BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Pilani Campus

EEE F434: Digital Signal Processing

**LAB 4: CIRCULAR CONVOLUTION, DISCRETE COSINE TRANSFORM (DCT) AND WINDOW FUNCTIONS**

**Learning Outcomes:**

1. learn the concepts associated with circular convolution: (1) effect of aliasing, (2) relationship between linear and circular convolutions.

2. understand audio compression using the DCT.

3. analyze different window functions in time and frequency domains.

**Note: Please write your MATLAB codes in this .doc file and save it. Capture and paste the snapshots of your plots, wherever required. Make sure you get it signed before leaving the lab.**

**Please make sure you add a title, axis labels, x-axis limit and y-axis limit, grid on, and legend (if required) to each of your figures.**

**PART A: Circular Convolution**

The input sequence x[n] and the impulse response of a system h[n] are given as:

x[n] = {1, 6, 1, 4}

h[n] = {4, 5, 0, 6, 0, 9}

a) Compute the output response using linear convolution. Plot the output sequence.

clear

x\_n=[1 6 1 4];

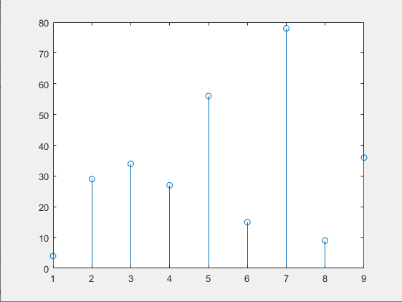
h\_n=[4 5 0 6 0 9];

w\_n= conv(x\_n,h\_n);

%W\_n=(w\_n)';

%N= 1:length(w\_n);

stem(w\_n);



b) Obtain the N-point circular convolution between x[n] and h[n] such that the output sequence from circular convolution is similar to the linear convolution. Use subplots to show your findings.

clear

x\_n=[1 6 1 4];

h\_n=[4 5 0 6 0 9];

w\_n= cconv(x\_n,h\_n);

%W\_n=(w\_n)';

%N= 1:length(w\_n);

W\_n=conv(x\_n,h\_n);

subplot(2,1,1);

stem(W\_n);

grid on

xlabel('n')

ylabel('W\_n')

title('Linear Convolution');

subplot(2,1,2);

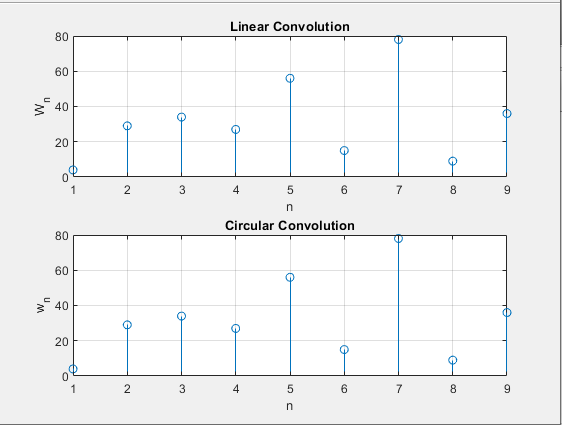
stem(w\_n);

grid on

xlabel('n');

ylabel('w\_n')

title('Circular Convolution');



c) Set the length N of circular convolution as the length maximum between x[n] and h[n]. Compute the N-point circular convolution. Plot the output sequence.

What are the observations with respect to the aliasing effect?

clear

x =[1 6 1 4];

h =[4 5 0 6 0 9];

mx= max(length(x),length(h));

xpad=[x zeros(1,mx-length(x))];

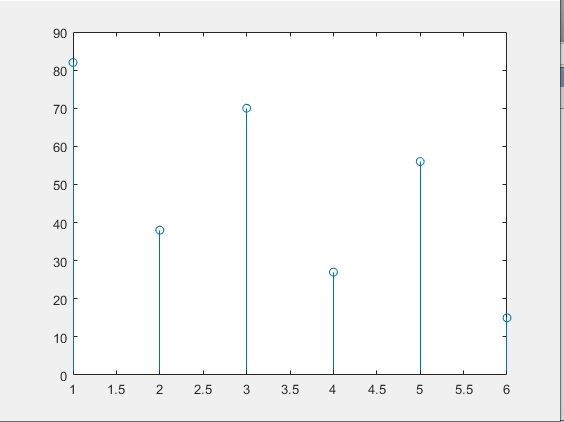
hpad=[h zeros(1,mx-length(h))];

w= ifft(fft(xpad).\*fft(hpad));

%W\_n=(w\_n)';

%N= 1:length(w\_n);

stem(w);



**PART B: Discrete Cosine Transform**

a) Read an audio file

To access the sound file (‘si1188.wav’):

<https://drive.google.com/file/d/0B7-qexRAlXuTUldYOUdPTXFYSW8/view?usp=sharing>

b) Compute and plot the Discrete Cosine transform of the signal.

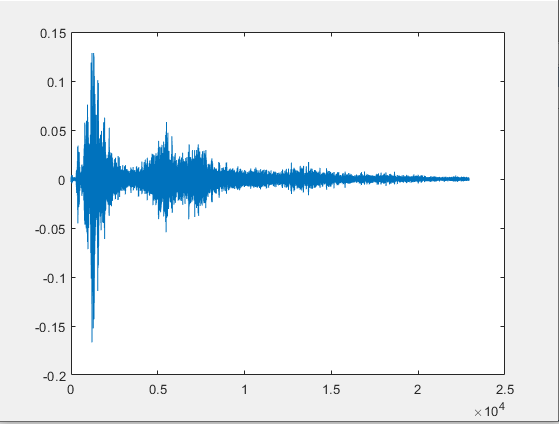
clear

[y,Fs]= audioread('C:\Users\user\Downloads\si1188.wav');

Y=dct(y);

N=0:length(Y)-1;

plot(N,Y);



c) Set the threshold at 10% of the maximum value of the DCT spectrum. Apply thresholding to the DCT spectrum; compute Inverse DCT of the thresholded spectrum. Plot the results.

clear

[y,Fs]= audioread('C:\Users\user\Downloads\si1188.wav');

Y=dct(y);

N=length(Y);

mx= max(abs(Y));

threshold= 0.1\*mx;

for c=1:N

if Y(c)<=threshold;

Y(c)=0;

end

end

plot(Y);

d) Repeat part c) for threshold value of 25% of the maximum value of the DCT spectrum.

clear

[y,Fs]= audioread('C:\Users\user\Downloads\si1188.wav');

Y=dct(y);

N=length(Y);

mx= max(abs(Y));

threshold= 0.25\*mx;

for c=1:N

if Y(c)<=threshold;

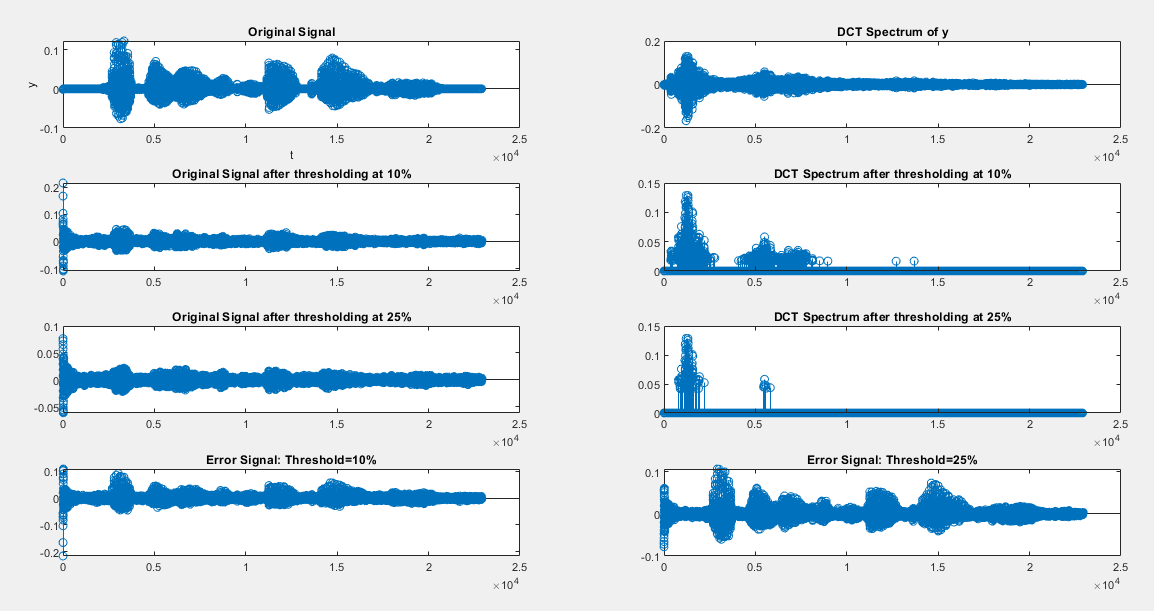
Y(c)=0;

end

end

plot(Y);

Plot the results b) to d) as a subplot with two columns and four rows. Column 1 should have the time domain signal with the corresponding DCT spectrum in column 2.



e) Calculate and plot the absolute errors in part c) and d), which is given by:

**Error = Original signal –Thresholded Signal**

The fourth row of the above subplot (i.e., from parts b,c,d) should plot the absolute error for 10% and 25% threshold value.

f) Calculate the energy loss in part c) and d), which is given by:

**Energy loss =**

Loss1=0.6019

Loss2=0.7597

clear

[y,Fs]= audioread('C:\Users\user\Downloads\si1188.wav');

Y=dct(y);

N=length(Y);

mx= max(Y);

threshold=0.1\*mx;

Y1=Y;

Y2=Y;

for c=1:N

if Y(c)<=threshold;

Y1(c)=0;

end

end

threshold2= 0.25\*mx;

for c=1:N

if Y(c)<=threshold2;

Y2(c)=0;

end

end

th1=idct(Y1);

th2=idct(Y2);

error1= y-th1;

error2= y-th2;

Nmr1=0;

Nmr2=0;

Dnr=0;

for c=1:N

Nmr1=Nmr1+ (y(c)-th1(c))^2;

Nmr2=Nmr2+ (y(c)-th2(c))^2;

Dnr=Dnr+ (y(c))^2;

end

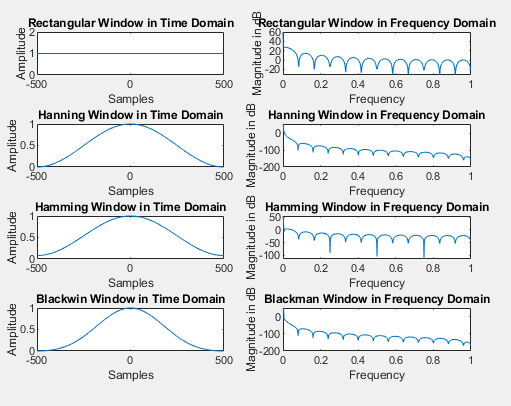
loss1= Nmr1/Dnr;

loss2= Nmr2/Dnr;

What are the observations with respect to the effect of thresholding on the absolute error and energy loss?

**PART C: Window Function**

Generate a time representation and magnitude spectrum (or frequency) of the following window functions: (1) Rectangular, (2) Hamming, (3) Hanning and (4) Blackman widow function, with N =1000. Plot these as a subplot with four rows and two columns. Column 1 should have the time domain signal with the corresponding magnitude spectrum in column 2. Make sure the x-axis for the magnitude spectrum is in Hz and the y-axis is in dB scale. Assume Fs = 10 kHz.



clear

L=1000;

Fs=10000;

nfft=1024;

timeaxis=-500:499;

freqaxis=0:2/nfft:2-2/nfft;

rectwin=rectwin(L);

hannwin=hann(L);

hammwin=hamming(L);

blacwin=blackman(L);

Rectwin=fft(rectwin,nfft);

Hannwin=fft(hannwin,nfft);

Hammwin=fft(hammwin,nfft);

Blacwin=fft(blacwin,nfft);

subplot(4,2,1)

plot(timeaxis,rectwin);

xlabel('Samples')

ylabel('Amplitude')

title('Rectangular Window in Time Domain');

subplot(4,2,2)

plot(freqaxis(1:nfft/2), 20\*log10(abs(Rectwin(1:nfft/2))));

xlabel('Frequency');

ylabel('Magnitude in dB')

title('Rectangular Window in Frequency Domain');

subplot(4,2,3)

plot(timeaxis,hannwin);

xlabel('Samples')

ylabel('Amplitude')

title('Hanning Window in Time Domain');

subplot(4,2,4)

plot(freqaxis(1:nfft/2), 20\*log10(abs(Hannwin(1:nfft/2))));

xlabel('Frequency');

ylabel('Magnitude in dB')

title('Hanning Window in Frequency Domain');

subplot(4,2,5)

plot(timeaxis,hammwin);

xlabel('Samples')

ylabel('Amplitude')

title('Hamming Window in Time Domain');

subplot(4,2,6)

plot(freqaxis(1:nfft/2), 20\*log10(abs(Hammwin(1:nfft/2))));

xlabel('Frequency');

ylabel('Magnitude in dB')

title('Hamming Window in Frequency Domain');

subplot(4,2,7)

plot(timeaxis,blacwin);

xlabel('Samples')

ylabel('Amplitude')

title('Blackwin Window in Time Domain');

subplot(4,2,8)

plot(freqaxis(1:nfft/2), 20\*log10(abs(Blacwin(1:nfft/2))));

xlabel('Frequency');

ylabel('Magnitude in dB')

title('Blackman Window in Frequency Domain');