COMMUNICATION SYSTEMS LAB 4 DATE-14/9/2021

NAME – JAGRIT LODHA

ID – 2019A3PS0165P

SECTION – P4

MATLAB

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 = 5; % ID No - 2019A3PS0165P

fs=4\*freq;

ts=1/fs;

t=0:ts:1; % To define the time axis

U=randi(5); % To generate random values form 1 to 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 (seconds)')

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')

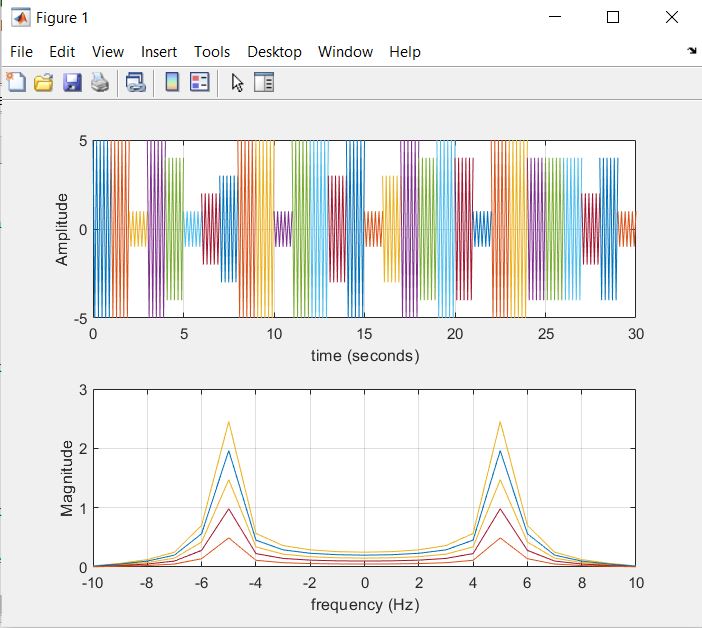
grid on

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.

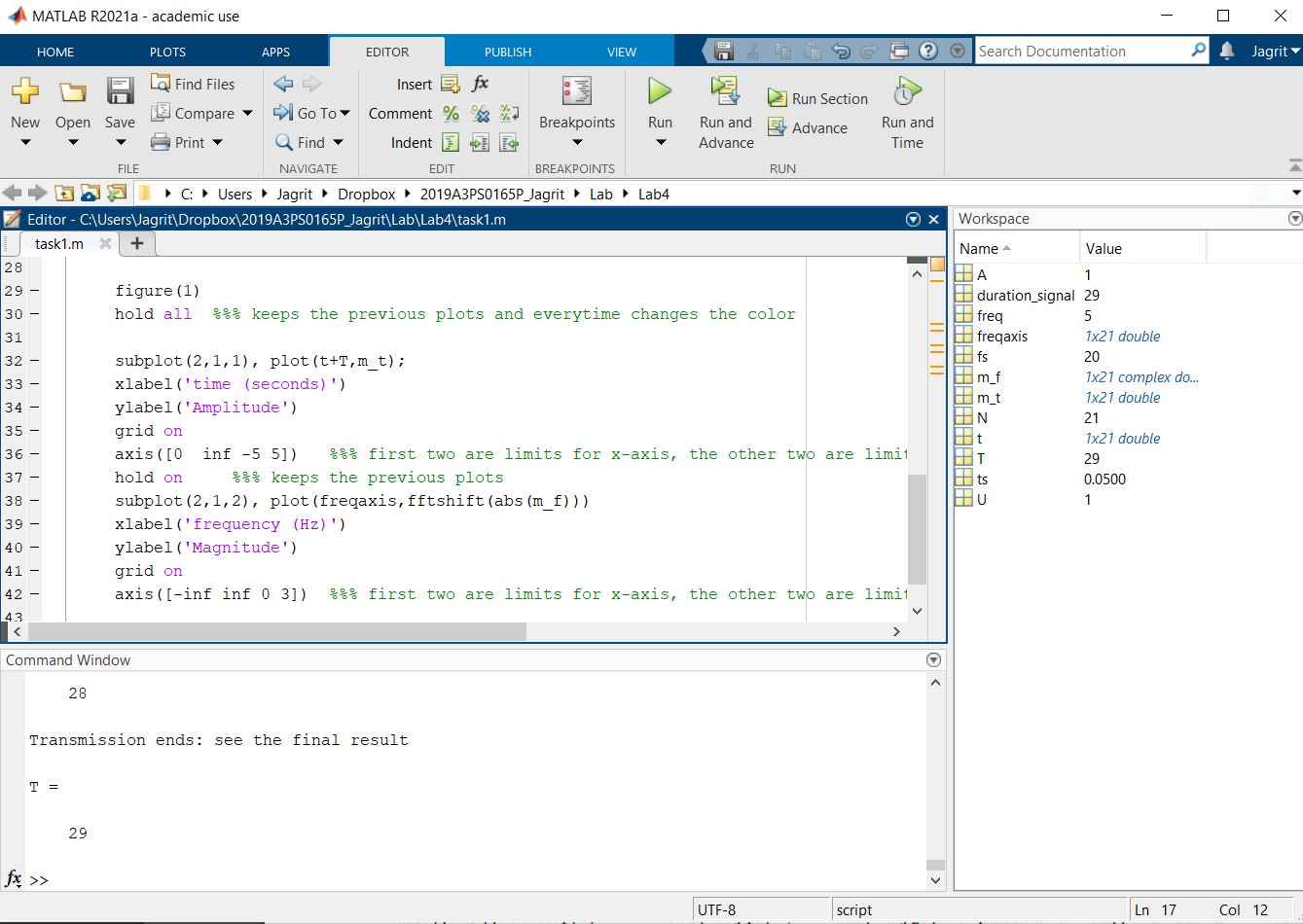
pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

End

Screenshot of MATLAB plot – (both time and frequency domain)



MATLAB Screenshot after executing -



NOTE –

* Last digit of ID no is 5.
* The cosine plot for each second might seem like zig-zag, it just appears so because of the small time interval of 1 second.
* We can see how random amplitudes of 1, 2, 3, 4 and 5 are being generated after each 1 sec.
* The frequency domain peaks are at 5 and -5. (because the cosine input is 2\*pi\*5\*t)
* Time domain plot –
  + X-axis (time) is from 0 to infinity as we need it from 0 to 30 seconds.
  + Y-axis (amplitude) is from 0 to 5 because the amplitude (U) takes random values from 1 to 5.
* Frequency-domain plot –
  + X-axis (frequency) is from -infinity to infinity because N (the frequency peak) could have taken any value. [N=0 to 9 in our case]
  + Y-axis (magnitude) is from 0 to 3 as the magnitudes of Fourier Transforms can be 0.5, 1, 1.5, 2 and 2.5, so an upper limit of 3 should be fine.

TASK 2 –

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=5; % ID No - 2019A3PS0165P

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("audio.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(m\_t);

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

xlim([0 30]);

xlabel('time')

ylabel('amplitude')

grid on

axis([0 inf -3 10])

hold on %%% keeps the previous plots

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

xlabel('frequency (Hz)')

ylabel('Magnitude')

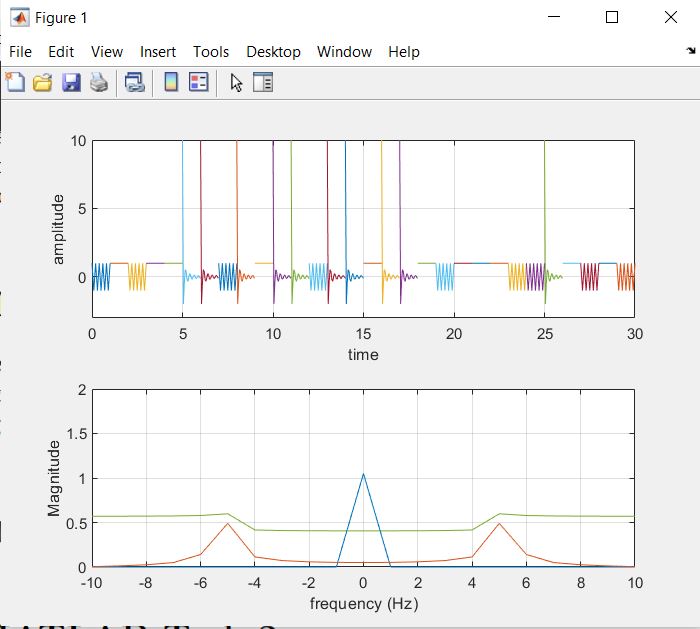
grid on

axis([-inf inf 0 2])

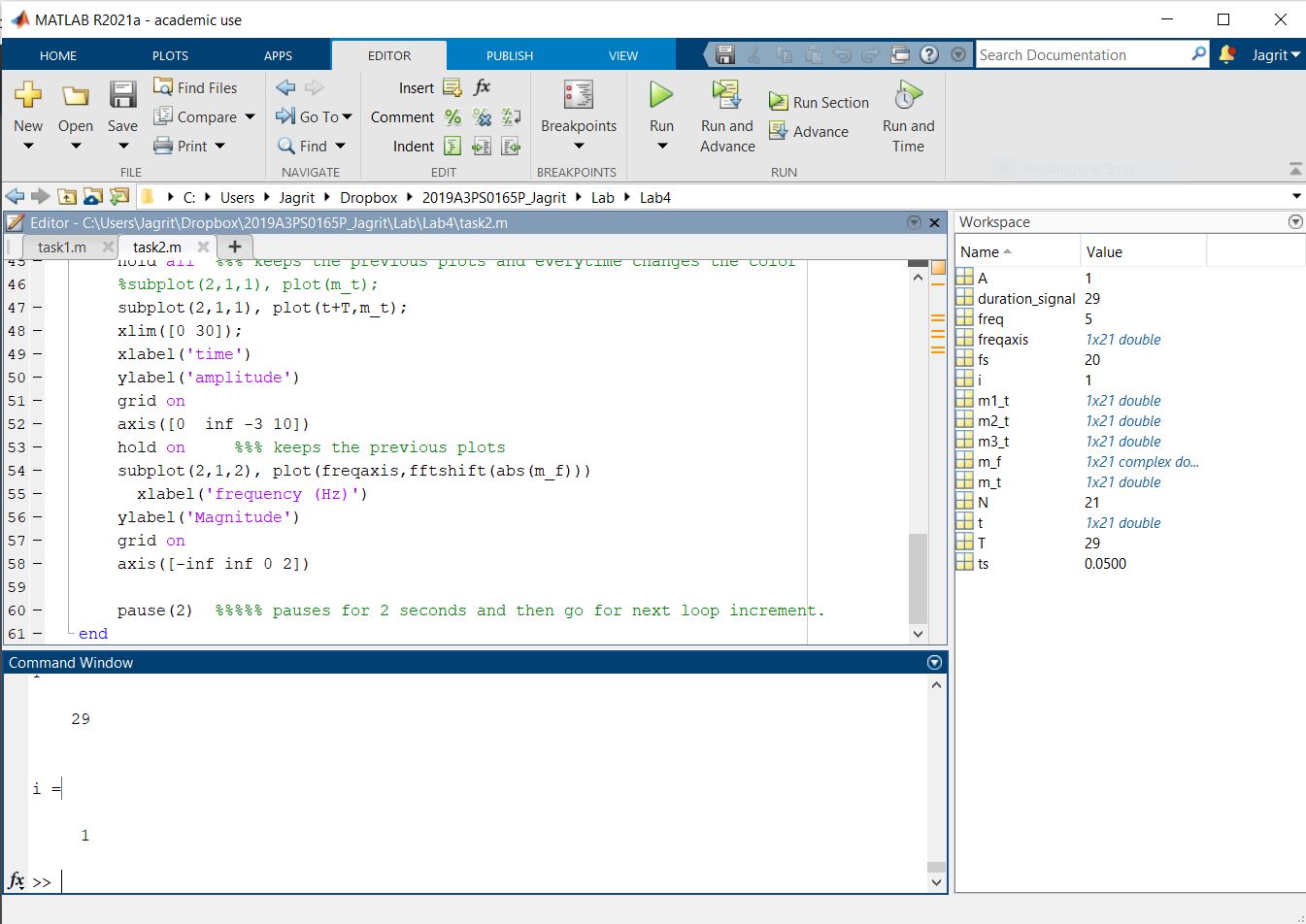
pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

end

Screenshot for Task 2 (without using audio file)



MATLAB Terminal Screenshot -



NOTE –

* We can clearly see how we are getting three different signals are being chosen and 3 different frequency plots as well.
* The x and y limits have been chosen similar to that of task 1.
* We get two frequency peaks at 5 and -5 because of the cosine signal and a rect spectrum for the sinc signal
* For the rect signal, we get only an impulse at f=0 because for rect(t/N) (N=5), the signal is 1 for every 1 second interval and hence it is being treated as a flatband signal, while in reality it is a rectangular pulse of width from -2.5 to 2.5.

TASK 3 –

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=50; % Just taken for sampling purposes

fs=2\*freq;

ts=1/fs;

t=0:ts:1;

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

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

N=5;

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

h\_t = 100\*sinc(100\*t); %%% To band-limit the signal to 50Hz

m\_t = conv(m1\_t, h\_t, '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 -inf inf])

hold on %%% keeps the previous plots

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

xlabel('frequency (Hz)')

ylabel('Magnitude')

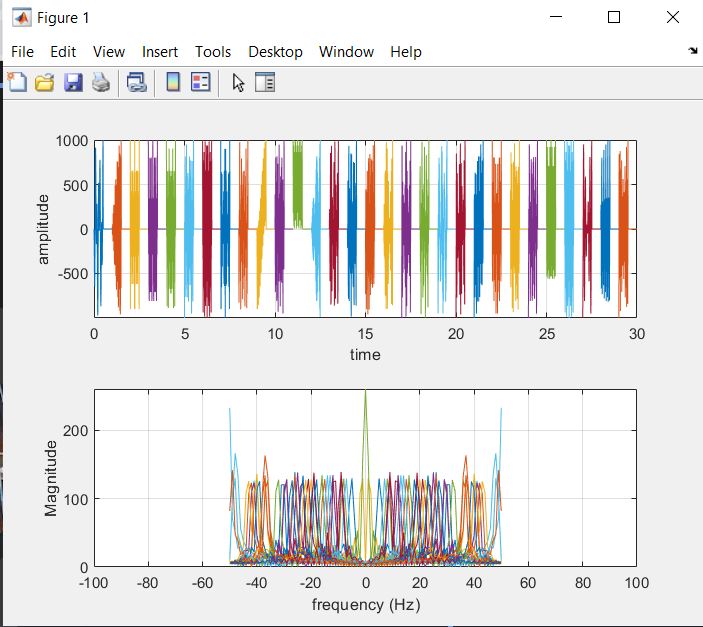
grid on

axis([-100 100 0 inf])

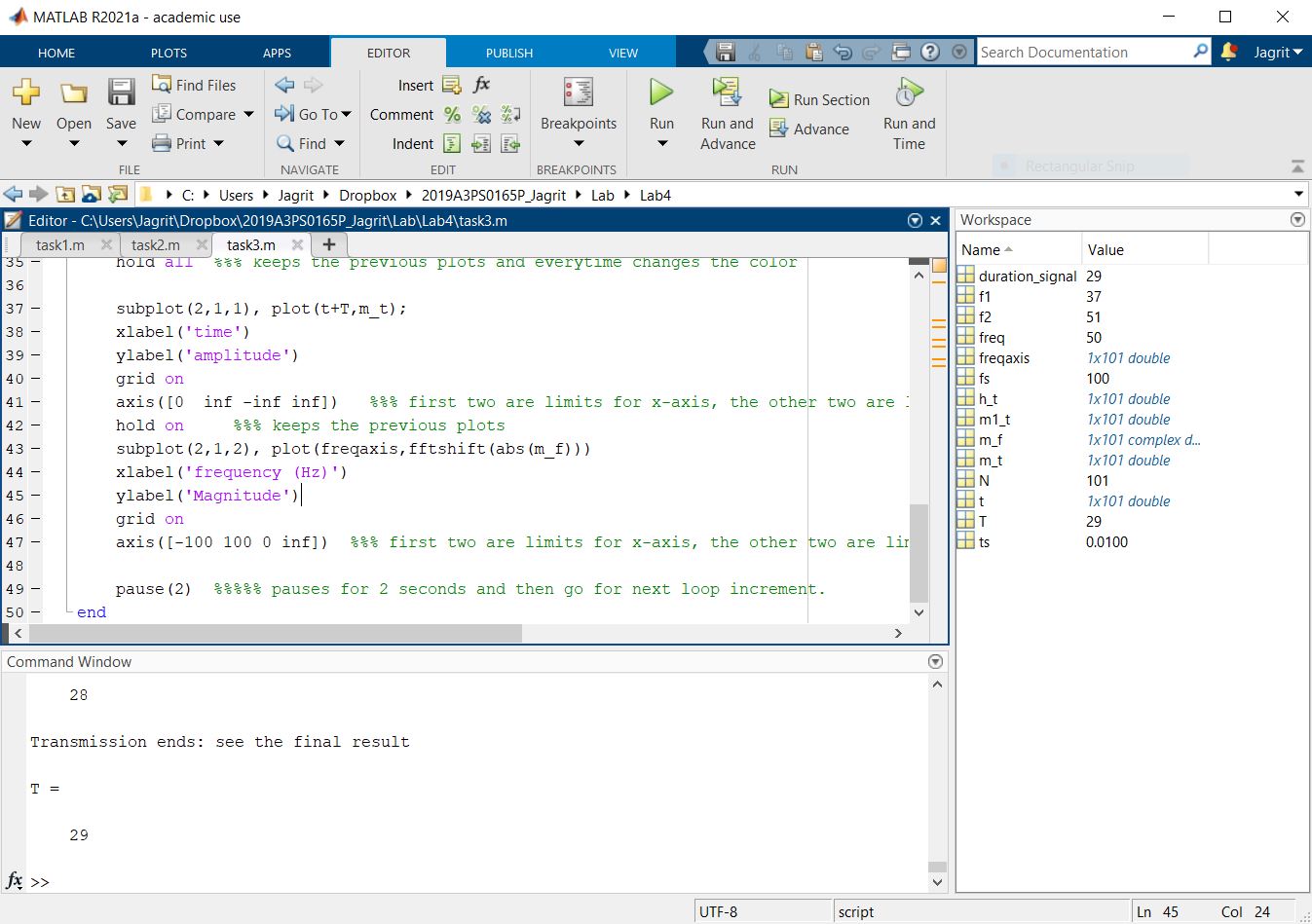
pause(2) %%%%% pauses for 2 seconds and then go for next loop increment.

End

Screenshot for plots –



Screenshot for MATLAB Terminal –



NOTE –

* We use the sinc function in time domain to limit the signal to 50Hz [ h(t) = 100\*sinc(100t) ]
* We can see in time-domain how in each second, a different f1 and f2 are being selected.
* In frequency domain, due to random values for f1 and f2, we get various peaks.
* We change the axis limits for the plots to account for the variety of amplitudes and frequency peaks.

TASK 4 –

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 = 5; % ID No - 2019A3PS0165P

fs=4\*freq;

ts=1/fs;

t=0:ts:1; % To define the time axis

U=randi(5); % To generate random values form 1 to 5

m\_t = 2\*freq\*sinc(2\*freq\*t);

mh\_t = hilbert(m\_t);

x\_t = m\_t + 1j\*mh\_t;

plot(t,real(mh\_t),t,imag(mh\_t))

legend('real','imaginary')

title('hilbert Function')

N=length(x\_t);

x\_f= fft(x\_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,x\_t);

xlabel('time (seconds)')

ylabel('Amplitude')

grid on

axis([0 inf -inf inf])

hold on %%% keeps the previous plots

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

xlabel('frequency (Hz)')

ylabel('Magnitude')

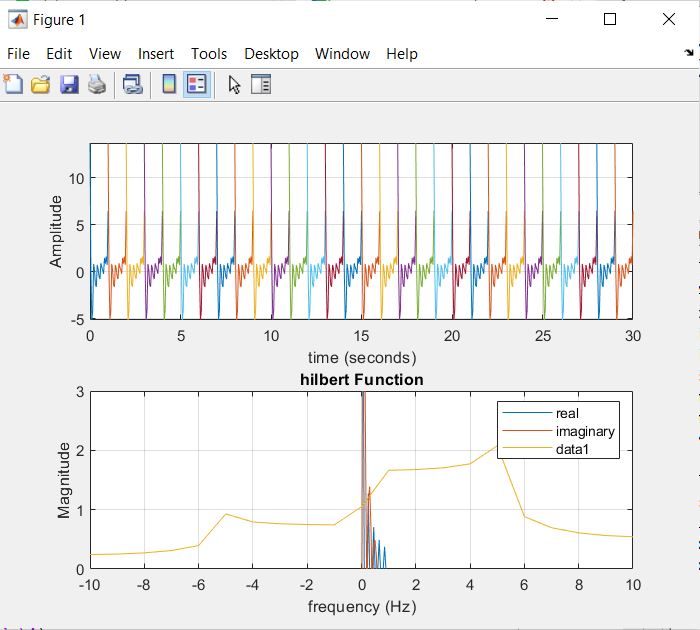
grid on

axis([-inf inf 0 3])

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

end

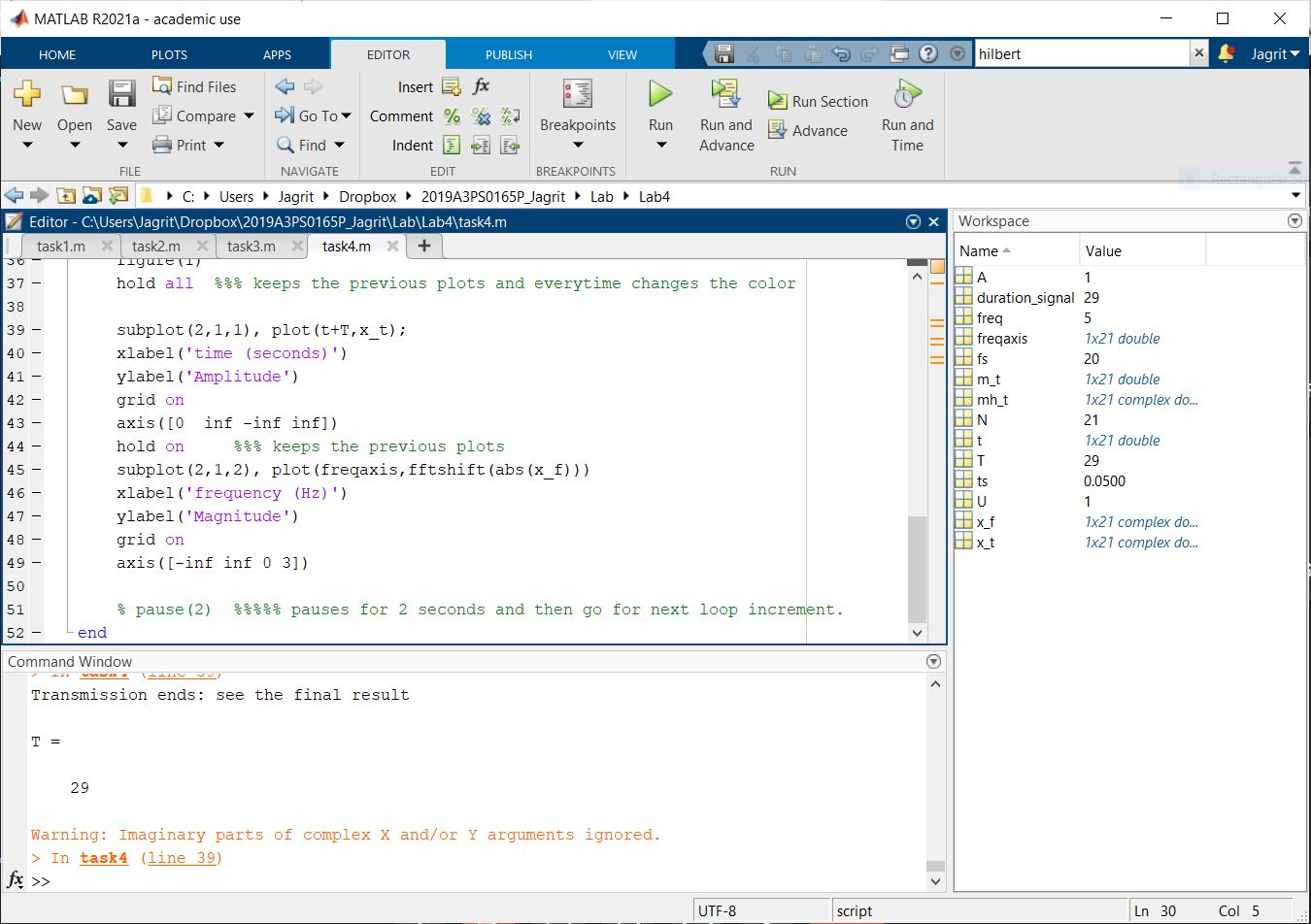
Screenshot of Plots –



NOTE –

We have plotted the real and imaginary part of Hilbert Transform of m(t) as given in the legend.

The yellow part shows the Frequency domain curve of x(t)



So we have successfully plotted x(t) for 30 seconds using the real-time approach.

We have also computed the Hilbert transform of m(t) and plotted it.