2021

Final Analogue Project Report

[AREEJ SALAHUDDIN 6389]
[SOHAILA HAZEM 6388]
[MARIAM ELKAMHAWY 6470]
[MANAR ABDELKADER 6485]
[KHALED GAMAL 6117]

Table of Contents

EXPERIMENT 1

Double Sideband Modulation

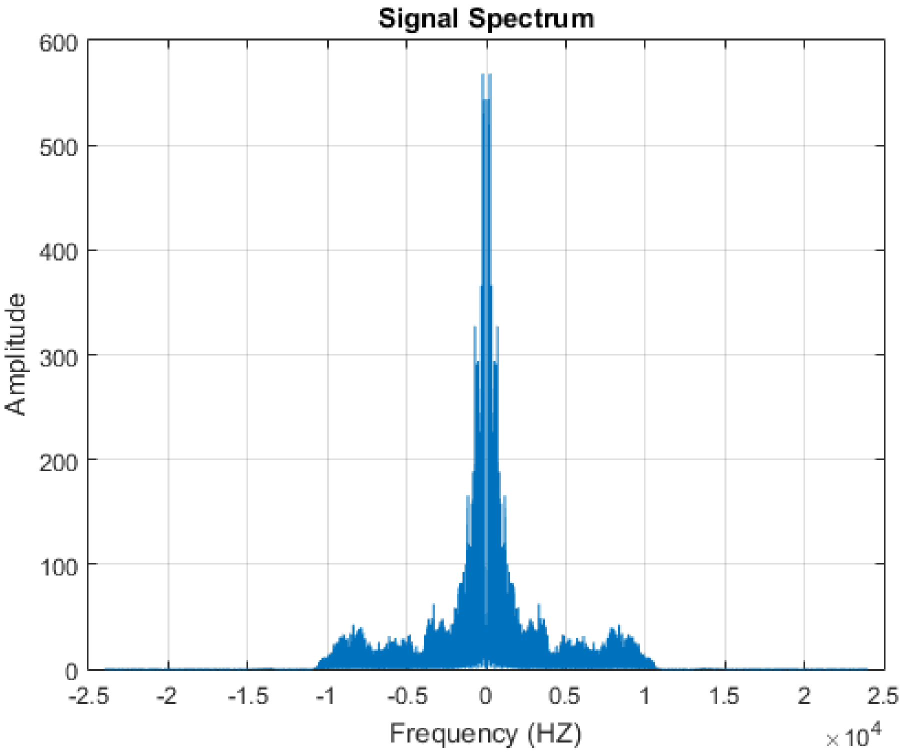
Code :

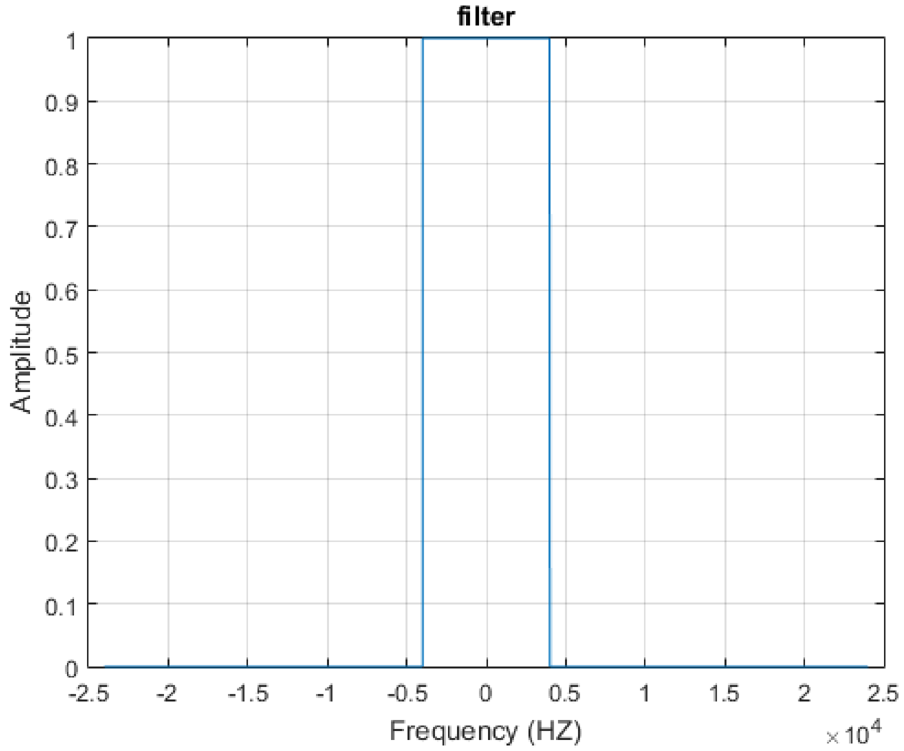
```
%%%%%%%%%%%%%%% question 1%%%%%%%%%%%%%%

[y , fs] = audioread('eric.wav');
% sound(y,fs);
plot_time(y,fs,'Signal time domain');
Yspectrum = (fftshift(fft(y)));
f = linspace(-fs/2,fs/2,length(y));
plot_frequency(abs(Yspectrum),fs,'Signal Spectrum');

%%%%%%%%%%%%%%% question 2 %%%%%%%%%%%%%%

filter = generate_filter(length(Yspectrum),fs);
plot_frequency(filter,fs,'filter');
filteredSpectrum = Yspectrum.* filter ;
ytime_filtered = real(ifft(ifftshift(filteredSpectrum))); % for sounding

%%%%%%%%%%%%%%% question 3 %%%%%%%%%%%%%%

plot_time(ytime_filtered,fs,'Filtered Signal in Time Domain');
plot_frequency(abs(filteredSpectrum),fs,'Filtered Signal in Frequency Domain');

%%%%%%%%%%%%%%% question 4 %%%%%%%%%%%%%%

% sound(ytime_filtered,fs);

%%%%%%%%%%%%%%% question 5 %%%%%%%%%%%%%%
fc = 100000;
FS = 5*fc;
resampledSignal = resample(ytime_filtered,FS,fs);
AC = 2 * max(abs(resampledSignal));
grid;

t = linspace(0,length(resampledSignal)/FS,length(resampledSignal));

%DSB-SC
carrierSignal = cos(2 * pi * fc * t);
modulated_time_DSB_SC= carrierSignal.'.* resampledSignal;
modulated_DSB_SC = fftshift(fft(modulated_time_DSB_SC));
f_new = linspace(-FS/2, FS/2,length(resampledSignal));
figure;
plot(f_new,abs(modulated_DSB_SC));
title('DSB-SC modulated signal in Frequency domain');
grid;
```

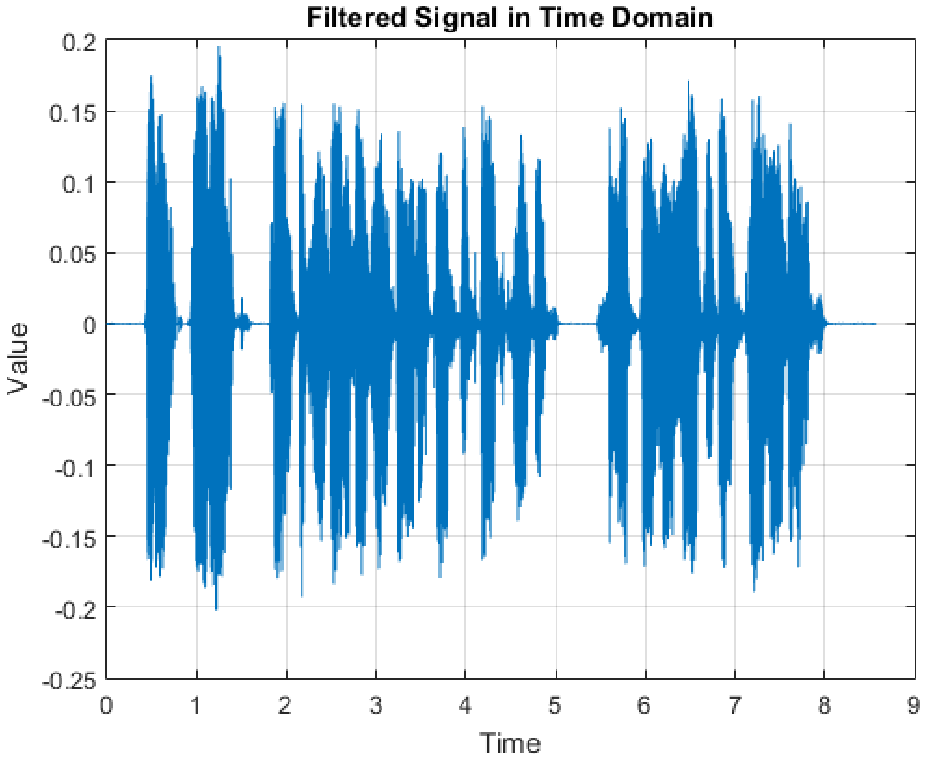
```
%DSB_TC
modulated_time_DSB_TC= (AC + resampledSignal).* carrierSignal.' ;
modulated_DSB_TC = fftshift(fft(modulated_time_DSB_TC));
figure;
plot(f_new,abs(modulated_DSB_TC));
title('DSB-TC modulated signal in Frequency domain');
grid;

plot_time(modulated_time_DSB_SC,FS,'DSB-SC modulated signal in time domain');
plot_time(modulated_time_DSB_TC,FS,'DSB-TC modulated signal in time domain');

%%%%%%%%%%%%%%% question 6 %%%%%%%%%%%%%%

%DSB_SC
envelope_SC = abs(hilbert(modulated_time_DSB_SC));
%DSB-TC
envelope_TC = abs(hilbert(modulated_time_DSB_TC));

%%%%%%%%%%%%%%% question 7 %%%%%%%%%%%%%%

%DSB-SC
figure;
plot(envelope_SC);
title('Envelope of DSB-SC signal');
grid;
envelopeResampled_SC=resample(envelope_SC,fs,FS);
%sound(envelopeResampled_SC,fs);

%DSB-TC
figure;
plot(envelope_TC);
title('Envelope of DSB-TC signal');
grid;
envelopeResampled_TC=resample(envelope_TC,fs,FS);
% sound(envelopeResampled_TC,fs);
% envelope detector should be used with TC only

%%%%%%%%%%%%%%% question 8 %%%%%%%%%%%%%%

% DSB-SC
for i = [0 10 30]
demodCoherent_SC = awgn(modulated_time_DSB_SC,i);
t = linspace(0,length(demodCoherent_SC)/FS,length(demodCoherent_SC));
demodCoherent_SC= demodCoherent_SC.*cos(2*pi*fc*t).';  %Coherent detection
Yr = fftshift(fft(demodCoherent_SC));  %Fourier transform
Yr_filter= generate_filter(length(modulated_time_DSB_SC),FS);
Yr_msg = Yr.*Yr_filter;  %Obtained audio message after coherent detection
plot_frequency(abs(Yr_msg),FS,'DSB-SC audio signal spectrum');
time_demodSC = ifft(ifftshift(Yr_msg));   %Inverse fourier transform
time_demodSC = resample(real(time_demodSC),fs,FS);
plot_time(time_demodSC,fs,['DSB-SC audio signal in time domain with noise: ',num2str(i),' db']);
```

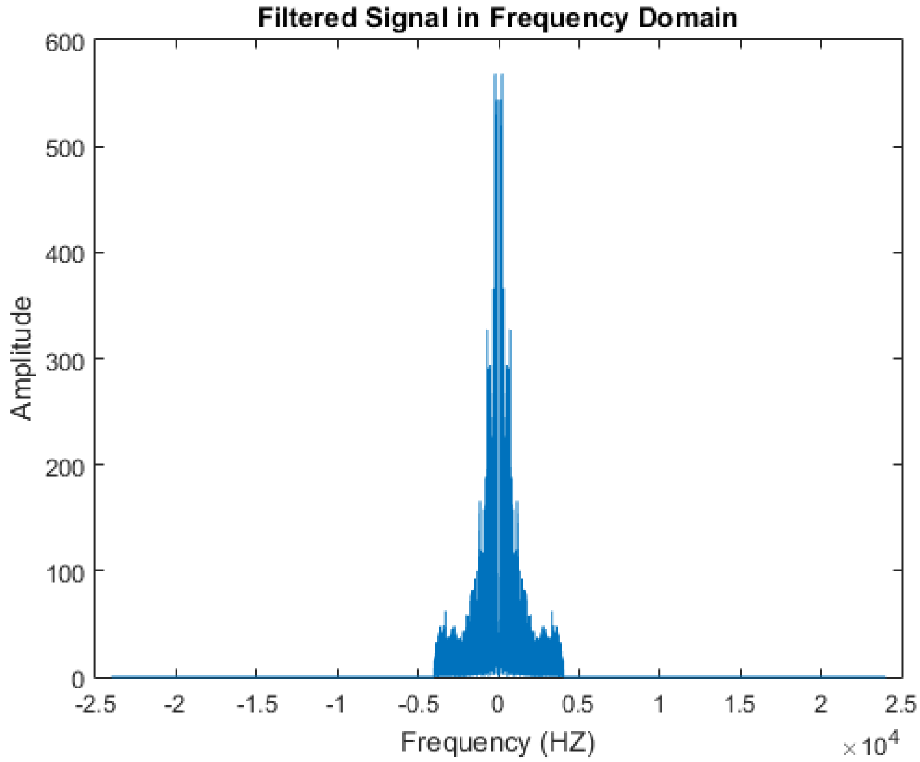
```
freq_demodSC = fftshift(fft(time_demodSC));
plot_frequency(real(abs(freq_demodSC)), fs, ['DSB-SC audio signal in freq domain with noise:
',num2str(i),' db']);

% audiowrite(['Signal_DSB_SC_SNR',num2str(i),'.wav'],time_demodSC,fs);
end

%%%%%%%%%%%%%%% question 9 %%%%%%%%%%%%%%
fc_new = 100100;

t = linspace(0,length(modulated_time_DSB_SC)/FS,length(modulated_time_DSB_SC));
demod_freqError= modulated_time_DSB_SC.*cos(2*pi*fc_new*t).';  %Coherent detection
demod_freqError = fftshift(fft(demod_freqError));
demod_freqError = demod_freqError.*Yr_filter;
plot_frequency(abs(demod_freqError),FS,'DSB-SC audio signal with frequency error spectrum');
time_demodSC = ifft(ifftshift(demod_freqError));   %Inverse fourier transform
time_demodSC = resample(real(time_demodSC),fs,FS);
plot_time(time_demodSC,fs,'DSB-SC audio signal in time domain with freq error');
freq_demodSC = fftshift(fft(time_demodSC));
plot_frequency(real(abs(freq_demodSC)), fs, 'DSB-SC audio signal in freq domain with freq
error');
%sound(time_demodSC,fs);
% audiowrite(['Signal_with_frequencyError.wav'],time_demodSC,fs);

%%%%%%%%%%%%%%% question 10 %%%%%%%%%%%%%%
phaseshift = degtorad(20);
t = linspace(0,length(modulated_time_DSB_SC)/FS,length(modulated_time_DSB_SC));
demod_phaseError= modulated_time_DSB_SC.*cos(2*pi*fc*t + phaseshift).';  %Coherent detection
demod_phaseError = fftshift(fft(demod_phaseError));
demod_phaseError = demod_phaseError.*Yr_filter;
plot_frequency(abs(demod_phaseError),FS,'DSB-SC audio signal with phase error spectrum');
time_demodSC = ifft(ifftshift(demod_phaseError));   %Inverse fourier transform
time_demodSC = resample(real(time_demodSC),fs,FS);
plot_time(time_demodSC,fs,'DSB-SC audio signal in time domain with phase error');
freq_demodSC = fftshift(fft(time_demodSC));
plot_frequency(real(abs(freq_demodSC)), fs, 'DSB-SC audio signal in freq domain with phase
error');
%sound(time_demodSC,fs);
% audiowrite(['Signal_with_phaseError.wav'],time_demodSC,fs);
```

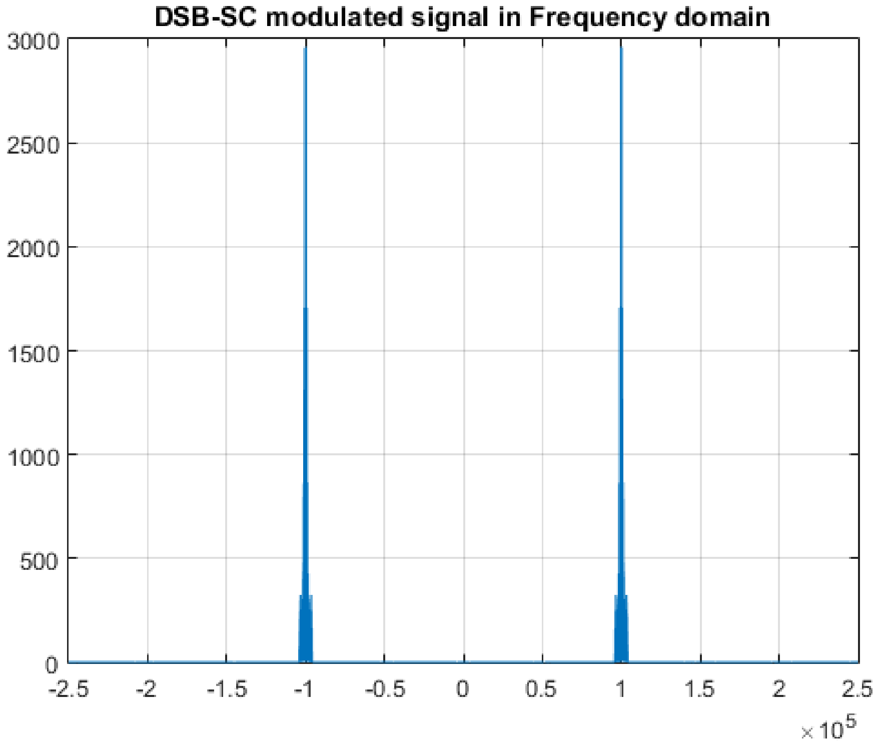
Figures:

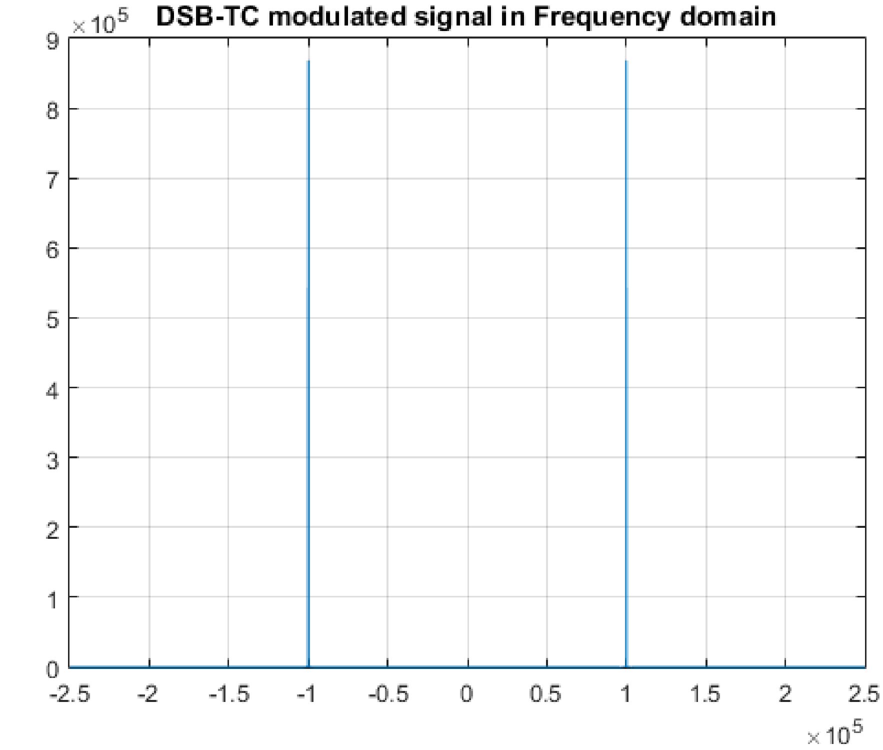

Signal Spectrum
Amplitude
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
$\times 10^4$

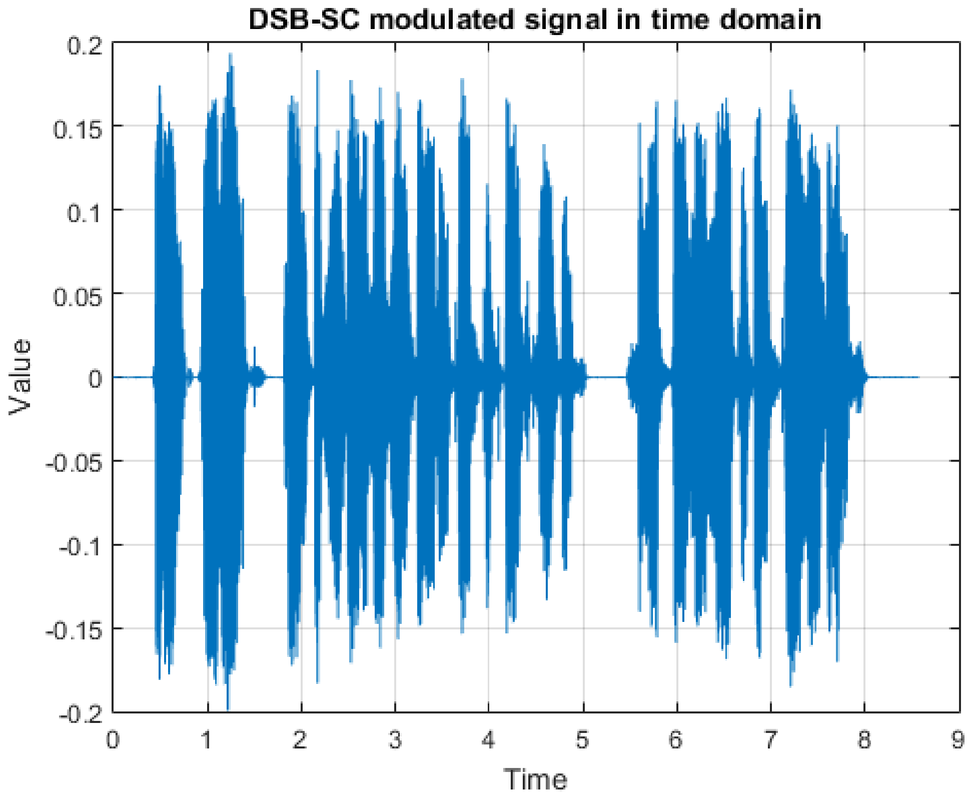

filter
Amplitude
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
$\times 10^4$

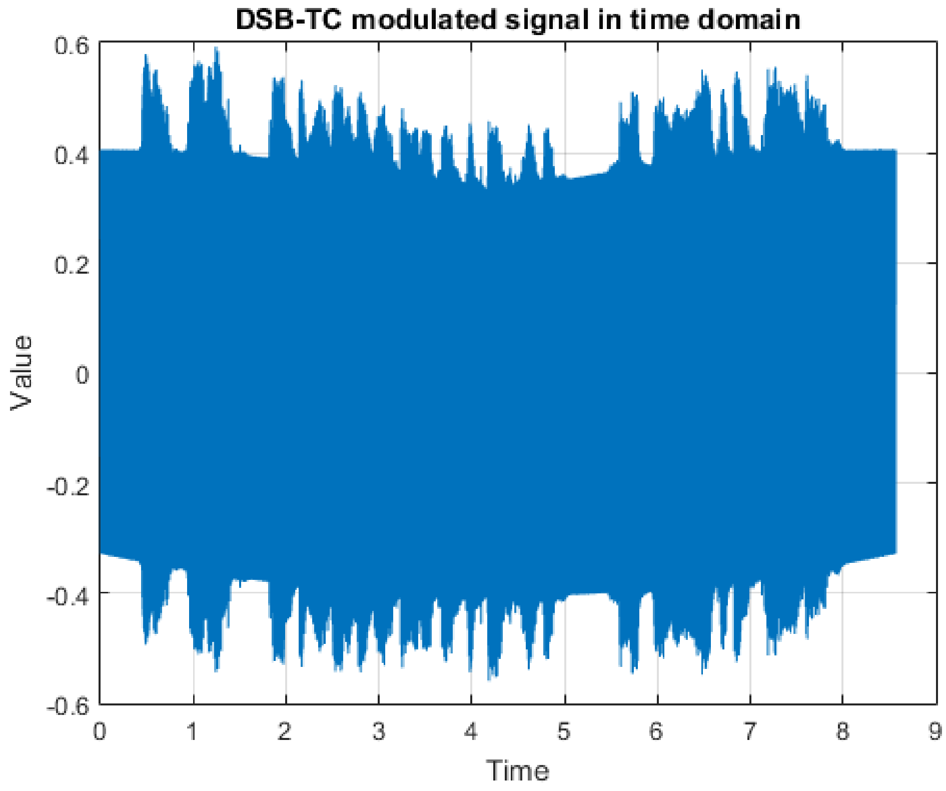

Filtered Signal in Time Domain
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9

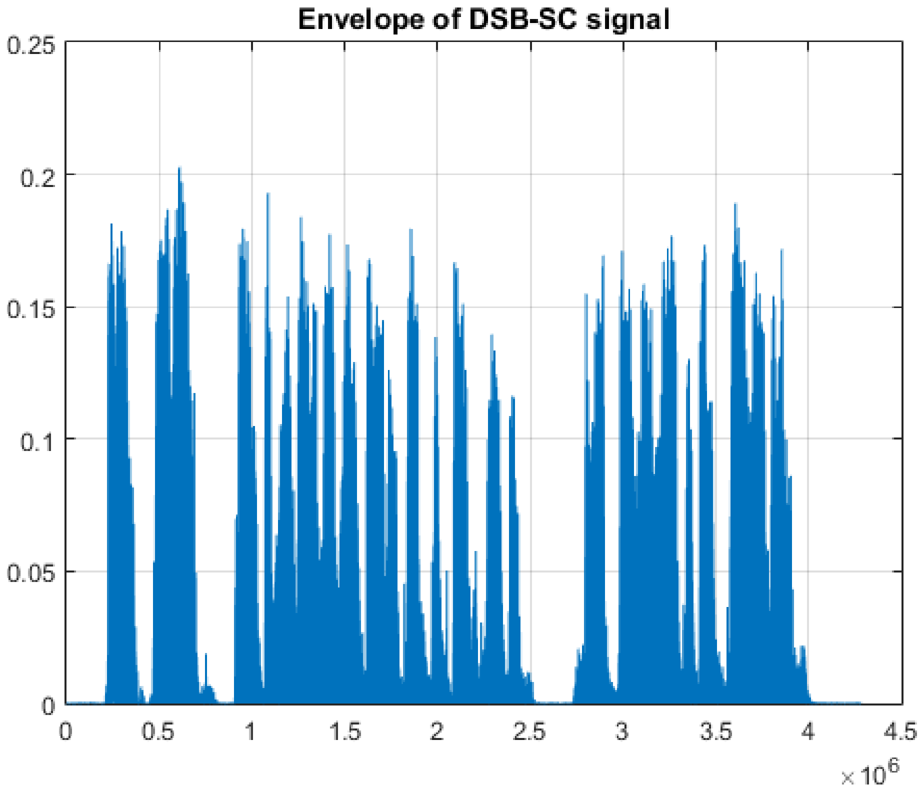

Filtered Signal in Frequency Domain
Amplitude
Frequency (HZ)
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^4

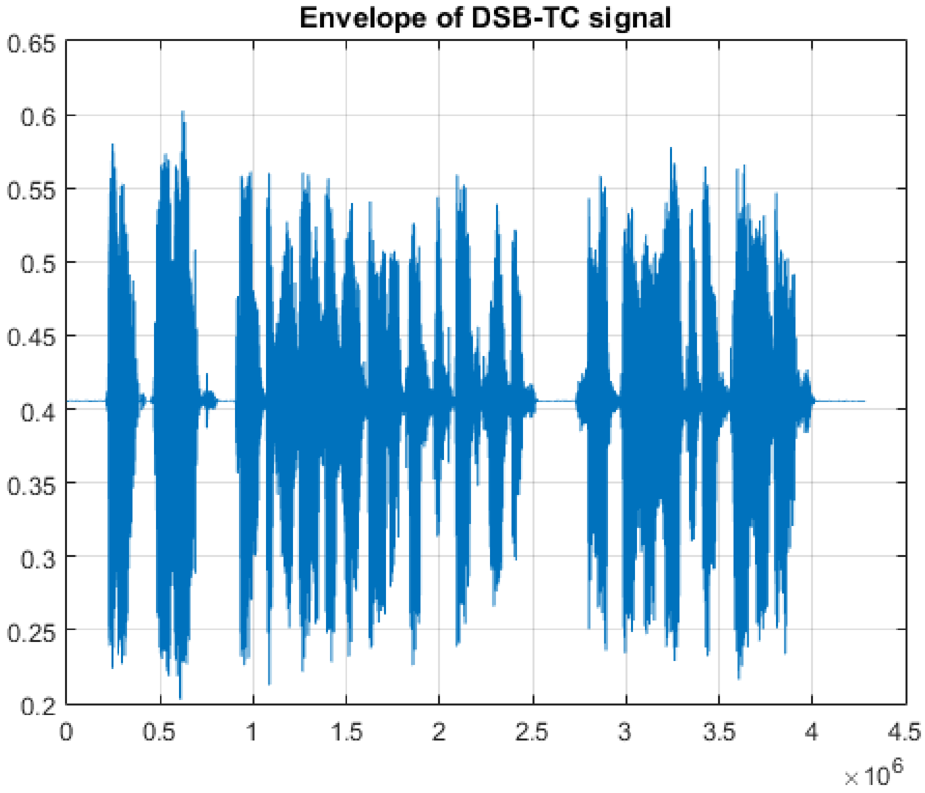

DSB-SC modulated signal in Frequency domain
3000
2500
2000
1500
1000
500
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
× 10^5


× 10^5
DSB-TC modulated signal in Frequency domain
9
8
7
6
5
4
3
2
1
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
× 10^5

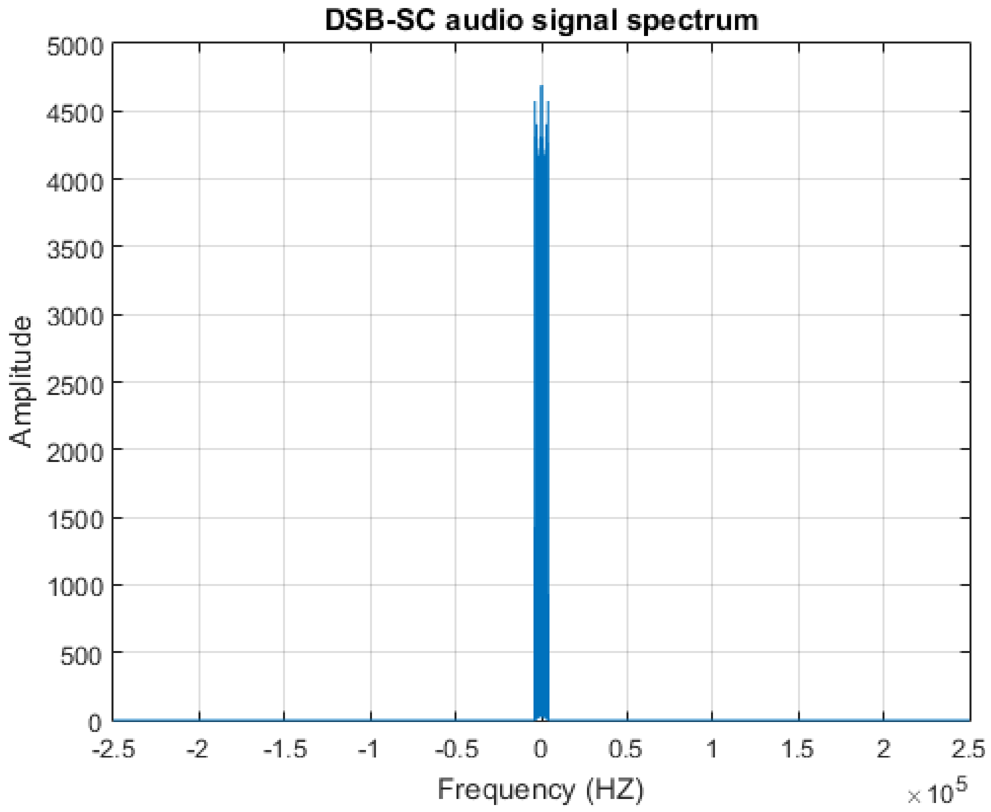

DSB-SC modulated signal in time domain
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
1
2
3
4
5
6
7
8
9

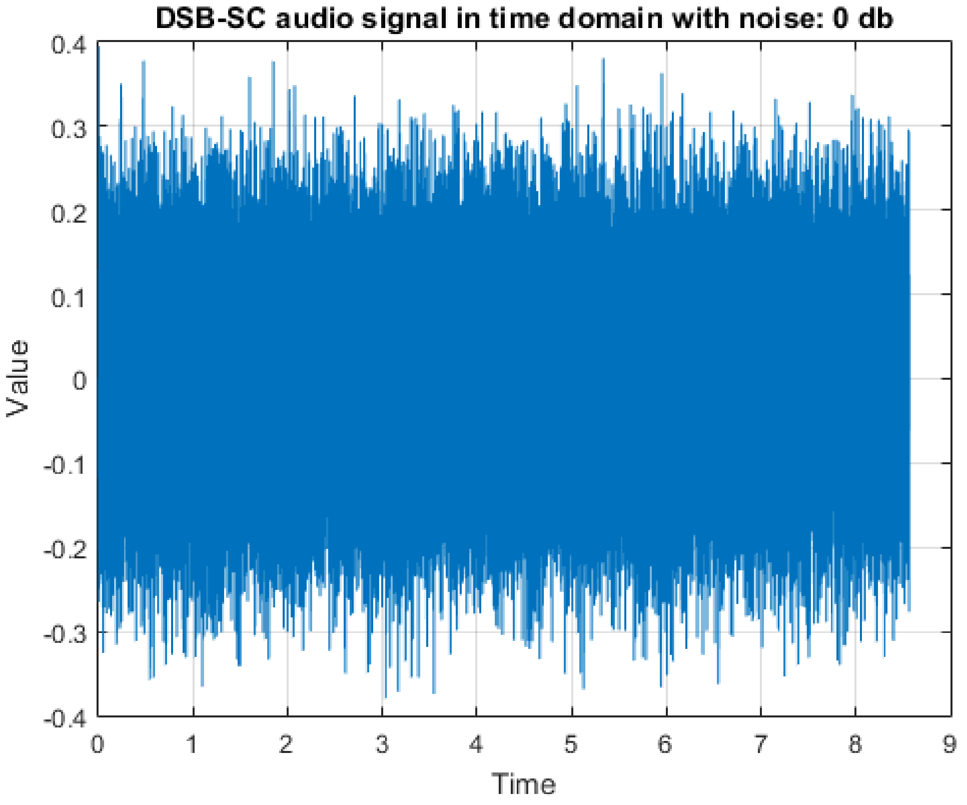

DSB-TC modulated signal in time domain
Value
Time
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
7
8
9

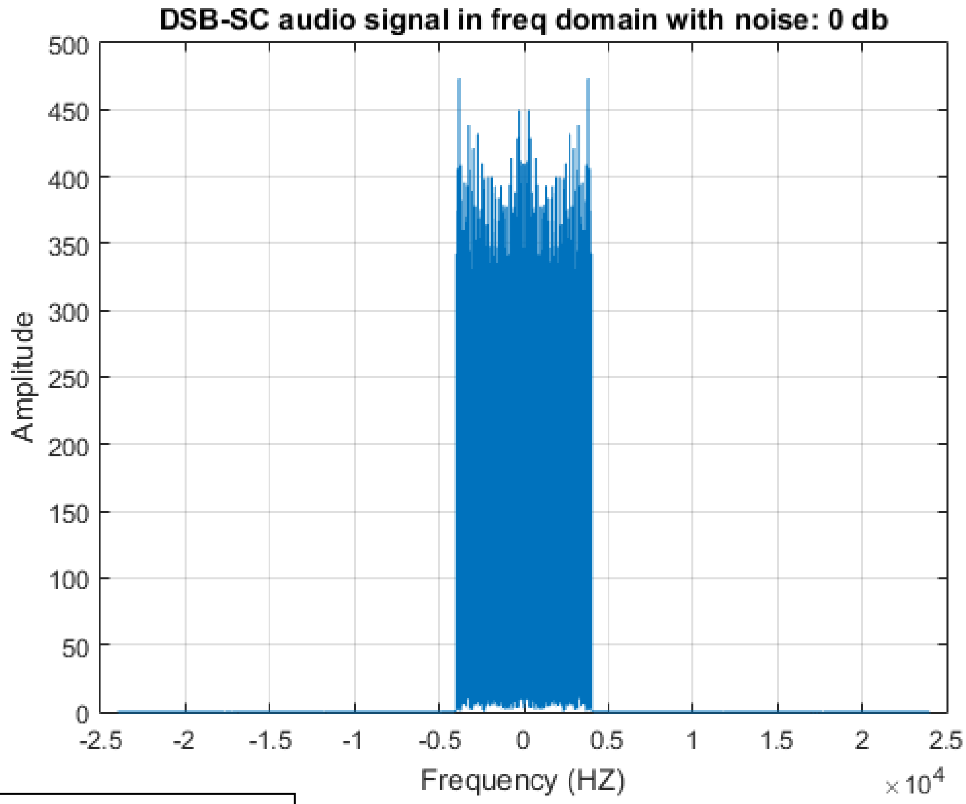

Envelope of DSB-SC signal
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
×10^6


Envelope of DSB-TC signal
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
×10^6

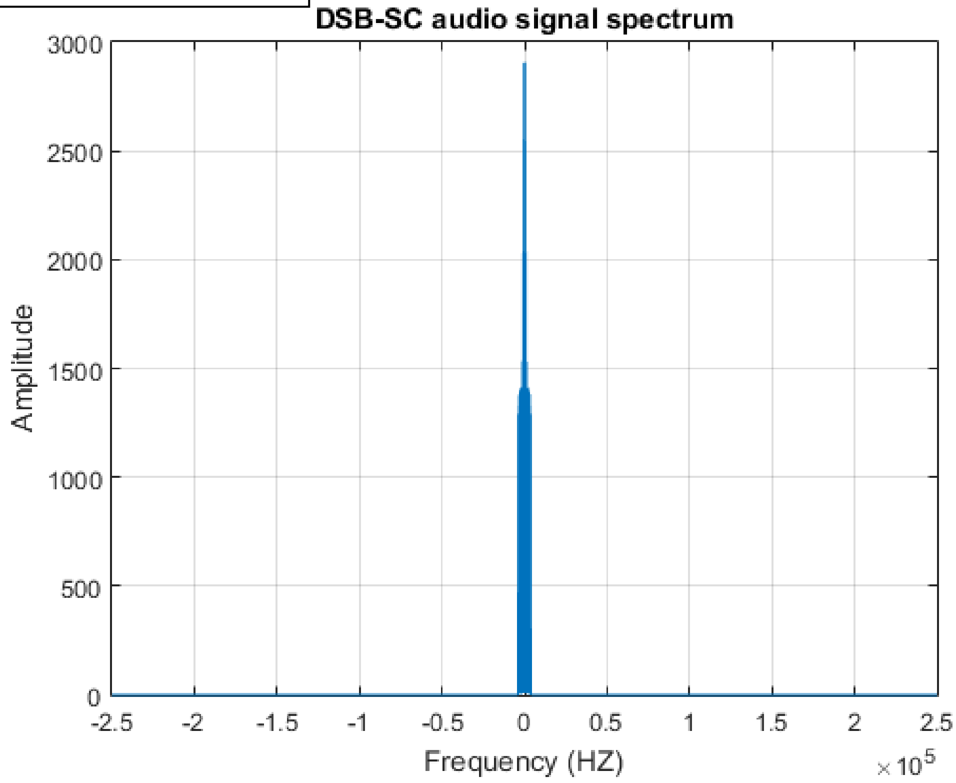
0 db Before Resampling: Fs = 5*Fc

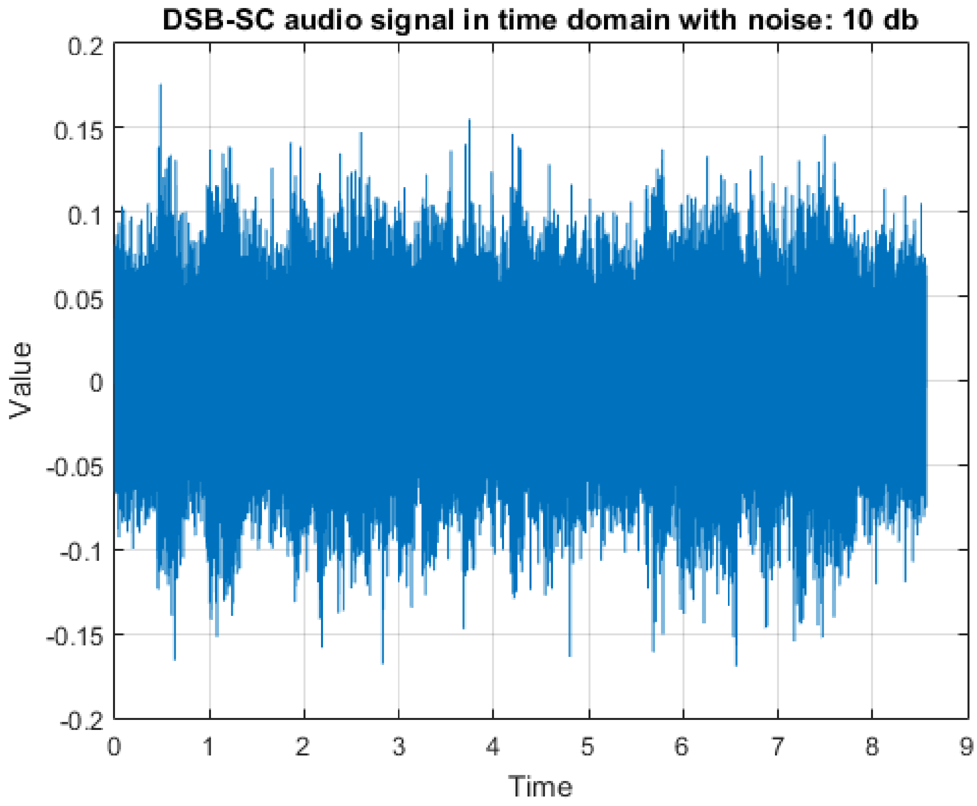


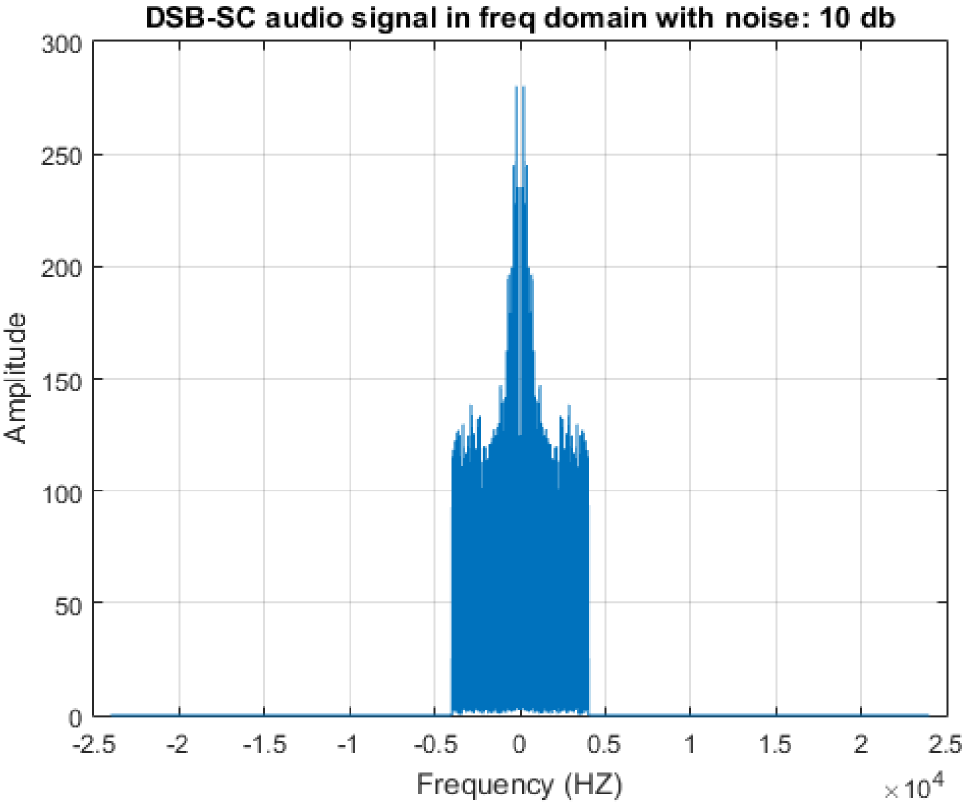




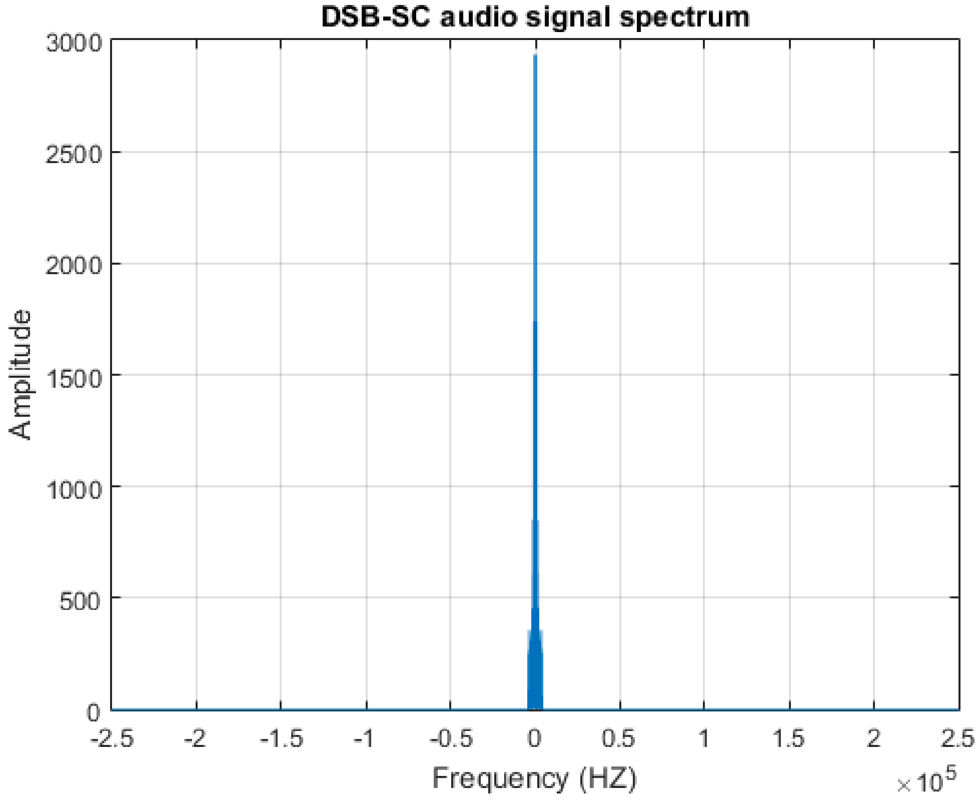
10 db Before Resampling: Fs = 5*Fc

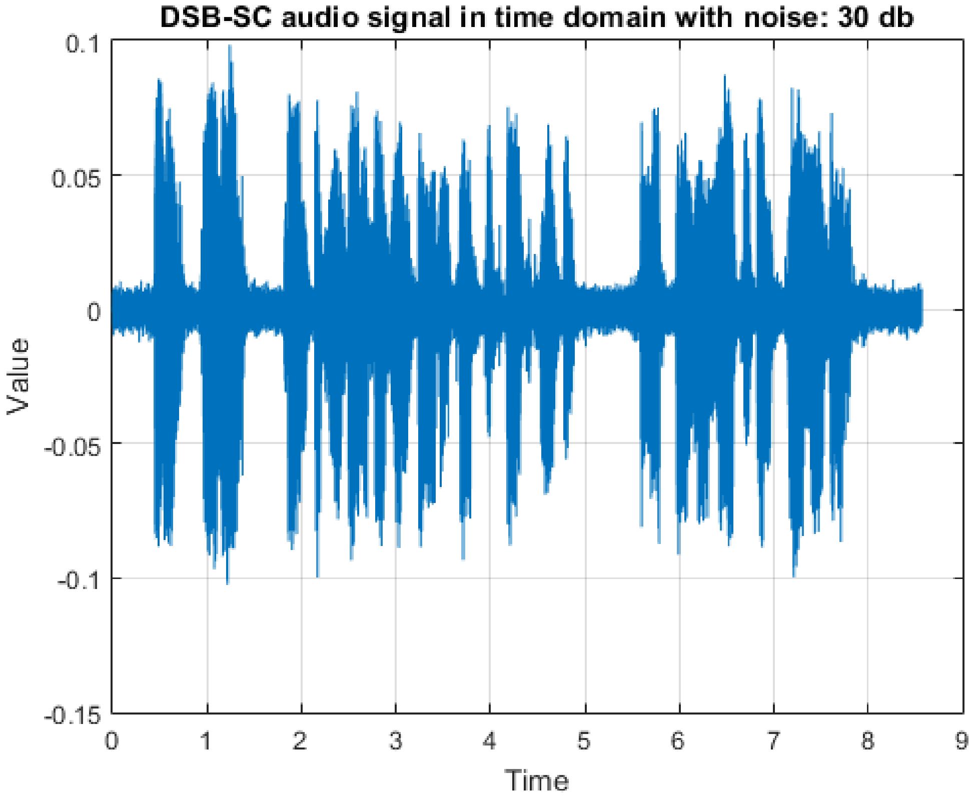


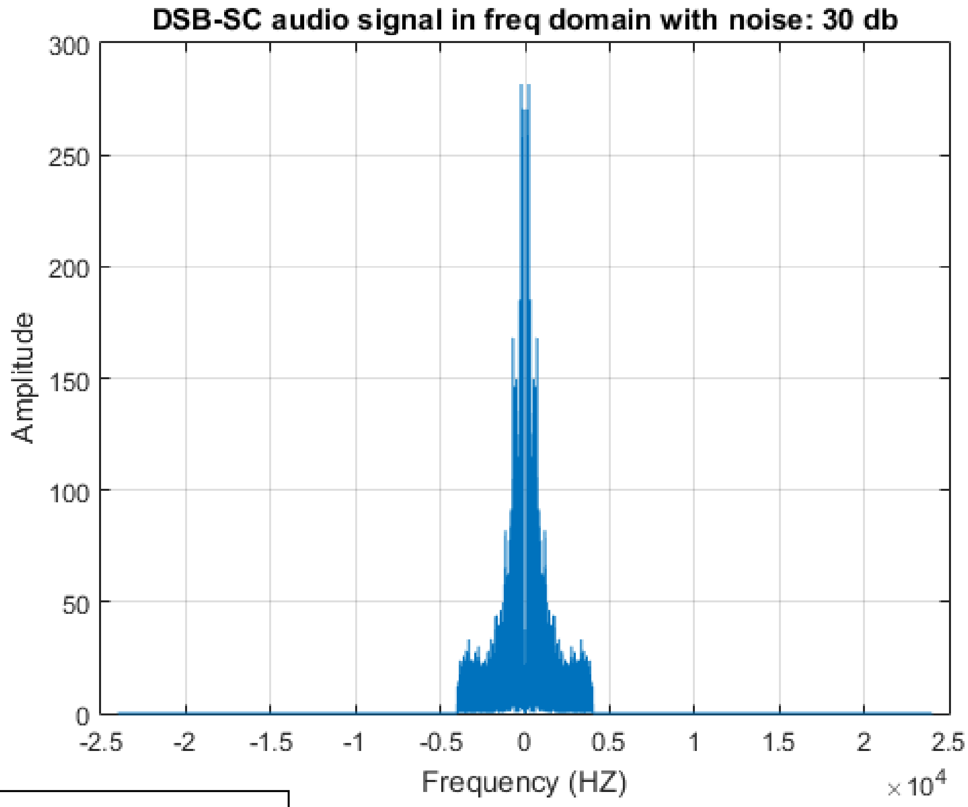

DSB-SC audio signal in time domain with noise: 10 db
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
1
2
3
4
5
6
7
8
9


DSB-SC audio signal in freq domain with noise: 10 db
Amplitude
Frequency (HZ)
300
250
200
150
100
50
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
$\times 10^4$

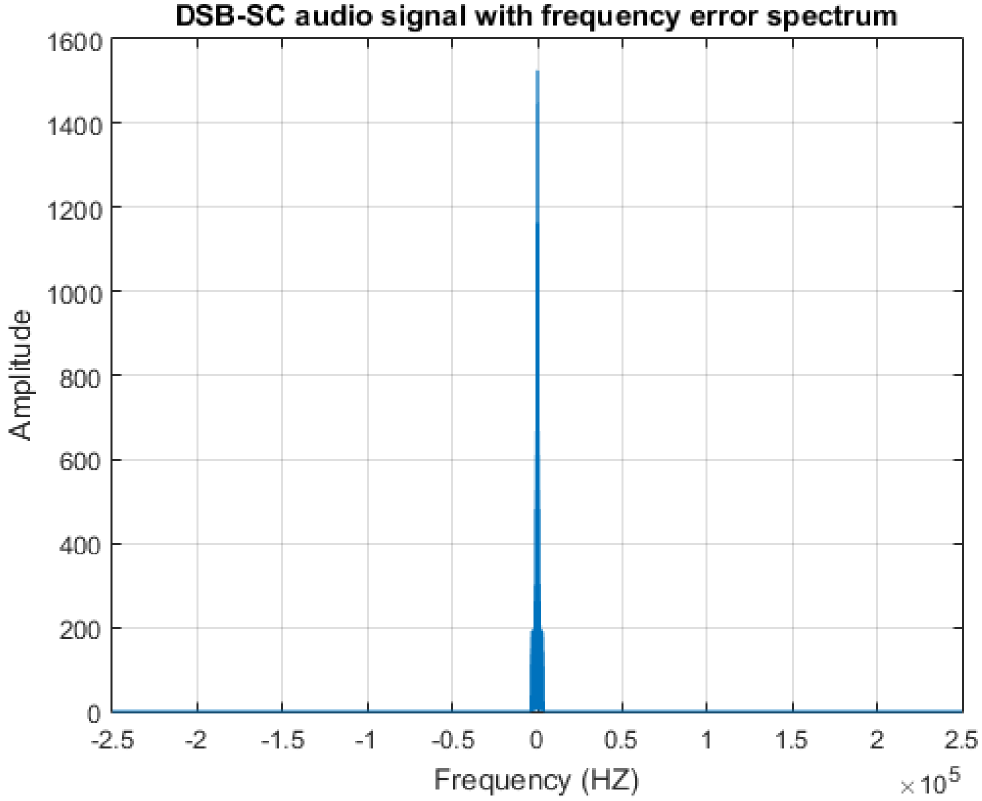
30 db Before Resampling: Fs = 5*Fc

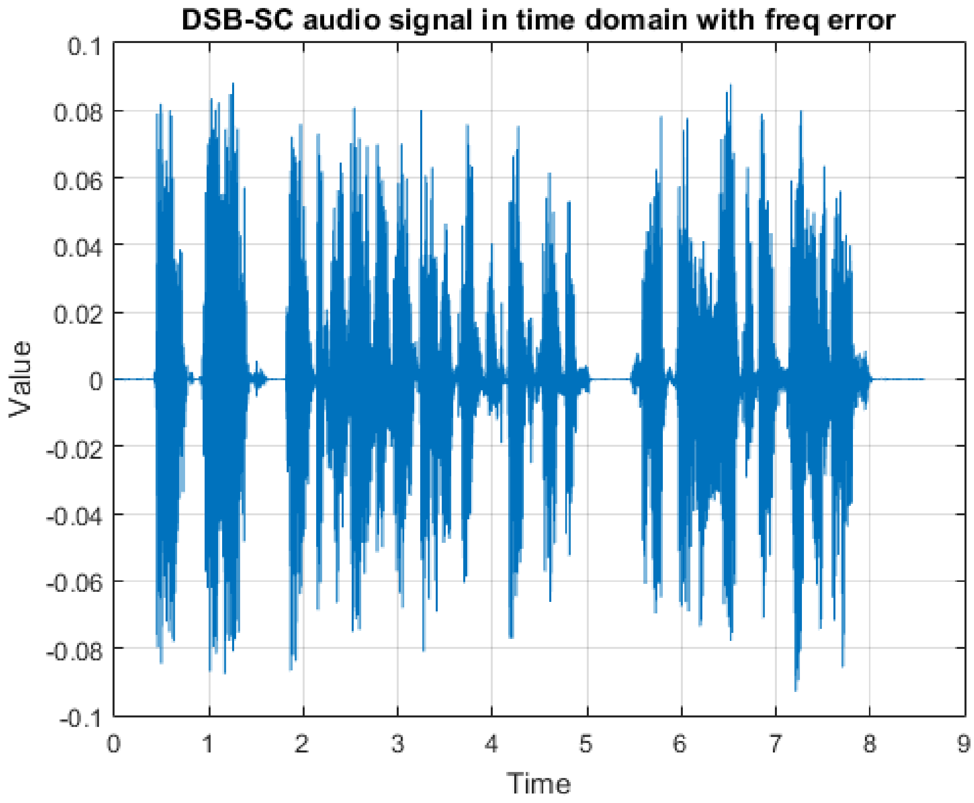


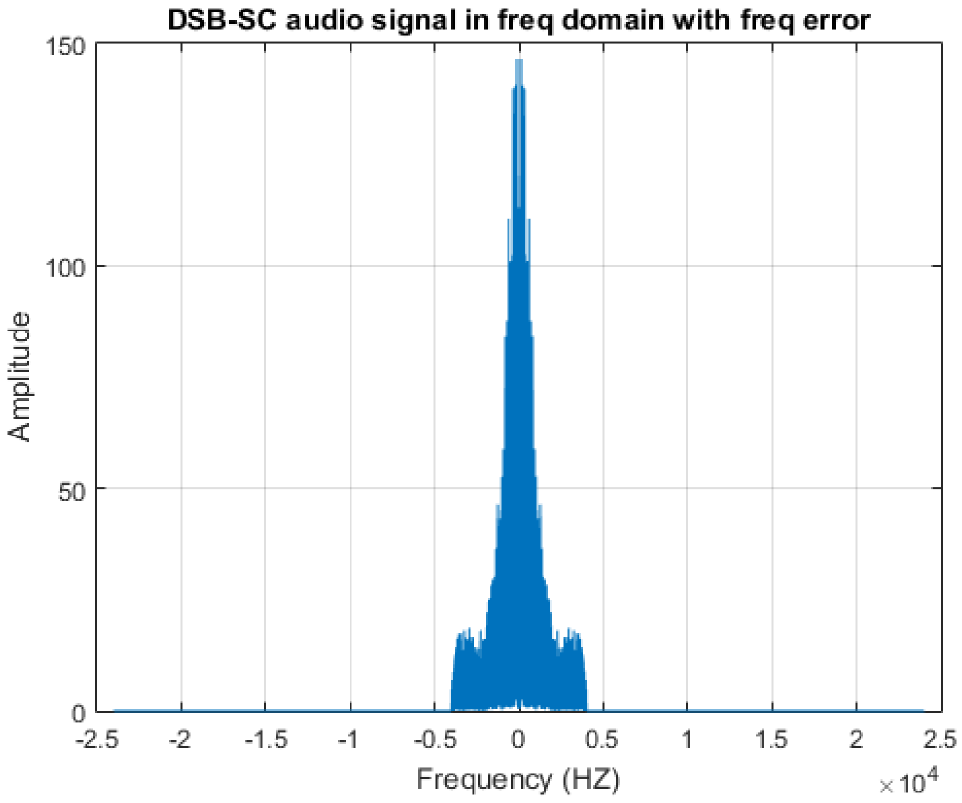




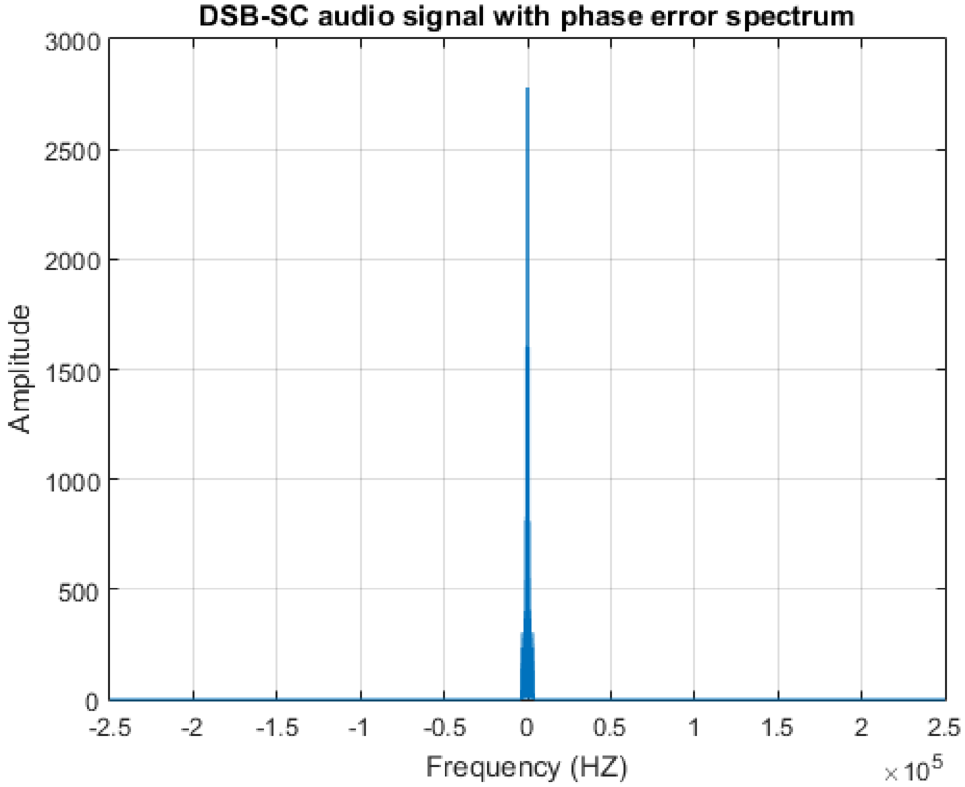
Before Resampling: Fs = 5*Fc

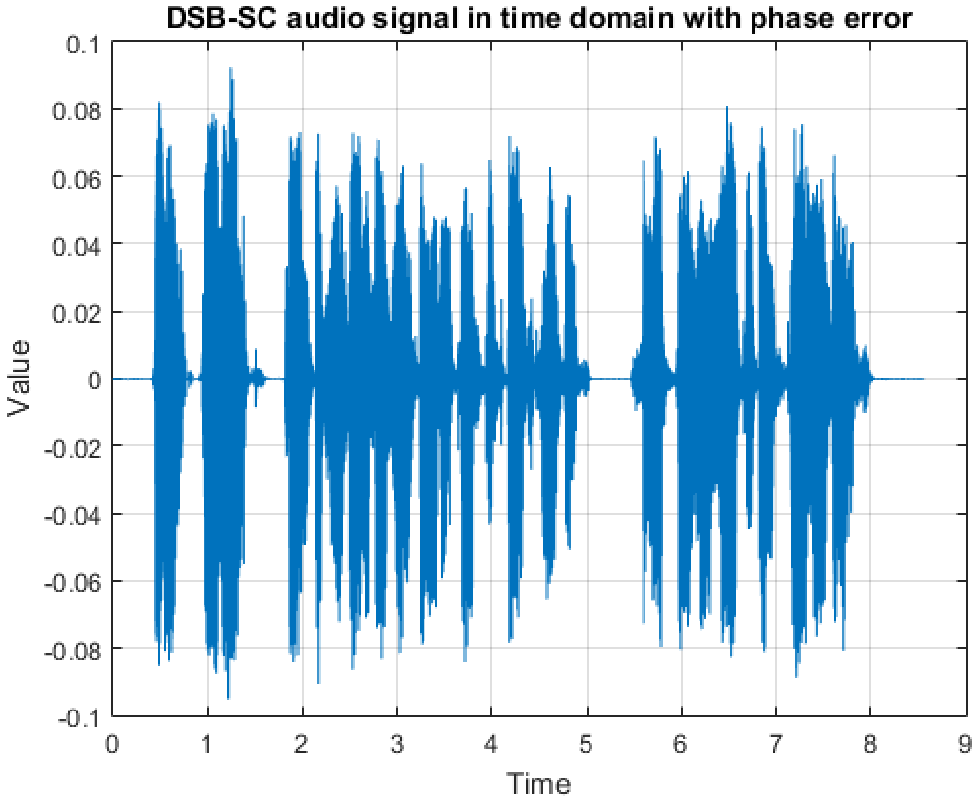


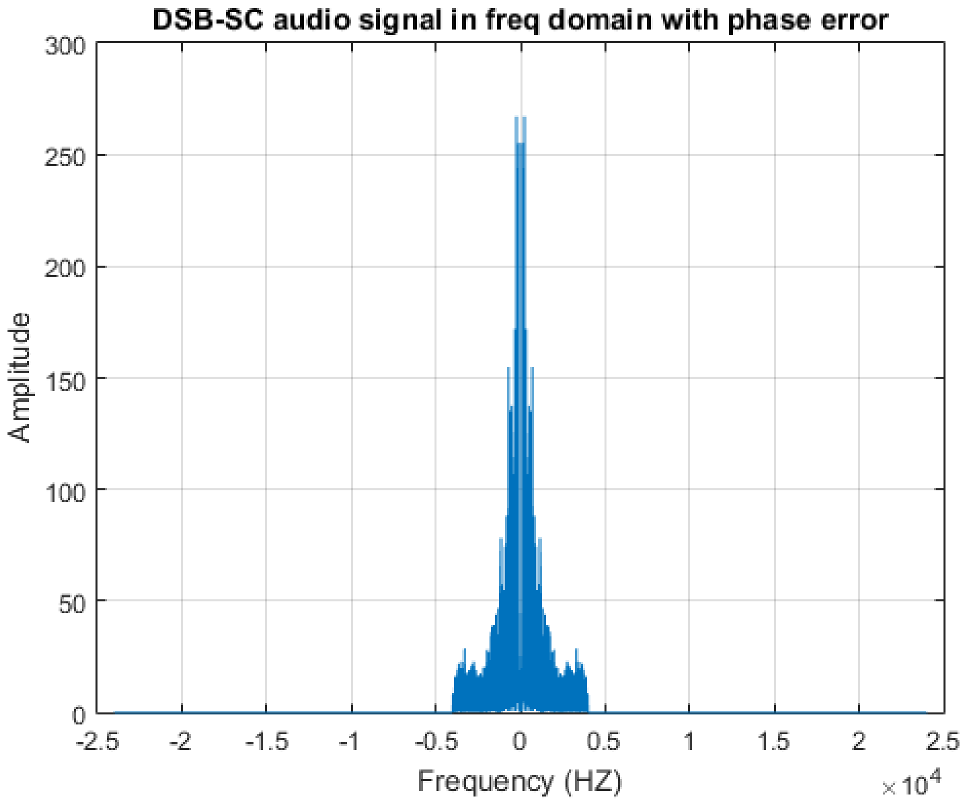

DSB-SC audio signal in time domain with freq error
Value
Time
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
1
2
3
4
5
6
7
8
9


DSB-SC audio signal in freq domain with freq error
Amplitude
Frequency (HZ)
150
100
50
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^4

Before Resampling: Fs = 5*Fc







Theoretical Questions:

Question 7: What observation can you make of this or which type of modulation the envelope detector can be used with?

➔ We observe that DSB-SC cannot be demodulated using the envelope detector while the DSB-TC can be demodulated using the envelope detector.

Question 9: Do you have a name for this phenomenon?

➔ Distortion occurs due to the frequency error and the name of this phenomenon is "Beating effect".

Question 10: *Repeat Question 9 for* Phase error?

➔ Attenuation occurs due to the Phase error and the name of this phenomenon is "Quadrature null effect".

Experiment (2):

SINGLE SIDEBAND MODULATION

Code :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%(1)%%%%%%%%%%%%%%%%%%%%%%%%%
[y , fs] = audioread('eric.wav');
% sound(y,fs);
y_Spectrum = fftshift(fft(y));
f = linspace(-fs/2,fs/2,length(y));
t = linspace(0, length(y)/fs, length(y));
plot_time(y,fs,'Signal in time domain');
plot_frequency(abs(y_Spectrum),fs,'Signal in Frequency Domain');


%%%%%%%%%%%%%%%%%%%%%%%%%%%(2)%%%%%%%%%%%%%%%%%%%%%%%%%
filter1 = ones(1,length(y_Spectrum)).*(f >= -4000 & f <= 4000);



%%%%%%%%%%%%%%%%%%%%%%%%%%%(3)%%%%%%%%%%%%%%%%%%%%%%%%%
filteredSpectrum = y_Spectrum.' .* filter1 ;
ytime_filtered = real(ifft(ifftshift(filteredSpectrum)));
plot_time(ytime_filtered,fs,'Filtered Signal in Time Domain');
plot_frequency(abs(filteredSpectrum),fs,'Filtered Signal in Frequency Domain');
% sound(ytime_filtered,fs);


%%%%%%%%%%%%%%%%%%%%%%%%%%%(4)%%%%%%%%%%%%%%%%%%%%%%%%%
fc = 100000;
FS = 5*fc;
resampledSignal = resample(ytime_filtered,FS,fs);
f_new = linspace(-FS/2, FS/2,length(resampledSignal));
t_new = linspace(0, length(resampledSignal)/FS, length(resampledSignal));
carrierSignal = cos(2 * pi * fc * t_new);
modulated_DSB_SC_time = carrierSignal.* resampledSignal;
modulated_DSB_SC_Spectrum = fftshift(fft(modulated_DSB_SC_time));
plot_time(modulated_DSB_SC_time,FS,'DSB-SC modulated Signal in Time Domain');
plot_frequency(abs(modulated_DSB_SC_Spectrum),FS,'DSB-SC modulated Signal in Frequency Domain');

%%%%%%%%%%%%%%%%%%%%%%%%%%%(5)%%%%%%%%%%%%%%%%%%%%%%%%%
filter2 = ones(1,length(modulated_DSB_SC_Spectrum)).*(f_new >= -fc & f_new <= fc);
modulated_SSB_Spectrum = modulated_DSB_SC_Spectrum.*filter2;
modulated_SSB_time = real(ifft(ifftshift(modulated_SSB_Spectrum)));
plot_time(modulated_SSB_time,FS,'SSB-SC modulated Signal in Time Domain (LSB) Ideal Filter');
plot_frequency(abs(modulated_SSB_Spectrum),FS,'SSB-SC modulated Signal in Frequency Domain (LSB)
Ideal Filter');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%(6)%%%%%%%%%%%%%%%%%%%%%%%%%
demodulated_time = modulated_SSB_time.*carrierSignal;
filter3 = designfilt('lowpassfir', 'FilterOrder', 8000, 'CutoffFrequency', 4000, 'SampleRate',
FS);
coherent_demodulated_time  = filter(filter3, demodulated_time);
coherent_demodulated_freq = fft(fftshift(coherent_demodulated_time));
plot_time(real(coherent_demodulated_time),FS,'Demodulated Signal in Time Domain (Coherent
Detection Ideal Filter)');
plot_frequency(abs(real(coherent_demodulated_freq)),FS,'Demodulated Signal in Frequency Domain
(Coherent Detection Ideal Filter)');
resampled_time = resample(coherent_demodulated_time, fs, FS);
resampled_freq = fftshift(fft(resampled_time));
plot_time(real(resampled_time),fs,'Demodulated Signal in Time Domain (Coherent Detection Ideal
Filter)');
plot_frequency(abs(real(resampled_freq)),fs,'Demodulated Signal in Frequency Domain (Coherent
Detection Ideal Filter)');
%sound(real(resampled_time),fs);


%%%%%%%%%%%%%%%%%%%%%%%%%%%(7)%%%%%%%%%%%%%%%%%%%%%%%%%
```

step (5) %%%

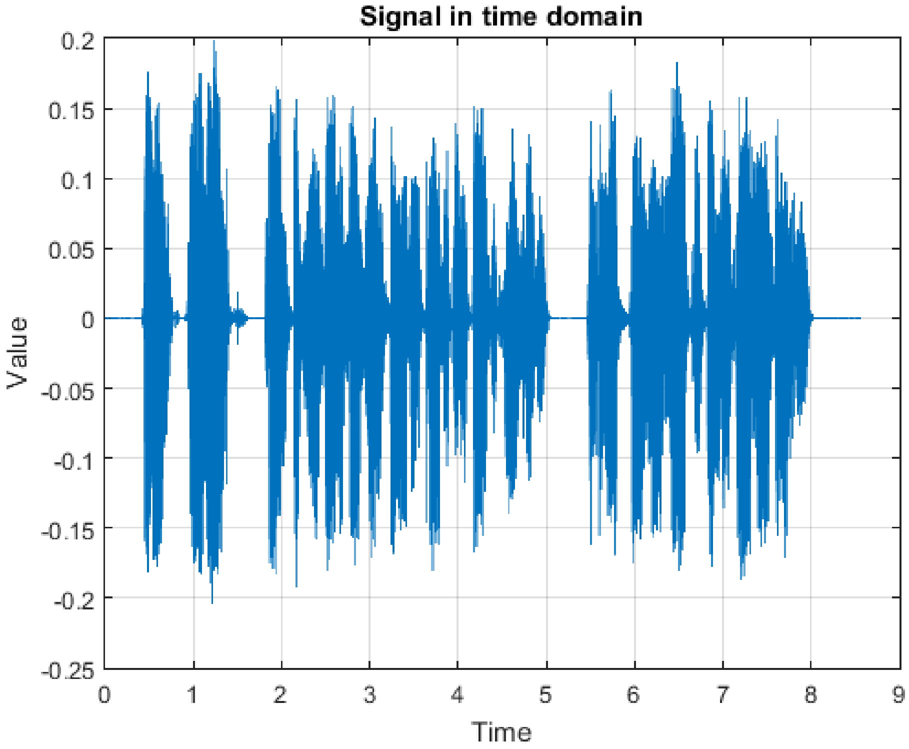
```
fcn = FS/2;
fc1 = (fc - 4000)/fcn;
fc2 = fc/fcn;
[b1, a1] = butter(4,[fc1 fc2]);
SSB_practical_time = filter(b1, a1, modulated_DSB_SC_time);
SSB_practical_Spectrum = real(fftshift(fft(SSB_practical_time)));
plot_time(SSB_practical_time,FS,'Butter-worth SSB-SC modulated Signal in Time Domain (LSB) 4th
order');
plot_frequency(abs(SSB_practical_Spectrum),FS,'Butter-worth SSB-SC modulated Signal in Frequency
Domain (LSB) 4th order');
```

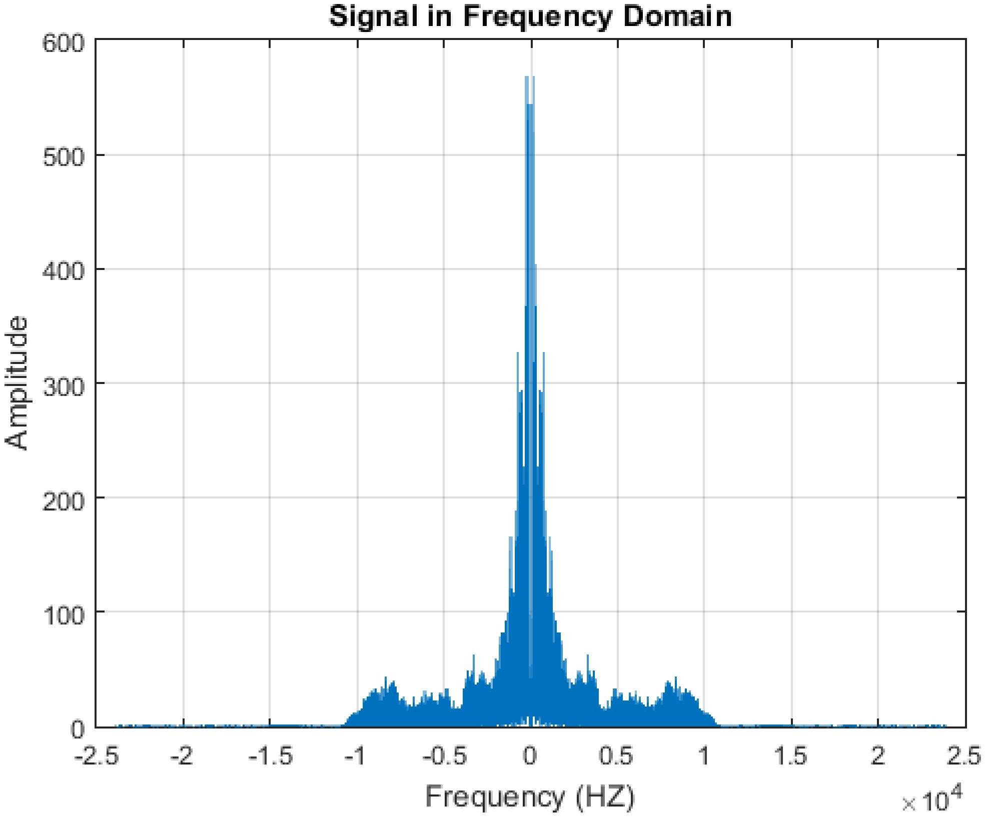
step (6) %%%

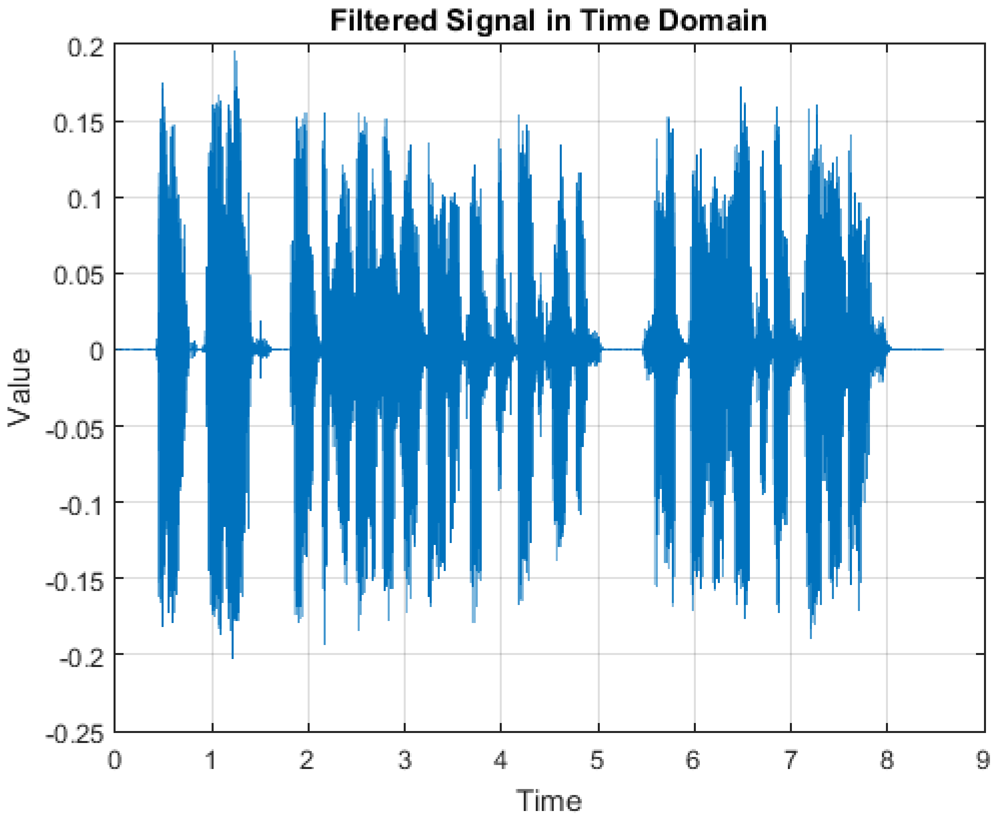
```
demodulated_time = SSB_practical_time .*carrierSignal;
demodulated_spectrum = fft(fftshift(demodulated_time));
[b2, a2] = butter(4,4000/fcn);
coherent_demodulated_time = filter(b2, a2, demodulated_time);
coherent_demodulated_Spectrum = fftshift(fft(coherent_demodulated_time));
plot_time(coherent_demodulated_time,FS,'Butter-worth Demodulated Signal in Time Domain (Coherent
Detection)');
plot_frequency(abs(coherent_demodulated_Spectrum),FS,'Butter-worth Demodulated Signal in
Frequency Domain (Coherent Detection)');
resampled_time = resample(coherent_demodulated_time, fs, FS);
resampled_freq = fftshift(fft(resampled_time));
plot_time(resampled_time,fs,'Butter-worth Demodulated Signal in Time Domain (Coherent
Detection)');
```

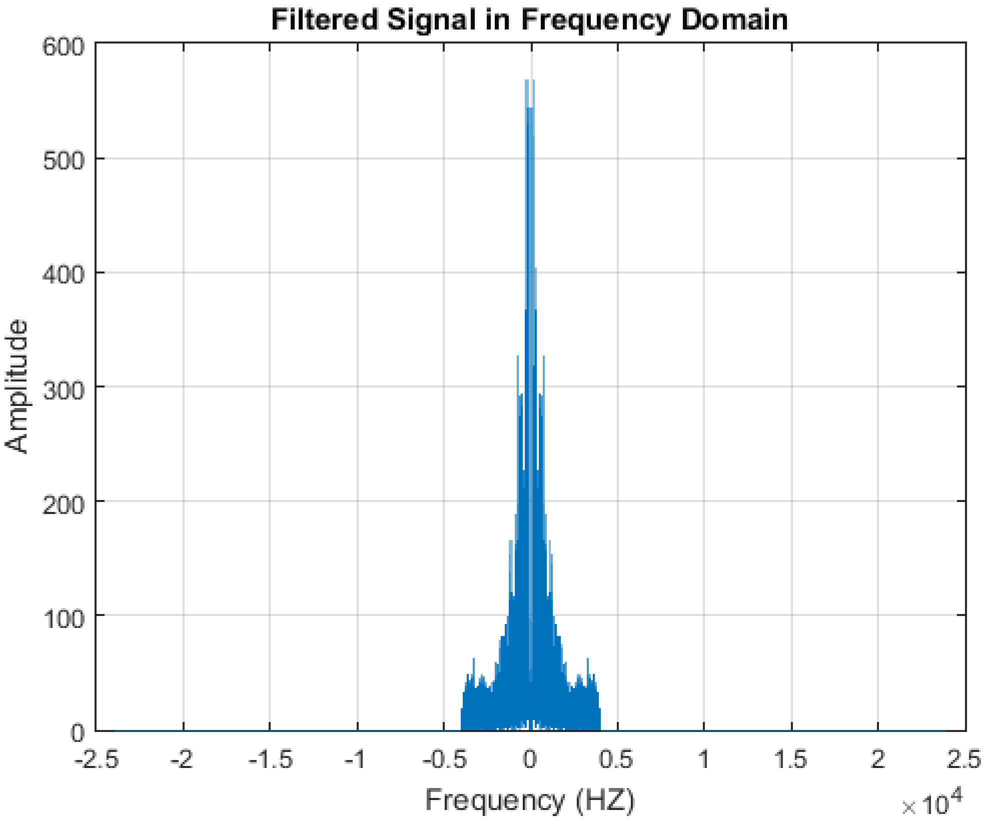
```
plot_frequency(abs(real(resampled_freq)),fs,'Butter-worth Demodulated Signal in Frequency Domain
(Coherent Detection)');
% sound(real(resampled_time),fs);


%%%%%%%%%%%%%%%%%%%%%%%%%%%(8)%%%%%%%%%%%%%%%%%%%%%%%%%
for i = [0 10 30]
demodulated_coherent_SSB_SC = awgn(modulated_SSB_time,i);
time = linspace(0, length(demodulated_coherent_SSB_SC)/FS, length(demodulated_coherent_SSB_SC));
demodulated_signal_noise_time = demodulated_coherent_SSB_SC.*cos(2*pi*fc*time);
filter4 = designfilt('lowpassfir', 'FilterOrder', 8000, 'CutoffFrequency', 4000, 'SampleRate',
FS);
demodulatedTime  = filter(filter4, demodulated_signal_noise_time);
demodulatedSpectrum = fftshift(fft(demodulatedTime));
plot_time(real(demodulatedTime),FS,['Demodulated SSB-SC signal in Time Domain:',num2str(i),'
db']);
plot_frequency(abs(demodulatedSpectrum),FS,['Demodulated SSB-SC signal in Frequency
Domain:',num2str(i),' db']);
resampled_demodulatedTime = resample(real(demodulatedTime),fs,FS);
resampled_demodulatedSpectrum = fftshift(fft(resampled_demodulatedTime));
plot_time(resampled_demodulatedTime,fs,['Demodulated SSB-SC signal in Time Domain with noise:
',num2str(i),' db']);
plot_frequency(real(abs(resampled_demodulatedSpectrum)), fs, ['Demodulated SSB-SC signal in
Frequency Domain with noise: ',num2str(i),' db']);
%audiowrite(['Signal_SSB_SC_SNR',num2str(i),'.wav'],resampled_demodulatedTime,fs);
%sound(resampled_demodulatedTime,fs);

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%(9)%%%%%%%%%%%%%%%%%%%%%%%%%
Ac = 2 * max(abs(resampledSignal));
modulated_DSB_TC_time = (Ac + resampledSignal).* carrierSignal ;
filter5 = designfilt('lowpassfir', 'FilterOrder', 8000, 'CutoffFrequency', fc, 'SampleRate', FS);
modulated_SSB_TC_Time  = filter(filter5, modulated_DSB_TC_time);
envelope_SSB_TC = abs(hilbert(modulated_SSB_TC_Time));
plot(envelope_SSB_TC);
title('Envelope of SSB-TC signal');
grid;
envelopeResampled_SSB_TC_time = resample(envelope_SSB_TC,fs,FS);
plot_time(real(envelopeResampled_SSB_TC_time), fs, 'Demodulated SSB-TC signal using Envelope
Detection in Time Domain');
% sound(envelopeResampled_SSB_TC,fs);
```

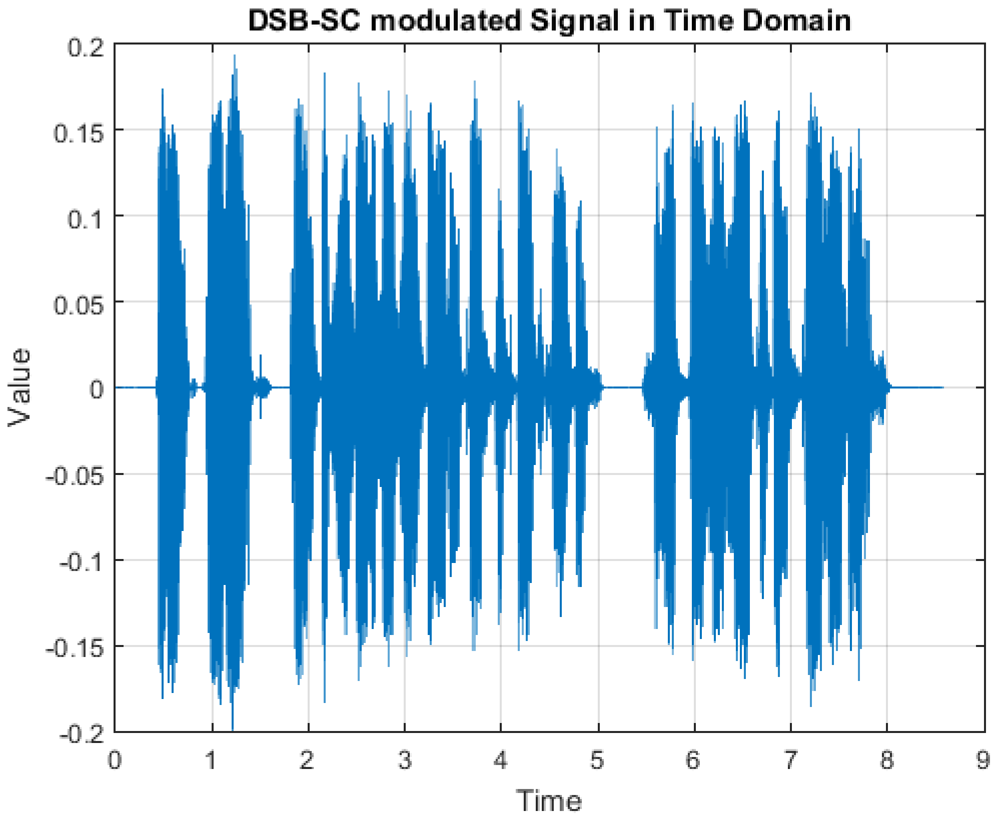
Figures:

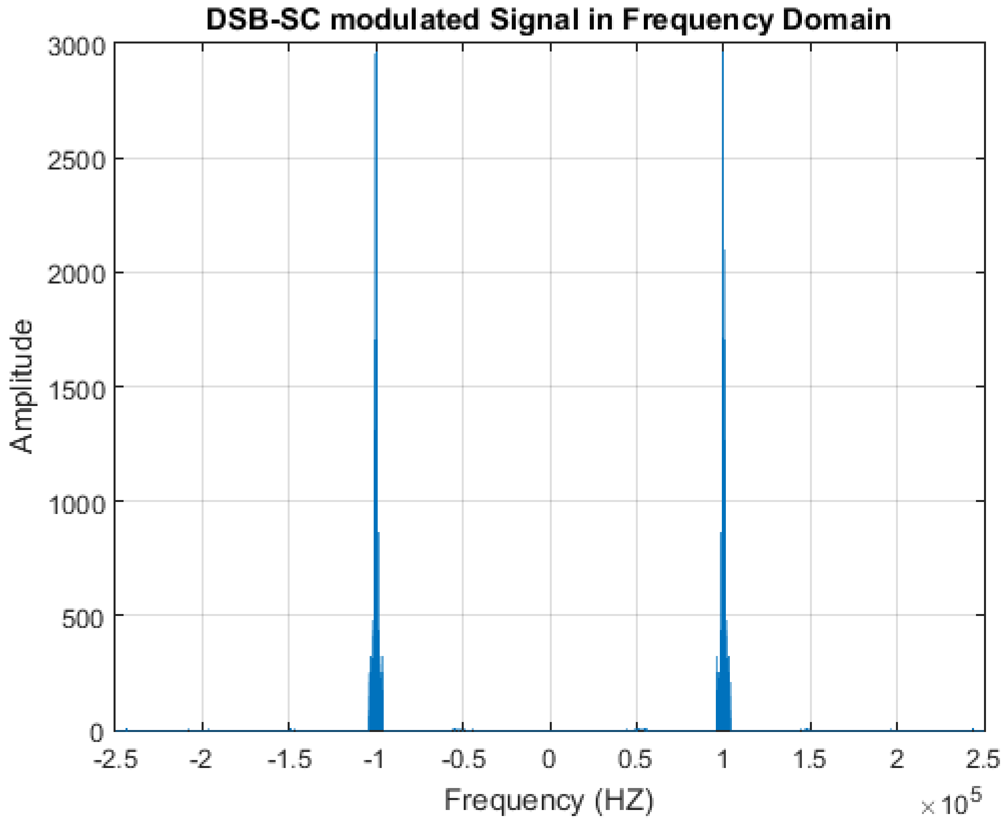

Signal in time domain
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9

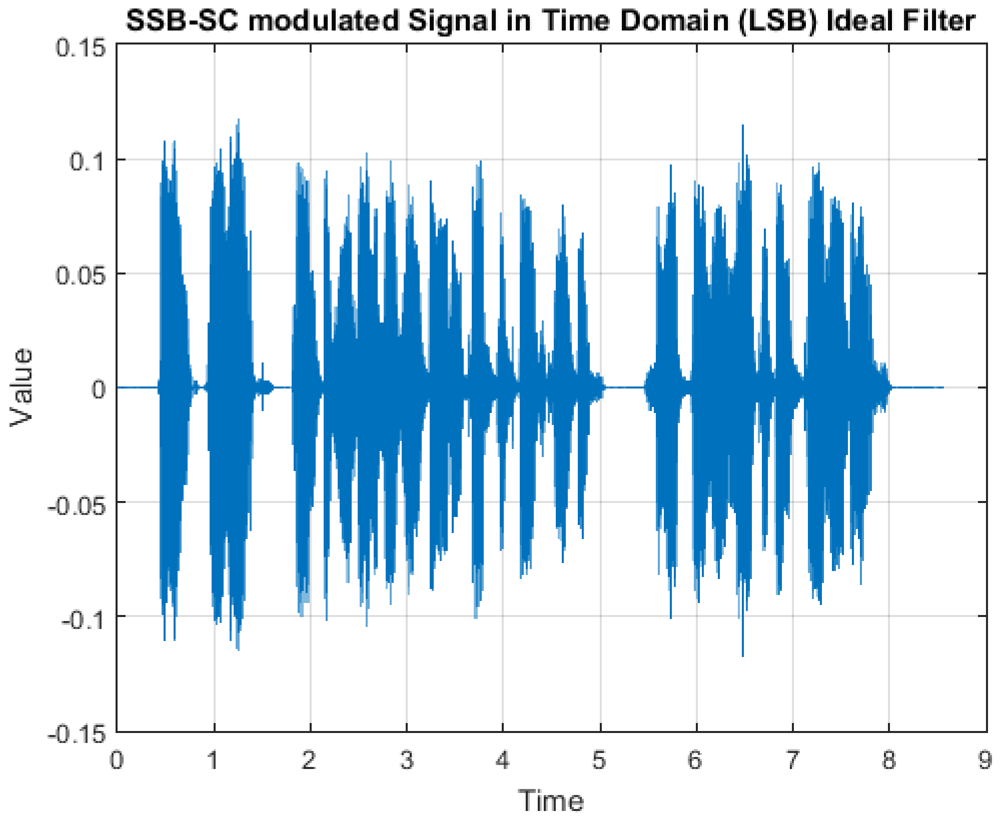

Signal in Frequency Domain
Amplitude
Frequency (HZ)
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^4

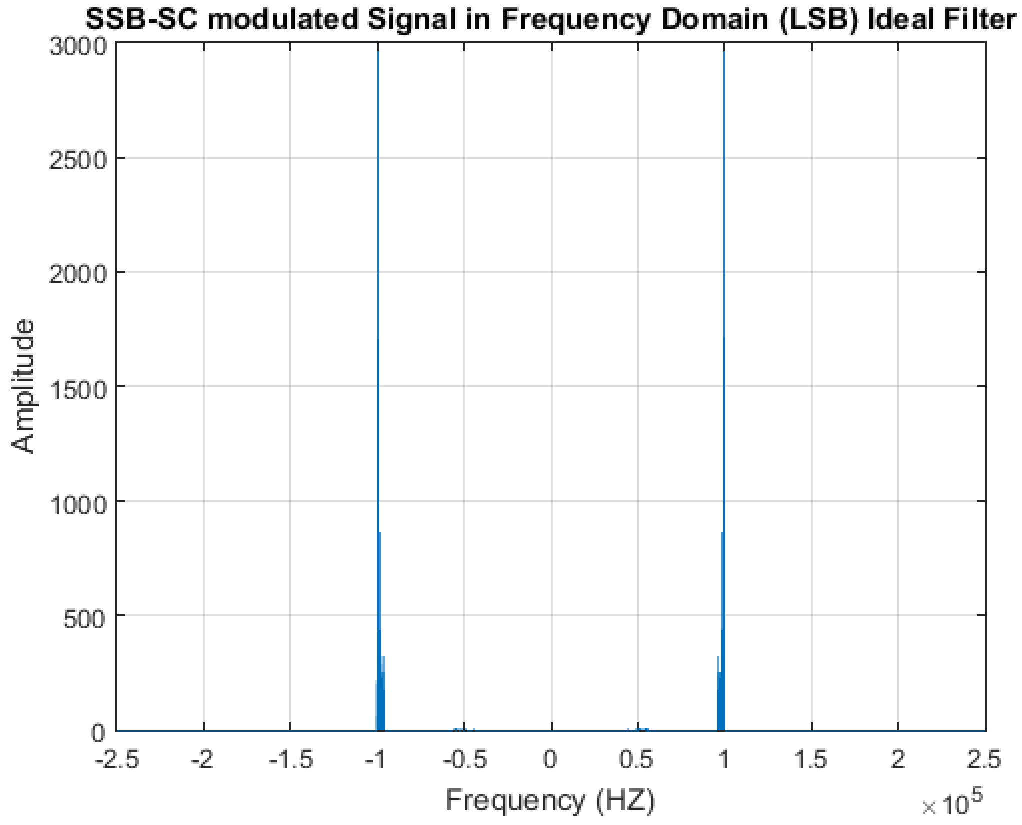

Filtered Signal in Time Domain
Value
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9
Time


Filtered Signal in Frequency Domain
Amplitude
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
×10^4

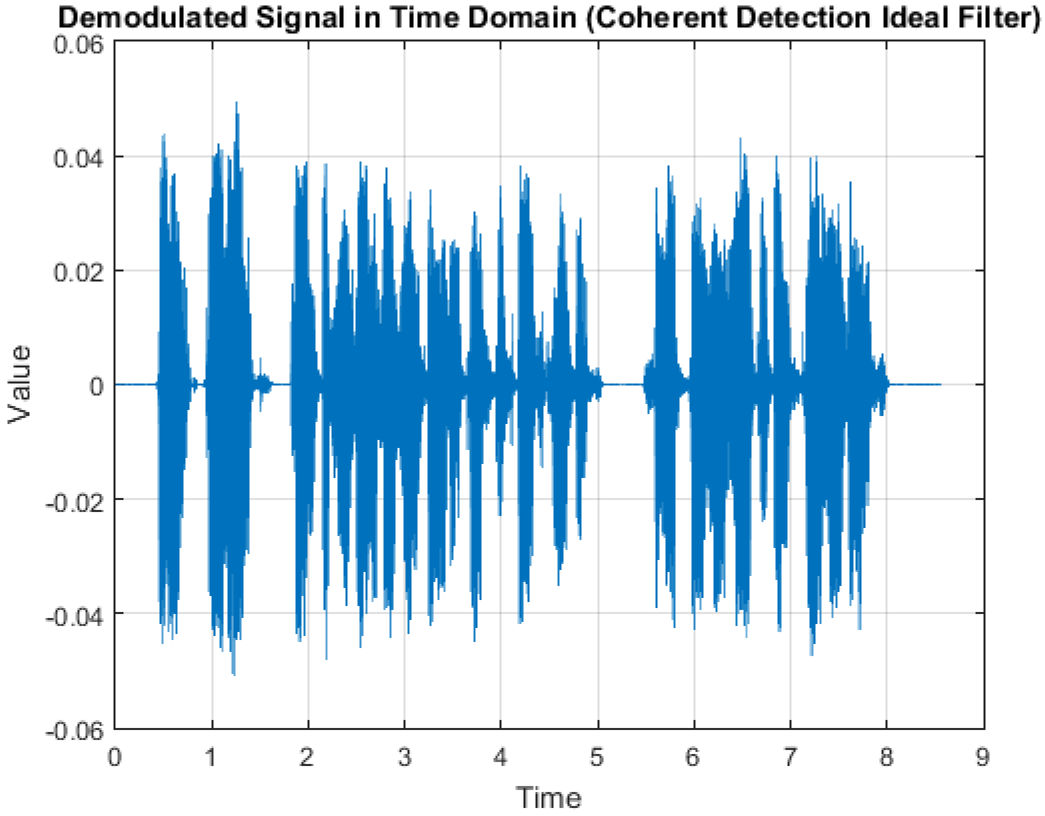

DSB-SC modulated Signal in Time Domain
Value
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
1
2
3
4
5
6
7
8
9
Time

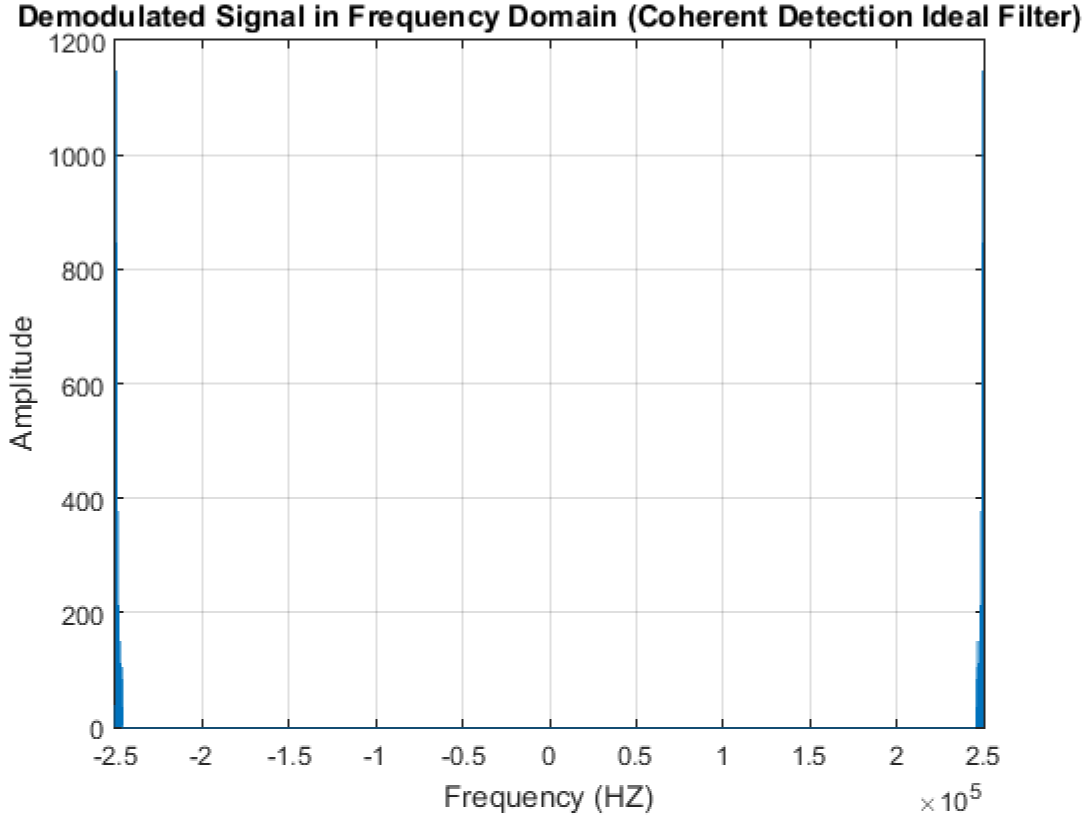

DSB-SC modulated Signal in Frequency Domain
Amplitude
3000
2500
2000
1500
1000
500
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
×10^5


SSB-SC modulated Signal in Time Domain (LSB) Ideal Filter
Value
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1
2
3
4
5
6
7
8
9
Time

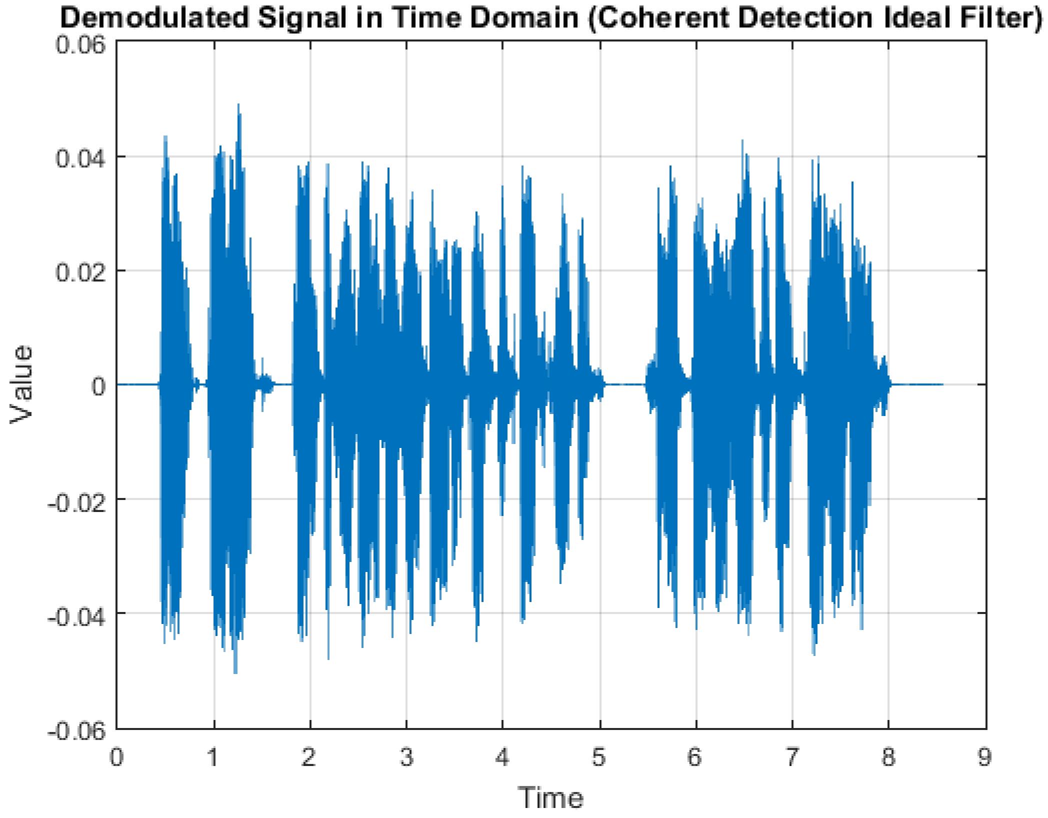

SSB-SC modulated Signal in Frequency Domain (LSB) Ideal Filter
Amplitude
3000
2500
2000
1500
1000
500
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
× 10^5

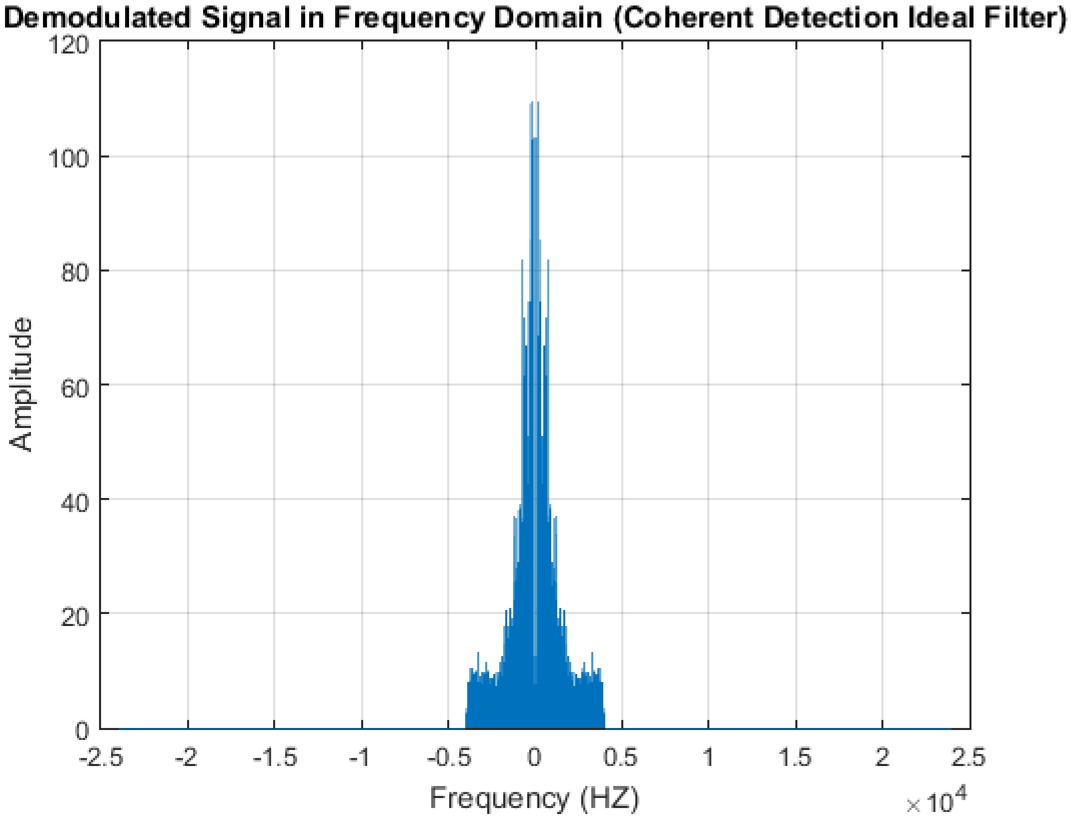
Before Resampling: Fs = 5*Fc

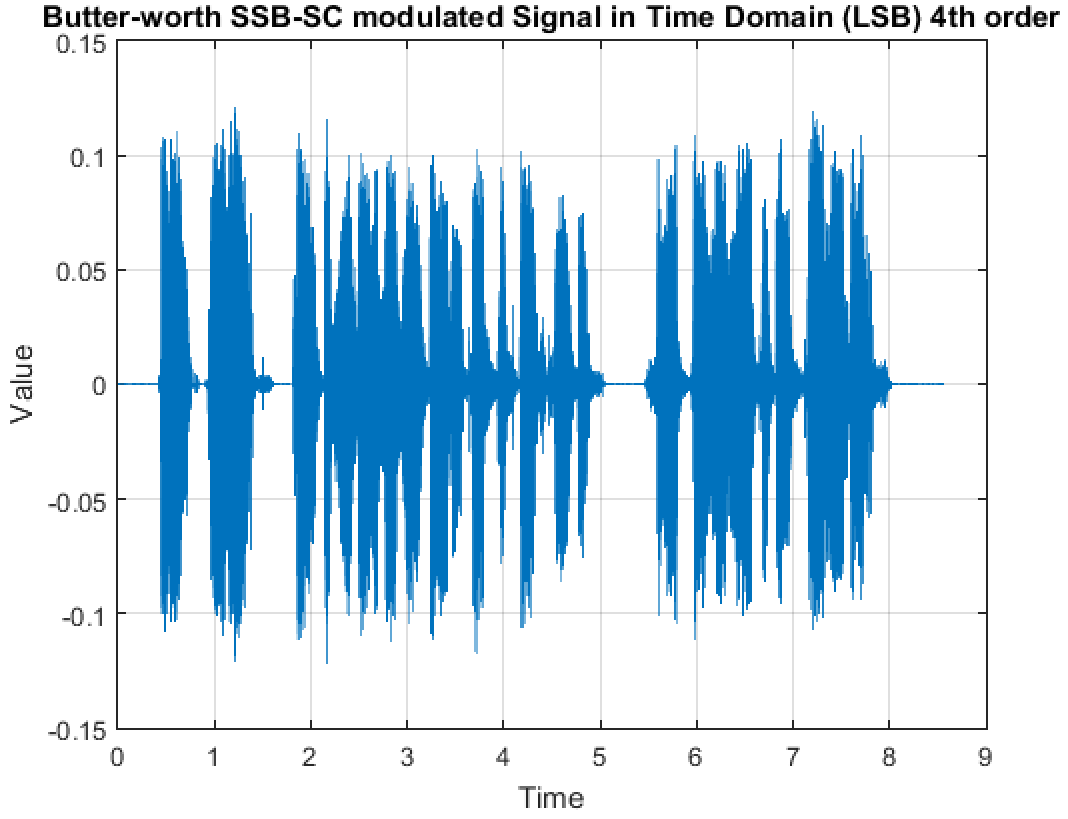


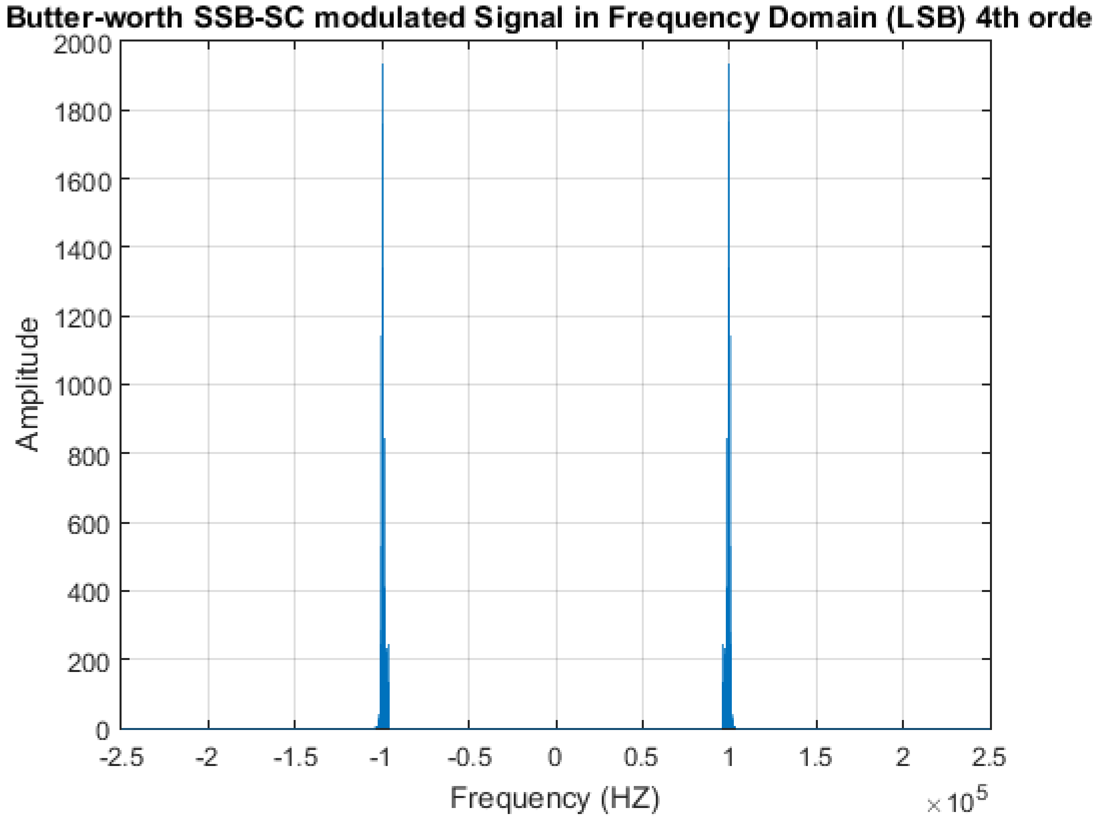


After Resampling: Fs = 48 KHz

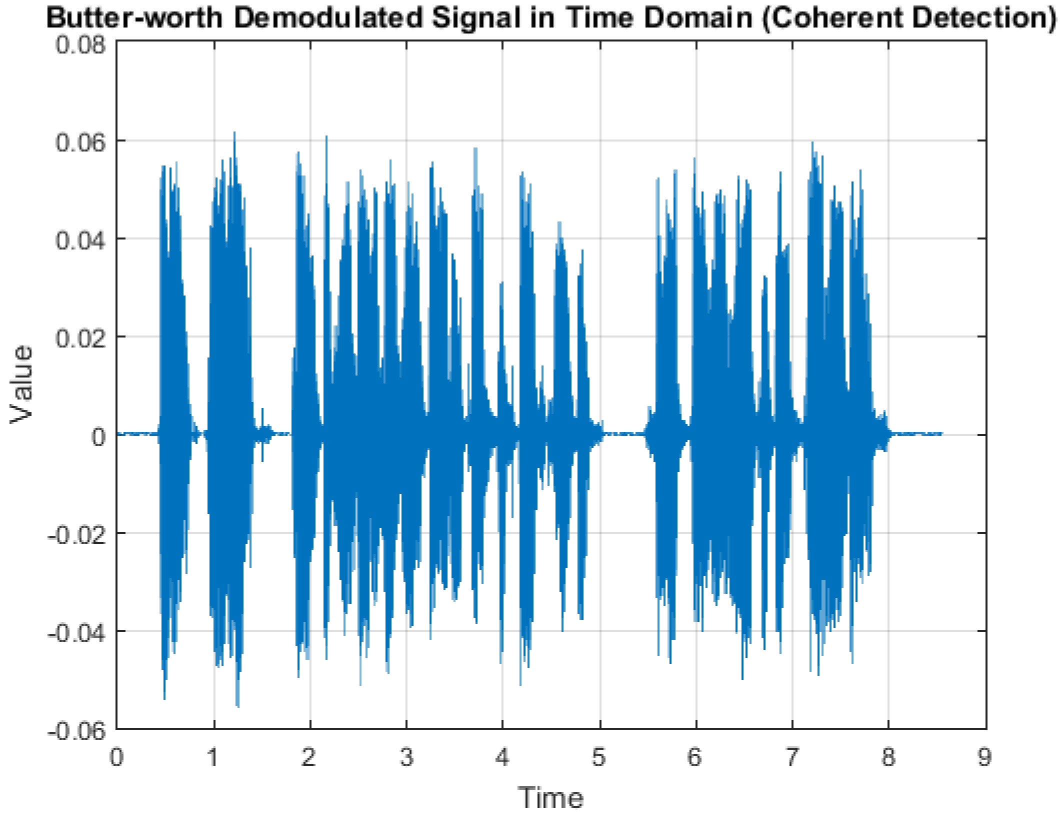


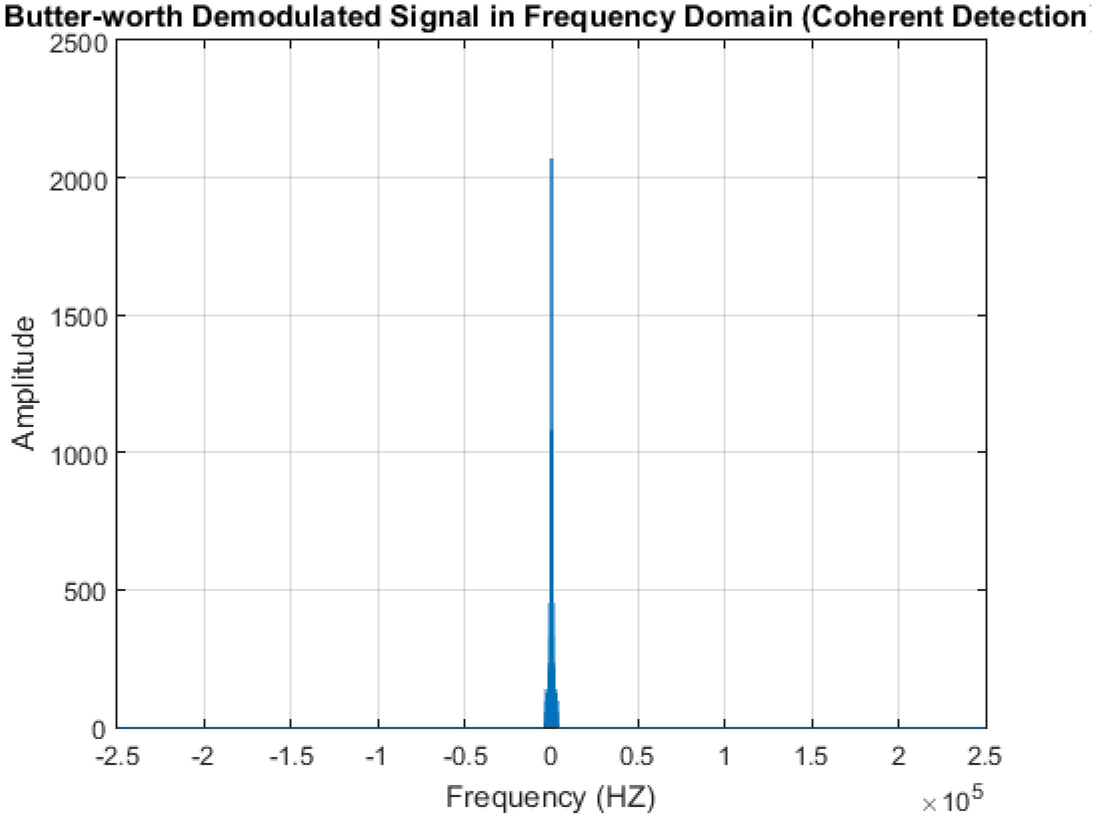



Butter-worth SSB-SC modulated Signal in Time Domain (LSB) 4th order
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
Value
0
1
2
3
4
5
6
7
8
9
Time

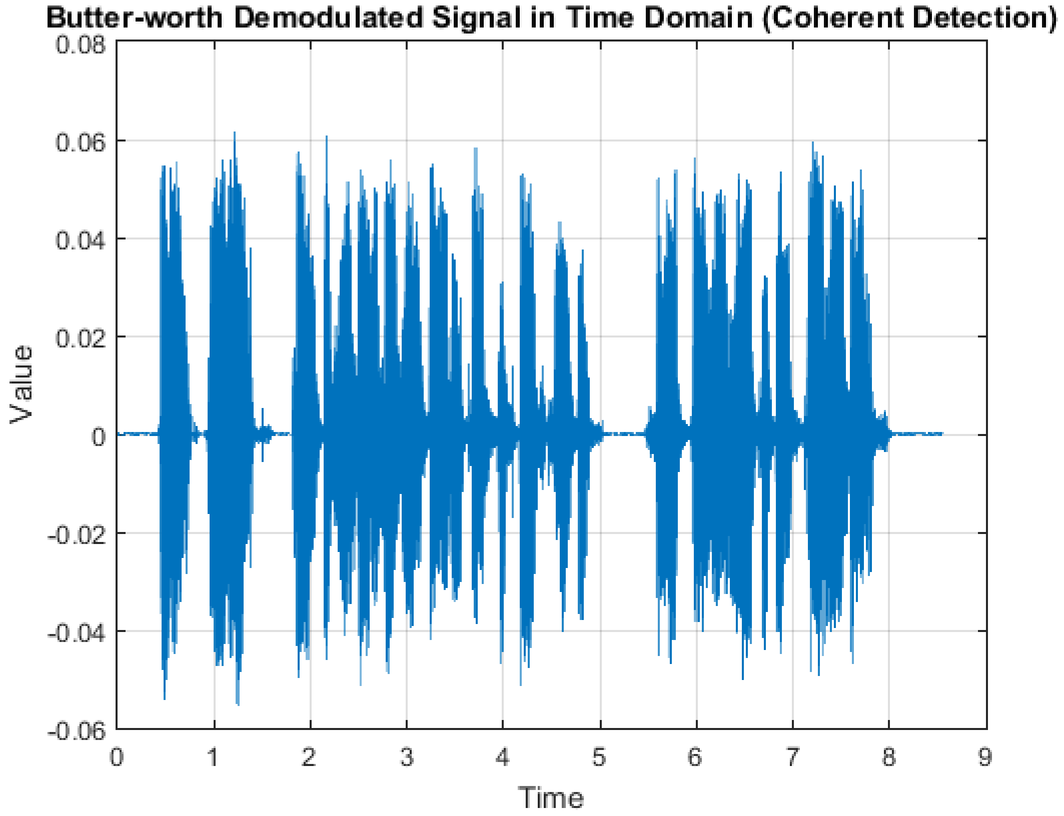

Butter-worth SSB-SC modulated Signal in Frequency Domain (LSB) 4th orde
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
× 10^5

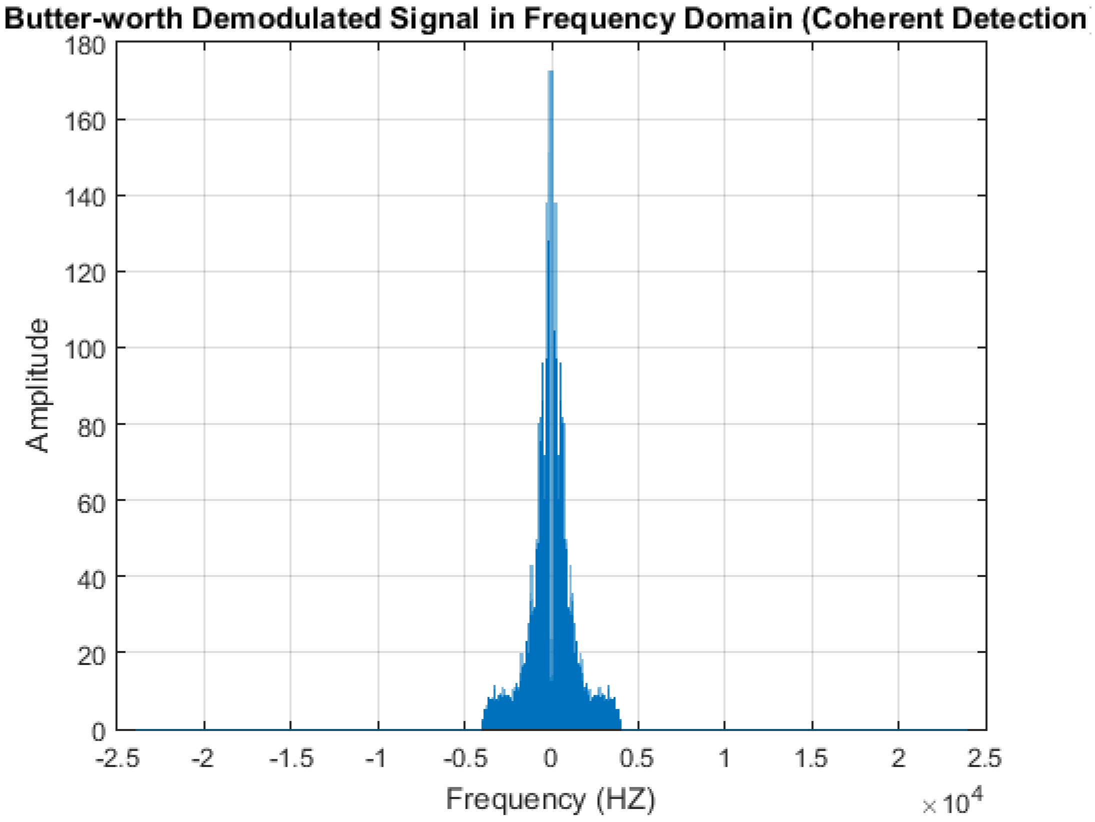
Before Resampling: Fs = 5*Fc



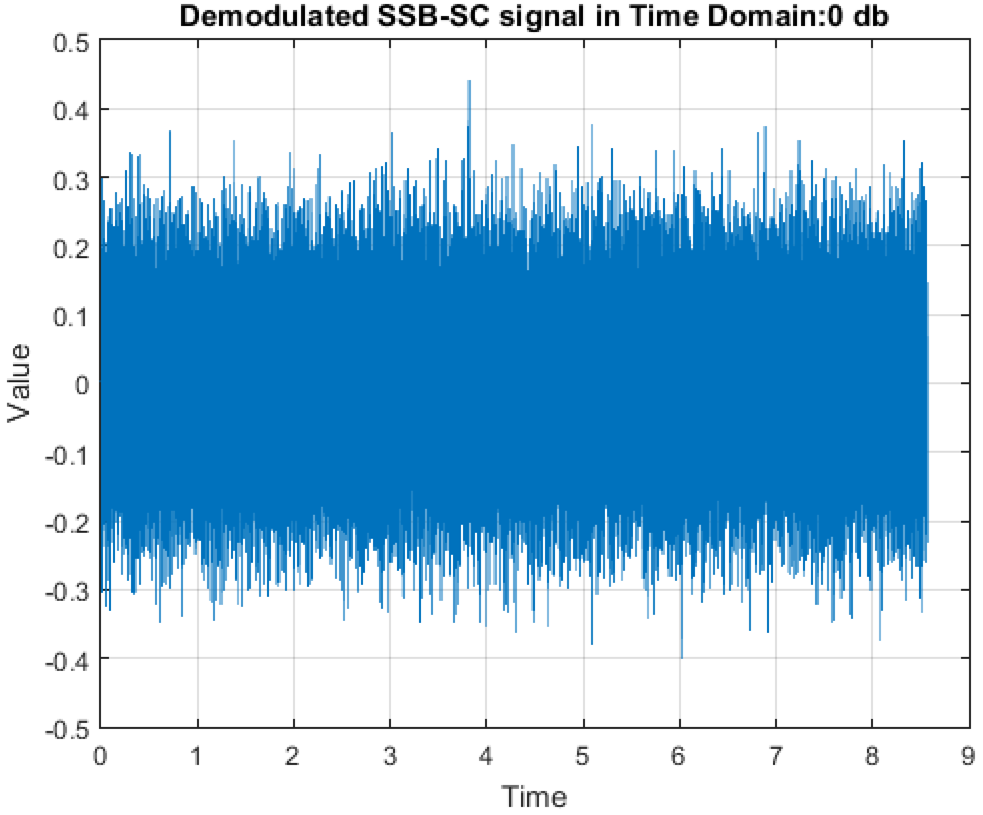


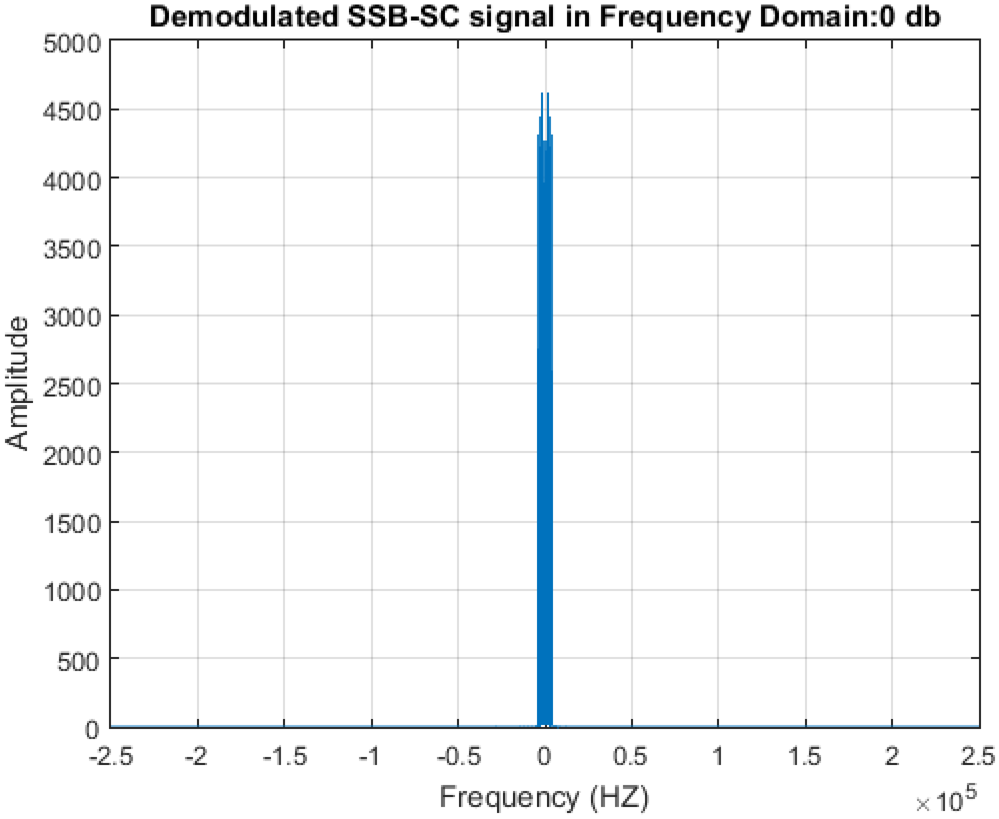
After Resampling: Fs = 48KHz


Butter-worth Demodulated Signal in Time Domain (Coherent Detection)

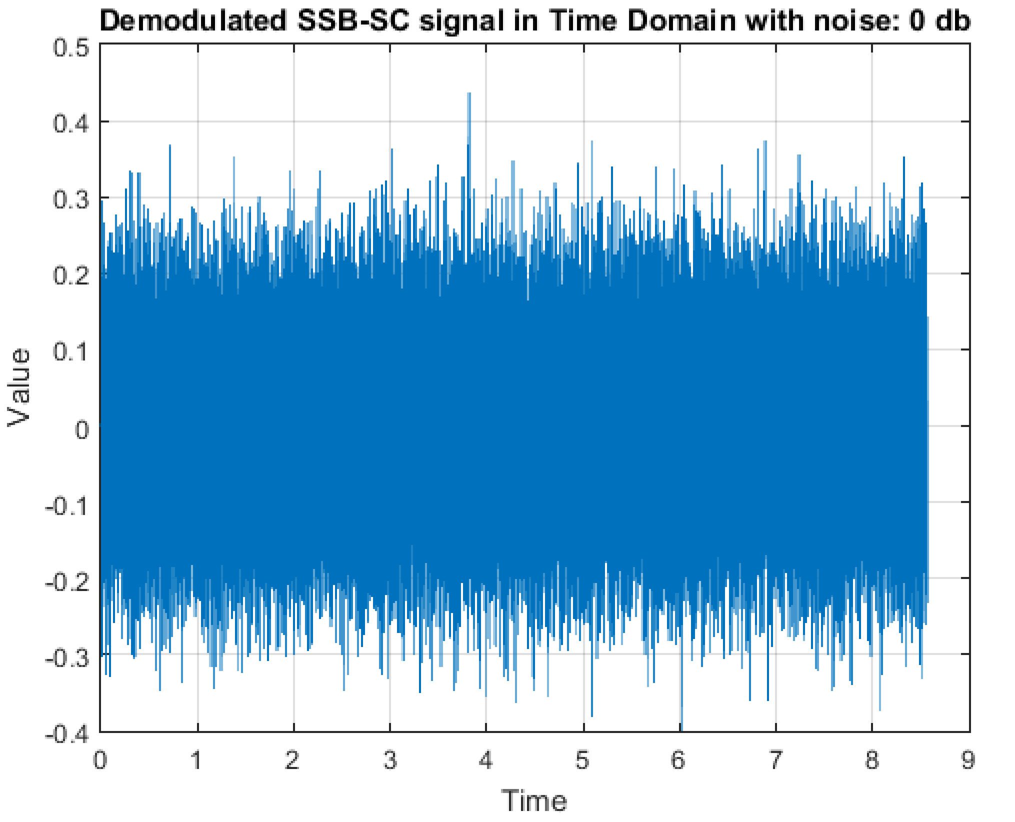

Butter-worth Demodulated Signal in Frequency Domain (Coherent Detection

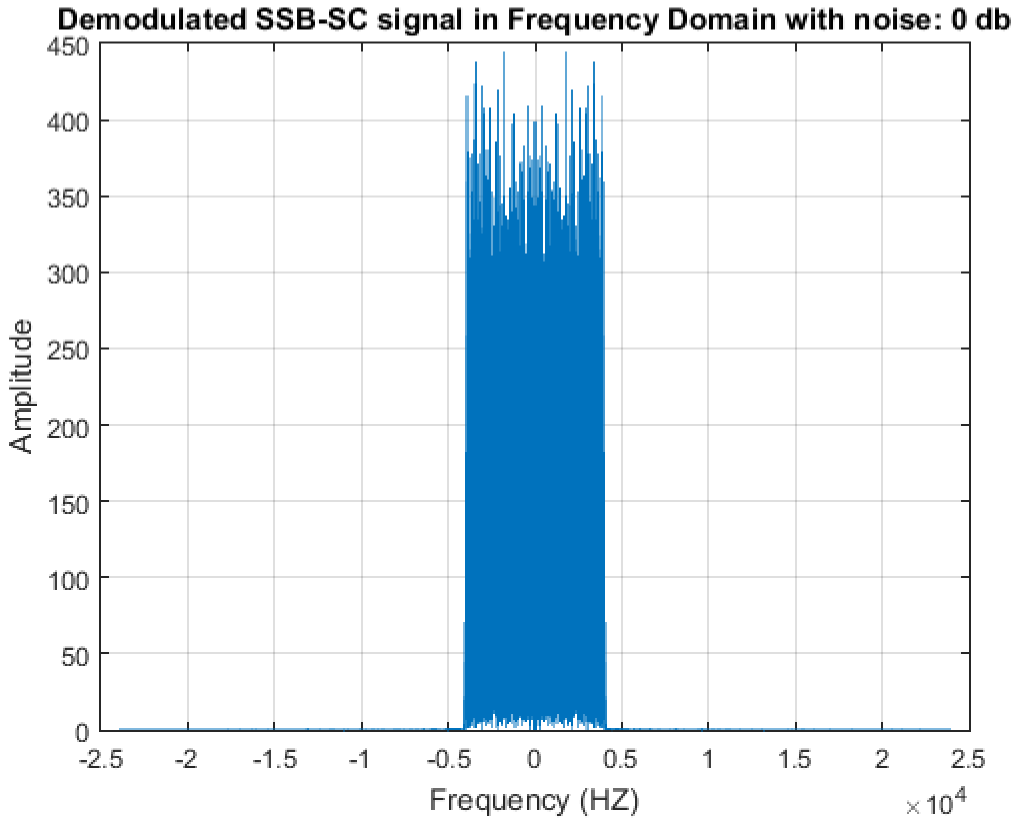
Before Resampling: Fs = 5*Fc



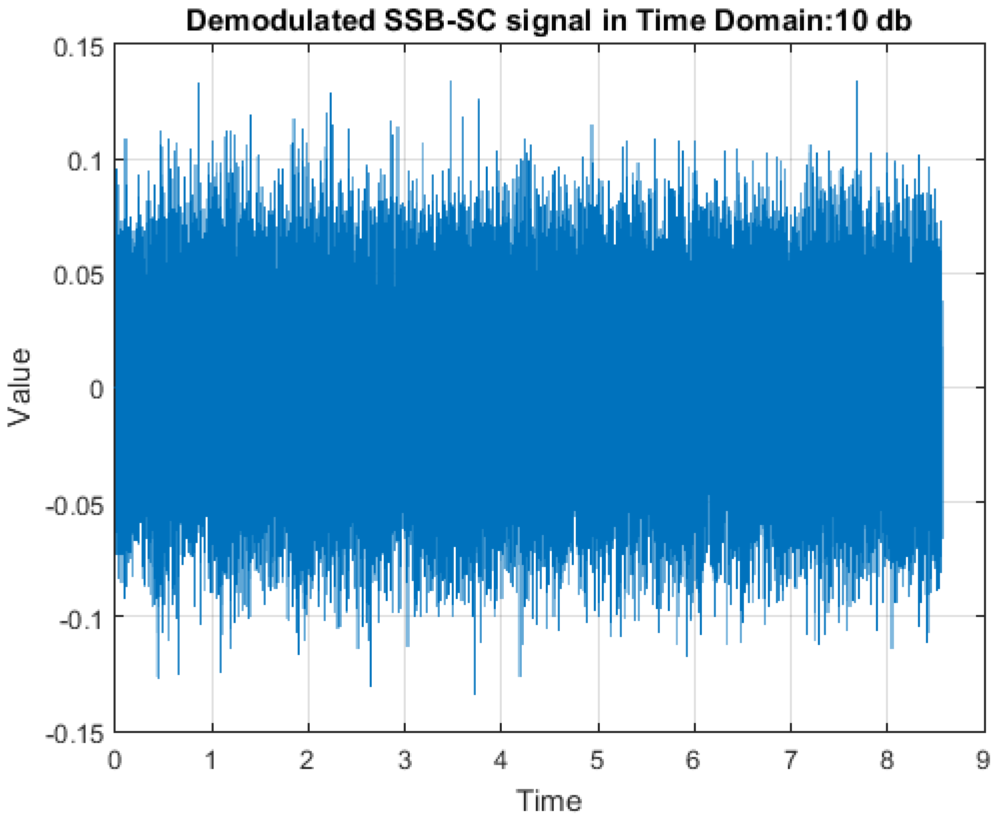


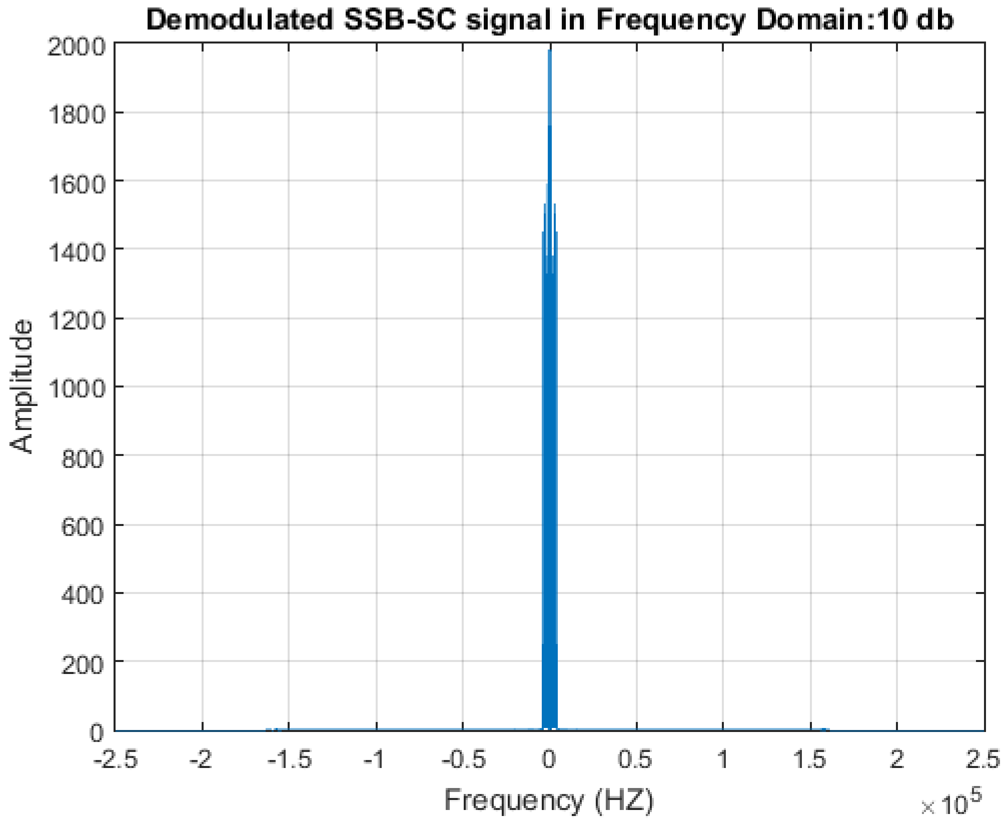
After Resampling: Fs = 48KHz



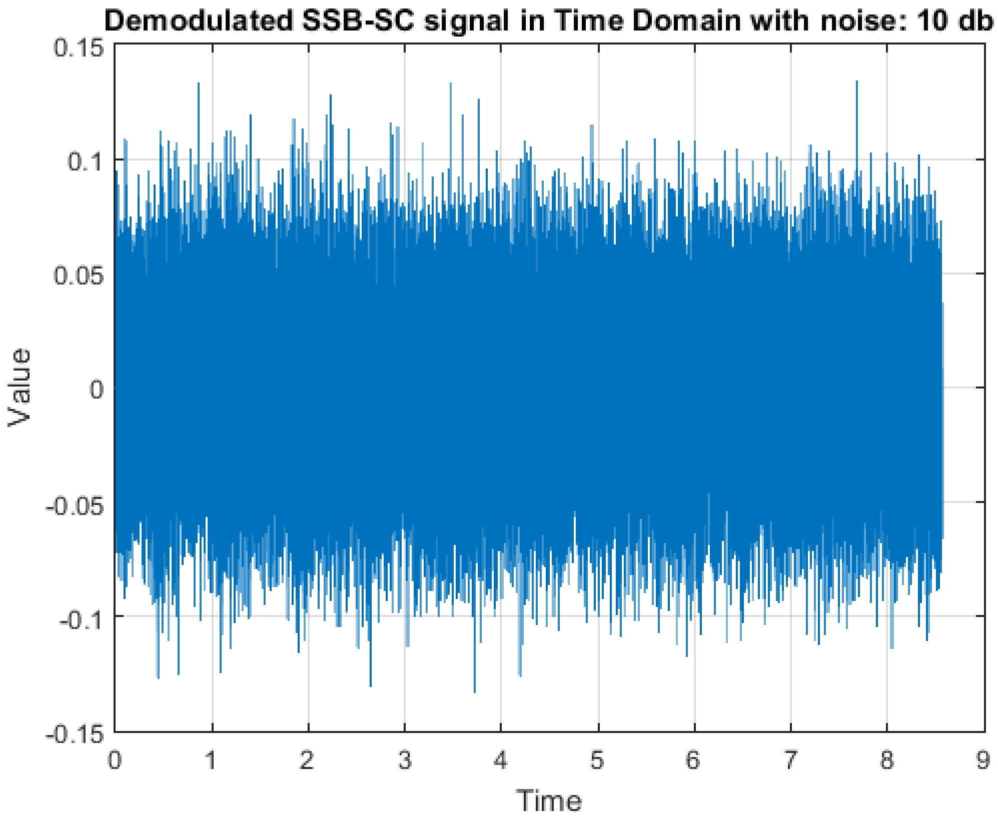


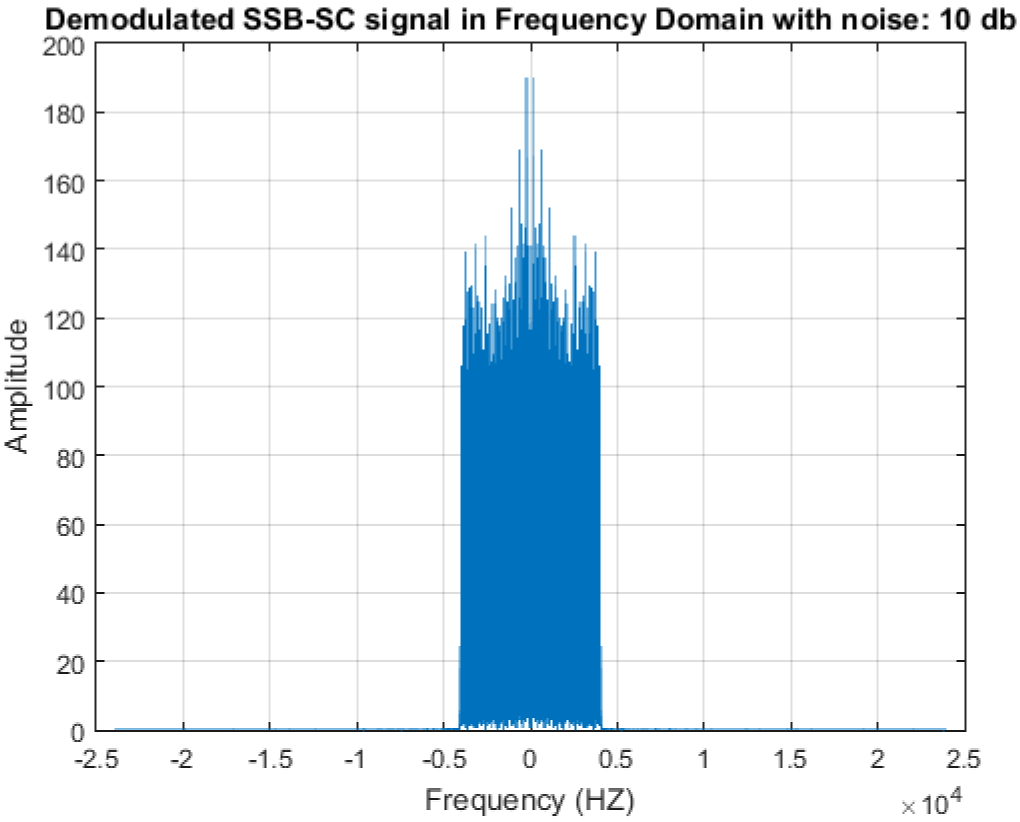
Before Resampling: Fs = 5*Fc


Demodulated SSB-SC signal in Time Domain:10 db

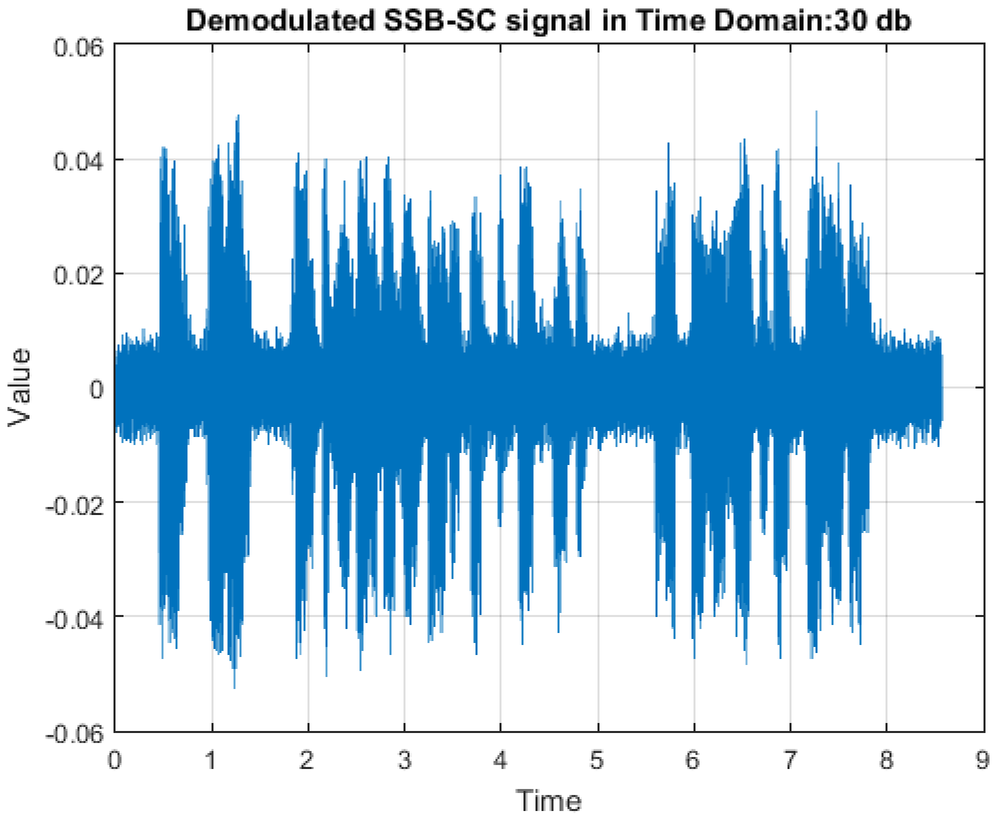

Demodulated SSB-SC signal in Frequency Domain:10 db

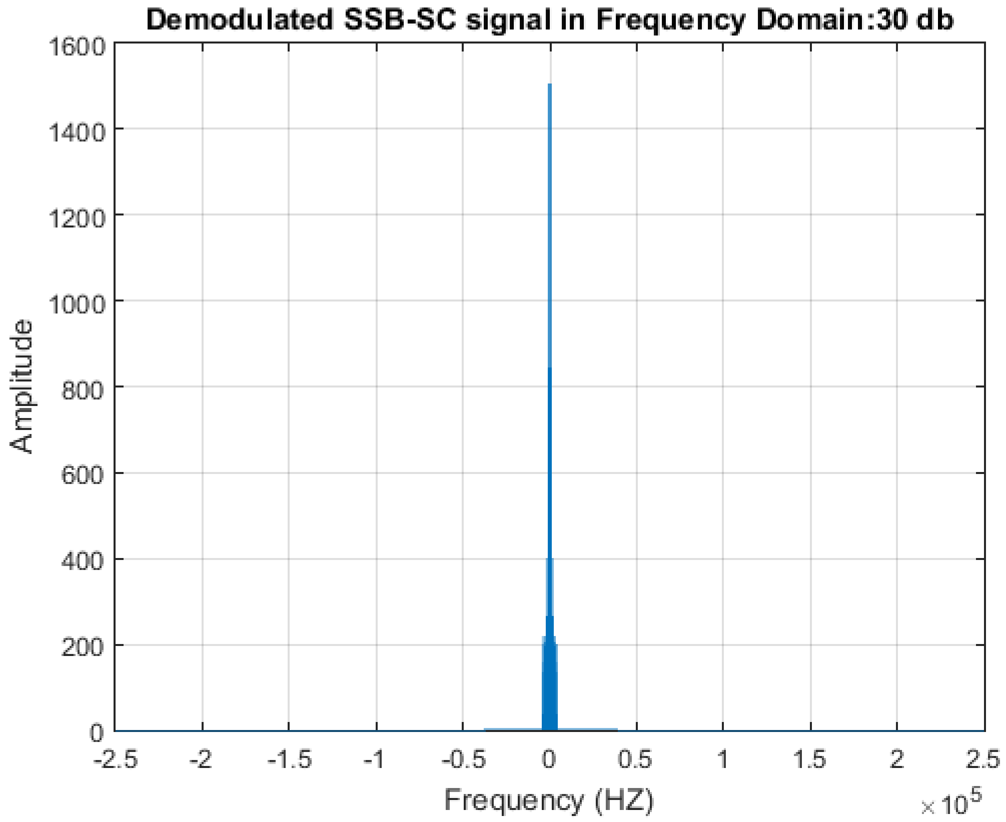
After Resampling: Fs = 48KHz



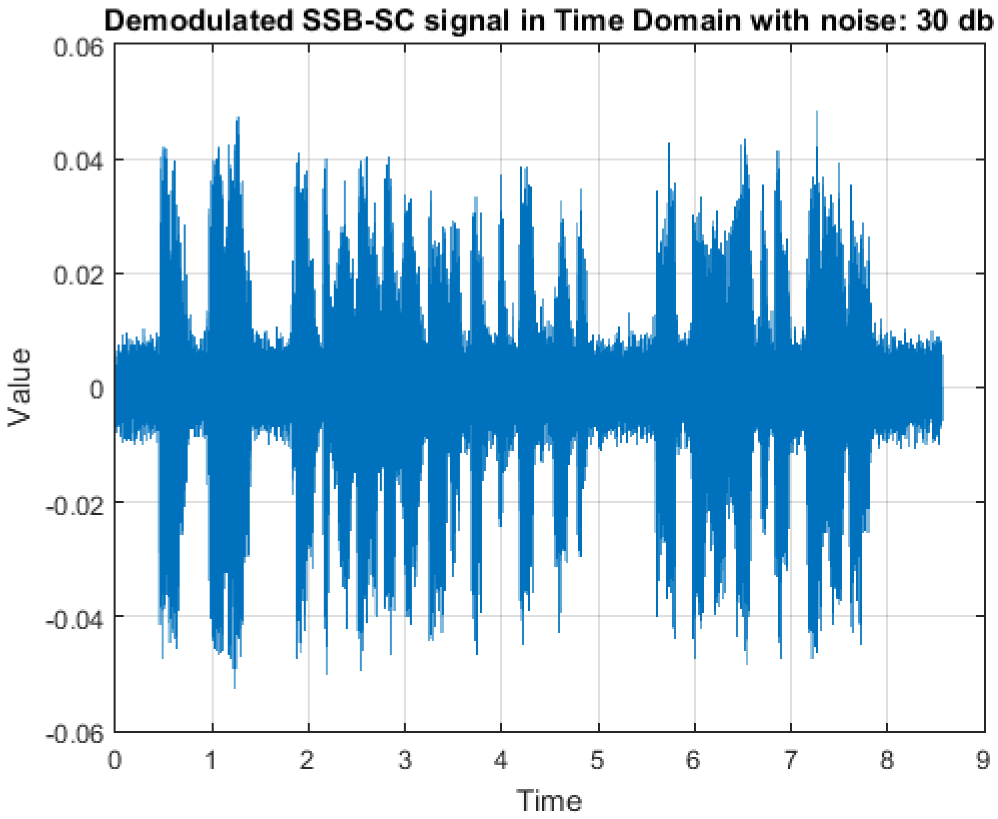


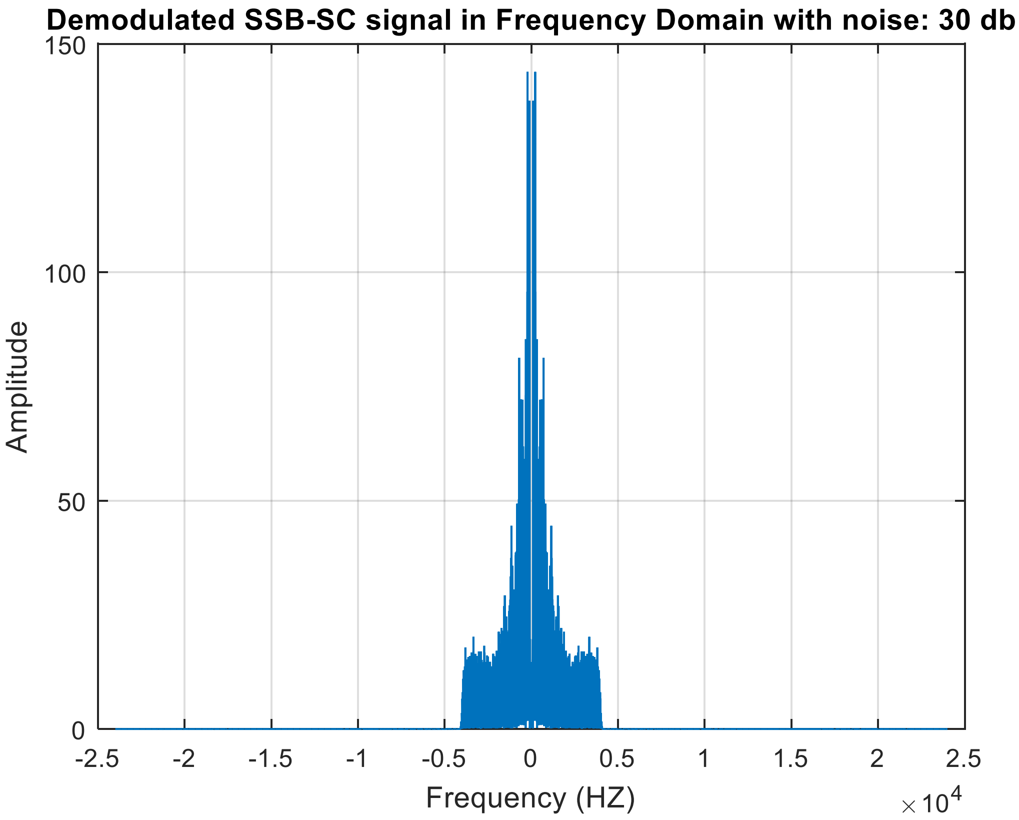
Before Resampling: Fs = 5*Fc

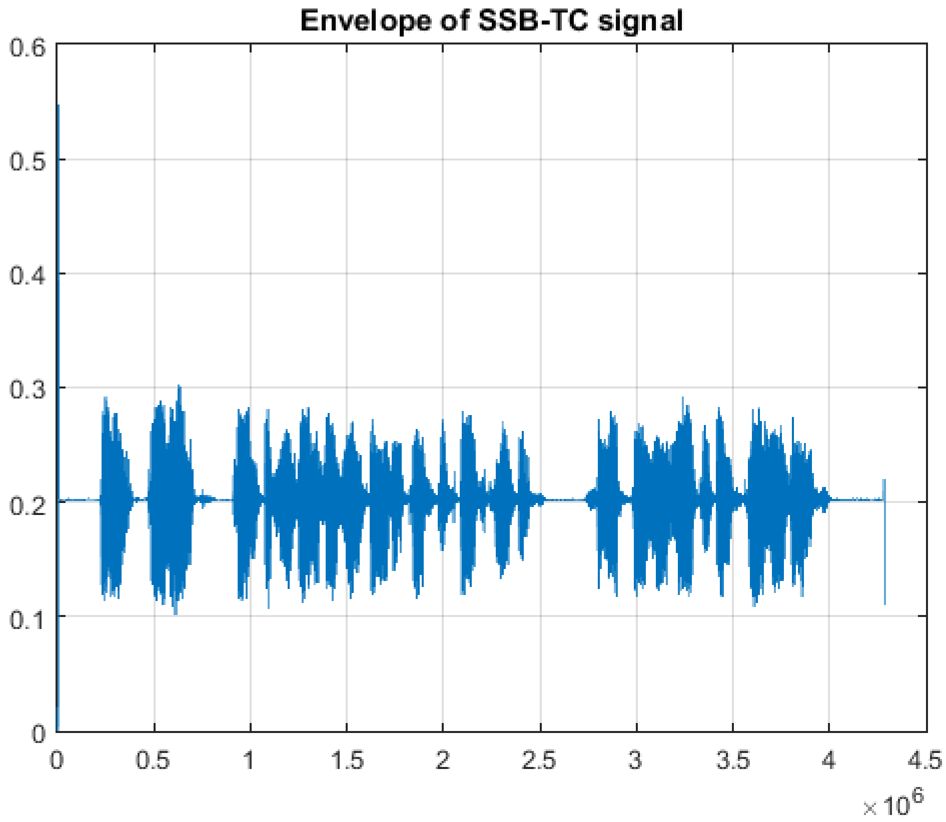

Demodulated SSB-SC signal in Time Domain:30 db

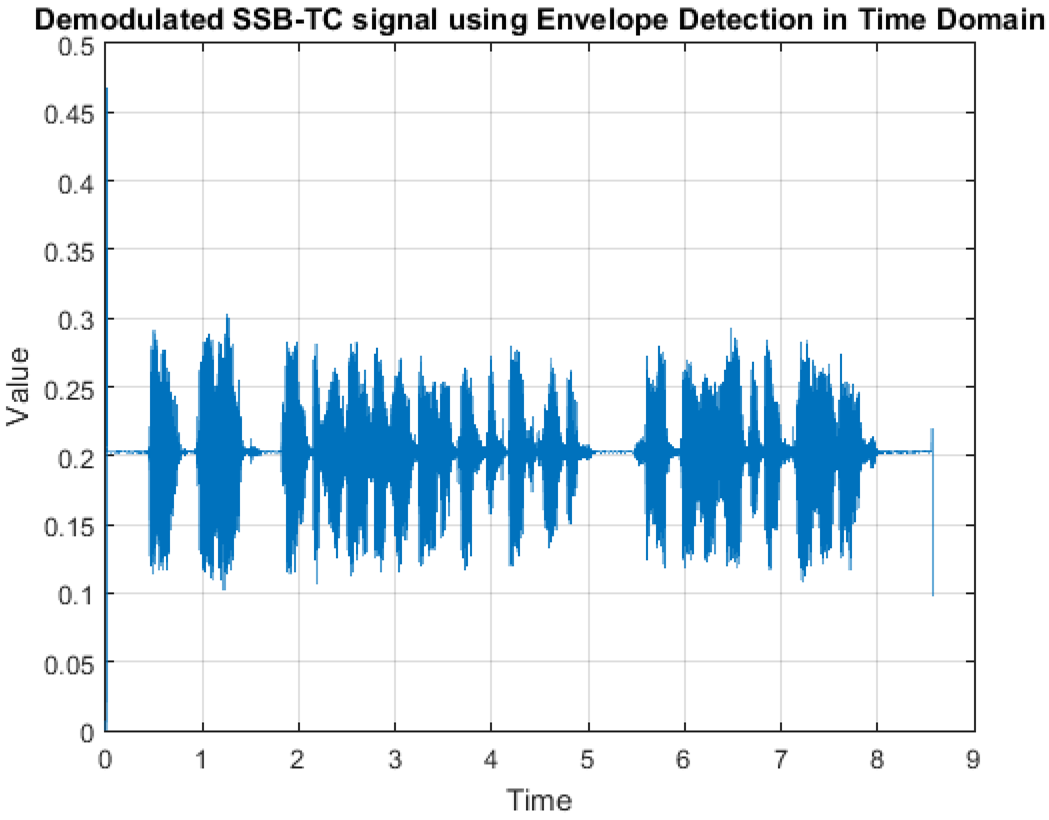

Demodulated SSB-SC signal in Frequency Domain:30 db

After Resampling: Fs = 48KHz


Demodulated SSB-SC signal in Time Domain with noise: 30 db


Demodulated SSB-SC signal in Frequency Domain with noise: 30 db


Envelope of SSB-TC signal
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
× 10^6


Demodulated SSB-TC signal using Envelope Detection in Time Domain
Value
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
4
5
6
7
8
9
Time

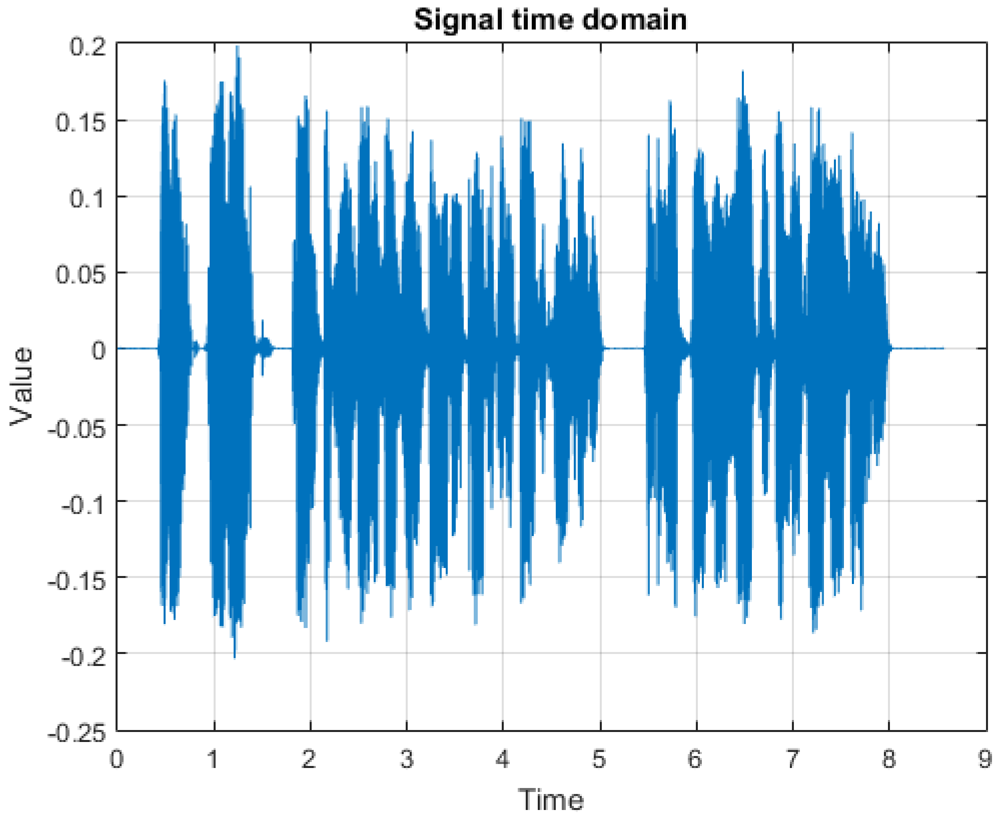
EXPERIMENT 3

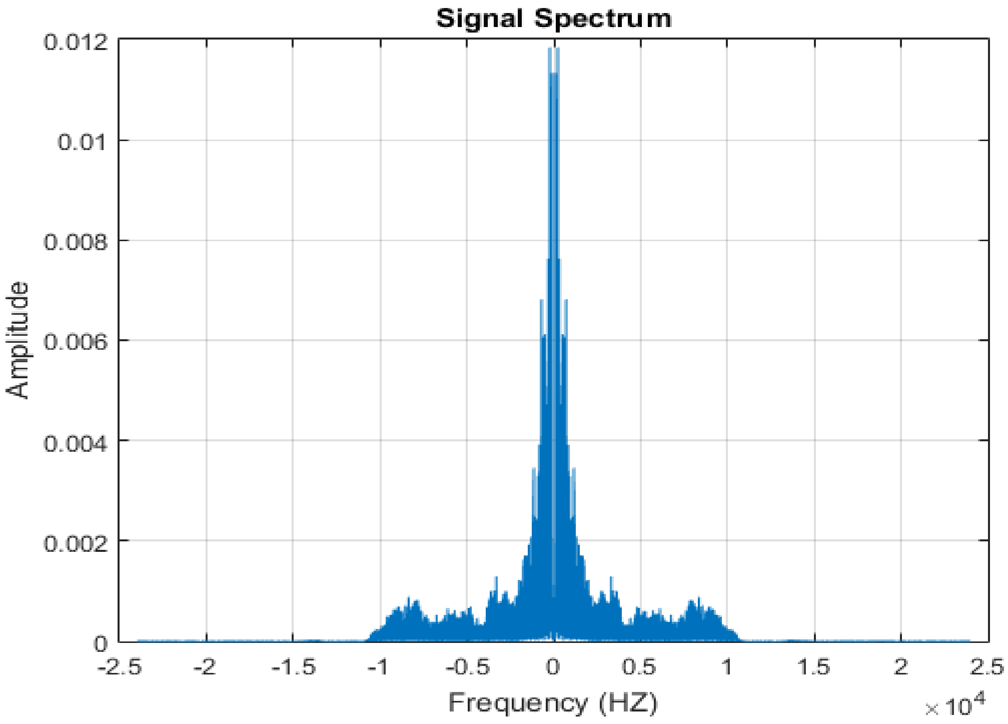
Frequency Modulation

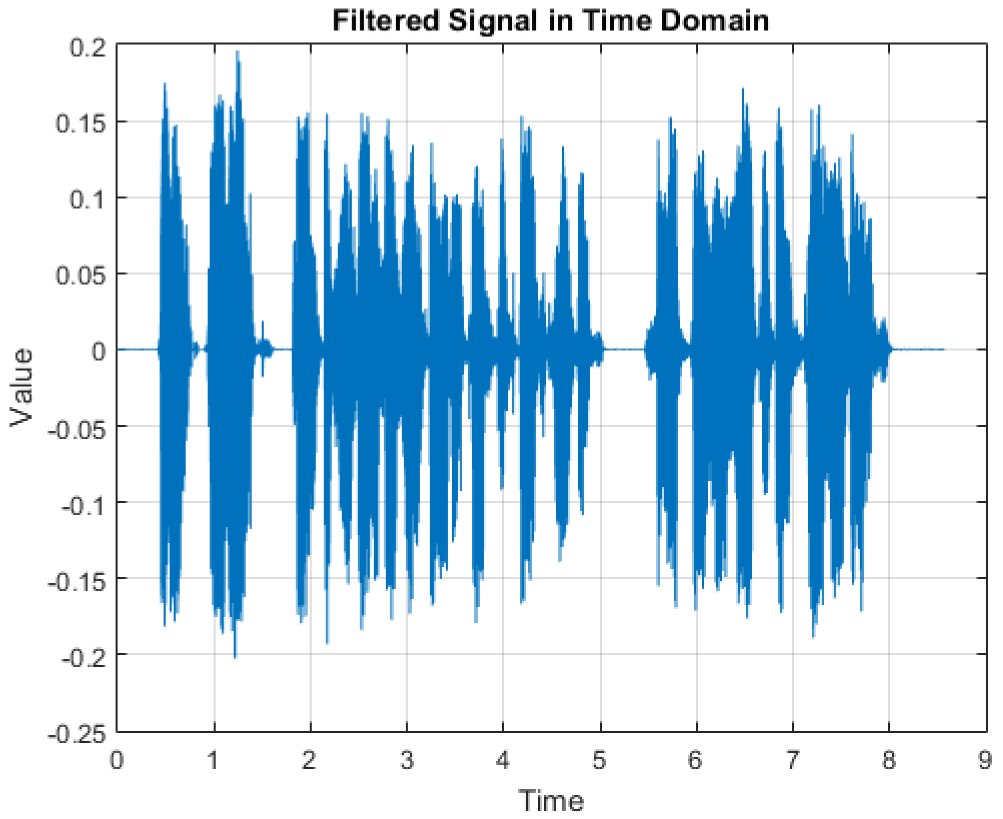
Code :

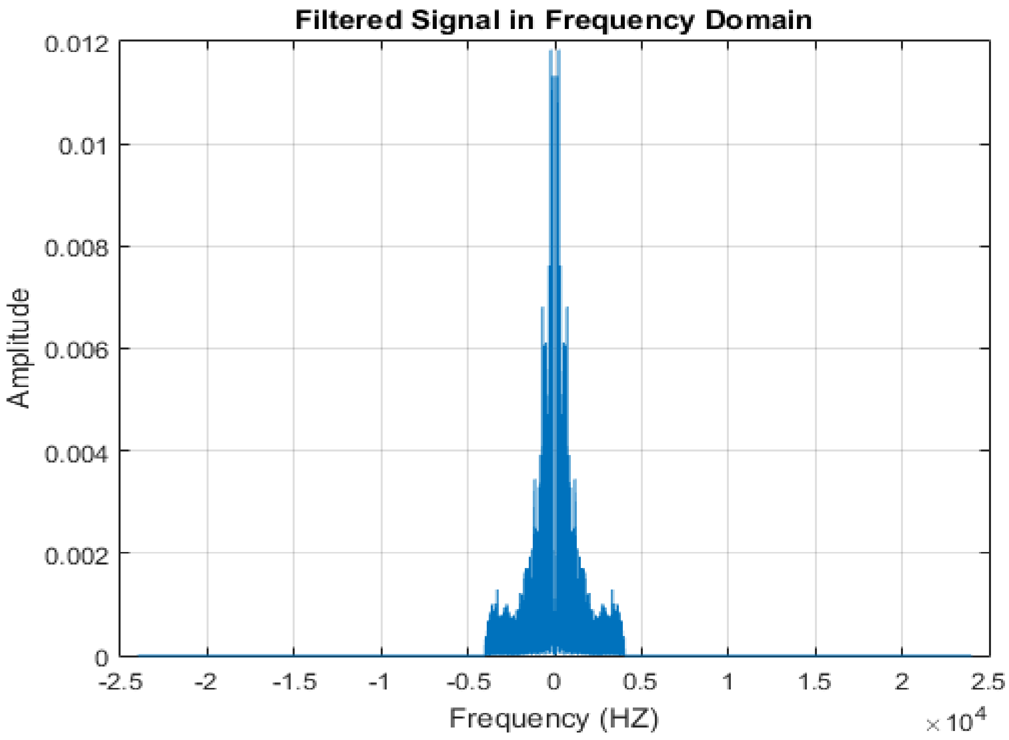
```
%%%%%%%%%question1%%%%%%%%%%

[y , fs] = audioread('eric.wav');
% sound(y,fs);
plot_time(y,fs,'Signal time domain');
y_Spectrum = (fftshift(fft(y)));
plot_frequency(abs(y_Spectrum),fs,'Signal Spectrum');
f = linspace(-fs/2,fs/2,length(y));

filter = generate_filter(length(y_Spectrum),fs);
filteredSpectrum = y_Spectrum.* filter ;
ytime_filtered= real(ifft(ifftshift(filteredSpectrum)));
plot_frequency(real(filter),fs,'Filter');
plot_time(ytime_filtered,fs,'Filtered Signal in Time Domain');
plot_frequency(abs(filteredSpectrum),fs,'Filtered Signal in Frequency Domain');
%sound(ytime_filtered,fs);

%%%%%%%%question2%%%%%%%%%%

fc = 100000;
FS = 5*fc;
resampledSignal = resample(ytime_filtered,FS,fs);
t = linspace(0,length(resampledSignal)/FS,length(resampledSignal));
A=max(abs(resampledSignal));
Kf= 0.2/(2*pi* max(cumsum(resampledSignal)*(1/FS)));
delta= Kf.*cumsum(resampledSignal).';
NBFM_time= A.*cos(2* pi*fc*t  + delta);
plot_time(NBFM_time, FS, 'NBFM modulated signal in Time Domain');
NBFM_frequency=abs(fftshift(fft(NBFM_time)));
plot_frequency(NBFM_frequency, FS, 'NBFM modulated signal in Frequency Domain');

%%%%%%%question3%%%%%%%%%%
%beta<<1

%%%%%%%question4%%%%%%%%%%

signalDiff_AM=diff(NBFM_time);
signal_envelope=abs(hilbert(signalDiff_AM));
signal_envelopeResampled=resample(signal_envelope,fs,FS);
plot_time(signal_envelopeResampled, fs, 'NBFM demodulated signal in Time Domain using envelope');
signal_envelopeResampledFreq=abs(fftshift(fft(signal_envelopeResampled)));
plot_frequency(signal_envelopeResampledFreq, fs, 'NBFM demodulated signal in Frequency Domain
using envelope');
% sound(signal_envelopeResampled,fs);
```

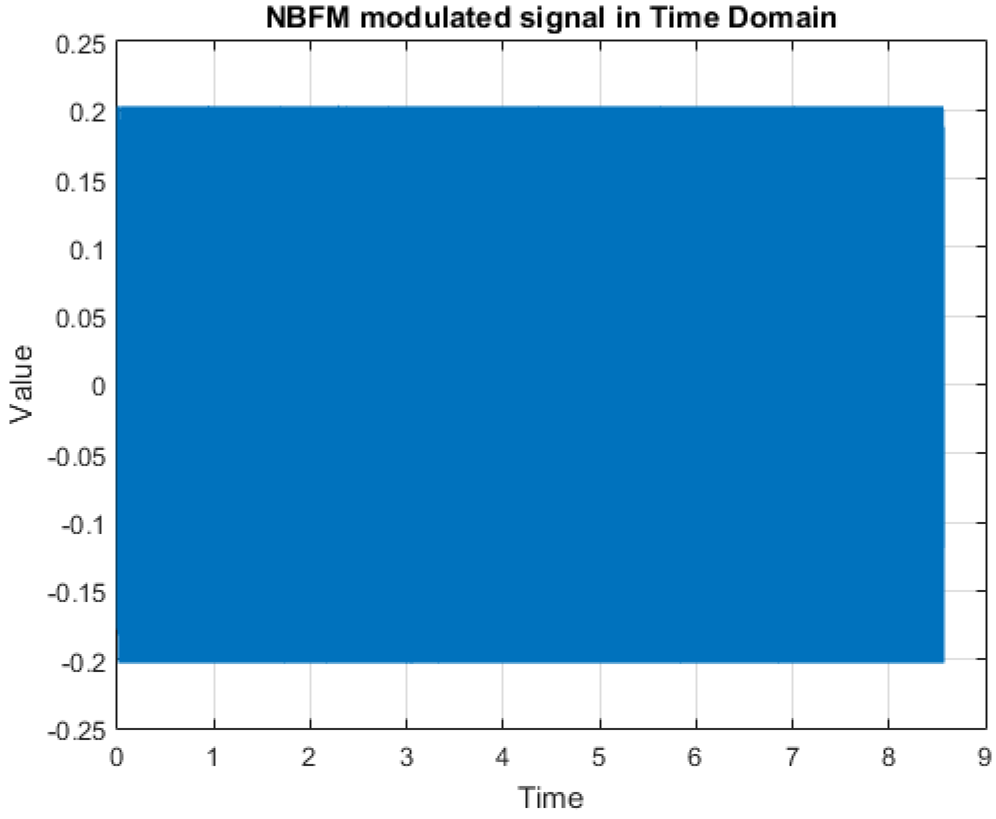
Figures:

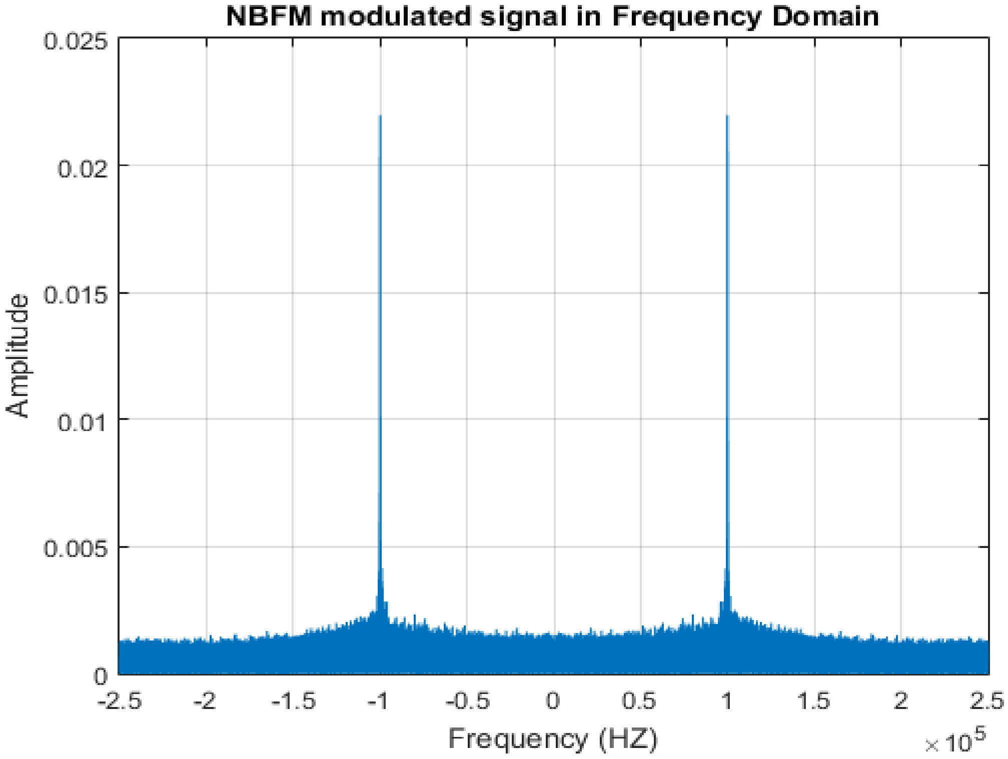

Signal time domain
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9

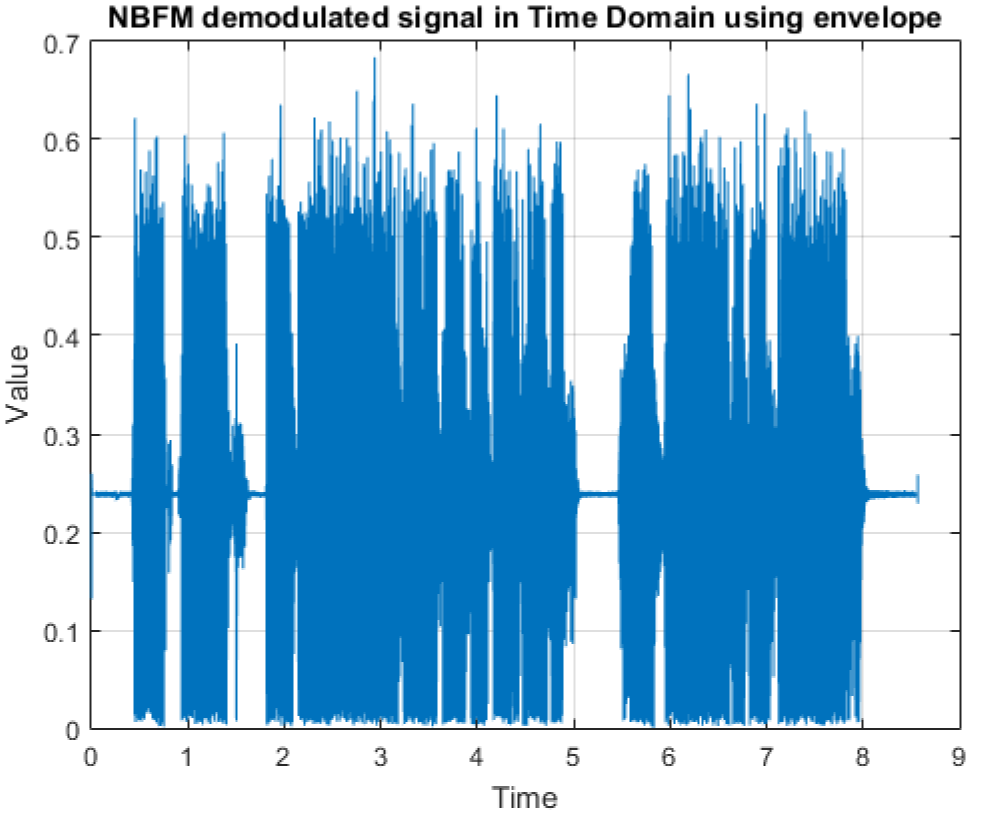

Signal Spectrum
Amplitude
Frequency (HZ)
0.012
0.01
0.008
0.006
0.004
0.002
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
× 10^4

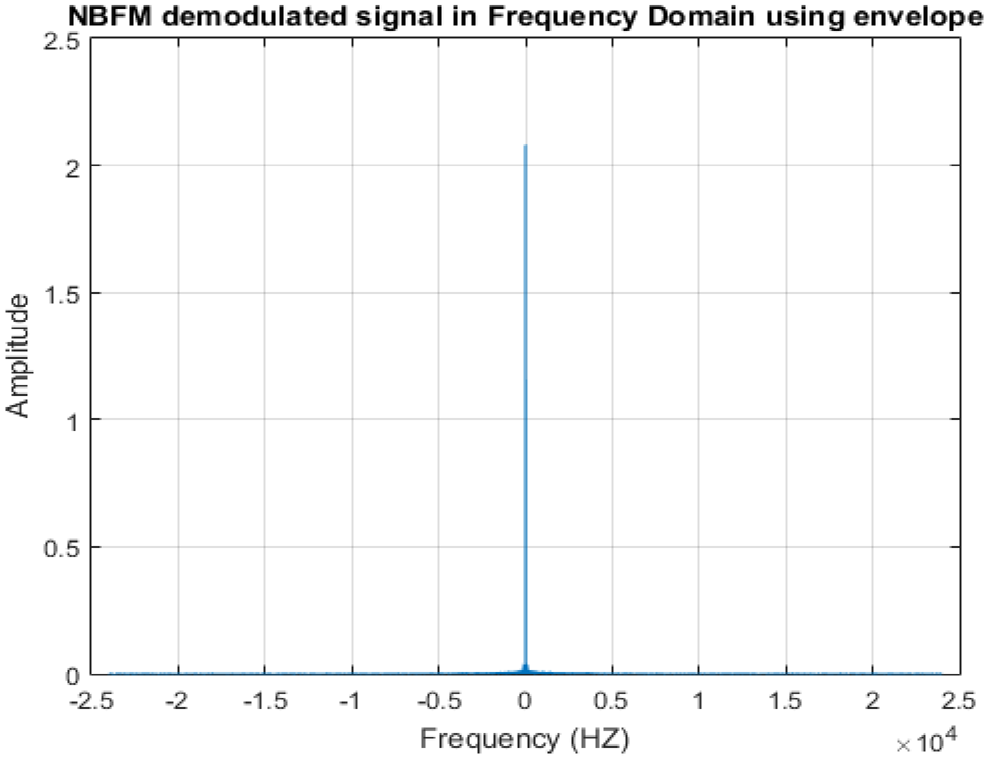

Filtered Signal in Time Domain
Value
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9


Filtered Signal in Frequency Domain
Amplitude
Frequency (HZ)
0.012
0.01
0.008
0.006
0.004
0.002
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
× 10^4


NBFM modulated signal in Time Domain
Value
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9
Time


NBFM modulated signal in Frequency Domain
Amplitude
0.025
0.02
0.015
0.01
0.005
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequency (HZ)
× 10^5


NBFM demodulated signal in Time Domain using envelope
Value
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
1
2
3
4
5
6
7
8
9


NBFM demodulated signal in Frequency Domain using envelope
Amplitude
Frequency (HZ)
0
0.5
1
1.5
2
2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^4

Theoretical Questions:

Question 2: What can you make out of the resulting plot?

- ➔ We obtain from the resulting plot of the NBFM spectrum that it is just like that of the DSB-TC.

Question 9: What is the condition we needed to achieve NBFM?

- ➔ Beta must be much smaller than 1 (beta<<1).

Functions Used

```
function filter = generate_filter(signal_length,fs)
    filter = ones(signal_length,1);
    f = linspace(-fs/2,fs/2,signal_length);
    for i = 1: signal_length
        if abs(f(i))>4000
            filter(i)=0;
        end
    end
end
```

```
function  plot_time(y,fs,title_label)
    t = linspace(0, length(y)/fs, length(y));
    figure();
    plot(t,y);
    title(title_label);
    xlabel('Time');
    ylabel('Value');
    grid;
end
```

```
function  plot_frequency(y,fs,title_label)
    f = linspace(-fs/2,fs/2,length(y));
    figure()
    plot(f,y);
    title(title_label);
    xlabel('Frequency (HZ)');
    ylabel('Amplitude');
    grid;
end
```