Luiz Medeiros

**Final Project and Take Home Exam**

Digital Signal Processing, Spring 2013

**Part I**

**1.**

%%%%%%%%%%%%%%%

clear, close all, clc;

%% A + B

[x,sr,b]=wavread('TONE');

lx=length(x); % Length of x[n];

t=[0:lx-1]/sr; % Time in Seconds;

figure,

subplot(2,1,1),

plot(x),xlabel('samples');

subplot(2,1,2),

plot(t,x),xlabel('Time in seconds');

%% C + D;

%nfft=2^(fix(log(lx)/log(2))+4); % makes it log 2;

nfft=2^(fix(log2(lx))+4); % Utilizes log2(x);

disp(nfft)

X=fft(x,nfft);

lX=length(X);

magX=abs(X(1:lX/2));

phaseX=angle(X(1:lX/2));

f=[0:lX/2-1]\*sr/lX; % f in Hz. fs/2 -> pi;

om=2\*pi\*f/sr; % omega; 2\* pi \* f;

figure,

grid minor,

subplot(2,1,1),

plot(om,20\*log10(magX)),

title('dB Plot of magX'),

subplot(2,1,2),

plot(om,magX),

title('magX vs Omega; 0-> pi');

%% E

figure,

grid minor,

subplot(2,1,1),

plot(f,20\*log10(magX)),

title('dB Plot of magX'),

subplot(2,1,2),

plot(f,magX),

title('magX vs f; 0-> pi');

%% F

% Using the Spectral tip, find the four highest peaks;

% f0=1330;f1=3989.94;f2=6649.93;f3=9309.88;

%% G

w1=1330/sr\*2\*pi;

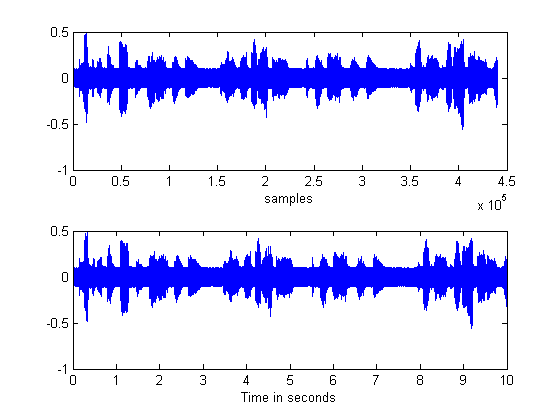
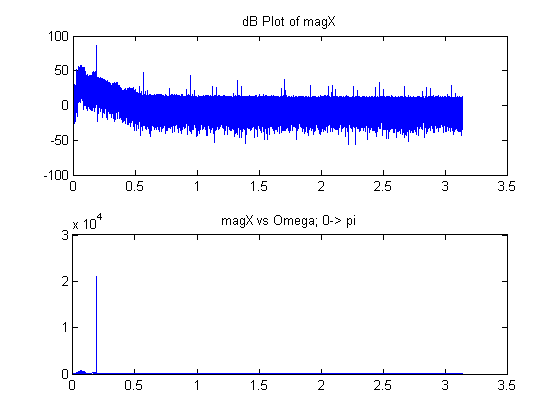
w2=3989.94/sr\*2\*pi;

w3=6649.93/sr\*2\*pi;

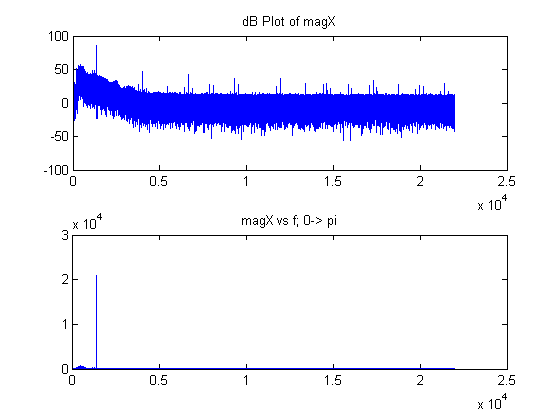
w4=9309.88/sr\*2\*pi;

%w5=11969.84/sr\*2\*pi;

w=[w1 w2 w3 w4];

****

Above, plots of x(t)

****

%% H

clear w0 H r a b k;

H=[zeros(5,nfft/2)];

r=0.99;

w0=[zeros(1,4)];

a=[zeros(4,3)];

b=[zeros(4,3)];

for k=1:4

w0=w(1,k);

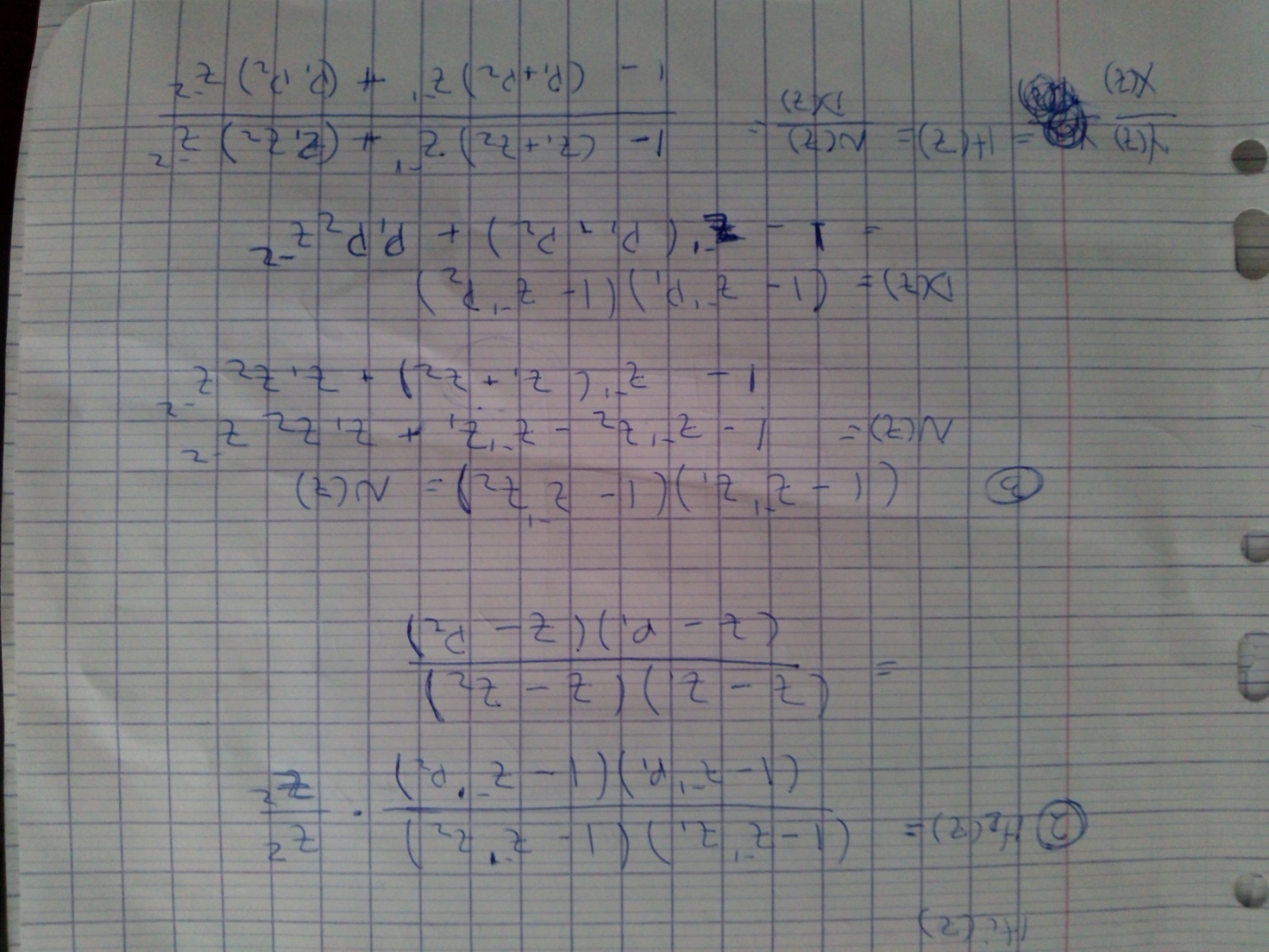
b(k,:)=[1 -2\*cos(w0) 1];

a(k,:)=[1 -2\*r\*cos(w0) r^2];

H(k,:)=freqz(b(k,:),a(k,:),nfft/2); % Part 4

end

%

For Numbers 2 and three of exercise H, please refer to the figure below:

%% H part 5

figure,

hold on,

grid minor,

for k=1:4

plot(om,20\*log10(abs(H(k,:))),'r');

end

plot(om,20\*log10(magX)),

title('dB of the filters and magX'),

xlabel('Filters are in Red');



%% I

clear y k;

y=x;

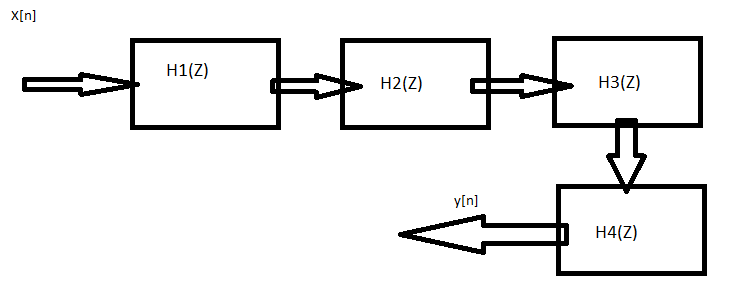
for k=1:4

y=filter(b(k,:),a(k,:),y);

%k;

end

Now, **I Part 1:**



%% I part 2

clear hImpf hImpt;

hImpf=zeros(4,length(f));

hImpt=zeros(4,length(t));

for k=1:4

hImpf(k,:)=filter(b(k,:),a(k,:),[1,zeros(1,length(f)-1)]);

hImpt(k,:)=filter(b(k,:),a(k,:),[1,zeros(1,length(t)-1)]);

end

figure,

hold on, grid minor,

title('hImpf vs t'),

for k=1:4

subplot(4,1,k),

plot(t,hImpt(k,:)),

title('hImpt vs t'),

axis([-0.01 0.1 -0.02 0.02]);

end

HImpf=fft(hImpf);

for k=1:4

magHImpf(k,:)=abs(HImpf(k,1:length(HImpf/2)));

end

figure,

hold on, grid minor,

title('magHImpf vs f'),

for k=1:4

subplot(4,1,k),

plot(f,magHImpf(k,:)),

title('magHImpf vs f'),

axis([-0.01 5 -0.5 0.5]);

end



%% J part 1;

% The combining operation we must use to obtain H(z),

% with all the frequencies we wish to filter is the convolution

% of all coefficients of b and the convolution of all coefficients

% of a. This will result in the b/a form that after the

% performing the z transform gives us our H(z).

ha=a(1,:);

hb=b(1,:);

for k=2:4

ha=conv(ha,a(k,:));

hb=conv(hb,b(k,:));

end

HH=freqz(hb,ha,nfft/2);

%% J part 2;

% The lengths N and M are respectively 9.

%% J part 3;

% The coefficients are as follows:

bn=hb

an=ha

%% J part 4;

s=filter(hb,ha,x);

S=fft(s,nfft);

magS=abs(S(1:nfft/2));

%% J part 4 a;

Y=fft(y,nfft);

magY=abs(Y(1:nfft/2));

figure,

subplot(2,1,1),

plot(f,magY),

title('magY vs f');

subplot(2,1,2),

plot(f,magS),

title('magS vs f');

figure,

subplot(2,1,1),

plot(f,20\*log10(magY)),

title('dB of magY vs f');

subplot(2,1,2),

plot(f,20\*log10(magS)),

title('dB of magS vs f');

figure,

subplot(2,1,1),

plot(t,s),title('s vs t');

subplot(2,1,2),plot(t,y),

title('y vs t');

% Graphically, I am not able to see any difference.

% I zoomed and analyzed the both images and graphs, however,

% no apparent difference was found.

%% J part 4 b;

soundsc(y,sr);

soundsc(s,sr);

% Within both hearings I hear a significant decrease in noise.

% The loud beep that followed the voice in the beginning is no longer

% present. However there is still some noise present, but I am

% not able to distinguish the noise or sound difference between

% either output.

%% J part 4 c;

d=s-y;

%% J part 4 c i;

figure,

subplot(2,1,1),

plot(t,s),title('s vs t');

subplot(2,1,2),plot(t,y),

title('y vs t');

figure,plot(t,d),title('d vs t');

% From the graph and vector entries, it is possible to see magnitudes

% which range to from numbers in the order of 10^-13.

% A very small order of error.

% From what was mentioned in class, this reapproves the point

% where the latter S(z) contains less error than Y(z).

%% J part 4 c ii;

D=fft(d,nfft);

magD=abs(D(1:nfft/2));

energy= (magD./magY).^2;

plot(om,energy),

title('energy vs omega');

[q w]=max(energy);

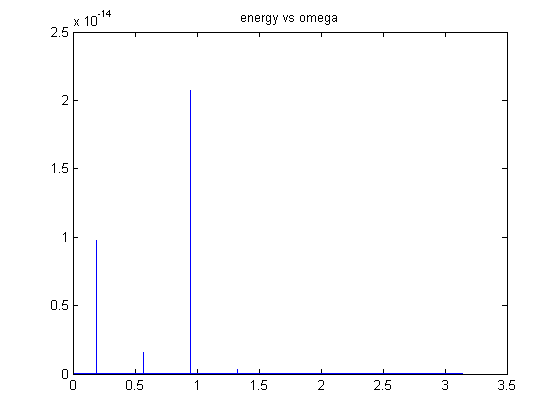
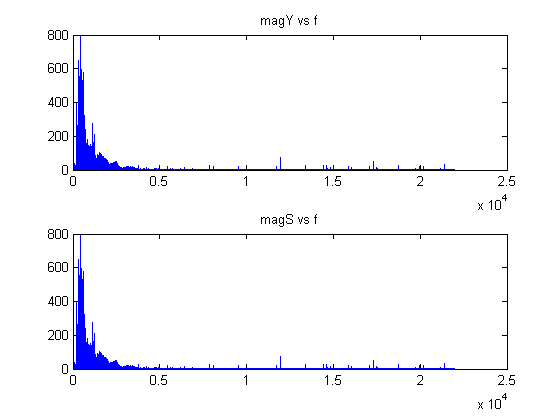
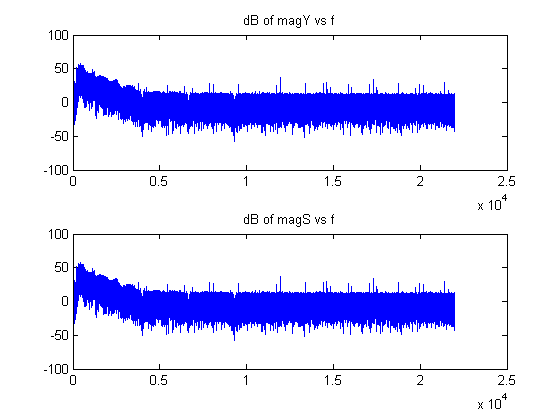
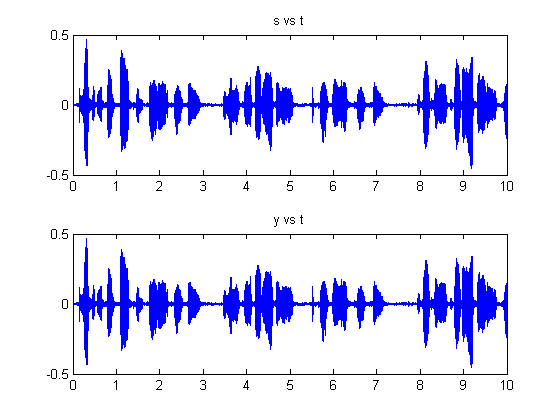
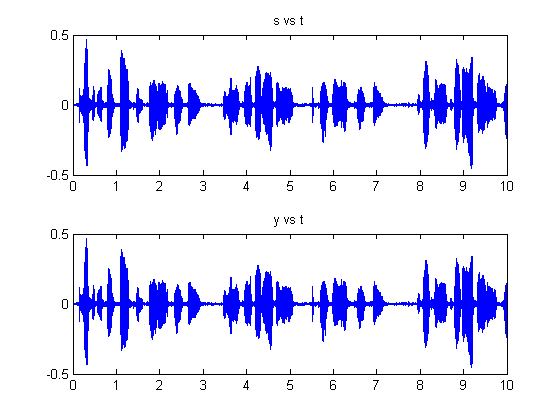
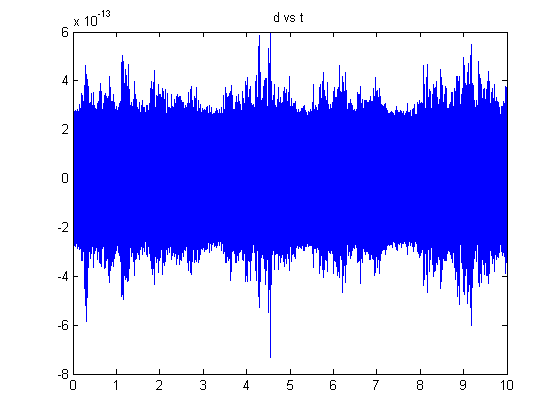
w\*2\*pi/lX;

% From analyzing the graph given, it is possible to see an

% energy spike in each frequency that we ommitted from the

% original signal. Very small (10^(-14)), but still something

% to consider.



For exercise **K** I will include the code, which may be reused to reach all the graphs and solutions to parts **a-f;**

%% K

% Luiz MEdeiros

%% K

clear all, close all, clc;

f1=1330;

f2=3989.94;

f3=6649.93;

f4=9309.88;

f0=[ f1 f2 f3 f4];

sr=44000;

t0=1/f0(1);

fs=sr;

td=10\*t0;

t=0:1/fs:td;

for k=1:4

x(k,:)=cos(2\*pi\*f0(k)\*t);

end

lx=length(x);

nfft=2^(fix(log2(lx))+2); % Utilizes log2(x);

xTest= x(1,:)-1/3\*x(2,:)+1/5\*x(3,:)-1/7\*x(4,:);

%% Plotting;

figure,

plot(t,xTest(1,:)),

title('xTest vs t');

%%

%% Building the filters

w0=f0/fs\*2\*pi;

H=[];

r=1.1;

a=[zeros(4,3)];

b=[zeros(4,3)];

% Part 4

for k=1:4

b(k,:)=[1 -2\*cos(w0(k)) 1];

a(k,:)=[1 -2\*r\*cos(w0(k)) r^2];

%H(k,:)=freqz(b(k,:),a(k,:),nfft,'whole');

yn(k,:)=filter(b(k,:),a(k,:),x(k,:));

end

%% K 1 a i

%

figure,

grid minor,

hold on,

subplot(411),title('x1 y1(red)'),hold,

subplot(412),title('x2 y2(red)'),hold,

subplot(413),title('x3 y3(red)'),hold,

subplot(414),title('x4 y4(red)'),hold;

for k=1:4

X(k,:)=fft(x(k,:),nfft);

magX(k,:)=abs(X(k,1:nfft/2));

end

for k=1:4

subplot(4,1,k),plot(t,x(k,:)),

subplot(4,1,k),plot(t,yn(k,:),'r');

end

for k=1:4

Yn(k,:)=fft(yn(k,:),nfft);

magYn(k,:)=abs(Yn(k,1:nfft/2));

end

lX=length(X);

f=[0:lX/2-1]\*fs/lX; % f in Hz. fs/2 -> pi;

figure,

grid minor,

hold on,

subplot(411),title('magx1 vs magy1(red)'),hold,

subplot(412),title('magx2 vs magy2(red)'),hold,

subplot(413),title('magx3 vs magy3(red)'),hold,

subplot(414),title('magx4 vs magy4(red)'),hold;

for k=1:4

subplot(4,1,k),plot(f,20\*log10(magX(k,:))),

subplot(4,1,k),plot(f,200\*log10(magYn(k,:)),'r');

end

%% Final Analysis

ha=[1 -2\*r\*cos(w0(1)) r^2];

hb=[1 -2\*cos(w0(1)) 1];

for k=2:4

ha=conv(ha,a(k,:));

hb=conv(hb,b(k,:));

end

HH=freqz(hb,ha,nfft/2);

%%

clear y k;

XTest=fft(xTest,nfft);

magXTest=abs(XTest(1:nfft/2));

lX=length(XTest);

f=[0:lX/2-1]\*fs/lX; % f in Hz. fs/2 -> pi;

y=xTest;

i=1;

col='kbgr';

figure,

grid minor,

hold on,

subplot(311),title('magXTest vs f'),hold,plot(f,magXTest,col(i)),

subplot(312),title('output(y) vs time'),hold,

subplot(313),title('magY vs f'),hold,

for k=1:4

y=filter(b(k,:),a(k,:),y);

Y=fft(y,nfft);

magY=abs(Y(1:nfft/2));

subplot(3,1,3),plot(f,magY,col(i)),

subplot(3,1,2),plot(t,y,col(i));

pause;

i=i+1;

%k;

end

%{

After analyzing the graphs given achieved through the

analysis, the filters seem to reduce the spectrum of all frequencies

significantly, except the first frequency at w0=.19;

The spectra do not disappear completely, which I suspect is due

to the time response of the filter.

When I tested with r=0.5 the response of the filter was incredibly

better! Beyond this, it completely cleaned the signal.

This demonstrates the consequences of remaining in the border

of convergence.

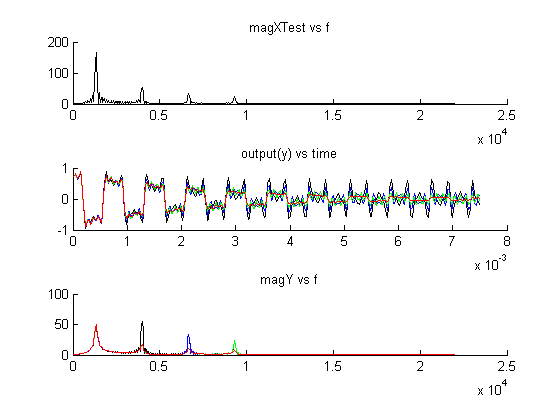
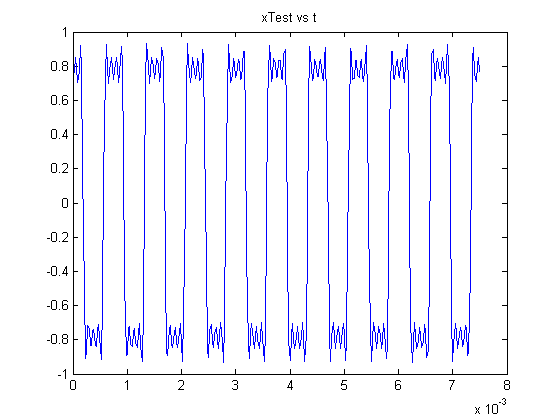
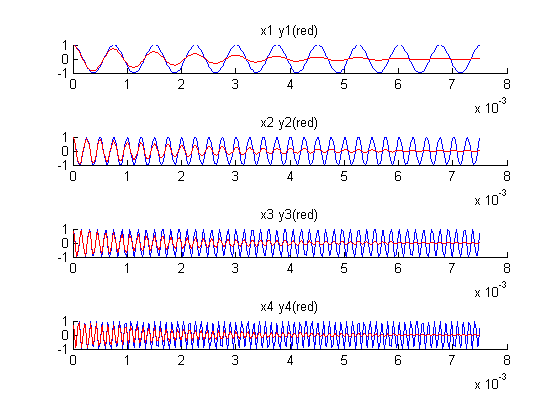
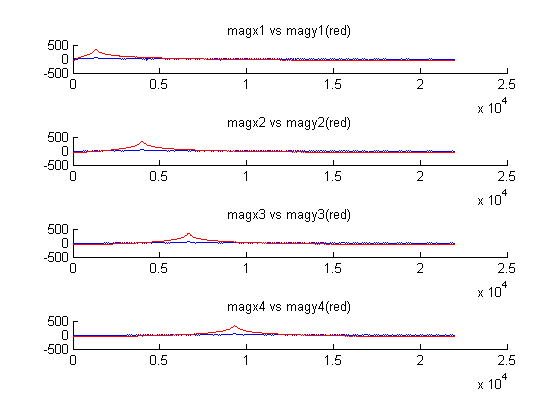
For r=1.1 the output goes insane in both domain.

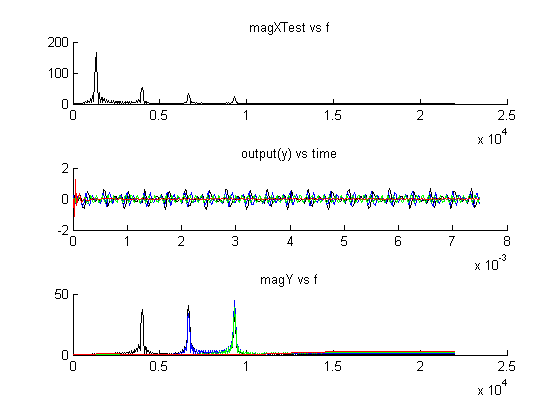
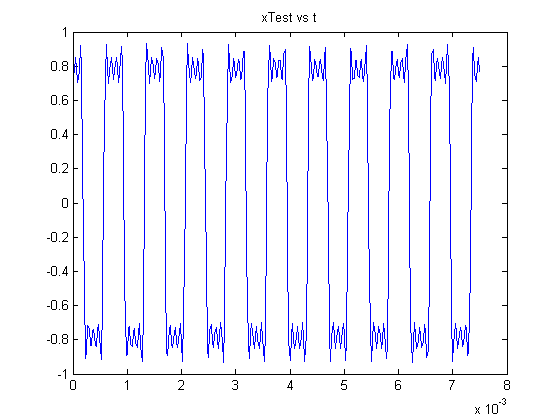
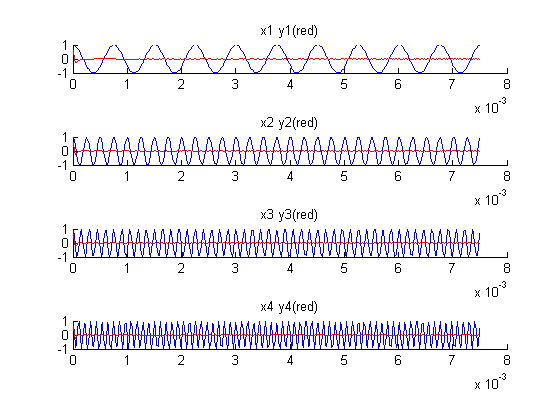
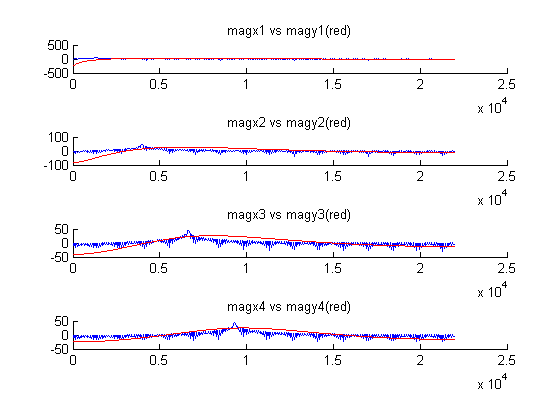
it blows up after 7 milliseconds in time domain and has

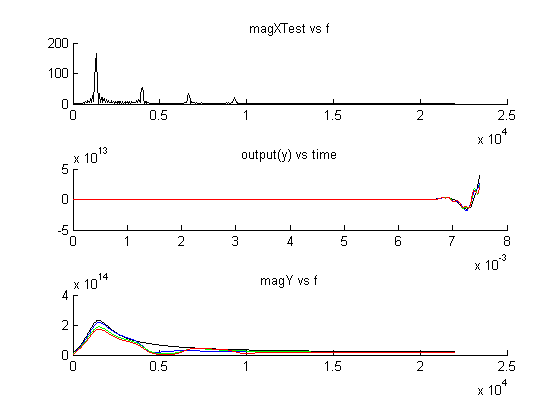
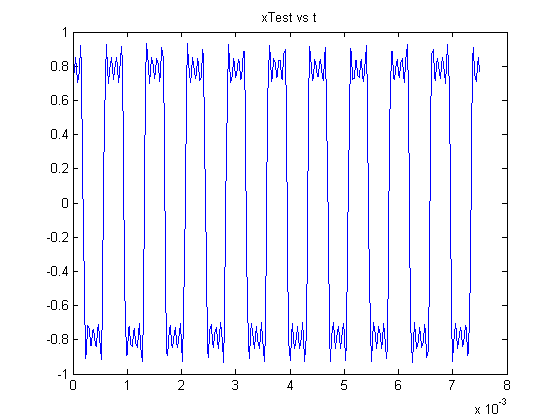
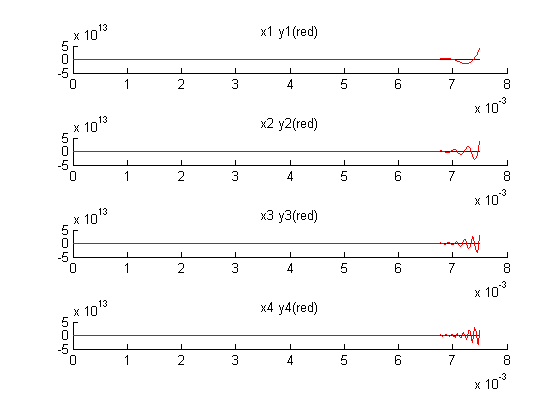
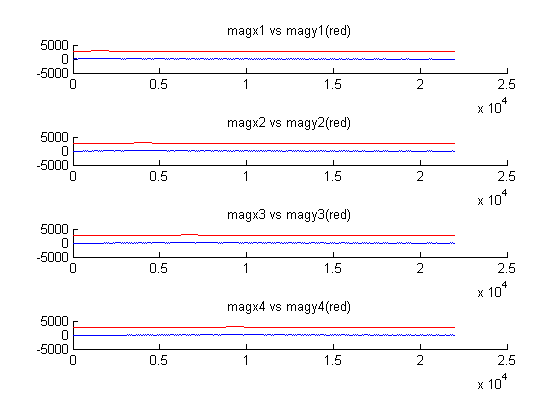
what almost looks like a bandpass outlook in the f domain.

%}

Now, We will start with the images from r=0.99, then r=0.5 and finally, r=1.1;







As we may note from these graphs, once |Pi|, or the magnitude of the poles, reaches values beyond the region of convergence, the system becomes unstable.

Through these graphs, it's also possible to note the response time of these filters, which are also influenced by the values of r.

**PART II**

Below is the code and graphs for exercise **A** of part II. A small analysis referring to the exercise sits within the code.

%% Part II of Final Project and Exam.

% Luiz Medeiros

%% A, 1 2

clc, clear, close all;

[y,sr,b,]=wavread('cleanTONE');

nsr=sr/8; % This is generating the desired sampling rate.

ay(1)=y(1); % First sample

for k=1:fix(length(y)/8) % We go up to length(y)/8 sample,

% Which is the last sample to be taken

% Considering our sampling strategy.

ay(1,k+1)=y(8\*k,1); % THe 8\*k is taking each Kth sample;

% In turn, we already has stored the first sample

%

end

at=[0:length(ay)-1]/nsr; % Generating our time with respect to

% Each sampling rate;

t=[0:length(y)-1]/sr;

%% A, 3;

figure,

subplot(2,1,1),

plot(at,ay),

title('ay, at');

%figure,

subplot(2,1,2),

plot(t,y),

title('y vs t');

%% A, 4,5;

ly=length(y);

lay=length(ay);

nffty= 2^(fix(log2(ly))+4);

nfftay=2^(fix(log2(lay))+4);

nfft=2^(fix(log2(ly))+4);

Y=fft(y,nffty);

% AY=fft(ay,nfftay);

AY=fft(ay,nfft);

magY=abs(Y(1:nffty/2));

% magAY=abs(A(Y1:nfftay/2));

magAY=abs(AY(1:nfft/2));

lY=length(Y);

lAY=length(AY);

f=[0:lY/2-1]\*sr/lY;

fAY=[0:lAY/2-1]\*nsr/lAY;

figure,

subplot(2,2,1),

% plot(fAY,magAY),

plot(fAY,magAY),

title('MagAY vs f');

%figure,

subplot(2,2,2),

plot(f,magY),

title('magY vs f');

subplot(2,2,3),

plot(fAY,20\*log10(magAY)),

title('dB(MagAY) vs f');

subplot(2,2,4),

plot(f,20\*log10(magY)),

title('dB(magY) vs f');

%% A 6, 7

% soundsc(y,sr);

% soundsc(ay,nsr);

%{

The output was quite distinct. There was a certain lower tone to the

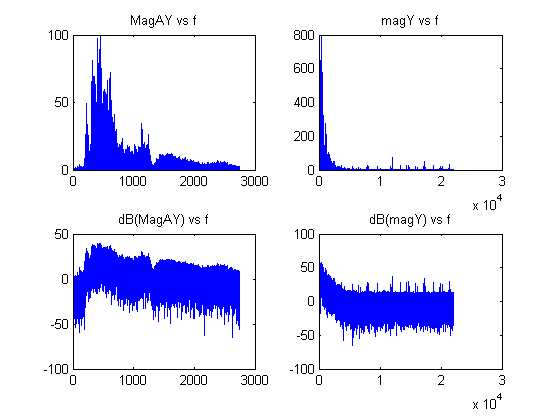
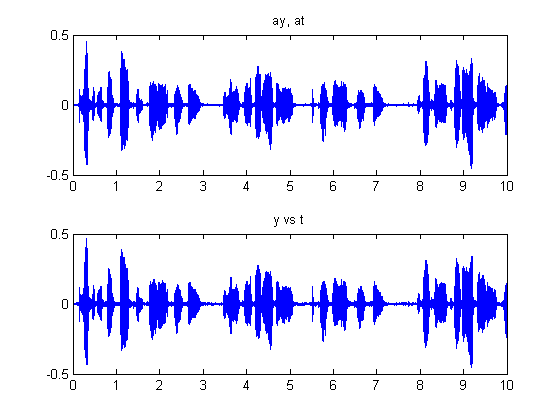
altered sampled output in comparison to the normal output.

In support of this observation, it is also possible to note the

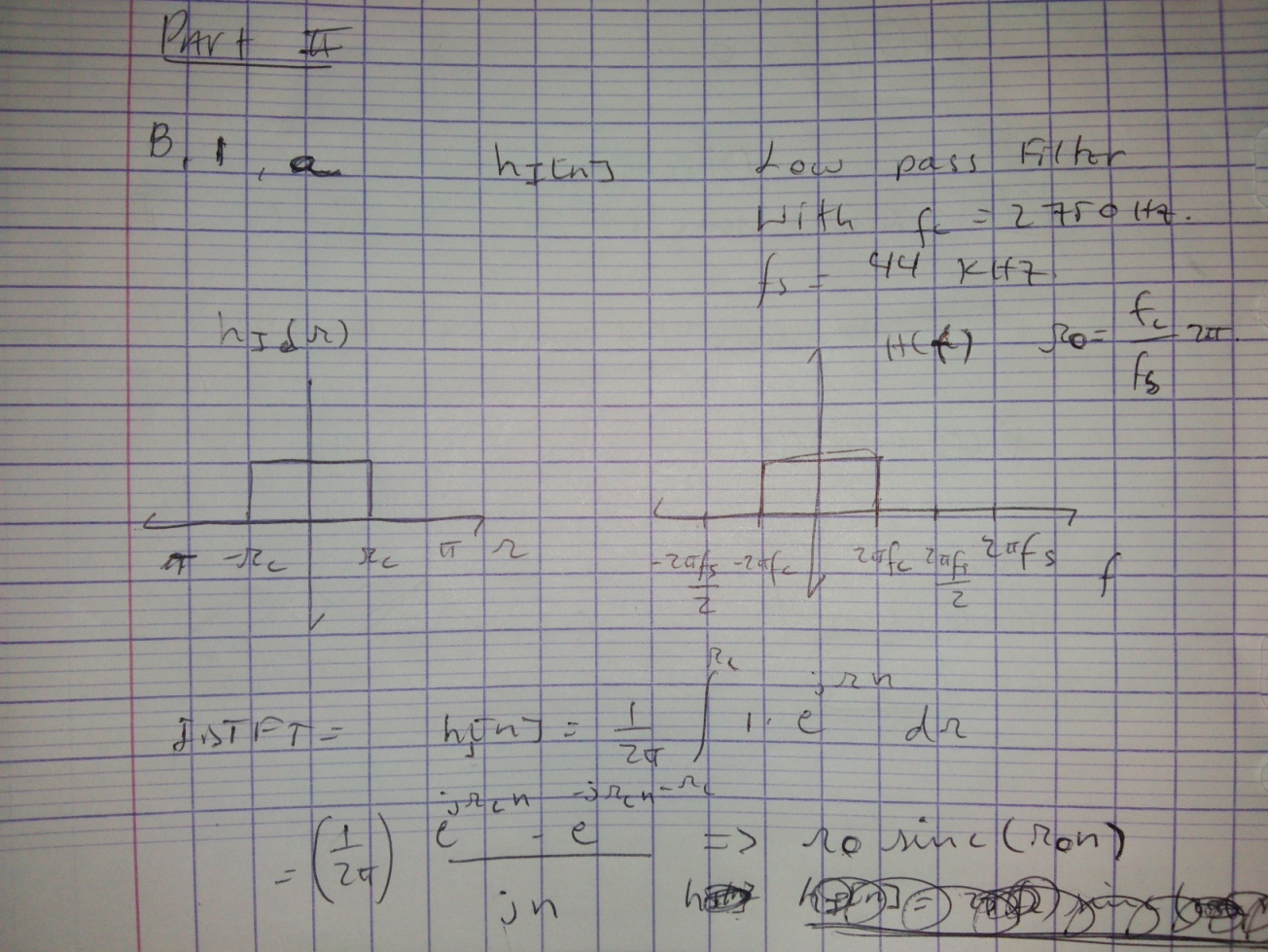
amplitude difference between MagAY and MagY; The difference

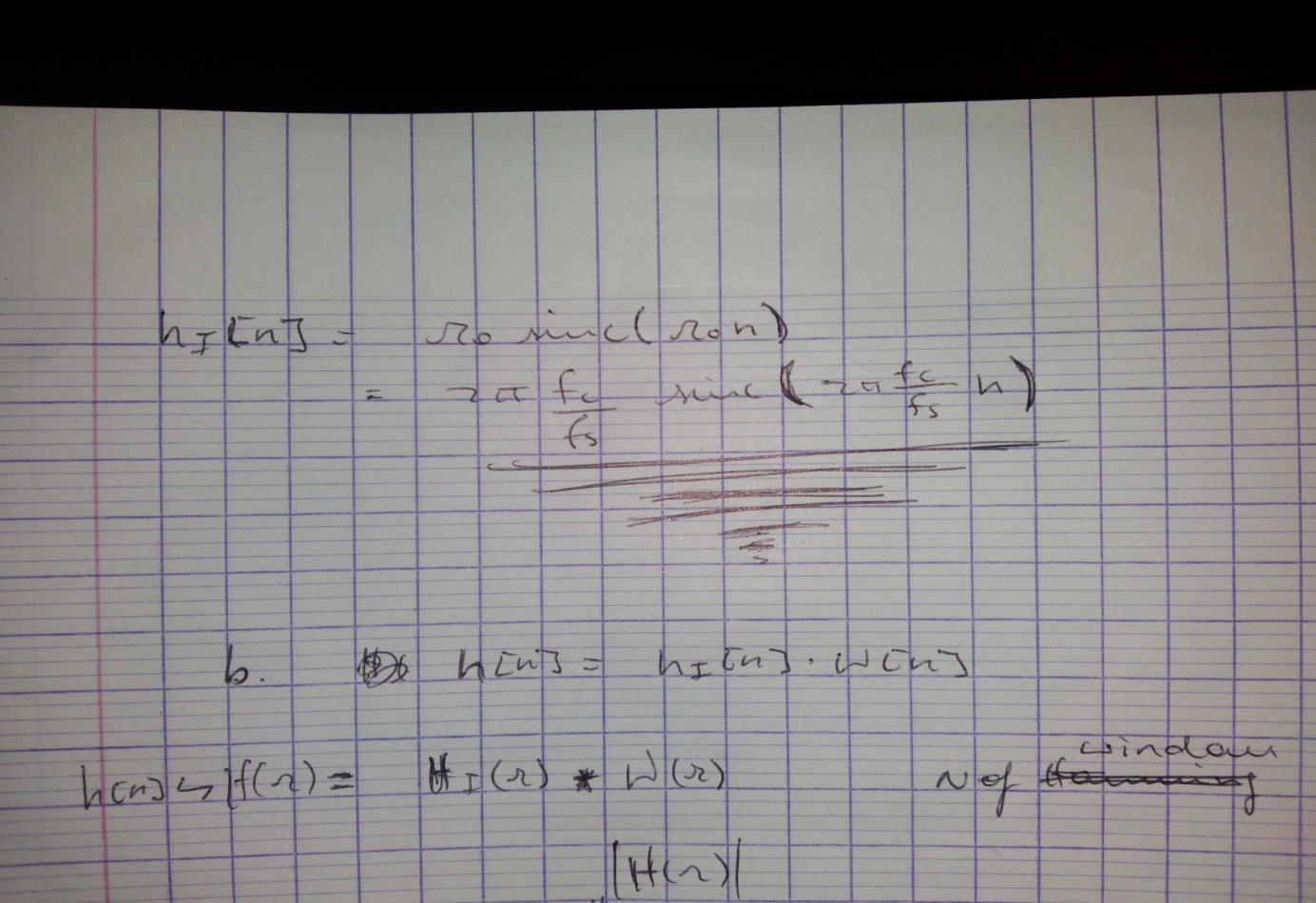
is exactly by the factor of 8.

%}



Now, On to assess Exercise **B** of this part.





% B 1 b;

fs=2750; % Desired Frequency in Hz;

sr=44000; % Sampling rate Hz;

Fc=fs/sr;

Wc=2\*pi\*Fc;

%nfft=2^12;

np=1:ly-1;

n=-ly+1:ly-1;

hp=sin(Wc\*np)./(pi\*np);

hpm=fliplr(hp);

h=[hpm 2\*Fc hp];

%w=hamming(50);

w=hamming(2\*length(np)+1)';

N=length(np);

hw=h.\*w;

%%

W=hamming(50+1);

[b,a]=fir1(50,Wc,W);

%%

s=filter(b,a,y);

%plot(t,s);

S=fft(s,nfft);

magS=abs(S(1:nfft/2));

%figure,

%plot(f,magS);

%%

lowy=s;

nsr=sr/8; % This is generating the desired sampling rate.

dlowy(1)=lowy(1); % First sample

for k=1:fix(length(lowy)/8) % We go up to length(y)/8 sample,

% Which is the last sample to be taken

% Considering our sampling strategy.

dlowy(1,k+1)=y(8\*k,1); % THe 8\*k is taking each Kth sample;

% In turn, we already has stored the first sample

%

end

dlowyt=[0:length(ay)-1]/nsr; % Generating our time with respect to

% Each sampling rate;

t=[0:length(y)-1]/sr;

figure,

plot(dlowyt,dlowy),

title('dlowy vs t');

%%

d=dlowy;

D=fft(d,nfft);

magD=abs(D(1:nfft/2));

figure,

subplot(2,1,1),

plot(fAY,magD),

title('magDLOWY vs f');

subplot(2,1,2),

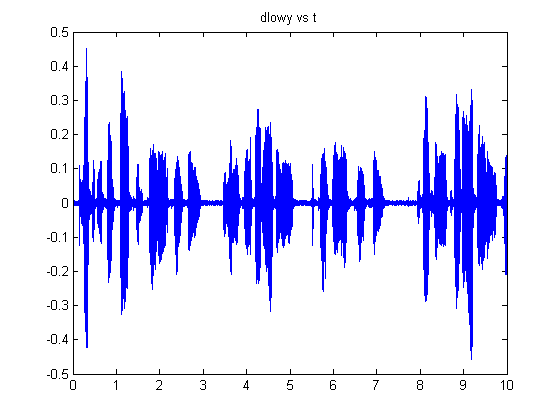
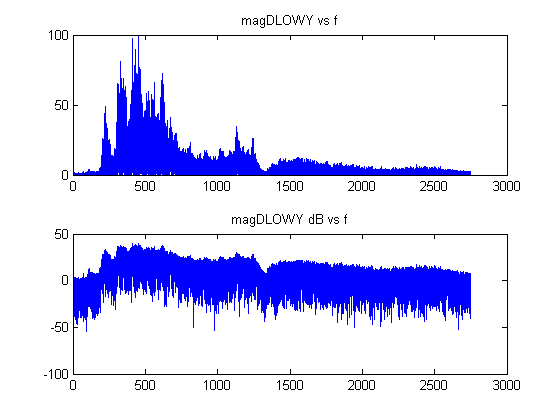
plot(fAY,20\*log10(magD));

title( 'magDLOWY dB vs f');

%% B 5

% soundsc(dlowy,nsr);

Below are the graphs which pertain to this set of code. The results consists of a signal which is significantly cleaner than the previous reults. By simple hearing of the final result it is possible to hear a cleaner voice, with what appears to be only noise from the recording (capturing) system.



**PART III**

**1**

Below is the code for Exercise 1, b.

%% 1 b

clear, clc, close all;

N=20;

x=ones(1,2\*(N-1)+1); % The +1 accounts for the 0;

om=[-3:0.0001:3]\*pi;

X=fft(x,N);

magX=abs(X);

magX=linspace(min(magX),max(magX),length(om));

% magX=min(magX):1./length(om)-1:max(magX)./length(om);

phaseX=angle(X);

% phaseX=min(phaseX):1./length(om)-1:max(magX)./length(om);

figure,

%subplot(2,1,1),

plot(om,magX),

title('magX vs om');

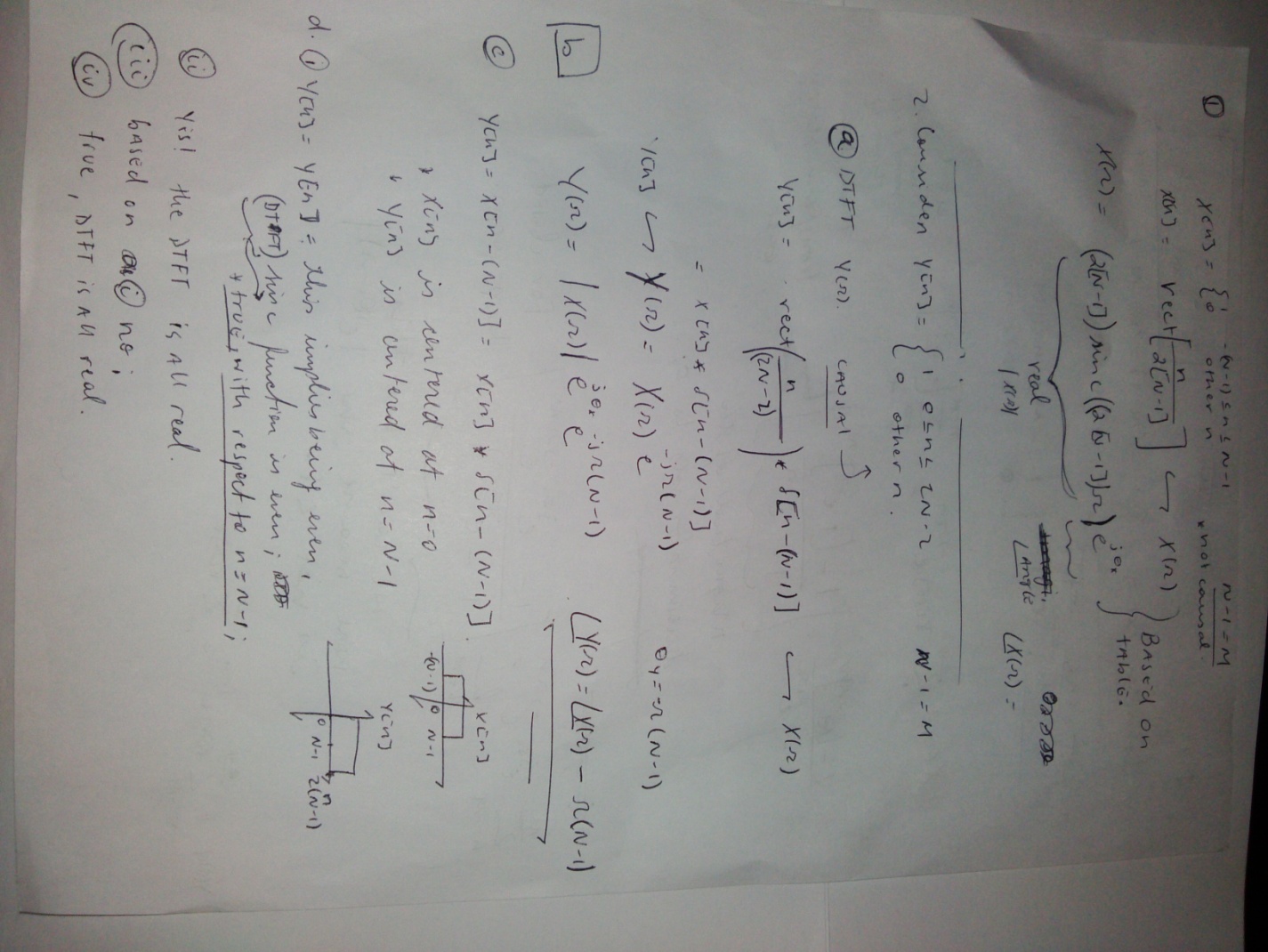
%subplot(2,1,2),

%plot(om,phaseX),

%title('phaseX vs om');



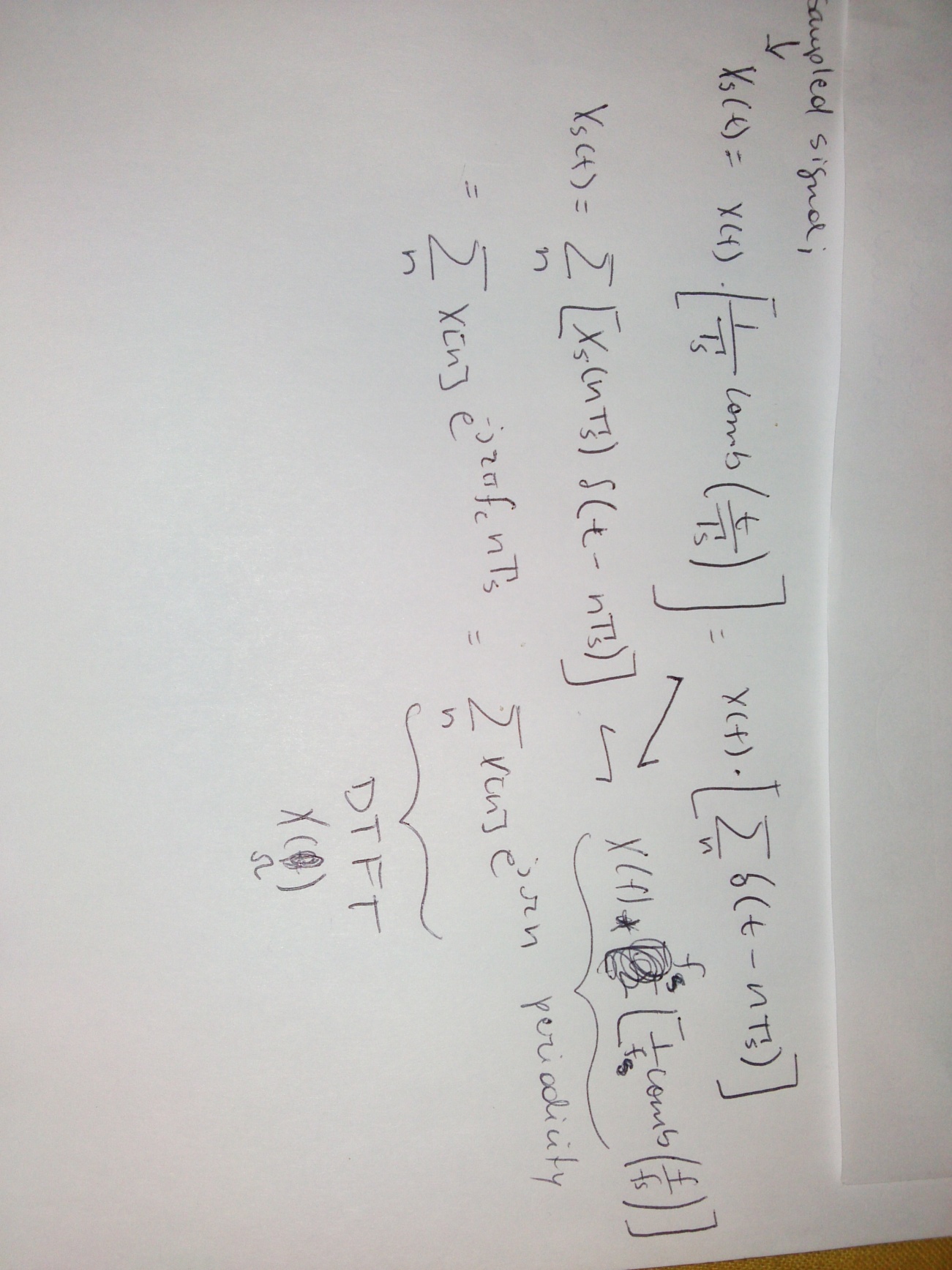
Below is the photo of work that pertains to Exercise one and two.



**C.**

**i.** MatLab does not agree with my calculations. According to my calculations, At Omega = 0, X(omega)= 2\*[N-1]. However, at 2\*pi and -2\*pi, it should equate.

**ii.**

****

**iii.** No, my matlab plot does not agree with my assessment of 1.c.i. I am assuming that it may be due to the fact that if you consider the x[n] as simply a discrete set of numbers, and you take N to be very large, you reach a point where |x[n]| is not < than 1, which happens to be a requirement for the DTFT to exist. I thus assume that this may have something to do with this lack of correlation between matlab results and analytical results. Regarding x[n] as a discrete rect function however changes the scenario, for we are able to directly go to the table and find an equivalent transform, the sinc function.

**D. i.** x[n] is even. with respect to n=0.

**ii.** true, all x[n] values are real, or 1.

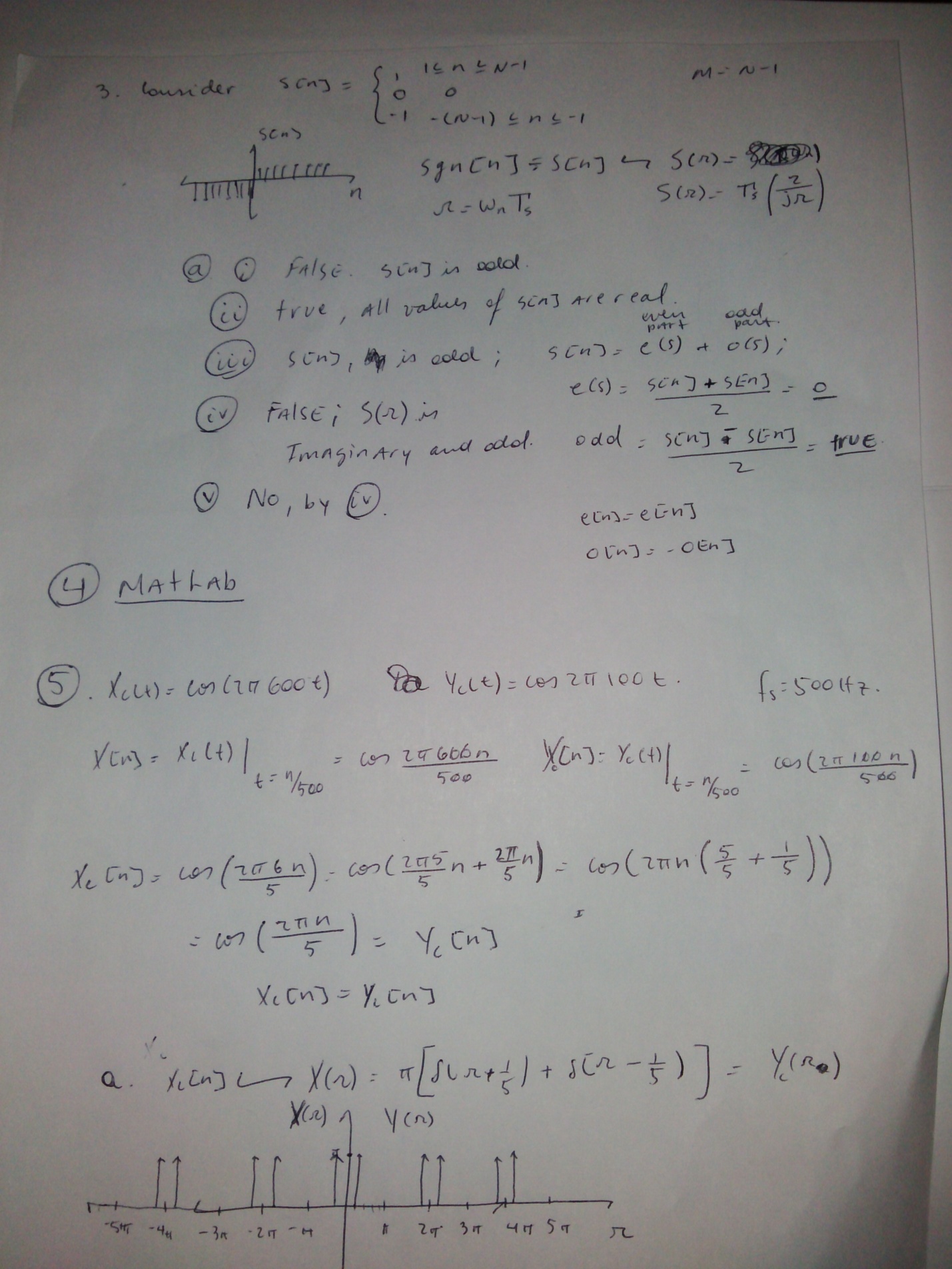
**iii.** False, Refer to statement in i.

**iv.** By the transform properties, it is real valued.

**v.**  It is also conjugate symmetric.

**e.**  The linear growth continued, and increased for every larger value of N.

**2** Number two is included in the picture above exercise C.

**3** Exercises 3 and 5 are included in the picutre below: 

**4** For question four, I will post the code, and later the graphs achieve. Posting the graphs, I will answer the analysis questions.

%% 4 a b

clear,clc,close all;

N=8;

x=ones(1,2\*(N-1)+1); % The +1 accounts for the 0;

nfft=[15 60 120];

X15=fft(x,15);

magX15=abs(X15(1:fix(nfft(1)/2)));

X60=fft(x,60);

magX60=abs(X60(1:fix(nfft(2)/2)));

X120=fft(x,120);

magX120=abs(X120(1:1:fix(nfft(3)/2)));

f15=[0:fix(length(X15)/2)-1]\*length(X15)/fix(length(X15)); % f in Hz. fs/2 -> pi;

f60=[0:fix(length(X60)/2)-1]\*length(X60)/fix(length(X60)); % f in Hz. fs/2 -> pi;

f120=[0:fix(length(X120)/2)-1]\*length(X120)/fix(length(X120)); % f in Hz. fs/2 -> pi;

om15=2\*pi\*f15/length(X15); % omega; 2\* pi \* f;

om60=2\*pi\*f60/length(X60);

om120=2\*pi\*f120/length(X120);

phaseX15=angle(X15(fix(1:length(X15)/2)));

phaseX60=angle(X60(fix(1:length(X60)/2)));

phaseX120=angle(X120(fix(1:length(X120)/2)));

figure,

subplot(3,1,1),

plot(om15,magX15),

hold,

plot(om15,phaseX15,'r'),

title('magX15 + phaseX15(red) vs om15'),

subplot(3,1,2),

plot(om60,magX60),

hold,

plot(om60,phaseX60,'r'),

title('magX60 + phaseX60(red) vs om60'),

subplot(3,1,3),

plot(om120,magX120),

hold,

plot(om120,phaseX120,'r'),

title('magX120 + phaseX120(red) vs om120');

%% 4 b.

x15=ifft(X15,15);

magx15=abs(x15(1:fix(nfft(1)/2)));

x60=ifft(X60,60);

magx60=abs(X60(1:fix(nfft(2)/2)));

x120=ifft(X120,120);

magx120=abs(x120(1:1:fix(nfft(3)/2)));

t15=[0:fix(length(x15))-1]/fix(length(x15));

t60=[0:fix(length(x60))-1]/fix(length(x60));

t120=[0:fix(length(x120))-1]/fix(length(x120));

phasex15=angle(x15);

phasex60=angle(x60);

phasex120=angle(x120);

figure,

subplot(3,1,1),

plot(t15,x15),

hold,

plot(t15,phasex15,'r'),

title('x15 + phasex15(red) vs t15'),

subplot(3,1,2),

plot(t60,x60),

hold,

plot(t60,phasex60,'r'),

title('x60 + phasex60(red) vs t60'),

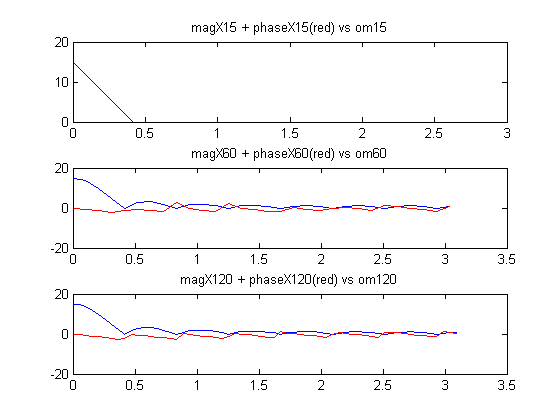
subplot(3,1,3),

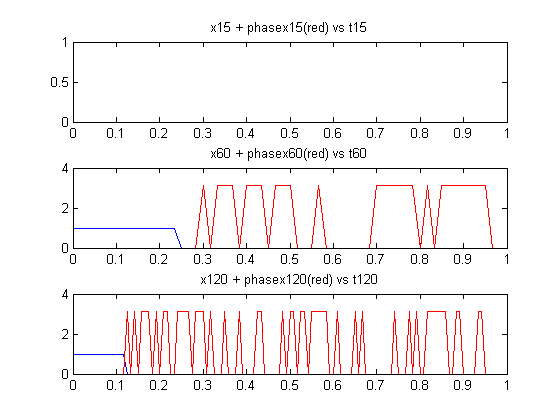
plot(t120,x120),

hold,

plot(t120,phasex120,'r'),

title('x120 + phasex120(red) vs t120');





The first graphs were as I predicted. However, the second graphs differ quite significantly from what I had in mind. I presume this may be due to the discrete numerical methods which Matlab utilizes, which significantly differ strict theoretical approaches.

**5**

Dr. Erdol, I would like to thank you for the effort you have put into teaching me and the rest of the class. Not only that but also the patience, second chances, intelligent comments and constructive criticism. I do understand I am nearly on the other side of the world, but I have to say, considering my time constraints and learning methods, being able to hear you teaching at 1.5 or even twice the regular speaking speed, or even skipping things I am already comfortable with, is very efficient (and funny sometimes). On the other hand, being able to stop you and make you repeat something 100 times without having you or the class frowning on me is effective.

I have found the subject of DSP to be extremely interesting and certainly plan to do further studies on it. I would love to do something related to bio signals. From cardiac to brain signals. If you have any suggestions for books or sources of information, please let me know. I will have sometime this summer to do my own studies, research and project.