# Project 7: Wavelet Transform

## Contents

## Course No: ECE 5256

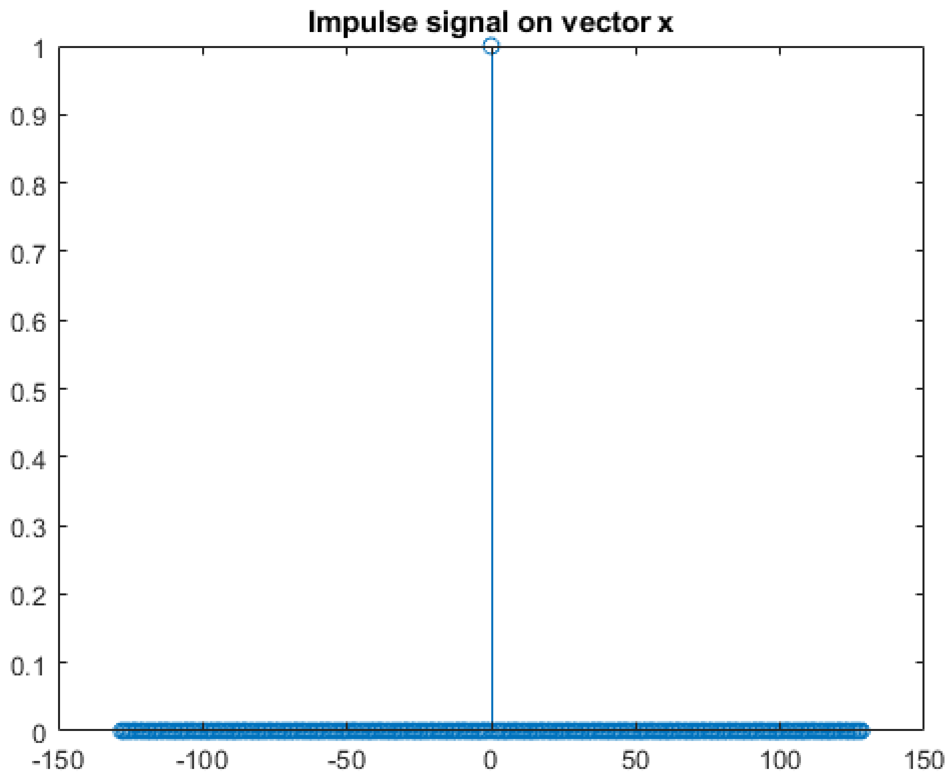
## Due Date: 3/28/2021

## Q1. Place an impulse in a vector x, and perform the inverse WT for different positions of the impulse to see wavelets and different scales and shifts. This will show what scale and shift of the wavelet transform corresponds to what particular location in the wavelet domain. For example, for a length 256 input, locations 129-256 correspond to the smallest scale, 65-128 to the next smallest, 33-64 to the next and so on. Verify that two adjacent impulses at the smallest scale corresponds to a wavlet shift of 2 pixels when the inverse transform is applied. Verify that at the next smallest scale the adjacent impulses correspond to a four-pixel shift. Next scale gives an 8-pixels shift and the next gives a 16-pixel shift. At every larger scale the wavelet will have more detail. Do this for four different levels. Use the wavelet ‘db2’ or similar and the commant four different scales and notice the increase in detail. Use the Matlab command waverec.m.

## Create a vector x

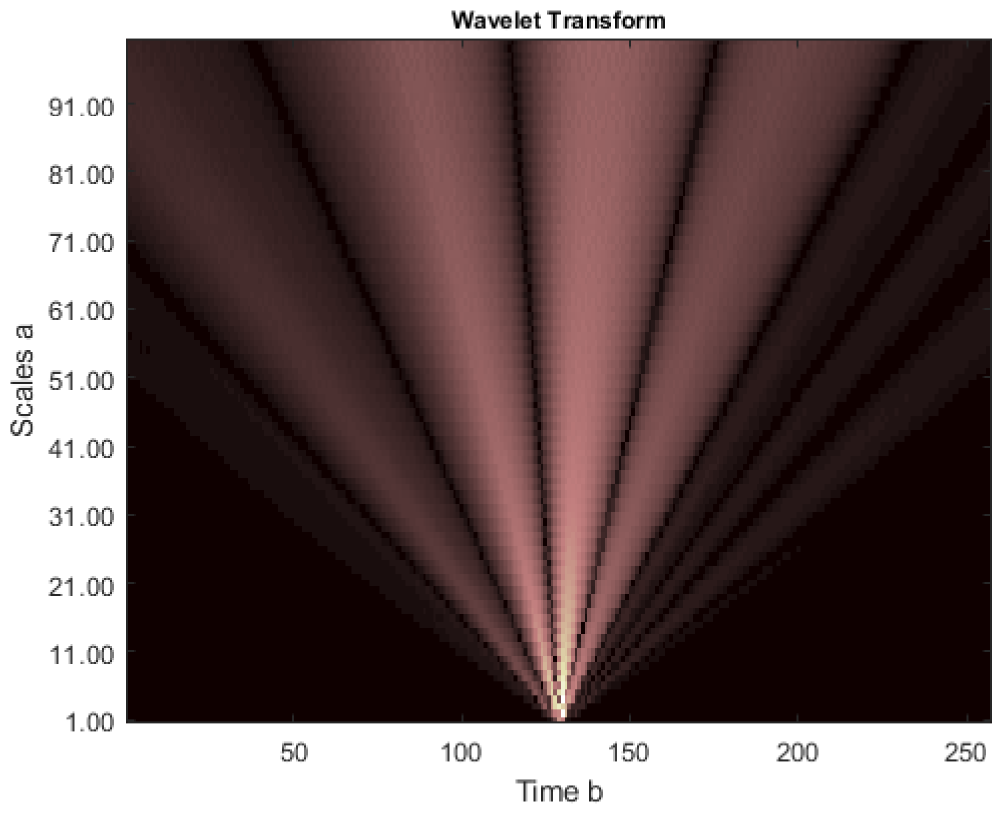
```
x=-128:1:128; %Time series
impulse=[zeros(1,128),ones(1,1),zeros(1,128)];
```

```
stem(x,impulse);
title("Impulse signal on vector x");
```

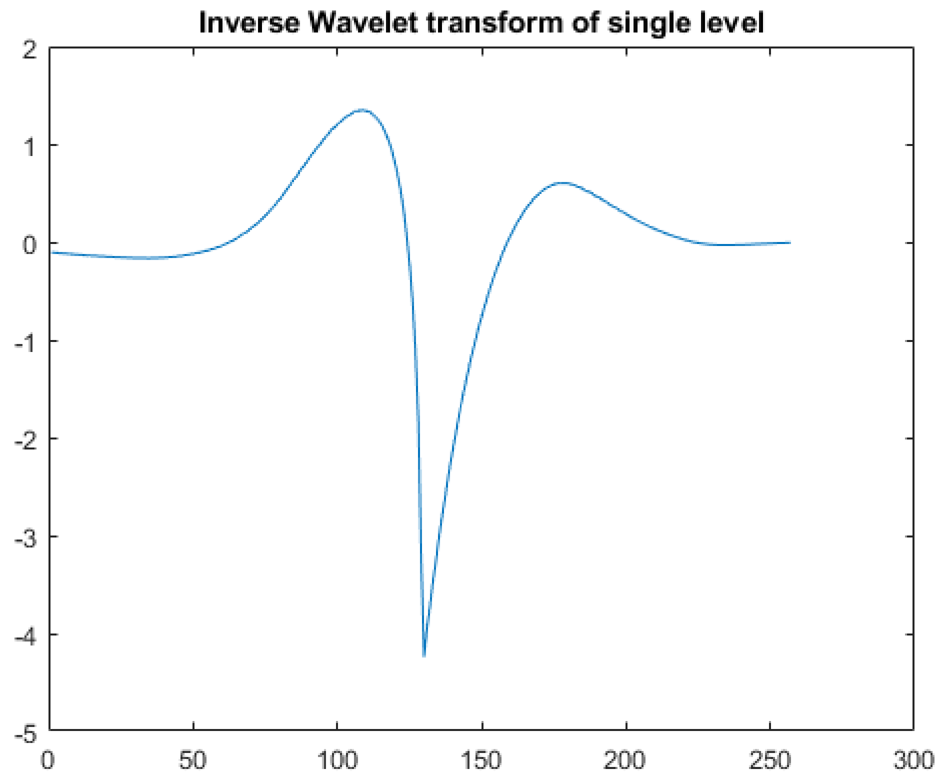


Apply Wavelet Transform

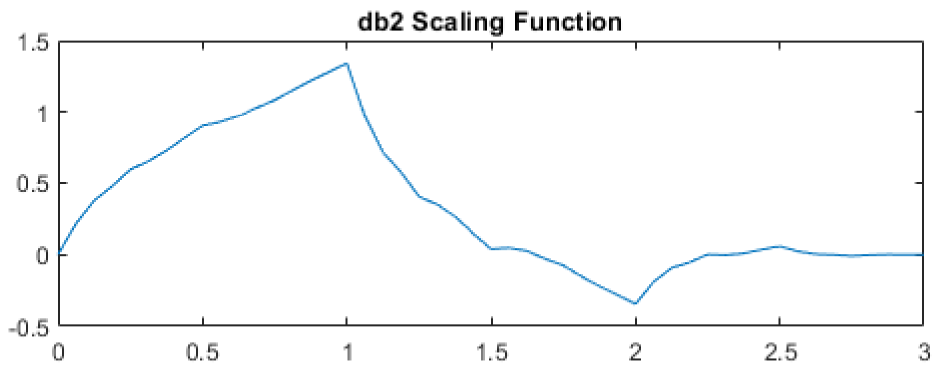
```
scales = 1:100;
wt = cwt(impulse,scales,'sym5','plot');
title("Wavelet Transform");
```

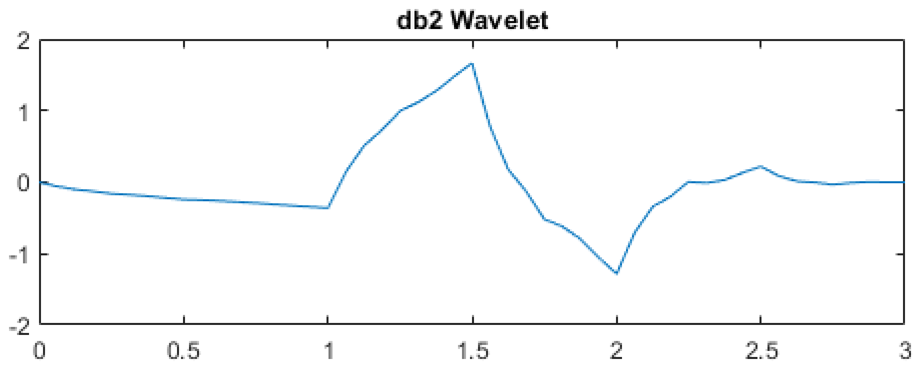


```
%Apply inverse wavelet transform
iwt=icwt(wt);
plot(iwt);
title("Inverse Wavelet transform of single level");
```



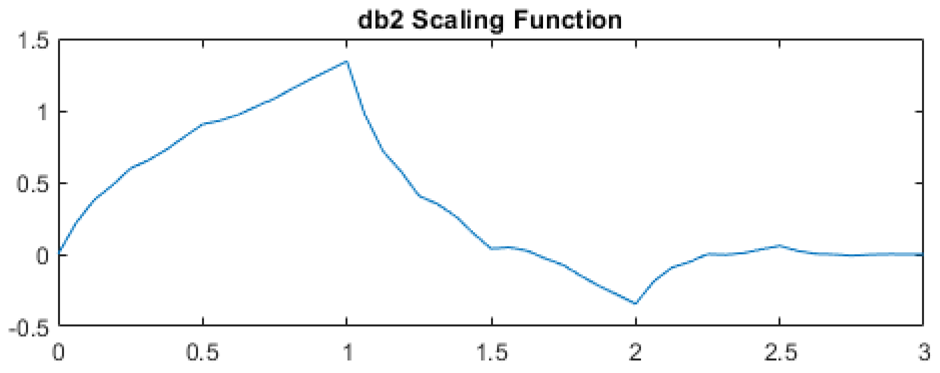
```
[phi,psi,xval] = wavefun('db2',4);
subplot(211);
plot(xval,phi);
title('db2 Scaling Function');
subplot(212);
plot(xval,psi);
title('db2 Wavelet');
```

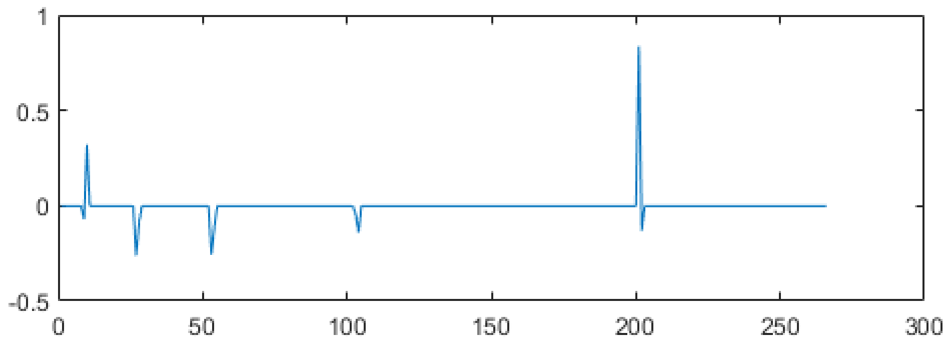




## Find the wavelet transform

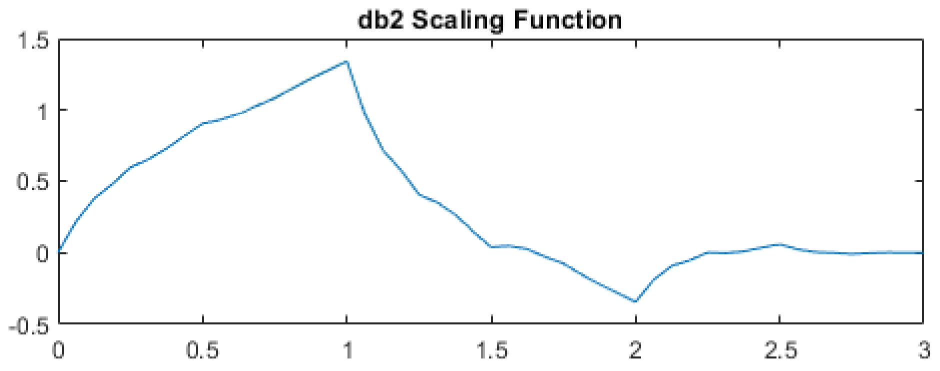
```
wv = 'db2';
[c,l] = wavedec(impulse,4,wv);
plot(c)
```

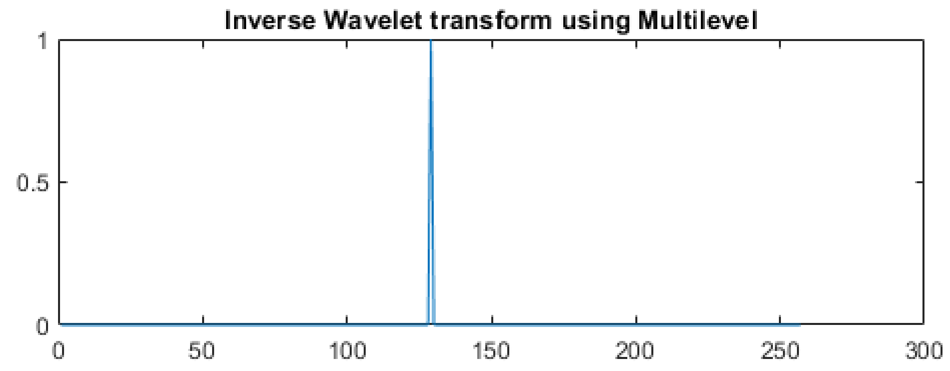




## Find Inverse wavelet using waverec

```
a = waverec(c,l,wv);
plot(a);
title("Inverse Wavelet transform using Multilevel");
```





## Verify that two adjacent impulses at the smallest scale

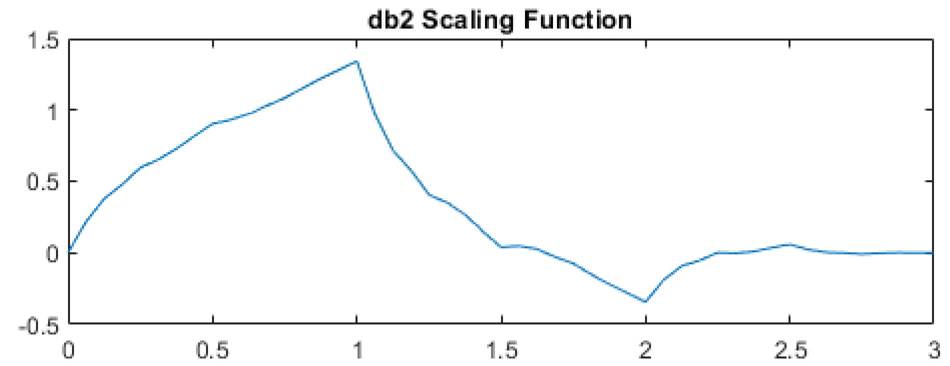
```
if a==18
    a=N/2;
elseif a==33
    a=N/4;
elseif a==66
    a=N/8;
elseif a==130
    a=N/16;
end
```

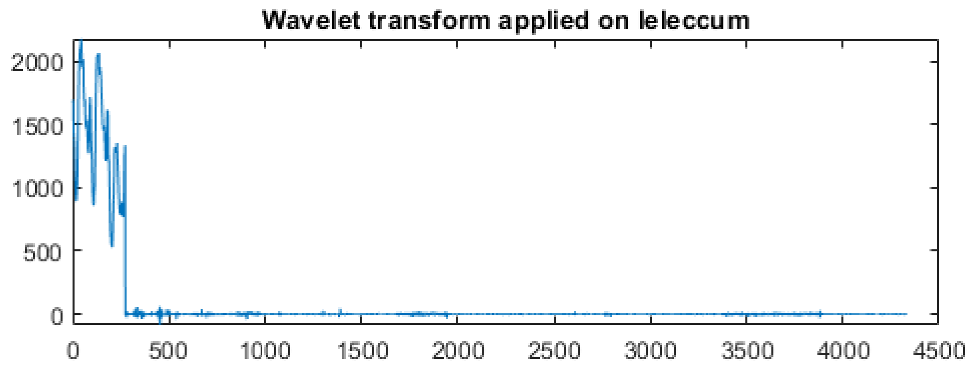
## Q2. Take the Wavelet transform of a signal. Something with a range of frequencies as in a short pulse, or pulses. Retain 25% of its largest coefficients (set the rest to 0) and reconstruct the signal. Do the same using the Fourier Transform and compute the MSE between the reconstructed signal and the original. Repeat for the top 10% of coefficients. Generally, the WT will give a lower MSE.

### Wavelet Transform

Create a signal

```
load leleccum
wv = 'db2';
[u,v] = wavedec(leleccum,4,wv);
plot(u);
title("Wavelet transform applied on leleccum");
```





### Retain 25% of its largest coefficients (set the rest to 0)for wavelet transform and reconstruct the signal.

```
u(1:1082) = zeros(size(u(1:1082)));
w = waverec(u,v,wv);
```

### Fourier Transform

```
f=fft(leleccum);
```

### Retain 25% of its largest coefficients for fourier transform(set the rest to 0) and reconstruct the signal.

```
f(1:864) = zeros(size(f(1:864)));
i_f = ifft(f);
```

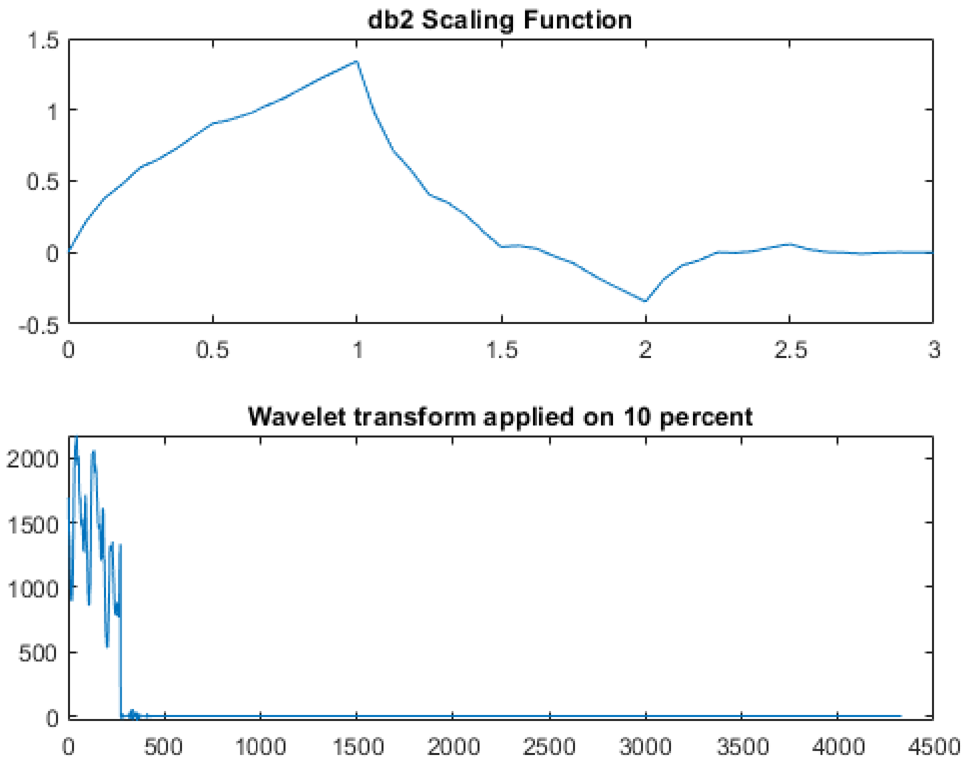
### MSE

```
m = (w-leleccum).^2/4320;%MSE for wavelet = 4.587488451398322e-11
min(m);
mf =(i_f-leleccum).^2/4320;%MSE for fourier = 3.8016
min(mf);
```

### Repeat for the top 10% of coefficients for wavelet

```
[u_t,v_t] = wavedec(leleccum,4,wv);
u_t(433:4329) = zeros(size(u_t(433:4329)));
w_t = waverec(u_t,v_t,wv);
```

```
plot(u_t);
title("Wavelet transform applied on 10 percent");
```



## Repeat for the top 10% of coefficients for wavelet

```
f_t=fft(leleccum);
```

```
f_t(432:4320) = zeros(size(f_t(432:4320)));
i_f_t = ifft(f_t);
```

## MSE for top 10 percent

```
m_t = (w_t-leleccum).^2/4320; %MSE for wavelet = 4.5874884513983 22e-11
min(m_t);
mf_t =(i_f_t-leleccum).^2/4320; %MSE for fourier = 1.38
% We can see that MSE for wavelet is comparatively less than MSE fourier
% transform
```

*Published with MATLAB® R2019a*