COMMUNICATION SYSTEMS LAB 6 DATE-05/10/2021

NAME – JAGRIT LODHA

ID – 2019A3PS0165P

SECTION – P4

MATLAB

TASK 1 –

clear all

close all

duration\_signal=29;

N=12; % Sum of last 3 digits of ID

for T = 0:duration\_signal %%%% Duration 30 seconds with interval of 1 sec.

if T==0

display('Transmission Started')

display (T)

elseif (T==duration\_signal)

display('Transmission ends: see the final result')

display (T)

else

display('Transmission in progress: please wait')

display (T)

end

fm = N;

fs = 4\*fm;

ts = 1/fs;

t = -0.5:ts:0.5;

i = randi(3);

display(i)

m1\_t = cos(2\*pi\*N\*t);

m2\_t = 2\*N\*sinc(2\*N\*t);

Nsym = 200; %filter span in symbol durations

L=10;

alpha=1;

m3\_t = 200\*(cos(200\*pi\*t)./(1-40000\*t.\*t)).\*sinc(200\*t);

switch i

case 1

m\_t = m1\_t;

case 2

m\_t = m2\_t;

case 3

m\_t = m3\_t;

end

Ac = 2; % Amplitude of carrier signal.

fc = 1000;

carrier = Ac.\*cos(2\*pi\*fc\*t); % carrier signal.

dsb\_sc = m\_t.\*carrier; %dsb sc modulated wave

h\_t = 2\*100\*sinc(2\*100\*t); % Channel of low bandwidth = 100Hz

y1\_t = conv(dsb\_sc, h\_t, 'same');

n\_t = 0.01\*randn(1, size(dsb\_sc,2));

y\_t = y1\_t + n\_t;

% DSB SC IN FREQUENCY DOMAIN

ld = length(dsb\_sc);

f = linspace(-fs/2,fs/2,ld);

DSB\_SC = fftshift(fft(dsb\_sc,ld)/ld); %frequency spectrum of dsb\_sc modulated signal

% DSB SC DEMODULATION TIME DOMAIN

pmo = y\_t.\*carrier;

pmo = pmo/(Ac\*Ac);

nf = fm/fs; %normalised frequency

[num, den] = butter(5,3\*nf); %butter worth lpf of 5th order

msg\_r = filter(num,den,pmo); %demodulated signal after passing through lpf

% DSB SC DEMODULATION FREQUENCY DOMAIN

lr = length(msg\_r);

fr = linspace(-fs/2,fs/2,lr); %frequency bins

MSG\_R = fftshift(fft(msg\_r,lr)/lr); %frequency spectrum of demodulated signal

% PLOTS

figure(1)

hold all;

subplot(4,1,1);

plot(t+T, m\_t);

title("MESSAGE SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

ylim([-5 30]);

grid on;

hold on;

subplot(4,1,2);

plot(t+T, carrier);

title("CARRIER SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

grid on;

hold on;

subplot(4,1,3);

plot(t+T, dsb\_sc);

title("MODULATED DSB SC SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

grid on;

hold on;

subplot(4,1,4);

plot(t+T, msg\_r);

title("DEMODULATED DSB SC SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

grid on;

hold on;

figure(2)

hold all

subplot(2,1,1);

plot(f, abs(DSB\_SC));

xlim([-15 15]);

title('DSB SC MODULATION IN FREQUENCY DOMAIN');

xlabel('frequency(hz)');

ylabel('amplitude');

grid on;

hold on;

subplot(2,1,2);

plot(fr, abs(MSG\_R));

xlim([-6 6]);

title('DSB SC DEMODULATION IN FREQUENCY DOMAIN');

xlabel('frequency(hz)');

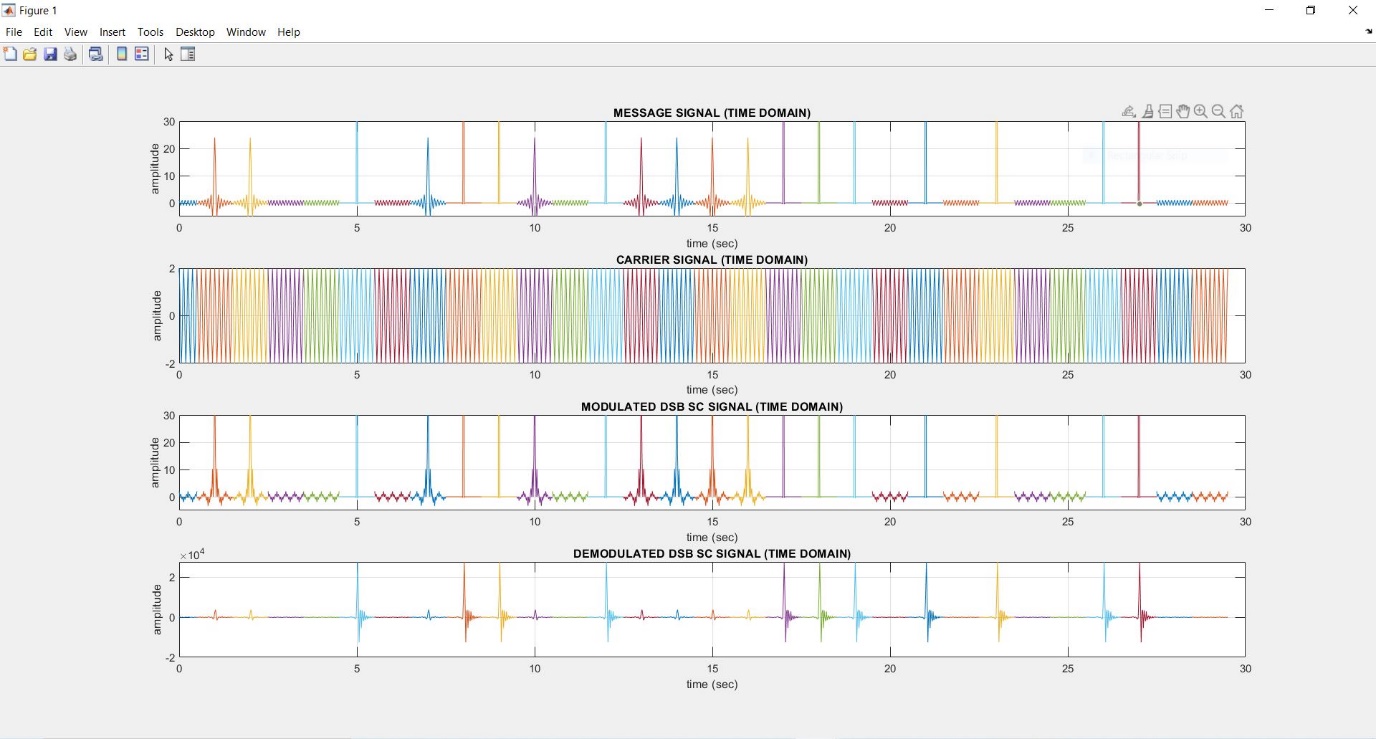
ylabel('amplitude');

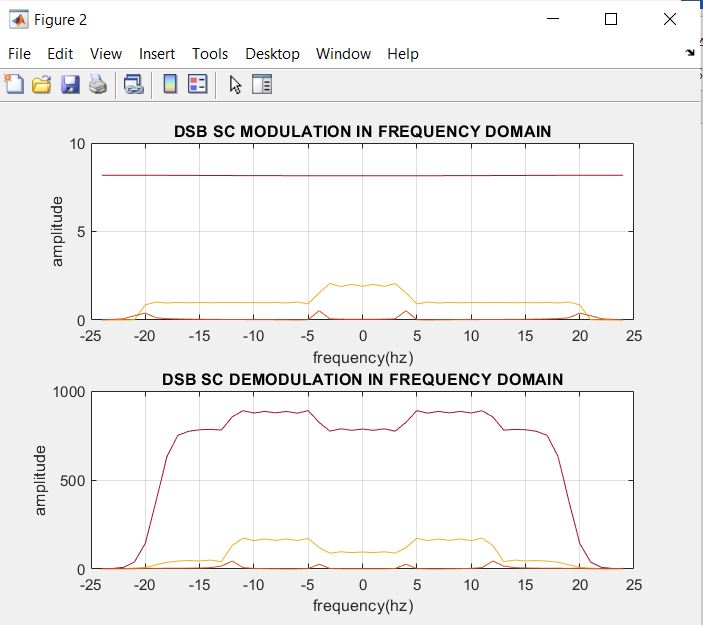
grid on;

pause(1) %%%%% pauses for 1 seconds and then go for next loop increment.

end

Required Plots -

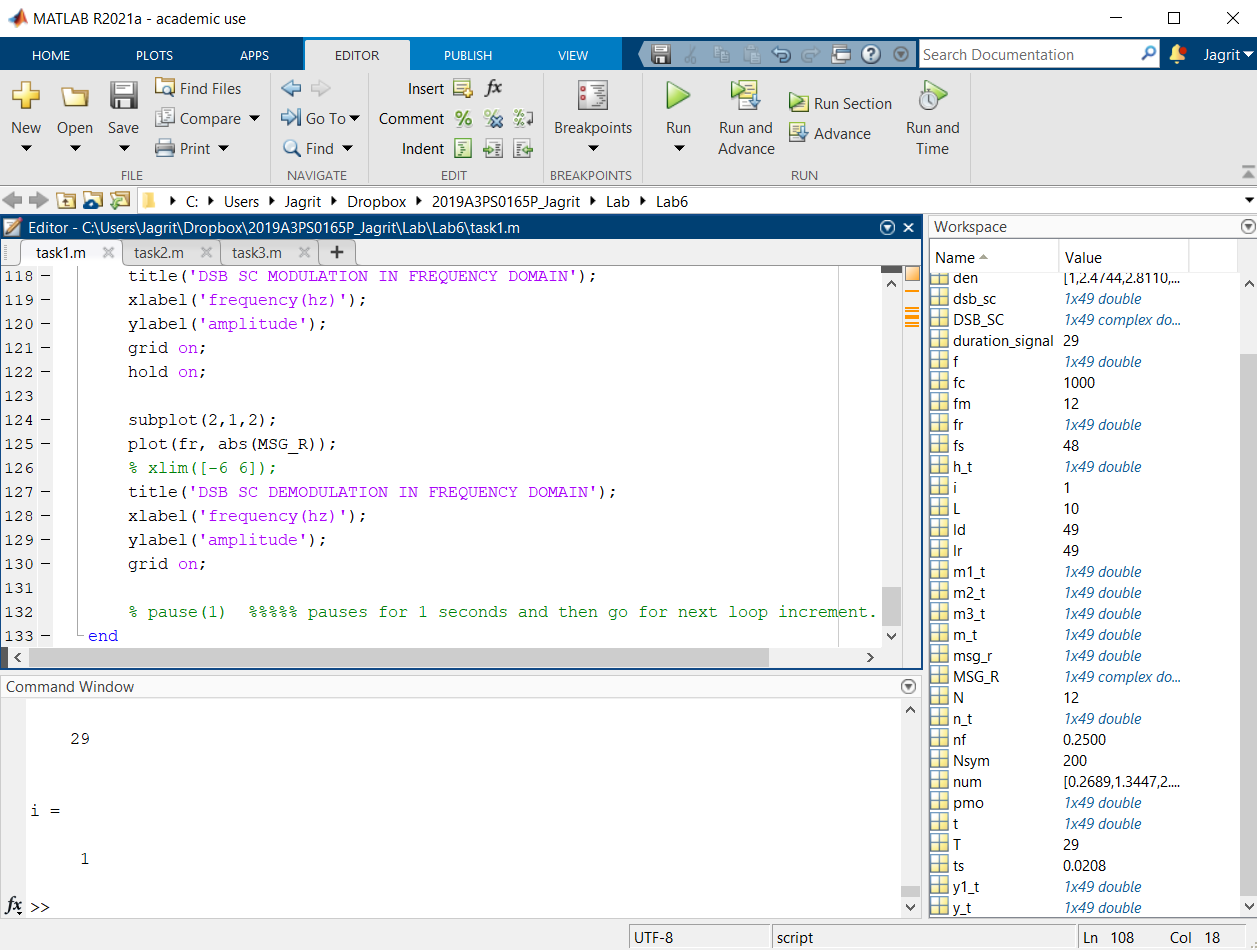




NOTE –

* For the first time-domain plot, we put the y-limits from -5 to 30 for better visualization. The sinc signal goes upto an amplitude of 24, the cos signal till 1 and the raised cosine pulse to 200 (which is not fully visible).
* Similarly in the third plot, the y-limit has been changed. (here the corresponding peak amplitudes are 48, 2 and 400).
* We can see how we get almost identical demodulated plots for sinc and cosine, while the raised cosine differs slightly, probably due to addition of AWGN.
* The same trend can be observed in the frequency domain as well.

Screenshot of MATLAB Terminal –



TASK 2 –

clear all

close all

duration\_signal=29;

N=12; % Sum of last 3 digits of ID

for T = 0:duration\_signal %%%% Duration 30 seconds with interval of 1 sec.

if T==0

display('Transmission Started')

display (T)

elseif (T==duration\_signal)

display('Transmission ends: see the final result')

display (T)

else

display('Transmission in progress: please wait')

display (T)

end

fm = N;

fs = 4\*fm;

ts = 1/fs;

t = -0.5:ts:0.5;

i = randi(3);

display(i)

m1\_t = cos(2\*pi\*N\*t);

m2\_t = 2\*N\*sinc(2\*N\*t);

Nsym = 200; %filter span in symbol durations

L=10;

alpha=1;

m3\_t = 200\*(cos(200\*pi\*t)./(1-40000\*t.\*t)).\*sinc(200\*t);

switch i

case 1

m\_t = m1\_t;

case 2

m\_t = m2\_t;

case 3

m\_t = m3\_t;

end

Ac = 2; % Amplitude of carrier signal.

fc = 1000;

carrier = Ac.\*cos(2\*pi\*fc\*t); % carrier signal.

dsb\_sc = m\_t.\*carrier; %dsb sc modulated wave

h\_t = 2\*100\*sinc(2\*100\*t); % Channel of low bandwidth = 100Hz

y1\_t = conv(dsb\_sc, h\_t, 'same');

n\_t = 0.01\*randn(1, size(dsb\_sc,2));

y\_t = y1\_t + n\_t;

% DSB SC IN FREQUENCY DOMAIN

ld = length(dsb\_sc);

f = linspace(-fs/2,fs/2,ld);

DSB\_SC = fftshift(fft(dsb\_sc,ld)/ld); %frequency spectrum of dsb\_sc modulated signal

% DSB SC DEMODULATION TIME DOMAIN

msg\_r = hilbert(y\_t).\*exp(-1i\*2\*pi\*fc\*t); %demodulated signal after passing through lpf

% DSB SC DEMODULATION FREQUENCY DOMAIN

lr = length(msg\_r);

fr = linspace(-fs/2,fs/2,lr); %frequency bins

MSG\_R = fftshift(fft(msg\_r,lr)/lr); %frequency spectrum of demodulated signal

% PLOTS

figure(1)

hold all;

subplot(4,1,1);

plot(t+T, m\_t);

title("MESSAGE SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

ylim([-5 30]);

xlim([0 30]);

grid on;

hold on;

subplot(4,1,2);

plot(t+T, carrier);

title("CARRIER SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

xlim([0 30]);

grid on;

hold on;

subplot(4,1,3);

plot(t+T, dsb\_sc);

title("MODULATED DSB SC SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

ylim([-5 30]);

xlim([0 30]);

grid on;

hold on;

subplot(4,1,4);

plot(t+T, msg\_r);

title("DEMODULATED DSB SC SIGNAL (TIME DOMAIN)");

xlabel('time (sec)');

ylabel('amplitude');

% ylim([-5 30]);

xlim([0 30]);

grid on;

hold on;

figure(2)

hold all

subplot(2,1,1);

plot(f, abs(DSB\_SC));

% xlim([-15 15]);

title('DSB SC MODULATION IN FREQUENCY DOMAIN');

xlabel('frequency(hz)');

ylabel('amplitude');

grid on;

hold on;

subplot(2,1,2);

plot(fr, abs(MSG\_R));

% xlim([-6 6]);

title('DSB SC DEMODULATION IN FREQUENCY DOMAIN');

xlabel('frequency(hz)');

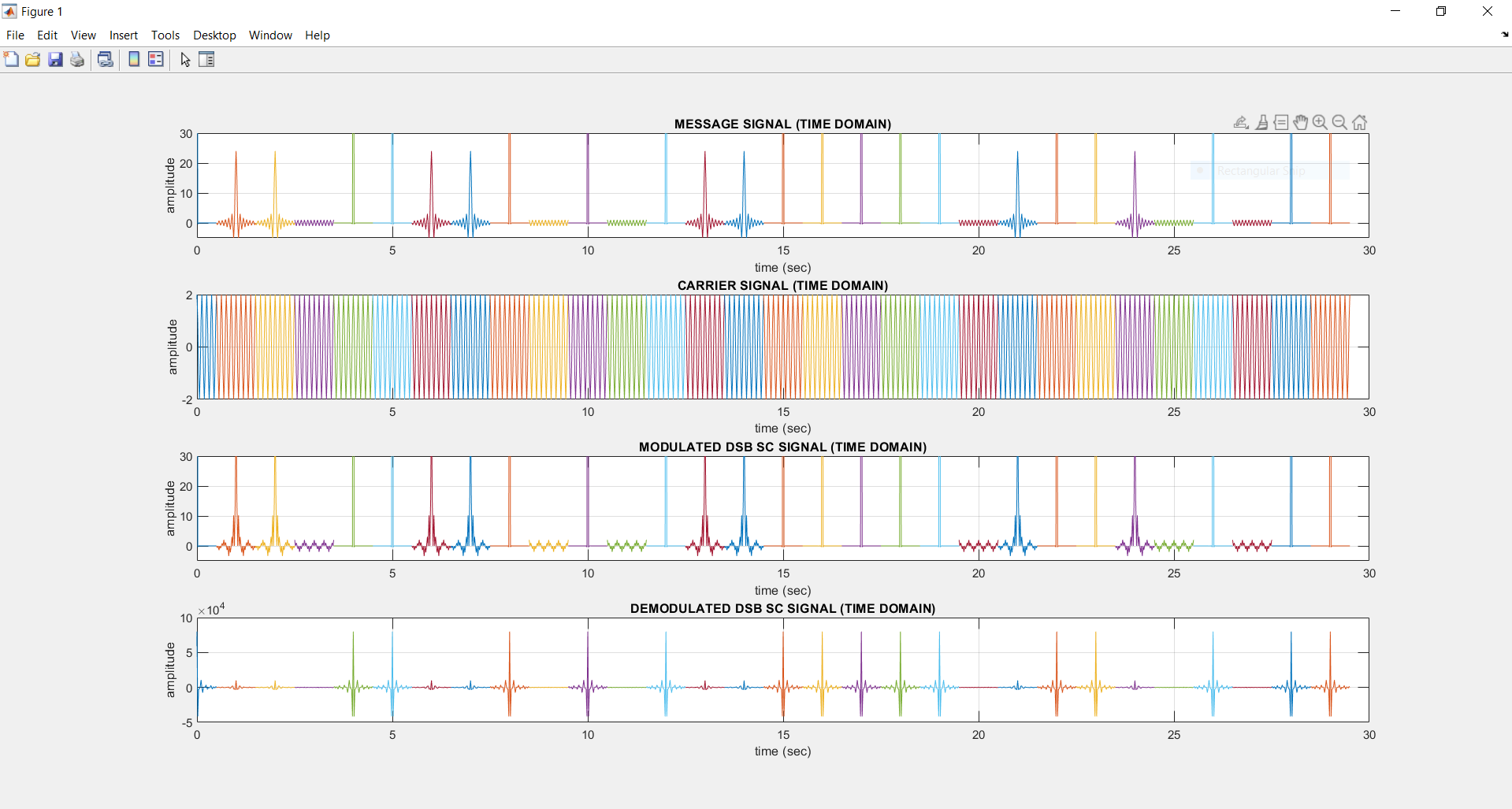
ylabel('amplitude');

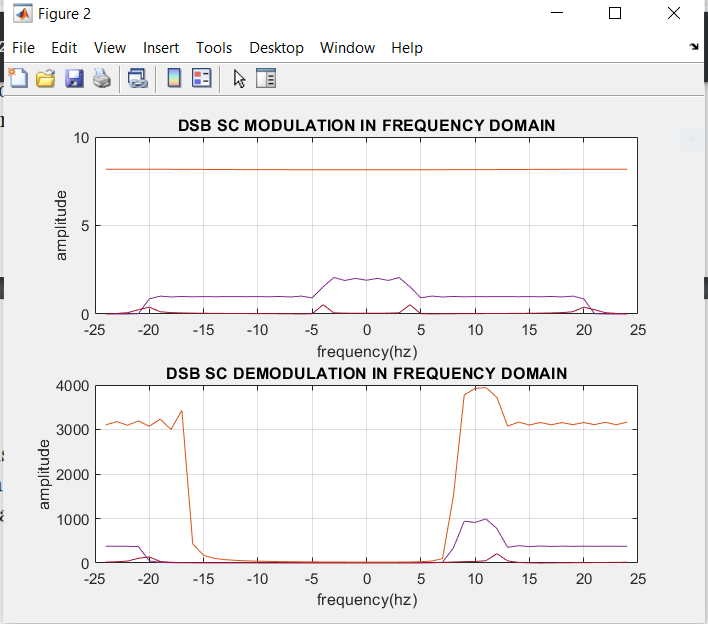
grid on;

% pause(1) %%%%% pauses for 1 seconds and then go for next loop increment.

end

Screenshot of required plots –

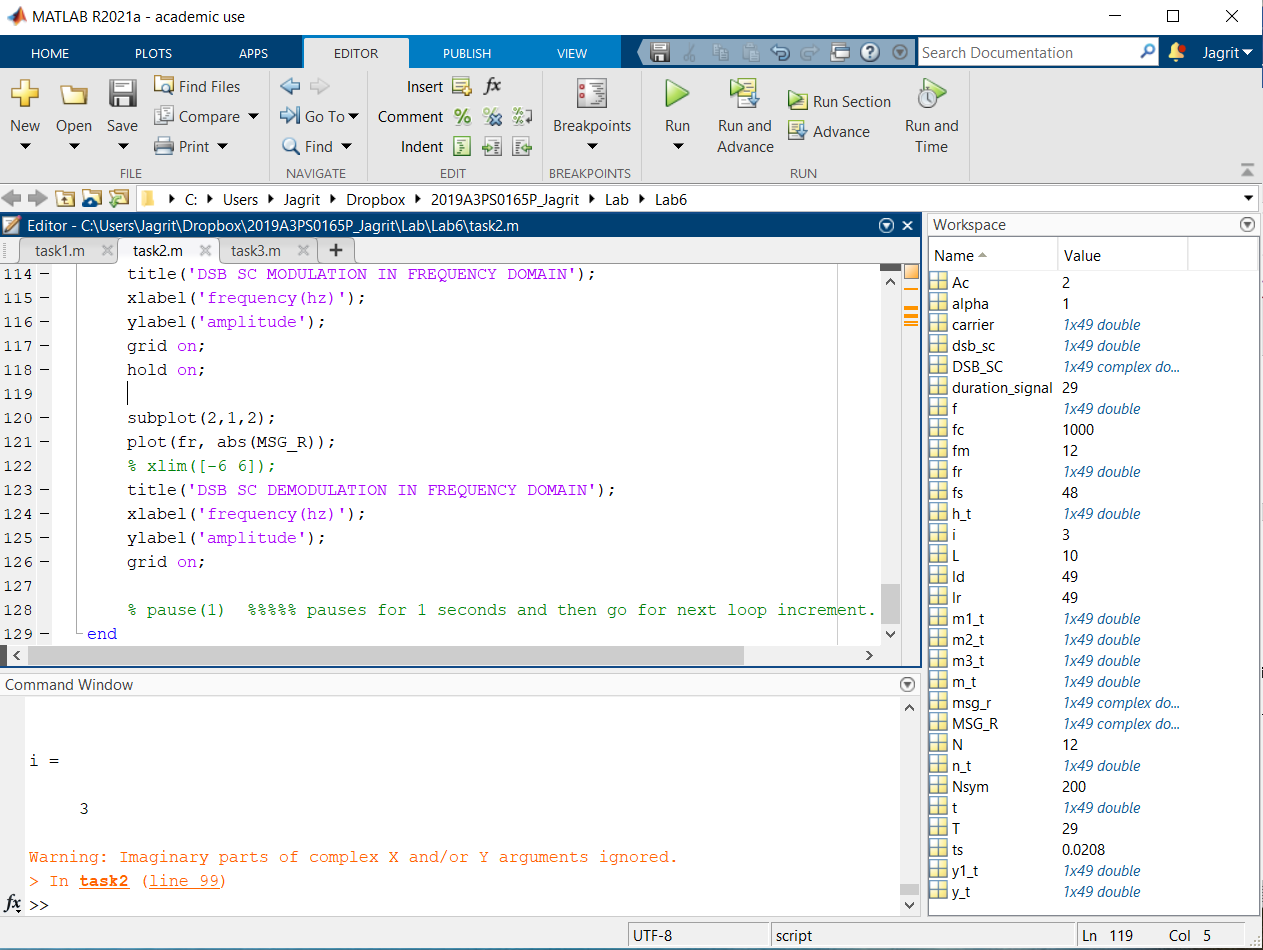




NOTE –

* The results obtained are almost same as those of task 1 with the exception that the raised cosine appears to have been demodulated much more accurately.
* The demodulated frequency domain is a bit distorted probably due to the real and imaginary factors of the Hilbert Filter that we have used.

Screenshot of MATLAB Terminal –



TASK 3 –

Will be trying it later with the downloaded audio file because the sampling frequency creates various complexities in the code.