Analog MATLAB project

Name	ID
Asmaa Hassan Mokhtar Aboshady	**19015430**
Perihan Hossam Eldin Imam	**20010392**
Abdelraouf Fathy Abdelraouf	**20010777**
Abdelrahman Mahmoud Mohamed Eldesoky	**20010860**

Experiment 1 DSBSC

The code:

```
clear all
close all
clc
%% Generating Signal
[Mt,fm] =audioread('eric.wav');
sound(Mt,fm);
fprintf('Original Sound is playing.. \n')
pause(length(Mt)/fm);
t=linspace(0,length(Mt)/fm,length(Mt));
figure('Name','Audio Signal');
subplot(4,1,1);
plot(t,Mt);
title('Time Domain');
Sf=fftshift(fft(Mt));
Sfvec=linspace(-fm/2,fm/2,length(Sf));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide ');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% filter
filter=ones(length(Mt),1);
f1=round((-4*1000+fm/2).*(length(Mt)/fm));
f2=round((4*1000+fm/2).*(length(Mt)/fm));
filter([1:f1  f2:end])=0;
Mf_Filtered=Sf;
Mf_Filtered=Mf_Filtered.*filter;
Mf_FilteredMag=abs(Mf_Filtered);
Mf_FilteredPhase=angle(Mf_Filtered);
Mt_Filtered=real(ifft(ifftshift(Mf_Filtered)));
sound(Mt_Filtered,fm);
fprintf('Filtered Sound is playing.. \n')
pause(length(Mt_Filtered)/fm);
```

```matlab
figure('Name','Filtered Audio Signal');
subplot(4,1,1);
plot(t,Mt_Filtered);
title('Time Domain');
subplot(4,1,2);
plot(Sfvec,Mf_Filtered);
title('Frequency Doamin');
subplot(4,1,3);
plot(Sfvec,Mf_FilteredMag);
title('Magneduide');
subplot(4,1,4);
plot(Sfvec,Mf_FilteredPhase);
title('Phase');
%% resample
fc=100000;
fs=5*fc;
Mt_Fil_res=resample(Mt_Filtered,fs,fm) ;
t1=linspace(0,length(Mt_Fil_res)/fs,length(Mt_Fil_res));
figure('Name','Resampled Filtered Audio Signal');
subplot(4,1,1);
plot(t1, Mt_Fil_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mt_Fil_res));
resvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_fil_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_fil_resphase);
title('Phase');
%% carrier
carrier_signal = cos(2*pi*fc*t1);
Ct=transpose(carrier_signal);
figure('Name','Carrier Signal');
subplot(4,1,1);
plot(t1, Ct);
title('Time Domain');
Sf=fftshift(fft(Ct));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
```

```
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% Modulated Signal
St=Mt_Fil_res .* Ct;
figure('Name','Modulated Signal');
subplot(4,1,1);
plot(t1, St);
title('Time Domain');
Sf=fftshift(fft(St));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% envelope detection
envolpe=abs(hilbert(St));
figure('Name','Recieved Signal with envelope detection');
subplot(4,1,1);
plot(t1, envolpe);
title('Time Domain');
Sf=fftshift(fft(envolpe));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
```

```
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% resample in envolpe detection
Mr_env_res=resample(envolpe,fm,fs) ;
t2=linspace(0,length(Mr_env_res)/fm,length(Mr_env_res));
sound(Mr_env_res,fm);
fprintf('Recieved Signal with envelope detection is playing.. \n');
pause(length(Mr_env_res)/fm);
figure('Name','Resampled Recieved Signal with envelope detection');
subplot(4,1,1);
plot(t2, Mr_env_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_env_res));
resvec=linspace(-fs/2,fs/2,length(Mr_env_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_env_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_env_resphase);
title('Phase');
%% coherent detection
Vt=St .* Ct;
figure('Name','Recieved Signal with coherent detection');
subplot(4,1,1);
plot(t1, Vt);
title('Time Domain');
Vf=fftshift(fft(Vt));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Vf);
title('Frequency Doamin');
Sfmag=abs(Vf);
Sfphase=angle(Vf);
subplot(4,1,3);
```

```
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% resample in coherent detection
Mr_coh_res=resample(Vt,fm,fs) ;
t3=linspace(0,length(Mr_coh_res)/fm,length(Mr_coh_res));
sound(Mr_coh_res,fm);
fprintf('Recieved Signal with coherent detection is
playing.. \n');
pause(length(Mr_coh_res)/fm);
figure('Name','Resampled Recieved Signal with coherent
detection');
subplot(4,1,1);
plot(t3, Mr_coh_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_coh_res));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_fil_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_fil_resphase);
title('Phase');
%% Adding Noise
%% SNR 0dB
Mr_SNR1=awgn(Mr_coh_res,0);
sound(Mr_SNR1,fm);
fprintf('Recieved Signal with coherent detection with SNR
0dB is playing.. \n');
pause(length(Mr_SNR1)/fm);
figure('Name','Resampled Recieved Signal with SNR 0dB ');
subplot(4,1,1);
plot(t3, Mr_SNR1);
title('Time Domain');
Mf_SNR1=fftshift(fft(Mr_SNR1));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
```

```
plot(resvec,Mf_SNR1);
title('Frequency Doamin');
Mf_SNR1_resmag=abs(Mf_SNR1);
Mf_SNR1_resphase=angle(Mf_SNR1);
subplot(4,1,3);
plot(resvec,Mf_SNR1_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_SNR1_resphase);
title('Phase');

%% SNR 10dB
Mr_SNR2=awgn(Mr_coh_res,10);
sound(Mr_SNR2,fm);
fprintf('Recieved Signal with coherent detection with SNR
10dB is playing.. \n');
pause(length(Mr_SNR2)/fm);
figure('Name','Resampled Recieved Signal with SNR 10dB');
subplot(4,1,1);
plot(t3, Mr_SNR2);
title('Time Domain');
Mf_SNR2=fftshift(fft(Mr_SNR2));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_SNR2);
title('Frequency Doamin');
Mf_SNR2_resmag=abs(Mf_SNR2);
Mf_SNR2_resphase=angle(Mf_SNR2);
subplot(4,1,3);
plot(resvec,Mf_SNR2_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_SNR2_resphase);
title('Phase');

%% SNR 30dB
Mr_SNR3=awgn(Mr_coh_res,30);
fprintf('Recieved Signal with coherent detection with SNR
30dB is playing.. \n');
sound(Mr_SNR3,fm);
pause(length(Mr_SNR3)/fm);
figure('Name','Resampled Recieved Signal with SNR 30dB');
subplot(4,1,1);
plot(t3, Mr_SNR3);
```

```
title('Time Domain');
Mf_SNR3=fftshift(fft(Mr_SNR3));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_SNR3);
title('Frequency Doamin');
Mf_SNR3_resmag=abs(Mf_SNR3);
Mf_SNR3_resphase=angle(Mf_SNR3);
subplot(4,1,3);
plot(resvec,Mf_SNR3_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_SNR3_resphase);
title('Phase');
%% coherent detection with Frequency error
%% Frequency error =100
Carrier_freq_error_1 = cos(2*pi*(fs+100*(10^3))*t1);
Ct_freq_error_1=transpose(Carrier_freq_error_1);
Vt=St .* Ct_freq_error_1;
figure('Name','Recieved Signal with Frequency error =100');
subplot(4,1,1);
plot(t1, Vt);
title('Time Domain');
Vf=fftshift(fft(Vt));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Vf);
title('Frequency Doamin');
Sfmag=abs(Vf);
Sfphase=angle(Vf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% resample in coherent detection with freq error
Mr_coh_res=resample(Vt,fm,fs) ;
t3=linspace(0,length(Mr_coh_res)/fm,length(Mr_coh_res));
sound(Mr_coh_res,fm);
fprintf('Recieved Signal with coherent detection with 
Frequency error =100 is playing.. \n');
pause(length(Mr_coh_res)/fm);
```

```
figure('Name','Resampled Recieved Signal with Frequency
error =100');
subplot(4,1,1);
plot(t3, Mr_coh_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_coh_res));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_fil_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_fil_resphase);
title('Phase');
%% Frequency error =100.1
Carrier_freq_error_2 = cos(2*pi*(fs+100*(10^3))*t1);
Ct_freq_error_2=transpose(Carrier_freq_error_2);
Vt=St .* Ct_freq_error_2;
figure('Name','Recieved Signal with Frequency error
=100.1');
subplot(4,1,1);
plot(t1, Vt);
title('Time Domain');
Vf=fftshift(fft(Vt));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Vf);
title('Frequency Doamin');
Sfmag=abs(Vf);
Sfphase=angle(Vf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
%% resample in coherent detection with freq error
Mr_coh_res=resample(Vt,fm,fs) ;
t3=linspace(0,length(Mr_coh_res)/fm,length(Mr_coh_res));
sound(Mr_coh_res,fm);
```

```
fprintf('Recieved Signal with coherent detection with
Frequency error =100.1 is playing.. \n');
pause(length(Mr_coh_res)/fm);
figure('Name','Resampled Recieved Signal with Frequency
error =100.1');
subplot(4,1,1);
plot(t3, Mr_coh_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_coh_res));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_fil_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_fil_resphase);
title('Phase');
%% phase error
%% error = 20 degree
Carrier_phase_error = cos(2*pi*fc*t1+((20*pi)/180));
Ct_phase=transpose(Carrier_phase_error);
Vt=St .* Ct_phase;
figure('Name','Recieved Signal with phase error = 20
degree');
subplot(4,1,1);
plot(t1, Vt);
title('Time Domain');
Vf=fftshift(fft(Vt));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Vf);
title('Frequency Doamin');
Sfmag=abs(Vf);
Sfphase=angle(Vf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
```

```
%% resample in coherent detection with phase error
Mr_coh_res=resample(Vt,fm,fs) ;
t3=linspace(0,length(Mr_coh_res)/fm,length(Mr_coh_res));
sound(Mr_coh_res,fm);
fprintf('Recieved Signal with coherent detection with phase
error = 20 degree is playing.. \n');
pause(length(Mr_coh_res)/fm);
figure('Name','Resampled Recieved Signal with phase error =
20 degree');
subplot(4,1,1);
plot(t3, Mr_coh_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_coh_res));
resvec=linspace(-fs/2,fs/2,length(Mr_coh_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_fil_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_fil_resphase);
title('Phase');
```

The outputs:

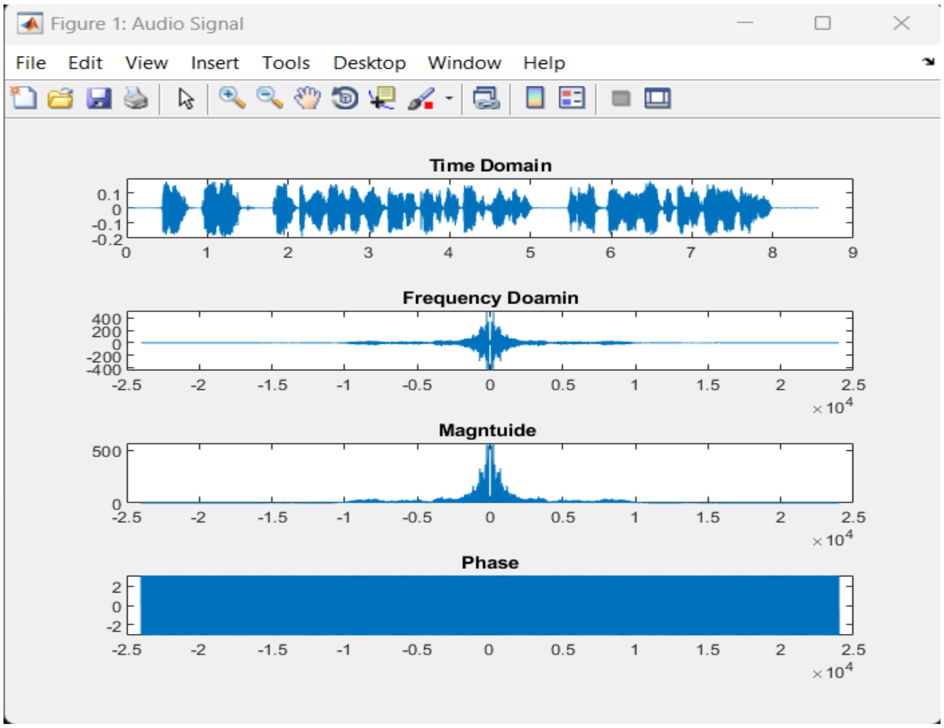


Figure 1: Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

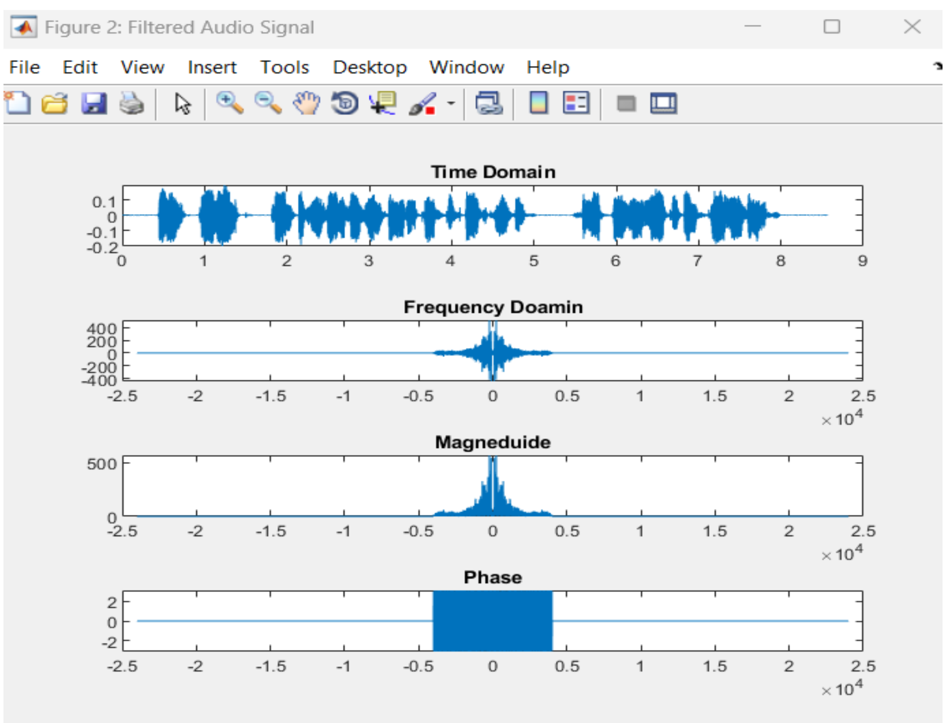


Figure 2: Filtered Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

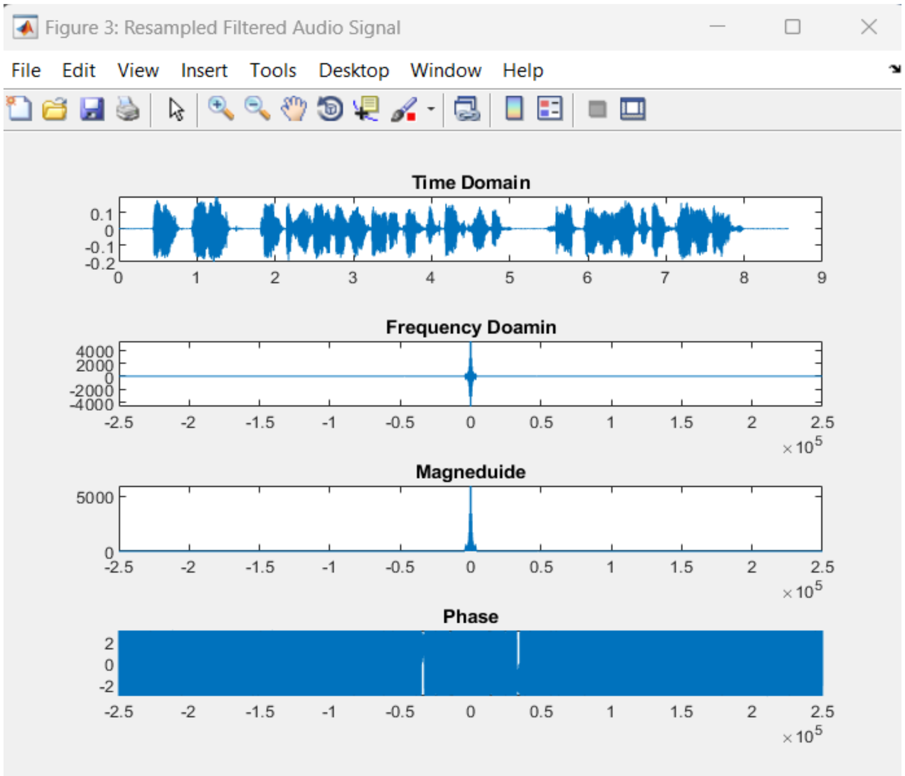


Figure 3: Resampled Filtered Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

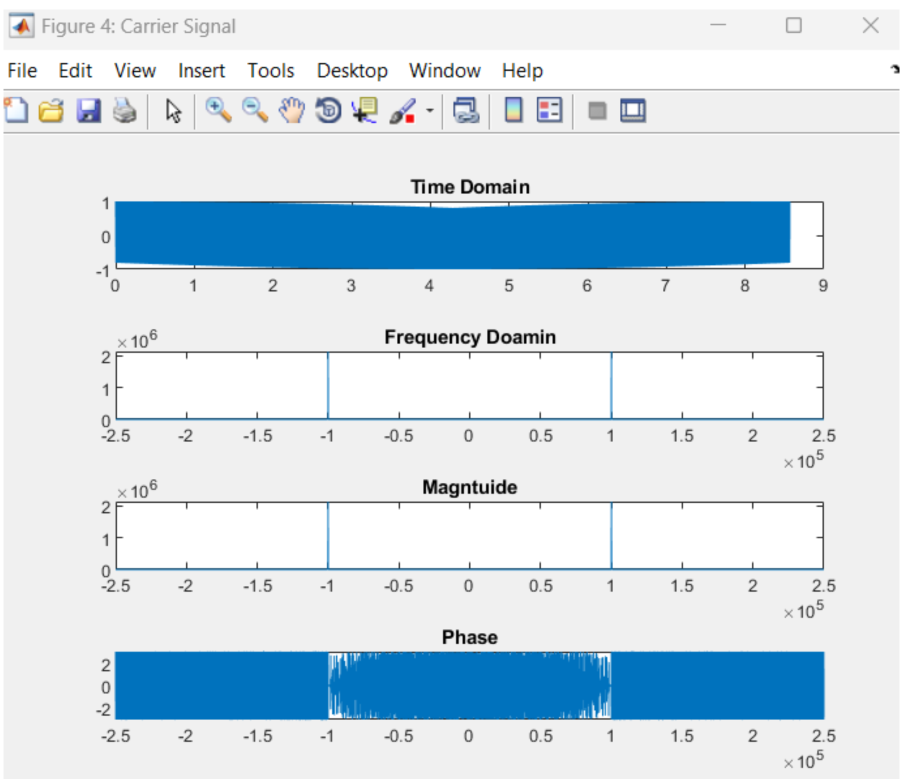


Figure 4: Carrier Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

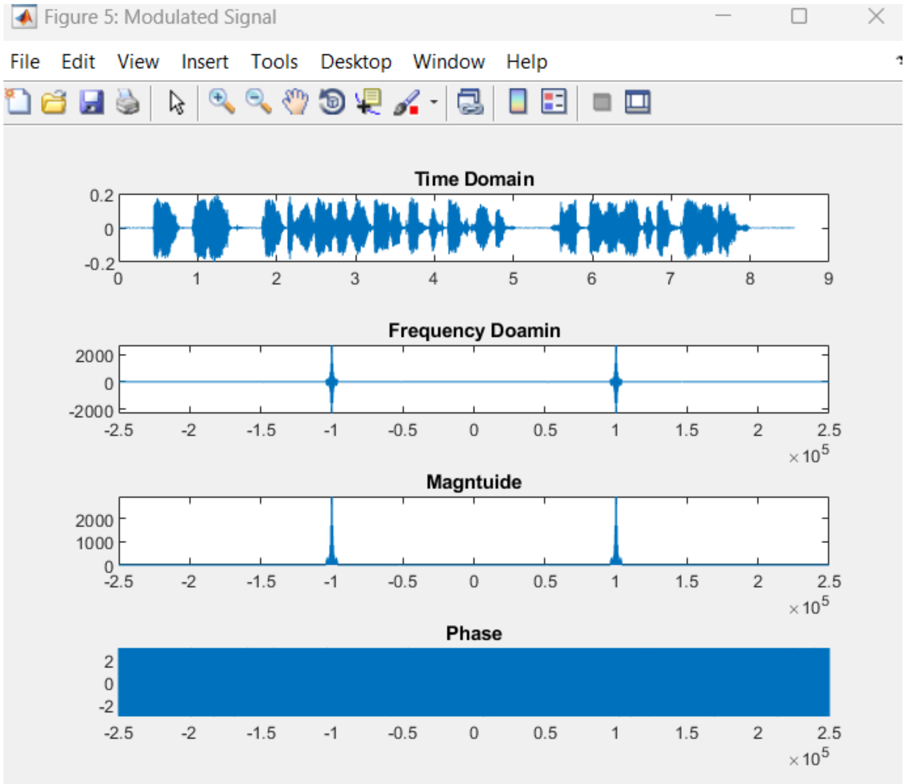


Figure 5: Modulated Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase
$\times 10^5$

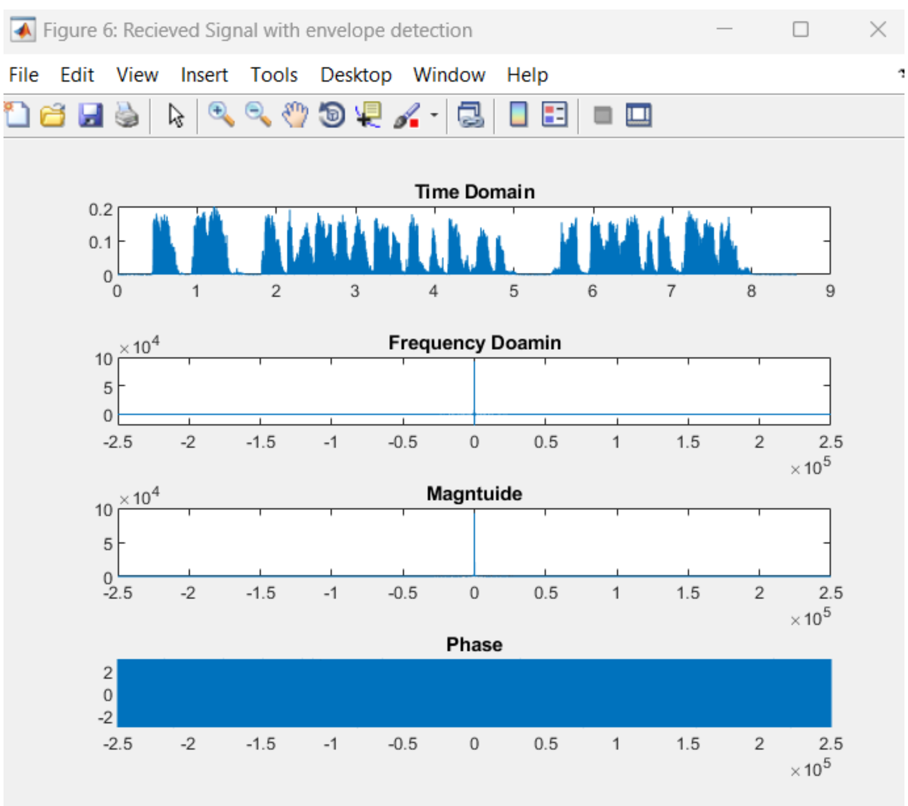


Figure 6: Recieved Signal with envelope detection
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase
$\times 10^4$
$\times 10^5$

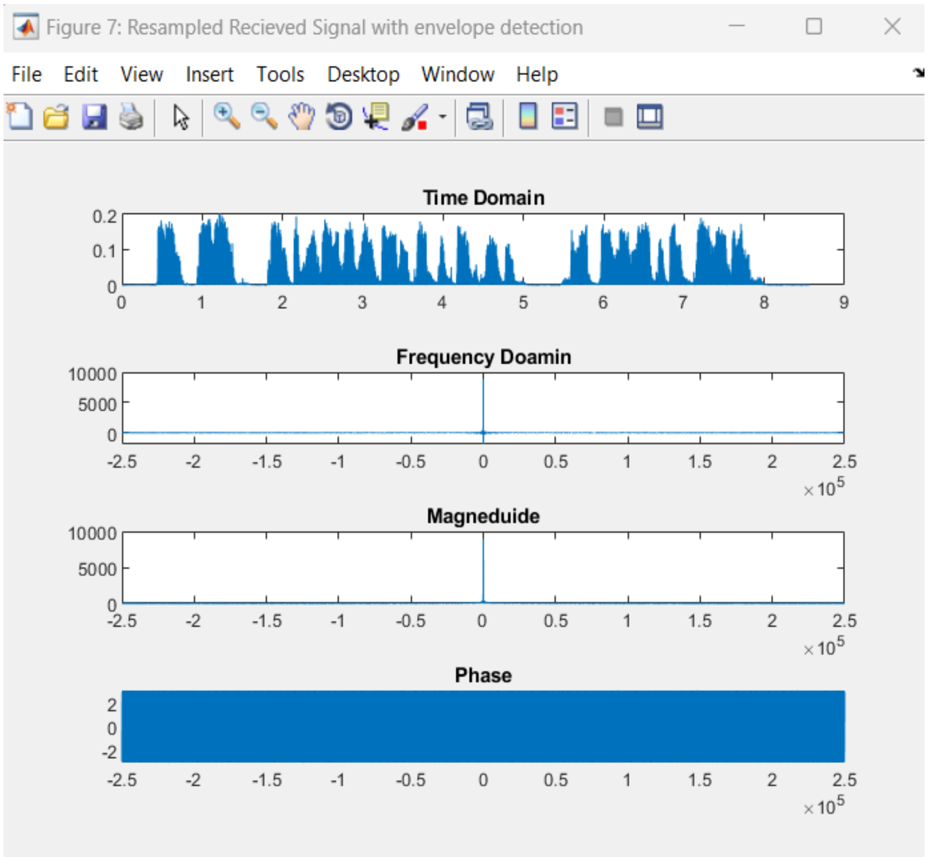


Figure 7: Resampled Recieved Signal with envelope detection
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

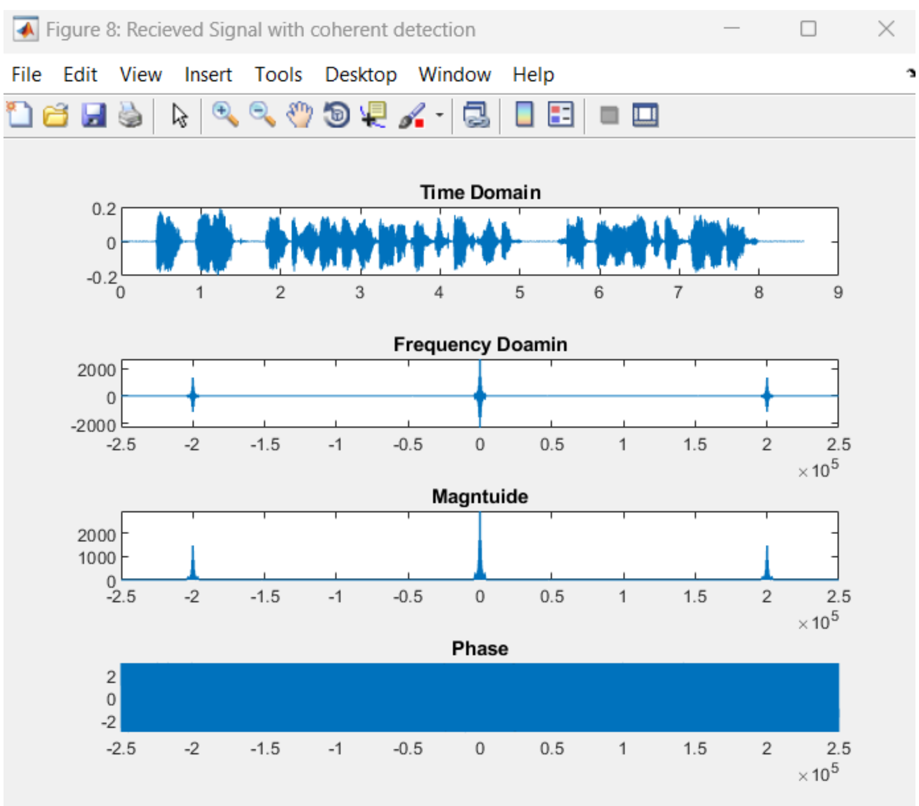


Figure 8: Recieved Signal with coherent detection
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

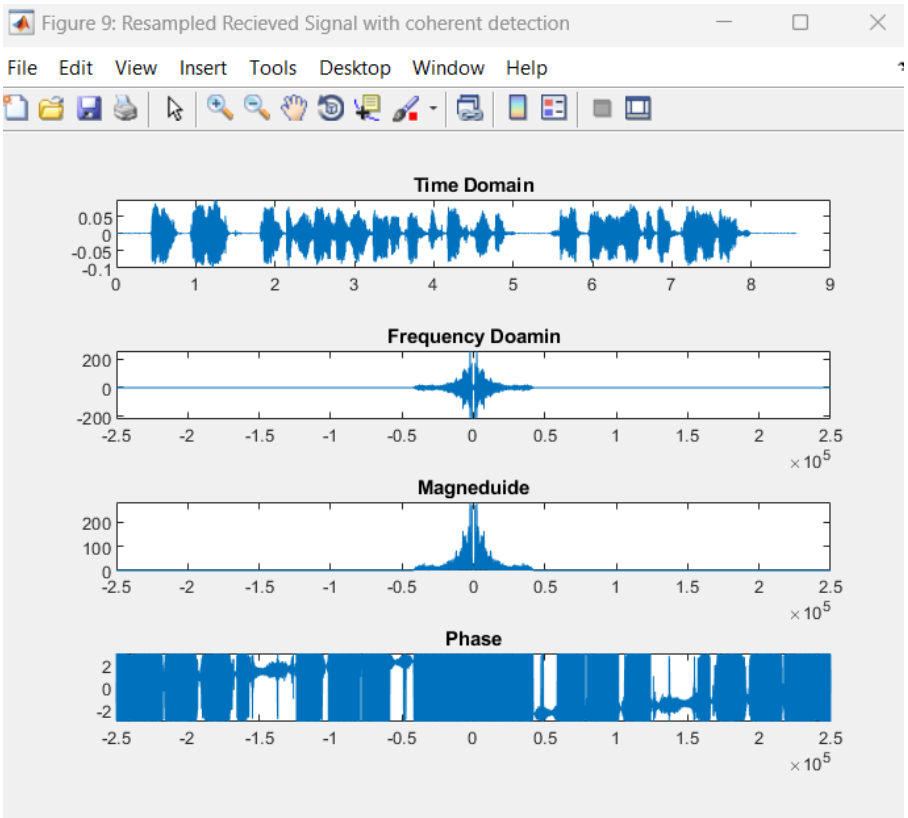


Figure 9: Resampled Recieved Signal with coherent detection
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

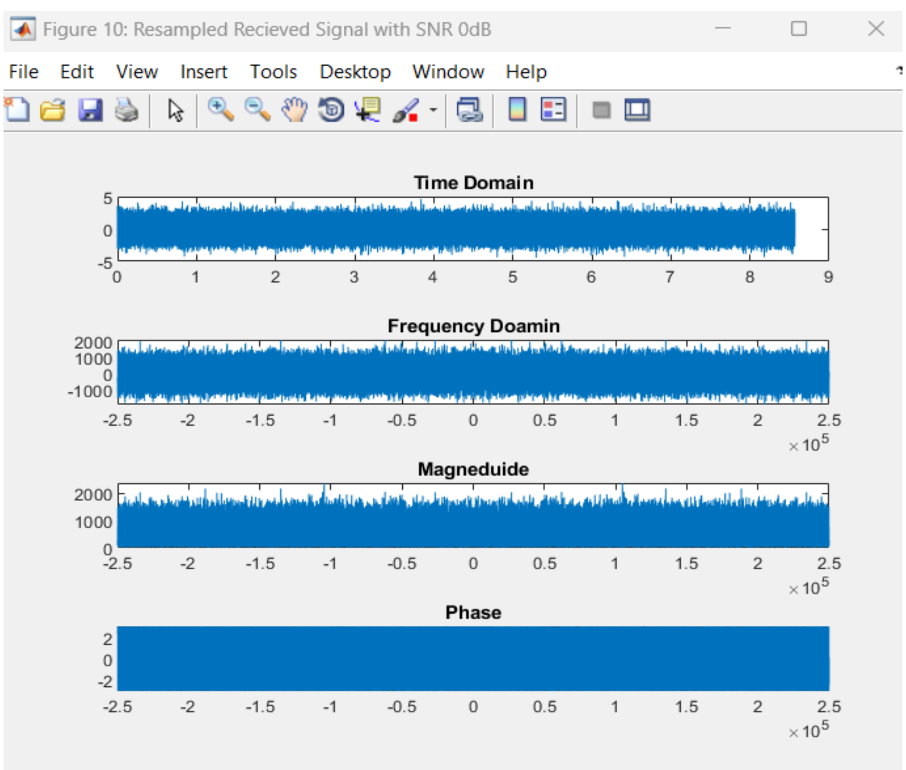


Figure 10: Resampled Recieved Signal with SNR 0dB
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

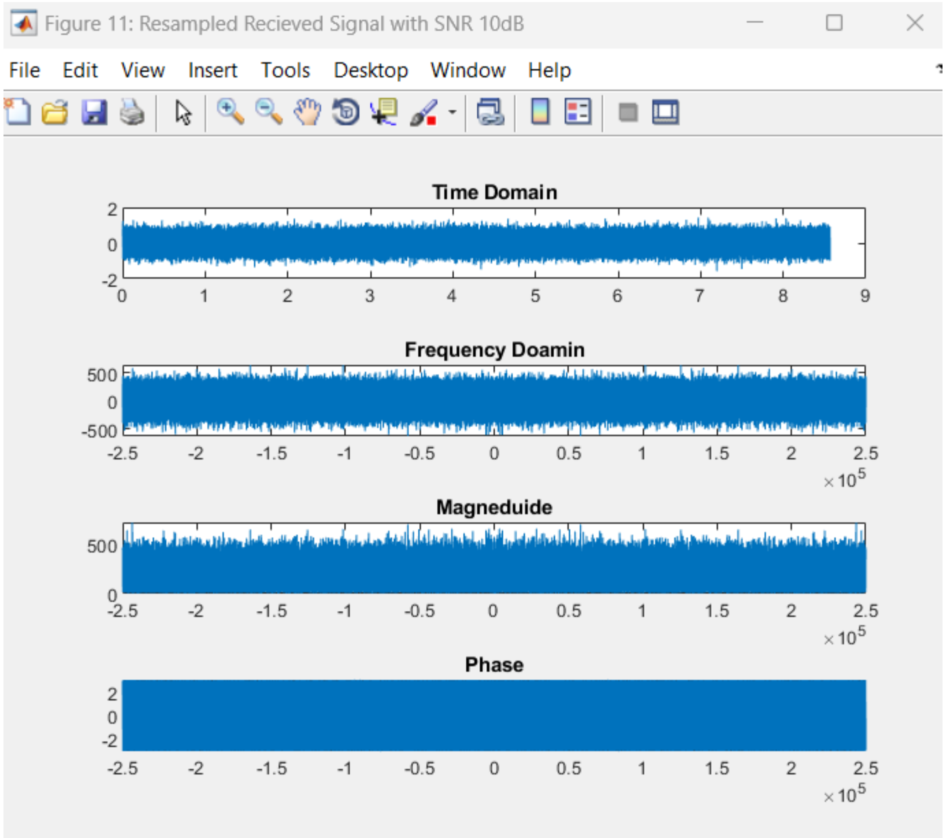


Figure 11: Resampled Recieved Signal with SNR 10dB
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

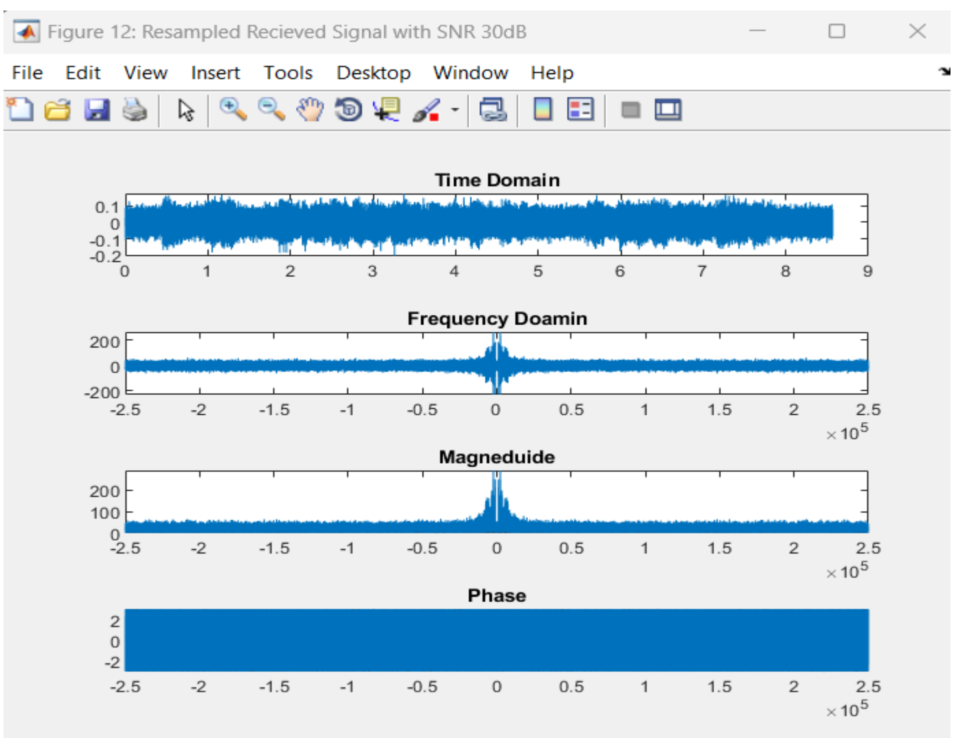


Figure 12: Resampled Recieved Signal with SNR 30dB
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

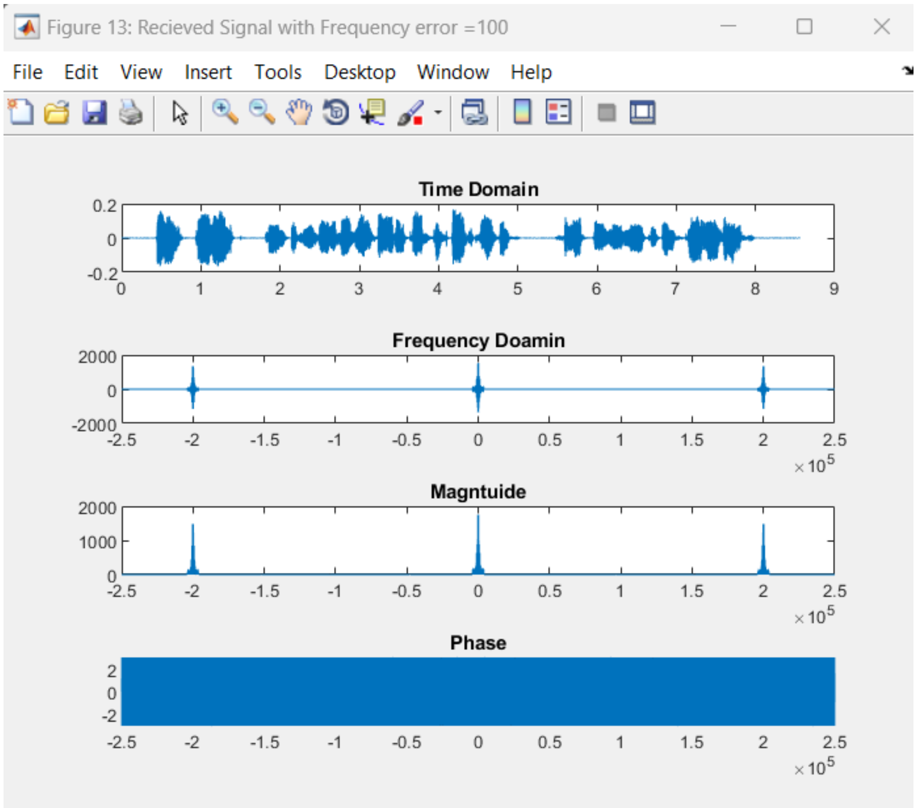


Figure 13: Recieved Signal with Frequency error =100
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

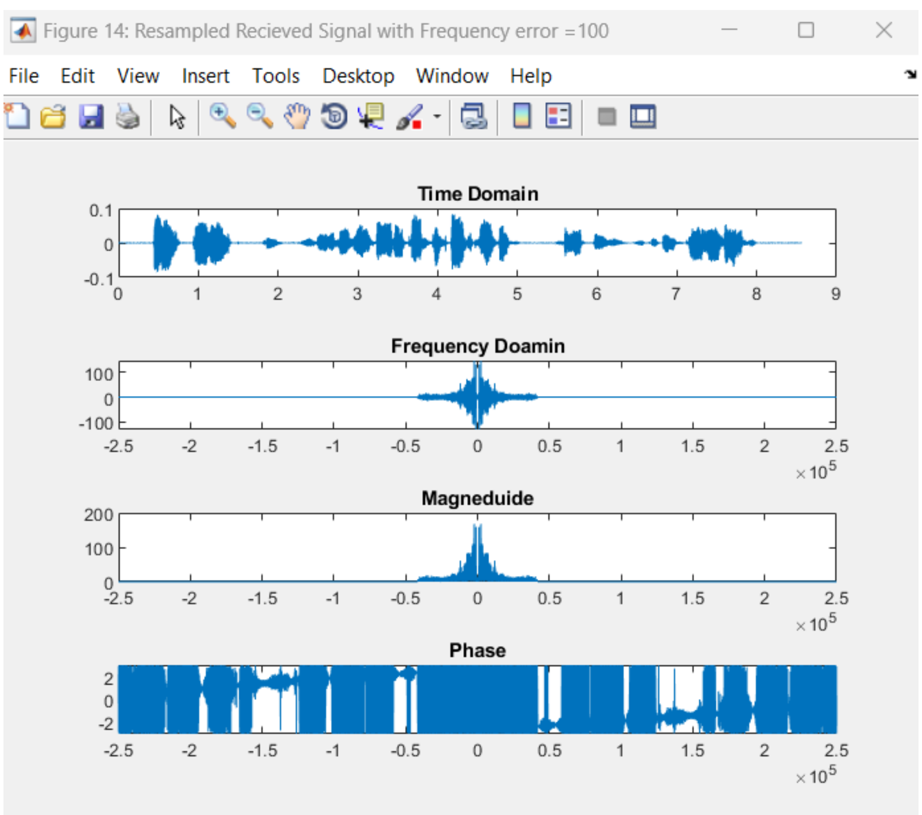


Figure 14: Resampled Recieved Signal with Frequency error =100
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

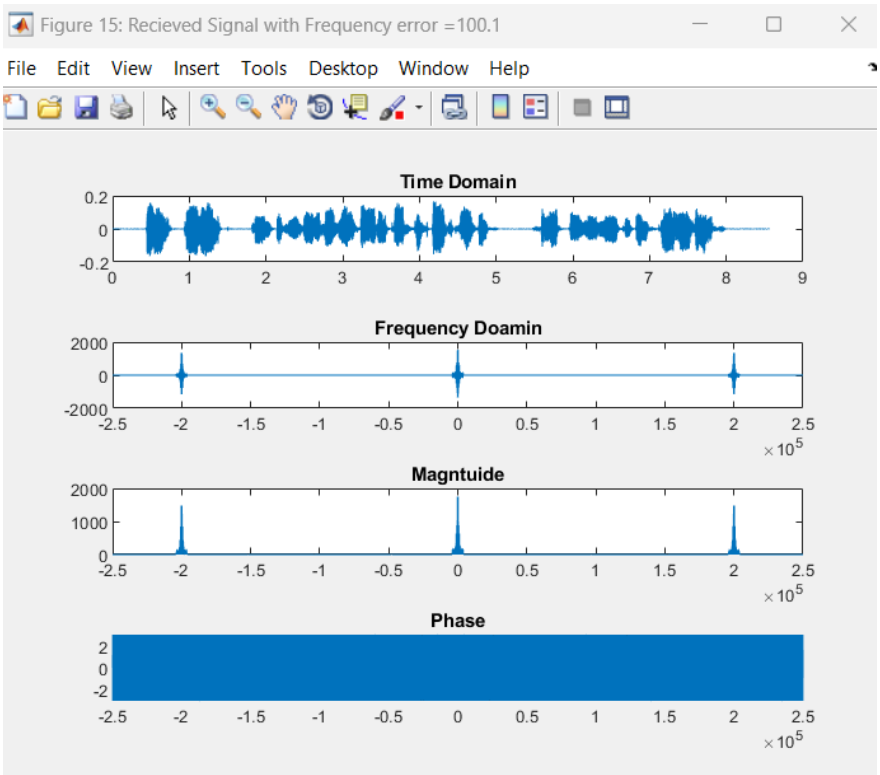


Figure 15: Recieved Signal with Frequency error =100.1
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase
×10^5

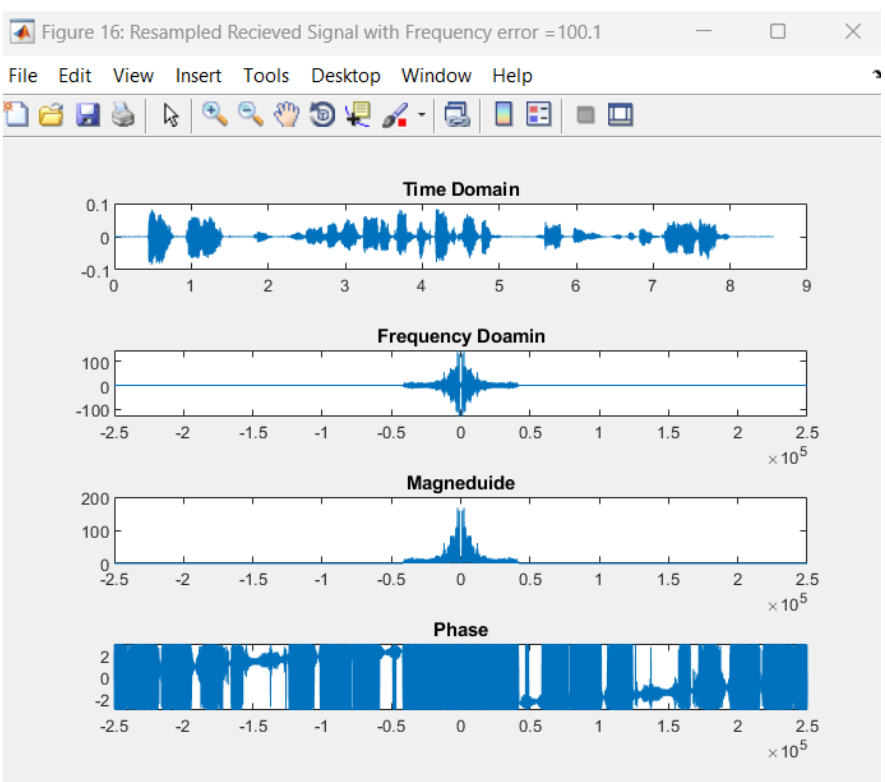


Figure 16: Resampled Recieved Signal with Frequency error =100.1
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase
×10^5

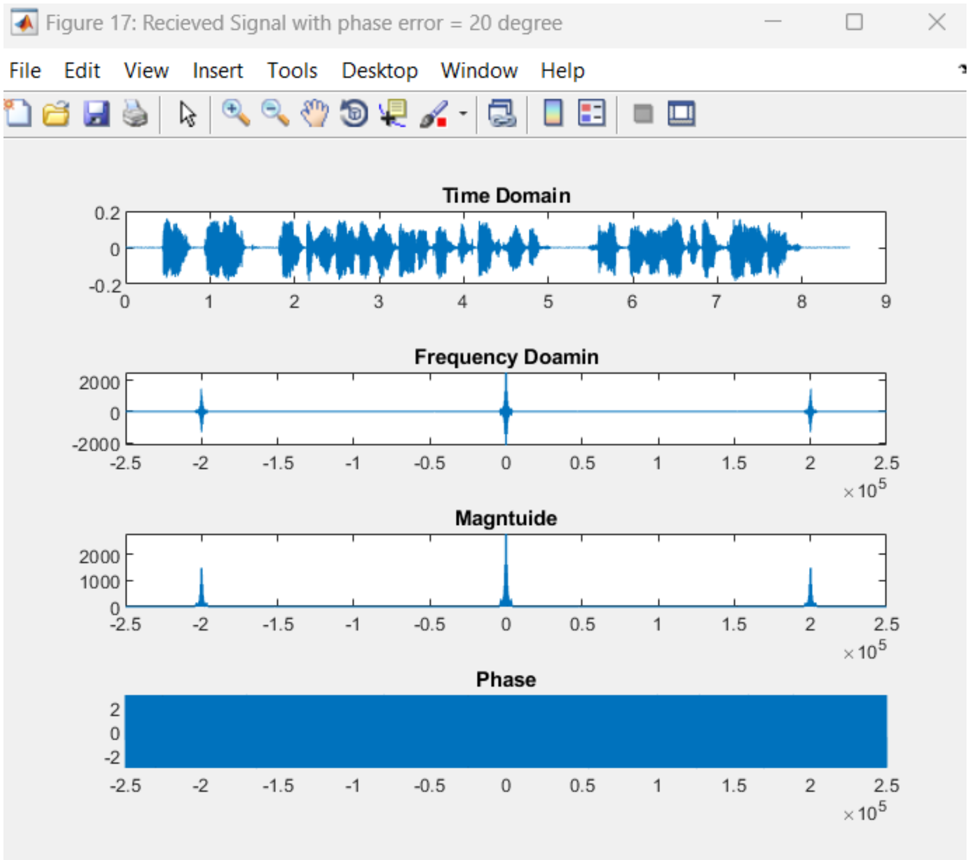


Figure 17: Recieved Signal with phase error = 20 degree
File Edit View Insert Tools Desktop Window Help
Time Domain
0.2
0
-0.2
0 1 2 3 4 5 6 7 8 9
Frequency Doamin
2000
0
-2000
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5
Magntuide
2000
1000
0
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5
Phase
2
0
-2
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5

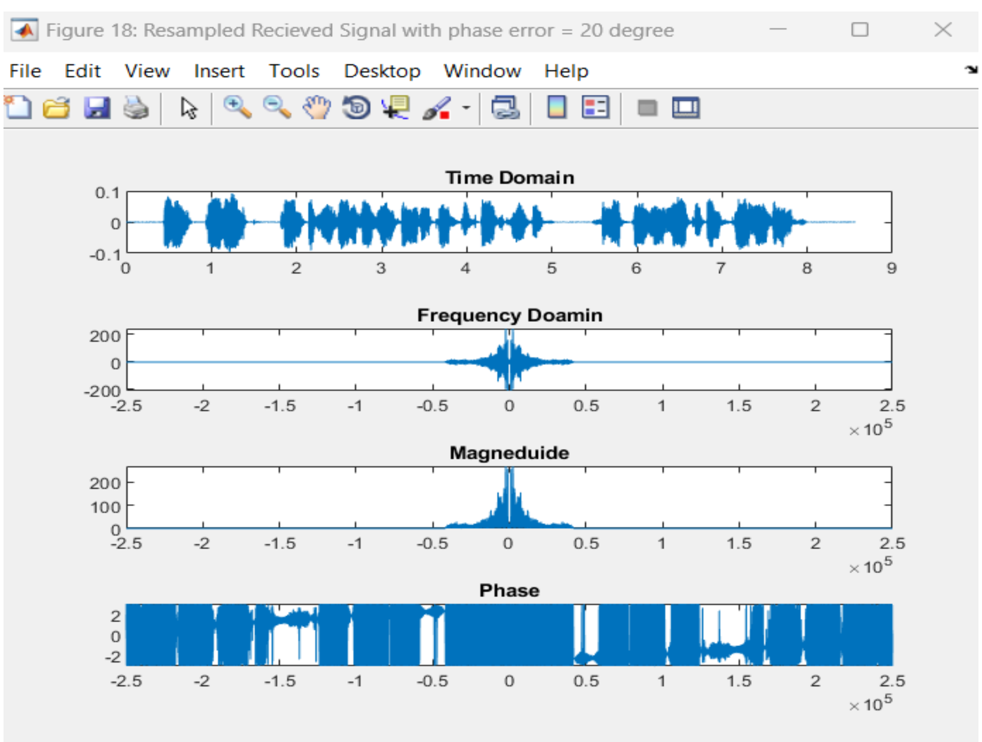


Figure 18: Resampled Recieved Signal with phase error = 20 degree
File Edit View Insert Tools Desktop Window Help
Time Domain
0.1
0
-0.1
0 1 2 3 4 5 6 7 8 9
Frequency Doamin
200
0
-200
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5
Magneduide
200
100
0
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5
Phase
2
0
-2
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
× 10^5

Experiment 1 DSBTC

<u>The code:</u>

```
clear all
close all
clc
%% Generating Signal
[Mt,fm] =audioread('eric.wav');
sound(Mt,fm);
fprintf('Original Sound is playing.. \n');
pause(length(Mt)/fm);
t=linspace(0,length(Mt)/fm,length(Mt));
figure('Name','Audio Signal');
subplot(4,1,1);
plot(t,Mt);
title('Time Domain');
Sf=fftshift(fft(Mt));
Sfvec=linspace(-fm/2,fm/2,length(Sf));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide ');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
```

```
%% filter
filter=ones(length(Mt),1);
f1=round((-4*1000+fm/2).*(length(Mt)/fm));
f2=round((4*1000+fm/2).*(length(Mt)/fm));
filter([1:f1  f2:end])=0;
Mf_Filtered=Sf;
Mf_Filtered=Mf_Filtered.*filter;
Mf_FilteredMag=abs(Mf_Filtered);
Mf_FilteredPhase=angle(Mf_Filtered);
Mt_Filtered=real(ifft(ifftshift(Mf_Filtered)));
sound(Mt_Filtered,fm);
fprintf('Filtered Sound is playing.. \n');
pause(length(Mt_Filtered)/fm);
figure('Name','Filtered Audio Signal');
subplot(4,1,1);
plot(t,Mt_Filtered);
title('Time Domain');
subplot(4,1,2);
plot(Sfvec,Mf_Filtered);
title('Frequency Doamin');
subplot(4,1,3);
plot(Sfvec,Mf_FilteredMag);
title('Magneduide');
subplot(4,1,4);
plot(Sfvec,Mf_FilteredPhase);
title('Phase');
```

```
%% resample
fc=100000;
fs=5*fc;
Mt_Fil_res=resample(Mt_Filtered,fs,fm) ;
t1=linspace(0,length(Mt_Fil_res)/fs,length(Mt_Fil_res));
figure('Name','Resampled Filtered Audio Signal');
subplot(4,1,1);
plot(t1, Mt_Fil_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mt_Fil_res));
resvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_env_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_env_resphase);
title('Phase');

%% carrier
carrier_signal = cos(2*pi*fc*t1);
Ct=transpose(carrier_signal);
figure('Name','Carrier Signal');
subplot(4,1,1);
plot(t1, Ct);
title('Time Domain');
Sf=fftshift(fft(Ct));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');

%% Modulated Signal
m=0.5;
A=max(Mt_Fil_res)/m;
St=(A+Mt_Fil_res).* Ct;
figure('Name','Modulated Signal');
subplot(4,1,1);
plot(t1, St);
title('Time Domain');
Sf=fftshift(fft(St));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
```

```
%% envolpe detection
envolpe=abs(hilbert(St))- 2* max(Mt_Fil_res);
figure('Name','Recieved Signal with envelope detection');
subplot(4,1,1);
plot(t1, envolpe);
title('Time Domain');
Sf=fftshift(fft(envolpe));
Sfvec=linspace(-fs/2,fs/2,length(Mt_Fil_res));
subplot(4,1,2);
plot(Sfvec,Sf);
title('Frequency Doamin');
Sfmag=abs(Sf);
Sfphase=angle(Sf);
subplot(4,1,3);
plot(Sfvec,Sfmag);
title('Magntuide');
subplot(4,1,4);
plot(Sfvec,Sfphase);
title('Phase');
```

```
%% resample in envolpe detection
Mr_env_res=resample(envolpe,fm,fs) ;
t2=linspace(0,length(Mr_env_res)/fm,length(Mr_env_res));
sound(Mr_env_res,fm);
fprintf('Recieved Signal with envelope detection is playing.. \n');
pause(length(Mr_env_res)/fm);
figure('Name','Resampled Recieved Signal with envelope detection');
subplot(4,1,1);
plot(t2, Mr_env_res);
title('Time Domain');
Mf_env_res=fftshift(fft(Mr_env_res));
resvec=linspace(-fs/2,fs/2,length(Mr_env_res));
subplot(4,1,2);
plot(resvec,Mf_env_res);
title('Frequency Doamin');
Mf_env_resmag=abs(Mf_env_res);
Mf_env_resphase=angle(Mf_env_res);
subplot(4,1,3);
plot(resvec,Mf_env_resmag);
title('Magneduide');
subplot(4,1,4);
plot(resvec,Mf_env_resphase);
title('Phase');
```

The outputs:

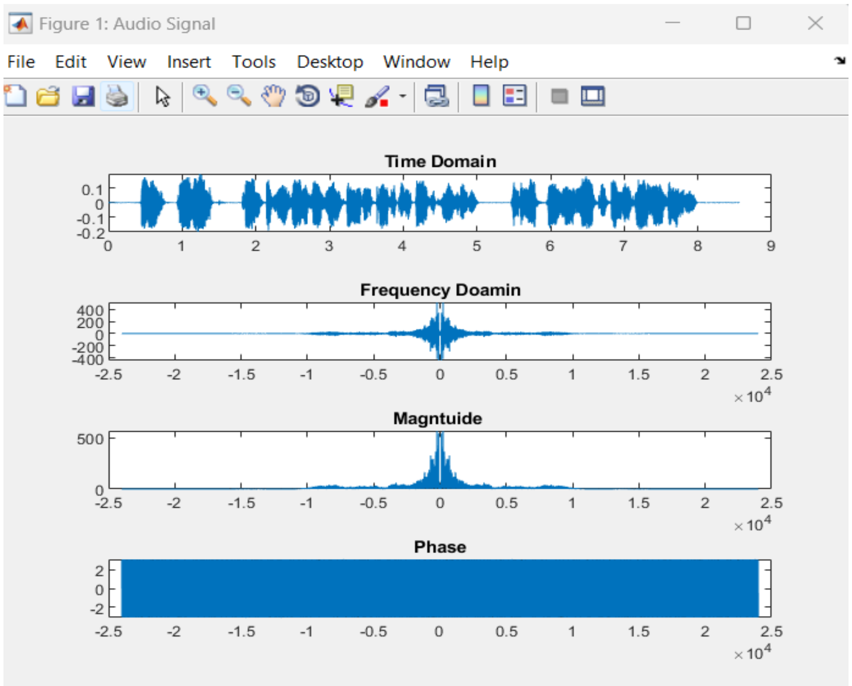


Figure 1: Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

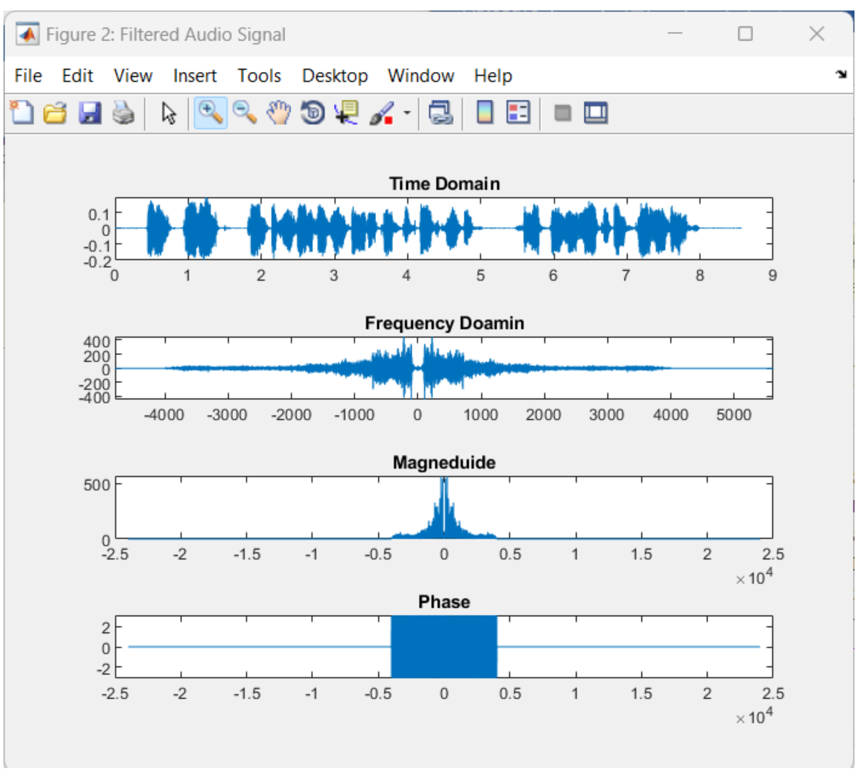


Figure 2: Filtered Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

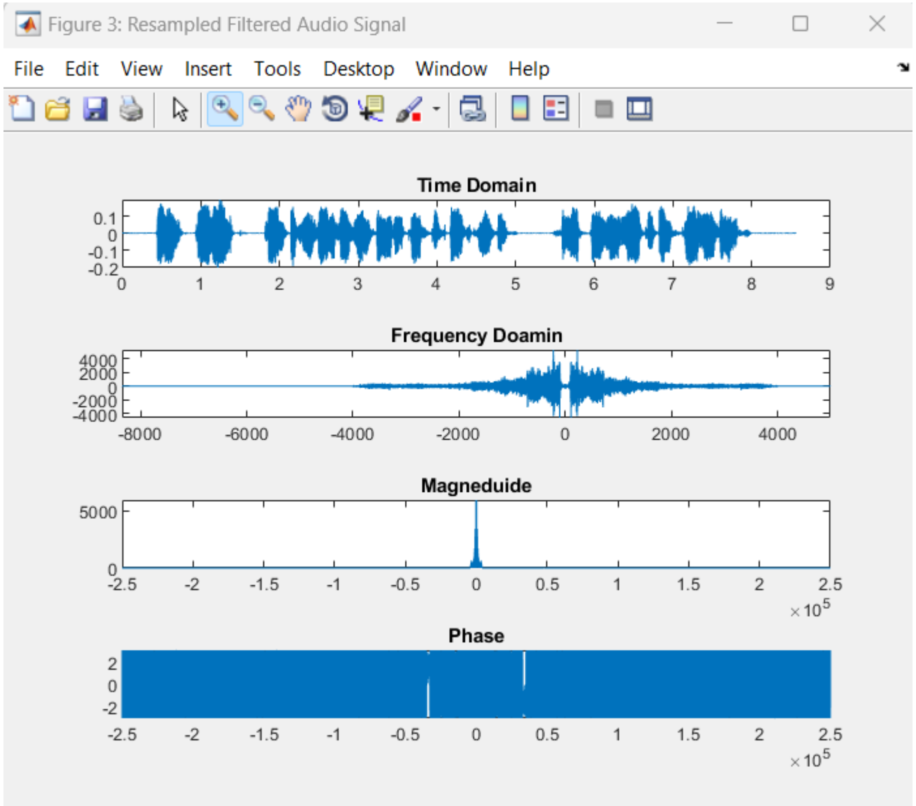


Figure 3: Resampled Filtered Audio Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magneduide
Phase

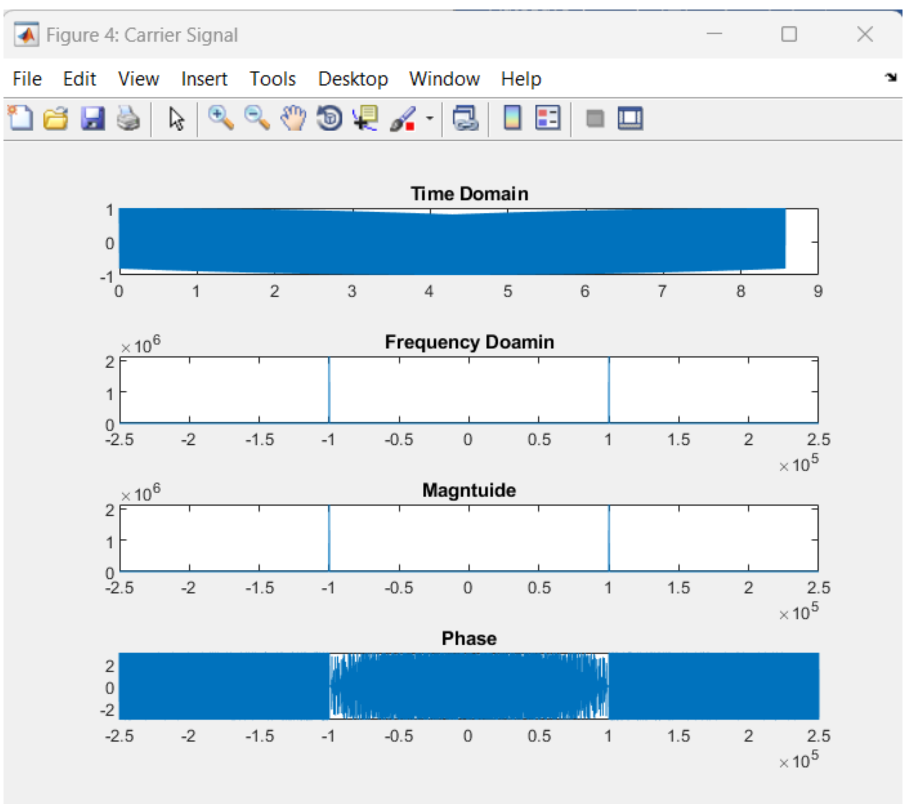


Figure 4: Carrier Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

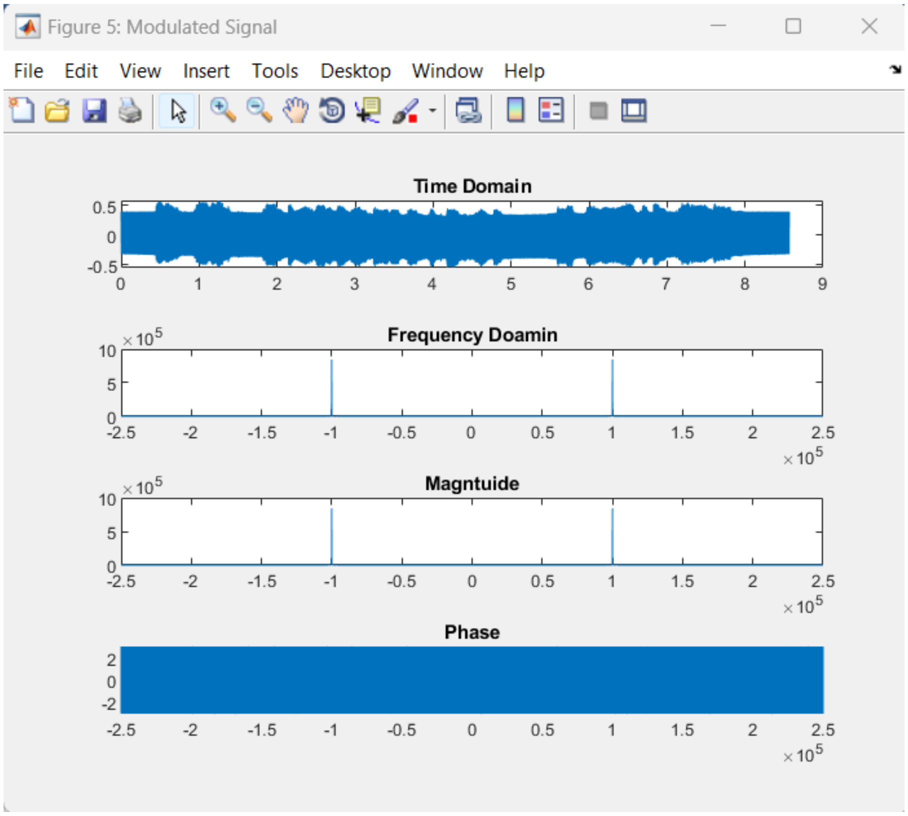


Figure 5: Modulated Signal
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase

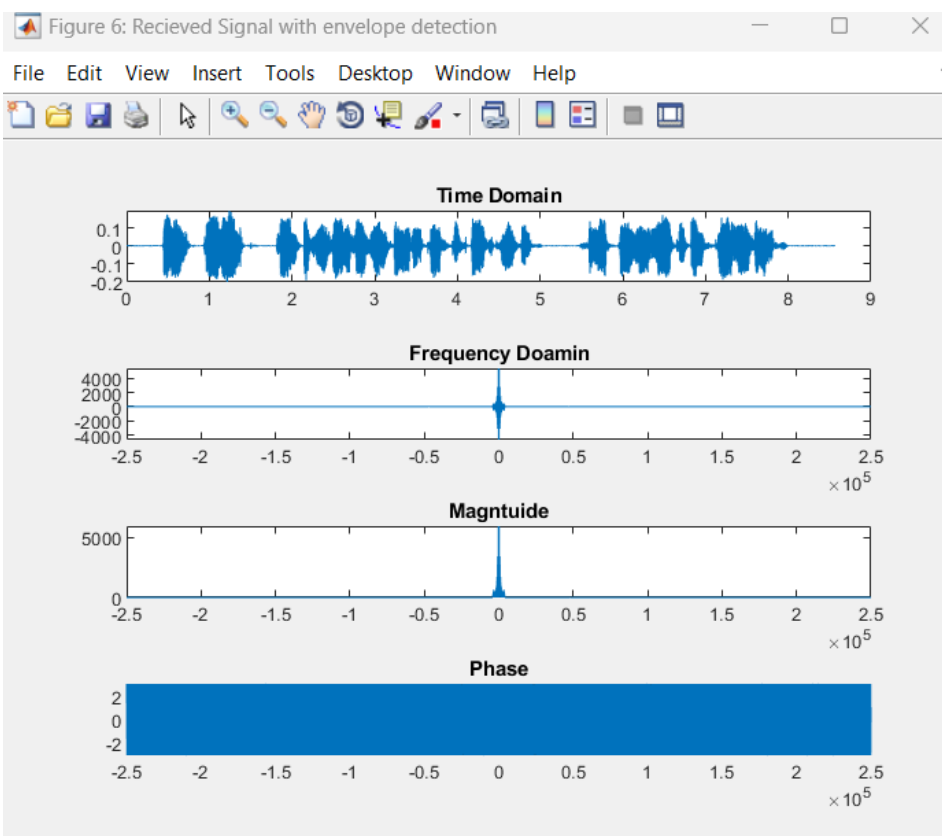
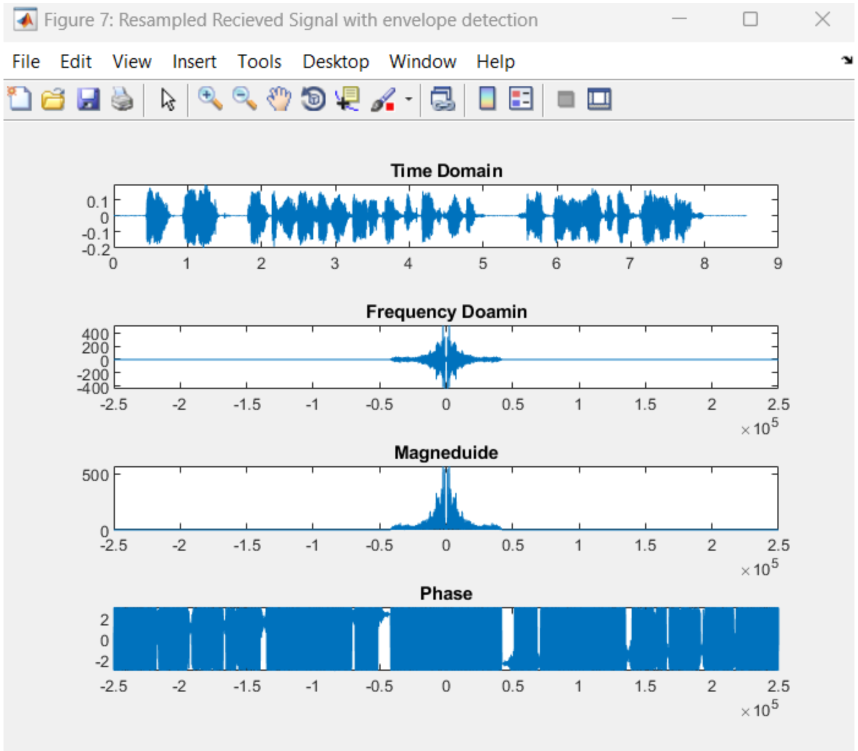


Figure 6: Recieved Signal with envelope detection
File Edit View Insert Tools Desktop Window Help
Time Domain
Frequency Doamin
Magntuide
Phase



What observation can you make of this or which type of modulation the envelope detector can be used with?

The envelope detection can only be used in DSBTC because the carrier is transmitted with the signal and when applied to DSBSC the signal appears distorted.

The higher the SNR the better the quality of the signal.

This is called beat effect, and it is a form of distortion.

Experiment 2 single sideband modulation

The bandwidth inefficiency stemming from the DSB transmission was the main reason why the single sideband (SSB) was developed. In SSB modulation, the bandwidth required for band pass transmission is equal to the bandwidth of that of the baseband. In other words, the band requirement for SSB is halved with respect to that of DSB which requires twice the baseband bandwidth. This reduction in transmission bandwidth is possible since the sideband are replicated twice over the positive and negative frequencies. However, the bandwidth reduction doesn't come entirely free.

SSB suffers from several disadvantages.

- it's difficult to design an ideal sharp edge filter, so this method is applicable only for signals have a dc null like audio signal.
- No ideal phase shifter over all band
- The generation is complex and expensive.

The code:

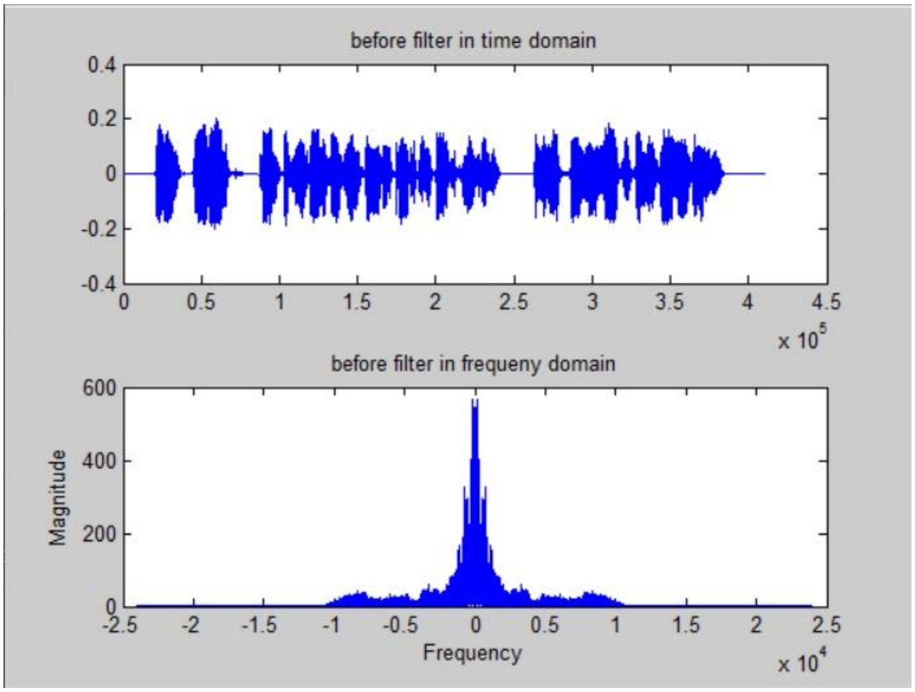
```
[signal,Fs] = audioread('E:\Downloads_2\eric.wav');
sound(signal,Fs);
pause(8)
signal = signal(:,1);
figure(1);
subplot(2,1,1);
plot(signal);
title('before filter in time domain');
faxis = linspace(-Fs/2,Fs/2,length(signal));
SIGNAL = fftshift(fft(signal));
subplot(2,1,2);
plot(faxis,abs(SIGNAL));
xlabel('Frequency');
ylabel('Magnitude');
title('before filter in frequeny domain');
%------------------Filter--------------------
n=length(signal)/Fs;
RightSide = round((Fs/2-4000)*n);
SIGNAL([1:RightSide length(signal)-RightSide+1:length(signal)]) = 0;
figure(2);
subplot(2,1,2);
plot(faxis,abs(SIGNAL));
xlabel('Frequency');
ylabel('Magnitude');
title('After filter in frequeny domain');
signal = real(ifft(ifftshift(SIGNAL)));
subplot(2,1,1);
plot(signal);
```

```
title('After filter in time domain');
sound(signal,Fs);
pause(8);
%-------------------5-DSB-SC--------------------------
CarrierFreq = 100000;
NewFs = 5 * CarrierFreq;
ResampledSignal = resample(signal,NewFs,Fs);
Time = linspace(0, length(ResampledSignal)/NewFs,
length(ResampledSignal));
ScCarrier = cos(2*pi*CarrierFreq*Time)';
DSB_SC = ResampledSignal .* ScCarrier;
fftDSB_SC = fftshift(fft(DSB_SC));
faxis = linspace(-NewFs/2,NewFs/2,length(fftDSB_SC));
figure(3);
plot(faxis,abs(fftDSB_SC));
title('DSB-SC Signal');
%-------------------SSB-SC-------------------------
n=length(DSB_SC)/NewFs;
Filter = ones(length(DSB_SC),1);
RightSide = round(n*(CarrierFreq+(NewFs/2)));
LeftSide = round(n*(-CarrierFreq+(NewFs/2)));
Filter([1:LeftSide RightSide:end]) = 0;
LSB_SC = Filter .*fftDSB_SC;
figure(4);
plot(faxis,abs(LSB_SC));
title('SSB-SC Signal');
%--------------------CohenrentDetection SSB-SC----------------
LSB_SC_T= real(ifft(ifftshift(LSB_SC)));
RecievedLSB = LSB_SC_T.*ScCarrier;
fftRecievedLSB = fftshift(fft(RecievedLSB));
n=length(fftRecievedLSB)/NewFs;
LPF = ones(length(fftRecievedLSB),1);
RightSide = round(n*(4000+(NewFs/2)));
LeftSide = round(n*(-4000+(NewFs/2)));
LPF([1:LeftSide RightSide:end]) = 0;
OutputSignal=fftRecievedLSB .*LPF;
OutputSignalT=real(ifft(ifftshift(OutputSignal)));
ResampledOutputT=resample(OutputSignalT,Fs,NewFs);
ResampledOutputF=fftshift(fft(ResampledOutputT));
Faxis=linspace(-Fs/2,Fs/2,length(ResampledOutputF));
Taxis=linspace(0,length(ResampledOutputT)/Fs,length(ResampledOutputT));
figure(5);
subplot(2,1,1);
plot(Taxis,ResampledOutputT);
title('SSB-SC Recieved signal in Time domain');
subplot(2,1,2);
plot(Faxis,abs(ResampledOutputF));
title('SSB-SC Recieved signal in Frequency domain');
sound(ResampledOutputT,Fs)
pause(10);
%--------------------4th order Butterworth filter---------------\
[b,a]=butter(4,[CarrierFreq-4000 CarrierFreq]/(NewFs/2),'Bandpass');
LSB_ButterT=filter(b,a,DSB_SC);
LSB_ButterF=fftshift(fft(LSB_ButterT));
```

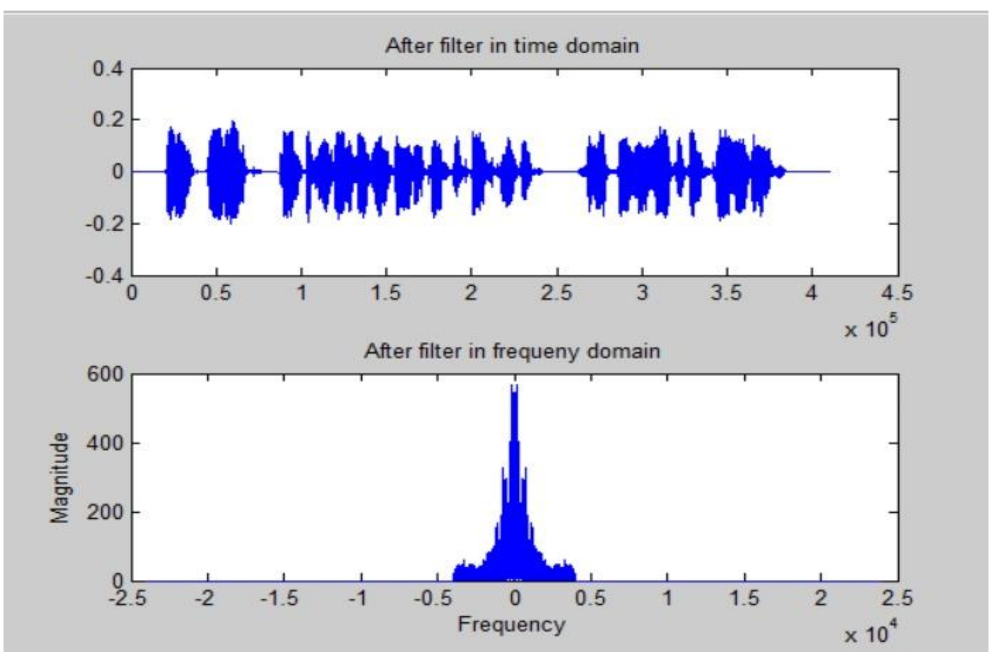
```
figure(6);
plot(faxis,abs(LSB_ButterF));
title('SSB-SC (Butterworth)');
%--------------------CohenrentDetection SSB-SC----------------
RecievedLSB = LSB_ButterT.*ScCarrier;
ResampledRecievedLSB=resample(RecievedLSB,Fs,NewFs);
[b,a]=butter(4,(4000 ./(2.*Fs)),'low');
OutputT=filter(b,a,ResampledRecievedLSB);
OutputF=fftshift(fft(ResampledRecievedLSB));
Faxis=linspace(-Fs/2,Fs/2,length(OutputF));
Taxis=linspace(0,length(OutputT)/Fs,length(OutputT));
figure(7);
subplot(2,1,2);
plot(Faxis,abs(OutputF));
title('SSB recieved signal in Frequency Domain(ButterWorth)');
subplot(2,1,1);
plot(Taxis,OutputT);
title('SSB recieved signal in Time Domain(ButterWorth)');
sound(OutputT,Fs) ;
pause(10);
%-----------------ssd_snr=0------------
lsb_Noise=awgn(LSB_SC_T,0);
recievedLsb_NoiseT=lsb_Noise.*ScCarrier;
recievedLsb_NoiseF=fftshift(fft(recievedLsb_NoiseT));
n=length(recievedLsb_NoiseF)/NewFs;
LPF = ones(length(recievedLsb_NoiseF),1);
RightSide = round(n*(4000+(NewFs/2)));
LeftSide = round(n*(-4000+(NewFs/2)));
LPF([1:LeftSide RightSide:end]) = 0;
OutputSignal_noise=recievedLsb_NoiseF .*LPF;
OutputSignal_noiseT=real(ifft(ifftshift(OutputSignal_noise)));
ResampledOutputT=resample(OutputSignal_noiseT,Fs,NewFs);
ResampledOutputF=fftshift(fft(ResampledOutputT));
Faxis=linspace(-Fs/2,Fs/2,length(ResampledOutputF));
Taxis=linspace(0,length(ResampledOutputT)/Fs,length(ResampledOutputT));
figure(8);
subplot(2,1,1);
plot(Taxis,ResampledOutputT);
title('SSB-SC Recieved signal with SNR=0 in Time domain');
subplot(2,1,2);
plot(Faxis,abs(ResampledOutputF));
title('SSB-SC Recieved signal with SNR=0 in Freq domain');
sound(ResampledOutputT,Fs)
pause(10);
% -----------------ssd_snr=10------------------------
lsb_Noise=awgn(LSB_SC_T,10);
recievedLsb_NoiseT=lsb_Noise.*ScCarrier;
recievedLsb_NoiseF=fftshift(fft(recievedLsb_NoiseT));
n=length(recievedLsb_NoiseF)/NewFs;
LPF = ones(length(recievedLsb_NoiseF),1);
RightSide = round(n*(4000+(NewFs/2)));
LeftSide = round(n*(-4000+(NewFs/2)));
LPF([1:LeftSide RightSide:end]) = 0;
OutputSignal_noise=recievedLsb_NoiseF .*LPF;
```

```
OutputSignal_noiseT=real(ifft(ifftshift(OutputSignal_noise)));
ResampledOutputT=resample(OutputSignal_noiseT,Fs,NewFs);
ResampledOutputF=fftshift(fft(ResampledOutputT));
Faxis=linspace(-Fs/2,Fs/2,length(ResampledOutputF));
Taxis=linspace(0,length(ResampledOutputT)/Fs,length(ResampledOutputT));
figure(9);
subplot(2,1,1);
plot(Taxis,ResampledOutputT);
title('SSB-SC Recieved signal with SNR=10 in Time domain');
subplot(2,1,2);
plot(Faxis,abs(ResampledOutputF));
title('SSB-SC Recieved signal with SNR=10 in Freq domain');
sound(ResampledOutputT,Fs)
pause(10);
%-----------------ssd_snr=30------------------------
lsb_Noise=awgn(LSB_SC_T,30);
recievedLsb_NoiseT=lsb_Noise.*ScCarrier;
recievedLsb_NoiseF=fftshift(fft(recievedLsb_NoiseT));
n=length(recievedLsb_NoiseF)/NewFs;
LPF = ones(length(recievedLsb_NoiseF),1);
RightSide = round(n*(4000+(NewFs/2)));
LeftSide = round(n*(-4000+(NewFs/2)));
LPF([1:LeftSide RightSide:end]) = 0;
OutputSignal_noise=recievedLsb_NoiseF .*LPF;
OutputSignal_noiseT=real(ifft(ifftshift(OutputSignal_noise)));
ResampledOutputT=resample(OutputSignal_noiseT,Fs,NewFs);
ResampledOutputF=fftshift(fft(ResampledOutputT));
Faxis=linspace(-Fs/2,Fs/2,length(ResampledOutputF));
Taxis=linspace(0,length(ResampledOutputT)/Fs,length(ResampledOutputT));
figure(10);
subplot(2,1,1);
plot(Taxis,ResampledOutputT);
title('SSB-SC Recieved signal with SNR=30 in Time domain');
subplot(2,1,2);
plot(Faxis,abs(ResampledOutputF));
title('SSB-SC Recieved signal with SNR=30 in Freq domain');
sound(ResampledOutputT,Fs)
pause(10);
%------------------------ SSB-TC---------------------------
Amplitude = max(ResampledSignal)./0.5;
transmitted_ssb=((Amplitude.*ScCarrier)+DSB_SC);
ffttransmitted_ssb=fftshift(fft(transmitted_ssb));
envelope=abs(hilbert(transmitted_ssb));
new=resample(envelope,Fs,NewFs);
Taxis=linspace(0,length(new)/Fs,length(new));
figure(11);
plot(Taxis,new);
title('SSB-TC Enelope Detector signal ');
sound(new,Fs);
```

1. Audio signal Before Filtering in Time and Frequency Domain

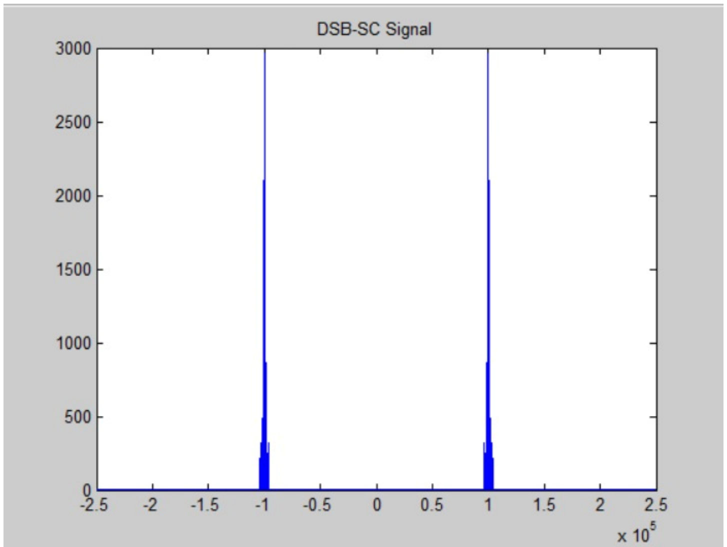


2. Audio signal After Filtering in Time and Frequency Domain To remove frequencies greater than 4KHz, we use Low Pass Filter(LPF) with cutoff frequency 4KHz So you can notice clearly from the graph that only frequency components below 4k passed.



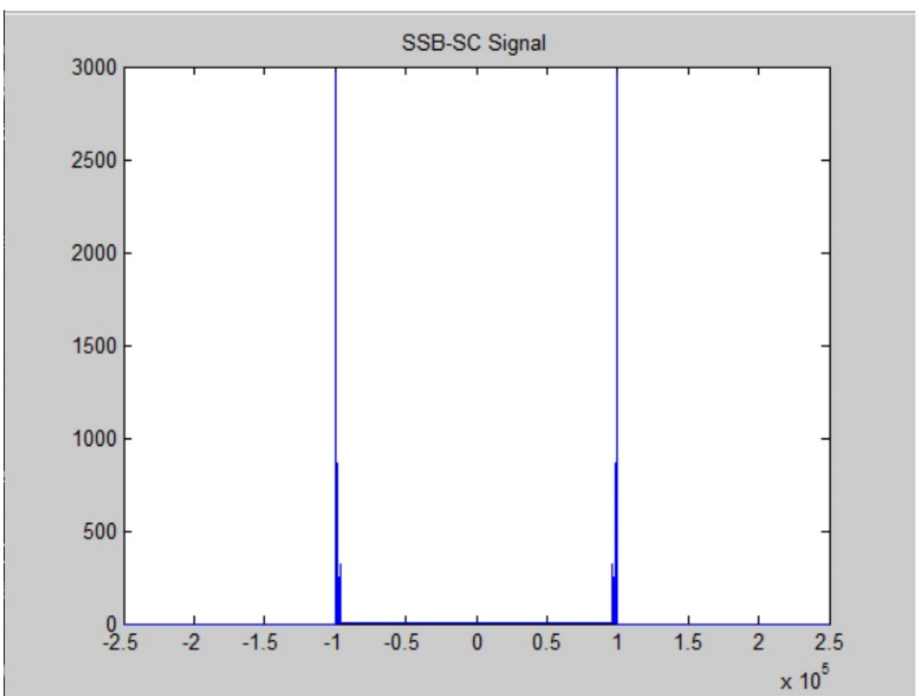
3. DSB-SC

We multiply signal by the carrier so we can notice that Signal has been shifted to +100KHz and -100KHz.



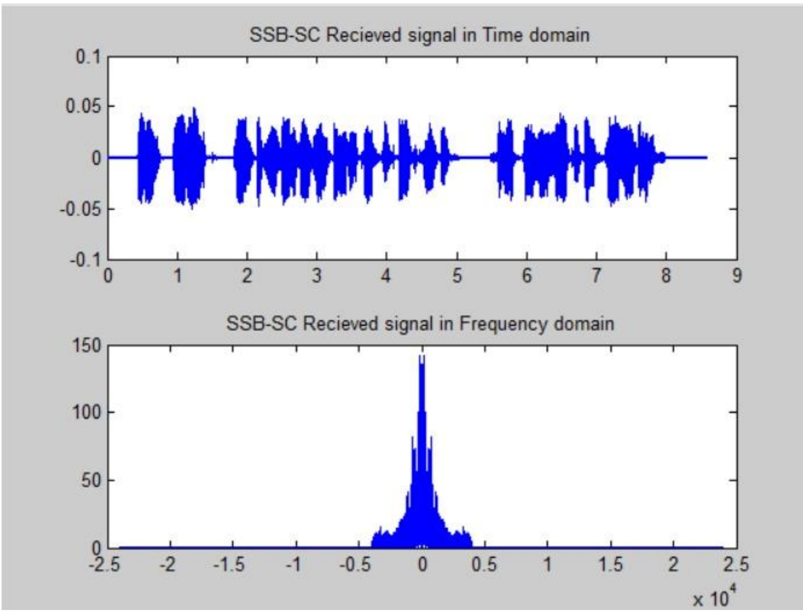
4. SSB-SC

Because ideal (band pass filter) BPF filter, we can notice that the USB side has been filtered out.



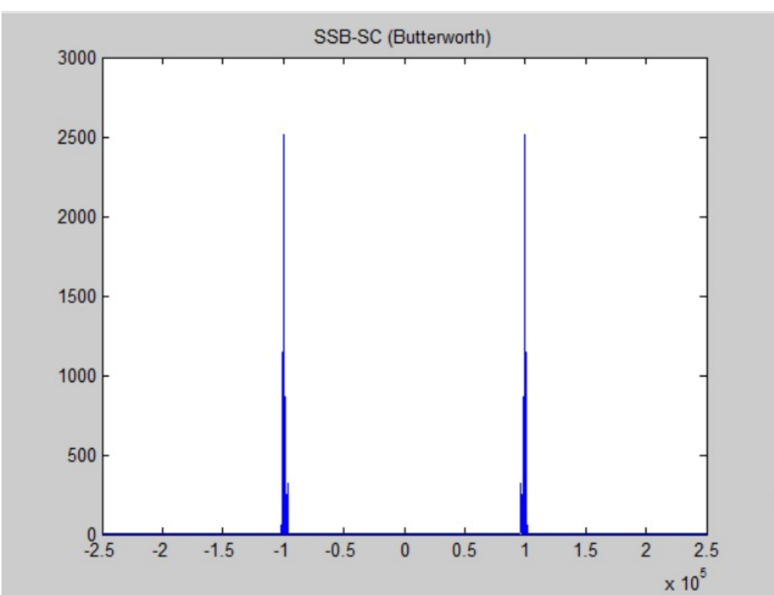
5. SSB-SC Coherent detection with no noise

We multiply the received Signal with carrier so there will be high Frequency at 2fc and frequency at the center (0,0), so we put a LPF filter to remove both noise and high frequency components.

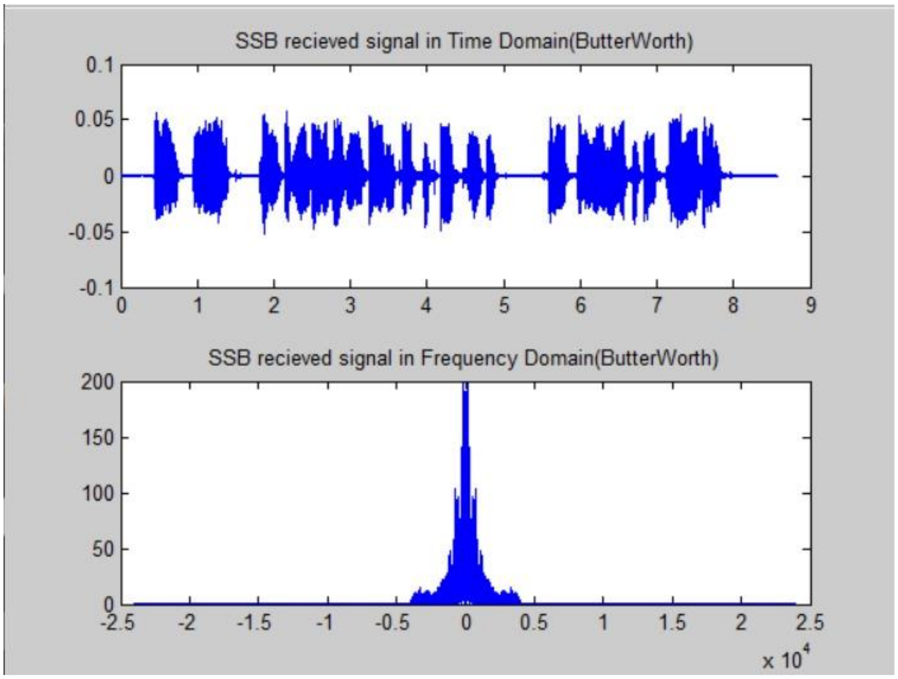


6. SSB-SC using practical filter (Butterworth)

The Butterworth filter is practical filter has a roll-off of some amount so we can see that not all of USB has filtered out.

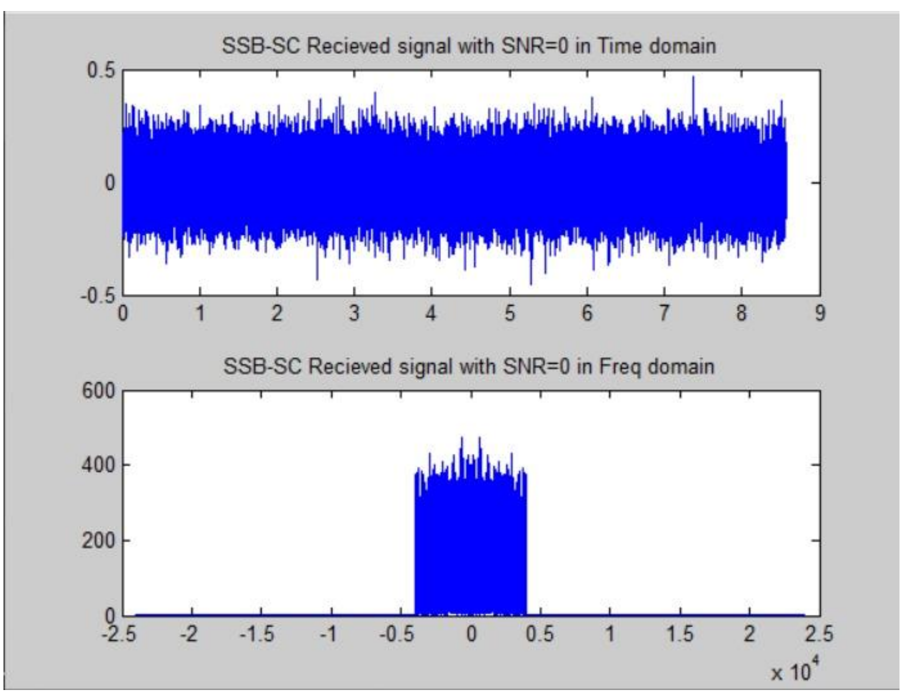


7. SSB-SC Received using practical filter (Butterworth).



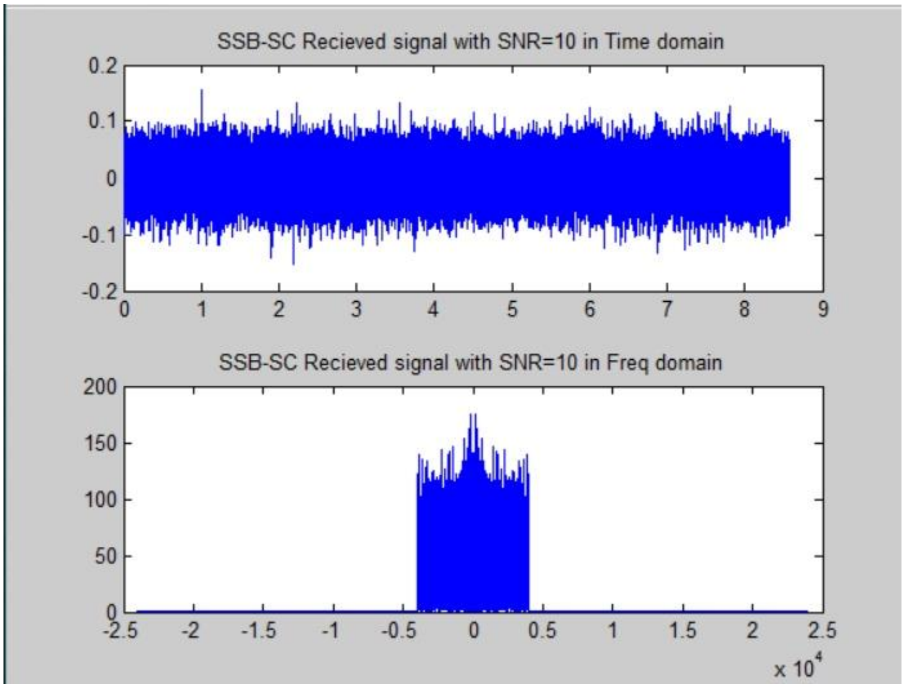
8. SSB-SC Received signal SNR = 0

As the SNR equal 0 signal is so noisy, and the power of noise is high.

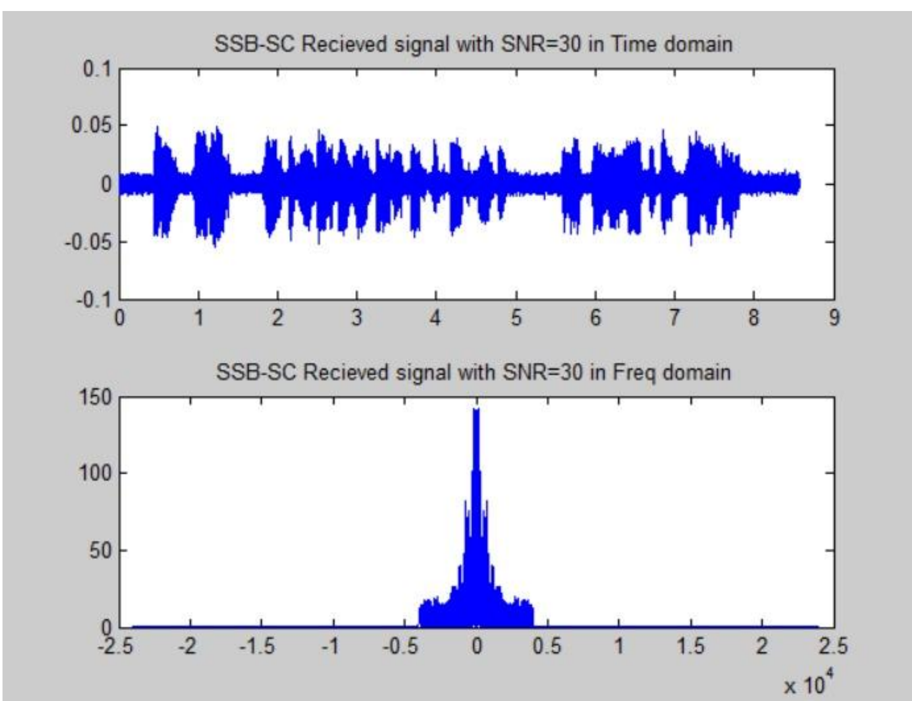


9. SSB-SC Received signal SNR =10

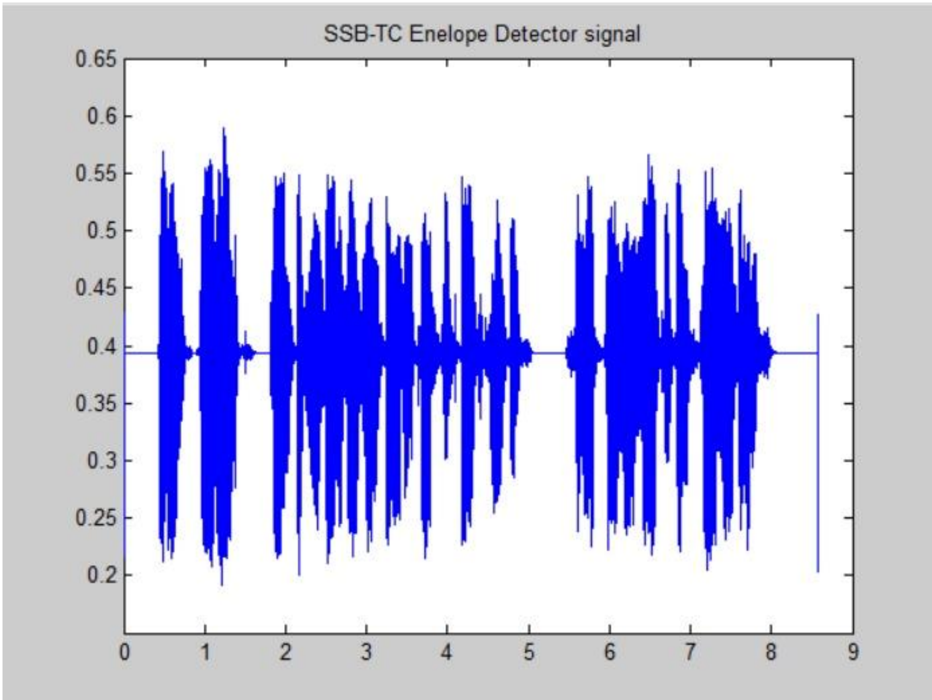
Less noise than SNR = 0 but still noisy and signal begin to appear.



10. SSB-SC Received signal SNR = 30 Noise get lower and signal appear



11. SSB-TC


SSB-TC Enelope Detector signal
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
1
2
3
4
5
6
7
8
9

Experiment 3: FM

Frequency modulation (FM) is a modulation type in which the instantaneous frequency of the carrier is changed according to the message amplitude. The motive behind the frequency modulation was to develop a scheme with inherent ability to combat noise. The noise, being usually modeled as additive, has a negative effect on the amplitude by introducing unavoidable random variations which are superimposed on the desired signal. Unlike the amplitude, frequency has a latent immunity against noise. Since it resides “away” from the amplitude, any changes in the amplitude would be completely irrelevant to the frequency. In other words, there is no direct correlation between the variation in amplitude and frequency, thus making FM a better candidate over AM with respect to noise immunity. However, what FM gains in noise immunity lacks in bandwidth efficiency. Since FM usually occupies larger bandwidth, AM is considered more bandwidth wise.

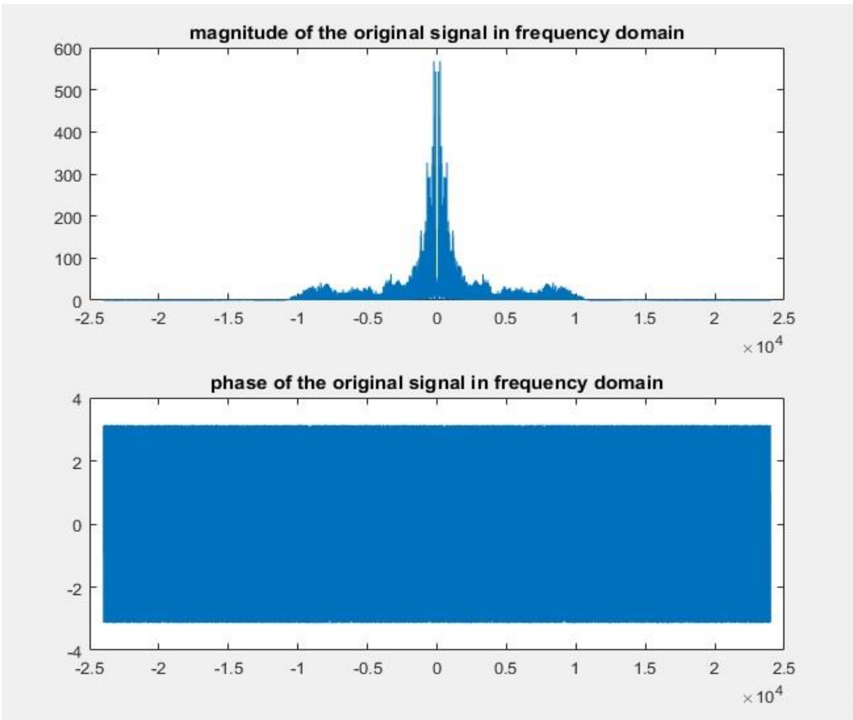
The code:

```
%%%%%%%insert signal%%%%%%%
clc;
[signal,fm]=audioread('eric.wav'); %read the audio file
Ts=1/fm;
signal_In_freq=fftshift(fft(signal)); %getting fourier
transform and swaping to center in zero
f_lineplot=linspace(-fm/2,fm/2,length(signal)); %frequency values in
x-axis
%%%%%%% plot signal (mag & angle)in Freq domain %%%%%%%
phase=angle(signal_In_freq*(180/pi));
magnitude=abs(signal_In_freq);
subplot(2,1,1)
plot(f_lineplot,magnitude)
title('magnitude of the original signal in frequency domain')
subplot(2,1,2)
plot(f_lineplot,phase)
title('phase of the original signal in frequency domain')
%%%%% Filter signal up 4000HZ %%%%%%
y=ones(length(signal),1);
n=length(signal)/fm;
RightSide = round((fm/2-4000)*n);
y([1:RightSide length(signal)-RightSide+1:length(signal)]) = 0;
filter_ferq=signal_In_freq.*y ;
filtered_signal_time=real(ifft(ifftshift(filter_ferq))); %%%m(t)
figure;
subplot(2,1,1)
plot(f_lineplot,abs(filter_ferq));
title('magnitude of the signal in frequency domain (After
```

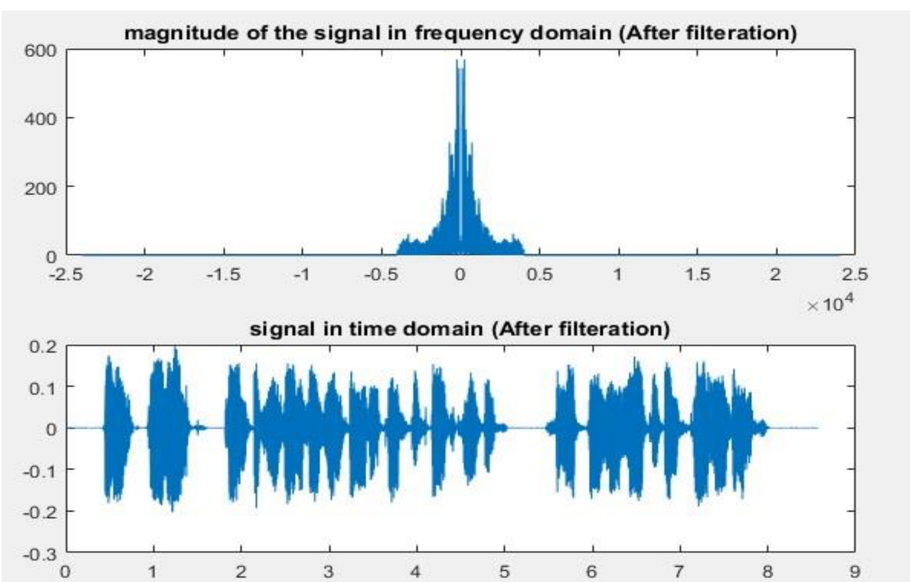
```
filteration)');
t=linspace(0,n,length(signal));
subplot(2,1,2)
plot(t,filtered_signal_time);
title('signal in time domain (After filteration)');
sound(filtered_signal_time,fm);
%%%%%%% NBFM %%%%%%%%
fc=100000;
fs=5*fc;
resample_signal_time=resample(filtered_signal_time,fs,fm);
%%%resampling by time rate=fs/fm
%%assume:
Ac=10;
kf=.05;
len=length(resample_signal_time);
integrat_mt=cumsum(resample_signal_time);
t=linspace(0,len/fs,len);
s_t= Ac*cos(2*pi*fc*t+2*pi*kf*integrat_mt'); %%%%narrow band
equation
s_f=fftshift(fft(s_t));
F_line= linspace(-fs/2,fs/2,length(s_f));
phase=angle(s_f*(180/pi));
magnitude=abs(s_f);
figure;
subplot(3,1,1)
plot(F_line,magnitude)
title('magnitude of the signal in frequency domain (NBFM)')
subplot(3,1,2)
plot(F_line,phase)
title('phase of the signal in frequency domain (NBFM)')
subplot(3,1,3)
plot(t,s_t)
title('the signal in time domain (NBFM)')
%%%%%%% demodulation %%%%%%%
diff_s_t=diff(s_t);
envelop_det=abs(hilbert(diff_s_t)); %%%%get one side of
signal
ev_time=(fftshift(fft(envelop_det)));
n = length(ev_time)/fs;
dc_f = (fs/2) - 1;
ev_time(((dc_f*n):end-(dc_f*n))) = 0; %%%dc block
f = linspace(-fs/2,fs/2,length(ev_time));
ev_time_real=real(ifft(ifftshift(ev_time)));
resample_f=resample(ev_time_real,fm,fs) %resampling with the old
frequency
t=linspace(0,length(resample_f)/fm,length(resample_f));
figure
subplot(2,1,1)
plot(f,abs(ev_time))
title('signal Magnitude (before resampling) in Frequency Domain')
subplot(2,1,2)
plot(t,resample_f)
title('signal (after resampling) in time Domain')
sound(resample_f,fm)
```

- # Figures:

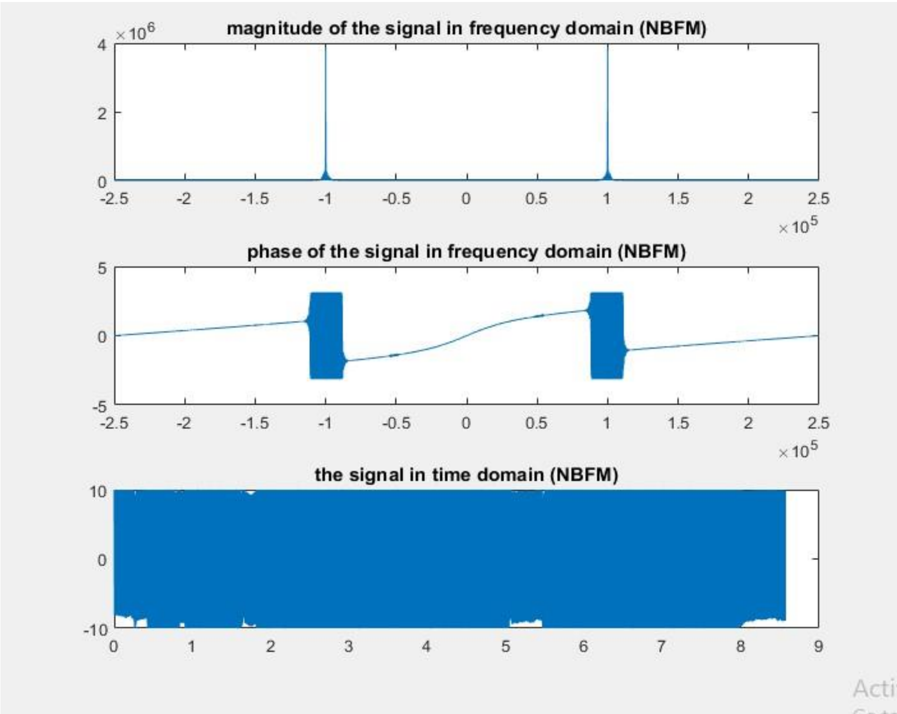
1. Sound signal has a sampling frequency Fs= 48 KHz. the spectrum of this signal (in frequency domain & time domain).



2. ideal Filter, remove all frequencies greater than 4KHz



3. the NBFM signal. Use a carrier frequency of 100kHz and a sampling frequency of $Fs = 5F$. the resulting spectrum:



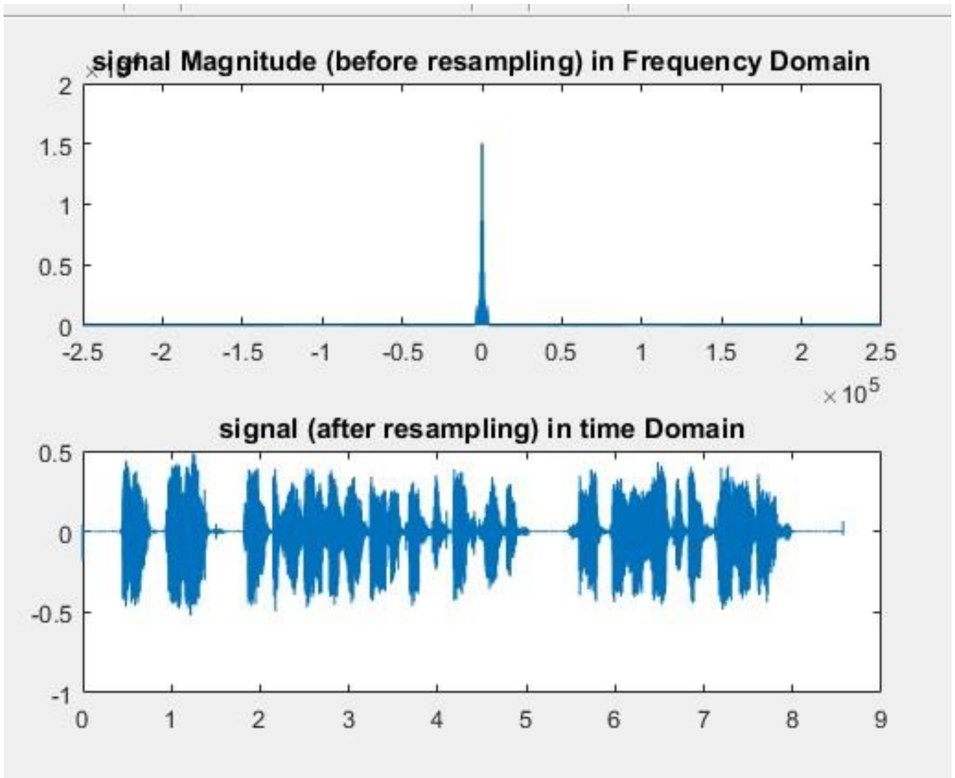
What can you make out of the resulting plot?

•The plot in NBFM is similar to the DSB-Tc's, the bandwidth of transmitted signal is doubled of frequency of original message (BW=2fm) so, using the NBFM is not useful .

What is the condition we needed to achieve NBFM?

1. BW = 2fm (β + 1), as β must be very small to make the BW ≈2fm.
2. Kf must be small.

4. Demodulate the NBFM signal using a differentiator and an ED:



Conclusion:

The plot of the spectrum, modulator, and demodulator are similar to the DSB-Tc.