Applications of DSP

Project 1

UNIVERSITY OF CENTRAL FLORIDA
DEPARTMENT OF ELECTRICAL ENGINEERING
February 14, 2017
Authored by**: Rajib Dey**

Applications of DSP

Project 1

Introduction

In this project, I have implemented the following system that was provided to us in the class.

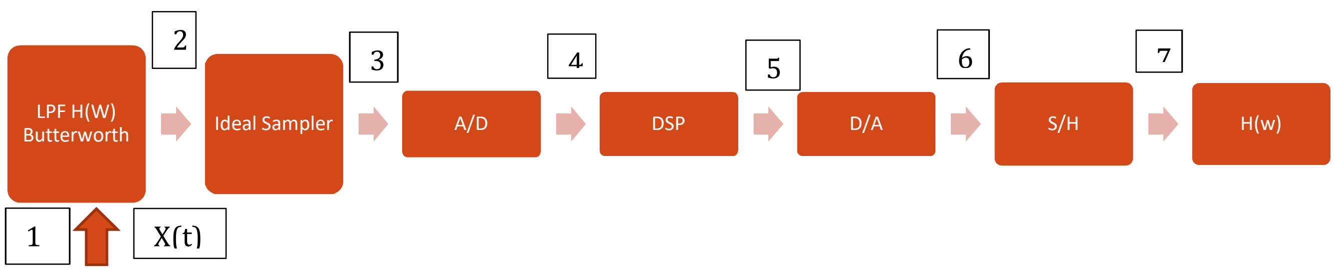

Figure 1: Design of the system

The given conditions are,

At [1],
$|x(w)|=1$, $\angle X(w)=0$. $H_A(w)$=antialiasing LPF, Butterworth, f_{3db}=2kHz
The useful band of input signal x(t)=0 to 2kHz

Problem a:

We calculated the Transfer Function of the 5^{th} and the 6^{th} order. As the Signal to Noise Ration at 6^{th} order turned out to be 63.2028, We can say the order of $H_A(w)$ is 6.

To calculate this, we took 100 points and the cut-off frequency $\omega c=2\pi*2000$. Then we used the following expressions to calculate the SNR:

$SNR=10*\log(\frac{S}{N})$

Where , $S= \sum_{L=0}^{N-1}(|X2(L\Delta\omega)|)*(|X2(L\Delta\omega)|)*\Delta\omega$

And $N = \sum_{L=0}^{N-1} X2(L\Delta\omega + 2\omega s) + X2\ (L\Delta\omega + \omega s) + X2(L\Delta\omega - \omega s) + X2(L\Delta\omega - 2\omega s)\hat{}2\ \Delta\omega$

```
SNR =

   63.2028

TF for the denormalized butterworth low pass filter is:
                                                  1
---------------------------------------------------------------------------------
                       5          4                    3          2
 6   8700290199134761 s     74641 s     2573137276597667 s     74641 s     38637 s
s  + ------------------ + -------- + ------------------ + -------- + ------- + 1
      2251799813685248      10000      281474976710656       10000     10000
```

Figure 2: Signal to Noise Ratio at 6th Order

<u>Problem b:</u>

Here, we create the Butterworth Filter using the given initialization values. Then we plot the Magnitude and Phase Signal Spectrum at point 2.

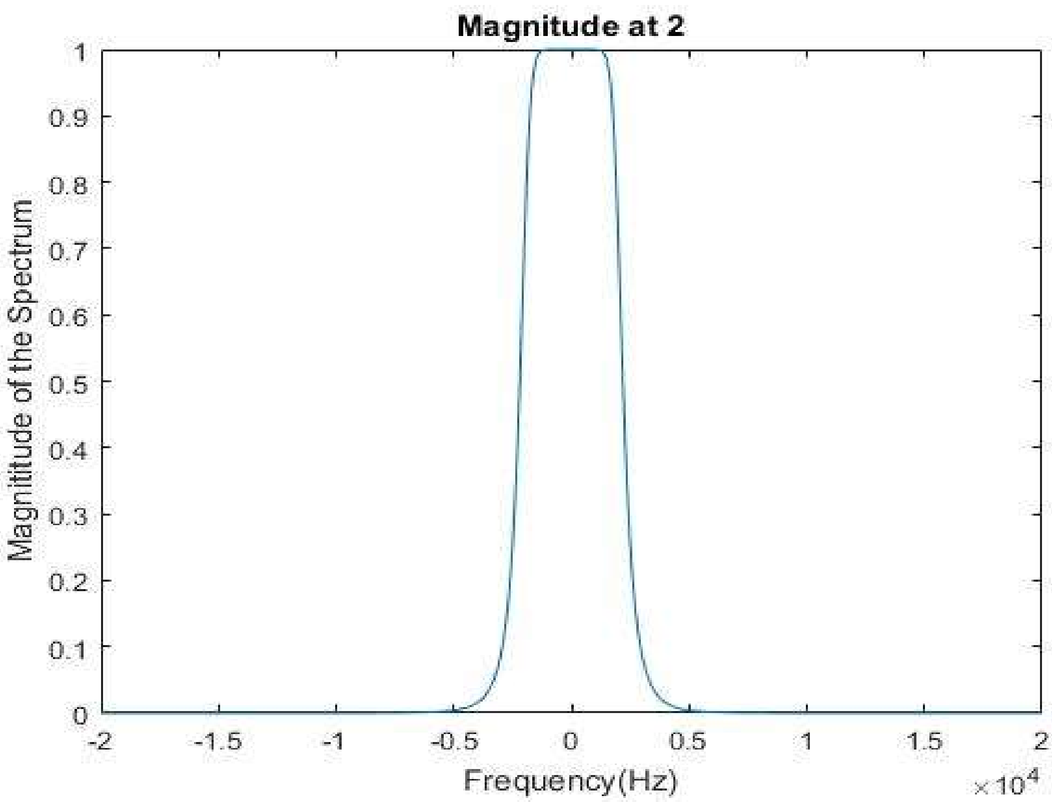

Figure 3: Magnitude Spectrum at 2

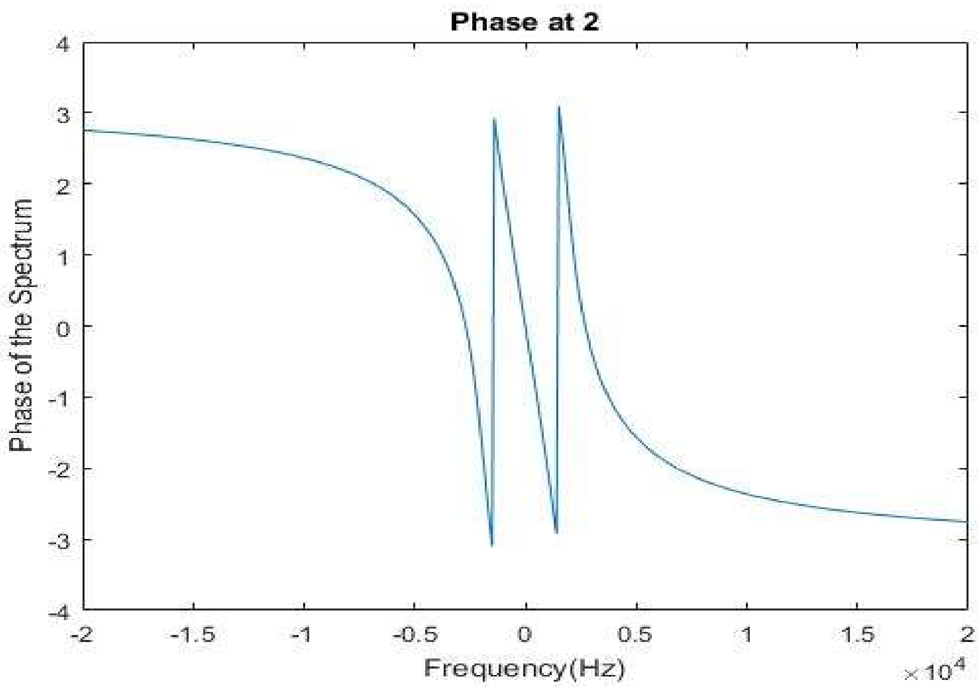

Figure 4: Phase Spectrum at 2

Problem c:

Here, we replaced $j\omega$ in butterworth filter with $j\omega$+or−ωs or $j\omega$+or−$2\omega s$ to create sample butterworth filter. Then we plotted the Magnitude and Phase diagram at point 3.

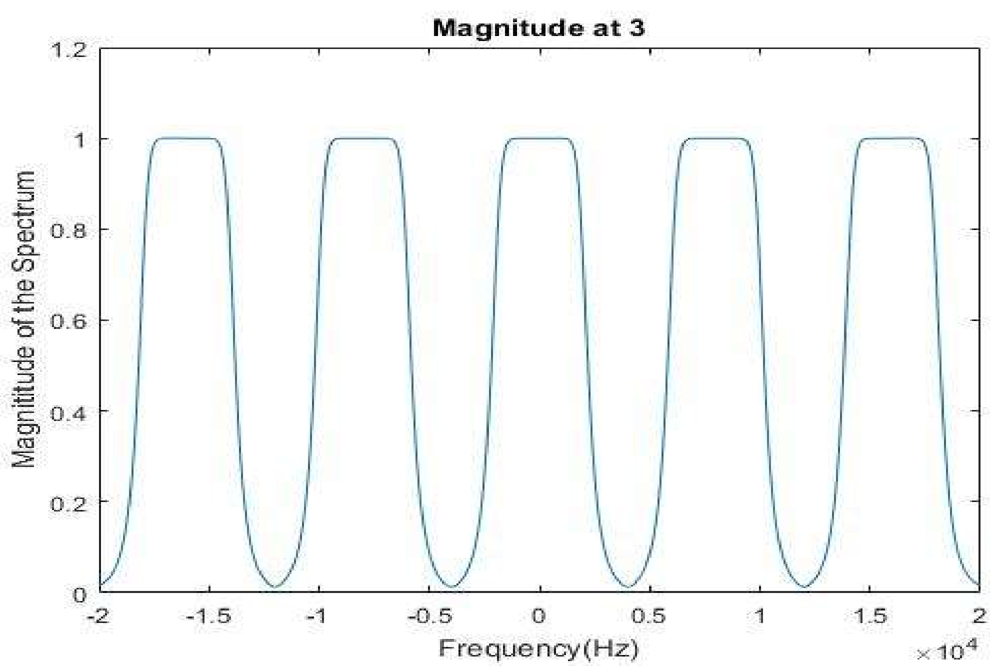

Figure 5: Magnitude Spectrum at 3

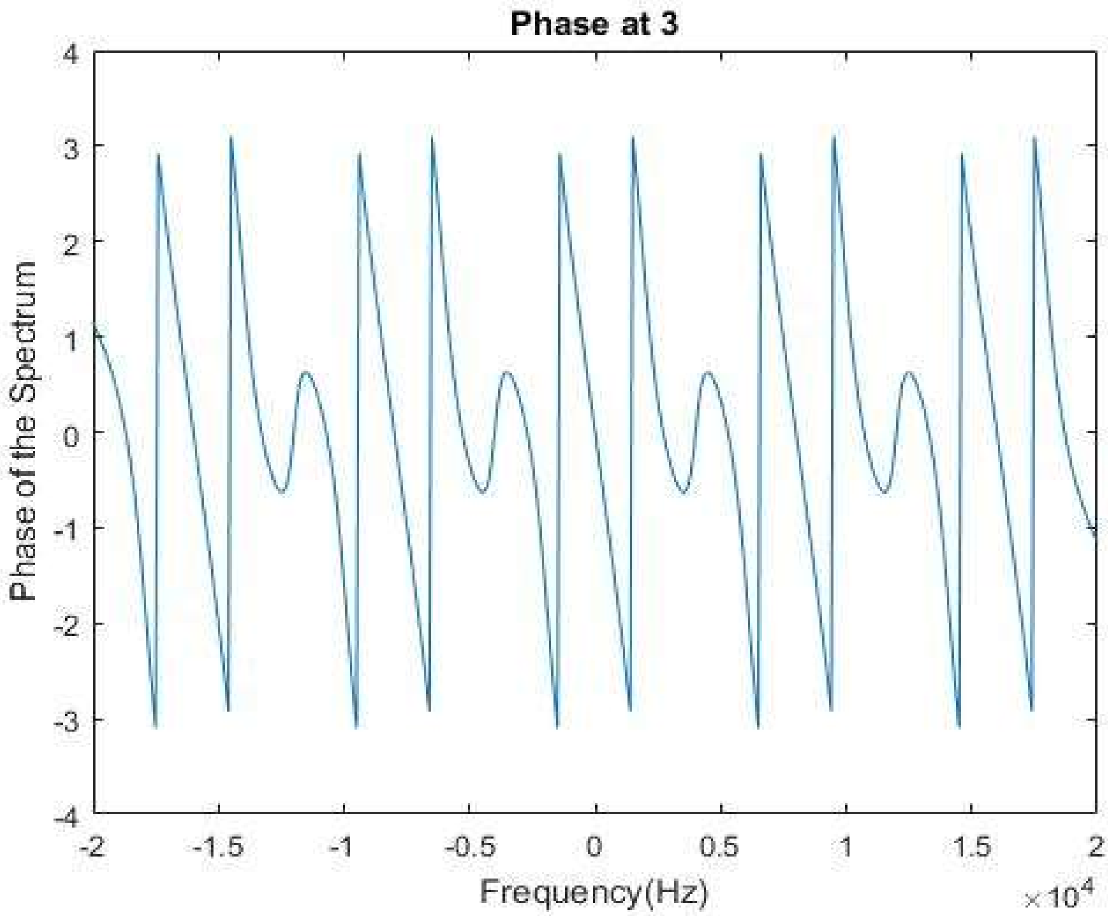

Figure 6: Phase Spectrum at 3

Problem d:

Here, we used the sawtooth function which gives us the triangular wave.

```
M=(1-sawtooth(w/8000,0.5))/2;
```

Then we multiply this triangular wave to the sample buttorworth filter we created in the previous problem. Then we plotted the Magnitude and Phase diagram at point 4.

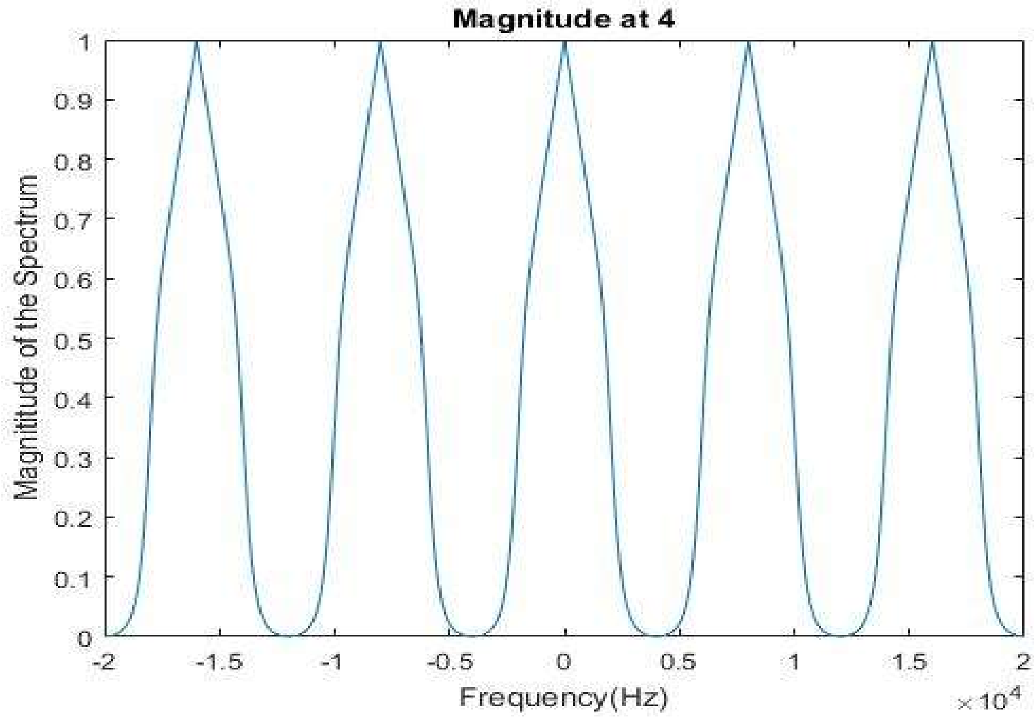

Figure 7: Magnitude Spectrum at 4

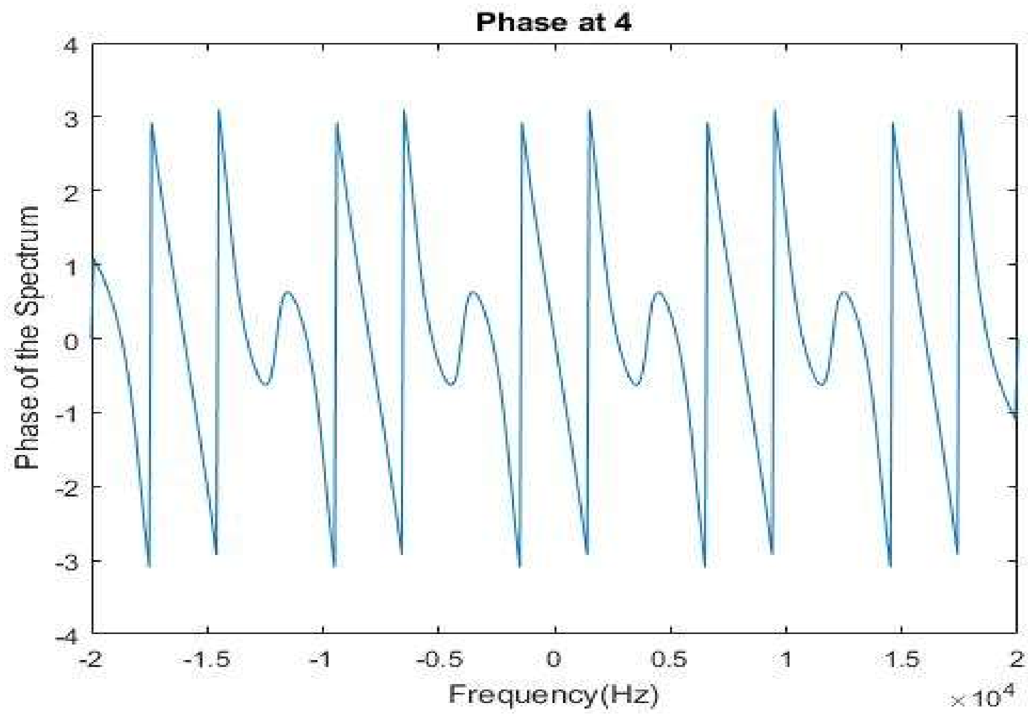

Figure 8: Phase Spectrum at 4

Problem e:

It passes through the sample and hold circuit after getting the output from the DSP block. It samples the signal at the given frequency and holds the value until the next sampling period. This is also known as the 'pulse widener'. To find the transfer function of the sample and hold subsystem we multiplied the previous signal with the following sinc function.

```
sinc(f/8000)
```

Then we plotted the Magnitude and Phase diagram at point 5.

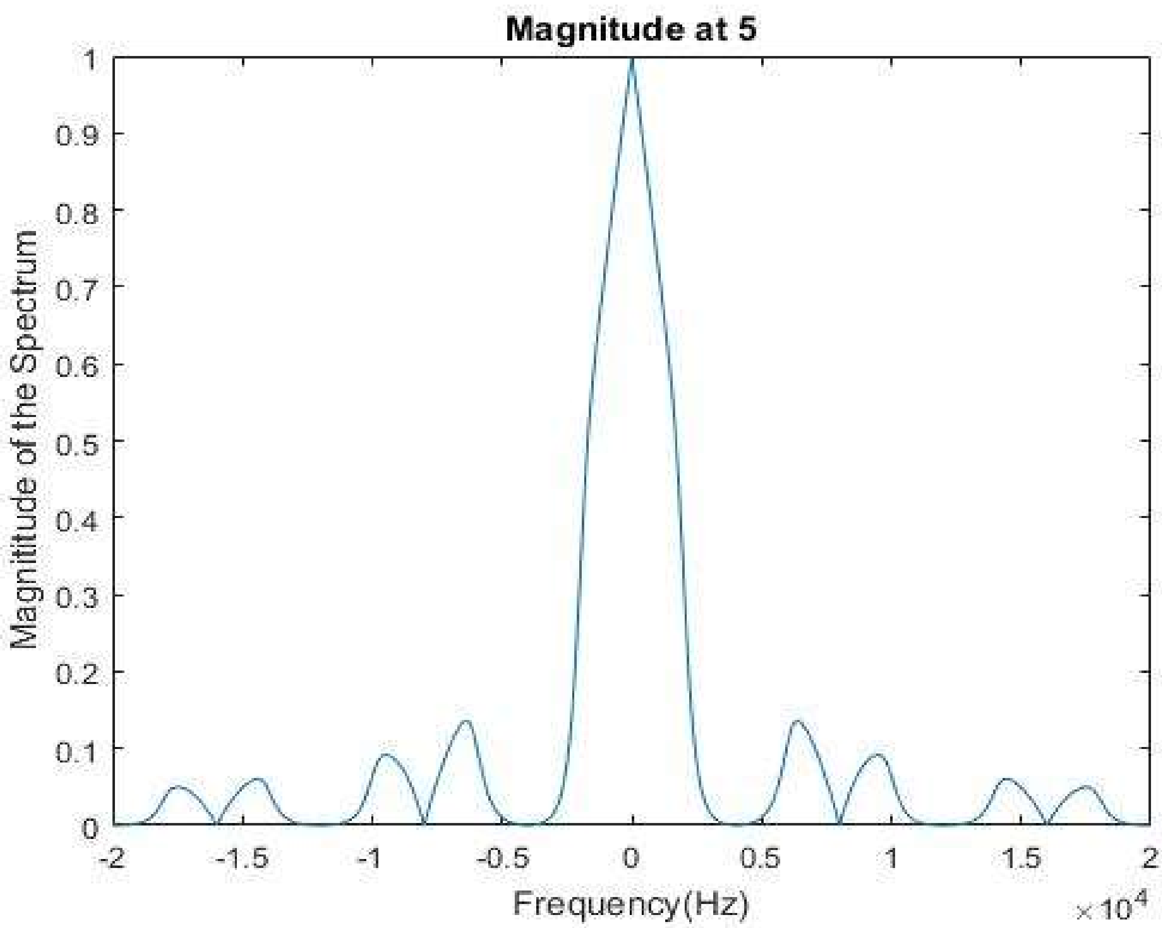

Figure 9: Magnitude Spectrum at 5

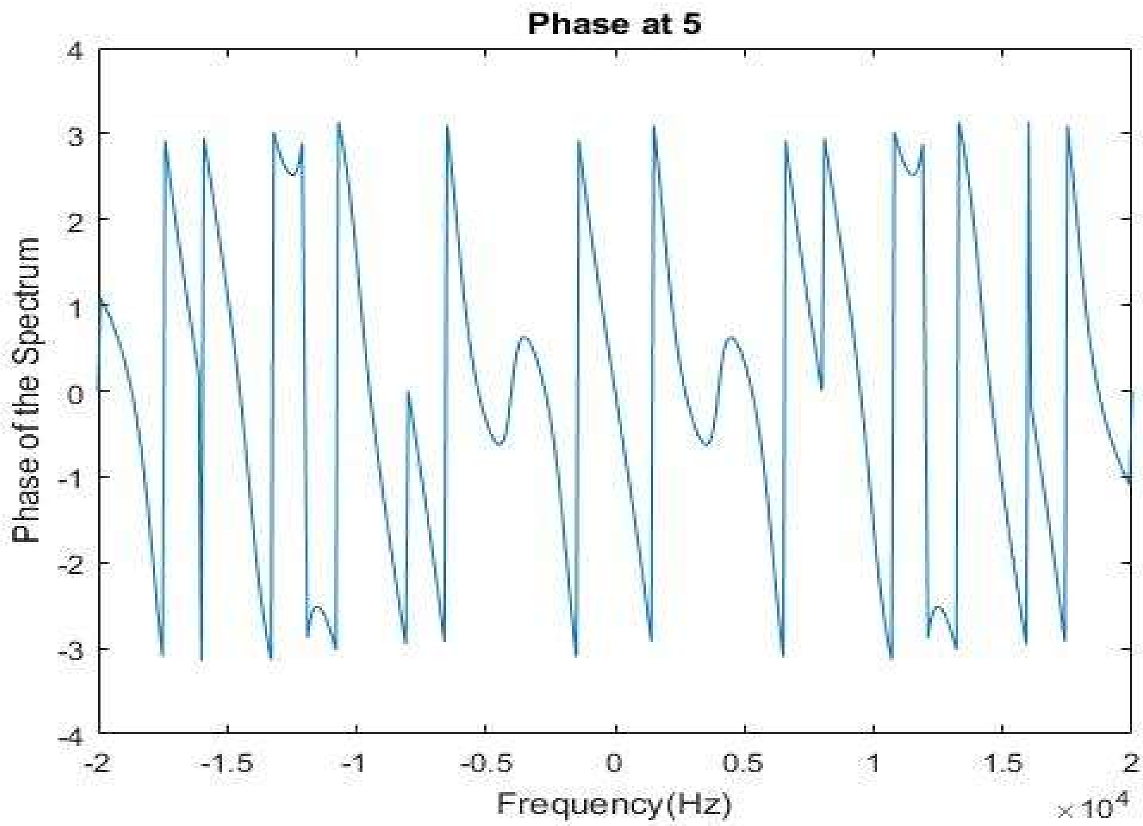

Figure 10: Phase Spectrum at 5

Problem f:

For the worst case analysis SNR_{jitter}>=60Db, The formula for the jitter noise

$SNR_{jitter}=3/2(\omega\tau^2)=3/16\pi^2=10\log(0.38*(Tsign\ \ /\tau)^2)$ becomes

$SNR_{jitter}\cong 10\log(4/100)+20\log(Tsign\ \ /\tau)$

By solving this, we get the maximum value of tau

```
MaxTau =
/    sqrt(358899852698093) sqrt(37778931862957161709568000000) \
| - --------------------------------------------------------- |
|                 37778931862957161709568000000               |
|                                                              |
|   sqrt(358899852698093) sqrt(37778931862957161709568000000)  |
|   ---------------------------------------------------------  |
\                37778931862957161709568000000                 /
```

Figure 11: Max Value of Tau

Problem g:

Here, We calculated $SNR_{quantiz}$ by using the following formulas

$SNR_{quantiz}=2^n$

$SNR_{quantiz}$ dB=10log(2^n)

Now step size $S=2V/2n$

So,

$SNR_{quantiz}=4V^2(2^n)^2/4V^2=2^{2n}$

$SNR_{quantiz}dB = 20n\log2 \cong 6n$

```
SNRquantize in dB =

SNRquantize =

   84.2884


SNRquantfinal =

    84
```

Figure 12: Value of $SNR_{quantiz}$

Problem h:

Here, we implemented HRw(j) by the following formula

```
HRw(j) = 1 ./ sinc(f(j)./8000)
```

Then we plotted HR(w).

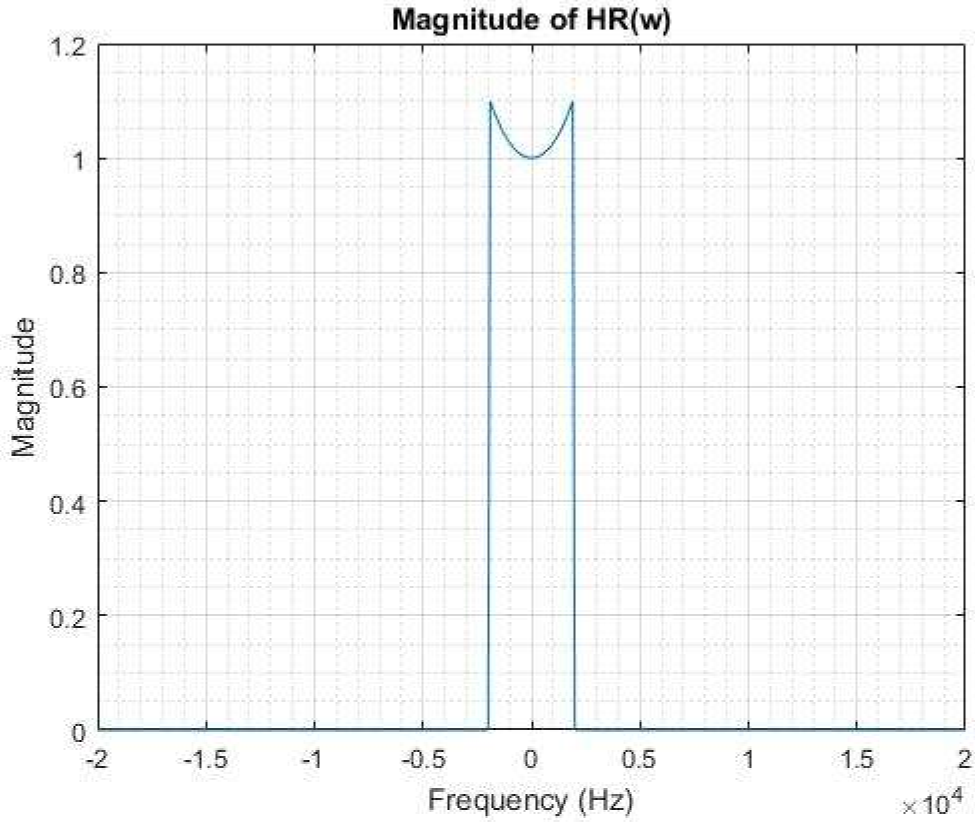

Figure 13: Magnitude Spectrum of HR(w)

Problem i:

Here, we multiplied HR(w) with the output at problem e to get the output at point 6. Then we plotted the Magnitude and Phase diagram at point 6.

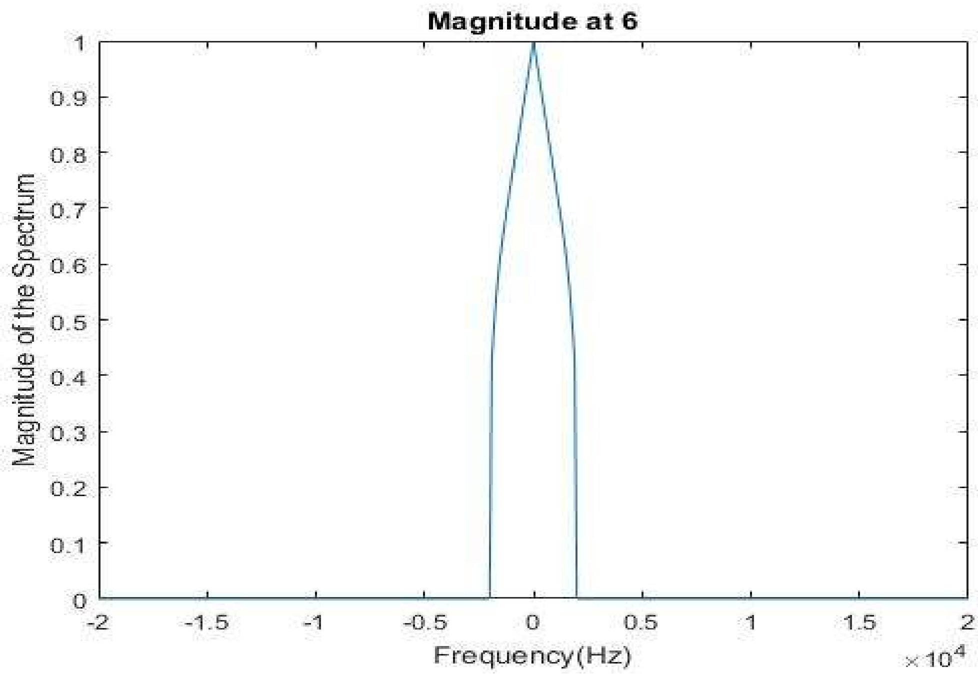

Figure 14: Magnitude Spectrum at 6

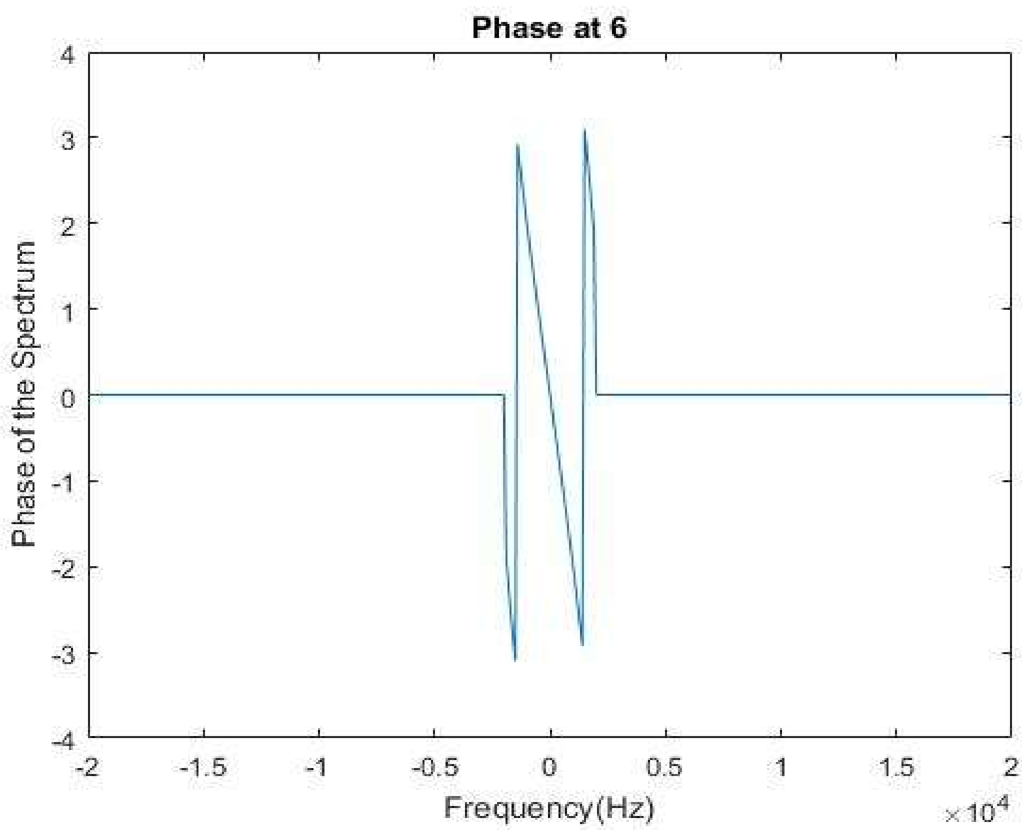

Figure 15: Phase Spectrum at 6

As there is noise in the system, generating from the components and from the sub-system the signal is not reconstructed perfectly.

Conclusion:

In this project we have worked on Low pass butterworth filter, ideal sampler, A/D converter and the sample and hold system. Then we used the reconstruction filter to get back the original signal. But due to the presence of noise, we could not get back the original signal.

The following is the codes and plots of the entire project in MATLAB.

MATLAB Code

```
%---------------------------------  Project 1 -------------------------
%--------------------------- Submitted by Rajib Dey -------------------------
close all
clc
clear all

n = 100;% number of points
B = 2*pi*2000;% 2000 Hz
dw = B/n;
wc = B;
syms s;
H1=1/(s+1)
H1=expand (H1)
% Denormalized 1st order Butterworth filter TF
DenormH1 = 1/((s/wc)+1)
pretty(DenormH1)
s1=0;
s2=0;
for L = 0:1:n-1
H1=1/((1j*L*dw)/(4000*pi) + 1);
absH1 = abs(H1);
s1 = s1 + (dw * absH1^2); % Signal for 1st order
end
ws = 2*pi*8000; % Sampling frequency 8k Hz
N1 = 0;
% Noise formula for both 1st order TFs
for L = 0:1:n-1
H1total=H1;
H1absolute=abs(H1total);
N1=N1+(dw*H1absolute^2);
end
%Antialiasing SNR for 1st order
```

```
SNRa1=10*log10(s1/N1)

%---------------------------------------------------------------------------
% Denormalized 5th and 6th order Butterworth filter TF
H5=1/((1+s)*(1+0.618*s+s^2)*(1+1.618*s+s^2)); % normalized 5th order TF
expand(H5)
% Denormalized 5th order Butterworth filter TF
DenH5 = 1/((s/wc)^5 + (809*(s/wc)^4)/250 +(1308981*(s/wc)^3)/250000 +
(1308981*(s/wc)^2)/250000 +(809*(s/wc))/250 + 1);
pretty(DenH5)
S6 = 0;
S5 = 0;
% Signal formula for 5th and 6th order TFs
for L = 0:n-1
H_Aw = 1./(2.53942e-25*(1j.*L.*dw).^6+1.23296e-20*(1j.*L.*dw).^5+2.99319e-
16*(1j.*L.*dw).^4+4.60671e-12*(1j.*L.*dw).^3+4.72667e-
8*(1j.*L.*dw).^2+3.07463e-4*(1j.*L.*dw).^1+1);
H5 = 1./((1j.*L.*dw).^5./(1024000000000000000*pi^5)
+(809.*(1j.*L.*dw).^4)./(64000000000000000*pi^4)
+(1308981.*(1j.*L.*dw).^3)./(16000000000000000*pi^3)
+(1308981.*(1j.*L.*dw).^2)./(4000000000000*pi^2)
+(809.*(1j.*L.*dw))./(1000000*pi) + 1);
absH_Aw = abs(H_Aw);
absH5 = abs(H5);
S6 = S6 + (dw * absH_Aw.^2); % Signal for 6th order
S5 = S5 + (dw * absH5.^2); % Signal for 5th order
end
N6 = 0;
N5 = 0;
% Noise formula for both 5th and 6th order TFs
for L = 0:n-1
H6pos1 = 1./(2.53942e-25*(1j.*(L.*dw-ws)).^6+1.23296e-20*(1j.*(L.*dw-
ws)).^5+2.99319e-16*(1j.*(L.*dw-ws)).^4+4.60671e-12*(1j.*(L.*dw-
ws)).^3+4.72667e-8*(1j.*(L.*dw-ws)).^2+3.07463e-4*(1j.*(L.*dw-ws)).^1+1);
H6neg1 = 1./(2.53942e-25*(1j.*(L.*dw+ws)).^6+1.23296e-
20*(1j.*(L.*dw+ws)).^5+2.99319e-16*(1j.*(L.*dw+ws)).^4+4.60671e-
12*(1j.*(L.*dw+ws)).^3+4.72667e-8*(1j.*(L.*dw+ws)).^2+3.07463e-
4*(1j.*(L.*dw+ws)).^1+1);
H6pos2 = 1./(2.53942e-25*(1j.*(L.*dw-2*ws)).^6+1.23296e-20*(1j.*(L.*dw-
2*ws)).^5+2.99319e-16*(1j.*(L.*dw-2*ws)).^4+4.60671e-12*(1j.*(L.*dw-
2*ws)).^3+4.72667e-8*(1j.*(L.*dw-2*ws)).^2+3.07463e-4*(1j.*(L.*dw-
2*ws)).^1+1);
H6neg2 = 1./(2.53942e-25*(1j.*(L.*dw+2*ws)).^6+1.23296e-
20*(1j.*(L.*dw+2*ws)).^5+2.99319e-16*(1j.*(L.*dw+2*ws)).^4+4.60671e-
12*(1j.*(L.*dw+2*ws)).^3+4.72667e-8*(1j.*(L.*dw+2*ws)).^2+3.07463e-
4*(1j.*(L.*dw+2*ws)).^1+1);
H6pos3 = 1./(2.53942e-25*(1j.*(L.*dw-3*ws)).^6+1.23296e-20*(1j.*(L.*dw-
3*ws)).^5+2.99319e-16*(1j.*(L.*dw-3*ws)).^4+4.60671e-12*(1j.*(L.*dw-
3*ws)).^3+4.72667e-8*(1j.*(L.*dw-3*ws)).^2+3.07463e-4*(1j.*(L.*dw-
3*ws)).^1+1);
H6neg3 = 1./(2.53942e-25*(1j.*(L.*dw+3*ws)).^6+1.23296e-
20*(1j.*(L.*dw+3*ws)).^5+2.99319e-16*(1j.*(L.*dw+3*ws)).^4+4.60671e-
12*(1j.*(L.*dw+3*ws)).^3+4.72667e-8*(1j.*(L.*dw+3*ws)).^2+3.07463e-
4*(1j.*(L.*dw+3*ws)).^1+1);
H6Total = H6pos1+H6neg1+H6pos2+H6neg2+H6pos3+H6neg3;
absH6Total = abs(H6Total);
```

```
H51 = 1./((1j.*(L.*dw-ws)).^5./(1024000000000000000*pi^5) +
(809.*(1j.*(L.*dw-ws)).^4)./(64000000000000000*pi^4) + (1308981.*(1j.*(L.*dw-
ws)).^3)./(16000000000000000*pi^3) + (1308981.*(1j.*(L.*dw-
ws)).^2)./(4000000000000*pi^2) + (809.*(1j.*(L.*dw-ws)))./(1000000*pi) + 1);
H5neg1 =1./((1j.*(L.*dw+ws)).^5./(1024000000000000000*pi^5)
+(809.*(1j.*(L.*dw+ws)).^4)./(64000000000000000*pi^4)
+(1308981.*(1j.*(L.*dw+ws)).^3)./(16000000000000000*pi^3)
+(1308981.*(1j.*(L.*dw+ws)).^2)./(4000000000000*pi^2)
+(809.*(1j.*(L.*dw+ws)))./(1000000*pi) + 1);
H52 = 1./((1j.*(L.*dw-2*ws)).^5./(1024000000000000000*pi^5) +
(809.*(1j.*(L.*dw-2*ws)).^4)./(64000000000000000*pi^4)
+(1308981.*(1j.*(L.*dw-2*ws)).^3)./(16000000000000000*pi^3)+
(1308981.*(1j.*(L.*dw-2*ws)).^2)./(4000000000000*pi^2) +(809.*(1j.*(L.*dw-
2*ws)))./(1000000*pi) + 1);
H5neg2 =1./((1j.*(L.*dw+2*ws)).^5./(1024000000000000000*pi^5)
+(809.*(1j.*(L.*dw+2*ws)).^4)./(64000000000000000*pi^4)
+(1308981.*(1j.*(L.*dw+2*ws)).^3)./(16000000000000000*pi^3)+
(1308981.*(1j.*(L.*dw+2*ws)).^2)./(4000000000000*pi^2)
+(809.*(1j.*(L.*dw+2*ws)))./(1000000*pi) + 1);
H5Total = H51 + H5neg1 + H52 + H5neg2;
absH5Total = abs(H5Total);
N6 = N6 + (dw * (absH6Total).^2);% Noise for 6th order
N5 = N5 + (dw * (absH5Total).^2);% Noise for 5th order
end
% Signal to noise ratio of 5th order
SNR5 = 10*log10(S5 ./ N5)
% Signal to noise ratio of 6th order
SNR = 10*log10(S6 ./ N6)
disp('TF for the denormalized butterworth low pass filter is:')
normHs=1./(s^6+3.863705*s^5+7.46410*s^4+9.14162*s^3+7.46410*s^2+3.86370*s+1);
simplify(normHs);
pretty(normHs);
denormHs=1./((s/wc)^6+3.863705*(s/wc)^5+7.46410*(s/wc)^4+9.14162*(s/wc)^3+7.4
6410*(s/wc)^2+3.86370*(s/wc)+1)
denormHs1=expand(denormHs)
simplify(denormHs1);
pretty(denormHs1);
[N,D] = numden(denormHs1)
%-------------------------------------------------------------------
% Problem b (Sketch the spectrum at 2)
% Initialization
f=-20000:100:20000;
w=2*pi*f;
fs=8000;
ws=2*pi*fs;
% Create Butterworth Filter
Butterworth=inline('1./(2.53942e-25*(j*w).^6+1.23296e-20*(j*w).^5+2.99319e-
16*(j*w).^4+4.60671e-12*(j*w).^3+4.72667e-8*(j*w).^2+3.07463e-
4*(j*w)+1)','w');
figure(1);
plot(f,abs(Butterworth(w)));
title('Magnitude at 2')
xlabel('Frequency(Hz)');
ylabel('Magnititude of the Spectrum');
figure(2);
plot(f,angle(Butterworth(w)));
title('Phase at 2')
```

```
xlabel('Frequency(Hz)');
ylabel('Phase of the Spectrum');
%-------------------------------------------------------------------
% Problem c (Sketch the spectrum at 3)
sample_Butterworth=Butterworth(w+2*ws)+Butterworth(w+ws)+Butterworth(w)+Butte
rworth(w-ws)+Butterworth(w-2*ws);
figure(3);
plot(f,abs(sample_Butterworth));
title('Magnitude at 3')
xlabel('Frequency(Hz)');
ylabel('Magnititude of the Spectrum');
figure(4);
plot(f,angle(sample_Butterworth));
title('Phase at 3')
xlabel('Frequency(Hz)');
ylabel('Phase of the Spectrum');
%-------------------------------------------------------------------
% Problem d (Sketch the spectrum at 4)
M=(1-sawtooth(w/8000,0.5))/2;
output_4=sample_Butterworth.*M;
figure(5);
plot(f,abs(output_4));
title('Magnitude at 4')
xlabel('Frequency(Hz)');
ylabel('Magnititude of the Spectrum');
figure(6);
plot(f,angle(output_4));
title('Phase at 4')
xlabel('Frequency(Hz)');
ylabel('Phase of the Spectrum');
%------------------------------------------------------------------
% Problem e (Sketch the spectrum at 5)
SH=sinc(f/8000);
output_5=output_4.*SH;
figure(7);
plot(f,abs(output_5));
title('Magnitude at 5')
xlabel('Frequency(Hz)');
ylabel('Magnititude of the Spectrum');
figure(8);
plot(f,angle(output_5));
title('Phase at 5')
xlabel('Frequency(Hz)');
ylabel('Phase of the Spectrum');

%-------------------------------------------------------------------
% Problem f ( Finding the maximum tau ( worst case analysis )
syms tau;
Tsignal = 1/2000;
taumax = solve(10*log10(0.038*Tsignal^2/tau^2)>=60,tau); % From SNRjitter
expression
disp('MaxTau = ')
pretty (2*taumax)
%-------------------------------------------------------------------
% Problem g ( Compute SNR quantiz in dB)
n=14;
disp('SNRquantize in dB = ')
```

```
SNRquantize = 20*n*log10(2)
SNRquantfinal=6*n
%-------------------------------------------------------------------
% Problem h ( Sketch HRw )
for i = -200:1:200
j = i+201;
f(j) = 100*i;
if i<=-20
HRw(j) = 0;
elseif i >= 20
HRw(j) = 0;
else
HRw(j) = 1 ./ sinc(f(j)./8000);% 8000 refers to 1/taw,taw = 125micro
end
end
figure(9)
plot(f,abs(HRw))
title('Magnitude of HR(w)')
xlabel('Frequency (Hz)')
ylabel('Magnitude')
grid on
grid minor
%-------------------------------------------------------------------
% Problem i ( Sketch spectrum at 6 )
output_6=output_5.*HRw;
figure(10);
plot(f,abs(output_6));
title('Magnitude at 6')
xlabel('Frequency(Hz)');
ylabel('Magnitude of the Spectrum');
figure(11);
plot(f,angle(output_6));
title('Phase at 6')
xlabel('Frequency(Hz)');
ylabel('Phase of the Spectrum');
```