Analog Matlab Report

Adham Mohamed Ahmed 19017255

Abdallah Ashraf Elsayed 19017266

Ahmed Elsayed Abdellah 19015193

Ragai Ahmed Abdelfatah 19015655

Ibrahim Abdelwahab Mohamed 19015169

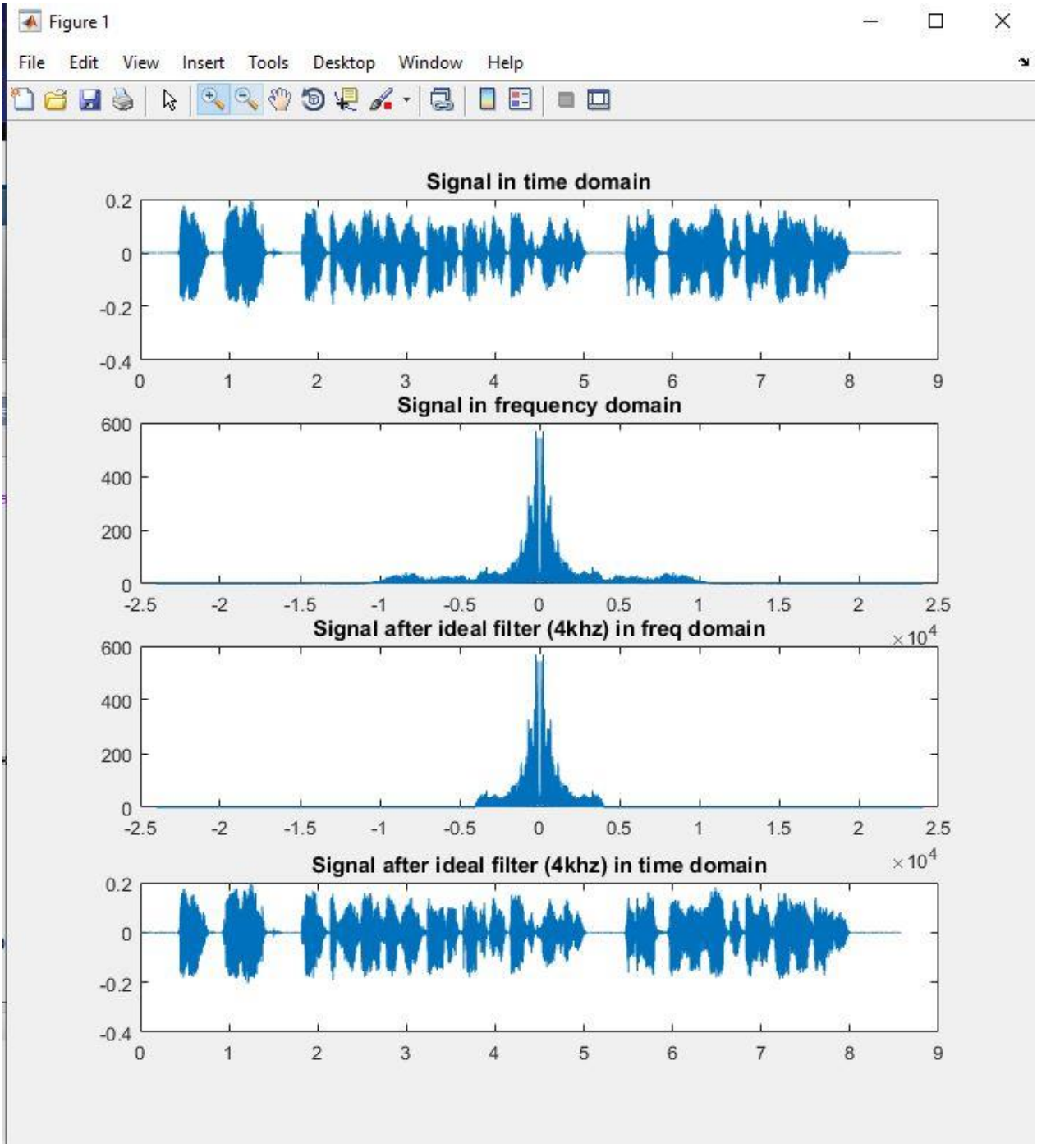
Experiment 1

Preparing the signal:

```
[xin ,fs] = audioread('eric.wav');

%sound(xin,fs);

% Extract only one channel if the audio is stereo

x = xin(:,1);

% Create time axis

t = linspace(0,length(x)/fs, length(x));

%plot in time domain

figure(1)

subplot(4,1,1)

plot(t,xin)

title('Original signal in Time domain')



%fft

X = fftshift(fft(x));

X_ABS = abs(X);

W = angle(X);

N = length(x);

F = linspace(-fs/2,fs/2,N);



%plot in frequency domain
```

```
subplot(4,1,2)

plot(F,X_ABS)

title('Original signal in Frequency domain')

%--ideal LPF at 4000hz--%

N = length(x);

n = N/fs;

r_limit = round((fs/2-4000)*n);

l_limit = (N-r_limit+1);

X([1:r_limit l_limit:N]) = 0;

X_ABS = abs(X);

%plot in frequency domain

subplot(4,1,3)

plot(F,X_ABS)

title('LPF(4khz) Signal in Frequency domain')



x = real(ifft(ifftshift(X)));

%plot in time domain

subplot(4,1,4)

plot(t,xin)

title('LPF(4khz) Signal in Time domain')



%sound(x,fs);
```

```
%audiowrite('C:\Users\Ragai\Desktop\Analog Project\Original audio filtered.wav',x,fs);
```



- After reading the audio file, we used an ideal 4khz LPF in the time domain and plotted it.

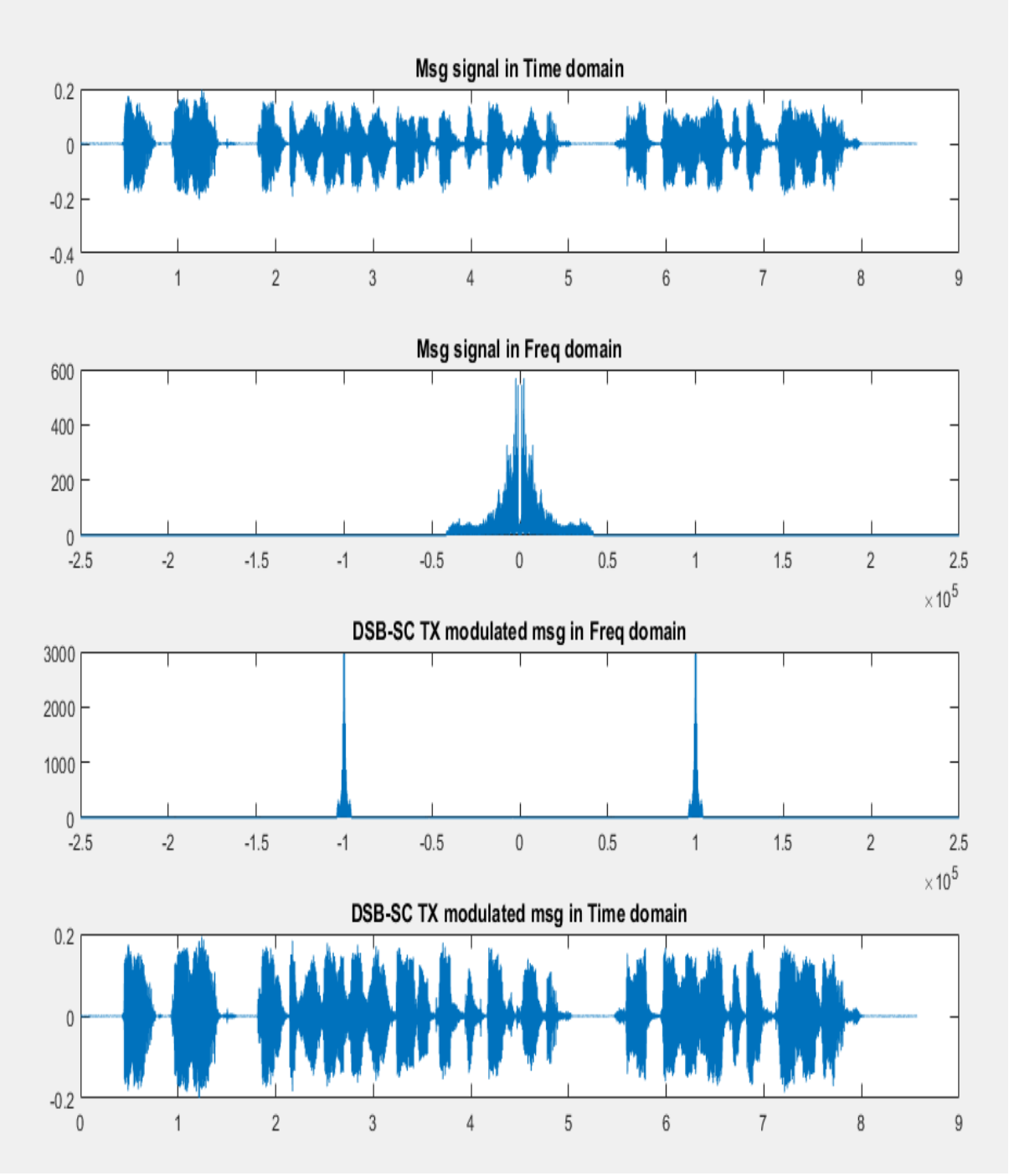
Computing the Double Side Band – Supressed Carrier transmitted signal:

```
%%%%%%%%%DSB-SC%%%%%%%%%%%%%%%%

fc = 100000;  %Carrier frequency

new_fs = 5*fc;  % new sample rate for message



% resampling message signal

msg_resampled = resample(x,new_fs,fs);



% Creating Time axis for resampled signal

t_end = length(msg_resampled)./new_fs;

t = linspace(0,t_end, length(msg_resampled));



% modulating

carrier = cos(2*pi*fc*t);

carrier = carrier';  % transpose to be a row matrix

% Shifting the signal to fc and -fc

send_signal = msg_resampled.*carrier;  % Multiplication in time domain



% freq domain

SEND_SIGNAL = fftshift(fft(send_signal));

SEND_SIGNAL_ABS = abs(SEND_SIGNAL);
```

```
N = length(send_signal);

F = linspace(-new_fs/2,new_fs/2,N);



figure(2)

subplot(5,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



MSG_resampled = fftshift(fft(msg_resampled));

MSG_resampled_MG = abs(X);

MSG_resampled_PHASE = angle(X);

N = length(X);

MSG_F = linspace(-new_fs/2,new_fs/2,N);



subplot(5,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')




subplot(5,1,3)

plot(F,SEND_SIGNAL_ABS)

title('DSB-SC TX modulated msg in Freq domain')
```

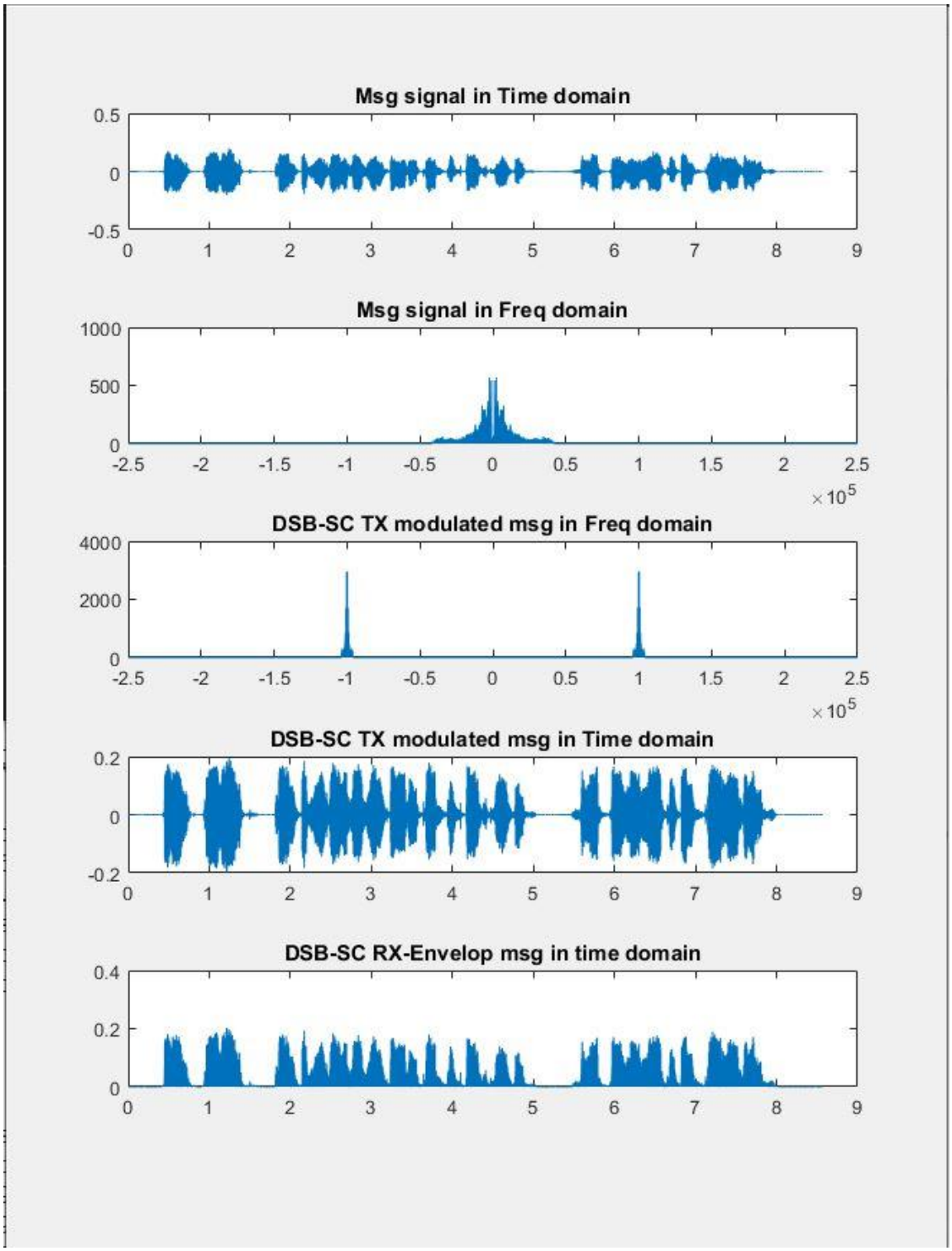
```
subplot(5,1,4)

plot(t,send_signal)

title('DSB-SC TX modulated msg in Time domain')
```

- Using carrier frequency of 100 KHz multiplied by the message signal. Plotting its frequency and time domain signals.


Msg signal in Time domain
0.2
0
-0.2
-0.4
0 1 2 3 4 5 6 7 8 9
Msg signal in Freq domain
600
400
200
0
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
x 10^5
DSB-SC TX modulated msg in Freq domain
3000
2000
1000
0
-2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5
x 10^5
DSB-SC TX modulated msg in Time domain
0.2
0
-0.2
0 1 2 3 4 5 6 7 8 9

DSB-SC envelope detector receiver:

```
%%%%%DSB-SC Envelope Receiver%%%%%%



envelop = abs(hilbert(send_signal));



subplot(5,1,5)

plot(t,envelop)

title('DSB-SC RX-Envelop msg in time domain')



% Resample down to play audio

original_audio = resample(envelop,fs,new_fs);

%sound(original_audio,fs);

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC
Envelope.wav',original_audio,fs);
```



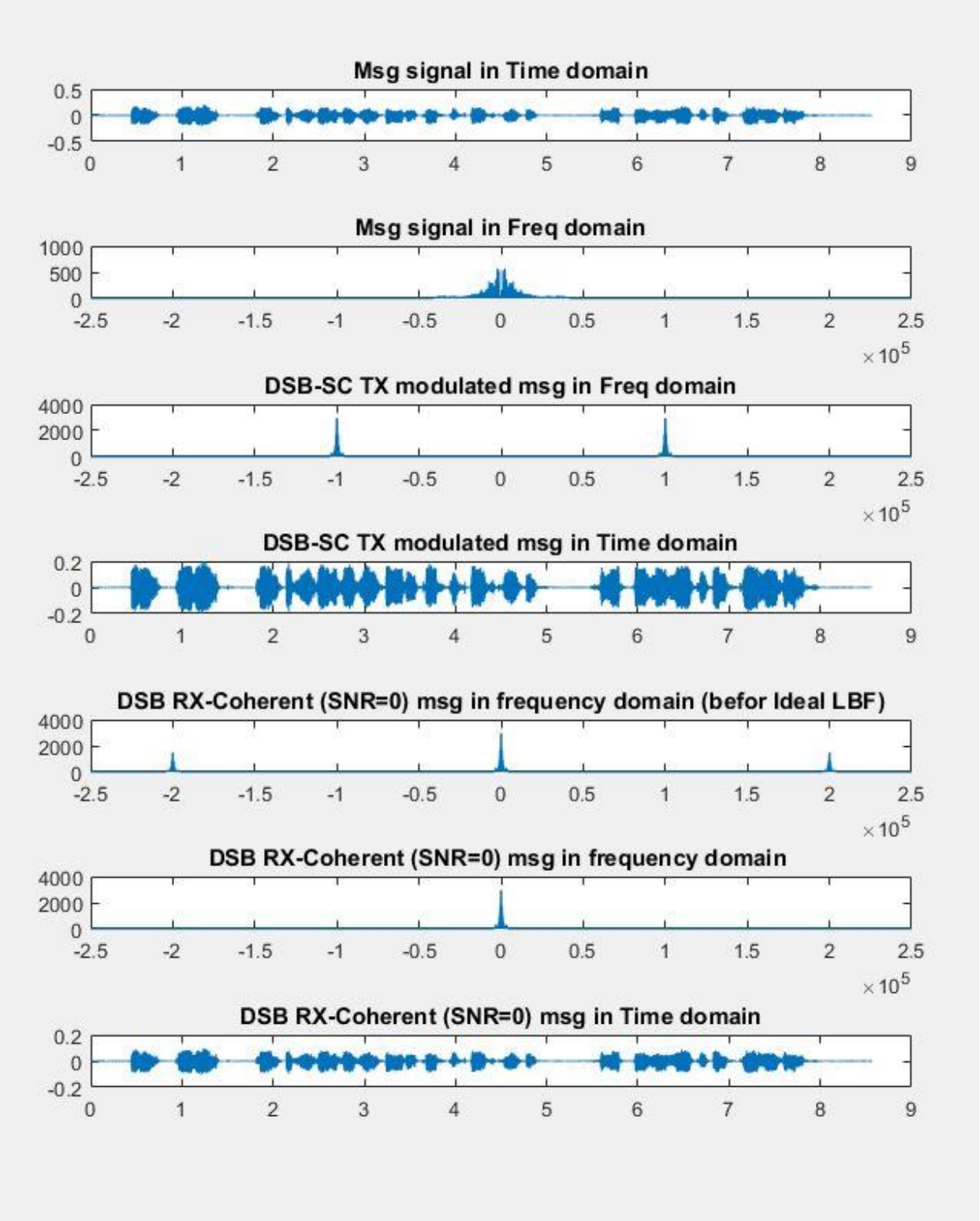
- Using an envelope detector we reobtain the message, but it seems malformed as it has some ticking noises also the bottom half of sent signal is missing.

DSB-SC Coherent detector receiver without noise:

```
%%%%%%%%COHERENT Detector 0 SNR %%%%%%%%

carrier = cos(2*pi*fc*t);

carrier = carrier';  % transpose to be a row matrix



% coherent (multiply msg with carrier)

coherent_detector_signal = send_signal.*carrier;



COHERENT_DETECTOR_SIGNAL = fftshift(fft(coherent_detector_signal));

% freq domain plot (befor Ideal LBF)

N = length(coherent_detector_signal);

F = linspace(-new_fs/2,new_fs/2,N);



figure(3)

subplot(7,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



subplot(7,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')
```

```
subplot(7,1,3)
plot(F,SEND_SIGNAL_ABS)
title('DSB-SC TX modulated msg in Freq domain')

subplot(7,1,4)
plot(t,send_signal)
title('DSB-SC TX modulated msg in Time domain')

subplot(7,1,5)
COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);
plot(F,COHERENT_DETECTOR_SIGNAL_ABS)
title('DSB RX-Coherent (SNR=0) msg in frequency domain (befor Ideal LBF)')


%------------- LBF ---------------%
N = length(coherent_detector_signal);
n = N/new_fs;
r_limit = round((new_fs/2-4000)*n);
l_limit = (N-r_limit+1);
COHERENT_DETECTOR_SIGNAL([1:r_limit l_limit:N]) = 0;
```

```
coherent_detector_signal_LPF = real(ifft(ifftshift(COHERENT_DETECTOR_SIGNAL)));

%------------- LBF ---------------%



% As a result of multipling the received signal with carrier there will be high freq
components

% at 2fc. So, we put the filter to remove them.



% freq domain plot (after Ideal LBF)

subplot(7,1,6)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent (SNR=0) msg in frequency domain ')




% time domain plot

subplot(7,1,7)

plot(t,coherent_detector_signal_LPF)

title('DSB RX-Coherent (SNR=0) msg in Time domain')



original_msg = resample(coherent_detector_signal_LPF,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC Coherent 0
SNR.wav',original_msg,fs)
```



- Sound is perfectly audible.

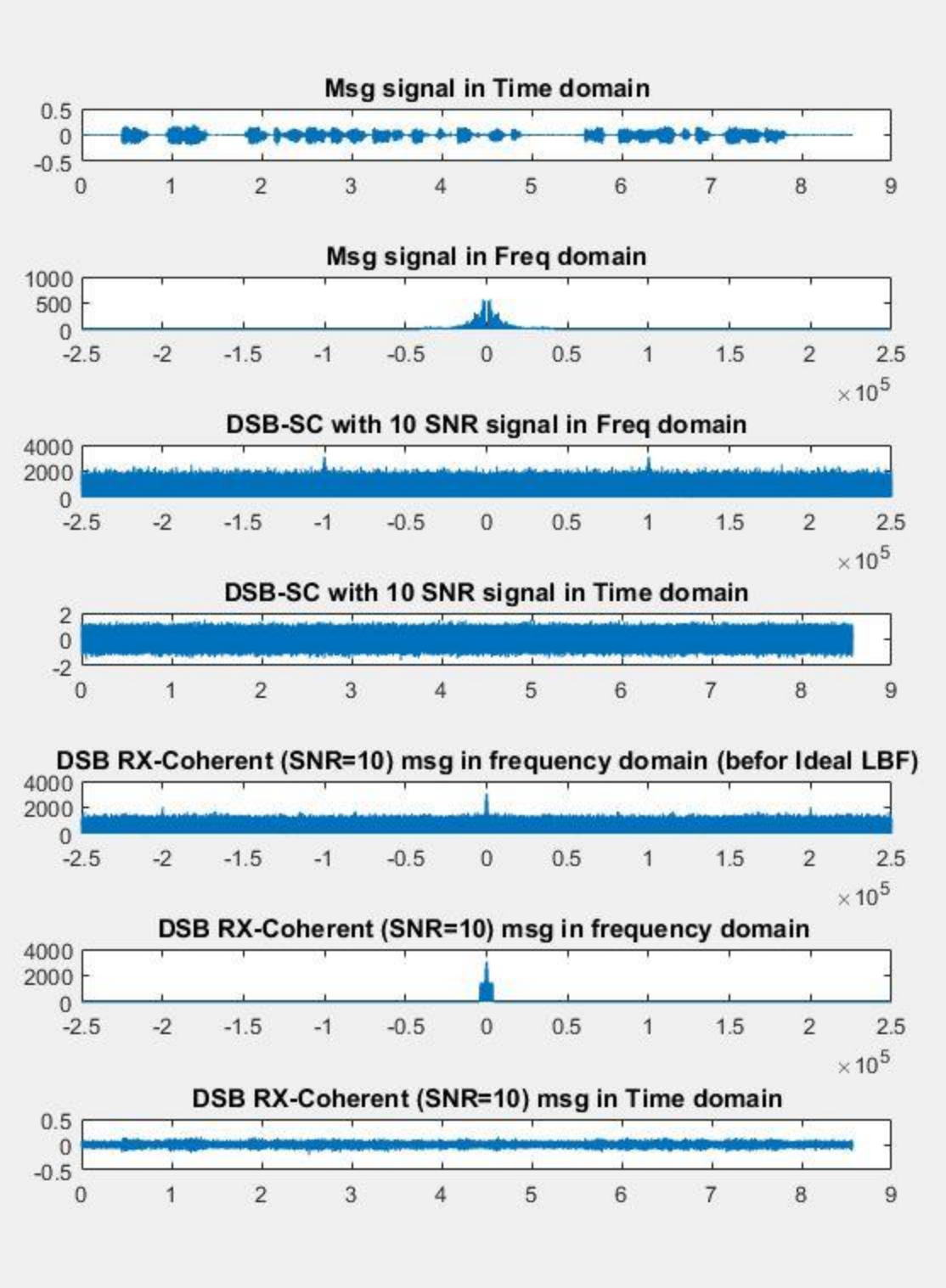
DSB-SC Coherent detector receiver with 10dB SNR:

```
%%%%%%%%%%%%%%%%%% COHERENT Detector 10 SNR %%%%%%%%%%%%%%%%%%%%

% add noise

send_signal_noisy = awgn(send_signal,10);

figure(4)

subplot(7,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



subplot(7,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')




send_signal_noisy_FREQ = fftshift(fft(send_signal_noisy));

N = length(send_signal_noisy_FREQ);

F = linspace(-new_fs/2,new_fs/2,N);



subplot(7,1,3)

plot(F,abs(send_signal_noisy_FREQ))

title('DSB-SC with 10 SNR signal in Freq domain')
```

```
subplot(7,1,4)
plot(t,send_signal_noisy)
title('DSB-SC with 10 SNR signal in Time domain')



carrier = cos(2*pi*fc*t);
carrier = carrier';  % transpose to be a row matrix


% coherent (multiply msg with carrier)
coherent_detector_signal = send_signal_noisy.*carrier;


COHERENT_DETECTOR_SIGNAL = fftshift(fft(coherent_detector_signal));
% freq domain plot (befor Ideal LBF)
N = length(coherent_detector_signal);
F = linspace(-new_fs/2,new_fs/2,N);
subplot(7,1,5)
COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);
plot(F,COHERENT_DETECTOR_SIGNAL_ABS)
title('DSB RX-Coherent (SNR=10) msg in frequency domain (befor Ideal LBF)')
```

```
%------------- LBF ---------------%

N = length(coherent_detector_signal);

n = N/new_fs;

r_limit = round((new_fs/2-4000)*n);

l_limit = (N-r_limit+1);

COHERENT_DETECTOR_SIGNAL([1:r_limit l_limit:N]) = 0;

coherent_detector_signal_LPF = real(ifft(ifftshift(COHERENT_DETECTOR_SIGNAL)));

%------------- LBF ---------------%



% As a result of multipling the received signal with carrier there will be high freq
components

% at 2fc. So, we put the filter to remove them.



% freq domain plot (after Ideal LBF)

subplot(7,1,6)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent (SNR=10) msg in frequency domain ')




% time domain plot

subplot(7,1,7)

plot(t,coherent_detector_signal_LPF)
```

```
title('DSB RX-Coherent (SNR=10) msg in Time domain')



original_msg = resample(coherent_detector_signal_LPF,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC Coherent 10
SNR.wav',original_msg,fs);
```



Sound is has so much static it is barely audible.

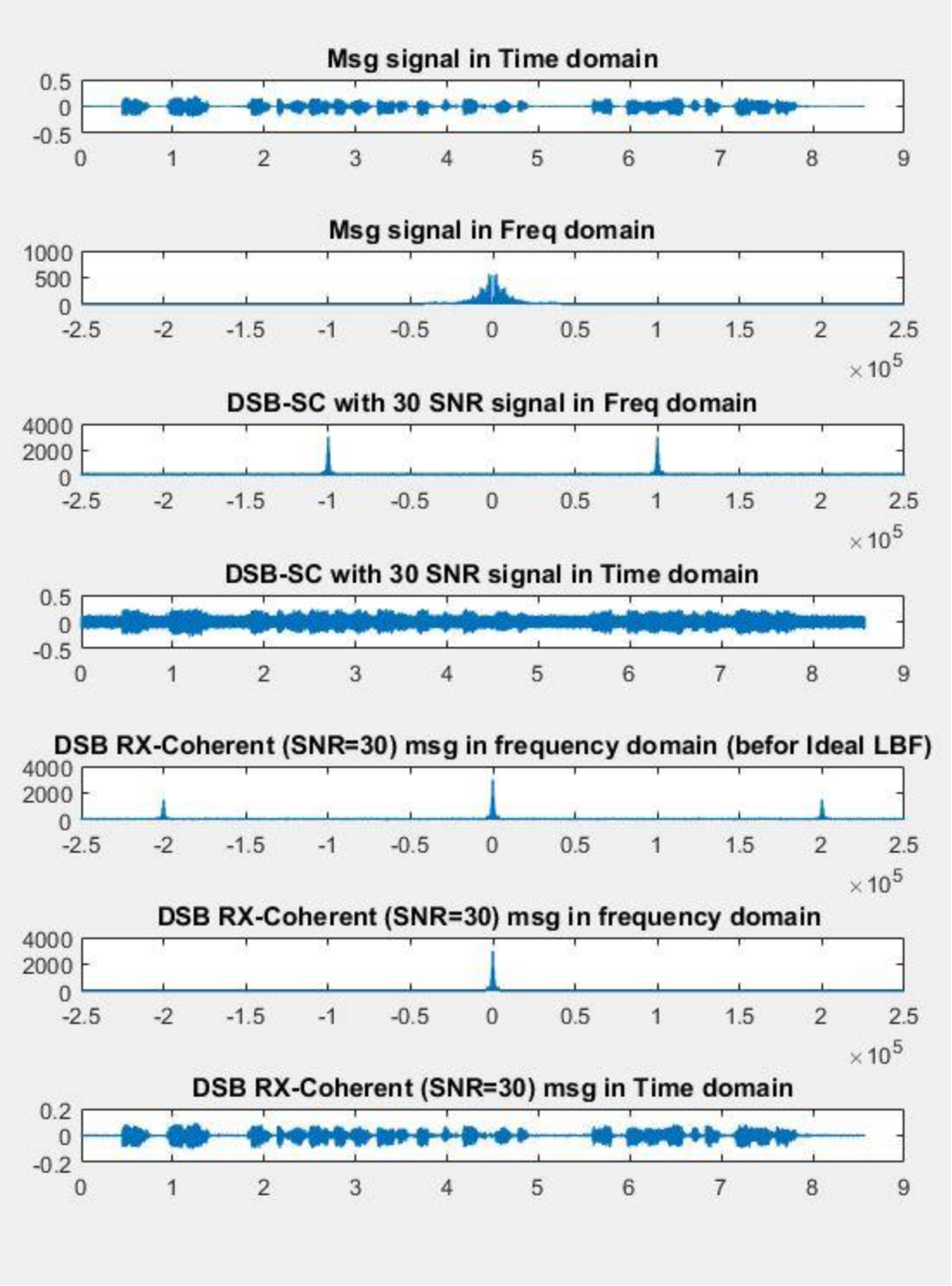
DSB-SC Coherent detector receiver with 30dB SNR:

```
%%%%%%%%%%%%%%%%% COHERENT Detector 30 SNR %%%%%%%%%%%%%%%%%%%

% add noise

send_signal_noisy = awgn(send_signal,30);

figure(5)

subplot(7,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



subplot(7,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')




send_signal_noisy_FREQ = fftshift(fft(send_signal_noisy));

N = length(send_signal_noisy_FREQ);

F = linspace(-new_fs/2,new_fs/2,N);



subplot(7,1,3)

plot(F,abs(send_signal_noisy_FREQ))
```

```
title('DSB-SC with 30 SNR signal in Freq domain')



subplot(7,1,4)

plot(t,send_signal_noisy)

title('DSB-SC with 30 SNR signal in Time domain')




carrier = cos(2*pi*fc*t);

carrier = carrier';  % transpose to be a row matrix



% coherent (multiply msg with carrier)

coherent_detector_signal = send_signal_noisy.*carrier;



COHERENT_DETECTOR_SIGNAL = fftshift(fft(coherent_detector_signal));

% freq domain plot (befor Ideal LBF)

N = length(coherent_detector_signal);

F = linspace(-new_fs/2,new_fs/2,N);

subplot(7,1,5)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent (SNR=30) msg in frequency domain (befor Ideal LBF)')
```

```
%------------- LBF ---------------%

N = length(coherent_detector_signal);

n = N/new_fs;

r_limit = round((new_fs/2-4000)*n);

l_limit = (N-r_limit+1);

COHERENT_DETECTOR_SIGNAL([1:r_limit l_limit:N]) = 0;

coherent_detector_signal_LPF = real(ifft(ifftshift(COHERENT_DETECTOR_SIGNAL)));

%------------- LBF ---------------%



% As a result of multipling the received signal with carrier there will be high freq
components

% at 2fc. So, we put the filter to remove them.



% freq domain plot (after Ideal LBF)

subplot(7,1,6)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent (SNR=30) msg in frequency domain ')




% time domain plot

subplot(7,1,7)
```

```
plot(t,coherent_detector_signal_LPF)

title('DSB RX-Coherent (SNR=30) msg in Time domain')



original_msg = resample(coherent_detector_signal_LPF,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC Coherent 30
SNR.wav',original_msg,fs);
```



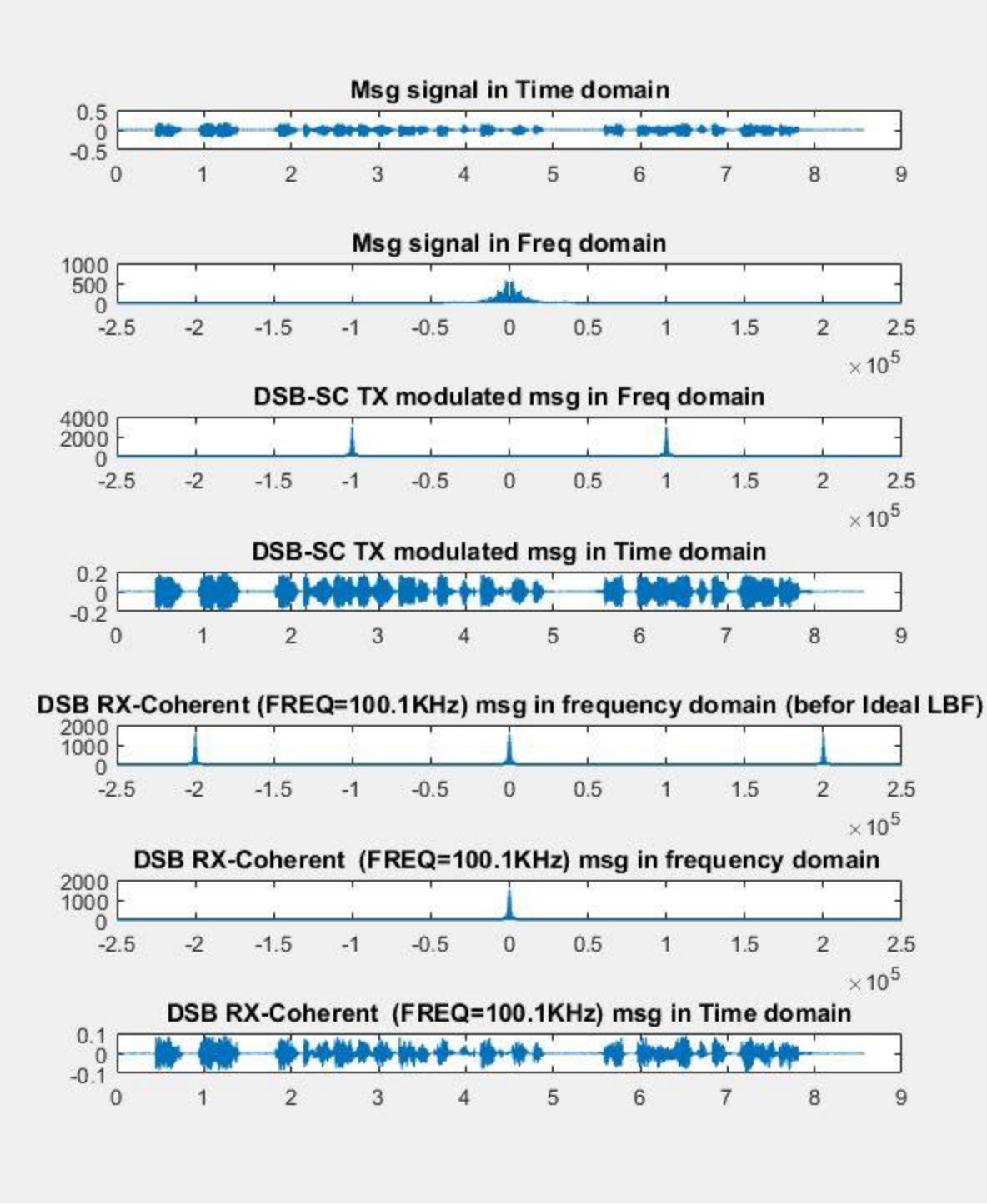
Sound is having static but is audible.

DSB-SC Coherent detector with incorrect frequency:

```
%%%%%%%%COHERENT Detector with frequency 100.1%%%%%%%%%%%%%%5

carrier = cos(2*pi*100100*t);

carrier = carrier';  % transpose to be a row matrix



% coherent (multiply msg with carrier)

coherent_detector_signal = send_signal.*carrier;



COHERENT_DETECTOR_SIGNAL = fftshift(fft(coherent_detector_signal));

% freq domain plot (befor Ideal LBF)

N = length(coherent_detector_signal);

F = linspace(-new_fs/2,new_fs/2,N);



figure(6)

subplot(7,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



subplot(7,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')
```

```
subplot(7,1,3)
plot(F,SEND_SIGNAL_ABS)
title('DSB-SC TX modulated msg in Freq domain')

subplot(7,1,4)
plot(t,send_signal)
title('DSB-SC TX modulated msg in Time domain')

subplot(7,1,5)
COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);
plot(F,COHERENT_DETECTOR_SIGNAL_ABS)
title('DSB RX-Coherent (FREQ=100.1KHz) msg in frequency domain (befor Ideal LBF)')


%------------- LBF ---------------%
N = length(coherent_detector_signal);
n = N/new_fs;
r_limit = round((new_fs/2-4000)*n);
l_limit = (N-r_limit+1);
COHERENT_DETECTOR_SIGNAL([1:r_limit l_limit:N]) = 0;
```

```
coherent_detector_signal_LPF = real(ifft(ifftshift(COHERENT_DETECTOR_SIGNAL)));

%------------- LBF ---------------%



% As a result of multipling the received signal with carrier there will be high freq
components

% at 2fc. So, we put the filter to remove them.



% freq domain plot (after Ideal LBF)

subplot(7,1,6)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent  (FREQ=100.1KHz) msg in frequency domain ')




% time domain plot

subplot(7,1,7)

plot(t,coherent_detector_signal_LPF)

title('DSB RX-Coherent  (FREQ=100.1KHz) msg in Time domain')



original_msg = resample(coherent_detector_signal_LPF,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC Coherent
(FREQ=100.1KHz).wav',original_msg,fs);
```



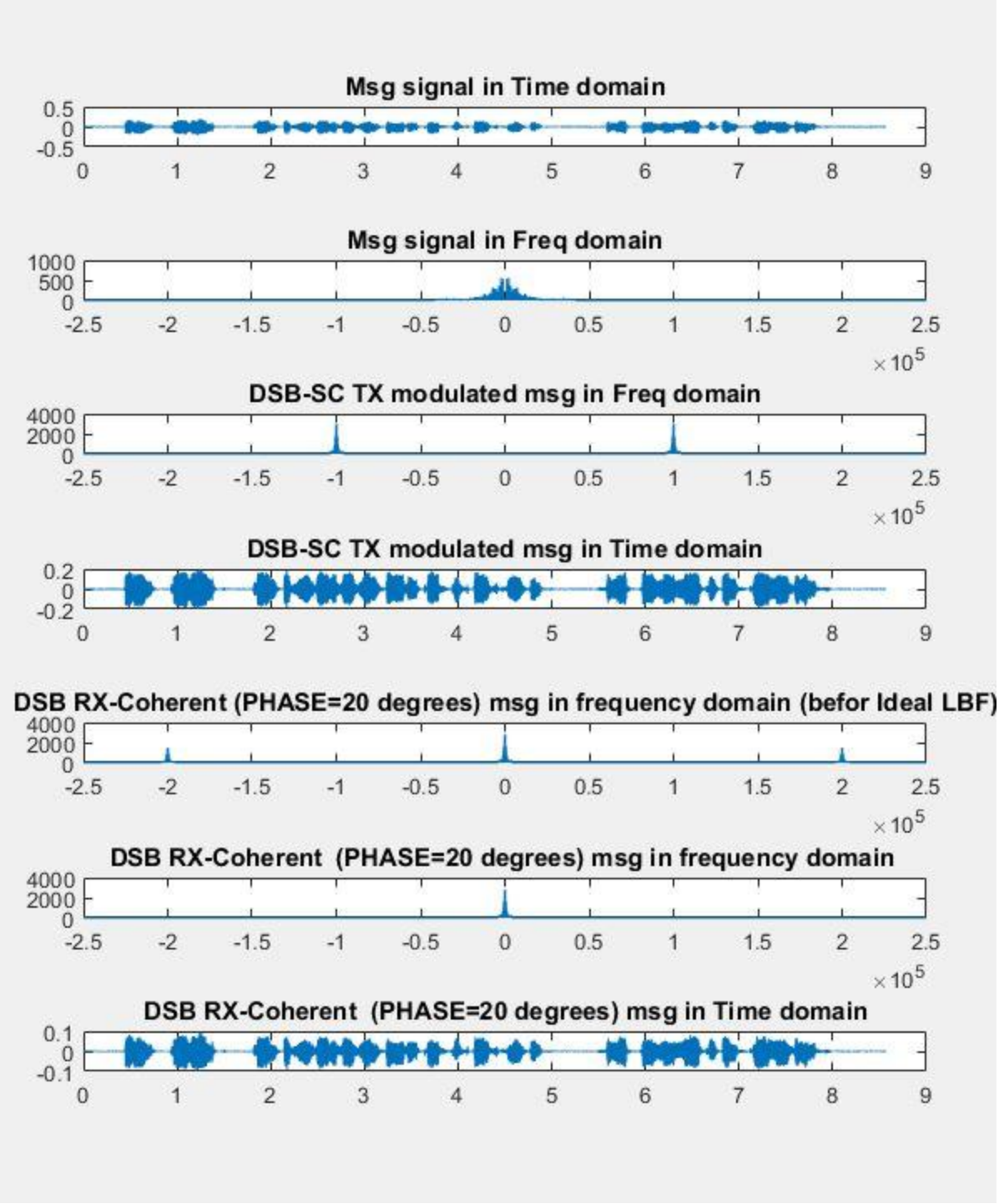
Not only the sound is attenuated but the message is messed up.

DSB-SC Coherent detector with incorrect Phase:

```
%%%%%%%%COHERENT Detector with phase shift 20 degrees %%%%%%%%%%%%%%5

carrier = cos(2*pi*fc*t+20/180*pi);

carrier = carrier';  % transpose to be a row matrix



% coherent (multiply msg with carrier)

coherent_detector_signal = send_signal.*carrier;



COHERENT_DETECTOR_SIGNAL = fftshift(fft(coherent_detector_signal));

% freq domain plot (befor Ideal LBF)

N = length(coherent_detector_signal);

F = linspace(-new_fs/2,new_fs/2,N);



figure(7)

subplot(7,1,1)

plot(t,msg_resampled)

title('Msg signal in Time domain')



subplot(7,1,2)

plot(MSG_F,MSG_resampled_MG)

title('Msg signal in Freq domain')
```

```
subplot(7,1,3)
plot(F,SEND_SIGNAL_ABS)
title('DSB-SC TX modulated msg in Freq domain')

subplot(7,1,4)
plot(t,send_signal)
title('DSB-SC TX modulated msg in Time domain')

subplot(7,1,5)
COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);
plot(F,COHERENT_DETECTOR_SIGNAL_ABS)
title('DSB RX-Coherent (PHASE=20 degrees) msg in frequency domain (befor Ideal LBF)')

%------------- LBF ---------------%
N = length(coherent_detector_signal);
n = N/new_fs;
r_limit = round((new_fs/2-4000)*n);
l_limit = (N-r_limit+1);
COHERENT_DETECTOR_SIGNAL([1:r_limit l_limit:N]) = 0;
```

```
coherent_detector_signal_LPF = real(ifft(ifftshift(COHERENT_DETECTOR_SIGNAL)));

%------------- LBF ---------------%



% As a result of multipling the received signal with carrier there will be high freq
components

% at 2fc. So, we put the filter to remove them.



% freq domain plot (after Ideal LBF)

subplot(7,1,6)

COHERENT_DETECTOR_SIGNAL_ABS = abs(COHERENT_DETECTOR_SIGNAL);

plot(F,COHERENT_DETECTOR_SIGNAL_ABS)

title('DSB RX-Coherent  (PHASE=20 degrees) msg in frequency domain ')




% time domain plot

subplot(7,1,7)

plot(t,coherent_detector_signal_LPF)

title('DSB RX-Coherent  (PHASE=20 degrees) msg in Time domain')



original_msg = resample(coherent_detector_signal_LPF,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-SC Coherent(PHASE=20
degrees).wav',original_msg,fs);
```

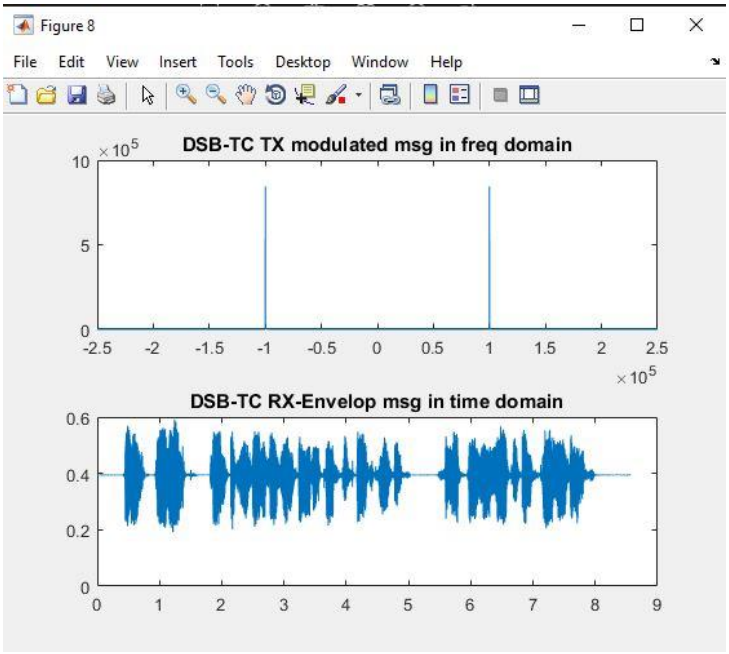


- Sound is attenuated but is intact.

DSB-TC Sender and envelope detector:

```
%%%%%%%%% DSP TC %%%%%%

A = max(msg_resampled)*2 ; % Amplitude of carier is double max value m=0.5



% modulating

carrier = cos(2*pi*fc*t);

carrier = carrier';

send_signal = (A + msg_resampled).*carrier;




% freq domain

SEND_SIGNAL = fftshift(fft(send_signal));

SEND_SIGNAL_ABS = abs(SEND_SIGNAL);

N = length(send_signal);

F = linspace(-new_fs/2,new_fs/2,N);



figure(7)

subplot(2,1,1)

plot(F,SEND_SIGNAL_ABS)

title('DSB-TC TX modulated msg in freq domain')
```

```
%------------------------- RX --------------------------%

%------------- Envelop--------------%

envelop = abs(hilbert(send_signal));



subplot(2,1,2)

plot(t,envelop)

title('DSB-TC RX-Envelop msg in time domain')




original_msg = resample(envelop,fs,new_fs);

%sound(original_msg,fs)

%audiowrite('C:\Users\Ragai\Desktop\Analog Project\DSB-TC.wav',original_msg,fs);
```



Experiment 2

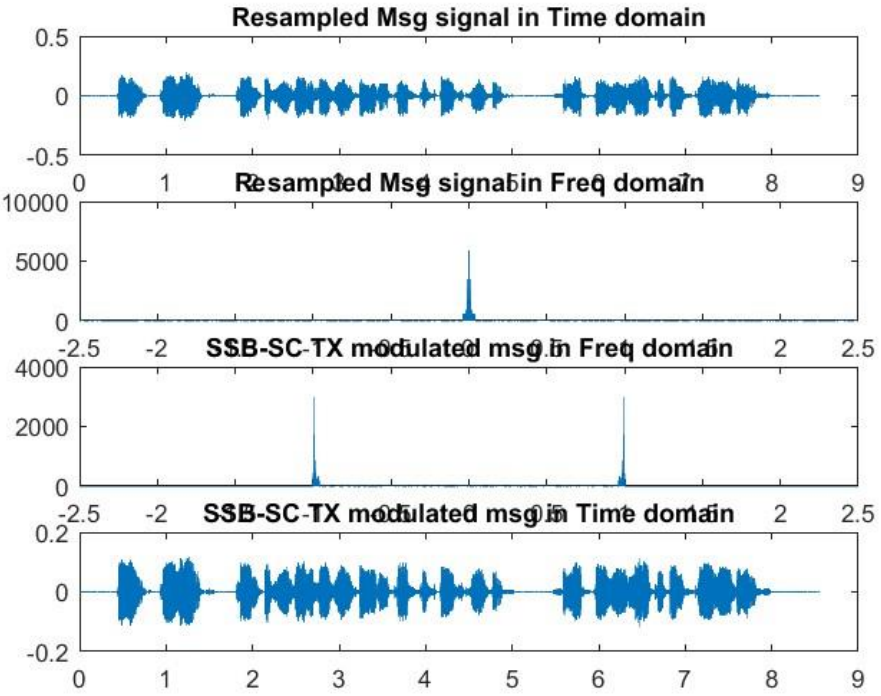
```
clear; clc;
% orignal sound
[xin ,fs] = audioread('eric.wav');
audio_length= length(xin)./fs;
t=linspace(0,audio_length,length(xin));
 figure (1) subplot(2,1,1)
plot(t,xin) title('signal in
Time domain')

f_xin=fftshift(fft(xin));
f_xin_mg= abs(f_xin); N =
length(xin);
f_vec = linspace(-fs/2,fs/2,N);
 subplot(2,1,2) plot(f_vec,f_xin_mg)
title(' signal in Frequency domain')

%%%%%%%%%LPF%%%%%%%%%
%filter at 4khz n =
N/fs;
right_band = round((fs/2-4000)*n); left_band
= (N-right_band+1); f_xin([1:right_band
left_band:N]) = 0;
 figure(2)
subplot(2,1,2)
plot(f_vec,abs(f_xin))
title('LPF(4khz) Signal in Frequency domain')
xin = real(ifft(ifftshift(f_xin)));


subplot(2,1,1) plot(t,xin)
title('LPF(4khz) Signal in Time domain')
```

transmitter

```
fc = 100000; fs_new = 5*fc; msg_resampled =
resample(xin,fs_new,fs); t_end =
length(msg_resampled)./fs_new;
```

```
t = linspace(0,t_end, length(msg_resampled));

% modulation
carrier = cos(2*pi*fc*t); carrier
= carrier';
 transmitted_m = msg_resampled.*carrier;
f_transmitted_m= fftshift(fft(transmitted_m));

f_t_magnitude= abs(f_transmitted_m); N=length(transmitted_m);
f_vec = linspace(-fs_new/2,fs_new/2,N);

% get the LSB usin LPF N =
length(msg_resampled); n = N/fs_new;
right_band = round((fs_new/2-100000)*n);
left_band = (N-right_band+1);
f_transmitted_m([1:right_band left_band:N]) = 0; f_t_magnitude
= abs(f_transmitted_m);
transmitted_m = real(ifft(ifftshift(f_transmitted_m)));

t_end = length(transmitted_m)./fs_new; t =
linspace(0,t_end, length(transmitted_m));

  figure (3)
subplot(4,1,1)
plot(t,msg_resampled)
title('Resampled Msg signal in Time domain')

f_resampled=fftshift(fft(msg_resampled));
f_resampled_mg=abs(f_resampled); N =
length(f_transmitted_m);
f_vec = linspace(-fs_new/2,fs_new/2,N);

subplot(4,1,2)
plot(f_vec,f_resampled_mg)
title('Resampled Msg signal in Freq domain')

subplot(4,1,3) plot(f_vec,abs(f_transmitted_m))
title('SSB-SC TX modulated msg in Freq domain')
```

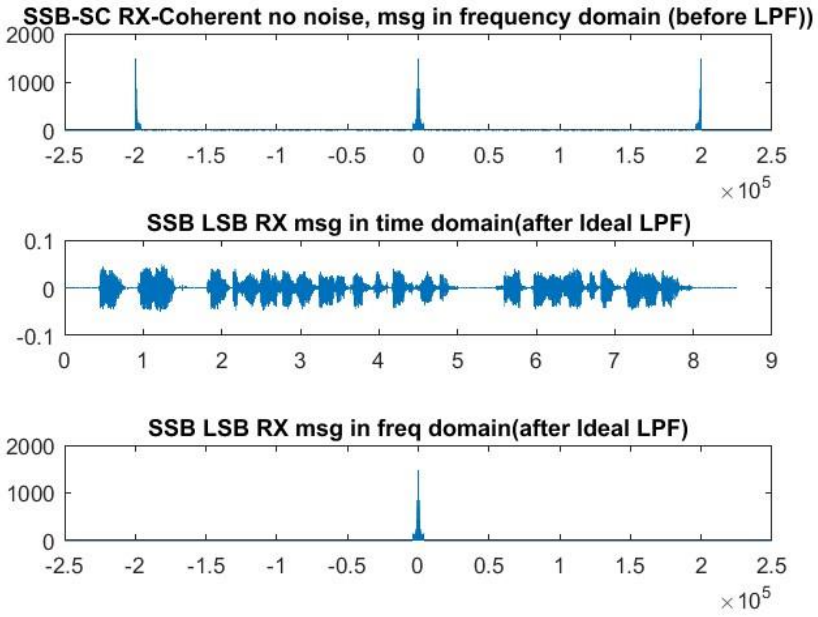
```
subplot(4,1,4) plot(t,transmitted_m)
title('SSB-SC TX modulated msg in Time domain')
```



receiver

```
%%%%%% Ideal LPF receiver %%%%%%%%%%%%% ideal_rx(transmitted_m,t,f_vec,fs_new,fs);

function ideal_rx(transmitted_m,t,f_vec,fs_new,fs) received_message
= transmitted_m.*(cos(2*pi*100000*t)'); f_received_message =
fftshift(fft(received_message)); received_message_mg =
abs(f_received_message);
  figure
subplot(3,1,1)
plot(f_vec,received_message_mg)
title('SSB-SC RX-Coherent no noise, msg in frequency domain (before LPF))')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Ideal LPF%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LPF to remove signal @ 2fc left_band = (N-
right_band+1); f_received_message([1:right_band
left_band:N]) = 0; received_message_mg =
abs(f_received_message);
```

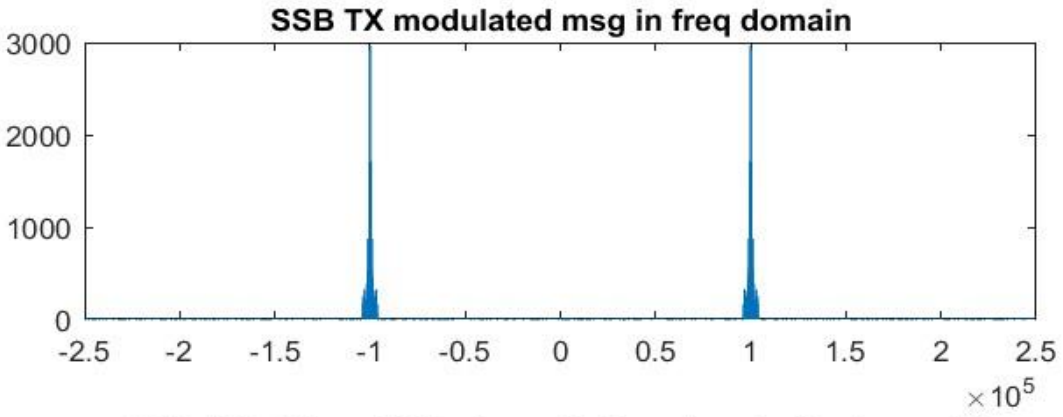
```
received_message = real(ifft(ifftshift(f_received_message)));
 subplot(3,1,2) plot(t,received_message) title('SSB LSB
RX msg in time domain(after Ideal LPF)')
 subplot(3,1,3)
plot(f_vec,received_message_mg) title('SSB LSB RX msg
in freq domain(after Ideal LPF)') original_msg =
resample(received_message,fs,fs_new);
%sound(original_msg, fs)
```

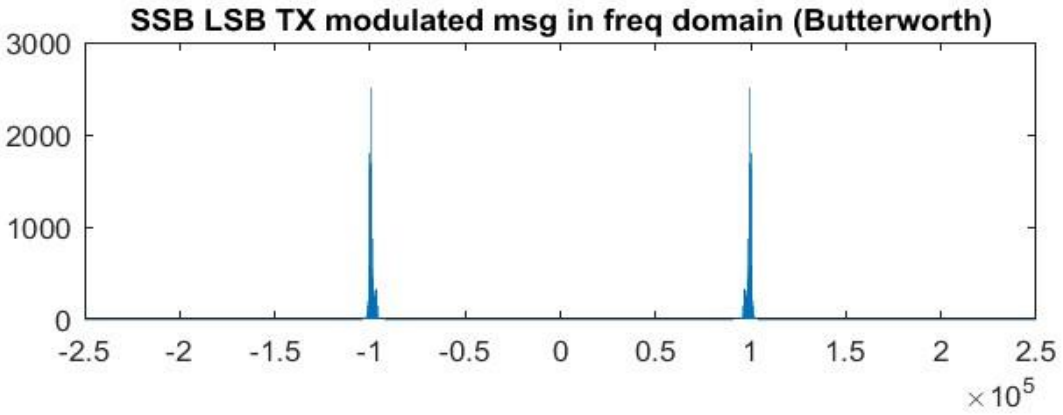


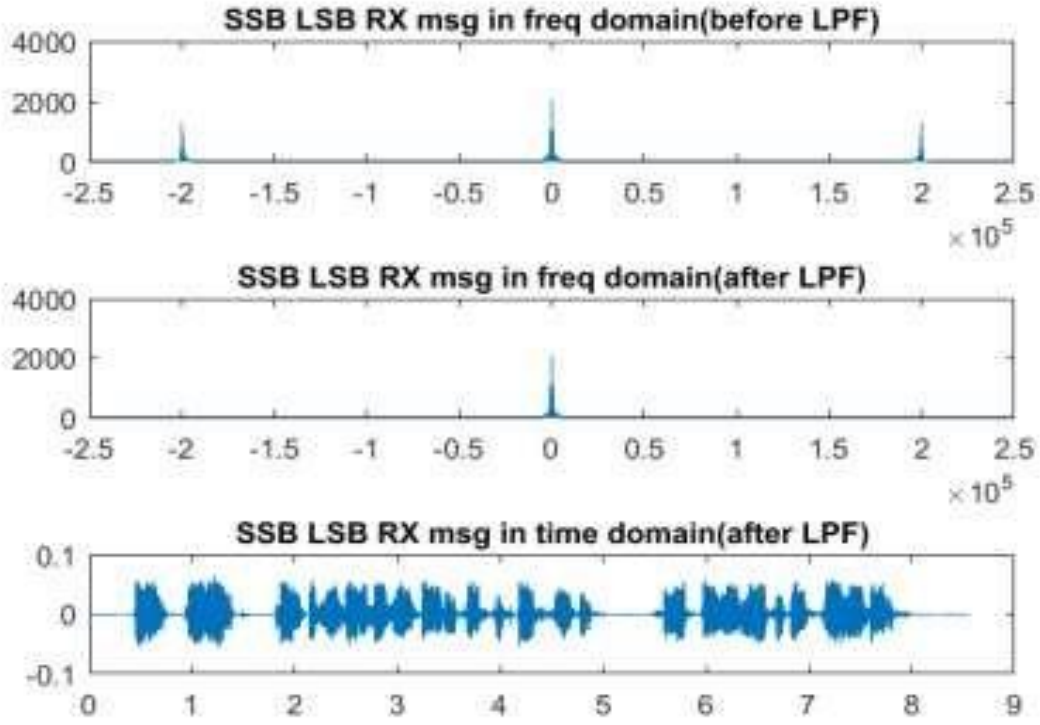
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Butterworth filter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
transmitted_m = msg_resampled.*carrier;


f_transmitted_m = fftshift(fft(transmitted_m));
f_transmitted_mg  =  abs(f_transmitted_m);  N  =
length(transmitted_m);
f_vec = linspace(-fs_new/2,fs_new/2,N);

%plot in freq figure(5)
subplot(2,1,1)
plot(f_vec,f_transmitted_mg)
title('SSB TX modulated msg in freq domain')
```

```
% butterworth bandpass filter to filterout the USB
[b, a]= butter(4, [(fc-4000)/(fs_new/2) fc/(fs_new/2)], 'bandpass');
 tx_msg_LSB =
filter(b,a,transmitted_m);
%plot in freq figure(5)
subplot(2,1,2)
plot(f_vec,TX_msg_LSB_F_mg)
title('SSB LSB TX modulated msg in freq domain (Butterworth)')

tr_msg_coh = tx_msg_LSB.*carrier;

tr_msg_coh_F = fftshift(fft(tr_msg_coh)); tr_msg_coh_F_mg
= abs(tr_msg_coh_F);

%plot in freq figure(6)
subplot(3,1,1)
plot(f_vec,tr_msg_coh_F_mg)
title('SSB LSB RX msg in freq domain(before LPF)')
%------------- LBF ---------------%
N = length(tr_msg_coh); n =
N/fs_new;
right_band = round((fs_new/2-4000)*n); left_band
= (N-right_band+1);
tr_msg_coh_F([1:right_band left_band:N]) = 0; tr_msg_coh_F_mg
= abs(tr_msg_coh_F);
tr_msg_coh_LPF = real(ifft(ifftshift(tr_msg_coh_F)));
%------------- LBF ---------------% subplot(3,1,2)
plot(f_vec,tr_msg_coh_F_mg)
title('SSB LSB RX msg in freq domain(after LPF)')

subplot(3,1,3) plot(t,tr_msg_coh_LPF) title('SSB
LSB RX msg in time domain(after LPF)')
%original_msg = resample(tr_msg_coh_LPF,fs,fs_new);
```

```
%sound(original_msg, fs)
```


SSB TX modulated msg in freq domain
3000
2000
1000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^5

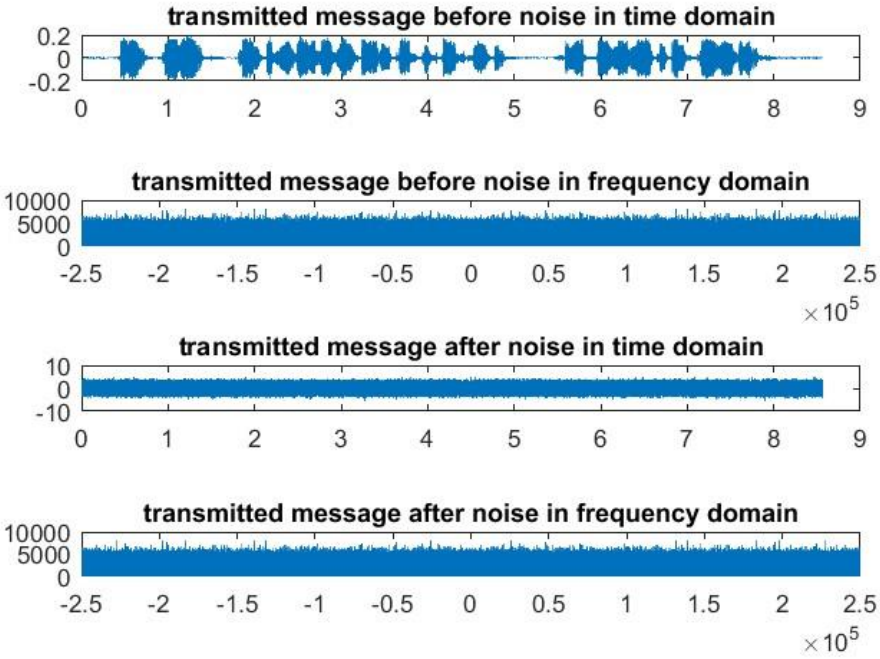

SSB LSB TX modulated msg in freq domain (Butterworth)
3000
2000
1000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^5

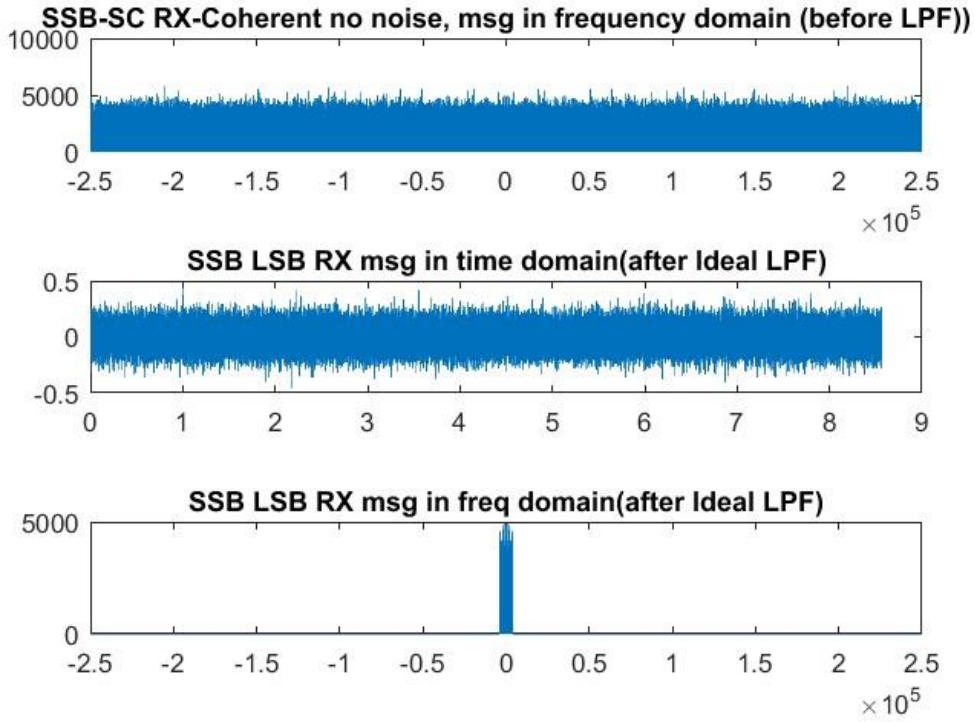

SSB LSB RX msg in freq domain(before LPF)
4000
2000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^5
SSB LSB RX msg in freq domain(after LPF)
4000
2000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
×10^5
SSB LSB RX msg in time domain(after LPF)
0.1
0
-0.1
0
1
2
3
4
5
6
7
8
9

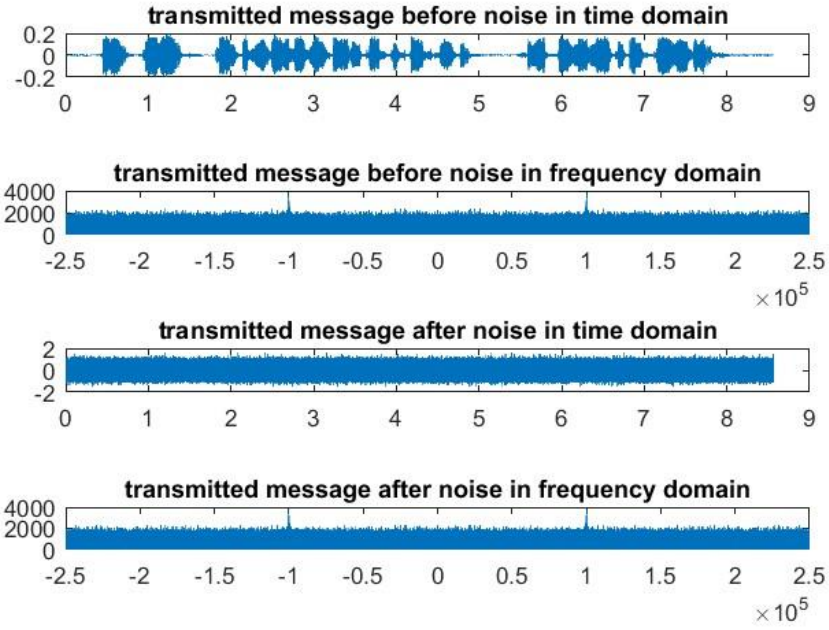
noise

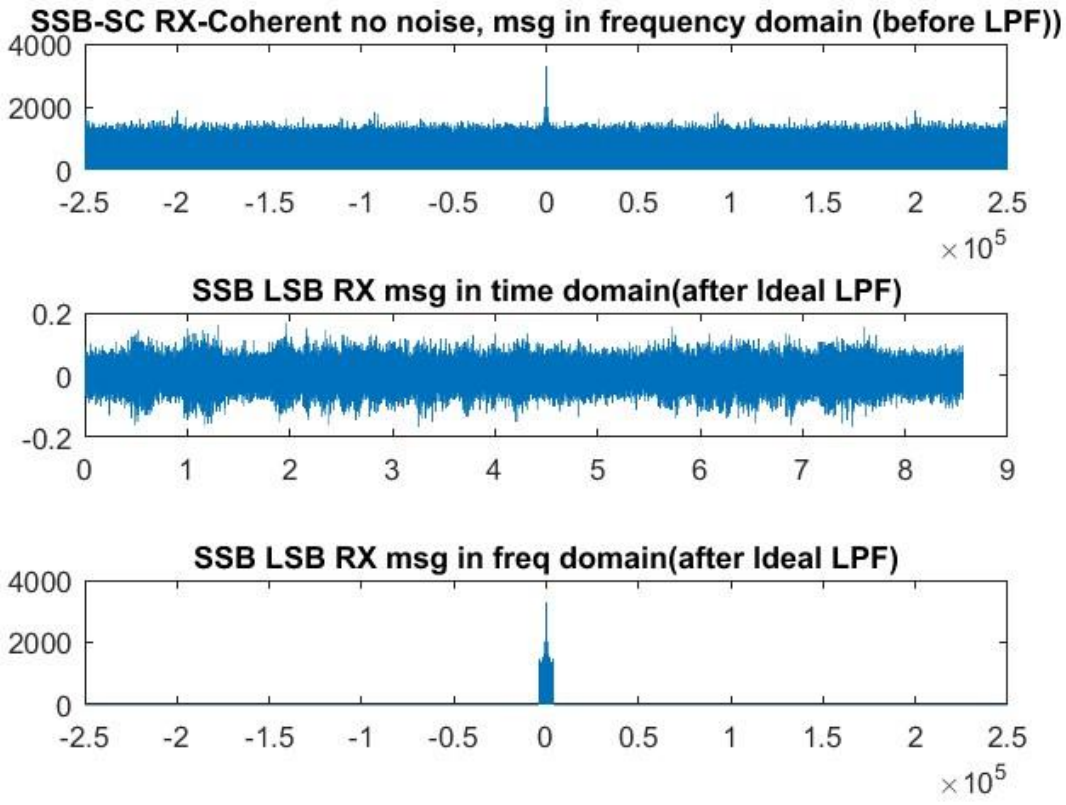
```
n_snr=0;
add_noise(transmitted_m,t,f_vec,n_snr); message_noise=
awgn(transmitted_m, n_snr);
ideal_rx(message_noise,t,f_vec,fs_new,fs);
 n_snr=10;
add_noise(transmitted_m,t,f_vec,n_snr);
message_noise= awgn(transmitted_m, n_snr);
ideal_rx(message_noise,t,f_vec,fs_new,fs);
  n_snr=30;
add_noise(transmitted_m,t,f_vec,n_snr);
message_noise= awgn(transmitted_m, n_snr);
ideal_rx(message_noise,t,f_vec,fs_new,fs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function add_noise(transmitted_m,t,f_vec, n_snr)

message_noise= awgn(transmitted_m, n_snr);
f_transmitted_m=abs(fftshift(fft(message_noise)));
f_message_noise=fftshift(fft(message_noise)); f_noise_mg=abs(f_message_noise);
 figure
subplot(4,1,1)
```

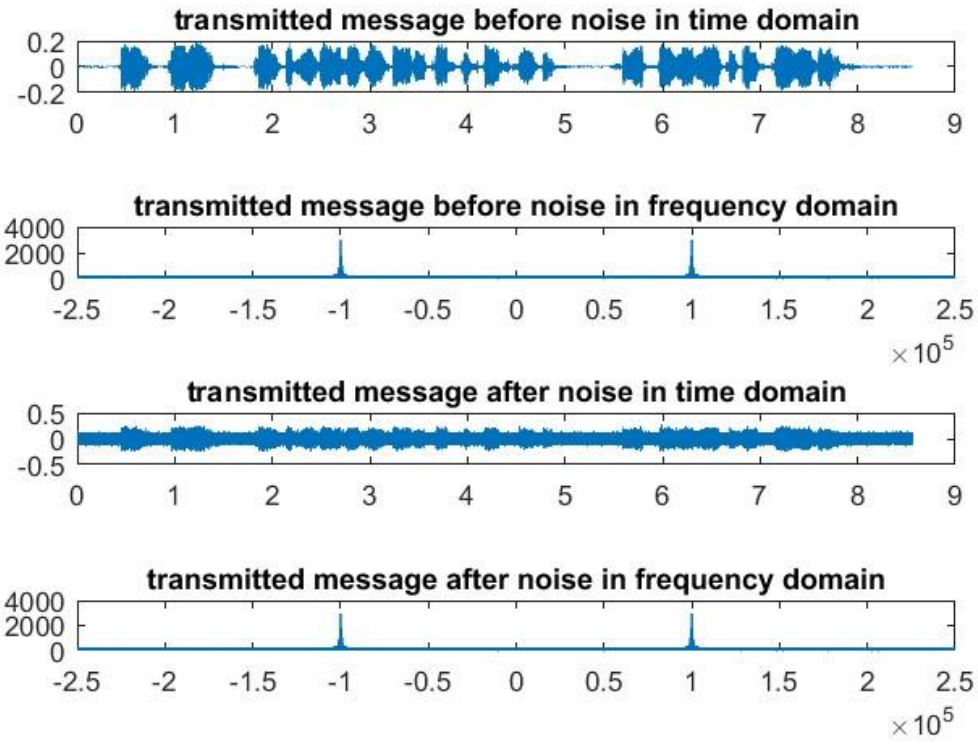
```
plot(t,transmitted_m)
title('transmitted message before noise in time domain')

subplot(4,1,2) plot(f_vec,f_transmitted_m)
title('transmitted message before noise in frequency domain')

subplot(4,1,3) plot(t,message_noise)
title('transmitted message after noise in time domain')

subplot(4,1,4) plot(f_vec,f_noise_mg)
title('transmitted message after noise in frequency domain')
```

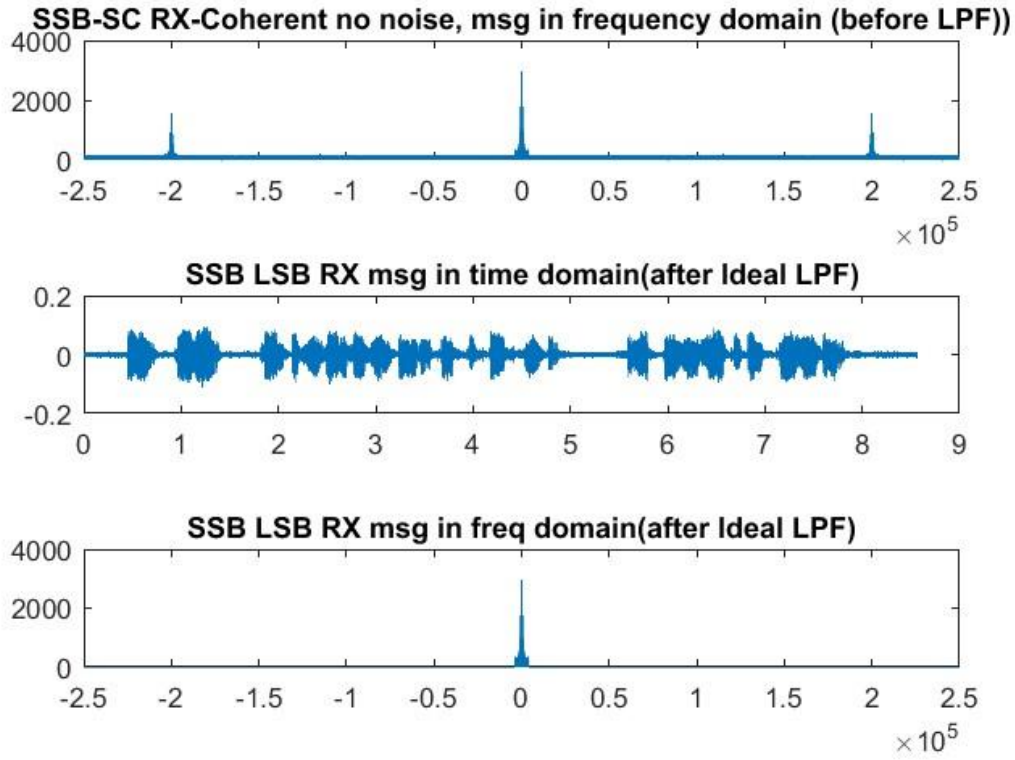

transmitted message before noise in time domain
transmitted message before noise in frequency domain
transmitted message after noise in time domain
transmitted message after noise in frequency domain
$\times 10^5$


SSB-SC RX-Coherent no noise, msg in frequency domain (before LPF))
SSB LSB RX msg in time domain(after Ideal LPF)
SSB LSB RX msg in freq domain(after Ideal LPF)
$\times 10^5$


transmitted message before noise in time domain
transmitted message before noise in frequency domain
transmitted message after noise in time domain
transmitted message after noise in frequency domain
×10^5


SSB-SC RX-Coherent no noise, msg in frequency domain (before LPF))
SSB LSB RX msg in time domain(after Ideal LPF)
SSB LSB RX msg in freq domain(after Ideal LPF)
×10^5


transmitted message before noise in time domain
transmitted message before noise in frequency domain
transmitted message after noise in time domain
transmitted message after noise in frequency domain
×10^5


SSB-SC RX-Coherent no noise, msg in frequency domain (before LPF))
SSB LSB RX msg in time domain(after Ideal LPF)
SSB LSB RX msg in freq domain(after Ideal LPF)
×10^5

9) SSB-TC:

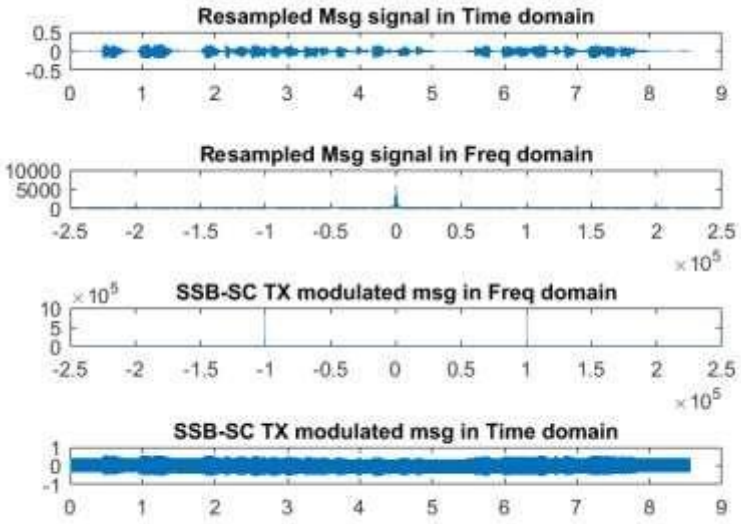
```
clear; clc;
[xin ,fs] = audioread('eric.wav'); audio_length=
length(xin)./fs;
t=linspace(0,audio_length,length(xin));


f_xin=fftshift(fft(xin));
f_xin_mg= abs(f_xin); N =
length(xin);
f_vec = linspace(-fs/2,fs/2,N);


%%%%%%%%%LPF%%%%%%%%% %filter at 4khz
n = N/fs; right_band = round((fs/2-
4000)*n); left_band = (N-
right_band+1); f_xin([1:right_band
left_band:N]) = 0;
 figure(1)
subplot(2,1,2)
plot(f_vec,abs(f_xin))
title('LPF(4khz) Signal in Frequency domain') xin
= real(ifft(ifftshift(f_xin)));
 subplot(2,1,1) plot(t,xin)
title('LPF(4khz) Signal in Time domain')
```

transmitter

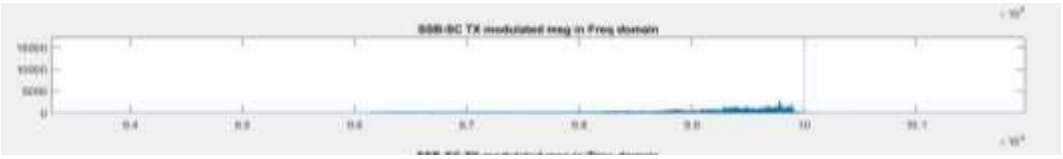
```
fc = 100000;
```

```
fs_new = 5*fc;
msg_resampled = resample(xin,fs_new,fs); A=max(msg_resampled)*2;
t_end = length(msg_resampled)./fs_new; t =
linspace(0,t_end, length(msg_resampled));

% modulation
carrier = cos(2*pi*fc*t); carrier
= carrier';
 transmitted_m = (A+msg_resampled).*carrier;
f_transmitted_m=
fftshift(fft(transmitted_m));
f_transmitted_mg= abs(f_transmitted_m);
N=length(transmitted_m); f_vec = linspace(-
fs_new/2,fs_new/2,N);




% get the LSB usin LPF N =
length(transmitted_m); f_vec =
linspace(-fs_new/2,fs_new/2,N); index =
f_vec>=fc+1; f_transmitted_mg(index) =
0;
% remove frequencies < -fc (remove negative HSB)
index2 = f_vec<=(-fc); f_transmitted_mg(index2)
= 0; f_transmitted_mg = abs(f_transmitted_mg);
 t_end = length(transmitted_m)./fs_new; t =
linspace(0,t_end, length(transmitted_m));

  figure (2)
subplot(4,1,1)
plot(t,msg_resampled)
title('Resampled Msg signal in Time domain')

f_resampled=fftshift(fft(msg_resampled));
f_resampled_mg=abs(f_resampled); N =
length(f_transmitted_mg); f_vec =
linspace(-fs_new/2,fs_new/2,N);
 subplot(4,1,2)
```

```
plot(f_vec,f_resampled_mg)
title('Resampled Msg signal in Freq domain')

subplot(4,1,3) plot(f_vec,abs(f_transmitted_mg))
title('SSB-SC TX modulated msg in Freq domain')

subplot(4,1,4) plot(t,transmitted_m)
title('SSB-SC TX modulated msg in Time domain')
```



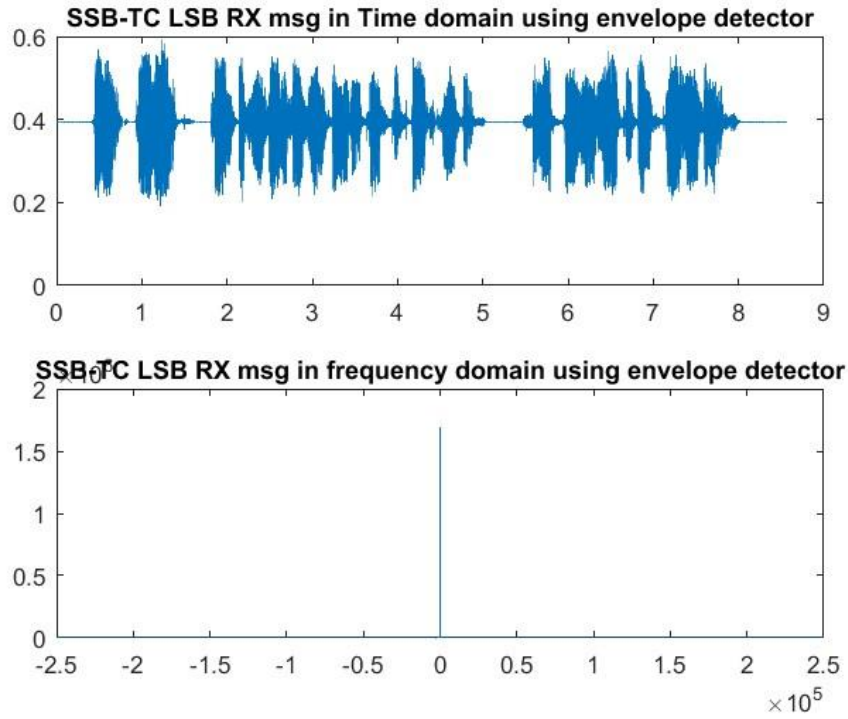
Zoomed version of the SSB-SC in frequency domain



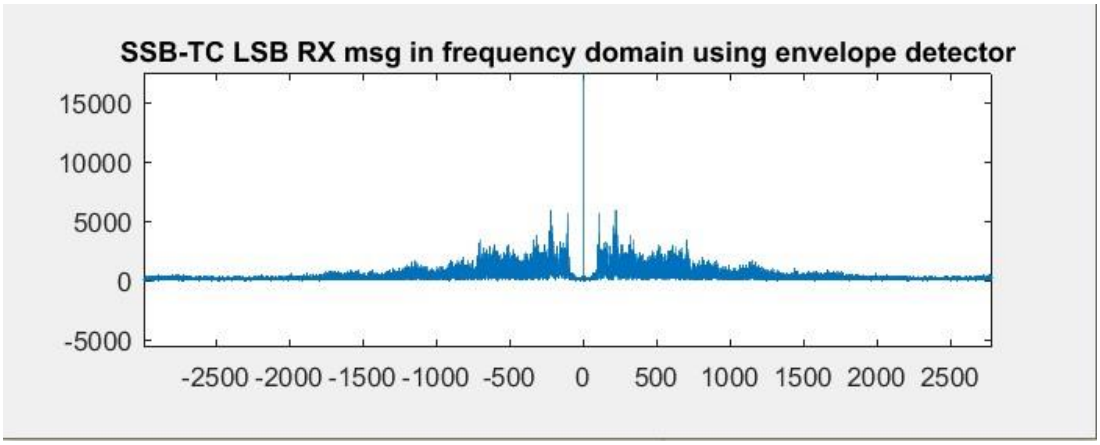
receiver

```
transmitted_m = real(ifft(ifftshift(f_transmitted_m)));
t_end = length(transmitted_m)./fs_new; t =
linspace(0,t_end, length(transmitted_m)); envelope =
abs(hilbert(transmitted_m)); figure(3) subplot(2,1,1)
plot(t,envelope)
```

```
title('SSB-TC LSB RX msg in Time domain using envelope detector')
f_envelope= fftshift(fft(envelope)); f_envelope_mg=
abs(f_envelope); subplot(2,1,2) plot(f_vec,f_envelope_mg)
title('SSB-TC LSB RX msg in frequency domain using envelope detector')
original_msg = resample(envelope,fs,fs_new); sound(original_msg, fs)
```

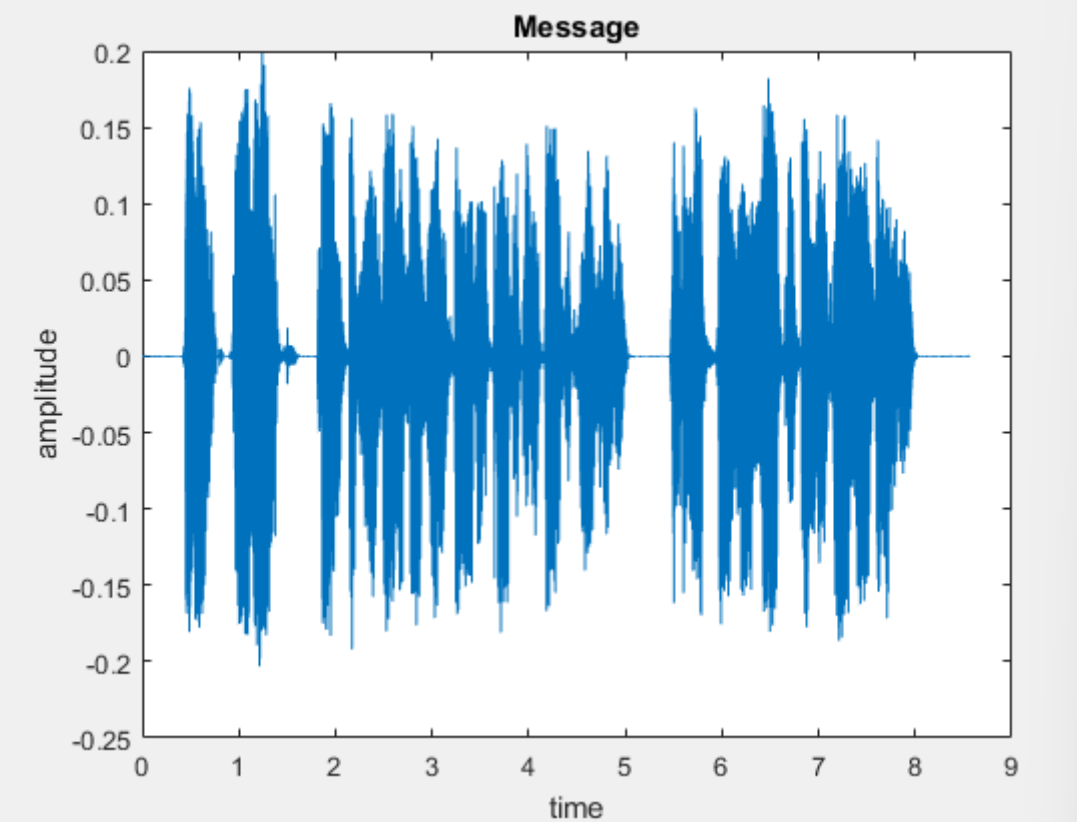


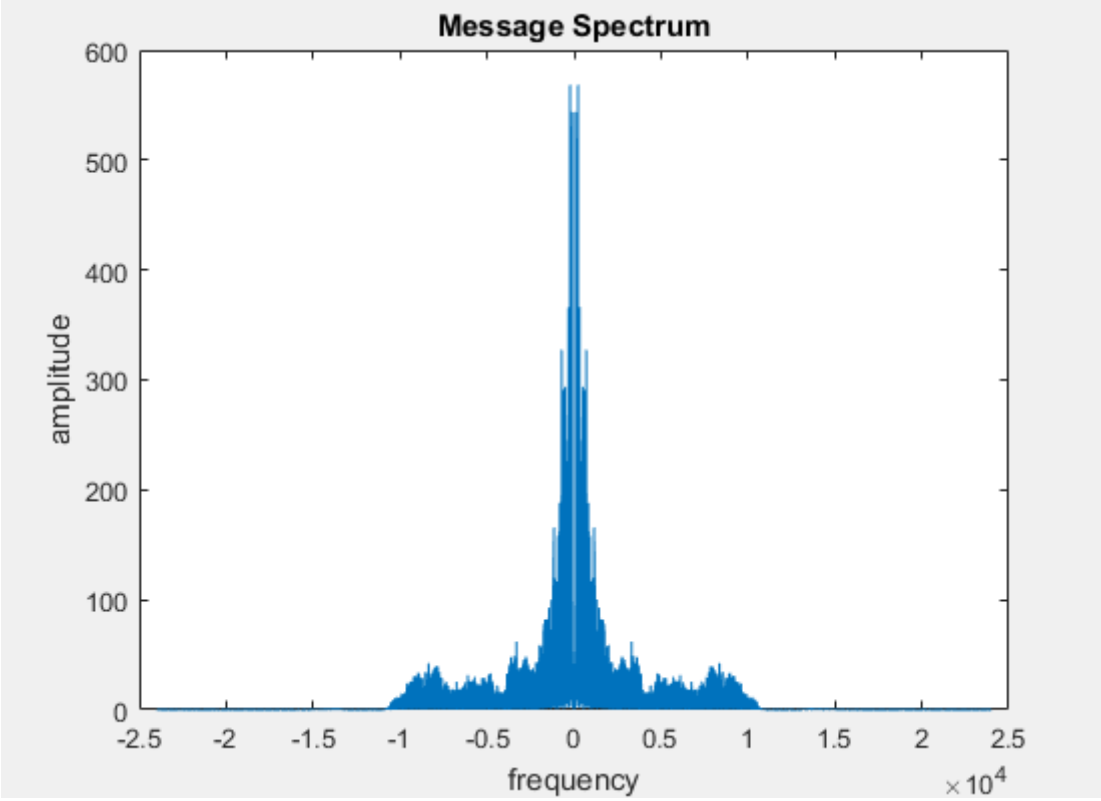
Zoomed SSB-TC after envelope detector

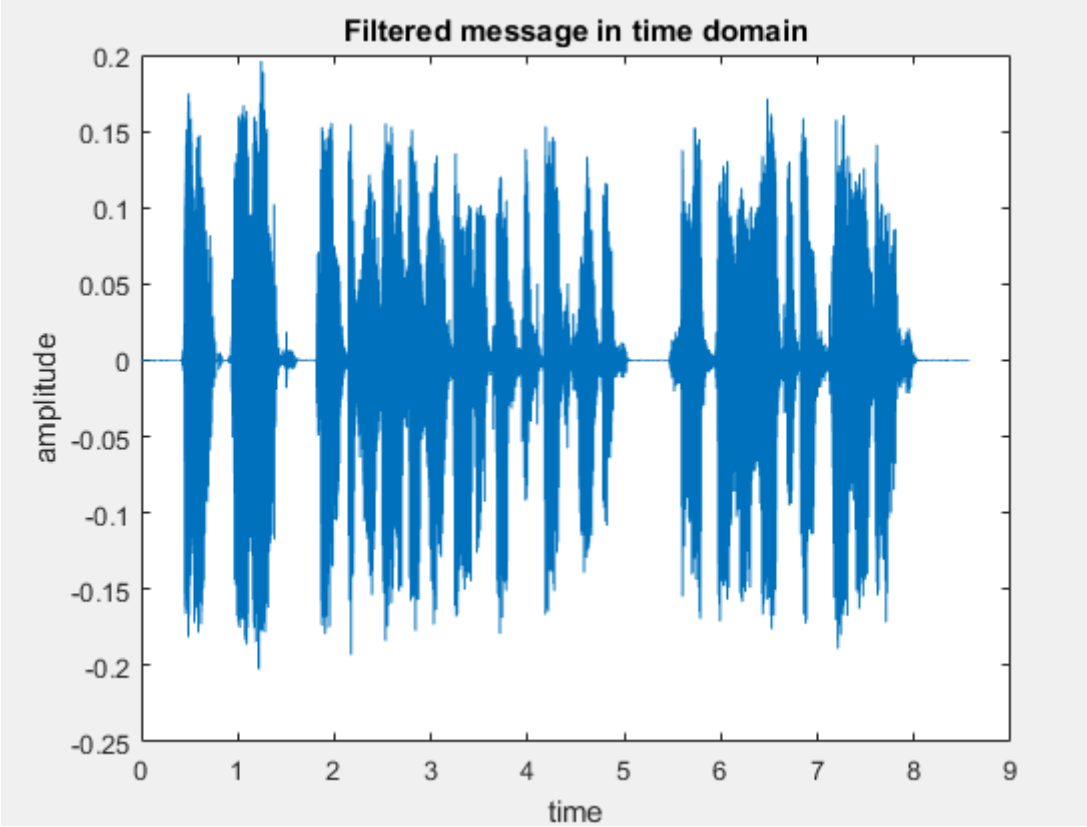


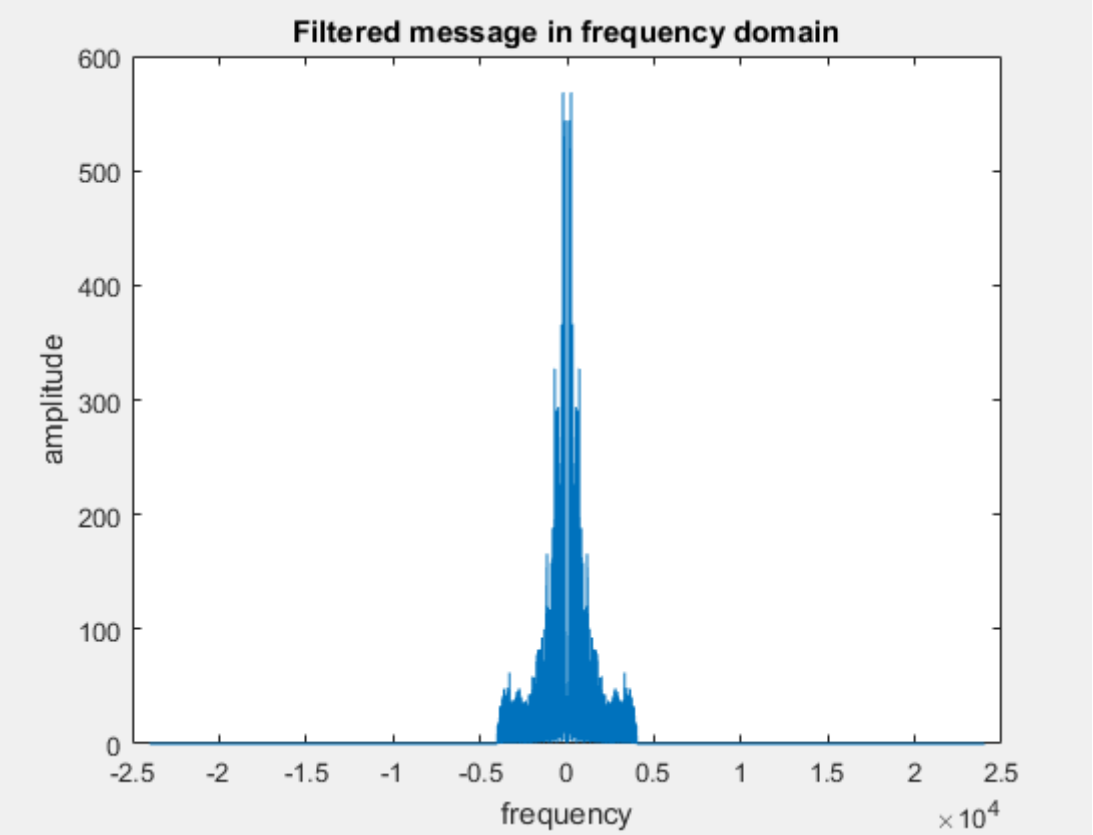
Experiment 3

```
%--------------------initializing-----------------------%
clc;
clear all;
close all;

%-----------------Uploading message---------------------%
[message,Fs]=audioread('eric.wav');

%-------------------Plotting message--------------------%

t    = linspace(0,(length(message)/Fs),length(message));
freq = linspace(-Fs/2,Fs/2,length(message));

MESSAGE=fftshift(fft(message));
figure;
plot(t,message);title 'Message';
xlabel 'time';ylabel 'amplitude';

figure;
plot(freq,abs(MESSAGE));title 'Message Spectrum';
xlabel 'frequency';ylabel 'amplitude';

%-----------------Filtering and Plotting---------------%

%constructing the filtered message
LPF  = [zeros(1,171354) ones(1,68541) zeros(1,171353)];
MESSAGE_filtered = LPF'.*MESSAGE;
message_filtered = ifft(ifftshift(MESSAGE_filtered));

%Ploting the filtered signal in time and frequency domain
figure;
plot(t,message_filtered);title 'Filtered message in time domain';
xlabel 'time';ylabel 'amplitude';
figure;
plot(freq,abs(MESSAGE_filtered));title 'Filtered message in frequency domain';
xlabel 'frequency';ylabel 'amplitude';
sound(message_filtered,Fs)
pause(8)
```

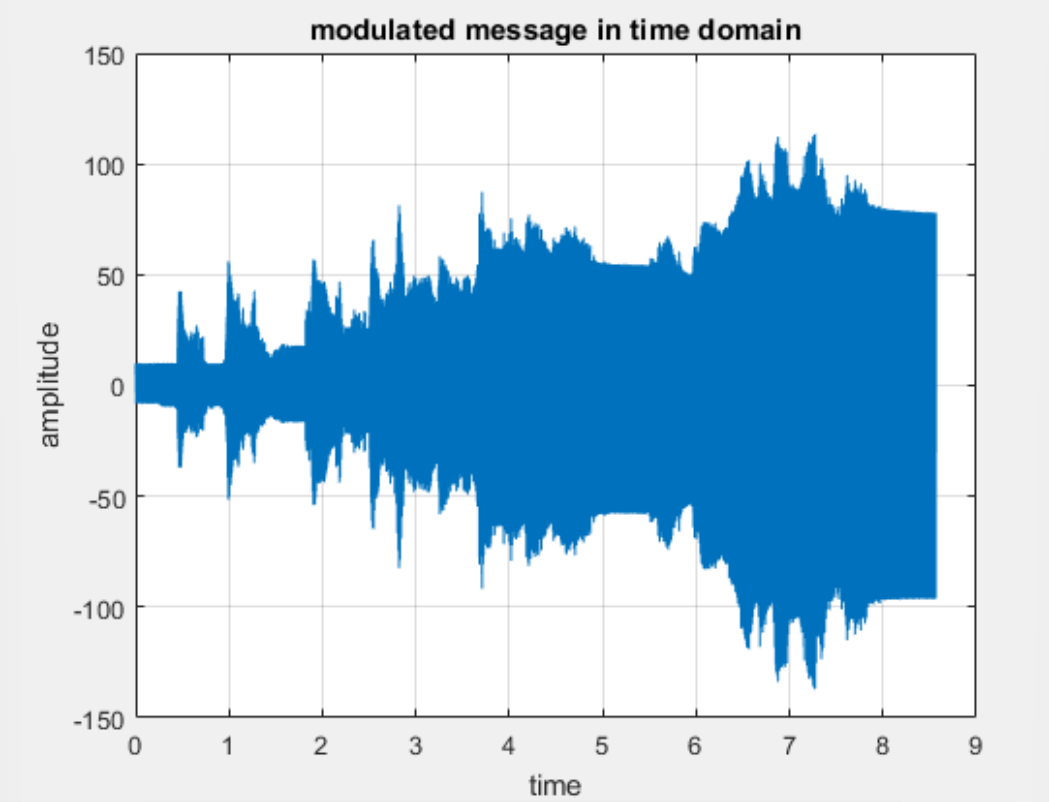

Message
amplitude
time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
7
8
9

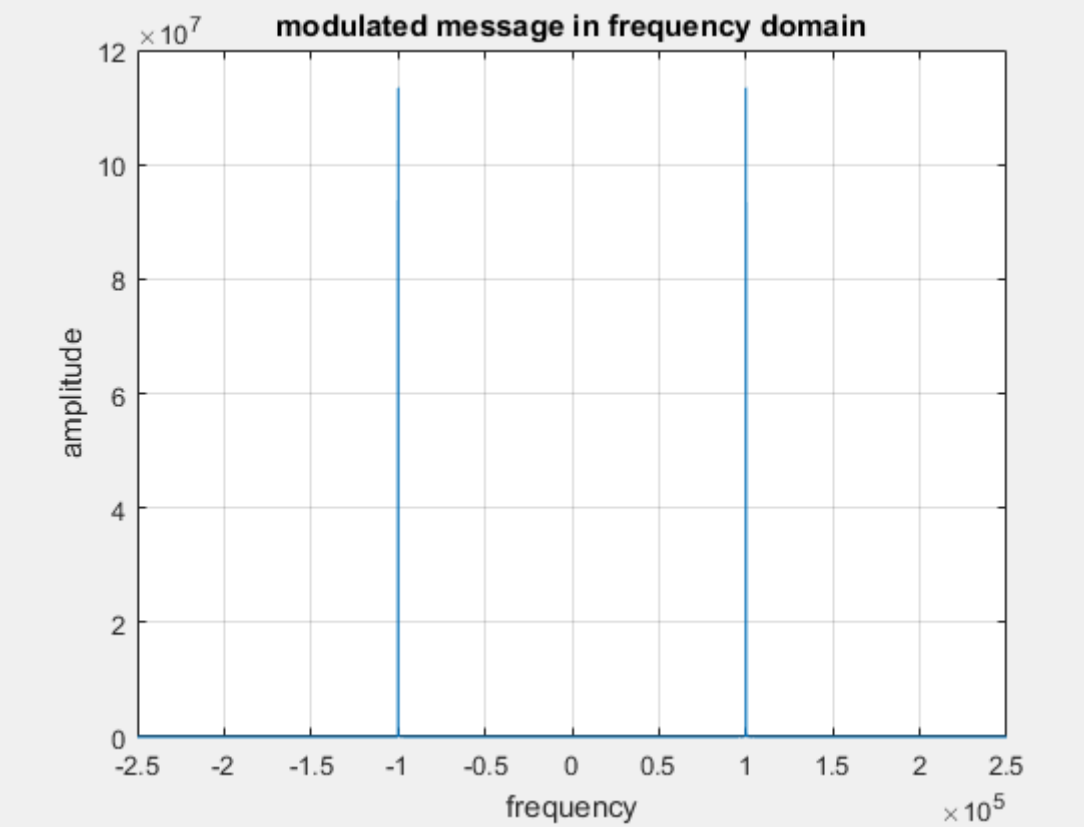

Message Spectrum
amplitude
frequency
×10^4
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


Filtered message in time domain
amplitude
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

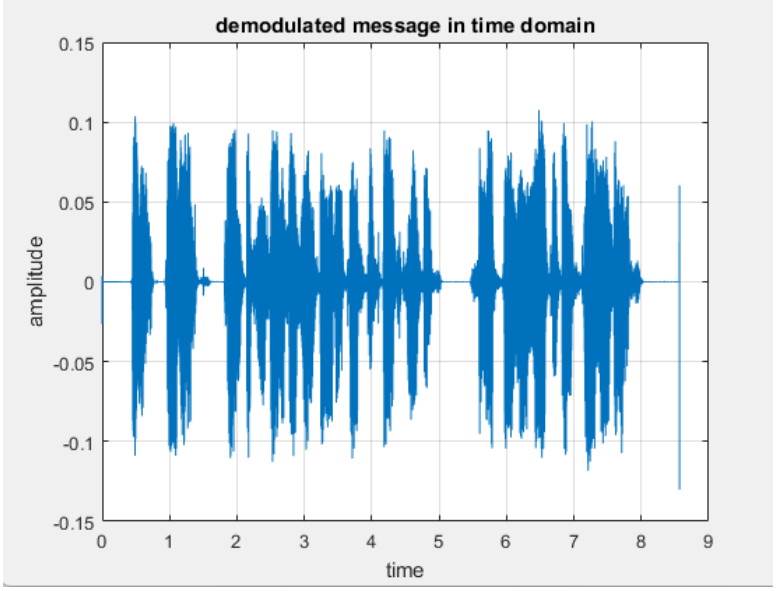

Filtered message in frequency domain
amplitude
frequency
×10^4
600
500
400
300
200
100
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

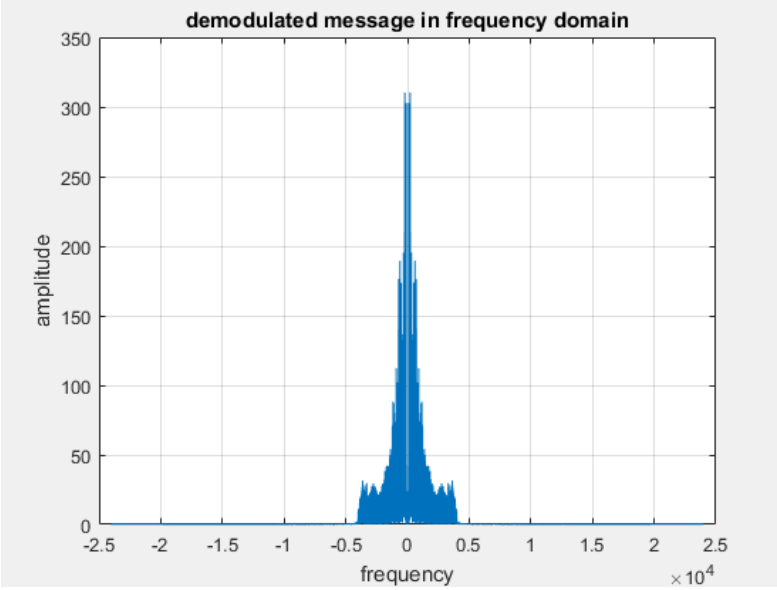
```
%--------------------Modulation----------------------%

%initialzing constants
fc=100000;
new_fs=5*fc;
kf=.2*pi;
A=10;



new_message=resample(message_filtered,new_fs,Fs);
durationofmessage=length(message_filtered)./Fs;
t=linspace(0,durationofmessage,length(new_message));

%integration of the message
integrate_me=cumsum(new_message);
integral_transpose=integrate_me.';

%modulation equation
modulated_signal=A*cos(2*pi.*fc*t)-kf.*integral_transpose.*sin(2*pi*fc.*t);

spectrum=fftshift(fft(modulated_signal));
f=linspace(-(new_fs)/2,(new_fs)/2,length(modulated_signal));

%plotting
figure;
plot(t,modulated_signal); grid on;title 'modulated message in time domain';
xlabel 'time';ylabel 'amplitude';

figure;
plot(f,abs(spectrum)); grid on;title 'modulated message in frequency domain';
xlabel 'frequency';ylabel 'amplitude';
```

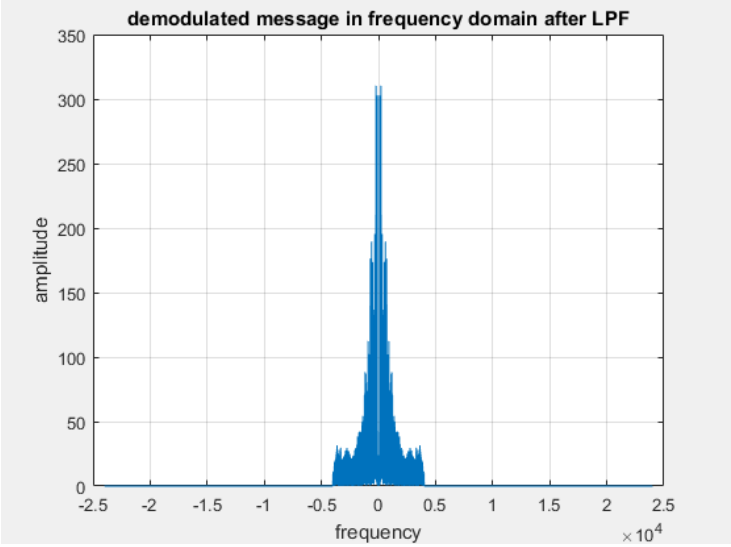

modulated message in time domain
amplitude
time
150
100
50
0
-50
-100
-150
0
1
2
3
4
5
6
7
8
9


modulated message in frequency domain
×10^7
amplitude
frequency
×10^5
12
10
8
6
4
2
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

```
%-------------------Demodulation--------------------%

%envelope detection and dc blocking
envelope=abs(hilbert(modulated_signal));
mm=diff(envelope);
receiver=resample(mm,Fs,new_fs);
receiver=receiver(2:end);
receiver_F=fftshift(fft(receiver));
receiver_F_mg=abs(receiver_F);
time=linspace(0,durationofmessage,length(receiver));

%plotting
figure;
plot(time,receiver); grid on;title 'demodulated message in time domain';
xlabel 'time';ylabel 'amplitude';


fx=linspace((-Fs/2),(Fs/2),length(receiver));
figure;
plot(fx,receiver_F_mg); grid on;title 'demodulated message in frequency domain';
xlabel 'frequency';ylabel 'amplitude';

%------------ LPF --------------%
N = length(receiver);
n = N/Fs;
right_band = round((Fs/2-4000)*n);
left_band = (N-right_band+1);
receiver_F([1:right_band left_band:N]) = 0;
receiver_LPF = real(ifft(ifftshift(receiver_F)));
%------------ LPF plotting --------------%
figure;
plot(fx,abs(receiver_F)); grid on;title 'demodulated message in frequency domain after LPF';
xlabel 'frequency';ylabel 'amplitude';
%-------------------End result----------------------%
sound(receiver_LPF,Fs);
```


demodulated message in time domain
amplitude
time
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1
2
3
4
5
6
7
8
9


demodulated message in frequency domain
amplitude
frequency
×10^4
350
300
250
200
150
100
50
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


demodulated message in frequency domain after LPF
amplitude
frequency
×10^4
350
300
250
200
150
100
50
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

Comments:

- 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.
- what FM gains in noise immunity lacks in bandwidth efficiency. Since FM usually occupies larger bandwidth, AM is considered more bandwidth wise.
- The output signal of NBFM is same as the original signal but it contains little noise.
- In NBFM, the BW of the modulated signal is equal to 2fm, this means that the spectrum, modulator, and demodulator are same as DSB-Tc.