**11. DESIGN OF FIR FILTER USING FREQUENCY SAMPLING METHOD**

// Design a FIR filter using Frequency Sampling method.

*//frequency sampling method*

clc;

clear;

close;

N=input("Enter the value of N:");

U=input("Enter the value of U:");

for n=0+U:1:N-1+U

h(n)=(1+cos(2\*%pi\*(7-n)/N))/N;

end

[hz,f]=frmag(h,256);

hz\_dB=20\*log10(hz)./max(hz);

figure;

plot(2\*f,hz\_dB);

a=gca();

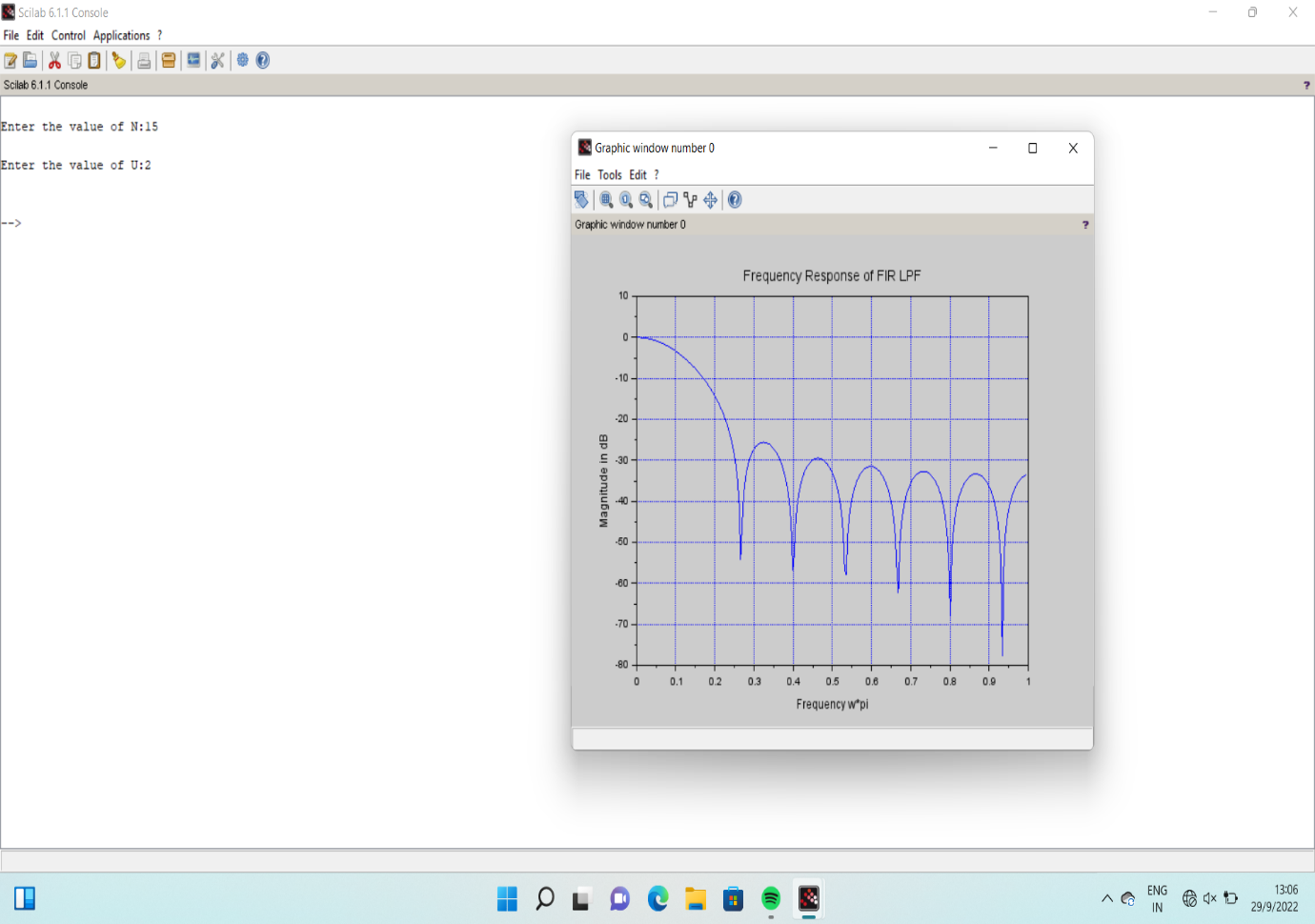
xlabel('Frequency w\*pi');

ylabel('Magnitude in dB');

title('Frequency Response of FIR LPF');

xgrid(2);

**OUTPUT**

****

**12 DOWN SAMPLING**

clc;

close;

N=input('input length of the input signal sequnce');

M=input('downsampling factor=');

*//generate the input sequence for the specified length N*

n=0:N-1;

*//m=0:N\*M-1;*

x=n;

disp(x);

*//generate the upsampled signal*

y=x([1:M:length(x)]);

disp(y);

*//plot the input sequence*

subplot(2,1,1);

plot2d3(n,x(1:N));

title('input sequence');

xlabel('time n');

ylabel('amplitude');

subplot(2,1,2);

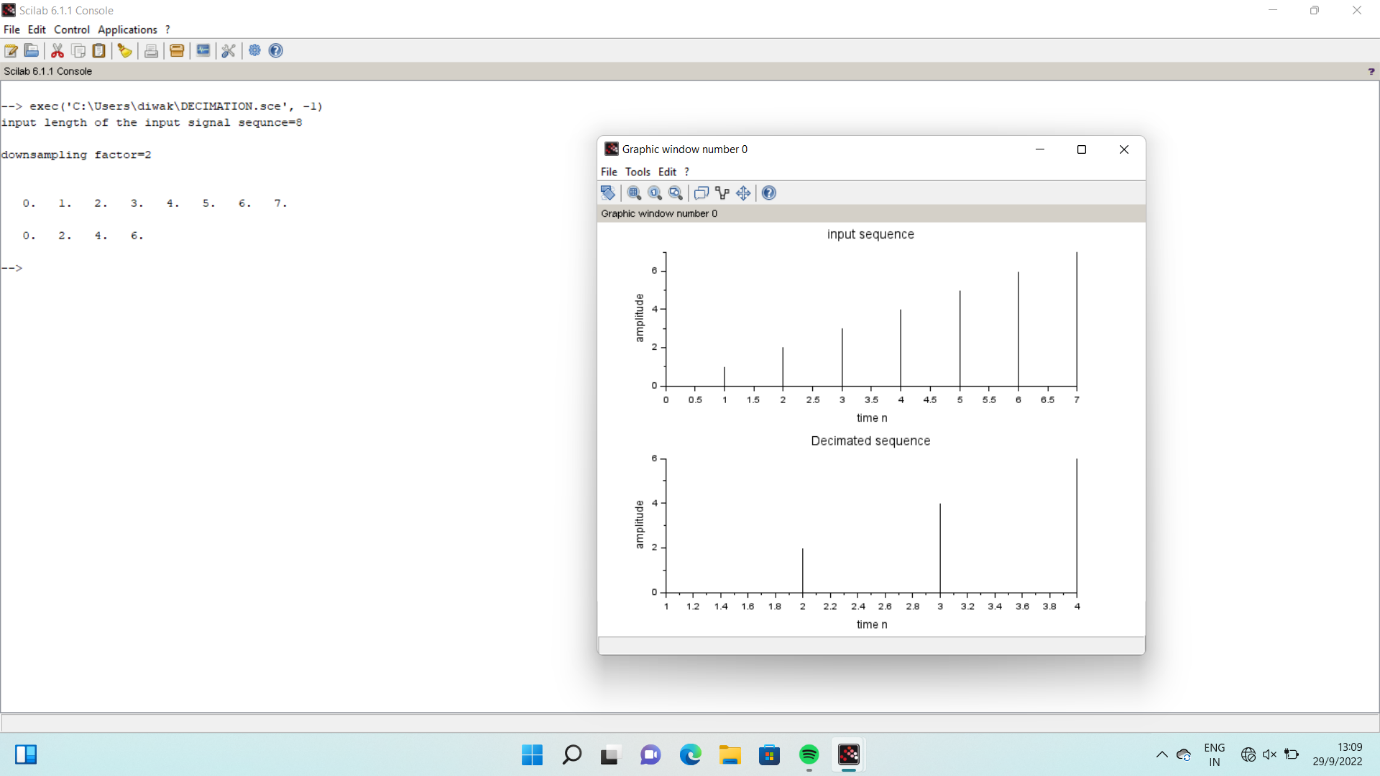
plot2d3(y);

title('Decimated sequence');

xlabel('time n');

ylabel('amplitude');

**OUTPUT**

****

**13 UP SAMPLING**

*//Multirate Signal Processing in scilab*

*//Upsampling a signal by a factor of 2*

clear;

clc;

n = 0:%pi/150:2\*%pi;

x = sin(%pi\*n); *//original signal*

upsampling\_x = zeros(1,2\*length(x)); *//upsampled by a factor of 2*

upsampling\_x(1:2:2\*length(x)) = x;

subplot(2,1,1)

plot(1:length(x),x);

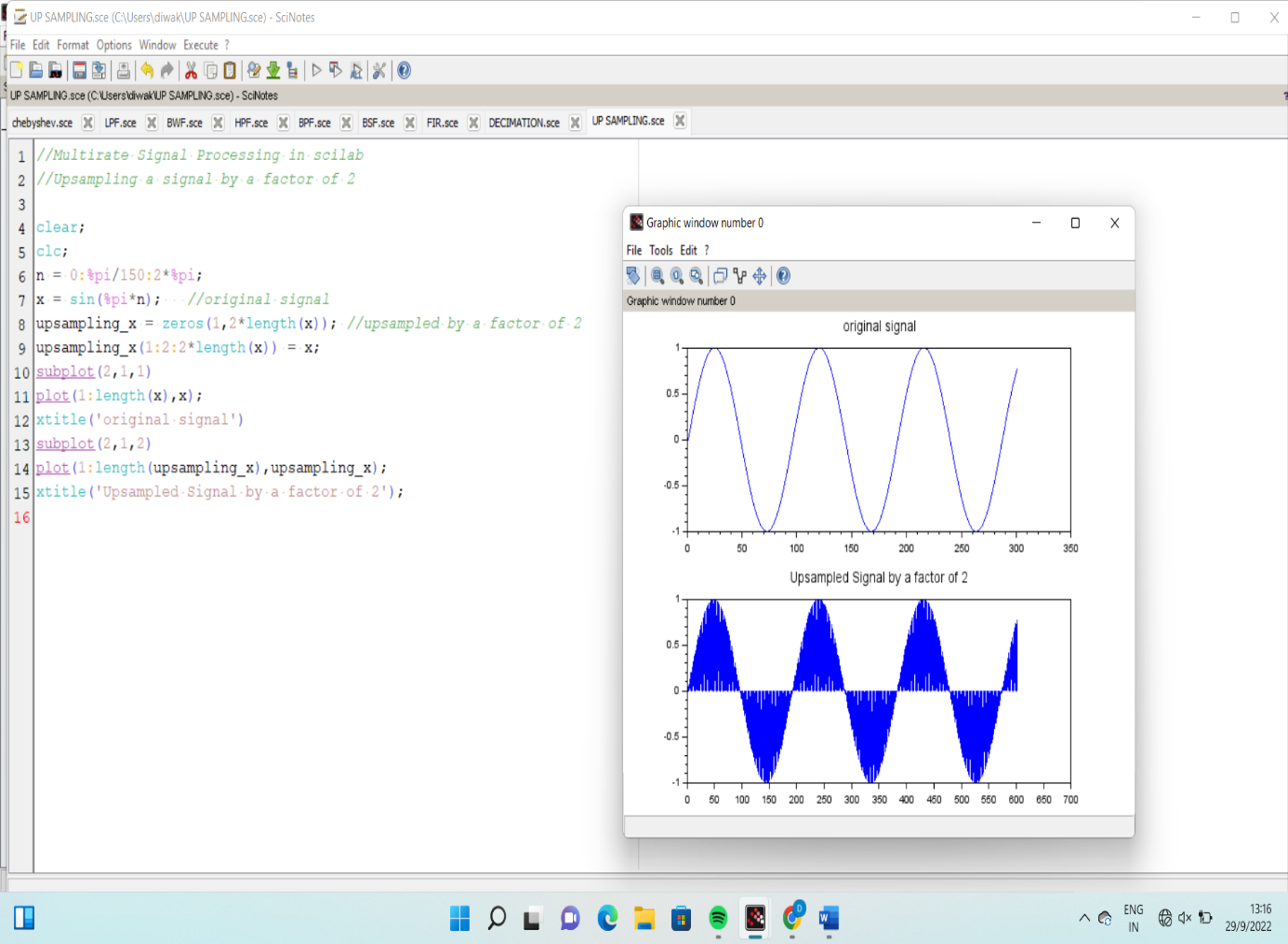
xtitle('original signal')

subplot(2,1,2)

plot(1:length(upsampling\_x),upsampling\_x);

xtitle('Upsampled Signal by a factor of 2');

**OUTPUT**

****

**14 DEAD BAND INTERVAL**

clc; clear; n=-1; y=12;

flag=1; while n<8

n=n+1;

y=[y 0.9\*y(n+1)];

yr=round(y);

end;

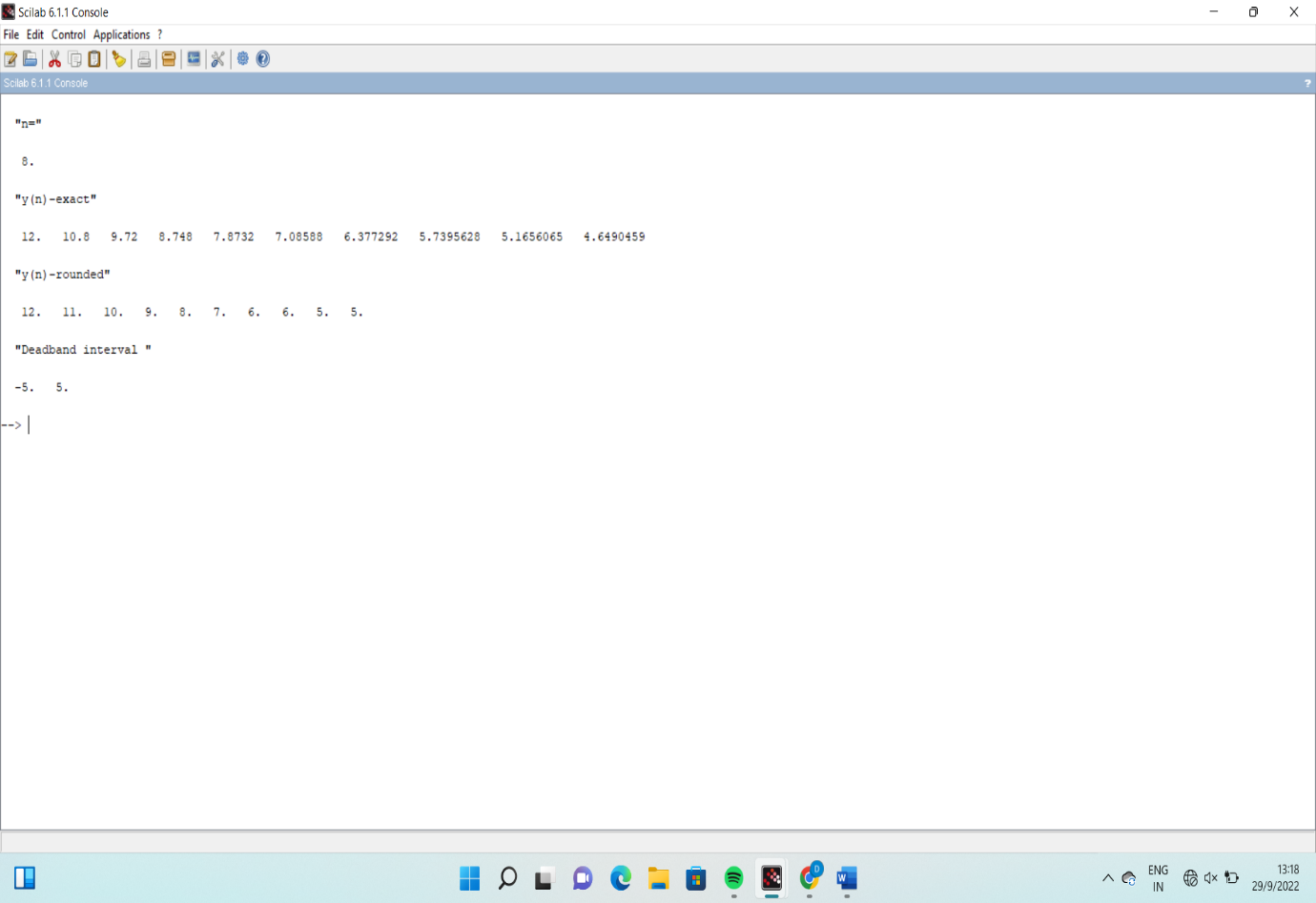
disp('n=',n);

disp('y(n)-exact',y);

disp('y(n)-rounded',yr);

disp('Deadband interval ',[-yr(n+2) yr(n+2)])

**OUTPUT**

****