**6. COMPUTATION OF DFT & IDFT USING FFT**

clc;

close;

*//Computation of DFT by using FFT*

x=input('Enter the input sequence');

n=0:1:length(x)-1;

subplot(3,1,1);

plot2d3(n,x);

xlabel('Time n');

ylabel('Amplitude');

title('Input Sequence');

Xk = fft(x);

disp(Xk);

K1=0:1:length(Xk)-1;

subplot(3,1,2);

plot2d3(K1,Xk);

xlabel('freq k');

ylabel('X(K)');

title('DFT of x(n)')

*//computation of IDFT from FFT%*

y=fft(Xk,1);

disp(y);

subplot(3,1,3);

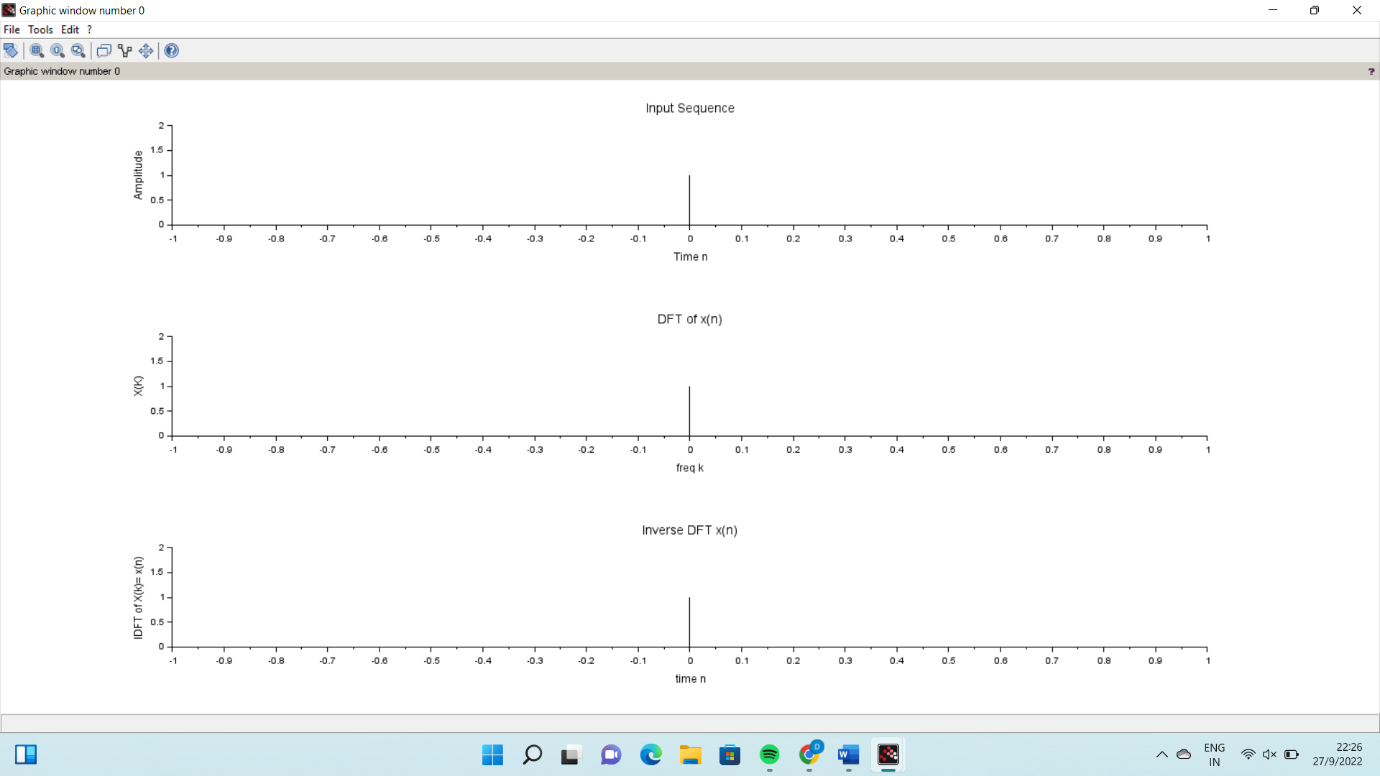
plot2d3(n,y);

xlabel('time n');

ylabel('IDFT of X(k)= x(n)');

title('Inverse DFT x(n)');

**OUTPUT**



**7. DESIGN OF IIR BUTTERWORTH FILTER**

*//Butterworth Lowpass Filter using bilinear transformation*

clc ;

close ;

wp=input('Enter the pass band frequency (Radians )= ' );

ws=input('Enter the stop band frequency (Radians )= ' );

alphap=input( ' Enter the pass band attenuation (dB)=' );

alphas=input( ' Enter the stop band attenuation(dB)=' );

T=input('Enter the Value of sampling Time=');

*//Pre warping- Bilinear Transformation*

omegap=(2/T)\*tan(wp/2);

disp('omegap=',omegap);

omegas=(2/T)\*tan(ws/2);

disp('omegas=',omegas);

*//Order of the filter*

N=log10(sqrt(((10^(0.1\*alphas))-1)/((10^(0.1\*alphap))-1)))/(log10(omegas/omegap));

disp('N=',N);

N=ceil(N);

disp('Round off value of N=',N);

*//Cut off frequency*

omegac=omegap/(((10^(0.1\*alphap)) -1)^(1/(2\* N)));

disp('omegac=',omegac);

disp('Normalised Analog LPF Transfer function H(S)=');

hs\_Normalised = analpf(N,'butt',[0,0],1);

disp('Analog LPF Transfer function H(S)=');

disp(hs\_Normalised);

hs= analpf(N,'butt',[0,0],omegac);

disp(hs);

z=poly(0,'z');*//Defining variable z*

Hz=horner(hs,(2/ T)\*((z -1)/(z+1)))*// Bilinear Transformation*

disp('Digital LPF Transfer function H(Z)=');

disp(Hz);

HW=frmag(Hz,512); *// Frequency response*

w=0:%pi/511:%pi ;

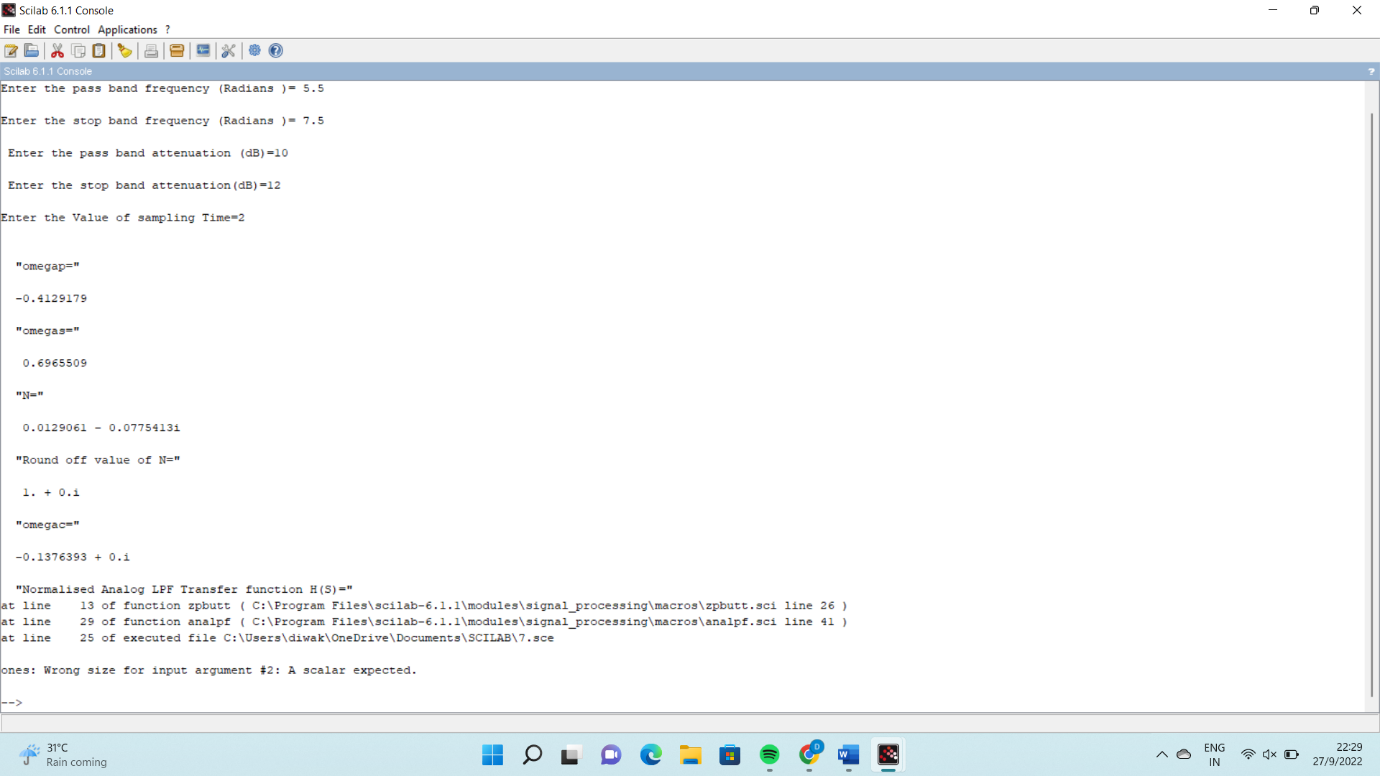
plot(w/%pi,abs(HW));

xlabel(' Normalized Digital Frequency w');

ylabel('Magnitude ');

title(' Frequency Response of Butterworth IIR LPF');

**OUTPUT**

****

**8. DESIGN OF IIR CHEBYSHEV FILTER**

clc ;

close ;

wp=input('Enter the pass band frequency (Radians )= ' );

ws=input('Enter the stop band frequency (Radians )= ' );

alphap=input( ' Enter the pass band attenuation (dB)=' );

alphas=input( ' Enter the stop band attenuation(dB)=' );

T=input('Enter the Value of sampling Time=');

*//Pre warping- Bilinear Transformation*

omegap=(2/T)\*tan(wp/2);

disp('omegap=',omegap);

omegas=(2/T)\*tan(ws/2);

disp('omegas=',omegas);

*//Order of the filter*

N=acosh(sqrt(((10^(0.1\*alphas))-1)/((10^(0.1\*alphap))-1)))/(acosh(omegas/omegap));

disp('N=',N);

N=ceil(N);

disp('Round off value of N=',N);

*//Cut off frequency*

omegac=omegap/(((10^(0.1\*alphap)) -1)^(1/(2\* N)));

disp('omegac=',omegac);

Epsilon = sqrt ((10^(0.1\*alphap))-1);

disp('Epsilon=',Epsilon);

[pols ,gn] = zpch1(N, Epsilon,omegap );

disp('Gain',gn);

disp('Poles',pols);

hs=poly(gn,'s','coeff')/real(poly(pols,'s'));

disp('Analog Low pass Chebyshev Filter Transfer function',hs);

z=poly(0,'z');*//Defining variable z*

Hz=horner(hs,(2/ T)\*((z -1)/(z+1)))*// Bilinear Transformation*

disp('Digital LPF Transfer function H(Z)=',Hz);

HW=frmag(Hz,512); *// Frequency response*

w=0:%pi/511:%pi ;

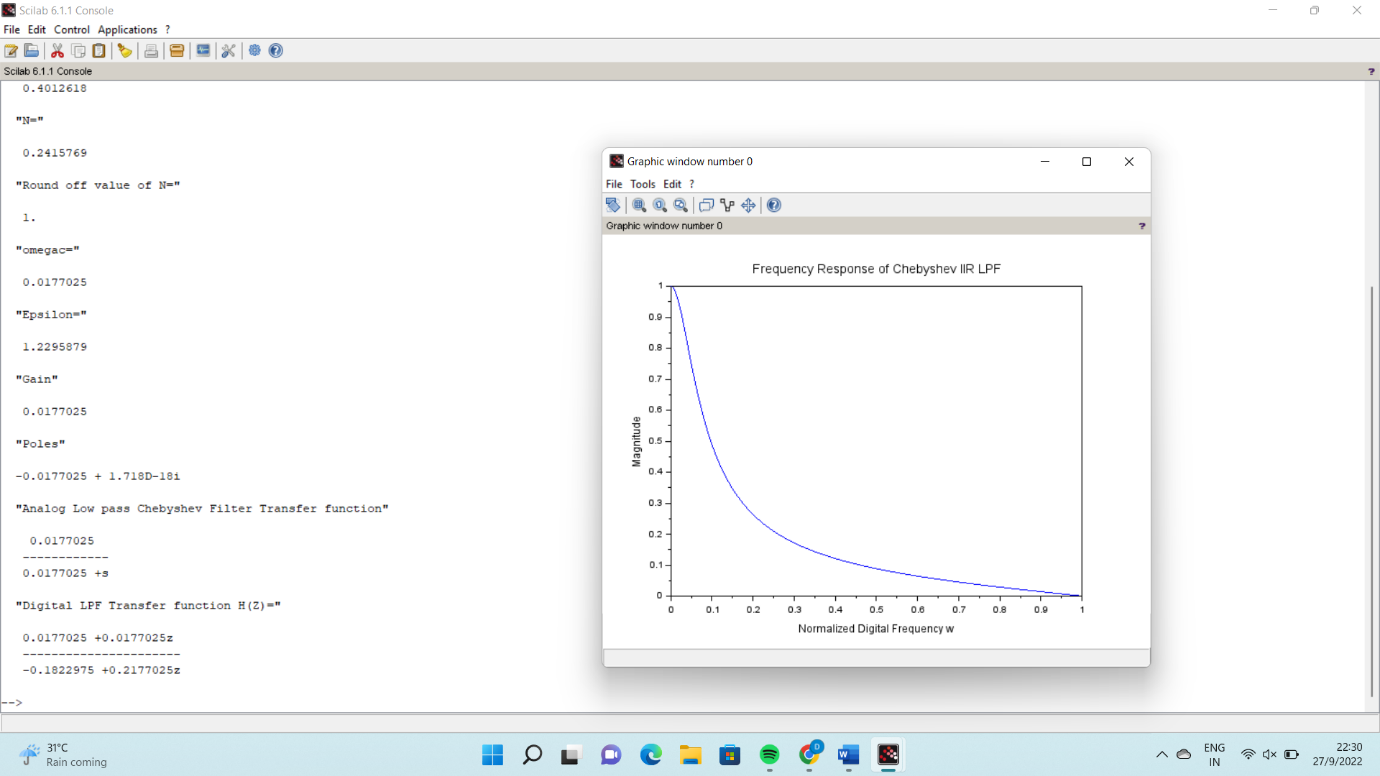
plot(w/%pi,abs(HW));

xlabel(' Normalized Digital Frequency w');

ylabel('Magnitude ');

title(' Frequency Response of Chebyshev IIR LPF');

**OUTPUT**

****