Communication systems Lab 4

Harshal Abhyankar 2019A3PS0282P

N=2.

**Task 1**

clear all

close all

duration\_signal=29;

A=1;

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

freq=2;

fs=4\*freq;

ts=1/fs;

t=0:ts:1;

U=randi(5);

m\_t=U\*cos(2\*pi\*freq\*t);

N=length(m\_t);

m\_f= fft(m\_t,N)/fs;

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

figure(1)

hold all %%% keeps the previous plots and everytime changes the color

subplot(2,1,1), plot(t+T,m\_t);

xlabel('time')

ylabel('amplitude')

grid on

axis([0 inf -5 5]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why 0 inf , and -5 5 are used here.

hold on %%% keeps the previous plots

subplot(2,1,2), plot(freqaxis,fftshift(abs(m\_f)))

xlabel('frequency (Hz)')

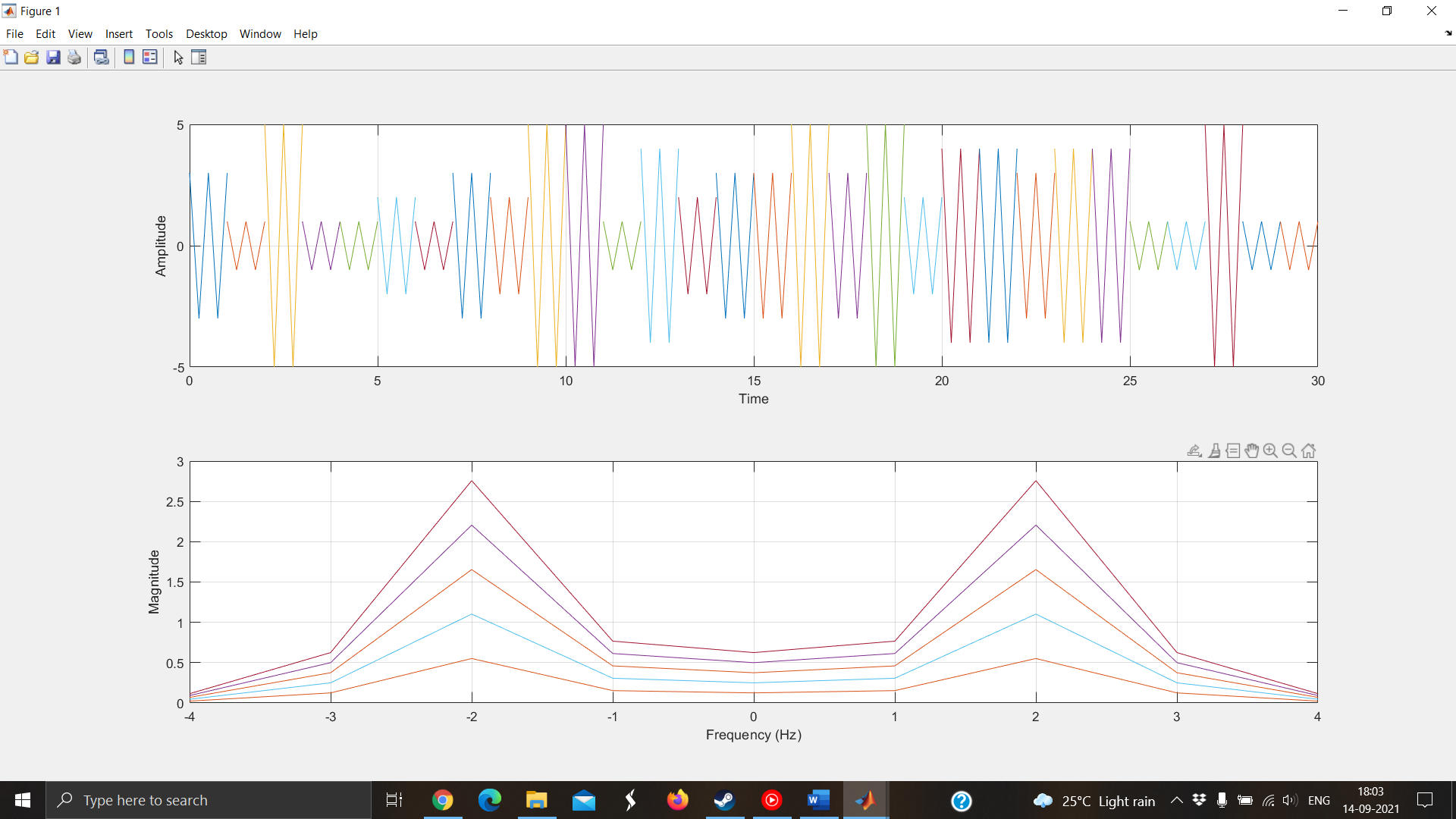
ylabel('Magnitude')

grid on

axis([-inf inf 0 3]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why -inf inf , and 0 3 are used here.

pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

end



**Conclusion: The required signal with random U for each second was successfully transmitted and displayed in real time. In the frequency domain there are peaks at -N and N respectively.**

**Task 2**

**Code:**

clear all

close all

duration\_signal=29;

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

freq=2;

fs=4\*freq;

ts=1/fs;

t=0:ts:1;

i=randi(3);

display(i)

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

m2\_t=2\*freq\*sinc(2\*freq\*pi\*t);

m3\_t=rectangularPulse(-freq/2,freq/2,t);

m4\_t=audioread("sample.wav");

switch i

case 1

m\_t=m1\_t

case 2

m\_t=m2\_t

case 3

m\_t=m3\_t

case 4

m\_t=m4\_t

end

N=length(m\_t);

m\_f= fft(m\_t,N)/fs;

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

figure(1)

hold all %%% keeps the previous plots and everytime changes the color

subplot(2,1,1), plot(t+T,m\_t);

xlim([0 30]);

xlabel('Time')

ylabel('Amplitude')

grid on

axis([0 inf -5 5]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why 0 inf , and -5 5 are used here.

hold on %%% keeps the previous plots

subplot(2,1,2), plot(freqaxis,fftshift(abs(m\_f)))

xlabel('Frequency (Hz)')

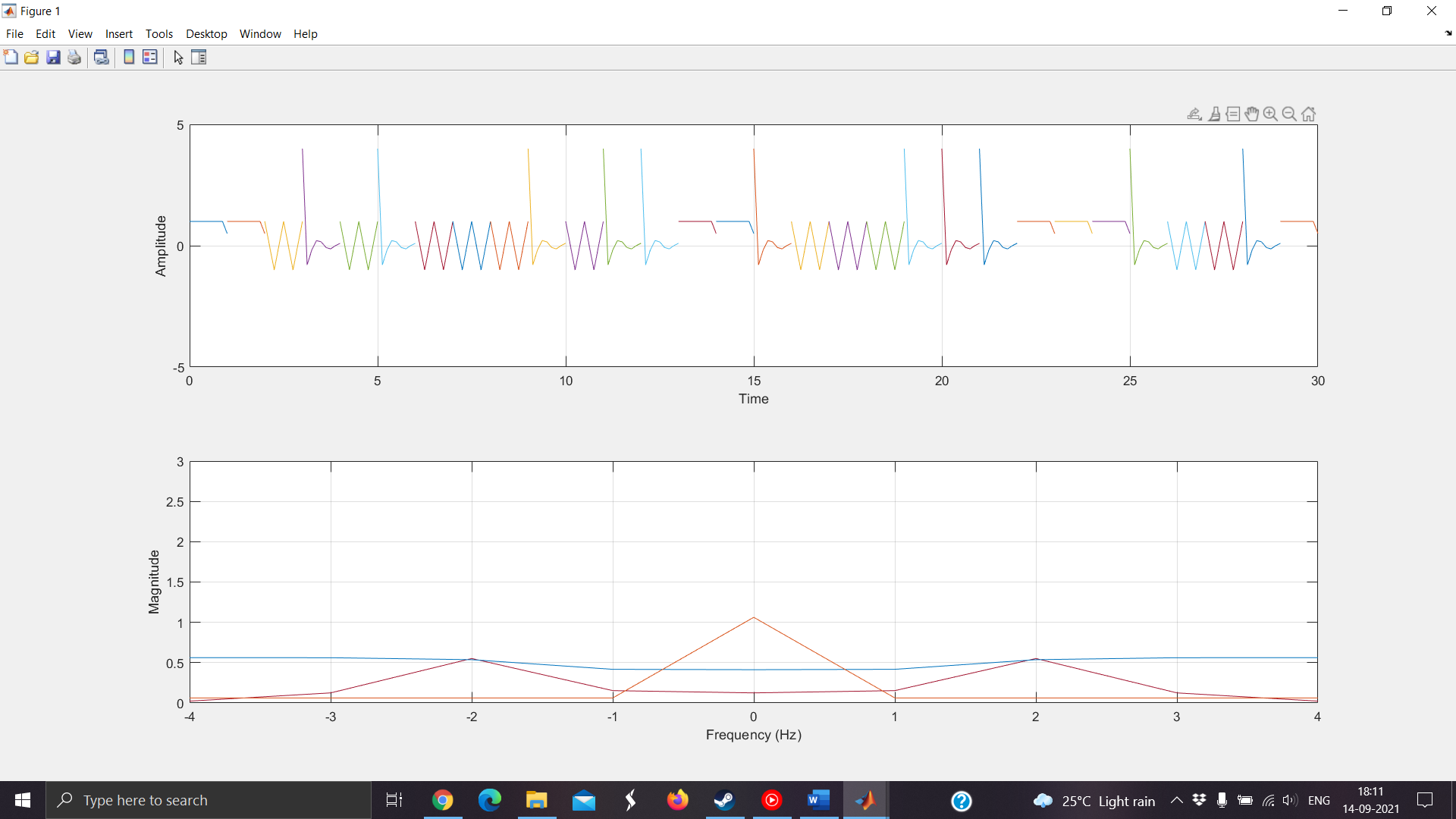
ylabel('Magnitude')

grid on

axis([-inf inf 0 3]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why -inf inf , and 0 3 are used here.

%pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

end



**Conclusion: In this task we plot a message selected randomly from the four signals each second and it is transmitted and displayed in real time.**

**Task 3**

Code:

clear all

close all

duration\_signal=29;

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

freq=100;

fs=3\*freq;

ts=1/fs;

t=0:ts:1;

f1 = randi([10 100],1,1);

f2 = randi([10 100],1,1);

N=2;

m1\_t = N\*cos(2\*pi\*f1\*t) + N\*cos(2\*pi\*f2\*t);

t1=-5:ts:5;

ht=100\*sinc(100\*t1);

m\_t=conv(m1\_t, ht, 'same');

N=length(m\_t);

m\_f= fft(m\_t,N)/fs;

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

figure(1)

hold all %%% keeps the previous plots and everytime changes the color

subplot(2,1,1), plot(t+T,m\_t);

xlabel('time')

ylabel('amplitude')

grid on

axis([0 inf -5 5]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why 0 inf , and -5 5 are used here.

hold on %%% keeps the previous plots

subplot(2,1,2), plot(freqaxis,fftshift(abs(m\_f)))

xlabel('frequency (Hz)')

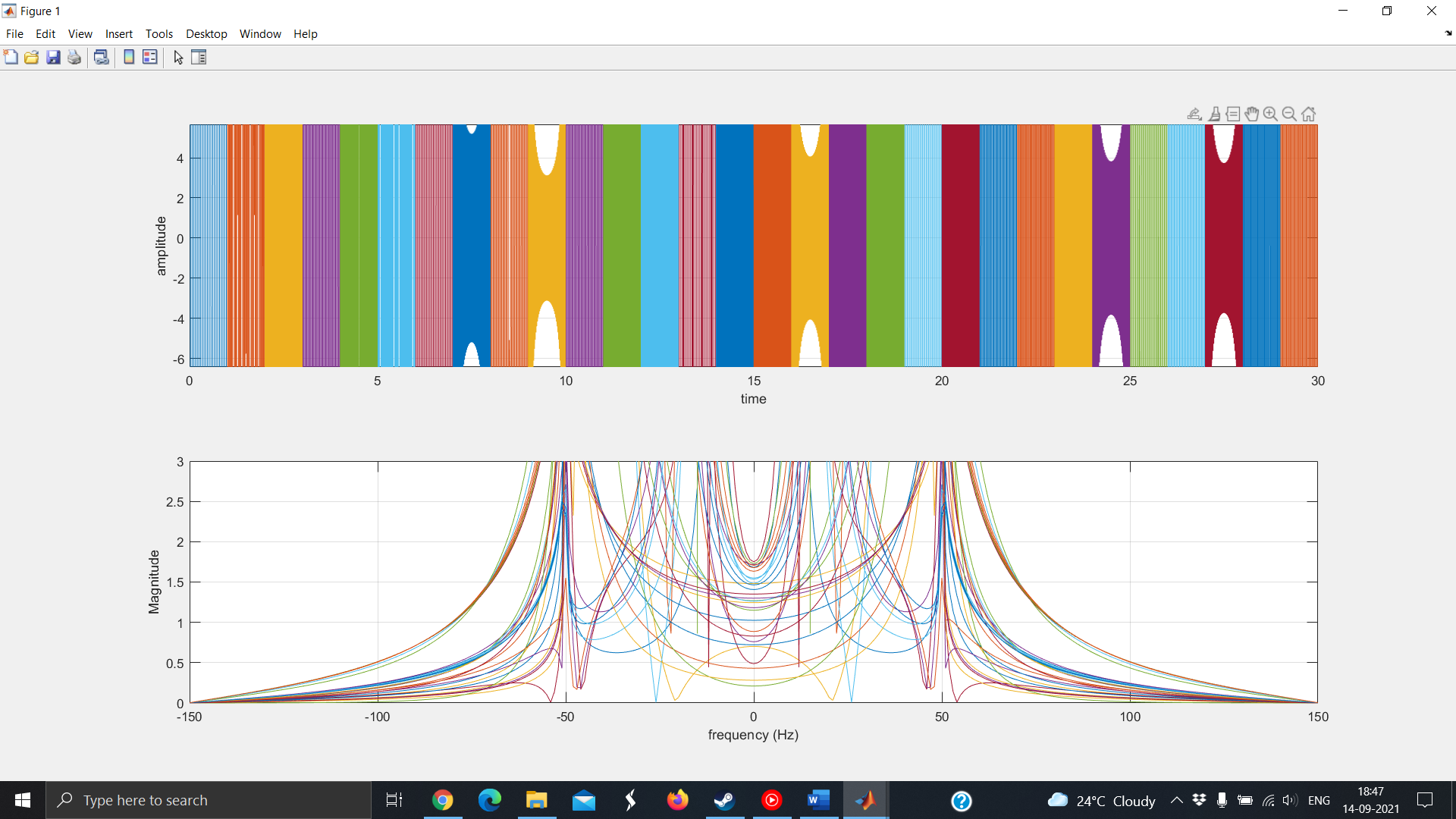
ylabel('Magnitude')

grid on

axis([-inf inf 0 3]) %%% first two are limits for x-axis, the other two are limits for y-axis: observe why -inf inf , and 0 3 are used here.

pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

end



**Conclusion: At each second the message signal is sum of two cos functions with frequencies f1 and f2 selected randomly which get cutoff at 50 Hz due to 50Hz band limited channel and we get peaks between -50 and 50 Hz in frequency domain.**

**Task 4**

Code:

clear all

close all

duration\_signal=29;

A=1;

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

freq=2;

fs=8\*freq;

ts=1/fs;

t=0:ts:1;

mt= 2\*freq\*sinc(2\*freq\*(t));

mht=hilbert(mt);

x\_t=mt+1j\*mht;

len=length(x\_t);

freqaxis=linspace(-fs/2, fs/2, len);

x\_f=fft(x\_t,len)/fs;

figure(1)

hold all %%% keeps the previous plots and everytime changes the color

subplot(2,1,1), plot(t+T,real(x\_t),t+T , imag(x\_t));

xlabel('time')

ylabel('amplitude')

grid on

hold on %%% keeps the previous plots

subplot(2,1,2), plot(freqaxis,fftshift(abs(x\_f)))

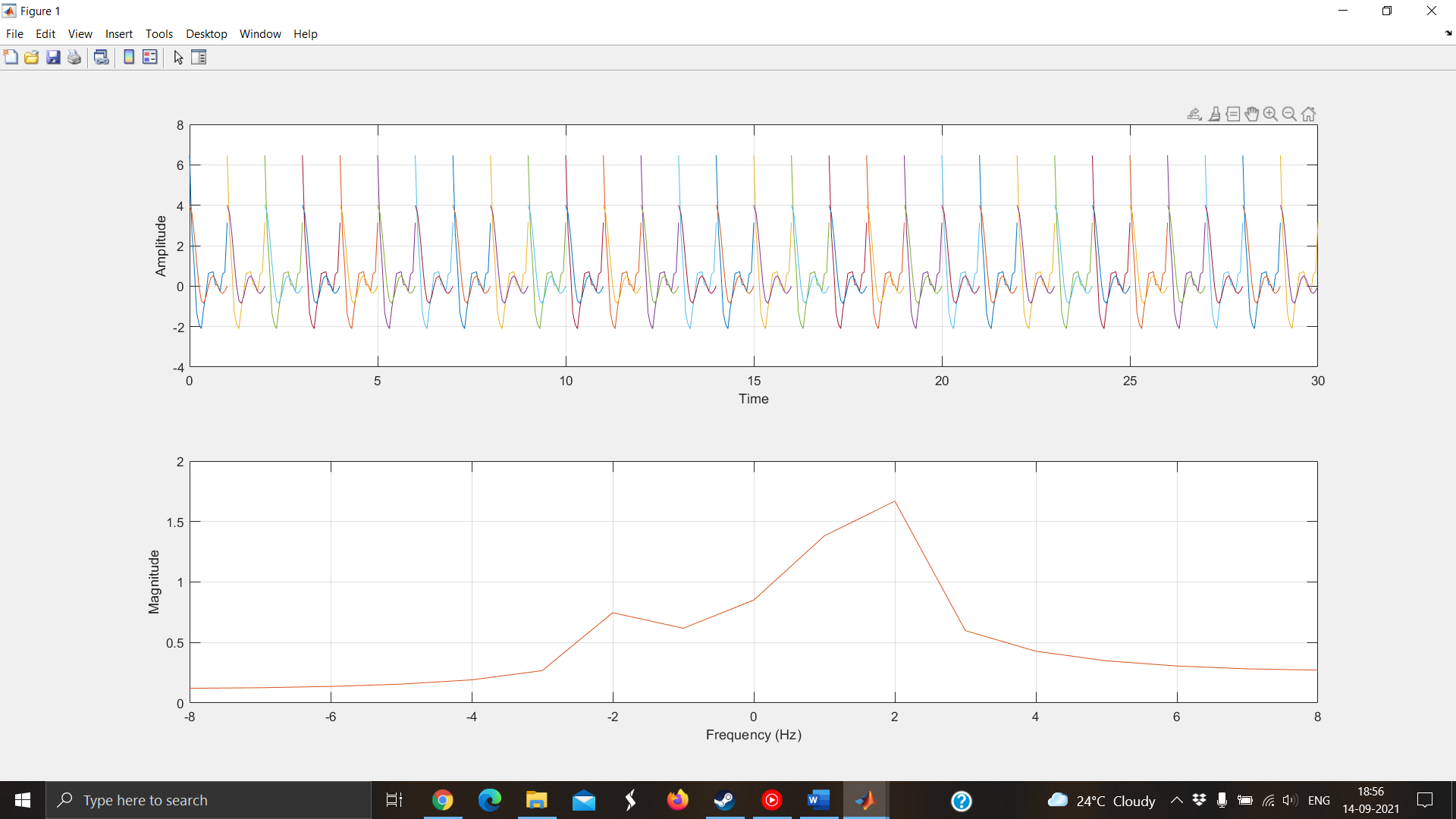
xlabel('frequency (Hz)')

ylabel('Magnitude')

grid on

pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

end



Conclusion: We are transmitting the same message each second with two parts which are the imaginary and real parts according to the Hilbert transform and in frequency domain it gets overlapped each second.