Table of Contents

Program to verify Sampling Theorem

```
clc;
clear all;
t = -100:0.01:100;
fm = 0.02;
x = cos(2*pi*t*fm);
subplot(2,2,1);
plot(t,x);
xlabel('time in sec')
ylabel('x(t)')
title('Continuous time signal')


fs1 =0.02 %fs1 < 2fm
n=-2:2;
x1 = cos(2*pi*fm*n/fs1)
subplot(2,2,2) ;
stem(n, x1)
hold on
subplot (2,2,2)
plot (n, x1, ':')
xlabel('n')
ylabel('x(n) ')
title('discrete time signal x(n) for fs<2f')
fs1 = 0.02 %fs1<2fm
```

```
fs2 = 0.04 %fs=2fm
n1 = -4:4;
x2 = cos (2*pi*fm*n1/fs2)
subplot (2,2,3)
stem (n1, x2)
hold on
subplot (2,2,3)
plot (n1, x2, ':')
xlabel ('n1')
ylabel('x(n)')
title("discrete time signal")


fs3 = 0.6 %fs3>2*fm
n2 = -50:50;
x3 = cos(2*pi*fm*n2/fs3)
subplot(2,2,4)
stem(n2,x3)
hold on
subplot(2,2,4)
plot(n2,x3,':')
xlabel('n3')
ylabel('x(n2)')
title('discrete time signal x(n2) for fs>2fm')
```

```
fs1 =

    0.0200


x1 =

     1     1     1     1     1


fs1 =

    0.0200


fs2 =

    0.0400


x2 =

     1    -1     1    -1     1    -1     1    -1     1


fs3 =
```

```
    0.6000


x3 =

  Columns 1 through 7

   -0.5000   -0.6691   -0.8090   -0.9135   -0.9781   -1.0000   -0.9781

  Columns 8 through 14

   -0.9135   -0.8090   -0.6691   -0.5000   -0.3090   -0.1045    0.1045

  Columns 15 through 21

    0.3090    0.5000    0.6691    0.8090    0.9135    0.9781    1.0000

  Columns 22 through 28

    0.9781    0.9135    0.8090    0.6691    0.5000    0.3090    0.1045

  Columns 29 through 35

   -0.1045   -0.3090   -0.5000   -0.6691   -0.8090   -0.9135   -0.9781

  Columns 36 through 42

   -1.0000   -0.9781   -0.9135   -0.8090   -0.6691   -0.5000   -0.3090

  Columns 43 through 49

   -0.1045    0.1045    0.3090    0.5000    0.6691    0.8090    0.9135

  Columns 50 through 56

    0.9781    1.0000    0.9781    0.9135    0.8090    0.6691    0.5000

  Columns 57 through 63

    0.3090    0.1045   -0.1045   -0.3090   -0.5000   -0.6691   -0.8090

  Columns 64 through 70

   -0.9135   -0.9781   -1.0000   -0.9781   -0.9135   -0.8090   -0.6691

  Columns 71 through 77

   -0.5000   -0.3090   -0.1045    0.1045    0.3090    0.5000    0.6691

  Columns 78 through 84

    0.8090    0.9135    0.9781    1.0000    0.9781    0.9135    0.8090

  Columns 85 through 91
```

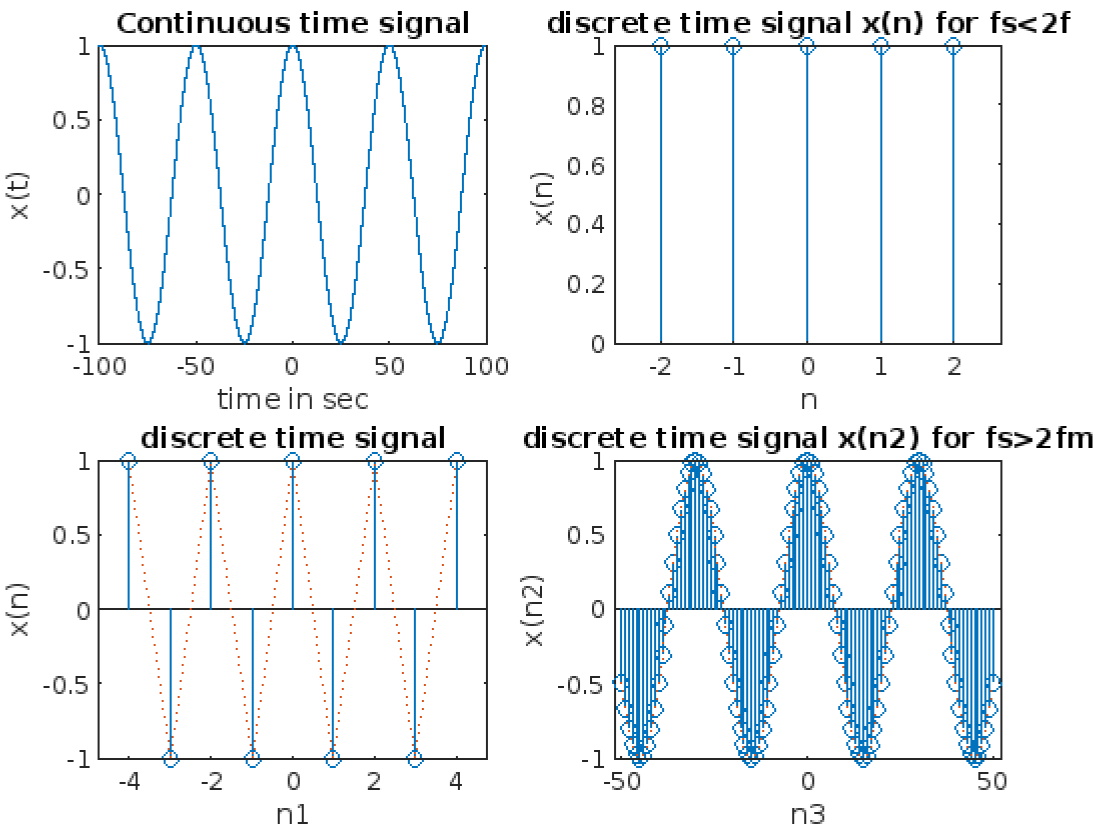
```
  0.6691    0.5000    0.3090    0.1045   -0.1045   -0.3090   -0.5000

Columns 92 through 98

 -0.6691   -0.8090   -0.9135   -0.9781   -1.0000   -0.9781   -0.9135

Columns 99 through 101

 -0.8090   -0.6691   -0.5000
```



Convolution

```
x = [1,2,3,4]
h = [-3,2,1]
y = conv(x,h)
```

```
x =

     1     2     3     4


h =
```

```
    -3     2     1


y =

    -3    -4    -4    -4    11     4
```

Linear Convolution

```
clc;
clear all;
close all;
x = [1,2,3];
h = [1,1,1];
y1 = conv(x,h)
stem(y1)
xlabel('n values');
ylabel('Amplitude');
title('Linear Convolution');

m = length(x);
m1 = length(h);
subplot(2,2,2);
```

```
X = [x ,zeroes(1,m)];
H = [h ,zeroes(1,m1)];
for i=1: (m1+m-1)
    y(i) = 0;
    for j=1:m

        if(i-j+1)>0
            y(i) = y(i) + X(j)*H(i-j+1)
        end
    end
end

subplot(2,2,3);
```

```
y1 =

     1     3     6     5     3


Undefined function 'zeroes' for input arguments of type 'double'.

Error in lab1 (line 82)
```

```
X = [x ,zeroes(1,m)];
        ^^^^^^^^^^^
```

Linear Convolution 2

```
% WAP of linear convolution
clc;
clear all;
close all;
x=[1,2];
h=[1,2,4];
%y1=conv(x,h)
%stem(y1)
%xlabel('n values');
%ylabel('Amplitude');
%title('Linear Convolution');
m=length(x);
m1=length(h);
subplot(2,2,1)
stem(x);
xlabel('n values');
ylabel('Amplitude');
title('x(n)');
subplot(2,2,2)
stem(h);
xlabel('n values');
ylabel('Amplitude');
title('h(n)');
X=[x,zeros(1,m)];
H=[h,zeros(1,m1)];
for i=1:(m1+m-1)
    y(i)=0;
    for j=1:m
        if(i-j+1)>0
            y(i)=y(i)+X(j)*H(i-j+1);
        end
    end
end
subplot(2,2,3);
stem(y);
xlabel('n values');
ylabel('Amplitude');
title('y output');
y1=conv(x,h)
subplot(2,2,4);
stem(y1)
xlabel('t values');
ylabel('Amplitude');
title('Linear convolution');
```

Circular Convolution

WAP to implement a code of circular convolution

```
clc;
clear all;
x = [1,1,2,2];
h = [1,2,3,4];
subplot(2,2,1);
stem(x);
xlabel('n values');
ylabel('Amplitude');
title('x(n)');
subplot(2,2,2);
stem(h);
xlabel('n values');
ylabel('Amplitudes');
title('h(n)');
N1 = length(x);
N2 = length(h);
N = max(N1,N2);
if(N1<N2)
    x = [x,zeros(1,N-N1)]
end
if(N1>=N2)
    h = [h,zeros(1,N-N2)]
end
y = zeros(1,N);
for m=1:N
    for n=1:N
        z = mod(m-n,N);
        y(m)=y(m) + x(n).*h(z+1);
    end
end
subplot(2,2,3)
stem(y)
xlabel('n values')
ylabel('Samples')
title('Circular Convolution')
```

Third One

```
x = [1,1,1,2,1,1];
h = [1,2,2,1];
subplot(2,2,1);
stem(x);
xlabel('n values');
ylabel('Amplitude');
title('x(n)');
subplot(2,2,2);
stem(h);
xlabel('n values');
ylabel('Amplitude');
title('h(n)');

N1 = length(x)
N2 = length(h)
```

```
N = N1 + N2 -1
x = [x,zeros(1,N-N1)]
h = [h,zeros(1,N-N2)]

y = zeros(1,N)

for n=0:N-1
    for m=0:N-1
        z=mod(n-m,N)
        y(n+1) = y(n+1) + x(m+1).*h(z+1)
    end
end

subplot(2,2,3);
stem(y);
xlabel('n values');
ylabel('amplitude');
title('Output y(n)');
disp('y = ')
disp(y)
```

OCT 14

To solve difference equation

```
clc;
x = [0 0 ones(1,10)];
y_past = [1 2];
y(1)= y_past(1);
y(2)= y_past(2);
for k=3: (length(x));
    y(k) = 1.5*y(k-1) - y(k-2) + 2*x(k-2);
end

disp('Solution for the given difference equation is');
disp(y);
stem(-2:(length(y)-3),y);
xlabel('input x(n)');
ylabel('output y(n)');
title('Differential Equation')
```

Oct 14 -2

Solving another difference equation

```
x = [0 0 ones(1,10)];
y_past = [1 2];
y(1)= y_past(1);
y(2)= y_past(2);
for k=4: (length(x));
    y(k) = 5*y(k-3) + 4*x(k-3);
end
```

```
disp('Solution for the given difference equation is');
disp(y);
stem(-3:(length(y)-4),y);
xlabel('input x(n)');
ylabel('output y(n)');
title('Difference Equation')
```

DFT

```
clc;
clear all;
x1 = input('Enter the sequence: ');
n = input('Enter the length: ');
m = fft(x1,n);
disp('N-point DFT of the given sequnce: ')
disp(m);
z = abs(fft(x1,n));
disp('magnitude of the DFT sequence is:');
disp(z);
N = 0:1:n-1;
subplot(2,2,1);
stem(N,z);
xlabel('length');
ylabel('output');
title('magnitude spectrum');
a = angle(m)
subplot(2,2,2);
stem(N,a);
xlabel('length');
ylabel('output');
title('phase spectrum');
```

Auto Co-relation Rxx

```
%Calculate Auto Corelation of sequence [1 2 3 4]

clc;
clear all;
x = input('ENter the input sequence');
Rxx = xcorr(x);
disp('Auto Correlated sequence,Rxx: ');
disp(Rxx);
t = -(length(x) -1) : 1 : (length(x) -1);
stem(t,Rxx);
xlabel('Time: ');
ylabel('Magnitude ');
title('Auto Correlation output of a sequence');
```

Cross Corelation

```
%Solve for the cross corelation to a sequence
```

```
clc;
clear all;
x=input('Enter the first sequence:');
Rxx=xcorr(x);
disp('auto correlated sequence, Rxx:');
disp(Rxx);
y=input('Enter the second sequence:');
Ryy=xcorr(y);
disp('cross correlated sequence, Ryy');
disp(Ryy);
Rxy=xcorr(x,y);
disp('cross correlated sequence, Rxy:');
disp(Rxy);
Ryx=xcorr(y,x);
disp('cross correlated sequence,Rxy:');
disp(Ryx);
t=-(length(x)-1):1:(length(x)-1);
stem(t, Rxy);
xlabel('Time');
ylabel('Magnitude')
title('Cross Relation of the given two sequences');
```

FT Rectangle Sir

```
clear all;
close all;
Fs = 40;
Ts = 1/Fs;
t = -1:Ts:1;
width = 1;
x = rectpuls(t, width);
figure,
plot(t,x),
xlabel('time'), ylabel('magnitude');
[M N] = size(t);
F = fft(x,N);
ff = (0:N-1)*Fs/N;
figure,
plot(ff,fftshift(abs(F)));
```

FT Rectangle GPT

```
clc;
clear all;
close all;

% Input parameters
a = input('Enter width of pulse: ');
t = linspace(-5, 5, 1000); % Define a range for t, to generate a smooth plot

% Generate the rectangular pulse
x = rectpuls(t, a);
```

```
% Plot the original pulse
figure;
subplot(2,1,1);
plot(t, x);
title('Rectangular Pulse');
xlabel('Time');
ylabel('Amplitude');
grid on;

% Compute Fourier Transform using FFT
X = fft(x);
f = linspace(-0.5, 0.5, length(X)) * length(t) / (t(end) - t(1)); % Frequency
axis

% Plot the magnitude of the Fourier Transform
subplot(2,1,2);
plot(f, fftshift(abs(X)));
title('Magnitude Spectrum of the Pulse');
xlabel('Frequency');
ylabel('Magnitude');
grid on;
```

Fourier Transform Square

```
clear all;
clc;
n=0:0.01:29;
x=square(2*pi*n);
x1=abs(fft(x));
subplot(2,1,1);
plot(n,x);
title('input signal');
subplot(2,1,2);
plot(x1,'r-x');
title('fourier transform');
```

Fourier Tranform Square:

```
clear all;
clc;

% Time vector
n = 0:0.01:30;
x = square(2 * pi * 0.5 * n); % Square wave with frequency 0.5 Hz

% Compute Fourier Transform
X = fft(x);
f = (0:length(X)-1) * (1 / max(n)); % Frequency axis

% Magnitude Spectrum
X_mag = abs(X);

% Plot input signal
```

```
subplot(2,1,1);
plot(n, x);
title('Input Signal (Square Wave)');
xlabel('Time');
ylabel('Amplitude');
grid on;

% Plot Fourier Transform
subplot(2,1,2);
plot(f, X_mag, 'r-x');
xlim([0 10]); % Adjust the x-axis limit for better visibility
title('Magnitude Spectrum of the Square Wave');
xlabel('Frequency');
ylabel('Magnitude');
grid on;
```

FT SINC

```
%Fourier Transform of sinc function
clc;
clear all;
close all;
t=input('Enter duration of sinc function')
x=sinc(5*t);
figure(1);
subplot(2,1,1);
plot(t,x);
X=fft(x);
subplot(2,1,2);
plot(t,fftshift(abs(X)));
```

FIR LPF/HPF

```
clc;
clear all;
close all;
N=30;
fs=8000;
fc=2000;
wc=fc/(fs/2);
h=fir1(N,wc,'low',hamming(N+1));
freqz(h,1,1024,fs);
```

FIR BPF/BSF

```
clc;
clear all;
close all;
N=10;
fs=8000;
fc1=2000;
fc2=3000;
```

```
wc1=fc1/(fs/2);
wc2=fc2/(fs/2);
h=fir1(N,[wc1,wc2],'bandpass',hamming(N+1));
freqz(h,1,1000,fs);
```

IIR Butterworth LPF/HPF

```
clc;
clear all;
close all;
fs=1000;
kp=3;
ks=60;
fp=40;
fstop=150;
wp=fp/(fs/2);
wstop=fstop/(fs/2);
[N wc]=buttord(wp,wstop,kp,ks)
[b a]=butter(N,wc,'low') % low or high as required
freqz(b,a,1000,fs)
```

IIR Butterworth BPF/BRF

```
clc;
clear all;
close all;
fs=1000;
kp=3;
ks=60;
fp1=200;
fp2=300;
fs1=100;
fs2=400;
wp1=fp1/(fs/2);
wp2=fp2/(fs/2);
ws1=fs1/(fs/2);
ws2=fs2/(fs/2);
[N wc]=buttord([wp1,wp2],[ws1,ws2],kp,ks)
[b a]=butter(N,wc,'bandpass')
freqz(b,a,1000,fs)
```

IIR Chebyshev LPF/HPF

```
clc;
clear all;
close all;
fs=1000;
kp=3;
ks=60;
fp=40;
fstop=150;
wp=fp/(fs/2);
```

```
wstop=fstop/(fs/2);
[N wc]=cheb1ord(wp,wstop,kp,ks)
[b a]=cheby1(N,kp,wc,'high')
freqz(b,a,1000,fs)
```

IIR Chebyshev BPF/BRF

```
clc;
clear all;
close all;
fs=1000;
kp=3;
ks=60;
fp1=200;
fp2=300;
fs1=100;
fs2=400;
wp1=fp1/(fs/2);
wp2=fp2/(fs/2);
ws1=fs1/(fs/2);
ws2=fs2/(fs/2);
[N wc]=cheb1ord([wp1,wp2],[ws1,ws2],kp,ks)
[b a]=cheby1(N,kp,wc,'bandpass')
freqz(b,a,1000,fs)
```

Published with MATLAB® R2024b