# ANALOG FINAL PROJECT

| IDs | Names |
|---|---|
| **18010248** | أحمد محمود محمد الدقماق |
| **18010459** | باسل مصطفى سليم سليمان علي |
| **18011912** | مي بنداري محمود سليمان |
| **18011918** | ميار جمال أحمد سعد صمادة |

COMM DEP.
Faculty of engineering

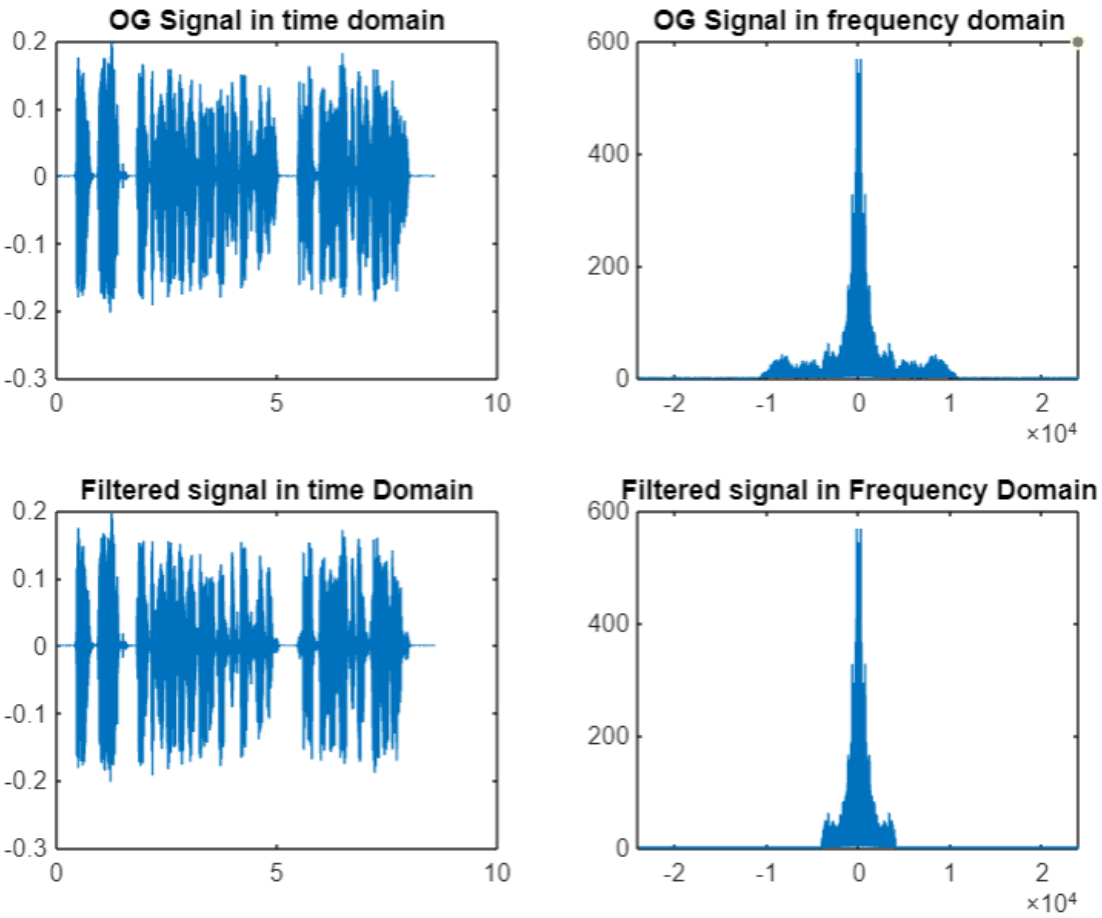
# EXPERIMENT ONE: DOUBLE SIDEBAND MODULATION

## (Points :1,2,3,4)

- **The code :**

```
%-------------------------------------------------------------------------
%-------------------------1-READ AUDIO-----------------------------------
%-------------------------------------------------------------------------
[y , fs] = audioread('eric.wav');
Y = fftshift(fft(y)); % to get spectrum of original signal
t = linspace(0,length(y)/fs,length(y));
figure; subplot(2,2,1) %PLOTTING OG AUDIO SIGNAL IN TIME DOMAIN
plot(t, y); title('OG Signal in time domain');
f = linspace(-fs/2,fs/2,length(Y));
subplot(2,2,2) %PLOTTING OG AUDIO SIGNAL IN FREQ DOMAIN
plot(f,abs(Y)); title('OG Signal in frequency domain');


%-------------------------------------------------------------------------
%------------2,3,4-IDEAL FILTER TO REMOVE FREQUENCES ABOVE 4 KHZ-----------
%-------------------------------------------------------------------------
Filter = rectpuls(f, 8000);
Filtered_Y = Y.* transpose(Filter); %transpose to be the same size
t=linspace(0,length(y)/fs,length(y));
subplot(2,2,4)  %PLOTTING FILTERED SIGNAL IN FREQ DOMAIN
plot(f,abs(Filtered_Y)); title('Filtered signal in Frequency Domain');
Filtered_y = real(ifft(ifftshift(Filtered_Y)));
subplot(2,2,3)  %PLOTTING FILTERED SIGNAL IN TIME DOMAIN
plot(t,Filtered_y); title('Filtered signal in time Domain');
player = audioplayer(Filtered_y,fs);
play(player)
```

- **The Plots :**



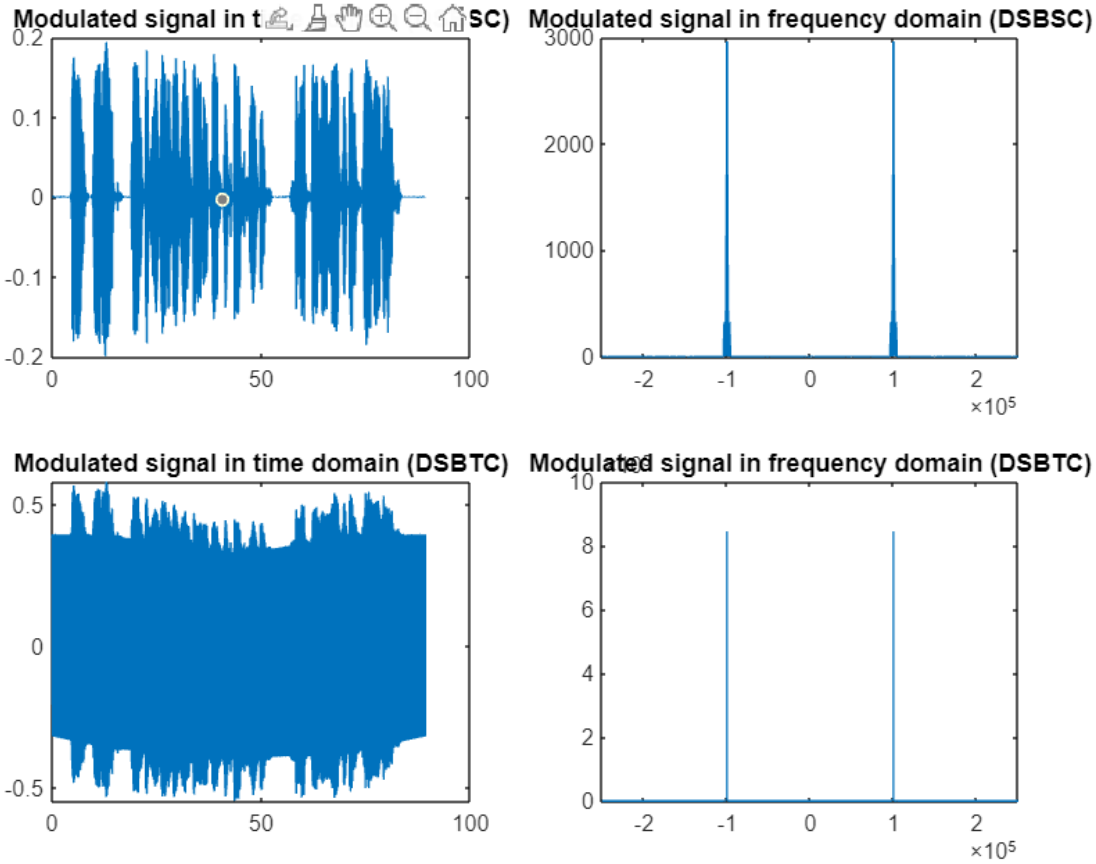

- **Comment :** we generate a rect pulse then we multiply the message with it so we have the filtered message
- The range of signal in freq domain after filtering is from -0.4*104 to 0.4*104

## (Point : 5)

- **The code :**

```
%--------------------------------------------------------------------------
%-------------------5-MODULATION OF DSBSC AND DSBTC-------------------------
%--------------------------------------------------------------------------
%Modulation for DSBSC
RSMBLE_DSBSC = resample(Filtered_y , 125 , 12);
t_resample = linspace(0,length(RSMBLE_DSBSC)/500000,length(RSMBLE_DSBSC));
DSBSC_T = transpose(RSMBLE_DSBSC).*cos(2*pi*100*1000*t_resample);
t1 = linspace(0,length(DSBSC_T)/fs,length(DSBSC_T));
DSBSC_F = fftshift(fft(DSBSC_T));
f_resample = linspace(-250000,250000,length(RSMBLE_DSBSC));
figure; subplot(2,2,1)
plot(t1 , DSBSC_T); title('Modulated signal in time domain (DSBSC)');
subplot(2,2,2)
plot(f_resample , abs(DSBSC_F)); title('Modulated signal in frequency domain (DSBSC)');
%Modulation of DSBTC
A = 2*max(RSMBLE_DSBSC);
DSBTC_T = A* cos(2*pi*100*1000*t_resample) + transpose(RSMBLE_DSBSC).*cos(2*pi*100*1000*t_resample);
t2 = linspace(0,length(DSBTC_T)/fs,length(DSBTC_T));
DSBTC_F = fftshift(fft(DSBTC_T));
subplot(2,2,3)
plot(t2 , DSBTC_T); title('Modulated signal in time domain (DSBTC)');
subplot(2,2,4)
plot(f_resample , abs(DSBTC_F)); title('Modulated signal in frequency domain (DSBTC)');
```

- **The Plots :**




- **Comments :**

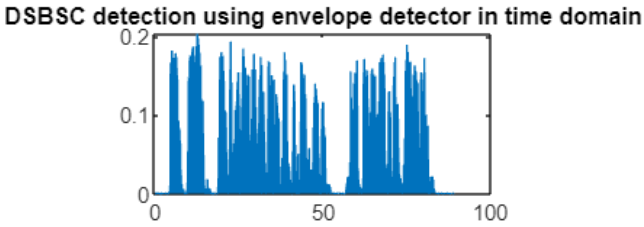
  - At first we resample the message then to do the modulation we multiply it with COS
  - At SC we get the message only ( upper side & lower side )
  - At TC we get the message + the carrier

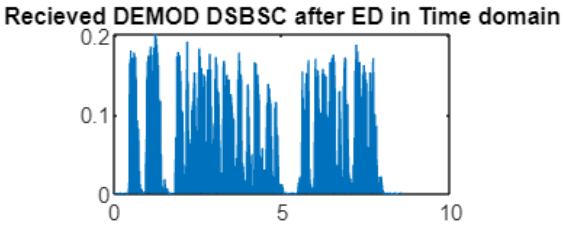
## (Points :6,7)

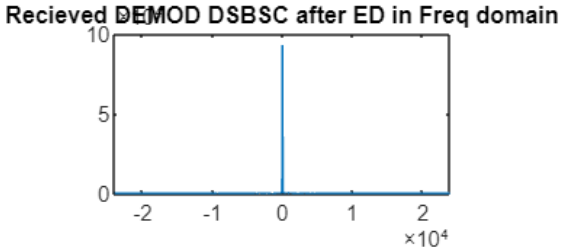
- **The code**

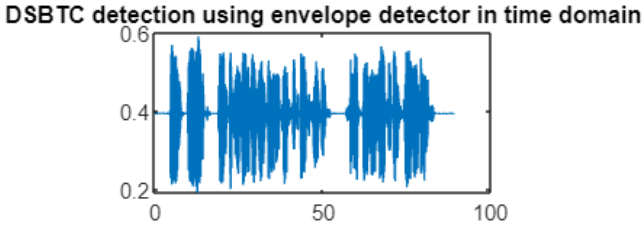
```
%--------------------------------------------------------------------------
%------------------6,7-DETECTION USNIG ENVELOPE DETECTOR------------------
%--------------------------------------------------------------------------
%Demodulation for DSBSC using envelope detector
ED_DSBSC = abs(hilbert(DSBSC_T));
figure; subplot(4,3,1)
plot(t1 , ED_DSBSC); title('DSBSC detection using envelope detector in time domain');
restored_DSBSC_T = resample(ED_DSBSC,12,125);
restored_DSBSC_F = fftshift(fft(ED_DSBSC));
t4 = linspace(0,length(restored_DSBSC_T)/fs,length(restored_DSBSC_T));
F_4 = linspace(-fs/2,fs/2,length(restored_DSBSC_F));
subplot(4,3,3)
plot(t4,restored_DSBSC_T); title('Recieved DEMOD DSBSC after ED in Time domain');
subplot(4,3,5)
plot(F_4,abs(restored_DSBSC_F)); title('Recieved DEMOD DSBSC after ED in Freq domain');
audiowrite('Envlope_DSBSC.wav',restored_DSBSC_T,fs);

%Demodulation for DSBTC using envelope detector
ED_DSBTC = abs(hilbert(DSBTC_T));
ED_DSBTC_WITHOUT_DC = ED_DSBTC - A;
subplot(4,3,7)
plot(t2 , ED_DSBTC); title('DSBTC detection using envelope detector in time domain');
restored_DSBTC_T = resample(ED_DSBTC_WITHOUT_DC,12,125);
restored_DSBTC_F = fftshift(fft(ED_DSBTC_WITHOUT_DC));
t3 = linspace(0,length(restored_DSBTC_T)/fs,length(restored_DSBTC_T));
F_3 = linspace(-fs/2,fs/2,length(restored_DSBTC_F));
subplot(4,3,9)
plot(t3,restored_DSBTC_T); title('Recieved DEMOD DSBTC after ED and removing DC in Time domain');
subplot(4,3,11)
plot(F_3,abs(restored_DSBTC_F)); title('Recieved DEMOD DSBTC after ED and removing DC in Time domain');
audiowrite('Envlope_DSBTC.wav',restored_DSBTC_T,fs);
```

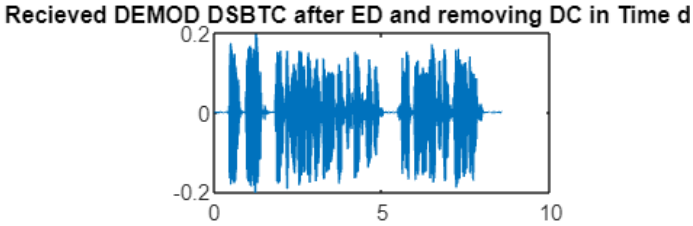
- **The Plots :**

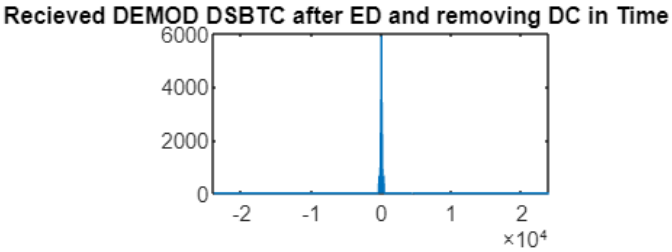



- **Comments:**
  - **After demodulation with envelope detector we get the original message**

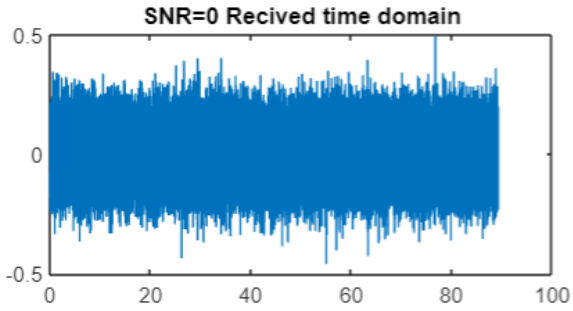
  - **We use Hilbert to get the envelope detector**

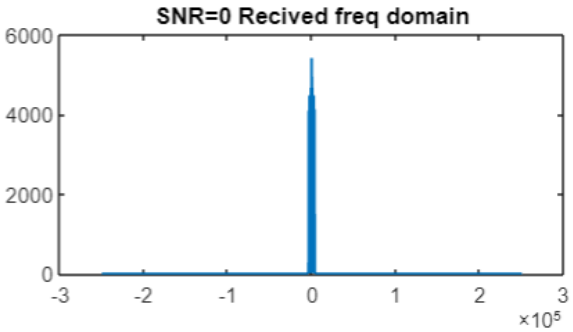
# (Point : 8)

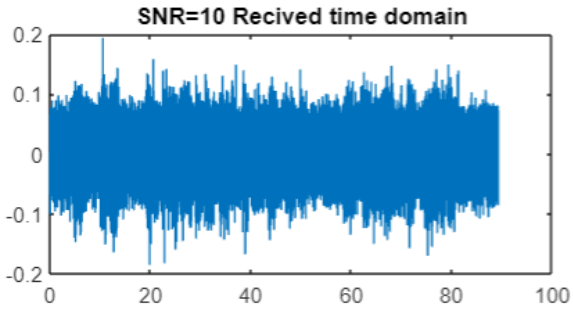
- **The code :**

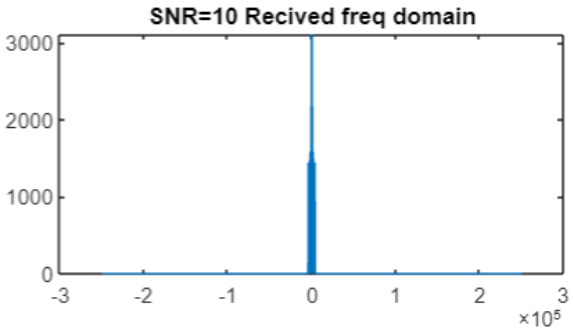
```
%--------------------------------------------------------------------------
%------------------8-DEMODULATION IN PRESENCE OF NOISE--------------------
%--------------------------------------------------------------------------
filter_2 = rectpuls(f_resample , 8000);

NOISE_0 = awgn(DSBSC_T,0);
NOISE_0_T = NOISE_0 .* cos(2*pi*100*1000*t_resample);
NOISE_0_F = fftshift(fft(NOISE_0_T));
NOISE_0_F_R = NOISE_0_F .* filter_2;
NOISE_0_T_R = real(ifft(ifftshift(NOISE_0_F_R)));
figure; subplot(3,2,1);plot(t1,NOISE_0_T_R); title('SNR=0 Recived time domain');
subplot(3,2,2);plot(f_resample,abs(NOISE_0_F_R)); title('SNR=0 Recived freq domain');
NOISE_0_T_R = resample(NOISE_0_T_R,12,125);
audiowrite('SNR=0.wav',NOISE_0_T_R,fs);

NOISE_10 = awgn(DSBSC_T,10);
NOISE_10_T = NOISE_10 .* cos(2*pi*100*1000*t_resample);
NOISE_10_F = fftshift(fft(NOISE_10_T));
NOISE_10_F_R = NOISE_10_F .* filter_2;
NOISE_10_T_R = real(ifft(ifftshift(NOISE_10_F_R)));
subplot(3,2,3);plot(t1,NOISE_10_T_R); title('SNR=10 Recived time domain');
subplot(3,2,4);plot(f_resample,abs(NOISE_10_F_R)); title('SNR=10 Recived freq domain');
NOISE_10_T_R = resample(NOISE_10_T_R,12,125);
audiowrite('SNR=10.wav',NOISE_10_T_R,fs);

NOISE_30 = awgn(DSBSC_T,30);
NOISE_30_T = NOISE_30 .* cos(2*pi*100*1000*t_resample);
NOISE_30_F = fftshift(fft(NOISE_30_T));
NOISE_30_F_R = NOISE_30_F .* filter_2;
NOISE_30_T_R = real(ifft(ifftshift(NOISE_30_F_R)));
subplot(3,2,5);plot(t1,NOISE_30_T_R); title('SNR=30 Recived time domain');
subplot(3,2,6);plot(f_resample,abs(NOISE_30_F_R)); title('SNR=30 Recived freq domain');
NOISE_30_T_R = resample(NOISE_30_T_R,12,125);
audiowrite('SNR=30.wav',NOISE_30_T_R,fs);
```

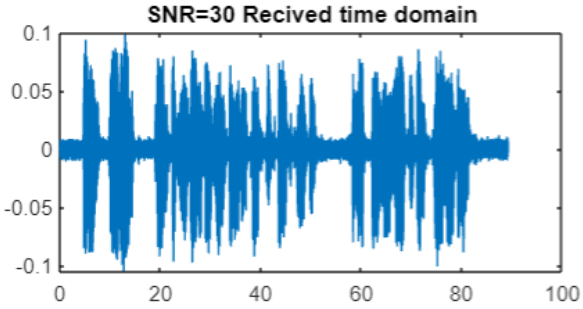
- **The Plots :**

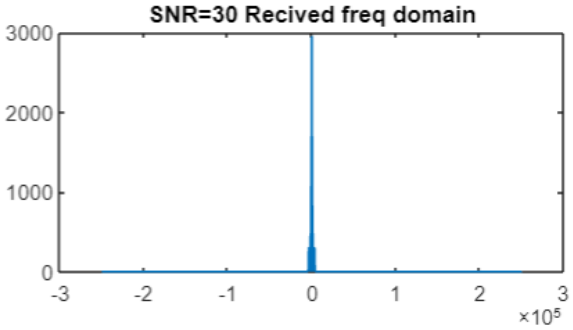

SNR=0 Recived time domain
0.5
0
-0.5
0
20
40
60
80
100


SNR=0 Recived freq domain
6000
4000
2000
0
-3
-2
-1
0
1
2
3
×10^5


SNR=10 Recived time domain
0.2
0.1
0
-0.1
-0.2
0
20
40
60
80
100


SNR=10 Recived freq domain
3000
2000
1000
0
-3
-2
-1
0
1
2
3
×10^5


SNR=30 Recived time domain
0.1
0.05
0
-0.05
-0.1
0
20
40
60
80
100


SNR=30 Recived freq domain
3000
2000
1000
0
-3
-2
-1
0
1
2
3
×10^5

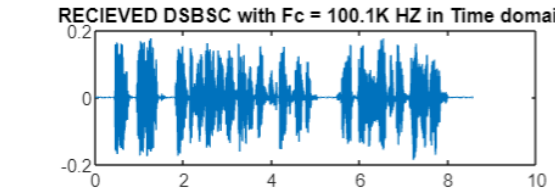
# (Points : 9,10)

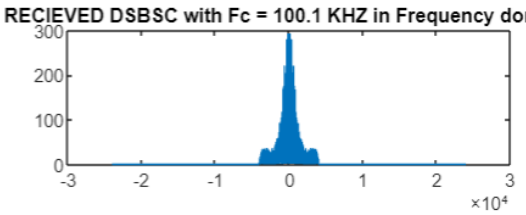
- **The code :**

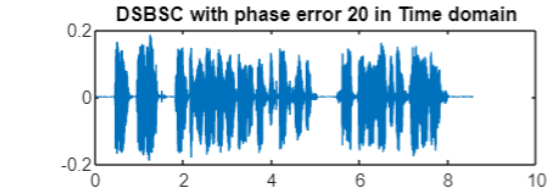
```
%--------------------------------------------------------------------------
%------------------9-10DEMODULATION IN PRESENCE OF NOISE-------------------
%--------------------------------------------------------------------------
%DSBDC with freq error = 100.1 instead of 100
carrier_error   =  cos(2*pi*100.1*1000*t_resample);
DSBSC_T_DEMOD_ERROR = DSBSC_T .* carrier_error;
DSBSC_F_DEMOD_ERROR =  fftshift(fft(DSBSC_T_DEMOD_ERROR ));
DSBSC_filtered_F_ERROR =  DSBSC_F_DEMOD_ERROR.*filter_2;
DSBSC_filtered_T_ERROR =  real(ifft(ifftshift(DSBSC_filtered_F_ERROR)));
DSBSC_filtered_T_ERROR =  2*DSBSC_filtered_T_ERROR;
recieved_filtered_T_ERROR = resample(DSBSC_filtered_T_ERROR,12,125);
recieved_filtered_F_ERROR = fftshift(fft(recieved_filtered_T_ERROR ));
figure; subplot(4,2,1)
plot(t3,recieved_filtered_T_ERROR); title('RECIEVED DSBSC with Fc = 100.1K HZ in Time domain ');
f2 = linspace(-fs/2,fs/2,length(recieved_filtered_F_ERROR));
subplot(4,2,2)
plot(f2,abs(recieved_filtered_F_ERROR)); title('RECIEVED DSBSC with Fc = 100.1 KHZ in Frequency domain ');
audiowrite('FREQ_ERROR.wav',recieved_filtered_T_ERROR,fs);

%DSBDC with phase error = 20
carrier_error_2  =  cos(2*pi*100*1000*t_resample+(20*pi/180));
DSBSC_T_DEMOD_ERROR_2 = DSBSC_T .* carrier_error_2;
DSBSC_F_DEMOD_ERROR_2 =  fftshift(fft(DSBSC_T_DEMOD_ERROR_2));
DSBSC_filtered_F_ERROR_2 =  DSBSC_F_DEMOD_ERROR_2.*filter_2;
DSBSC_filtered_T_ERROR_2 =  real(ifft(ifftshift(DSBSC_filtered_F_ERROR_2)));
DSBSC_filtered_T_ERROR_2 =  2*DSBSC_filtered_T_ERROR_2;
recieved_filtered_T_ERROR_2 = resample(DSBSC_filtered_T_ERROR_2,12,125);
recieved_filtered_F_ERROR_2 = fftshift(fft(recieved_filtered_T_ERROR_2));
subplot(4,2,3)
plot(t3,recieved_filtered_T_ERROR_2); title('DSBSC with phase error 20 in Time domain');
f3 = linspace(-fs/2,fs/2,length(recieved_filtered_F_ERROR_2));
subplot(4,2,4)
plot(f3,abs(recieved_filtered_F_ERROR_2)); title('DSBSC with phase error 20 in Frequency domain');
audiowrite('PAHSE_ERROR.wav',recieved_filtered_T_ERROR_2,fs);
```

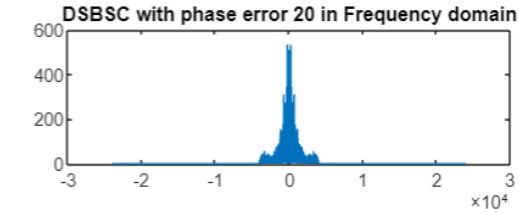
```
%DSBDC with freq error = 100.1 instead of 100 AND SNR = 30
carrier_error   =  cos(2*pi*100.1*1000*t_resample);
DSBSC_T_DEMOD_ERROR_SNR30 = NOISE_30 .* carrier_error;
DSBSC_F_DEMOD_ERROR_SNR30 =  fftshift(fft(DSBSC_T_DEMOD_ERROR_SNR30 ));
DSBSC_filtered_F_ERROR_SNR30 =  DSBSC_F_DEMOD_ERROR_SNR30.*filter_2;
DSBSC_filtered_T_ERROR_SNR30 =  real(ifft(ifftshift(DSBSC_filtered_F_ERROR_SNR30)));
DSBSC_filtered_T_ERROR_SNR30 =  2*DSBSC_filtered_T_ERROR_SNR30;
recieved_filtered_T_ERROR_SNR30 = resample(DSBSC_filtered_T_ERROR_SNR30,12,125);
recieved_filtered_F_ERROR_SNR30 = fftshift(fft(recieved_filtered_T_ERROR_SNR30 ));
subplot(4,2,5)
plot(t3,recieved_filtered_T_ERROR_SNR30); title('RECIEVED DSBSC with Fc = 100.1KHZ and SNR=30 in Time domain');
f2 = linspace(-fs/2,fs/2,length(recieved_filtered_F_ERROR_SNR30));
subplot(4,2,6)
plot(f2,abs(recieved_filtered_F_ERROR_SNR30)); title('RECIEVED DSBSC with Fc = 100.1 KHZ and SNR=30 in Frequency domain ');
audiowrite('FREQ_ERROR_SNR=30.wav',recieved_filtered_T_ERROR_SNR30,fs);

%DSBDC with phase error = 20 AND SNR = 30
carrier_error_2  =  cos(2*pi*100*1000*t_resample+(20*pi/180));
DSBSC_T_DEMOD_ERROR_2_SNR30 = NOISE_30 .* carrier_error_2;
DSBSC_F_DEMOD_ERROR_2_SNR30 =  fftshift(fft(DSBSC_T_DEMOD_ERROR_2_SNR30));
DSBSC_filtered_F_ERROR_2_SNR30 =  DSBSC_F_DEMOD_ERROR_2_SNR30.*filter_2;
DSBSC_filtered_T_ERROR_2_SNR30 =  real(ifft(ifftshift(DSBSC_filtered_F_ERROR_2_SNR30)));
DSBSC_filtered_T_ERROR_2_SNR30 =  2*DSBSC_filtered_T_ERROR_2_SNR30;
recieved_filtered_T_ERROR_2_SNR30 = resample(DSBSC_filtered_T_ERROR_2_SNR30,12,125);
recieved_filtered_F_ERROR_2_SNR30 = fftshift(fft(recieved_filtered_T_ERROR_2_SNR30));
subplot(4,2,7)
plot(t3,recieved_filtered_T_ERROR_2_SNR30); title('DSBSC with phase error 20 and SNR=30 in Time domain');
f3 = linspace(-fs/2,fs/2,length(recieved_filtered_F_ERROR_2_SNR30));
subplot(4,2,8)
plot(f3,abs(recieved_filtered_F_ERROR_2_SNR30)); title('DSBSC with phase error 20 and SNR=30 in Frequency domain');
audiowrite('PAHSE_ERROR_SNR30.wav',recieved_filtered_T_ERROR_2_SNR30,fs);
%----------------------END-------------------------------------------------
```

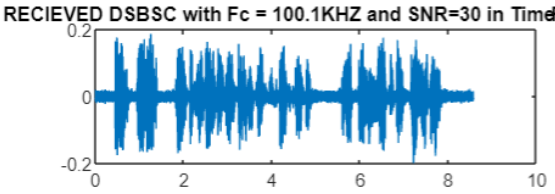
- **The Plots :**

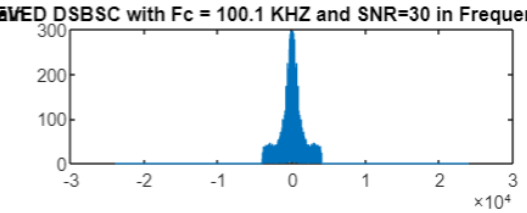

RECIEVED DSBSC with Fc = 100.1K HZ in Time domain
0.2
0
-0.2
0
2
4
6
8
10

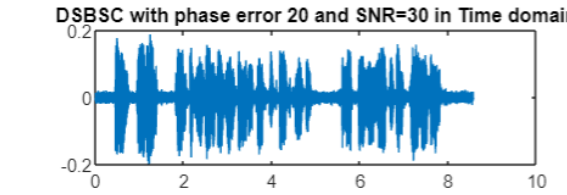

RECIEVED DSBSC with Fc = 100.1 KHZ in Frequency domain
300
200
100
0
-3
-2
-1
0
1
2
3
×10⁴

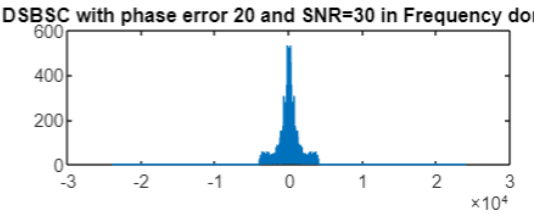

DSBSC with phase error 20 in Time domain
0.2
0
-0.2
0
2
4
6
8
10


DSBSC with phase error 20 in Frequency domain
600
400
200
0
-3
-2
-1
0
1
2
3
×10⁴


RECIEVED DSBSC with Fc = 100.1KHZ and SNR=30 in Time domain
0.2
0
-0.2
0
2
4
6
8
10


RECIEVED DSBSC with Fc = 100.1 KHZ and SNR=30 in Frequency domain
300
200
100
0
-3
-2
-1
0
1
2
3
×10⁴


DSBSC with phase error 20 and SNR=30 in Time domain
0.2
0
-0.2
0
2
4
6
8
10


DSBSC with phase error 20 and SNR=30 in Frequency domain
600
400
200
0
-3
-2
-1
0
1
2
3
×10⁴

# EXPERIMENT TWO: SINGLE SIDEBAND MODULATION

## ⟶ (Points :1,2,3,4,5,6,7,8)

- **The code :**

```
%--------------------------------------------------------------------------
%-------------------------1-READ AUDIO-------------------------------------
%--------------------------------------------------------------------------
[y , fs] = audioread('eric.wav');
Y = fftshift(fft(y));
t = linspace(0,length(y)/fs,length(y));
figure; subplot(4,4,1)
plot(t, y); title('OG Signal in time domain');
f = linspace(-fs/2,fs/2,length(Y));
subplot(4,4,2)
plot(f,abs(Y)); title('OG Signal in frequency domain');


%--------------------------------------------------------------------------
%------------2,3,-IDEAL FILTER TO REMOVE FREQUENCES ABOVE 4 KHZ-----------
%--------------------------------------------------------------------------
Filter = rectpuls(f, 8000);
Filtered_Y = Y.* transpose(Filter);
t=linspace(0,length(y)/fs,length(y));
subplot(4,4,4)
plot(f,abs(Filtered_Y)); title('Filtered signal in Frequency Domain');
Filtered_y = real(ifft(ifftshift(Filtered_Y)));
subplot(4,4,3)
plot(t,Filtered_y); title('Filtered signal in time Domain');
player = audioplayer(Filtered_y,fs);
play(player)
```

### Comment :

- we generate a rect pulse then we multiply the message with it so we have the filtered message
- The range of signal in freq domain after filtering is from -0.4*104 to 0.4*104

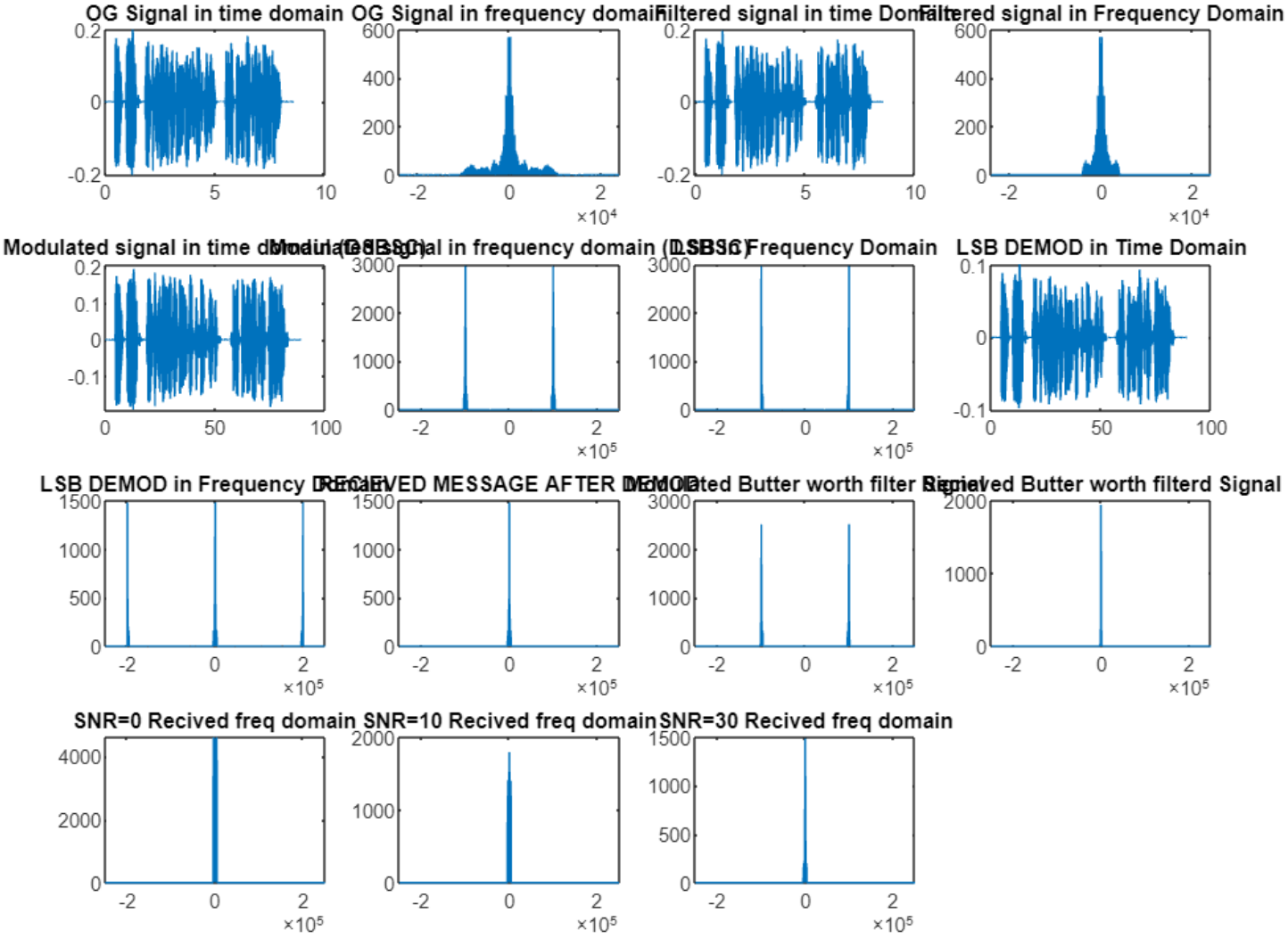
```
%-------------------------------------------------------------------------
%-------------------4-MODULATION OF DSBSC---------------------------------
%-------------------------------------------------------------------------
%Modulation for DSBSC
RSMBLE_DSBSC = resample(Filtered_y , 125 , 12);
t_resample = linspace(0,length(RSMBLE_DSBSC)/500000,length(RSMBLE_DSBSC));
DSBSC_T = transpose(RSMBLE_DSBSC) .* cos(2*pi*100*1000*t_resample);
t1 = linspace(0,length(DSBSC_T)/fs,length(DSBSC_T));
DSBSC_F = fftshift(fft(DSBSC_T));
f_resample = linspace(-250000,250000,length(DSBSC_T));
subplot(4,4,5)
plot(t1 , DSBSC_T); title('Modulated signal in time domain (DSBSC)');
subplot(4,4,6)
plot(f_resample , abs(DSBSC_F)); title('Modulated signal in frequency domain (DSBSC)
%Demodulation of DSBSC
DSBSC_demodulated_time = DSBSC_T .* cos(2*pi*100*1000*t_resample);
DSBSC_demodulated_freq = fftshift(fft(DSBSC_demodulated_time));
```

```
%-------------------------------------------------------------------------
%-----------------------------LSB-----------------------------------------
%-------------------------------------------------------------------------
fc=100e3;
Fs=500e3;
Y1 = DSBSC_F;
Filter1 = rectpuls(f_resample, 200000);
LSB = Y1 .* Filter1;
F_1=linspace(-Fs/2,Fs/2,length(LSB));
subplot(4,4,7);plot(F_1,abs(LSB)); title('LSB in Frequency Domain');
T_lsb = real(ifft(ifftshift(LSB)));
T_lsb_DEMOD = T_lsb .* cos(2*pi*100*1000*t_resample);
F_lsb_DEMOD = fftshift(fft(T_lsb_DEMOD));
F_2 = linspace(-Fs/2,Fs/2,length(F_lsb_DEMOD));
subplot(4,4,8);plot(t1,T_lsb_DEMOD); title('LSB DEMOD in Time Domain');
subplot(4,4,9);plot(F_2,abs(F_lsb_DEMOD)); title('LSB DEMOD in Frequency Domain');
filter_2 = rectpuls(F_2 , 8000);
RES_F_lsb_DEMOD = F_lsb_DEMOD .* filter_2;
RES_T_lsb_DEMOD = real(ifft(ifftshift(RES_F_lsb_DEMOD)));
RES_T_lsb_DEMOD = resample(RES_T_lsb_DEMOD,12,125);
F_3=linspace(-Fs/2,Fs/2,length(RES_F_lsb_DEMOD));
subplot(4,4,10);plot(F_3,abs(RES_F_lsb_DEMOD)); title('RECIEVED MESSAGE AFTER DEMOD');
audiowrite('LSB_SSB_Received.wav',RES_T_lsb_DEMOD,fs);
```

```
%-------------------------------------------------------------------------
%------------7-MODULATION OF DSBSC USING BUTTERWORTH FILTER--------------
%-------------------------------------------------------------------------
[B ,A] = butter(4,[fc-4000 fc]/(Fs/2));
Filterd_LSB_SSB_Butter = filter(B,A,DSBSC_T');
Filterd_LSB_SSB_Butter = Filterd_LSB_SSB_Butter';
Filterd_LSB_SSB_Butter_F = fftshift(fft(Filterd_LSB_SSB_Butter));
subplot(4,4,11);plot(f_resample,abs(Filterd_LSB_SSB_Butter_F)); title('Modulated Butter worth filter Signal');
%Reciving the Signal%
Filterd_LSB_SSB_Butter_R = Filterd_LSB_SSB_Butter .* cos(2*pi*100*1000*t_resample);
[B,A] =butter(4,4000/Fs/2);
Filterd_LSB_SSB_Butter_R = filter(B,A,Filterd_LSB_SSB_Butter_R);
Filterd_LSB_SSB_Butter_R_F = real(fftshift(fft(Filterd_LSB_SSB_Butter_R)));
subplot(4,4,12);plot(f_resample,abs(Filterd_LSB_SSB_Butter_R_F)); title('Recieved Butter worth filterd Signal');
Filterd_LSB_SSB_Butter_R =resample(Filterd_LSB_SSB_Butter_R,12,125);
audiowrite('Butter_ssb.wav',Filterd_LSB_SSB_Butter_R,fs);
```

```
%-------------------------------------------------------------------------
%-------------------8-DEMODULATION IN PRESENCE OF NOISE-------------------
%-------------------------------------------------------------------------
filter_2 = rectpuls(f_resample , 8000);

NOISE_0 = awgn(DSBSC_T,0);
NOISE_0_T = NOISE_0 .* cos(2*pi*100*1000*t_resample);
NOISE_0_F = fftshift(fft(NOISE_0_T));
NOISE_0_F_R = NOISE_0_F .* filter_2;
NOISE_0_T_R = real(ifft(ifftshift(NOISE_0_F_R)));
figure; subplot(3,2,1);plot(t1,NOISE_0_T_R); title('SNR=0 Recived time domain');
subplot(3,2,2);plot(f_resample,abs(NOISE_0_F_R)); title('SNR=0 Recived freq domain');
NOISE_0_T_R = resample(NOISE_0_T_R,12,125);
audiowrite('SNR=0.wav',NOISE_0_T_R,fs);

NOISE_10 = awgn(DSBSC_T,10);
NOISE_10_T = NOISE_10 .* cos(2*pi*100*1000*t_resample);
NOISE_10_F = fftshift(fft(NOISE_10_T));
NOISE_10_F_R = NOISE_10_F .* filter_2;
NOISE_10_T_R = real(ifft(ifftshift(NOISE_10_F_R)));
subplot(3,2,3);plot(t1,NOISE_10_T_R); title('SNR=10 Recived time domain');
subplot(3,2,4);plot(f_resample,abs(NOISE_10_F_R)); title('SNR=10 Recived freq domain');
NOISE_10_T_R = resample(NOISE_10_T_R,12,125);
audiowrite('SNR=10.wav',NOISE_10_T_R,fs);

NOISE_30 = awgn(DSBSC_T,30);
NOISE_30_T = NOISE_30 .* cos(2*pi*100*1000*t_resample);
NOISE_30_F = fftshift(fft(NOISE_30_T));
NOISE_30_F_R = NOISE_30_F .* filter_2;
NOISE_30_T_R = real(ifft(ifftshift(NOISE_30_F_R)));
subplot(3,2,5);plot(t1,NOISE_30_T_R); title('SNR=30 Recived time domain');
subplot(3,2,6);plot(f_resample,abs(NOISE_30_F_R)); title('SNR=30 Recived freq domain');
NOISE_30_T_R = resample(NOISE_30_T_R,12,125);
audiowrite('SNR=30.wav',NOISE_30_T_R,fs);
```

- **The Plots :**

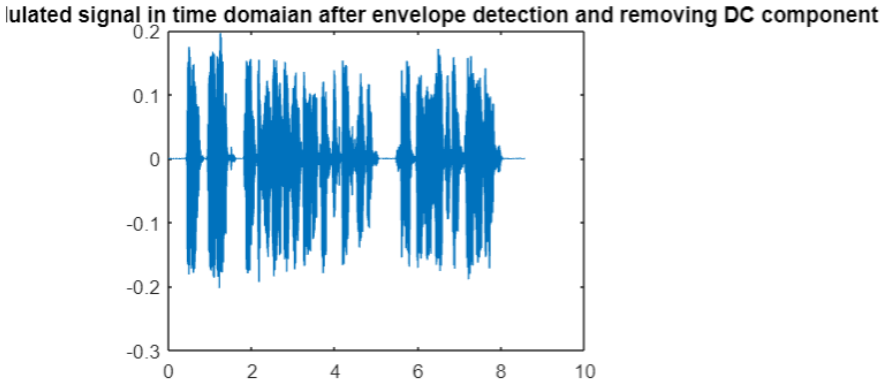

OG Signal in time domain
OG Signal in frequency domain
Filtered signal in time Domain
Filtered signal in Frequency Domain
Modulated signal in time domain (USB)
Modulated signal in frequency domain (USB)
LSB DEMOD in Frequency Domain
LSB DEMOD in Time Domain
LSB DEMOD in Frequency Domain
RECEIVED MESSAGE AFTER DEMOD
Modulated Butter worth filter Signal
Received Butter worth filterd Signal
SNR=0 Recived freq domain
SNR=10 Recived freq domain
SNR=30 Recived freq domain

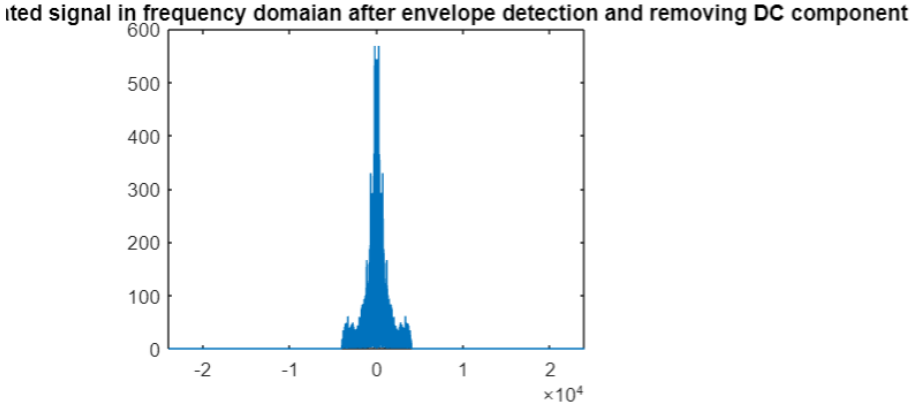
# (Point : 9)

- ## The Code :

```
%--------------------------------------------------------------------------
%-------------------------9-SSBTC------------------------------------------
%--------------------------------------------------------------------------
A = 2*max(RSMBLE_DSBSC);
SSBTC_T = A* cos(2*pi*100*1000*t_resample) + transpose(RSMBLE_DSBSC).*cos(2*pi*100*1000*t_resample);
SSBTC_F = fftshift(fft(SSBTC_T));
%DEMODULATION
ED_SSBTC = abs(hilbert(SSBTC_T ));
ED_SSBTC_WITHOUT_DC = ED_SSBTC - A;
restored_SSBTC_T = resample(ED_SSBTC_WITHOUT_DC,12,125);
restored_SSBTC_F = fftshift(fft(restored_SSBTC_T));
t2 = linspace(0,length(restored_SSBTC_T)/fs,length(restored_SSBTC_T));
F_3 = linspace(-fs/2,fs/2,length(restored_SSBTC_F));
figure;subplot(2,2,1)
plot(t2,restored_SSBTC_T); title('SSB-TC modulated signal in time domaian after envelope detection and removing DC component');
subplot(2,2,3)
plot(F_3,abs(restored_SSBTC_F)); title('SSB-TC modulated signal in frequency domaian after envelope detection and removing DC component');
audiowrite('ED_SSBTC.wav',restored_SSBTC_T,fs);
%----------------------END---------------------------------------------------
```

- ## The plots :



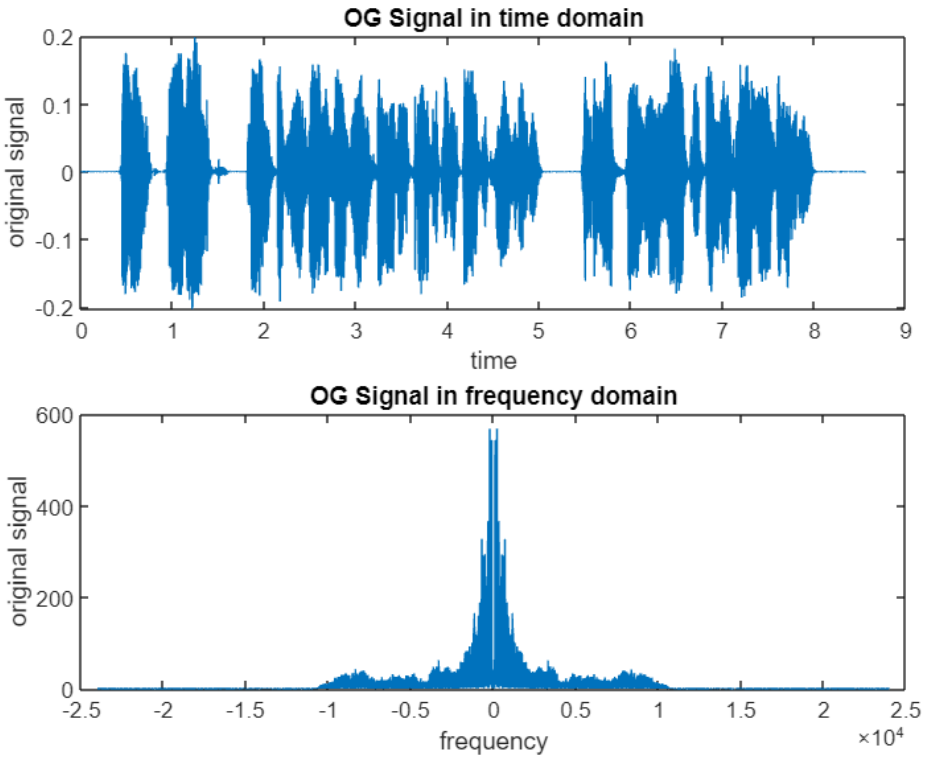
# EXPERIMENT THREE: FREQUENCY MODULATION

## ⟶ (Point :1)

- **The code :**

```
%-------------------------------------------------------------------------
%-------------------------1-READ AUDIO-----------------------------------
[y , fs] = audioread('eric.wav'); %read data(audio) from the file and returns sampled data(y) and sample rate fs
Y = fftshift(fft(y)); % getting fourir of the signal using fast fourier transform
t = linspace(0,length(y)/fs,length(y));
figure (1)
subplot(2,1,1) %PLOTTING OG AUDIO SIGNAL IN TIME DOMAIN
plot(t, y)
xlabel('time') , ylabel('original signal')
title('OG Signal in time domain')
f = linspace(-fs/2,fs/2,length(Y));
subplot(2,1,2) %PLOTTING OG AUDIO SIGNAL IN FREQ DOMAIN
plot(f,abs(Y))
xlabel('frequency') , ylabel('original signal')
title('OG Signal in frequency domain');
```

- **The Plots:**

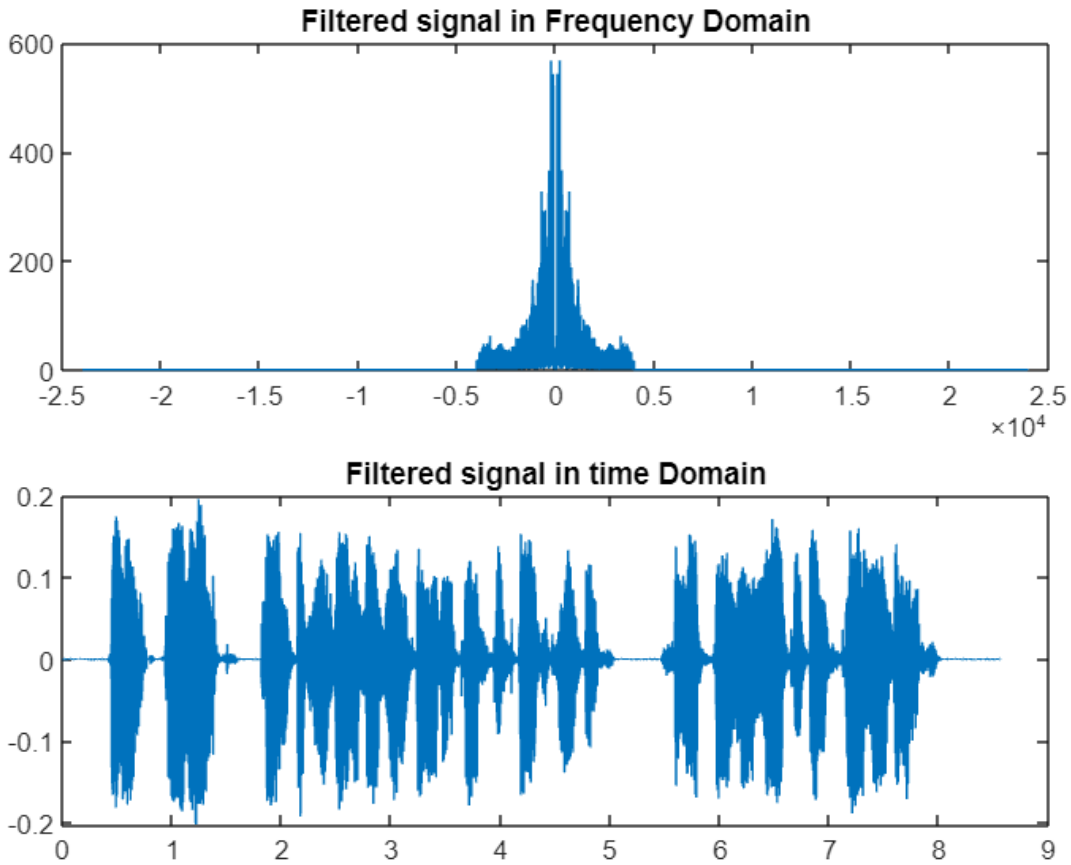


# (Point : 2,3,4)

- ## The Code :

```
%------------2,3,4-IDEAL FILTER TO REMOVE FREQUENCES ABOVE 4 KHZ-----------
%--------------------------------------------------------------------------
Filter = rectpuls(f, 8000); %the ideal filter using rectpuls which returns a rectangular of width 8KHZ
Filtered_Y = Y.* transpose(Filter); %transpose to be the same size then
% the filter passes signal in the range of 4KHZ
t=linspace(0,length(y)/fs,length(y));
figure (2)
subplot(2,1,1)  %plotting spectrum of the filtered signal
plot(f,abs(Filtered_Y))
title('Filtered signal in Frequency Domain')
Filtered_y = real(ifft(ifftshift(Filtered_Y)));%getting the filtered signal in time
% domain using inverse fourier transform
subplot(2,1,2) %plotting the filtered signal in time domain
plot(t,Filtered_y); title('Filtered signal in time Domain');
%player = audioplayer(Filtered_y,fs);
%play(player)
```

- ## The Plots :



## (Point : 5,6)

what is the condition we needed to achieve NBFM?

**Ans:** **B.W** ↓↓ **Kf** ↓↓

- **The Code :**

```
%-------------------------------------------------------------------------
%---------4-demodulation-----------------------------------------------
%envelope detector
B=A*msg_int; %from the modulated signal equation envelope = sqr(A.^2 + A.^2 integrated msg.^2)
ED=sqrt(A.^2 + B.^2);
%extracting message from the envelope by using differentiator
% Differentiaing the signal decreases the length by 1 so we add zero at the
Received_msg = zeros(length(msg),1);
Received_msg(2:end) =(diff(ED));
Received_msg_f = fftshift(fft(Received_msg));
figure(4)
subplot(2,1,1) %plotting recived signal in time domain
plot(t,Received_msg)
title('Message in time Domain')
subplot(2,1,2) %plotting recived signal in frequency domain
plot(f,abs(Received_msg_f));
title('Message in Frequency Domain')
```

- **The Plots :**