (abs(d5)>d5thresh).*d5;
    a4t = idwt(a5_,d5_,wname);
    a4_ = (abs(a4t)>a4thresh).*a4t;
    d4_ = (abs(d4)>d4thresh).*d4;
    a3t = idwt(a4_,d4_,wname);
    a3_ = (abs(a3t)>a3thresh).*a3t;
    d3_ = (abs(d3)>d3thresh).*d3;
    a2t = idwt(a3_,d3_,wname);
    a2_ = (abs(a2t)>a2thresh).*a2t;
    d2_ = (abs(d2)>d2thresh).*d2;
    a1t = idwt(a2_,d2_,wname);
```

```
    a1_ = (abs(a1t)>a1thresh).*a1t;
    d1_ = (abs(d1)>d1thresh).*d1;
    x_ = idwt(a1_,d1_,wname);
    E = norm(x_ - xorig)/norm(xorig)*100;
    if E < min
        min = E;
        number = (i-1)/N;
    end
end
    disp(['Percent Relative Error: ', num2str(min)]);
    disp(['threshold: ', num2str(number)]);
```

Explanation part:

Use the provided code and change only one threshold value gradually from 0 to 1 for each time. And record the threshold value that generated the smallest percent relative error. And then do that for all the threshold value and use these values in the waveletdenoise.m

Chosen threshold value: a1thresh = 0.02, d1thresh = 0.01, a2thresh = 0.02, d2thresh = 0.014, a3thresh = 0.02, d3thresh = 0.018, a4thresh = 0.016, d4thresh = 0.02, a5thresh = 0.006, d5thresh = 0.02

The percent relative error is 7.8351.