Class works:

Task 1: To perform IIR system using MATLAB

Code:

%IIR system

clc;

clear all;

close all;

Nx=51;

b=[0.5,0.7,0.6,0.4];

a=[1,0.4,-0.3,0.2];

n=(0:Nx-1);

x=sin(2\*pi\*0.125\*n);

y=filter(b,a,x);

figure(1)

stem(n,x),xlabel('Sample index k'),

ylabel('X'),

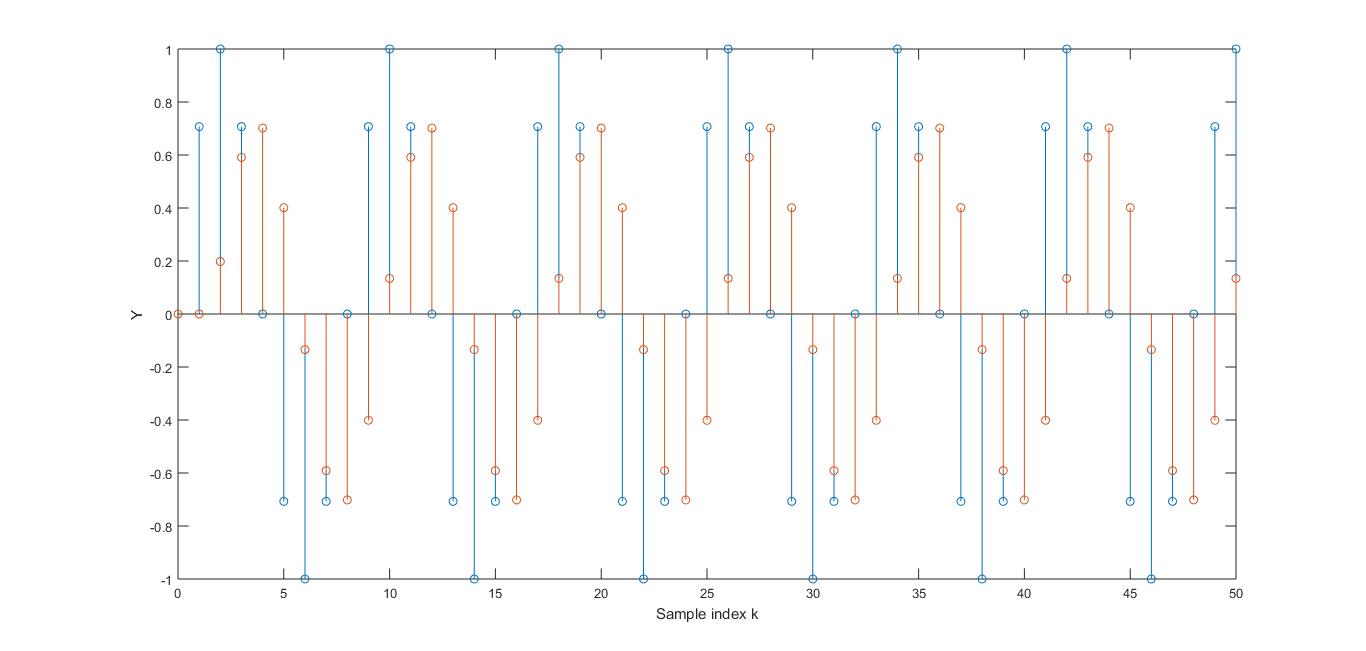
hold on

stem(n,y),xlabel('Sample index k'),

ylabel('Y'),

Output:

Figure 4.1 : to perform IIR system using MATLAB



Task 2: To perform FIR system using MATLAB

Code:

%FIR

Nx=51;

b=[0.5,0.7,0.6,0.4];

a=1;

n=(0:Nx-1);

x=sin(2\*pi\*0.125\*n);

y=filter(b,a,x);

figure(1)

stem(n,x),xlabel('Sample index k'),

ylabel('X'),

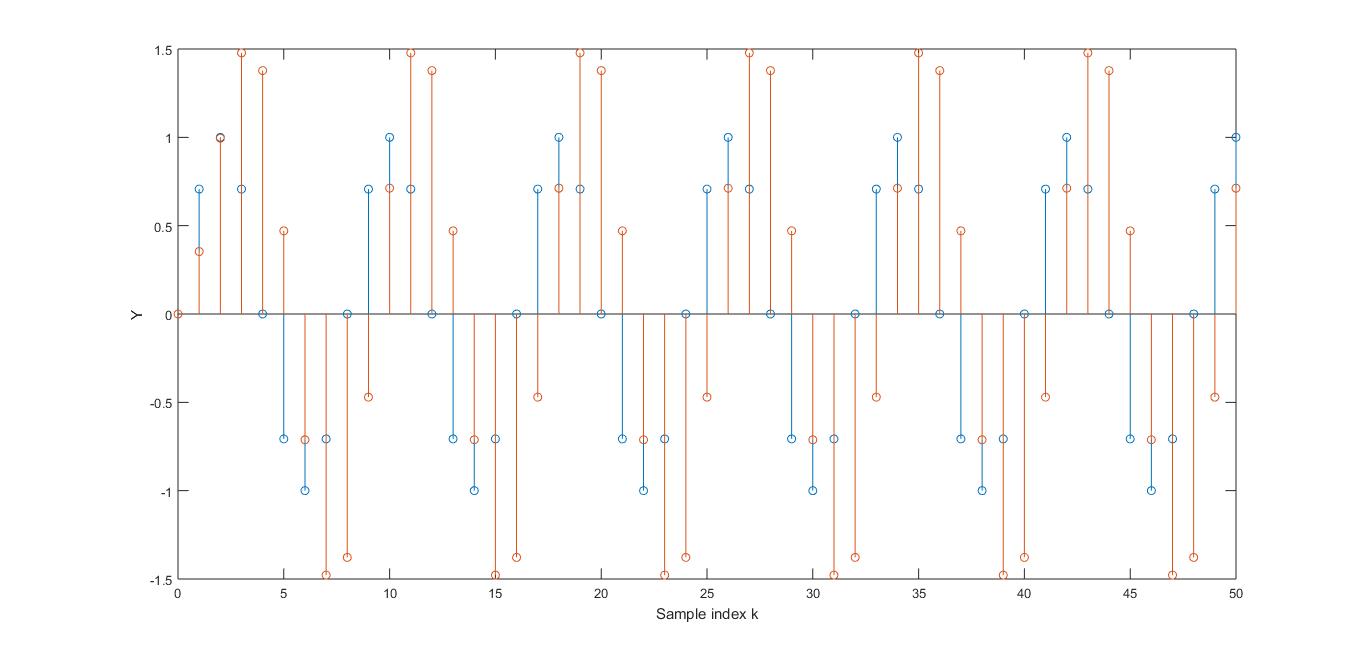
hold on

stem(n,y),xlabel('Sample index k'),

ylabel('Y'),

Output:

Figure 4.2 : to perform FIR system using MATLAB



Task 3: To perform FIR cascade form for the LTI system of transfer function

Code:

format long

num=input('Numerator coefficient vector=');

FIR\_cascade= roots([num])

Output:

Numerator coefficient vector=[1 2 0 3 0 -5]

FIR\_cascade =

-2.374751129250894 + 0.000000000000000i

-1.105257528190224 + 0.000000000000000i

0.268350131758975 + 1.395509126085460i

0.268350131758975 - 1.395509126085460i

0.943308393923172 + 0.000000000000000i

Task 4: factorization of a cascade realization for IIR Transfer Function

Code:

format long

num=input('Numerator coefficient vector=');

den=input('Denominator coefficient vector=');

[z,p,k]=tf2zp(num,den)

sos=zp2sos(z,p,k)

Output:

Numerator coefficient vector=[0 .5 .77 0 .8]

Denominator coefficient vector=[1 2 0 3 -5]

z =

-1.957540408099285 + 0.000000000000000i

0.208770204049642 + 0.879640394007509i

0.208770204049642 - 0.879640394007509i

p =

-2.678405406629794 + 0.000000000000000i

-0.119284504658155 + 1.421821927645391i

-0.119284504658155 - 1.421821927645391i

0.916974415946106 + 0.000000000000000i

k =0.500000000000000

sos =Columns 1 through 2

0 0.500000000000000

1.000000000000000 -0.417540408099284

Columns 3 through 4

0.978770204049642 1.000000000000000

0.817352220868614 1.00000000000000

Columns 5 through 6

1.761430990683688 -2.456029233411247

0.238569009316309 2.035806386984795

Task 5: parallel realization of an iir transfer Function

Code:

num=input('Numerator coefficient vector=');

den=input('Denominator coefficient vector=');

[r1,p1,k1]=residuez(num,den);

[r2,p2,k2]=residue(num,den);

disp('Parallel form I')

disp('Residues are');disp(r1);

disp('poles are at');disp(p1);

disp('Constant value');disp(k1);

disp('Parallel form II')

disp('Residues are');disp(r2);

disp('poles are at');disp(p2);

disp('Constant value');disp(k2);

Output:

Numerator coefficient vector=[2 3 .7 8]

Denominator coefficient vector=[4 9 .2 3]

Parallel form I

Residues are

0.056496083611412 + 0.000000000000000i

-1.111581375139039 + 0.103230326430492i

-1.111581375139039 - 0.103230326430492i

disp('poles are at');disp(p1);

poles are at

-2.363143325430491 + 0.000000000000000i

0.056571662715245 + 0.560511856507696i

0.056571662715245 - 0.560511856507696i

Constant value

2.666666666666667

Parallel form II

Residues are

-0.133508342899272 + 0.000000000000000i

-0.120745828550364 - 0.617214629029749i

-0.120745828550364 + 0.617214629029749i

poles are at

-2.363143325430491 + 0.000000000000000i

0.056571662715245 + 0.560511856507696i

0.056571662715245 - 0.560511856507696i

Constant value

0.500000000000000

Task 6: Spectral analysis of signal

Code:

clc;

clear all;

close all;

f1=30;%signal freq

fs=128; %sampling freq

N=256;%no. of samples

N1=1024;%no. of FFT points

n=0:N-1;% index n

f=(0:N1-1)\*fs/N1; %defining the frequency points [axis]

x=cos(2\*pi\*f1\*n/fs);

XR=abs(fft(x,N1));%magnitude of FFT using no windowing

xh=hamming(N);%hamming samples

xw=x.\*xh';%window the signal

XH=abs(fft(xw,N1));%magnitude of windowed signal

subplot(2,1,1);

plot(f(1:N1/2),20\*log10(XR(1:N1/2)/max(XR)));

title('spectrum of x(t) using rectangular windows');

grid;

xlabel('Frequency,Hz');

ylabel('Normalized Magnitude,[dB]');

subplot(2,1,2);

plot(f(1:N1/2),20\*log10(XH(1:N1/2)/max(XH)));

title('spectrum of x(t) using rectangular windows');

grid;

xlabel('Frequency,Hz');

ylabel('Normalized Magnitude,[dB]');

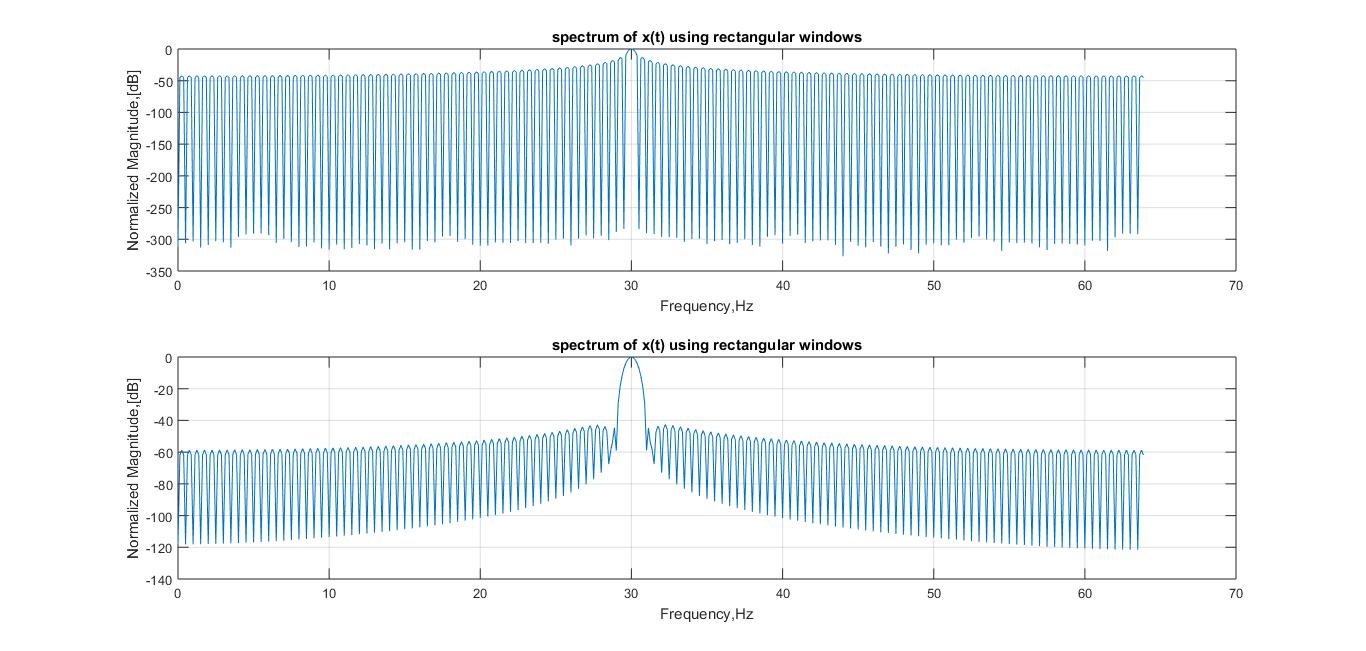


Figure 4.3 : Spectrum analysis using rectangular window

Home works:

Task 1: To perform IIR system using MATLAB

Code:

Nx=51;

b=[0,.28,.44,.09];

a=[.6,.5,.19,-.5];

n=(0:Nx-1);

x=sin(2\*pi\*0.125\*n);

y=filter(b,a,x);

figure(1)

stem(n,x),xlabel('Sample index k'),

ylabel('X'),

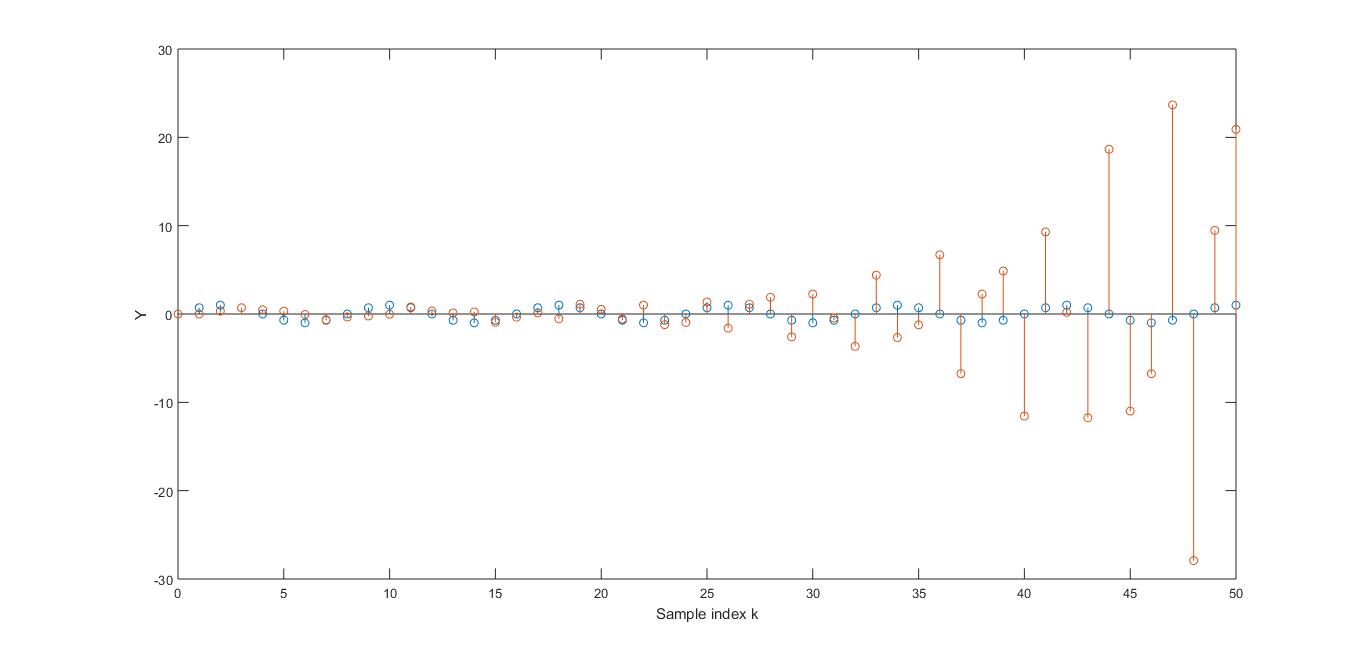
hold on

stem(n,y),xlabel('Sample index k'),

ylabel('Y'),

Output:

Figure 4.4 : to perform IIR system using MATLAB



Task 2: To perform FIR system using MATLAB

Code:

Nx=51;

b=[0,.28,.44,.09];a=1;

n=(0:Nx-1);

x=sin(2\*pi\*0.125\*n);

y=filter(b,a,x);

figure(1)

stem(n,x),xlabel('Sample index k'),

ylabel('X'),

hold on

stem(n,y),xlabel('Sample index k'),

ylabel('Y'),

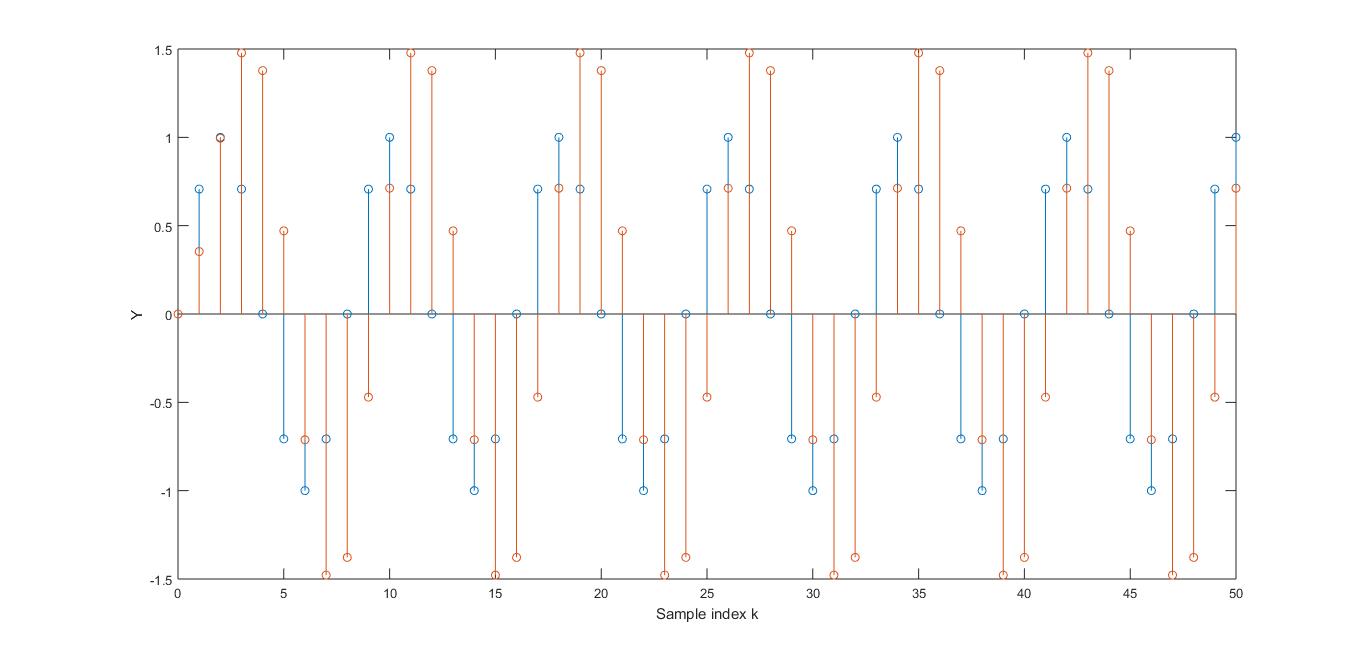
Output:

Figure 4.5 : to perform FIR system using MATLAB

Task 3: factorization of a cascade realization for IIR Transfer Function

Code:

format long

num=input('Numerator coefficient vector=');

den=input('Denominator coefficient vector=');

[z,p,k]=tf2zp(num,den)

sos=zp2sos(z,p,k)

Output:

Numerator coefficient vector=[0 0 0 2 4]

Denominator coefficient vector=[1 .6 -.16 -.96 0]

z = -2

p =

0.000000000000000 + 0.000000000000000i

-0.732860062728479 + 0.756187502774937i

-0.732860062728479 - 0.756187502774937i

0.865720125456957 + 0.000000000000000i

k = 2

sos =Columns 1 through 4

0 0 2.000000000000000 1.000000000000000

0 1.000000000000000 2.000000000000000 1.000000000000000

Columns 5 through 6

-0.865720125456957 0

1.465720125456958 1.108903410895386

Task 4: parallel realization of an IIR transfer Function

Code:

num=input('Numerator coefficient vector=');

den=input('Denominator coefficient vector=');

[r1,p1,k1]=residuez(num,den);

[r2,p2,k2]=residue(num,den);

disp('Parallel form I')

disp('Residues are');disp(r1);

disp('poles are at');disp(p1);

disp('Constant value');disp(k1);

disp('Parallel form II')

disp('Residues are');disp(r2);

disp('poles are at');disp(p2);

disp('Constant value');disp(k2);

Output:

Numerator coefficient vector=[15 -6]

Denominator coefficient vector=[3 .5 -.5]

Parallel form I

Residues are

5.400000000000000

-0.400000000000000

9 disp('poles are at');disp(p1);

poles are at

-0.500000000000000

0.333333333333333

Constant value

Parallel form II

Residues are

5.400000000000000

-0.400000000000000

poles are at

-0.500000000000000

0.333333333333333

Constant value