Mariam Mohamed Bahaa 6401

Yara Tarek Fathy 6445

Nour Mohamed Gaber 6392

Nada Ossama Soffar 6503

Sara Tarek Awny 6475

Hania Mohamed Abo Elamayem 6257

Final Project Report

**EXPERIMENT ONE: DOUBLE SIDEBAND MODULATION**

**Signal in time and frequency domains:**

%Plotting of original signal in TIME domain

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

figure();

subplot(2,1,1);

plot(t,y);

title('Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

%Plotting of original signal in FREQUENCY domain

yf = fftshift(fft(y));

f = linspace(-fs/2, fs/2, length(yf));

subplot(2,1,2);

plot(f, abs(yf)./fs);

title('Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

A screenshot of a computer

Description automatically generated with medium confidence

Filter in frequency domain:

%filter

filter\_freq = 4000;

lp = ones(length(yf),1);

lp(abs(f)>filter\_freq) = 0;

yf\_filtered = (lp.\*yf);

%plot the filter

f2 = linspace(-fs/2,fs/2,length(lp));

figure();

plot(f2,abs(lp)./fs);

title('Filter');

xlabel('Frequency');

ylabel('Amplitude');

Chart

Description automatically generated

Signal after filter:

%plot the filtered signal

f2 = linspace(-fs/2,fs/2,length(yf\_filtered));

figure();

subplot(2,1,2);

plot(f2,abs(yf\_filtered)./fs);

title('Filtered Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

yt\_filtered = ifft(ifftshift(yf\_filtered));

subplot(2,1,1);

plot(t,real(yt\_filtered));

title('Filtered Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

%sound(real(yt\_filtered), fs);

A screenshot of a computer

Description automatically generated with medium confidence

Signal after DSBSC modulation:

%DSBSC mod

fc=100000;

fs\_new=5\*fc;

yt\_new = resample(yt\_filtered,fs\_new,fs);

t\_new=linspace(0, length(yt\_new)/fs\_new, length(yt\_new));

carrier=transpose(cos(2\*pi\*fc\*t\_new));

yt\_dsbsc=yt\_new.\*carrier;

yf\_dsbsc=real(fftshift(fft(yt\_dsbsc)));

f\_new=linspace(-fs\_new/2,fs\_new/2,length(yt\_dsbsc));

%plot

figure();

subplot(2,1,1);

plot(t\_new,real(yt\_dsbsc));

title('DSBSC modulated Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

plot(f\_new,abs(yf\_dsbsc)./fs\_new);

title('DSBSC modulated Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

Timeline

Description automatically generated

DSBSC after envelope demodulation:

%DSBSC env demod

yt\_sc\_env = resample(abs(hilbert(real(yt\_dsbsc))),fs,fs\_new);

yf\_sc\_env = real(fftshift(fft(yt\_sc\_env)));

%plot

figure();

subplot(2,1,1);

t1=linspace(0,length(yt\_sc\_env)/fs,length(yt\_sc\_env));

plot(t1,real(yt\_sc\_env));

title('DSBSC envelope demodulated Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f1 = linspace(-fs/2, fs/2, length(yf\_sc\_env));

plot(f1,abs(yf\_sc\_env)./fs);

title('DSBSC envelope demodulated Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_env),fs); %sound of DSBSC env demod

Timeline

Description automatically generated

DSBSC after coherent detection with no SNR:

%DSBSC coherent demodulation and SNR

t\_coh=linspace(0,length(yt\_dsbsc)/fs\_new,length(yt\_dsbsc));

carrier\_coh\_no\_snr=transpose(cos(2\*pi\*fc\*t\_coh));

yt\_sc\_coh=yt\_dsbsc.\*carrier\_coh\_no\_snr;

yf\_sc\_coh=fftshift(fft(yt\_sc\_coh));

lp\_sc\_coh= 2\*ones(length(yt\_dsbsc),1);

lp\_sc\_coh(abs(f)>4000) = 0;

yf\_sc\_no\_snr\_filtered = lp\_sc\_coh.\*yf\_sc\_coh;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_no\_snr\_filtered));

yt\_sc\_coh\_no\_snr=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_no\_snr=real(fftshift(fft(yt\_sc\_coh\_no\_snr)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_no\_snr)/fs,length(yt\_sc\_coh\_no\_snr));

plot(t\_after\_resample,real(yt\_sc\_coh\_no\_snr));

title('DSBSC coherent demodulated Signal no snr in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_no\_snr));

plot(f\_after\_resample,abs(yf\_sc\_coh\_no\_snr)./fs);

title('DSBSC coherent demodulated Signal no snr in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_coh\_no\_snr),fs);

A screenshot of a computer

Description automatically generated with medium confidence

DSBSC after coherent detection with SNR=0:

%SNR=0

yt\_coh\_0\_snr=awgn(yt\_dsbsc,0);

t\_coh=linspace(0,length(yt\_coh\_0\_snr)/fs\_new,length(yt\_coh\_0\_snr));

carrier\_coh\_0\_snr=transpose(cos(2\*pi\*fc\*t\_coh));

temp\_before\_filter=yt\_coh\_0\_snr.\*carrier\_coh\_0\_snr;

yf\_coh\_0\_snr=fftshift(fft(temp\_before\_filter));

f\_SNR0=linspace(-(fs\_new)/2,(fs\_new)/2,length(yf\_coh\_0\_snr));

lp\_sc\_coh= 2\*ones(length(yt\_coh\_0\_snr),1);

lp\_sc\_coh(abs(f\_SNR0)>4000) = 0;

yf\_sc\_0\_snr\_filtered = lp\_sc\_coh.\*yf\_coh\_0\_snr;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_0\_snr\_filtered));

yt\_sc\_coh\_0\_snr=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_0\_snr=real(fftshift(fft(yt\_sc\_coh\_0\_snr)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_0\_snr)/fs,length(yt\_sc\_coh\_0\_snr));

plot(t\_after\_resample,real(yt\_sc\_coh\_0\_snr));

title('DSBSC coherent demodulated Signal SNR=0 in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_0\_snr));

plot(f\_after\_resample,abs(yf\_sc\_coh\_0\_snr)./fs);

title('DSBSC coherent demodulated Signal SNR=0 in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_coh\_0\_snr),fs);

Timeline

Description automatically generated

DSBSC after coherent detection with SNR=10:

%SNR=10

yt\_coh\_10\_snr=awgn(yt\_dsbsc,10);

t\_coh=linspace(0,length(yt\_coh\_10\_snr)/fs\_new,length(yt\_coh\_10\_snr));

carrier\_coh\_10\_snr=transpose(cos(2\*pi\*fc\*t\_coh));

temp\_before\_filter=yt\_coh\_10\_snr.\*carrier\_coh\_10\_snr;

yf\_coh\_10\_snr=fftshift(fft(temp\_before\_filter));

f\_SNR10=linspace(-(fs\_new)/2,(fs\_new)/2,length(yf\_coh\_10\_snr));

lp\_sc\_coh= 2\*ones(length(yt\_coh\_10\_snr),1);

lp\_sc\_coh(abs(f\_SNR10)>4000) = 0;

yf\_sc\_10\_snr\_filtered = lp\_sc\_coh.\*yf\_coh\_10\_snr;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_10\_snr\_filtered));

yt\_sc\_coh\_10\_snr=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_10\_snr=real(fftshift(fft(yt\_sc\_coh\_10\_snr)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_10\_snr)/fs,length(yt\_sc\_coh\_10\_snr));

plot(t\_after\_resample,real(yt\_sc\_coh\_10\_snr));

title('DSBSC coherent demodulated Signal SNR=10 in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_10\_snr));

plot(f\_after\_resample,abs(yf\_sc\_coh\_10\_snr)./fs);

title('DSBSC coherent demodulated Signal SNR=10 in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_coh\_10\_snr),fs);

Timeline

Description automatically generated

DSBSC after coherent detection with SNR=30:

%SNR=30

yt\_coh\_30\_snr=awgn(yt\_dsbsc,30);

t\_coh=linspace(0,length(yt\_coh\_30\_snr)/fs\_new,length(yt\_coh\_30\_snr));

carrier\_coh\_30\_snr=transpose(cos(2\*pi\*fc\*t\_coh));

temp\_before\_filter=yt\_coh\_30\_snr.\*carrier\_coh\_30\_snr;

yf\_coh\_30\_snr=fftshift(fft(temp\_before\_filter));

f\_SNR30=linspace(-(fs\_new)/2,(fs\_new)/2,length(yf\_coh\_30\_snr));

lp\_sc\_coh= 2\*ones(length(yt\_coh\_30\_snr),1);

lp\_sc\_coh(abs(f\_SNR30)>4000) = 0;

yf\_sc\_30\_snr\_filtered = lp\_sc\_coh.\*yf\_coh\_30\_snr;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_30\_snr\_filtered));

yt\_sc\_coh\_30\_snr=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_30\_snr=real(fftshift(fft(yt\_sc\_coh\_30\_snr)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_30\_snr)/fs,length(yt\_sc\_coh\_30\_snr));

plot(t\_after\_resample,real(yt\_sc\_coh\_30\_snr));

title('DSBSC coherent demodulated Signal SNR=30 in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_30\_snr));

plot(f\_after\_resample,abs(yf\_sc\_coh\_30\_snr)./fs);

title('DSBSC coherent demodulated Signal SNR=30 in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_coh\_30\_snr),fs);

A screenshot of a computer

Description automatically generated

DSBSC coherent detection with phase error:

%phase error

t\_coh=linspace(0,length(yt\_dsbsc)/fs\_new,length(yt\_dsbsc));

carrier\_coh\_phase\_error=transpose(cos(2\*pi\*fc\*t\_coh+(20\*(pi/180))));

temp\_before\_filter=yt\_dsbsc.\*carrier\_coh\_phase\_error;

yf\_sc\_coh\_phase\_error=fftshift(fft(temp\_before\_filter));

lp\_sc\_coh= 2\*ones(length(yt\_dsbsc),1);

lp\_sc\_coh(abs(f)>fs\_new) = 0;

yf\_sc\_phase\_error\_filtered = lp\_sc\_coh.\*yf\_sc\_coh\_phase\_error;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_phase\_error\_filtered));

yt\_sc\_coh\_phase\_error=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_phase\_error=real(fftshift(fft(yt\_sc\_coh\_phase\_error)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_phase\_error)/fs,length(yt\_sc\_coh\_phase\_error));

plot(t\_after\_resample,real(yt\_sc\_coh\_phase\_error));

title('DSBSC coherent demodulated Signal phase error in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_phase\_error));

plot(f\_after\_resample,abs(yf\_sc\_coh\_phase\_error)./fs);

title('DSBSC coherent demodulated Signal phase error in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

% sound(real(yt\_sc\_coh\_phase\_error),fs);

A screenshot of a computer

Description automatically generated with medium confidence

DSBSC coherent detection with frequency error:

%frequency error

t\_coh=linspace(0,length(yt\_dsbsc)/fs\_new,length(yt\_dsbsc));

carrier\_coh\_freq\_error=transpose(cos(2\*pi\*100100\*t\_coh));

temp\_before\_filter=yt\_dsbsc.\*carrier\_coh\_freq\_error;

yf\_sc\_coh\_freq\_error=fftshift(fft(temp\_before\_filter));

lp\_sc\_coh= 2\*ones(length(yt\_dsbsc),1);

lp\_sc\_coh(abs(f)>fs\_new) = 0;

yf\_sc\_freq\_error\_filtered = lp\_sc\_coh.\*yf\_sc\_coh\_freq\_error;

temp\_after\_filter=ifft(ifftshift(yf\_sc\_freq\_error\_filtered));

yt\_sc\_coh\_freq\_error=resample(temp\_after\_filter,fs,fs\_new);

yf\_sc\_coh\_freq\_error=real(fftshift(fft(yt\_sc\_coh\_freq\_error)));

%plot

figure();

subplot(2,1,1);

t\_after\_resample=linspace(0,length(yt\_sc\_coh\_freq\_error)/fs,length(yt\_sc\_coh\_freq\_error));

plot(t\_after\_resample,real(yt\_sc\_coh\_freq\_error));

title('DSBSC coherent demodulated Signal frequency error in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f\_after\_resample=linspace(-fs/2,fs/2,length(yf\_sc\_coh\_freq\_error));

plot(f\_after\_resample,abs(yf\_sc\_coh\_freq\_error)./fs);

title('DSBSC coherent demodulated Signal frquency error in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_sc\_coh\_freq\_error),fs);

A screenshot of a computer

Description automatically generated with medium confidence

DSBTC modulated signal:

%DSBTC mod

Am=max(abs(real(yt\_new))); %max gain of message

yt\_dsbtc=(2\*Am+real(yt\_new)).\*carrier;

yf\_dsbtc=real(fftshift(fft(yt\_dsbtc)));

%plot

figure();

subplot(2,1,1);

plot(t\_new,real(yt\_dsbtc));

title('DSBTC modulated Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

plot(f\_new,abs(yf\_dsbtc));

xlim([-10500 10500])

title('DSBTC modulated Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

Diagram

Description automatically generated with low confidence

DSBTC envelope demodulated signal:

%DSBTC envelope demodulation

yt\_tc\_env = resample(abs(hilbert((yt\_dsbtc))),fs,fs\_new)-(2\*Am);

yf\_tc\_env = real(fftshift(fft(yt\_tc\_env)));

%plot

figure();

subplot(2,1,1);

t1=linspace(0,length(yt\_tc\_env)/fs,length(yt\_tc\_env));

plot(t1,yt\_tc\_env);

title('DSBTC envelope demod Signal in time domain');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

f1 = linspace(-fs/2, fs/2, length(yf\_tc\_env));

plot(f1,abs(yf\_tc\_env)./fs);

title('DSBTC envelope demod Signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

%sound(real(yt\_tc\_env),fs);

Timeline

Description automatically generated

**Comments**:

DSBTC envelope modulated sound is so much better and clearer than DSBSC envelope demodulated, therefore envelope modulation should be used with DSBTC.

Adding the frequency causes error of 100Hz

Vx(t)= Ac m(t) cos(2fct) cos(2π(fc+f)t)

= (Ac m(t) )/2 [ cos(2π(2fc+f)t) +cos(2πft)] f<<<

= (Ac m(t) )/2 cos(2πft)

f must =0, else it is distorted

**EXPERIMENT TWO: SINGLE SIDEBAND MODULATION**

**%(5)Obtain the SSB by filtering out the USB (we need to get LSB only) of the DSB-SC modulated**

**%signal using an ideal filter then Plot the spectrum again.**

f\_ssb=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_dsbsc));

lpfc = 2.\*ones(length(yf\_dsbsc),1);

lpfc(abs(f\_ssb)>fc) = 0;

yf\_lssb = lpfc.\*yf\_dsbsc;

yt\_ssb = ifft(ifftshift(yf\_lssb));

t\_ssb=linspace(0, length(yt\_ssb)/fs\_new, length(yt\_ssb));

figure();

subplot(2,1,1);

plot(f\_ssb,abs(yf\_lssb)./fs\_new);

title('Single SideBand Modulation (LSB) frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

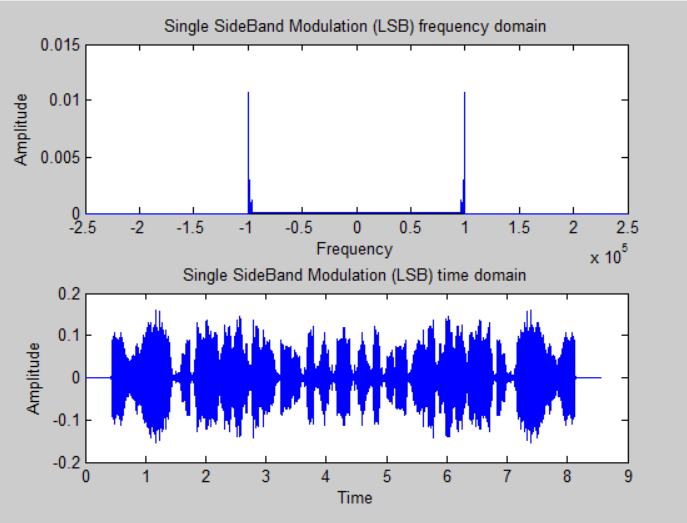
subplot(2,1,2);

plot(t\_ssb,real(yt\_ssb));

title('Single SideBand Modulation (LSB) time domain');

xlabel('Time');

ylabel('Amplitude');



**%(6)-Use coherent detection with no noise interference to get the received signal (to demodulate the**

**%SSB-SC) and play the file back also sketch the received waveform and spectrum**

**%SSB coherent demod time domain**

carrier1=cos(2\*pi\*fc\*t\_ssb).';

yt\_ssbdemod = carrier1.\*yt\_ssb;

**%SSB coherent demod freq domain**

yf\_ssbdemod = fftshift(fft(yt\_ssbdemod));

**%filter ssb**

yf\_ssbdemod(f\_ssb>4000) = 0;

yf\_ssbdemod(f\_ssb<-4000) = 0;

yf\_ssbdemod2 = yf\_ssbdemod.\*2;

**%filtered ssb demodulated**

f\_ssbf=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_ssbdemod2));

yt\_ssbdemodf = ifft(ifftshift(yf\_ssbdemod2));

yt\_ssbdemodf = resample(yt\_ssbdemodf,fs,fs\_new);

t\_ssbf=linspace(0, length(yt\_ssbdemodf)/fs\_new, length(yt\_ssbdemodf));

figure();

subplot(2,1,1);

plot(f\_ssbf,abs(yf\_ssbdemod2)./fs);

title('SSB cohDemodulated (Frequency)');

xlabel('Frequency');

ylabel('Amplitude');

subplot(2,1,2);

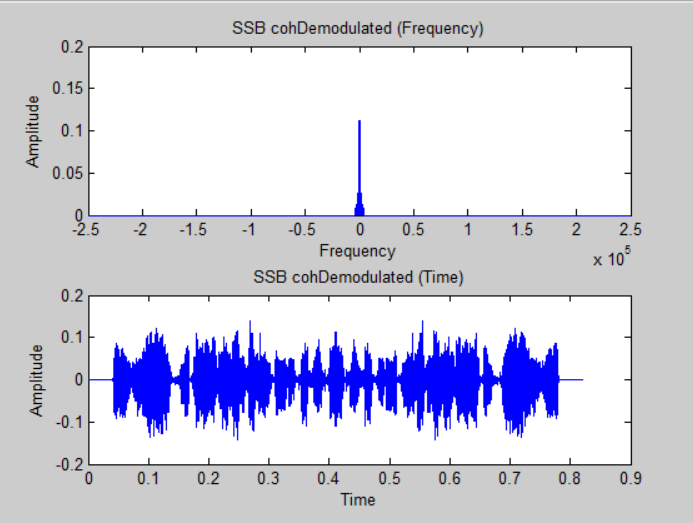
plot(t\_ssbf,real(yt\_ssbdemodf));

title('SSB cohDemodulated (Time)');

xlabel('Time');

ylabel('Amplitude');

%sound(real(yt\_ssbdemodf),fs);



**%(7)Repeat steps 5 and 6, only this time. Use a practical 4th order Butterworth filter.**

**%ssb butterworth**

flow=(fc-4e3)/(fs\_new/2);

fhigh=fc/(fs\_new/2);

fcut=[flow,fhigh];

[a,b] = butter(4,fcut);

yt\_ssb2 = filter(a,b,yt\_dsbsc).\*2;

yf\_ssb2 = fftshift(fft(yt\_ssb2));

f\_ssb2=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_ssb2));

t\_ssb2 = linspace(0, length(yt\_ssb2)/fs\_new, length(yt\_ssb2));

figure();

subplot(2,1,1);

plot(f\_ssb2,abs(yf\_ssb2)./fs\_new);

title('Single SideBand butter (frequency)');

xlabel('Frequency');

ylabel('Amplitude');

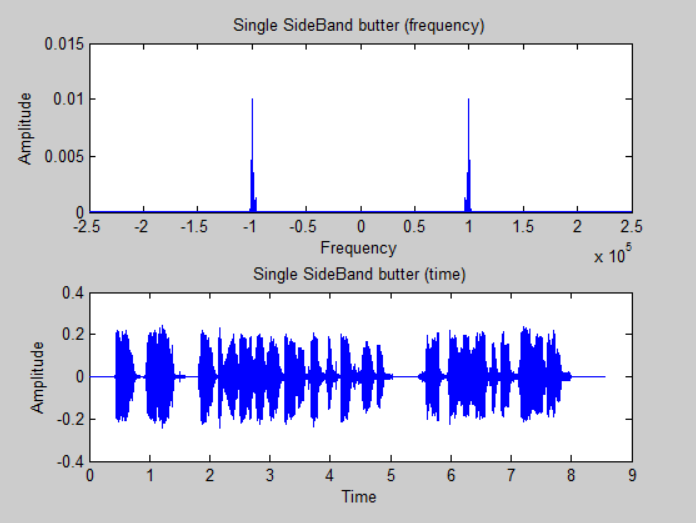
subplot(2,1,2);

plot(t\_ssb2,real(yt\_ssb2));

title('Single SideBand butter (time)');

xlabel('Time');

ylabel('Amplitude');



**%ssb butter cohdemodulation**

carrier5=cos(2\*pi\*fc\*t\_ssb2).';

yt\_ssbbutterdemod = carrier5.\*yt\_ssb2;

**%SSB cohbutter demod freq domain**

yf\_ssbbutterdemod = fftshift(fft(yt\_ssbbutterdemod));

**%filter ssb**

filter\_freq = 4000;

lp4 = 2.\*ones(length(yf\_ssbbutterdemod),1);

lp4(abs(f\_ssb2)>filter\_freq) = 0;

**%filtered ssb demodulated**

yf\_ssbbutterdemodf = lp4.\*yf\_ssbbutterdemod;

f\_ssbbutterf=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_ssbbutterdemodf));

yt\_ssbbutterdemodf = ifft(ifftshift(yf\_ssbbutterdemodf));

yt\_ssbbutterdemodf = resample(yt\_ssbbutterdemodf,fs,fs\_new);

t\_ssbbutterf = linspace(0, length(yt\_ssbbutterdemodf)/fs\_new, length(yt\_ssbbutterdemodf));

figure();

subplot(2,1,1);

plot(f\_ssbbutterf,abs(yf\_ssbbutterdemodf)./fs);

title('SSB butter cohdemod(frequency)');

xlabel('Frequency');

ylabel('ssb');

subplot(2,1,2);

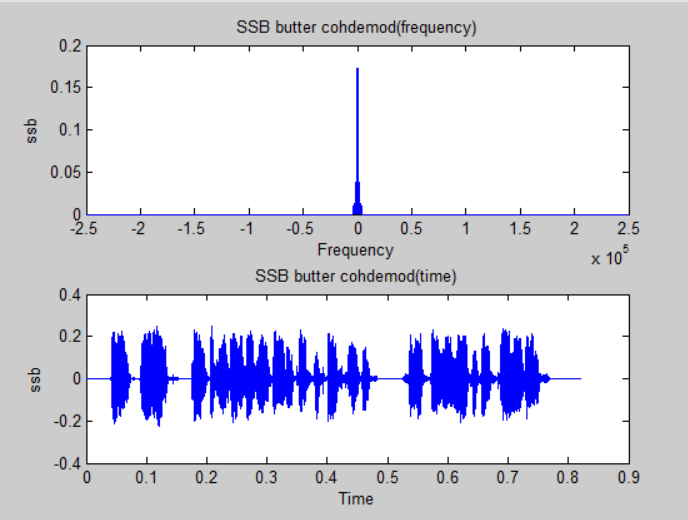
plot(t\_ssbbutterf,real(yt\_ssbbutterdemodf));

title('SSB butter cohdemod(time)');

xlabel('Time');

ylabel('ssb');

%sound(real(yt\_ssbbutterdemodf),fs);



**%(8) For the ideal filter case, get the received signal again but when noise is added to SSB-SC with**

**%SNR = 0, 10, and 30 also play the received sound and sketch the received waveform/spectrum in**

**%each case.**

**%SNR 0**

yt\_SNR0 = awgn(yt\_ssb, 0);

t\_SNR0=linspace(0, length(yt\_SNR0)/fs\_new, length(yt\_SNR0));

carrier2=cos(2\*pi\*fc\*t\_SNR0).';

yt\_SNR0demod = carrier2.\*yt\_SNR0;

yf\_SNR0demod = fftshift(fft(yt\_SNR0demod));

f\_SNR0=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR0demod));

**%LPF SNR0**

filter\_freq = 4000;

lp3 = 2.\*ones(length(yf\_SNR0demod),1);

lp3(abs(f\_SNR0)>filter\_freq) = 0;

**%filtered ssb demodulated**

yf\_SNR0demodf = lp3.\*yf\_SNR0demod;

f\_SNR0f=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR0demodf));

yt\_SNR0demodf= ifft(ifftshift(yf\_SNR0demodf));

yt\_SNR0demodf = resample(yt\_SNR0demodf,fs,fs\_new);

t\_SNR0f=linspace(0, length(yt\_SNR0demodf)/fs\_new, length(yt\_SNR0demodf));

figure();

subplot(2,1,1);

plot(f\_SNR0f,abs(yf\_SNR0demodf)./fs);

title('Single SideBand SNR0 demod (frequency)');

xlabel('Frequency');

ylabel('ssb');

subplot(2,1,2);

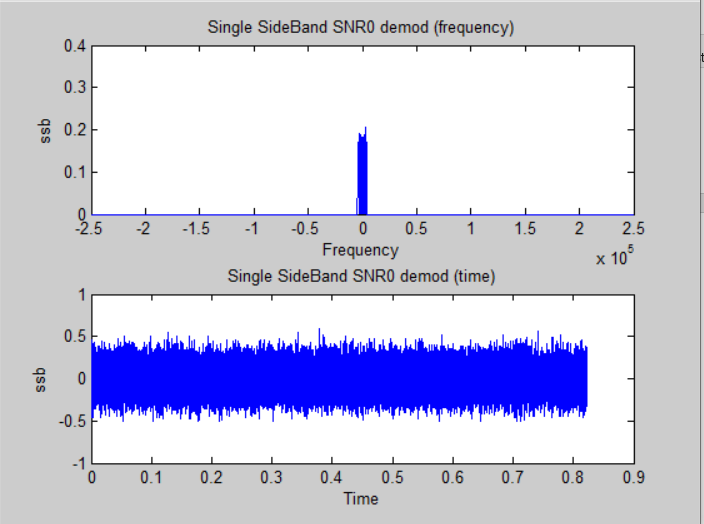
plot(t\_SNR0f,real(yt\_SNR0demodf));

title('Single SideBand SNR0 demod (time)');

xlabel('Time');

ylabel('ssb');

%sound(real(yt\_SNR0demodf),fs);



**%SNR 10**

yt\_SNR10 = awgn(yt\_ssb, 10);

t\_SNR10=linspace(0, length(yt\_SNR10)/fs\_new, length(yt\_SNR10));

carrier3=cos(2\*pi\*fc\*t\_SNR10).';

yt\_SNR10demod = carrier3.\*yt\_SNR10;

yf\_SNR10demod = fftshift(fft(yt\_SNR10demod));

f\_SNR10=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR10demod));

**%lpf**

filter\_freq = 4000;

lp10 = 2.\*ones(length(yf\_SNR10demod),1);

lp10(abs(f\_SNR10)>filter\_freq) = 0;

**%filtered ssb demodulated**

yf\_SNR10demodf = lp10.\*yf\_SNR10demod;

yt\_SNR10demodf= ifft(ifftshift(yf\_SNR10demodf));

f\_SNR10f=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR10demodf));

yt\_SNR10demodf = resample(yt\_SNR10demodf,fs,fs\_new);

t\_SNR10f=linspace(0, length(yt\_SNR10demodf)/fs\_new, length(yt\_SNR10demodf));

figure();

subplot(2,1,1);

plot(f\_SNR10f,abs(yf\_SNR10demodf)./fs);

title('Single SideBand SNR10 demod (frequency)');

xlabel('Frequency');

ylabel('ssb');

subplot(2,1,2);

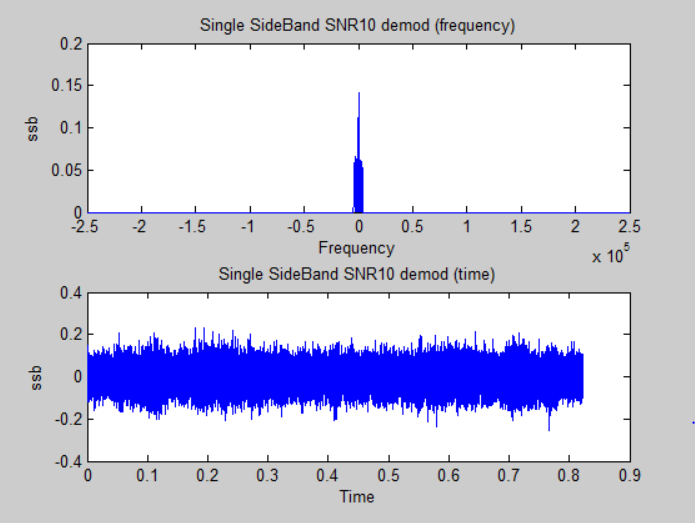
plot(t\_SNR10f,real(yt\_SNR10demodf));

title('Single SideBand SNR10 demod (time)');

xlabel('Time');

ylabel('ssb');

%sound(real(yt\_SNR10demodf),fs);



**%SNR 30**

yt\_SNR30 = awgn(yt\_ssb, 30);

t\_SNR30=linspace(0, length(yt\_SNR30)/fs\_new, length(yt\_SNR30));

carrier4=cos(2\*pi\*fc\*t\_SNR30).';

yt\_SNR30demod = carrier3.\*yt\_SNR30;

yf\_SNR30demod = fftshift(fft(yt\_SNR30demod));

f\_SNR30=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR30demod));

**%LPF SNR30**

filter\_freq = 4000;

lp30 = 2.\*ones(length(yf\_SNR30demod),1);

lp30(abs(f\_SNR30)>filter\_freq) = 0;

%filtered ssb demodulated

yf\_SNR30demodf = lp30.\*yf\_SNR30demod;

yt\_SNR30demodf = ifft(ifftshift(yf\_SNR30demodf));

f\_SNR30f=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_SNR30demodf));

yt\_SNR30demodf = resample(yt\_SNR30demodf,fs,fs\_new);

t\_SNR30f=linspace(0, length(yt\_SNR30demodf)/fs\_new, length(yt\_SNR30demodf));

figure();

subplot(2,1,1);

plot(f\_SNR30f,abs(yf\_SNR30demodf)./fs\_new);

title('Single SideBand SNR30 demod (frequency)');

xlabel('Frequency');

ylabel('ssb');

subplot(2,1,2);

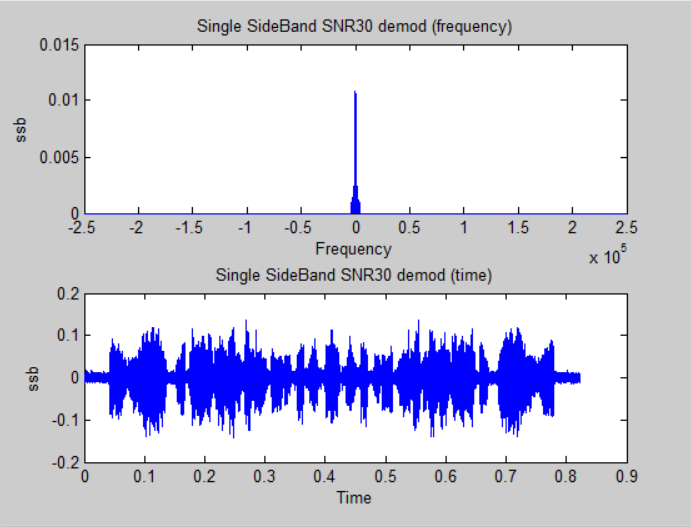
plot(t\_SNR30f,yt\_SNR30demodf);

title('Single SideBand SNR30 demod (time)');

xlabel('Time');

ylabel('ssb');

%sound(real(yt\_SNR30demodf),fs);



**%SSB-TC**

f\_ssbtc=linspace(-(fs\_new/2),(fs\_new/2),length(yf\_dsbtc));

yf\_dsbtc(f\_ssbtc>100001) = 0;

yf\_dsbtc(f\_ssbtc<-100001) = 0;

yt\_dsbtcf=ifft(ifftshift(yf\_dsbtc));

t\_ssbtc3 = linspace(0, length(yt\_dsbtcf)/fs\_new, length(yt\_dsbtcf));

figure();

subplot(2,1,1);

plot(t\_ssbtc3,real(yt\_dsbtc));

title('SSBTC Signal modulated');

xlabel('Time');

ylabel('Amplitude');

subplot(2,1,2);

plot(f\_new,abs(yf\_dsbtc)./fs\_new);

title('SSBTC Signal modulated');

xlabel('Frequency');

ylabel('Amplitude');

yt\_ssbtc\_env = resample(abs(hilbert(yt\_dsbtc)),fs,fs\_new)-(2\*Am);

yf\_ssbtc\_env = real(fftshift(fft(yt\_ssbtc\_env)));

t\_ssbtc = linspace(0, length(yt\_ssbtc\_env)/fs, length(yt\_ssbtc\_env));

figure();

plot(t\_ssbtc,yt\_ssbtc\_env);

title('SSBTC Signal demodulated');

xlabel('Time');

ylabel('Amplitude');

%sound(real(yt\_ssbtc\_env),fs);

A screenshot of a computer

Description automatically generated with medium confidence

**EXPERIMENT THREE: FREQUENCY MODULATION**

%NBFM

%Modulation

Fc\_NBFM = 100000;

%sampling frequency

Fs\_NBFM = 5\*Fc\_NBFM;

%Amplitude of the message

Am = max(abs(message));

%equation:

%cos(2\*pi\*fc\*t + 2\*pi\*kf\*integ\_message);

%expression : 2\*pi\*kf\*integ\_message < 1 rad

Resampled\_NBFM = resample(yt\_filtered,Fs\_NBFM,fs);

integ\_message = cumsum(Resampled\_NBFM)/Fs\_NBFM;

Kf = 0.2/(2\*pi\*max(integ\_message));

%carrier amplitude

Ac\_NBFM = max(abs(Resampled\_NBFM));

time\_NBFM = linspace(0, (length(Resampled\_NBFM))/Fs\_NBFM, length(Resampled\_NBFM));

time\_NBFM = transpose(time\_NBFM);

freq\_NBFM = linspace((-Fs\_NBFM)/2, Fs\_NBFM/2, length(Resampled\_NBFM));

NBFM\_Signal = Ac\_NBFM\*cos(2\*pi\*Fc\_NBFM\*time\_NBFM + 2\*pi\*Kf\*integ\_message);

NBFM\_signal\_spectrum = fftshift (fft(NBFM\_Signal)./Fs\_NBFM);

figure()

subplot(2,1,1);

plot(time\_NBFM, real(NBFM\_Signal));

title('NBFM modulated signal in time domain');

xlabel('Time');

ylabel('Amplitude');

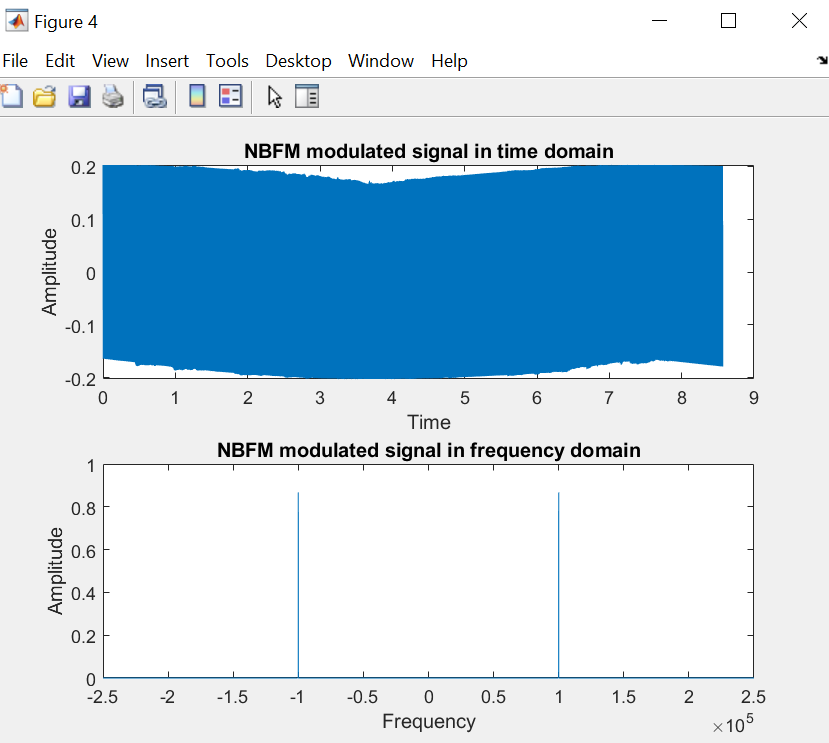
subplot(2,1,2);

plot(freq\_NBFM, abs(NBFM\_signal\_spectrum));

title('NBFM modulated signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');



**\*What can you make out of the resulting plot?**

From the time domain, we could conclude that the signal resulted from the Narrow Band FM is similar to DSB-TC. Meanwhile, we can conclude, from the frequency domain, that the spectrum is mostly the carrier due to the small variations in the frequency.

**\*What is the condition we needed to achieve NBFM?**

Given that: U(t) = Ac\* cos(2\*pi\*fc\*t + (t)), where = 2\*pi\* kf \* integration(message)

For the Narrow Band: 1 rad so,

And given that Beta = , where beta is the frequency deviation ratio or modulation index , is the frequency deviation Generally:

to obtain a Narrow Band: and the bandwidth is approximately 2fm

\*Kf which is the frequency deviation constant or frequency sensitivity must be small to decrease the frequency deviation.

%Demodulation

Output\_Signal\_NBFM\_diff = diff(NBFM\_Signal)\*Fs\_NBFM;

Output\_Signal\_demod\_time = abs(hilbert(Output\_Signal\_NBFM\_diff));

Output\_Signal\_demod\_time = Output\_Signal\_demod\_time - abs(mean(Output\_Signal\_demod\_time));

Output\_Signal\_demod\_time\_resampled = resample(Output\_Signal\_demod\_time,fs,Fs\_NBFM);

Output\_Signal\_demod\_freq = fftshift(fft(Output\_Signal\_demod\_time\_resampled)./Fs\_NBFM);

figure()

subplot(2,1,1);

time\_demod =linspace(0, (length(Output\_Signal\_demod\_time\_resampled))/fs, length(Output\_Signal\_demod\_time\_resampled));

plot(time\_demod,0.01\*Output\_Signal\_demod\_time\_resampled);

ylim([-0.2 0.2]);

title('Demodulated signal in time domain');

xlabel('Time');

ylabel('Amplitude');

sound(Output\_Signal\_demod\_time\_resampled,fs);

subplot(2,1,2);

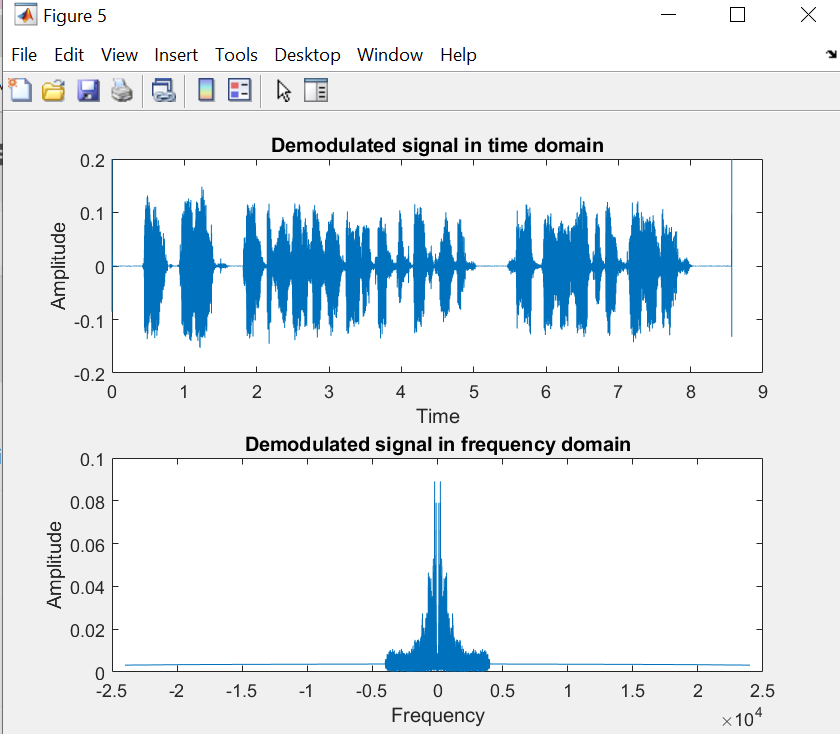
freq\_demod = linspace(-fs/2, fs/2, length(Output\_Signal\_demod\_freq));

plot(freq\_demod,abs(Output\_Signal\_demod\_freq));

title('Demodulated signal in frequency domain');

xlabel('Frequency');

ylabel('Amplitude');

****