DSB and FM Modulation

Matlab Project

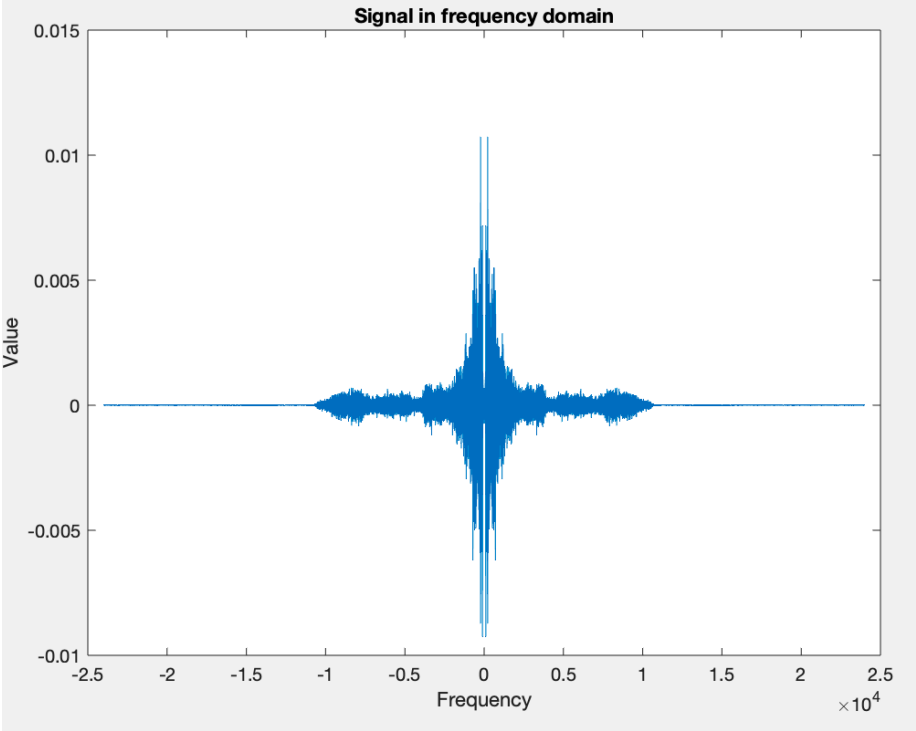
Tarek Mohamed Hany	5344
Youssef Hussein	5659
Omar Radwan	6013
Omar Walid Kamal	6022
Verginia Ehab	5320

Some defined functions used in our code:

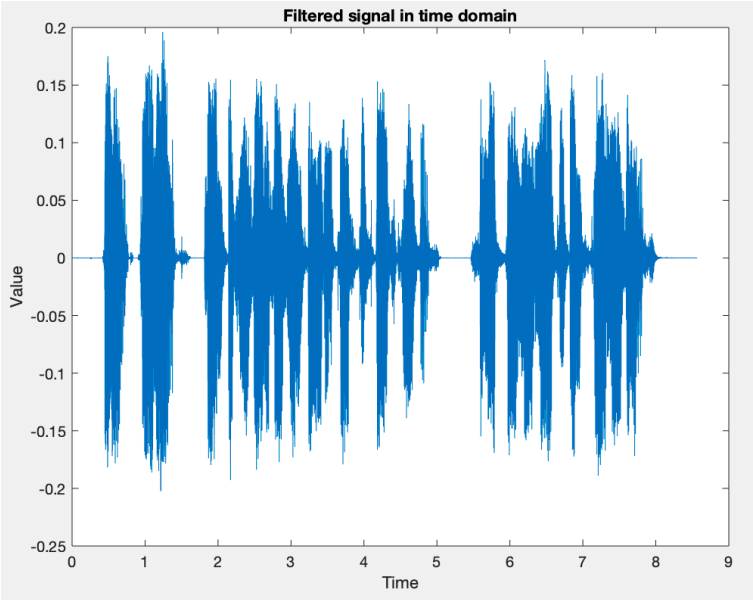
```
function  plot_in_frequency(yf,fs,title_label)
    f = linspace(-fs/2,fs/2,length(yf));
    figure();
    %plot(f,yf);
    plot(f,yf./fs);
    title(title_label);
    xlabel('Frequency');
    ylabel('Value');
end

function  plot_in_time(yt,fs,title_label)
    t = linspace(0, length(yt)/fs, length(yt));
    figure();
    plot(t,yt);
    title(title_label);
    xlabel('Time');
    ylabel('Value');
end

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

function [yt_demod,yf_demod] = env_demod(st,fs_cur,fs_res,is_snr,snr)

    if is_snr == 1
        st = awgn(st, snr);
    end

    yt_demod = resample(abs(hilbert(st)),fs_res,fs_cur); %envelope detector and resample
    yf_demod = fftshift(fft(yt_demod));
end

function [yt_demod, yf_demod] = coh_demod(st,fs_bef,fs_aft,is_snr,snr,fc,phase,f_filter)
if is_snr == 1
    st = awgn(st, snr);
end

t = linspace(0,length(st)/fs_bef, length(st)); %(x2-x1)/(n-1) = 1/5*fc, linspace(x1,x2,n)
carrier_t = cos(2*pi*fc*t+phase).';
tmp = st.*carrier_t; %m(t)*c(t)
filter = generate_filter(length(tmp),fs_bef,f_filter); %filter in frequency domain

tmp = fftshift(fft(tmp)).*filter;
tmp = ifft(ifftshift(tmp));
yt_demod = resample(tmp,fs_aft,fs_bef); %mutiply by carrier,filter and resample
yf_demod = fftshift(fft(yt_demod));
end
```

1- DSB Modulation:

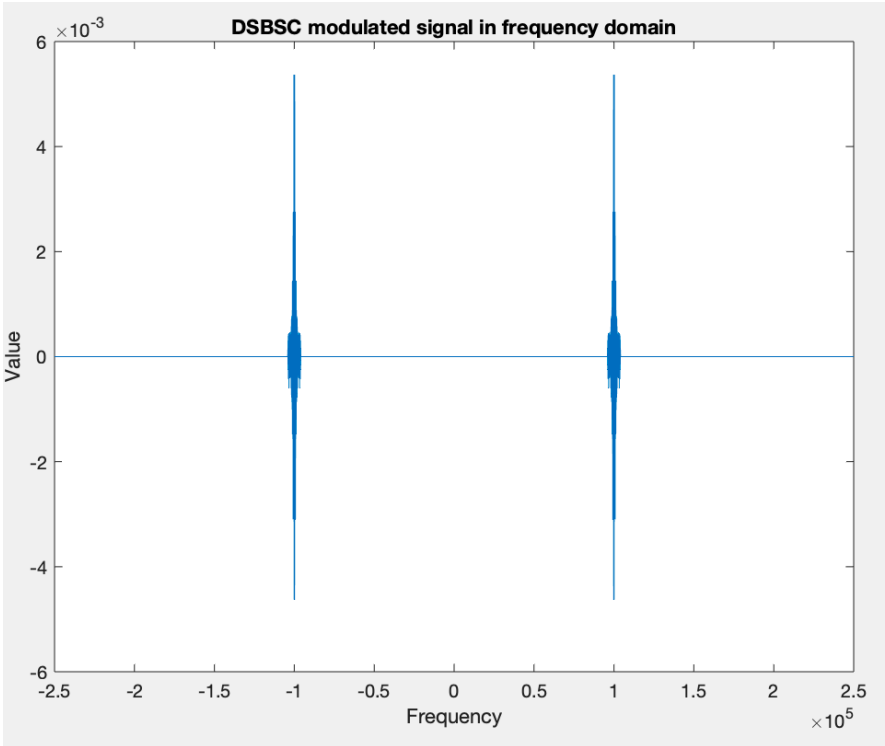
```
%-----------------------------------------------read file-----------------------------------------------------------
FILE = 'eric.wav';
fc = 100000;
fs_res = 5*fc;
modulation_index = 0.5;
[yt, fs]= audioread(FILE);
max_amp = max(abs(yt));
f_filter = 4000;
%-----------------------------------------------plotting-----------------------------------------------------------
%plot_in_time(yt,fs,'Signal in time domain');
%-----------------------------------------------message in frequency domain----------------------------------------
yf = fftshift(fft(yt));
%-----------------------------------------------plotting-----------------------------------------------------------
plot_in_frequency(real(yf),fs,'Signal in frequency domain');
```



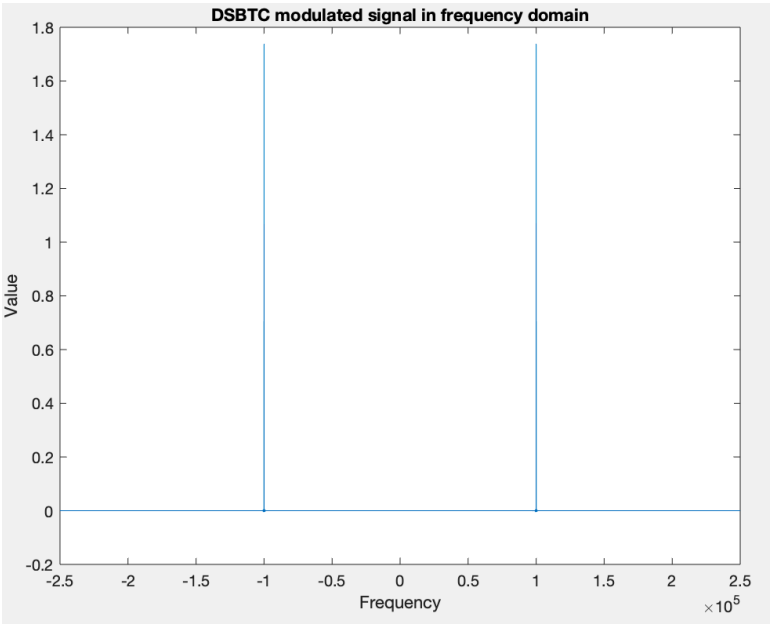

```
%-----------------------------------------------filter-------------------------------------------------------------
filter = generate_filter(length(yf),fs,f_filter);
yf_filtered = filter.*yf;
%-----------------------------------------------plotting-----------------------------------------------------------
%plot_in_frequency(real(filter),fs,'Filter');
%plot_in_frequency(real(yf_filtered),fs,'Filtered message in frequency domain');
%-----------------------------------------------back to time domain------------------------------------------------
yt_filtered= real(ifft(ifftshift(yf_filtered)));
%-----------------------------------------------sound--------------------------------------------------------------
%sound(yt_filtered,fs);
%-----------------------------------------------plotting-----------------------------------------------------------
t = linspace(0,length(yt_filtered)/fs, length(yt_filtered));
plot_in_time(yt_filtered,fs, 'Filtered signal in time domain');
```



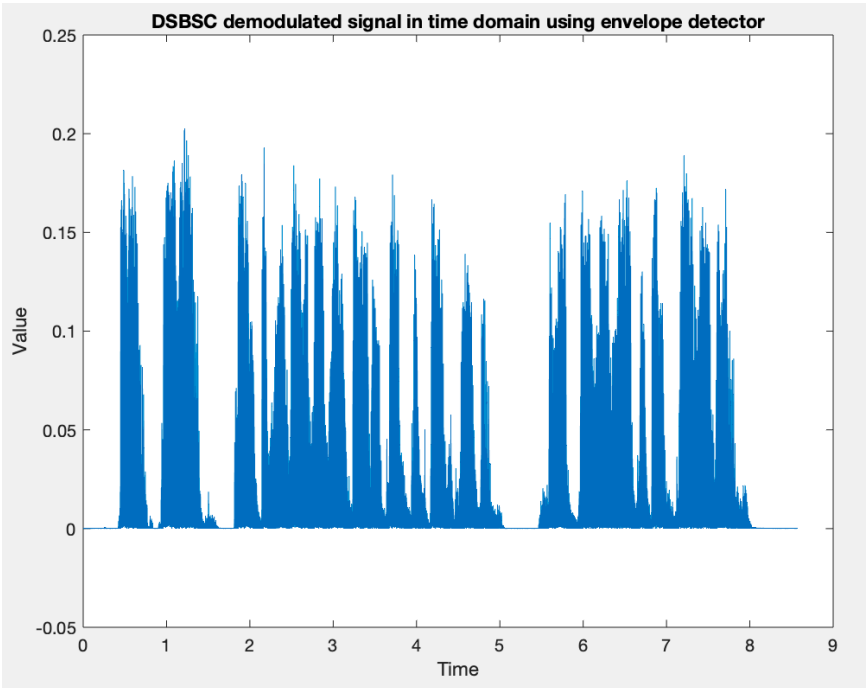

```
%-------------------------------------------------resample-------------------------------------------------------------------
yt_resampled = resample(yt_filtered,fs_res,fs);
%-------------------------------------------------carrier signal-------------------------------------------------------------------
t_carr = linspace(0,length(yt_resampled)/fs_res, length(yt_resampled)); %(x2-x1)/(n-1) = 1/5*fc, linspace(x1,x2,n)
carrier_t = cos(2*pi*fc*t_carr).';
carrier_f = fftshift(fft(carrier_t));

%-------------------------------------------------DSBSC modulation-------------------------------------------------------------
yt_sc = carrier_t.*yt_resampled;
yf_sc = fftshift(fft(yt_sc));
%sound(yt_sc, fs);


%-------------------------------------------------plotting-------------------------------------------------------------------
%plot_in_time(yt_sc,fs_res,'DSBSC modulated signal in time domain');
plot_in_frequency(real(yf_sc),fs_res,'DSBSC modulated signal in frequency domain');
```



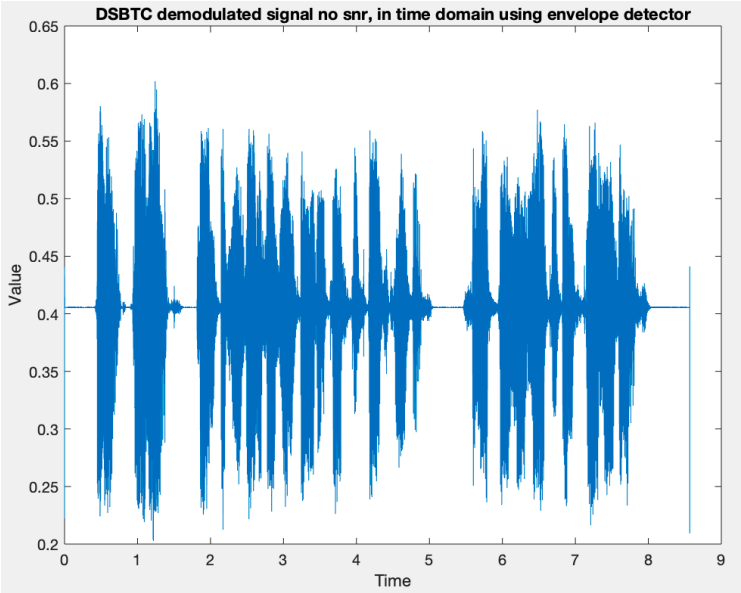
```
%-------------------------------------------------DSBTC modulation-----------------------------------------------------
max_message = max(abs(yt_resampled));
A = 2*max_message;
yt_tc = (A+yt_resampled).*carrier_t;
yf_tc = fftshift(fft(yt_tc));
%-------------------------------------------------plotting-------------------------------------------------------------
%plot_in_time(yt_tc,fs_res,'DSBTC modulated signal in time domain');
plot_in_frequency(real(yf_tc),fs_res,'DSBTC modulated signal in frequency domain');
%sound(yt_tc,fs);
```



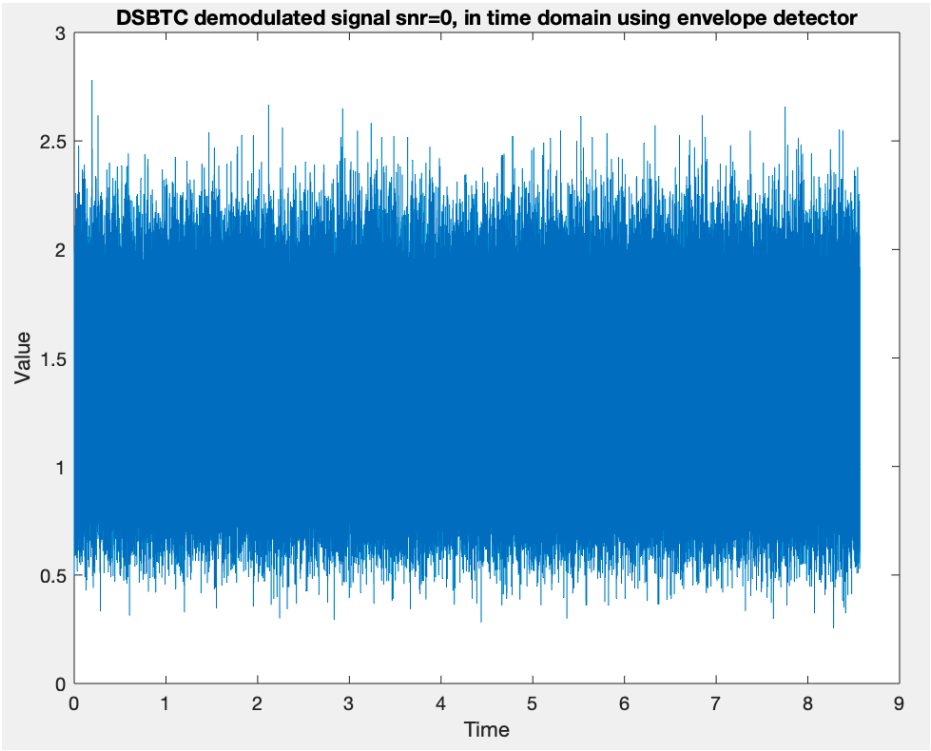

```
%-------------------------------------------------DSBSC envelope detector demodulation---------------------------------
[yt_sc_env, yf_sc_env] = env_demod(yt_sc,fs_res,fs,0,0);
%-------------------------------------------------plotting-------------------------------------------------------------
plot_in_time(yt_sc_env, fs, 'DSBSC demodulated signal in time domain using envelope detector');
%plot_in_frequency(real(yf_sc_env), fs, 'DSBSC demodulated signal in frequency domain using envelope detector');
%-------------------------------------------------sound----------------------------------------------------------------
%sound(yt_sc_env,fs); %should not be good
```



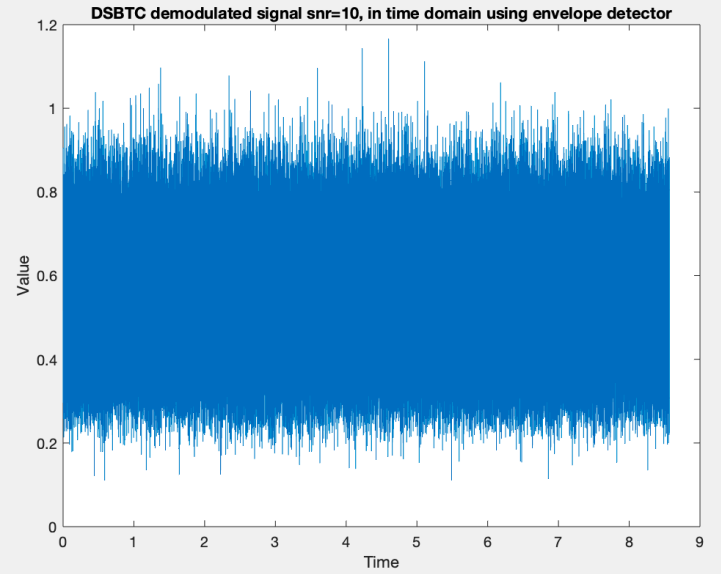
```
%----------------------------------------DSBTC no snr envelope detector demodulation----------------------------------------
[yt_tc_env, yf_tc_env] = env_demod(yt_tc,fs_res,fs,0,0);
%-----------------------------------------------plotting-----------------------------------------------------------------
plot_in_time(yt_tc_env, fs, 'DSBTC demodulated signal no snr, in time domain using envelope detector');
%plot_in_frequency(real(yf_tc_env), fs, 'DSBTC demodulated signal no snr in frequency domain using envelope detector');
%-----------------------------------------------sound--------------------------------------------------------------------
%sound(yt_tc_env,fs); %should be good
```



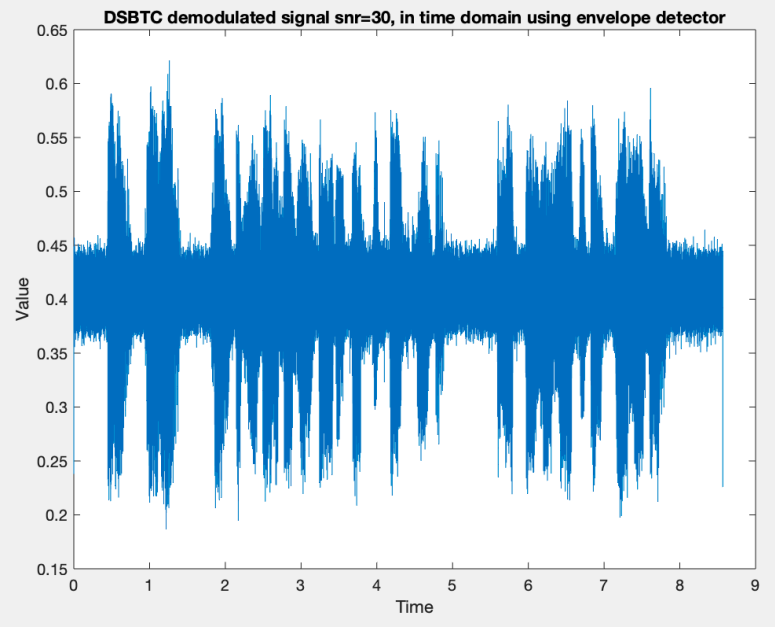

```
%-----------------------------------------------DSBTC 0 snr envelope detector demodulation----------------------------------------
[yt_tc_env, yf_tc_env] = env_demod(yt_tc,fs_res,fs,1,0);
%-----------------------------------------------plotting-----------------------------------------------------------------
plot_in_time(yt_tc_env, fs, 'DSBTC demodulated signal snr=0, in time domain using envelope detector');
%plot_in_frequency(real(yf_tc_env), fs, 'DSBTC demodulated signal snr=0, in frequency domain using envelope detector');
%-----------------------------------------------sound--------------------------------------------------------------------
%sound(yt_tc_env,fs);
```



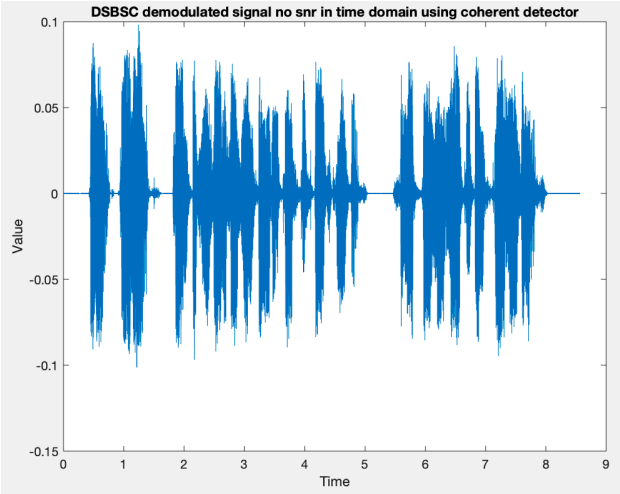
```
%-----------------------------------------------DSBTC 10 snr envelope detector demodulation-----------------------------------------
[yt_tc_env, yf_tc_env] = env_demod(yt_tc,fs_res,fs,1,10);
%-----------------------------------------------plotting----------------------------------------------------------------------------
plot_in_time(yt_tc_env, fs, 'DSBTC demodulated signal snr=10, in time domain using envelope detector');
%plot_in_frequency(real(yf_tc_env), fs, 'DSBTC demodulated signal snr=10, in frequency domain using envelope detector');
%-----------------------------------------------sound-------------------------------------------------------------------------------
%sound(yt_tc_env,fs);
```

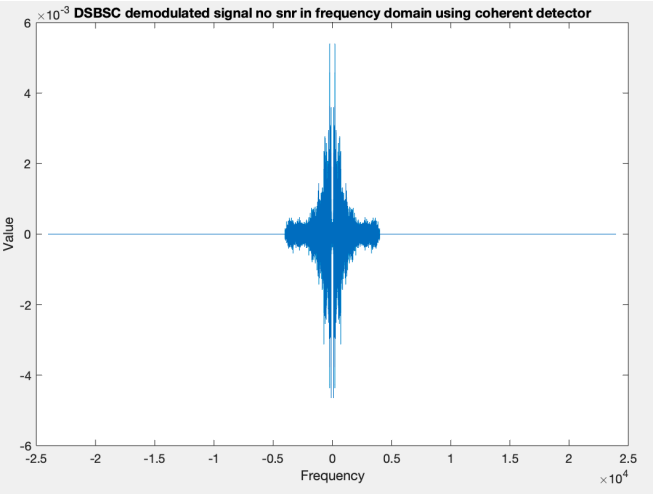



```
%-----------------------------------------------DSBTC 30 snr envelope detector demodulation-----------------------------------------
[yt_tc_env, yf_tc_env] = env_demod(yt_tc,fs_res,fs,1,30);
%-----------------------------------------------plotting----------------------------------------------------------------------------
plot_in_time(yt_tc_env, fs, 'DSBTC demodulated signal snr=30, in time domain using envelope detector');
%plot_in_frequency(real(yf_tc_env), fs, 'DSBTC demodulated signal snr=30, in frequency domain using envelope detector');
%-----------------------------------------------sound-------------------------------------------------------------------------------
%sound(yt_tc_env,fs);
```

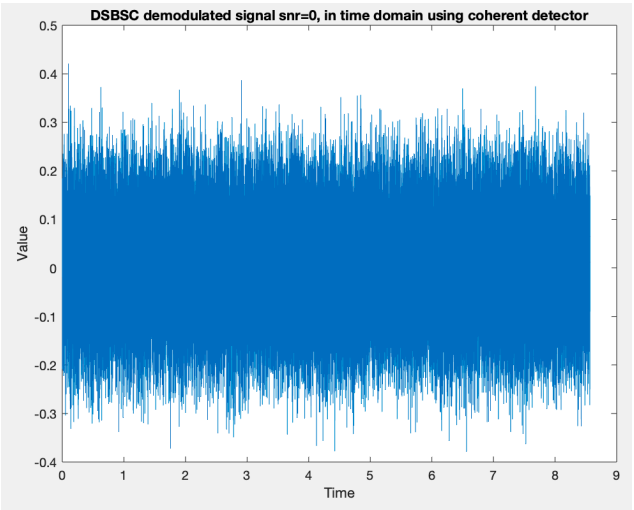


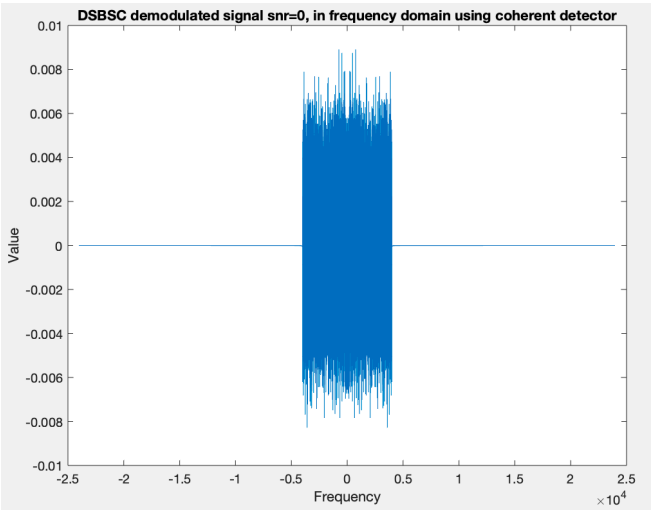
```
%----------------------------------------------DSBSC no snr coherent demodulation-----------------------------------------
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,0,0,fc,0,f_filter);
%----------------------------------------------plotting-------------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal no snr in time domain using coherent detector');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal no snr in frequency domain using coherent detector');
%----------------------------------------------sound----------------------------------------------------------------------
% sound(real(yt_sc_coh),fs); %should be good
```



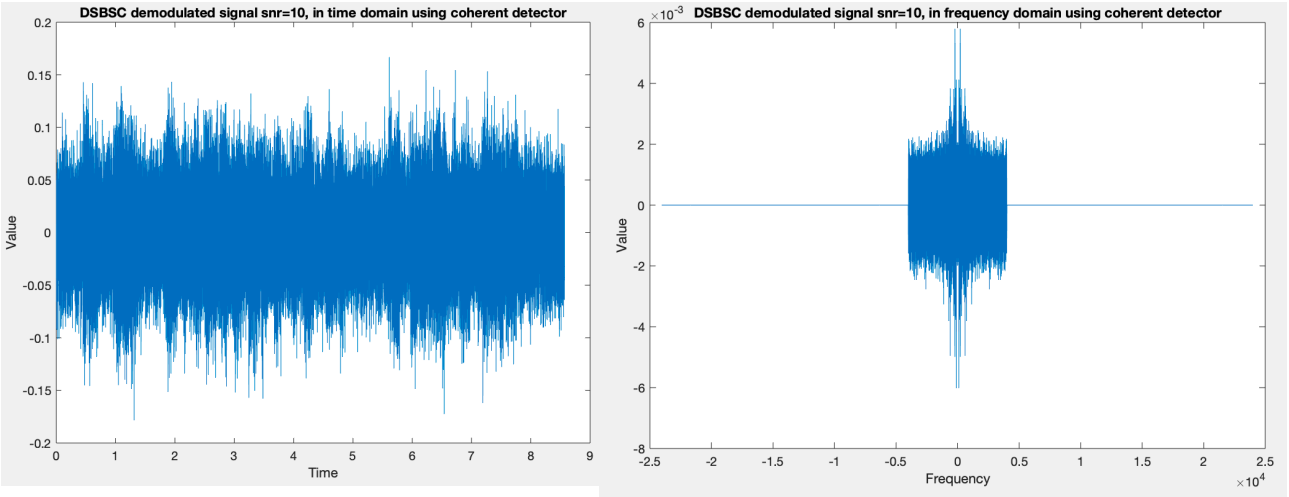



```
%----------------------------------------------DSBSC 0 snr coherent demodulation-------------------------------------
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,1,0,fc,0,f_filter);
%----------------------------------------------plotting-------------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal snr=0, in time domain using coherent detector');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal snr=0, in frequency domain using coherent detector');
%----------------------------------------------sound----------------------------------------------------------------------
%sound(real(yt_sc_coh),fs); %should be good
```

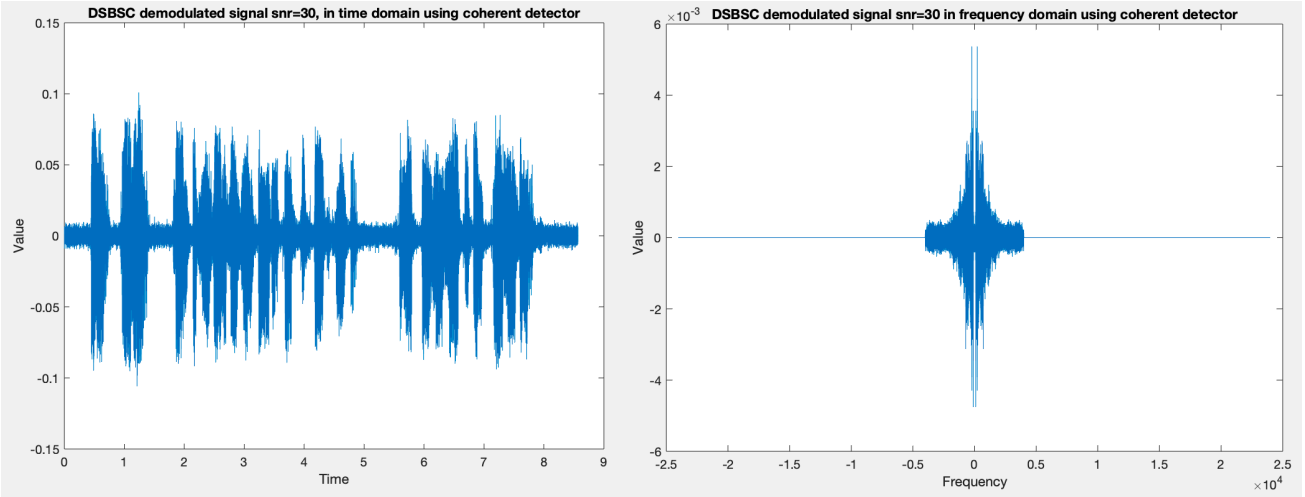




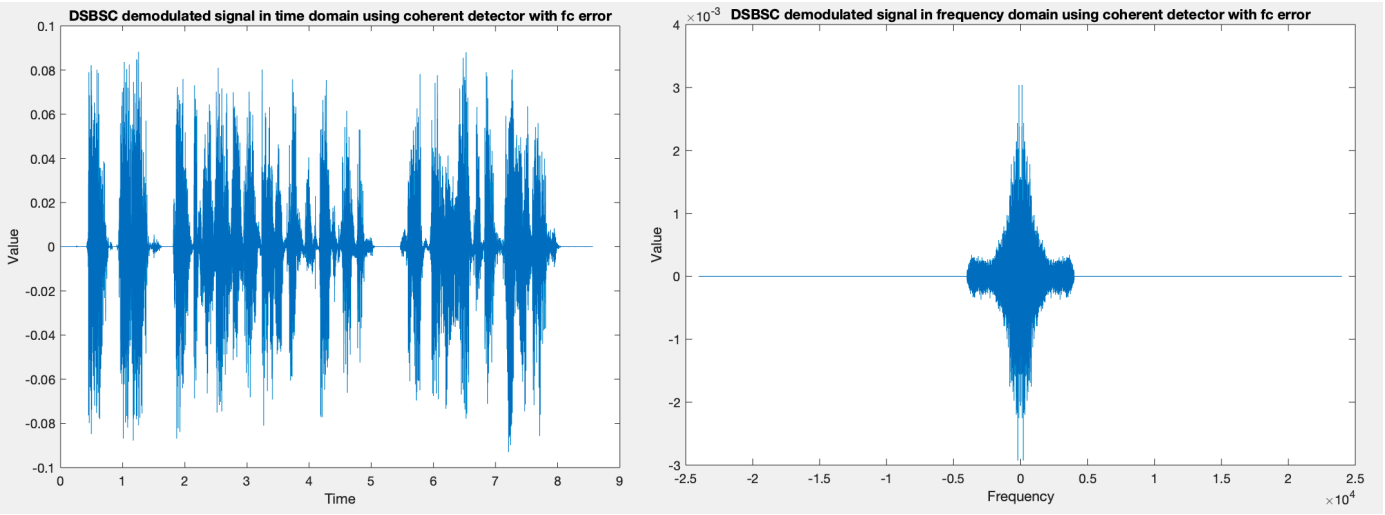
```
%-----------------------------------------------DSBSC 10 snr coherent demodulation-----------------------------------------------
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,1,10,fc,0,f_filter);
%-----------------------------------------------plotting-----------------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal snr=10, in time domain using coherent detector');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal snr=10, in frequency domain using coherent detector');
%-----------------------------------------------sound--------------------------------------------------------------------------
%sound(real(yt_sc_coh),fs);
```



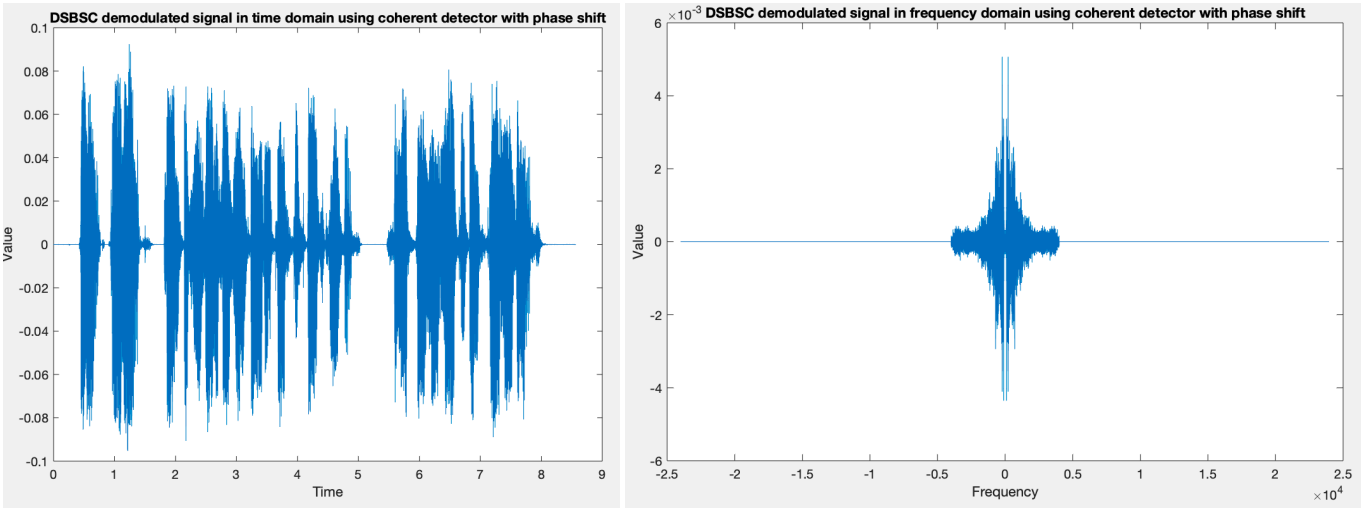

```
% ---------------------------------- DSBSC 30 snr coherent demodulation-----------------------------------------------
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,1,30,fc,0,f_filter);
%-----------------------------------------------plotting-----------------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal snr=30, in time domain using coherent detector');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal snr=30 in frequency domain using coherent detector');
%-----------------------------------------------sound--------------------------------------------------------------------------
%sound(real(yt_sc_coh),fs);
```



```
%----------------------------------------DSBSC no snr coherent demodulation with fc error----------------------------------------
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,0,0,fc+100,0,f_filter);
%-----------------------------------------------plotting-----------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal in time domain using coherent detector with fc error');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal in frequency domain using coherent detector with fc error');
%-----------------------------------------------sound--------------------------------------------------------------------
% sound(real(yt_sc_coh),fs);
```

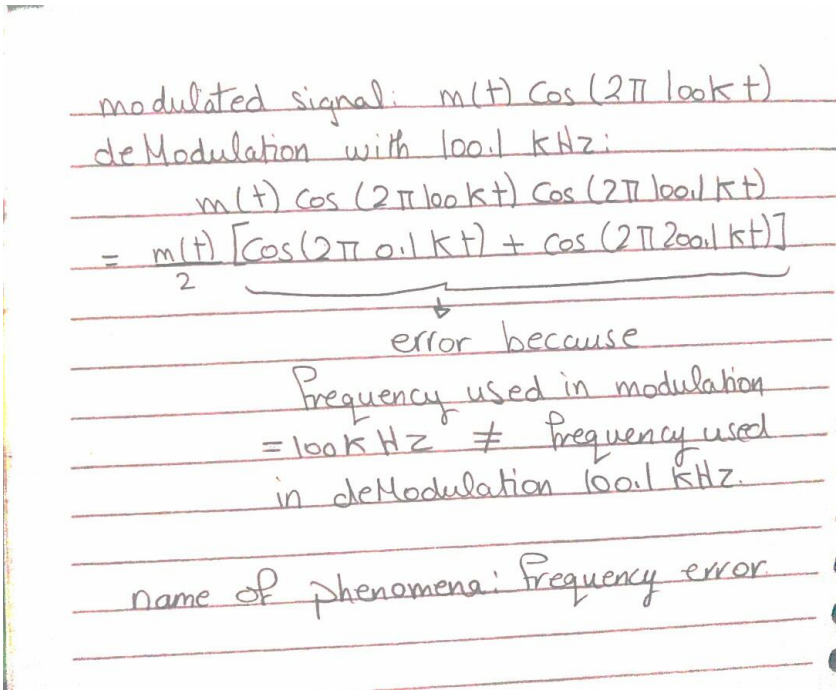



```
%---------------------------------------- DSBSC no snr coherent demodulation with phase shift ----------------------------------
phaseshift = degtorad(20);
[yt_sc_coh, yf_sc_coh] = coh_demod(yt_sc,fs_res,fs,0,0,fc,phaseshift,f_filter);
%-----------------------------------------------plotting-----------------------------------------------------------------
plot_in_time(real(yt_sc_coh), fs, 'DSBSC demodulated signal in time domain using coherent detector with phase shift');
plot_in_frequency(real(yf_sc_coh), fs, 'DSBSC demodulated signal in frequency domain using coherent detector with phase shift');
%-----------------------------------------------sound--------------------------------------------------------------------
%sound(real(yt_sc_coh),fs);
```



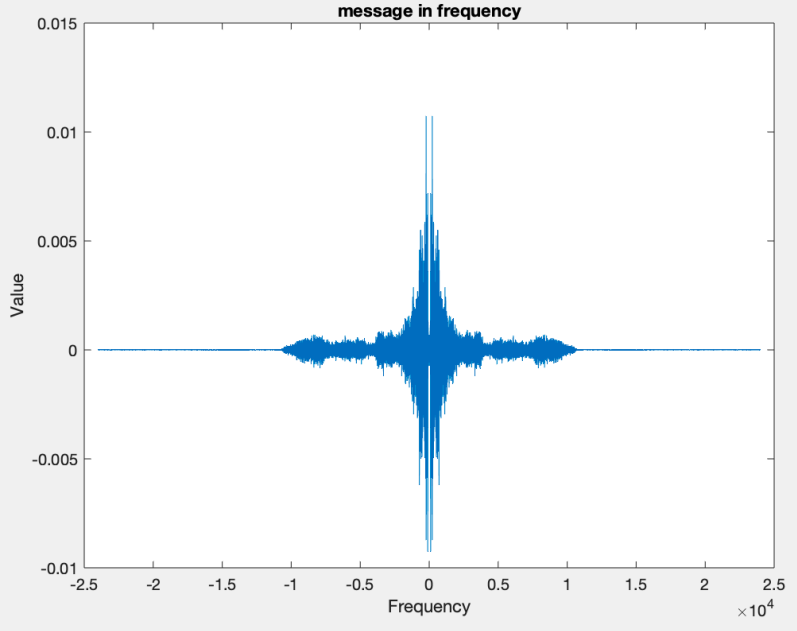
Conclusions on DSB:

- Envelope Detector can be used to detect DSBTC but not good with DSBSC.
- As the snr increases, the signal becomes clearer

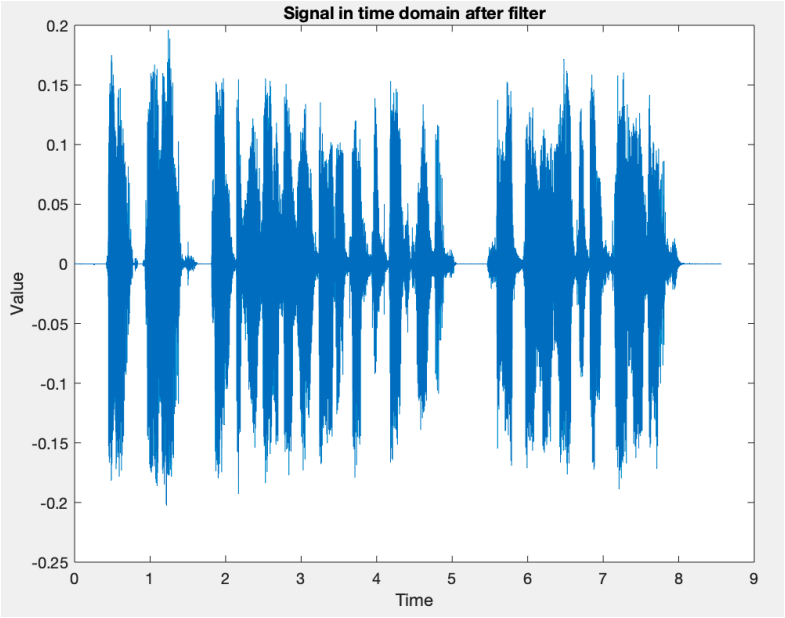


2- FM Modulation:

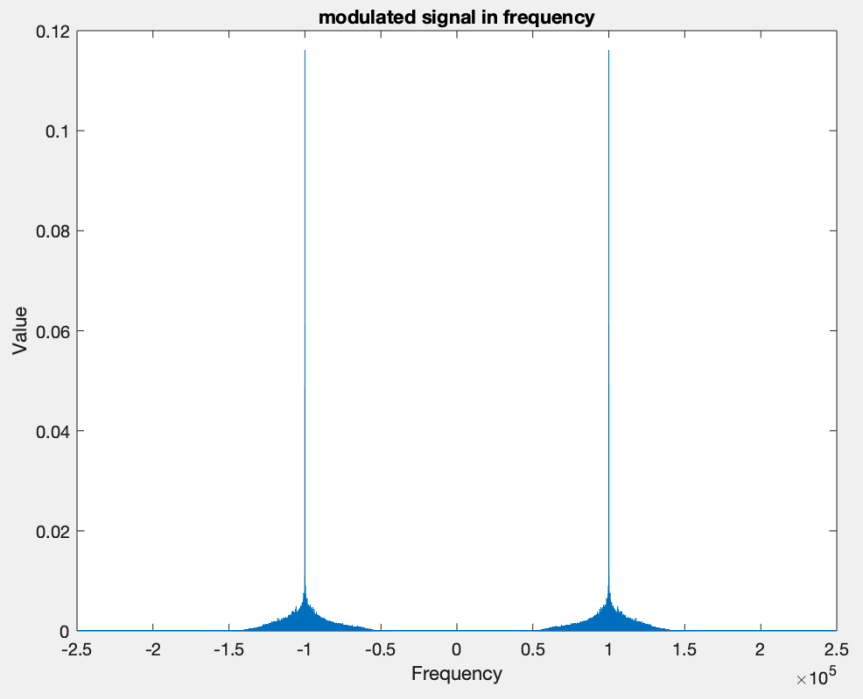
```
%-----------------------------------------------read file-----------------------------------------------------------
FILE = 'eric.wav';
fc = 100000;
[yt, fs]= audioread(FILE);
fs_res = 5*fc;
f_filter = 4000;
%-----------------------------------------------plotting-----------------------------------------------------------
xt = linspace(0,length(yt)/fs,length(yt));
%plot_in_time(yt,fs,'Message in time domain');
plot_in_frequency(real(fftshift(fft(yt))),fs, 'message in frequency');
```




```
%-----------------------------------------------message in frequency domain-----------------------------------------
yf = fftshift(fft(yt));
xf = linspace(-fs/2,fs/2,length(yf));
%-----------------------------------------------filter-------------------------------------------------------------
filter = generate_filter(length(yf),fs,f_filter);
yf_filtered = filter.*yf;
%-----------------------------------------------back to time domain------------------------------------------------
yt_filtered= real(ifft(ifftshift(yf_filtered)));
plot_in_time(yt_filtered, fs, 'Signal in time domain after filter' )
```



```
%-------------------------------------------------sound-------------------------------------------------------------
% sound(yt_filtered,fs);
%-------------------------------------------------plotting----------------------------------------------------------
xt = linspace(0,length(yt_filtered)/fs, length(yt_filtered));
%-------------------------------------------------resmple-----------------------------------------------------------
yt_resampled = resample(yt_filtered,fs_res,fs);
%-------------------------------------------------carrier signal----------------------------------------------------
t = linspace(0,length(yt_resampled)/fs_res, length(yt_resampled)); %(x2-x1)/(n-1) = 1/5*fc, linspace(x1,x2,n) lama radawan yeegy neb2a
nshofha
carrier = cos(2*pi*fc*t).';
carriersin = sin(2*pi*fc*t).';
yt_carrier=carrier;
yf_carrier=fftshift(fft(yt_carrier));
xt_carrier=t;
xf_carrier=linspace(-fc/2,fc/2,length(yt_carrier));
%-------------------------------------------------FM Modulation-----------------------------------------------------
%-------------------------------------------------NBFM--------------------------------------------------------------
A = max(abs(yt));
kf = pi; %de ely btefre2 fel amplitude bta3 elcarrier
beta = (kf*A)/(2*pi*fs_res);
m_int = kf.*cumsum(yt_resampled).'; % Integrating Msg
St = A.*cos(2*pi*fc*t + m_int);
%plot_in_time(St, fs_res, 'modulated in time');
plot_in_frequency(abs(fftshift(fft(St))),fs_res,'modulated signal in frequency');
%St = A.*cos(2*pi*fc*t)-m_int;%.*sin(2*pi*fc*t);
fourier = fftshift(fft(St));
```




```
%-------------------------------------------------Demodulation------------------------------------------------------
%------------------------------convert FM to AM-----------------------------
%AM = A.*(2*pi*fc+kf*yt_resampled).*sin(2*pi*fc*t+m_int);
AM = diff(St);
AM = [0 AM];
%plot_in_frequency(abs(fftshift(fft(AM))), fs, 'lel');
[yt_demod, yf_demod] = env_demod(AM,fs_res,fs,0,0);

%plot_in_time(yt_demod,fs,'Envelope Detector');
%plot_in_frequency(yf_demod,fs,'demodulated signal in frequency');

sound(yt_demod,fs);
```

Notes on FM Modulation:

- To be narrow-band FM, β should be small, BW=2fm(1+ β) =2fm where β <<< 1

- After plotting the spectrum of the NBFM signal, we can observe that the spectrum is similar to that of DSBTC, so we can use Envelope Detector for demodulation.