EEL 4930/5934 BioSignals Processing

Luiz Medeiros

Assignment 3- Due: November 15

Please write a report answering the following questions. Submit the report and all the m-files you used to answer the questions on Blackboard in Module Assignments / Assignment 3.

1. Exercises with known functions: Generate a discrete-time signal  sampled from  at the rate of  and duration of  seconds. We will refer to the signal as the MATLAB variable ***x***.
   1. Calculate ***fx*** the 1024-point DFT of ***x***. Using the ***subplot*** option, ***plot*** ***x*** versus *time* in seconds and magnitude of ***fx*** versus true frequency on Hz.
   2. Using ***wavedec*** with the wavelet choice ***db4*** (Daubechies 4) find the three level wavelet transform coefficients. You will have four sets of coefficients (detail 1, detail 2, detail 3, and approximation coefficients) sampled at different rates.
      1. Draw a block diagram that shows the filtering and downsampling operations on the data ***x*** that result in these coefficients. Indicate on the bloc diagram where these coefficients are and their sampling rates.
      2. Compute the DFT of each coefficient set and plot their magnitude versus true frequency in Hz.
      3. Compare the spectra above with the spectral magnitude of ***x*** and comment on how they are related.
   3. Using the ***waverec*** instruction, and the output of the ***wavedec*** used in 1.b, show that the reconstruction process works to restore ***x*** by comparing the output of ***waverec*** and ***x***.
2. Use the ***wfilters*** instruction to find the coefficients of the four filters ( high and low decomposition and high and low reconstruction) and their 256-point DFT’s. Show that they are perfect reconstruction filters by adding the squared magnitude of the low decomposition filter and the high decomposition filter. Show the same for the reconstruction filters.
3. Please study the following programs contained in this folder and answer the questions below:
   1. ***wavdecsines.m***
      1. In two or three sentences describe what this program does.   
         The program creates a sample x, based on a series of cosines. Then, it calculates the FFT and produces a wavelet decomposition of that signal. The program then provides a plot and follows to utilize the wavelet decomposition algorithm. Finally it provides an insight of the decomposition and re-composition process with a final graph providing and error assessment.
      2. What is stored in the variable **C**?  
         C is the vector containing the respective decomposed signal. It contain each portion of the signal, whose lengths are indicated by the other returned variable, L.
      3. How does this program extract the wavelet coefficients?  
         The program utilizes the Wavedec algorithm to extract the coefficients.
      4. What does N=3 indicate?  
         N=3 signifies the depth of the decomposition that we are going to. Thus, N=3 -> level #
      5. This program returns Approximation coefficients in cell array wc(1), detail coefficients at level 3 in wc(2), detail coefficients at level 2 in wc(3) and detail coefficients at level 1 in wc(4).
         1. What are the lengths of each set of coefficients?  
            Lengths are as follows: 109, 109, 208, 406;
         2. Account for the length of detail coefficients at level 1.  
            Length of Detail coefficients at level 1 is equivalent to the highest portion of the filter and contains the components with the highest frequencies. As we can see through the graph, the components shown are at f>40;
         3. What is the sampling rate of the detail coefficients at level 2? Explain.  
            We began with an initial sampling rate of 200. Following from that sampling rate we decreased by 2, to level 1, where we have 100. Then to level 2, where we would have sr=50;
      6. Consider the variables d4, d3, d2 and d1.
         1. What are stored them?  
            These variables contain the reconstructed components of the wavelets.
         2. What is the sampling rate of the samples in them? Explain.  
            sr=200, 100, 50, 25.
   2. ***wavdecNrec.m:*** What does this program show?   
      This program decimates the signal, and then puts the signal together.
   3. ***Makewlt.m:*** What does this program deliver? What are stored in **wlt** and **scfn** in lines 45 and 46?  
      The program delivers the respective wavelets utilized to decimate each function.   
      Stored in the wlt and scfn variables is the information regarding each respective fundamental wavelet.
   4. ***wavdecToyWavelets1.m***:
      1. Insert comments in the code repeated below so that the program is clear to a person with programming and signal processing expertise but no familiarity to this program.
      2. Add code to add a title and axes labels for each graph.

clear; clc; close all;

%load scfn and wlt for db8.

load scfnL3db3 %scfnL3db3 % here we are loading the stored variables;

load wltL3db3 %wltL3db3

M=length(wlt); % here we are extracting the length of the signal or number of samples

%let the wavelet and scfn be of 50 ms duration

sr=M/50\*10^3; % Fs= sampling frequency= 1/T, where T=sampling period=50ms. sr=Fs;

%generate a toy signal from linear combinations of the scn

n=20;

cf=randn(1,n); % Here we are creating a random number array of 20.

t=[0:1/sr:4]'; % by utilizing the given sampling rate sr, we split a total time

% of 4 seconds into our sampling period, 1/sr;

lx=length(t);

tr=[1:n]\*M;

x=zeros(lx,1); % create an x array of zeros as big as our respective sample #

for k=1:n

x(tr(k):tr(k)+M-1)=x(tr(k):tr(k)+M-1)+cf(k)\*scfn; % Creating a signal based on randomizing the

% wavelets functions achieved from the previous exercise.

end

figure

plot(t,x),title('Random Wavelet based signal'),xlabel('t,time'),ylabel('x');

dbname='db8'; % Daubechis 8

nfft=1024; % Defining an nfft to perform the fft .

fx=fft(x,nfft);% performing the fft

afx=abs(fx(1:nfft/2+1)); % achieving absolute value of the fft ;

f=(0:nfft/2)/nfft\*sr; % defining the discrete frequency domain based on the nfft point DFT.

fgn=1 % figure iteration number.

figure(fgn)

subplot(211),plot(t,x), title ('random sig x vs t '),

subplot(212),plot(f,afx),title('abs value of the fft of x'),xlabel('f'),ylabel('abs(X)');

N=3; % filter level to decimate waves.

[C,L]=wavedec(x,N,dbname); % Decimation of waves. returning values and coefficients to variables C and L

begin=1;

fgn=fgn+1; % incrementing the figure counter

figure(fgn)

for k=1:N+1

fin=begin+L(k)-1;

wc(k)={C(begin:fin)}; % Allocating the respective decimated signal to separate cells

d=cell2mat(wc(k)); %placing the cells in one variable.

fd=fft(d,nfft); % taking the fft of the cell with the decimated signals.

afd=abs(fd(1:nfft/2+1)); % Now the absolute value of the function.

begin=fin+1;

if k<2

p=N % in the case we are going through our first iteration , we set p to level 3

else

p=p-1;

end

t=(0:L(k)-1)/(sr/2^p); % creating the appropriate time domain.

f=(0:nfft/2)/nfft\*sr/2^p; %frequency representation.

figure(fgn)

subplot(N+1,2,2\*k-1),plot(t,d) % ploting the raw value of the decimated portion of the function

subplot(N+1,2,2\*k),plot(f,afd) % plotting the abs value of the same.

end

wx=ndwt(x,N,dbname,'mode','per'); % a struct with all of the pertaining information

% of a non dcimated transform.This includes:

% level , N, signal , x, decimation algorithm name, dbname, mode, and

t=0:1/sr:4;

f=(0:nfft/2)/nfft\*sr;

%a=indwt(wx,'a',0);

d4=indwt(wx,'d',4); % inverse the decimation by level

d3=indwt(wx,'d',3);

d2=indwt(wx,'d',2);

d1=indwt(wx,'d',1);

unc=[d1 d2 d3 d4];

xx=sum(unc,2); % summing the inverse of the decimated signal.

recer=x-xx; % checking the difference between original signal and attained inversely decimate signal.

figure

plot(xx), title('inversely decimated x')

hold

pause

plot(x,'r'), title('original x signal'),

plot(recer,'g','LineWidth',2),title('difference between the two different xs');

func=fft(unc,nfft); % going through the fft algorithm of the inverse operation.

afunc=abs(func(1:nfft/2+1,:));

figure

for k=1:N+1

subplot(N+1,2,2\*k-1), plot(t,unc(:,k))

subplot(N+1,2,2\*k),plot(f,afunc(:,k))

end

%comparisons

d=cell2mat(wc(1));

figure

subplot(211),stem(d(1:600))

subplot(212), stem(tr/8+37,cf) % stem plot of the signal.

figure

plot(t,x),title( 'original signal'),

hold

pause,figure, % I added the figure command because it is easier to just add it

% as another figure to view next to the original signal.

plot(t,unc(:,N+1),'r') % plot of the signal after gone through decimation and inverse decimation.

1. Make use of the given program ***wavdecToyWavelets1.m*** to analyze the EEG signal given in EEG4wlt.mat using your assigned wavelets and N=4,5,6. da Costa: db1,db12, sym, Evans: db1, db11, Ibrahim: db1, db10, coif4, Karp: db1,db9, Lieber: db1,db7,dmey, Martinez:db1,db7, Madeiros: db1,db6, Miranda: db1,db5, Muhamed Ali: db1, db8, coif5, Nair: db1,db14, Quevillon:db1, db3, db8, Ramdhan:db1, db4, db10, Reddy:db1,db2,db11, Wilson:db1, db2, db10, Zhang: db1,db10, coif3

As a result of added levels of to the WT algorithms, we are able to see more details from the signal.

This can be very useful for the human and computer to trigger and learn from specific patterns of behavior that occur at given times and/or frequency points.

%% 4.

% Medeiros: db1,db6;

clear; clc; close all;

load EEG4wlt;

sr=256; % Fs= sampling frequency= 1/T, where T=sampling period=50ms. sr=Fs;

% This is more than sufficient to sample the given signal.

t=[1/sr:1/sr:(length(x)/sr)]'; % by utilizing the given sampling rate sr, we split a total time

% time=sapmles/sampling rate=3600 second=60 minutes= 1hr.

lx=length(t);

figure

plot(t,x),title('EEG Signal'),xlabel('t,time'),ylabel('x');

dbname='db1'; % Daubechis 8

nfft=1024; % Defining an nfft to perform the fft .

fx=fft(x,nfft);% performing the fft

afx=abs(fx(1:nfft/2+1)); % achieving absolute value of the fft ;

f=(0:nfft/2)/nfft\*sr; % defining the discrete frequency domain based on the nfft point DFT.

fgn=1 % figure iteration number.

figure(fgn)

subplot(211),plot(t,x), title ('EEG sig x vs t '),

subplot(212),plot(f,afx),title('abs value of the fft of x'),xlabel('f'),ylabel('abs(X)');

N=4; % filter level to decimate waves.

for a=1:3

[C,L]=wavedec(x,N,dbname); % Decimation of waves. returning values and coefficients to variables C and L

begin=1;

fgn=fgn+1; % incrementing the figure counter

figure(fgn)

str='';

for k=1:N+1

fin=begin+L(k)-1;

wc(a\*k)={C(begin:fin)}; % Allocating the respective decimated signal to separate cells

d=cell2mat(wc(a\*k)); %placing the cells in one variable.

fd=fft(d,nfft); % taking the fft of the cell with the decimated signals.

afd=abs(fd(1:nfft/2+1)); % Now the absolute value of the function.

begin=fin+1;

if k<2

p=N % in the case we are going through our first iteration , we set p to level 3

else

p=p-1;

end

t=(0:L(k)-1)/(sr/2^p); % creating the appropriate time domain.

f=(0:nfft/2)/nfft\*sr/2^p; %frequency representation.

figure(fgn)

str=num2str(k);

subplot(N+1,2,2\*k-1),plot(t,d),title(['decimated signal x ',str,'N is ',num2str(N)]) % ploting the raw value of the decimated portion of the function

subplot(N+1,2,2\*k),plot(f,afd),title(['decimated abs of the fft of x ',str,'N is ',num2str(N)]) % plotting the abs value of the same.

end

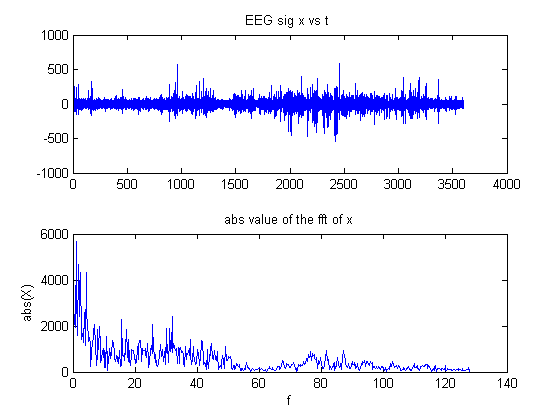
fgn=fgn+1;

N=N+1;

end

The difference between the code for db1 and db6 is only the declared string.

Thus, below are the pictures for db1:

And below follows for db6:

