[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

**SIGNALS AND SYSTEMS**

**PROJECT-02**

**DR.B NIRANJAN KRUPA**

**Name: AKASH RAVI BHAT**

**SRN:PES1UG21EC025**

**Sem:3**

**Sec:A**

**1)Obtaining the Fourier Series of a given signal and displaying its partial sums.**

**Code:**

% Obtaining the Fourier Series of a given Signal

clc;clear;close;

T=3;%period of the signal

t=-T/2:0.01:T/2;

x=t>=-.5&t<=0.5;%signal x(t)

plot(t,x);a=gca();a.Box='off';title('x(t) periodic with period T=3');

%to find Fourier series coefficients ck

for k=-15:15

ck(k+16)=sum(x.\*[exp(-j\*(2\*pi/T)\*k.\*t)]);%analysis equation

ck(k+16)=0.01\*ck(k+16)/T;

end

disp('Fourier Series coefficients ck');disp(ck);

%to get back x(t) from ck

t=-T/2:0.01:T/2;

for i=1:length(t) % length(var) returns the length of the largest array

% dimension in X

x\_new(i)=0;

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

for k=-15:15

x\_new(i)=x\_new(i)+ck(k+16)\*exp(j\*(2\*pi/T)\*k\*t(i));%synthesis equation

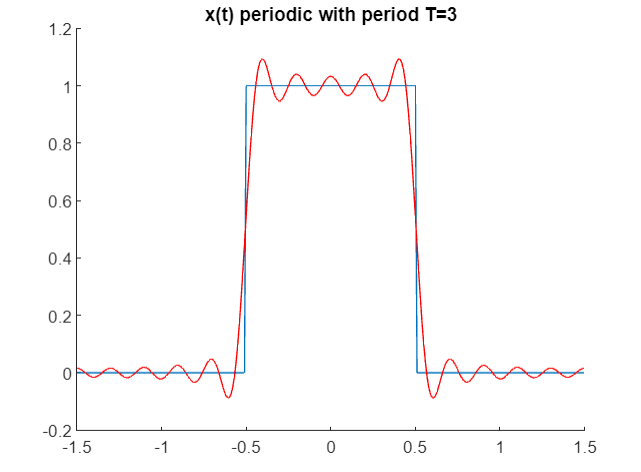
end

end

hold on;%plotting values on same fig window without deleting previous plot

plot(t,real(x\_new),'r');

hold off;

****

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

**2)Obataining a Fourier Transform of a given signal and plotting its given spectrum.**

**Code:**

%Obtaining Fourier Transform of a given Signal

clc;clear;close;

delta\_t=0.01;t=0:delta\_t:10;%time axis

x=exp(-2\*t); %signal x(t)

w=-10:0.01:10;

%find X(jw)

for j=1:length(w)

% length(var) returns the length of the largest array

% dimension in X

Xjw(j)=sum(x.\*exp(-i\*w(j)\*t))\*delta\_t; %analysis equation //X(jw) is used to find the frequency response

end

%plot x(t)

subplot(311);plot(t,x);title('x(t)=e^(^-^a^t^)u(t),a>0');

%plot magnitude spectrum

subplot(312); plot(w,abs(Xjw));

%Y = abs( X ) returns the absolute value of each element in array X // abs to get the **absolute** value for every element in the input S

title('magnitude spectrum');

xlabel('frequency w');

ylabel('|X(jw)|');

%plot phase spectrum

subplot(313); plot(w,angle(Xjw));

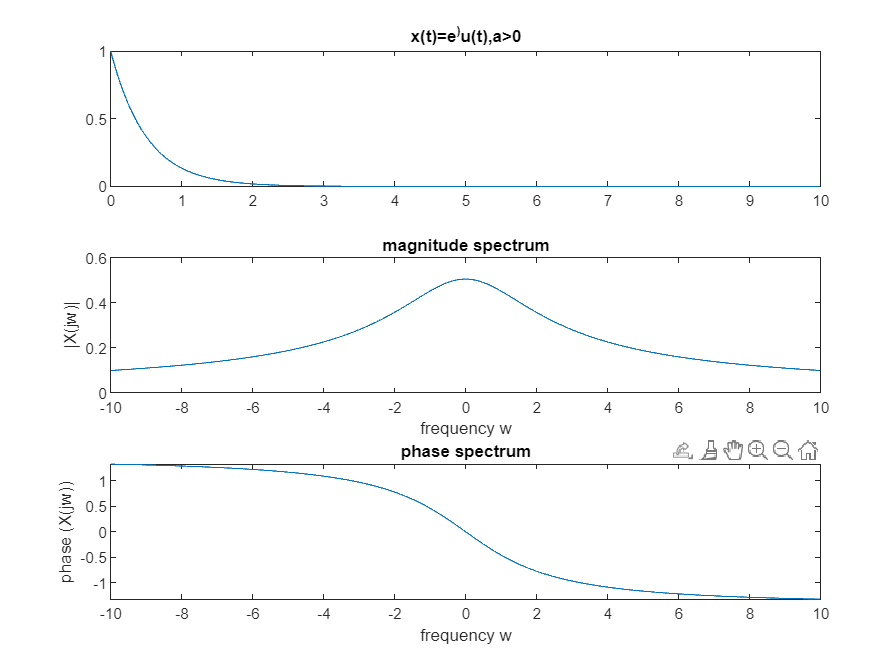
%theta = angle( z ) returns the phase angle in the interval [-π,π] for each element of a complex array z

title('phase spectrum');

xlabel('frequency w');

ylabel('phase (X(jw))');

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

****

**3)Verification of properties of Fourier representations of continuous signals.**

% Verification of properties of Fourier representation of

% continuous signals

clc;clear;close;

%these are the variables we require

syms t w;

%this is to make sure the figure pops up in a window named figure1

figure(1);

%title given

title('Time Scaling Property');

%the signal u(t+1)-u(t-1)

x1=heaviside (t+1)-heaviside(t-1);

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

%subplot(nrows, ncols, index, \*\*kwargs) divides the area to fit many plots

%in ine figure

subplot(231);

%range from [-4,4] plot plots a function between specified limits.

fplot(x1,[-4,4]);

title('x1=u(t+1)-u(t-1)');

xlabel('time t');

%plot function u(t/2+1)-u(t/2-1)

x2=heaviside(t/2+1)-heaviside(t/2-1);

subplot(232);

fplot(x2,[-4,4]);

title('x2=u(t/2+1)-u(t/2-1)');

xlabel('time t');

x3=heaviside(2\*t+1)-heaviside(2\*t-1);

subplot(233);

fplot(x3,[-4,4]);

title('x3=u(2\*t+1)-u(2\*t-1)');

xlabel('time t');

%fourier(f) returns the Fourier Transform of f. By default, the function symvar

% determines the independent variable, and w is the transformation variable.

X1W=fourier(x1,w);

subplot(234);

%Absolute value and complex magnitude collapse all in page Syntax Y = abs(X)

% Description example Y = abs(X) returns the absolute value of each element in array X.

fplot(abs(X1W),[-20,20]);%plots function described in arguments,here abs(X1)

title('X1W');

xlabel('\omega');

%fourier(f,transVar) uses the transformation variable transVar instead of w.

X2W=fourier(x2,w);

subplot(235);

fplot(abs(X2W),[-20,20]);

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

title('X2W');

xlabel('\omega');

X3W=fourier(x3,w);

subplot(236);

fplot(abs(X3W),[-20,20]);

title('X3W');

xlabel('\omega');

%2 time shifting property

figure(2);

t0=2;

y1=heaviside(t)-heaviside(t-1);

subplot(321);

fplot(y1,[-4,4]);

title('y1=u(t)-u(t-1)');

xlabel('time t');

%time is shifted by t0

y2=heaviside(t-t0)-heaviside(t-1-t0);

subplot(322);

fplot(y1,[-4,4]);

title('y2=u(t-t0)-u(t-1-t0)');

xlabel('time t');

%fourier of the time shifted

Y1W=fourier(y1,w)

subplot(323);

fplot(abs(Y1W), [-20,20]);

title('abs (Y1W)');

xlabel('\omega');

subplot(325);

fplot(angle(Y1W),[-20,20]);

title('angle (Y1W)');

xlabel('\omega');

Y2W=fourier(y2,w)

subplot(324);

fplot(abs(Y2W), [-20,20]);

title('abs (Y2W)');

xlabel('\omega');

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

subplot(326);

fplot(angle(Y2W),[-20,20]);

title('angle (Y2W)');

xlabel('\omega');

%Simplify a symbolic expression that contains logarithms and powers.

% By default, simplify does not combine powers and logarithms because

% combining them is not valid for generic complex values.

timeshift\_result=simplify(Y1W\*exp(-j\*w\*t0)-Y2W)

%if results is 0 then it satisfies time shifting property

%same magnitude spectrum for both but different phase

%to verify linearity property

figure(3);

z1=exp(-t)\*(heaviside(t)-heaviside(t-1));

subplot(231);

fplot(z1,[-2,6]);

title('z1=exp(-t)\*u(t)-u(t-1)');

xlabel('time t');

z2=(heaviside(t-3)-heaviside(t-4));

subplot(232);

fplot(z2,[-2,6]);

title('u(t-3)-u(t-4)');

xlabel('time t');

z12=2\*z1 + 3\*z2;

%linear combination of z1 and z2

subplot(233);

fplot(z12, [-2,6]);

title('signal z12=2\*z1+3\*z2');

xlabel('time t');

Z1W=fourier (z1,w)

subplot(234);

fplot(abs(Z1W),[-20,20]);

title('abs(Z1W)');

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

xlabel('\omega');

Z2W=fourier (z2,w)

subplot(235);

fplot(abs(Z2W),[-20,20]);

title('abs(Z2W)');

xlabel('\omega');

Z12W=fourier (z12,w)

subplot(236);

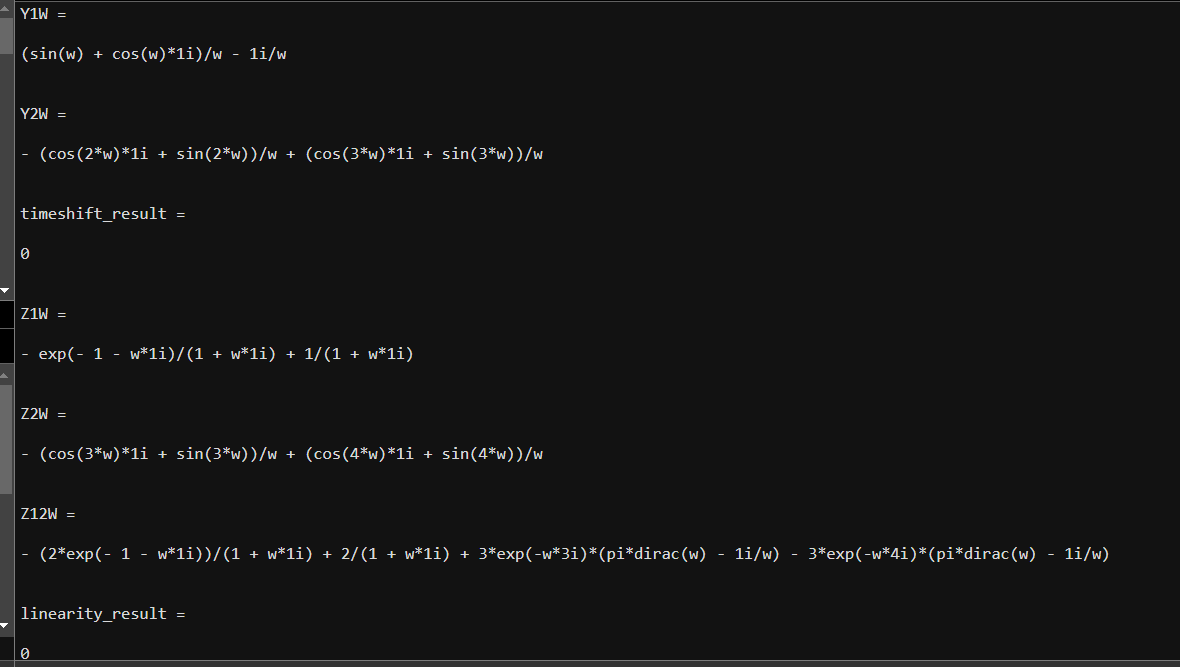
fplot(abs(Z12W),[-20,20]);

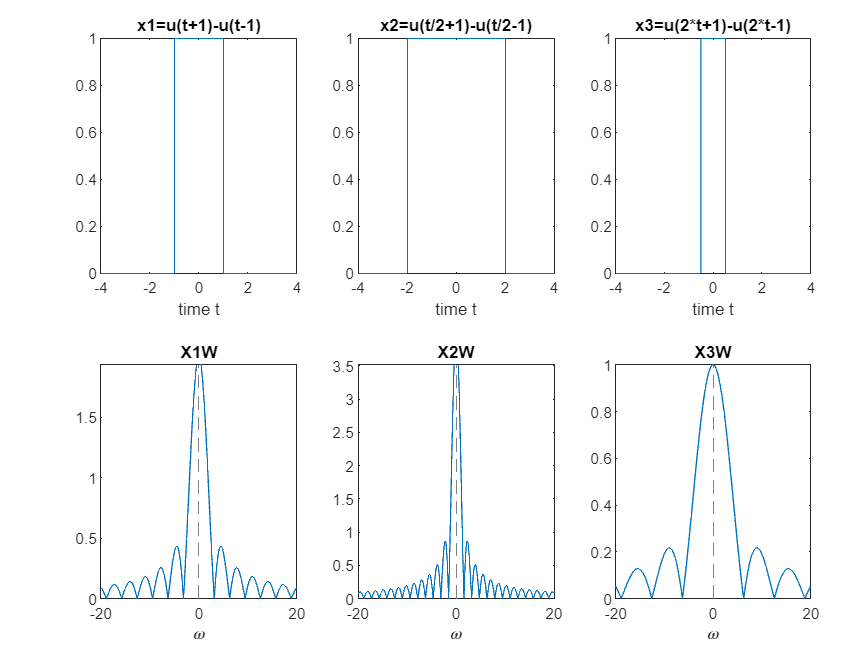
title('abs(Z12W)');

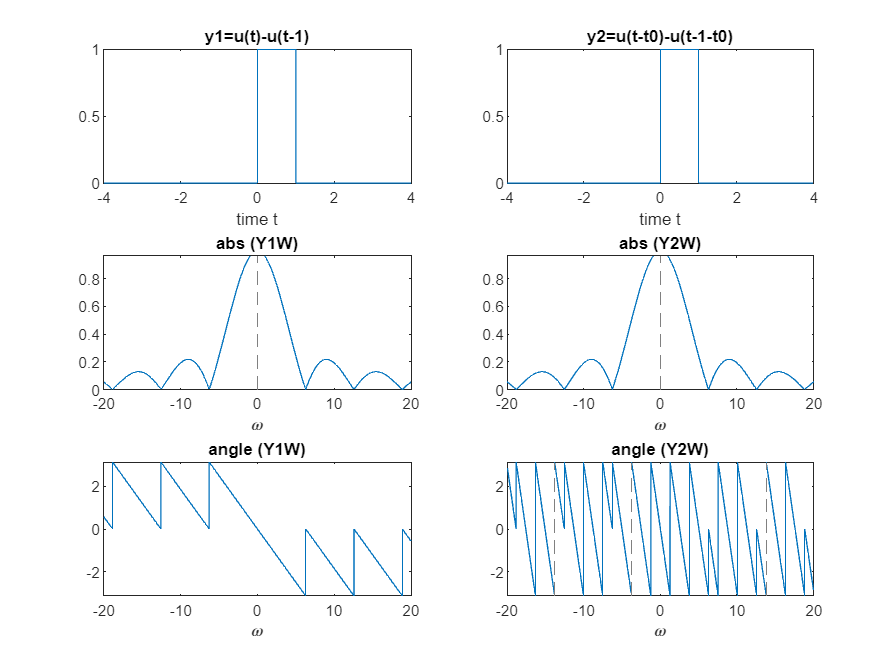
xlabel('\omega');

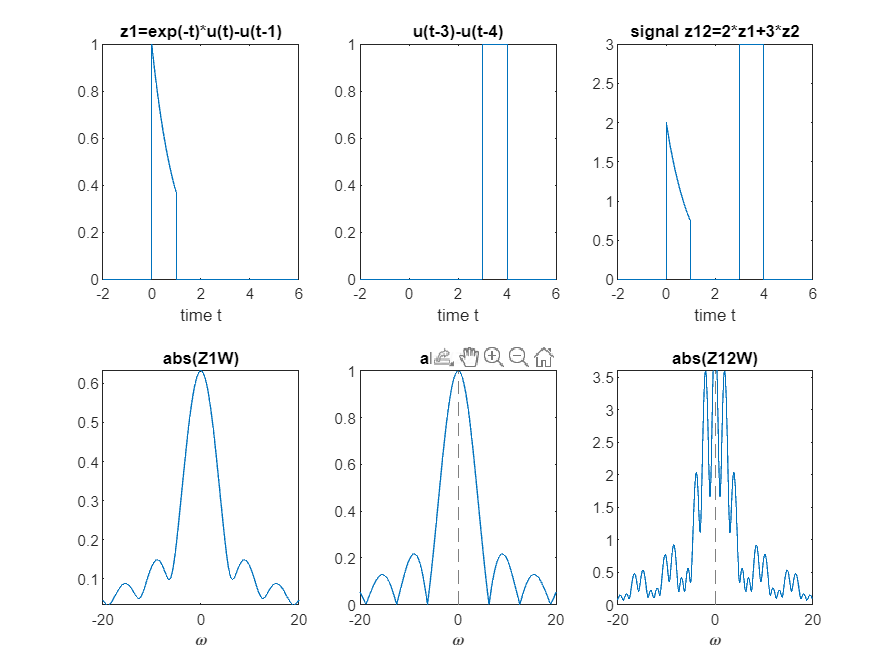
linearity\_result=simplify((2\*Z1W+3\*Z2W)-Z12W)

%if result is 0, then it satisfy linearity property

****

****

****

****

**4)Obtaining DTFT of a given signals and verification of its properties.**

**Code:**

% Obtainig the DTFT of a given signal and verification of its properties

clc;clear;close;

n=[-10:10];

xn1=(0.5.^n).\*(n>=0); %signal x[n] for n=-10 to 10

xn2=(0.5.^-n).\*(-n>=0); %signal xn12

figure(1);

subplot (321);

stem(n,xn1);

title('input sequence x(n)');

subplot(322);

stem(n,xn2);

title('input sequence x(-n)');

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

%define the frequency axis rom -3pi to 3pi;

w=-3\*pi:(pi/500):3\*pi;

X1jw=DTFT(n,xn1,w);

subplot(323);

plot(w,abs(X1jw));

title('Magnitude Spectrum');

xlabel('frequency w');

ylabel('|X1(w)|');

subplot(325);

plot(w,angle(X1jw));

title('Phase Spectrum');

xlabel('frequency w');

ylabel('ang(X1(w))');

%(1) from magnitude spectrum and phase spectrum we note that DTFT is

%periodic with 2\*pi

%(2) to verify time reversal property DTFT {x[-n]} is x(-jw)

X2jw=DTFT(n,xn2,w);

subplot(324);

plot(w,abs(X2jw));

title('Magnitude Spectrum');

xlabel('frequency w');

ylabel('|X2(w)|');

subplot(326);

plot(w,angle(X2jw));

title('Phase Spectrum');

xlabel('frequency w');

ylabel('ang(X2(w))');

%(3) to verify Parseval's relation (low of conservation of energy)

% leftside= rightside, so Parseval's relation hold good

leftside=sum(abs(xn1).^2)

delta\_w=pi/500;

omega=-pi:delta\_w:pi;

X1jomega=DTFT(n,xn1,omega);

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

rightside=(1/(2\*pi)\*sum(abs(X1jomega).^2)\*delta\_w)

disp('LHS=RHS Parsevals Relation is verified');

%(4) to verify frequency shifting property let w0=1

% we can observe that magnitude spectrum of xn3 is same as

%magnitude spectrum of xn1 but its shifted by w0

w0=1;

xn3=xn1.\*exp(j\*w0\*n);

X3jw=DTFT(n,xn3,w)

figure(2);

subplot(2,1,1);

plot(w,abs(X1jw));

title('Magnitude Spectrum of xn3=xn1');

xlabel('frequency w');

ylabel('|X1(w)|');

subplot(2,1,2);

plot(w,abs(X3jw));

title('Magnitude Spectrum of xn3=xn1.\*exp(j\*w0\*n where) where w0=1');

xlabel('frequency w');

ylabel('|X3(w)|');

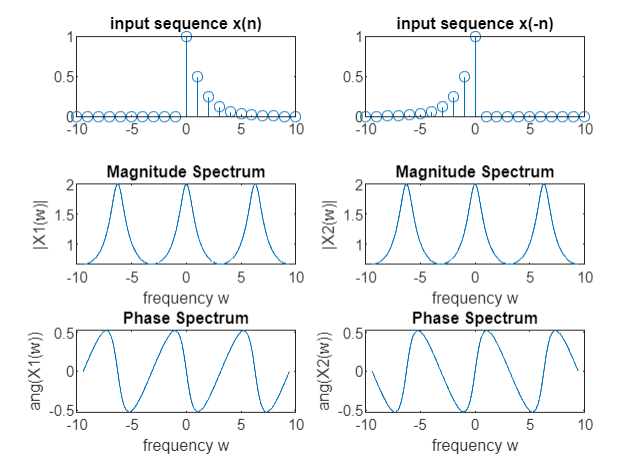
function [Xjw] = DTFT(n,xn,w)

for a=1:length(w)

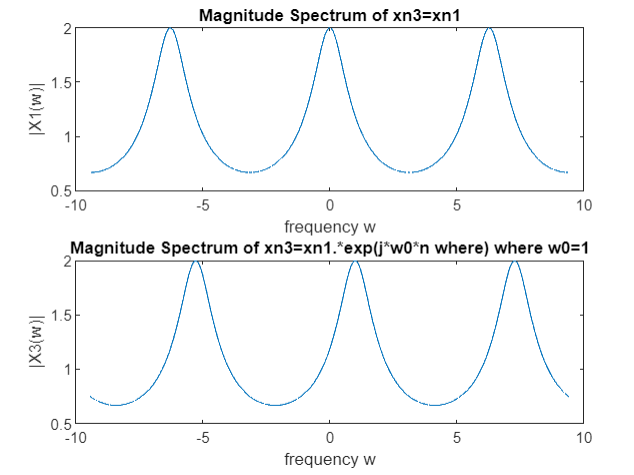
Xjw(a)=sum(xn.\*exp(-j\*w(a)\*n));

end

end

****

[*Published with MATLAB® R2022b*](https://www.mathworks.com/products/matlab)

****