**ME (Embedded Systems)**

**DIGITAL SIGNAL PROCESSING**

**Lab Assignment 2**

**Name : SRILATHA P Date : 10-04-2023**

**Reg. No. : 221039001**

**1. Write a Matlab program to Design an ideal linear phase bandpass FIR filter with**

**cutoff frequencies pi/6 rads and pi/3 rads, using frequency sampling technique. Assume**

**25 tap coefficients.**

**Code :**

clc;

clear all;

wc1=pi/6;

wc2=pi/3;

N=25;

%linear phase factor

alpha=(N-1)/2;

%dft samples indexed with k

hk=[zeros(1,N)];

k=0:1:N-1;

for i=1:N

w=2\*pi\*k(i)/N;

if(w>=wc1 && w<=wc2)

hk(i)=exp(-1i\*w\*alpha);

end

end

%magnitude and phase of DFT samples

Hmag=abs(hk);

Hphase=phase(hk);

%comb filter

num=[1,zeros(1,N-1),-1];

den=N;

Hc=tf(num,den,0.01,'Variable','z^-1'); %assuming sampling time for input as 0.01s

disp("Comb filter transfer function:");

Hc

%Resonator

%Since N=25 there is no N/2 term

% H(0)

num=[Hmag(1)];

den=[1,-1];

Hr=tf(num,den,0.01,'Variable','z^-1');

% find Hk(z) for values of k=1 to N-1/2 and add to Hr

for i=1:(N-1)/2

if(Hmag(i+1)~=0)

num=[cos(Hphase(i+1)),cos(Hphase(i+1)-2\*pi\*i/N)];

den=[1,-2\*cos(2\*pi\*i/N),1];

Hkz=tf(num,den,0.01,'Variable','z^-1');

Hr=Hr+2\*Hmag(i+1)\*Hkz;

end

end

disp("Resonator Tranfer function");

Hr

%Multiply tranfer function of comb filter and resonator

Hz=Hc\*Hr;

disp("Final transfer function:");

Hz

[num,den]=tfdata(Hz,'v');

w=0:.001\*pi:pi;

Hw=freqz(num,den,w);

Domega=w/pi;

plot(Domega,abs(Hw));

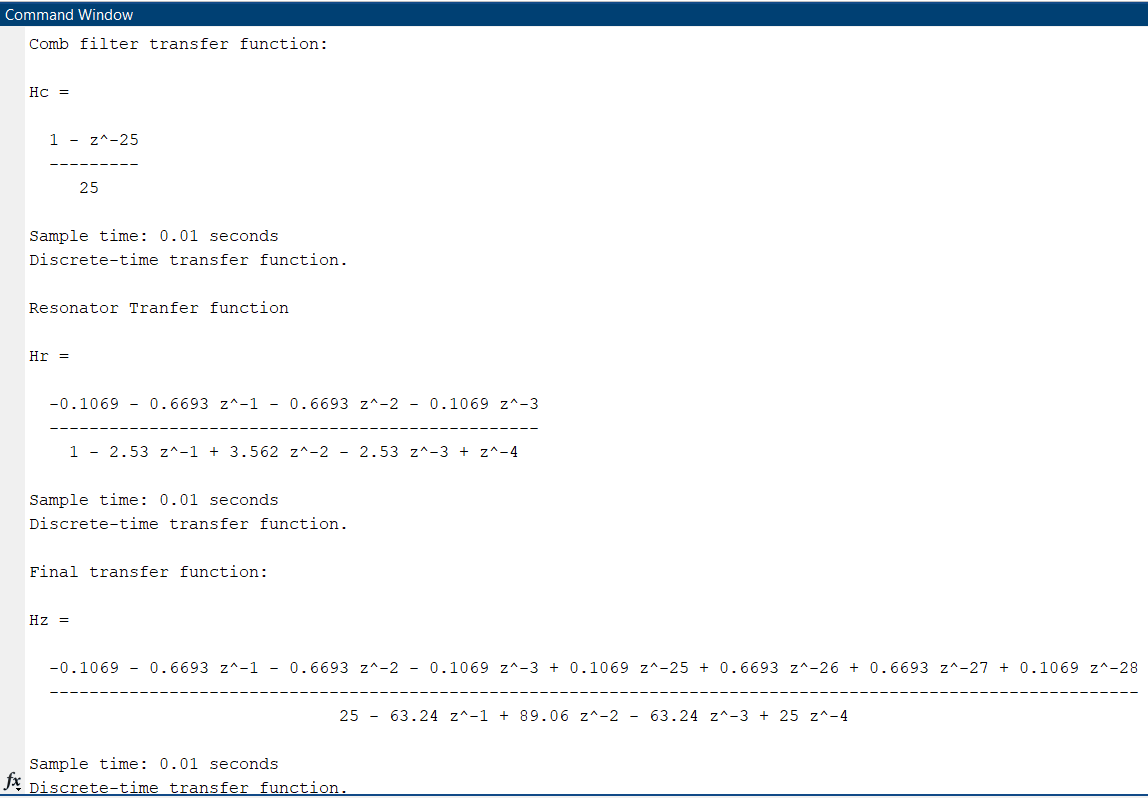
title('Frequency Response of the Filter')

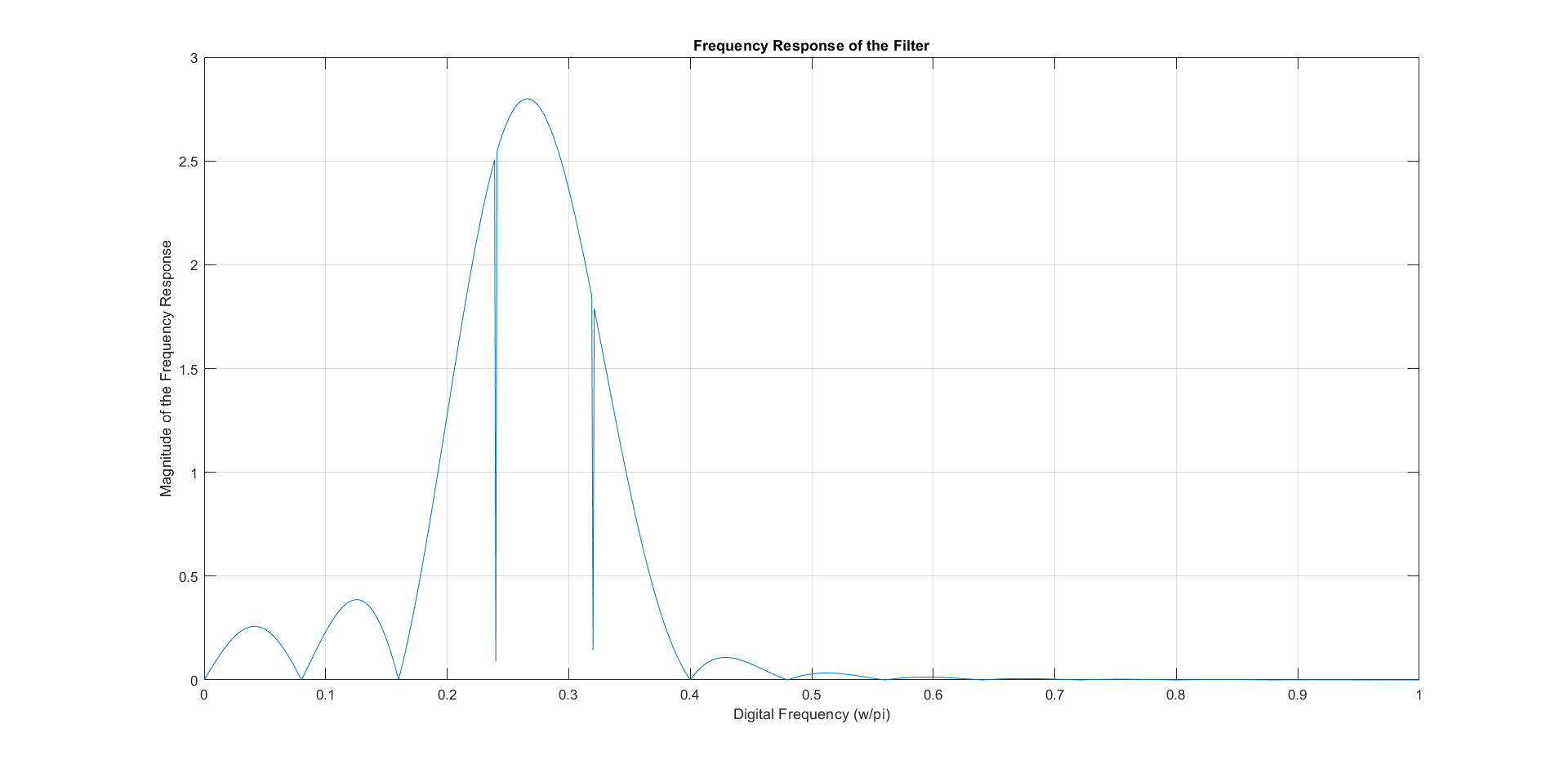
xlabel('Digital Frequency (w/pi)')

ylabel('Magnitude of the Frequency Response')

grid on

**Output:**





**2. Write a Matlab program to Design a digital Butterworth filter using Impulse Invariance transformation to meet the following specification.**

**(Do not use built-in functions to design the analog filter)**

0 ≥ ⎪H(ej Ω)⎪dB ≥ -1; for 0 ≤ Ω ≤ 20 rad/sec.

⎪H(ej Ω)⎪dB < -60; for Ω ≥ 200 rad/sec.

**Assume a sampling period be = 0.01 sec.**

**Code:**

clc;

clear all;

ap=-1;

as=-60;

wp=20;

ws=200;

T=0.01;

Fs=1/T;

%calculate N

N=ceil(log10((10^(-ap/10)-1)/(10^(-as/10)-1))/(2\*log10(wp/ws)));

disp("N:");

disp(N);

%calculate cutoff frequency

wc=wp/((10^(-ap/10)-1)^(1/(2\*N)));

disp("Cutoff Freq:");

disp(wc);

% transfer function by finding poles

sk=[]

hs=1;

for k=0:N-1

sk(k+1)=wc\*exp(-1i\*(2\*k+1+N)\*pi/(2\*N));

tf(abs(sk(k+1)),[1,-sk(k+1)])

hs=hs\*tf(abs(sk(k+1)),[1,-sk(k+1)]);

end

disp("poles:");

disp(sk);

hs

%frequency response of analog filter

[num,den]=tfdata(hs,'v');

fre1=0:1:300;

[resps]=freqs(num,den,fre1);

mags=20\*log10(abs(resps));

plot(fre1,mags);

title('Analog Butterworth Filter Frequency Response')

xlabel('Frequency in rad/s');

ylabel('Magnitude in dB.');

grid;

zoom;

%frequency response of digital filter

figure;

[BZ,AZ]=impinvar(num,den,Fs);

fre2=0:0.01:3;

[respz]=freqz(BZ,AZ,fre2);

magz=20\*log10(abs(respz));

f2=fre2\*Fs;

plot(f2,magz);

title('Digital Butterworth Filter Frequency Response')

xlabel('Frequency in rad/s ');

ylabel('Magnitude in dB.');

grid on;

**Output:**

