**Fourier Transform (FT):**

We want to find the amplitude spectrum of the two-frequency signal:

x(t)=cos(2π100t)+cos(2π500t)

We begin by creating a vector, x, with sampled values of the continuous time function. If we want to sample the signal every 0.0002 seconds and create a sequence of length 250, this will cover a time interval of length 250\*0.0002 = 0.05 seconds. A plot of this signal is generated using the following MATLAB code:

>> clear

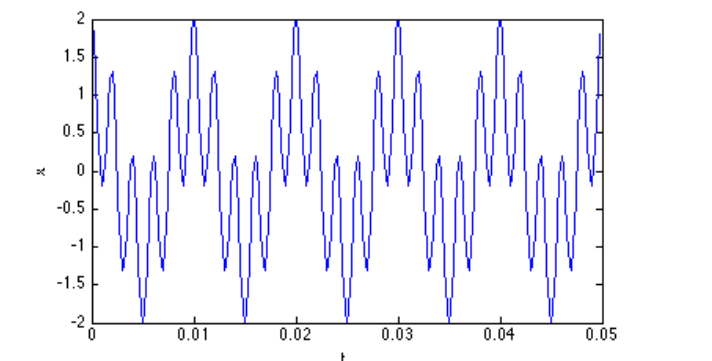
>> N=250;

>> ts=.0002;

>> t=[0:N-1]\*ts;

>> x=cos(2\*pi\*100\*t)+cos(2\*pi\*500\*t);

>> plot(t,x)



We can find approximate the Fourier transform integral for 0 ≤f ≤800 Hz using:

>> k=0;

>> for f=0:1:800

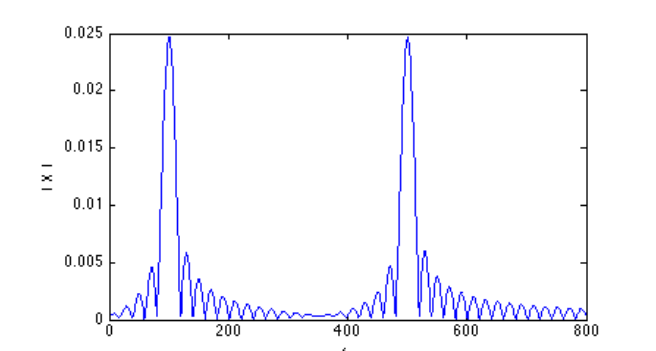
k=k+1;

X(k)=trapz(t, x.\*exp(-j\*2\*pi\*f\*t));

end

>> f=0:800;

>> plot(f, abs(X))



As expected the peaks in the spectrum are at 100 and 500 Hz the two frequencies contained in the signal. Theoretically, we expect to see impulse functions at these two frequencies and zero at every other frequency.

clc

clear all

close all

Fa=10;

T=1/Fa;

t=0:T/99:T;

y=5\*sin(2\*pi\*Fa\*t) +2\*sin(2\*pi\*2\*Fa\*t)+2\*sin(2\*pi\*3\*Fa\*t);

figure(1)

plot(t,y);

Fs=640;

Ts=1/Fs;

N=T/Ts;

n=0:N-1;

yy=5\*sin(2\*pi\*Fa\*n\*Ts)+ 2\*sin(2\*pi\*2\*Fa\*n\*Ts)+ 2\*sin(2\*pi\*3\*Fa\*n\*Ts);

figure(2)

stem(n,yy)

h=[];

b=[];

for(k=1:1:N)

for (n=1:1:N)

ff=yy(n)\*exp(-j\*2\*pi\*(k-1-(N/2))\*(n-1-(N/2))/N);

h=[ h ff];

end

p=sum(h);

b=[b p];

h=0;

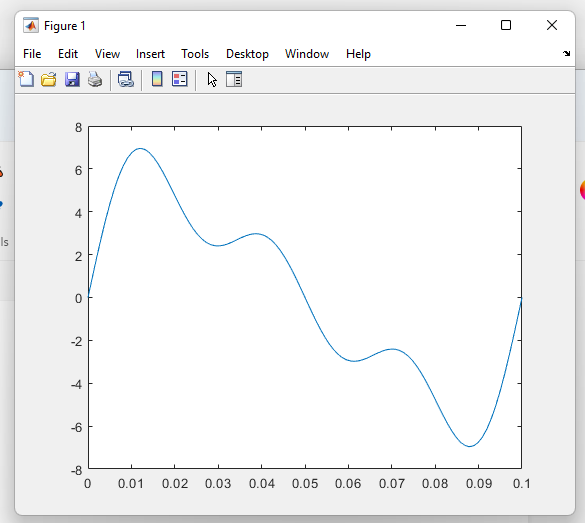
end

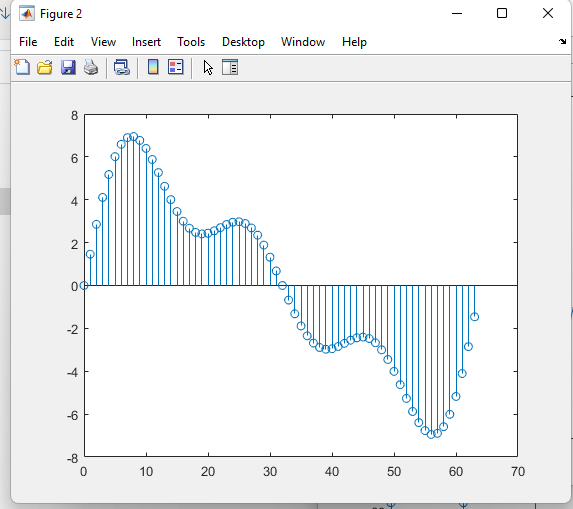
figure(3)

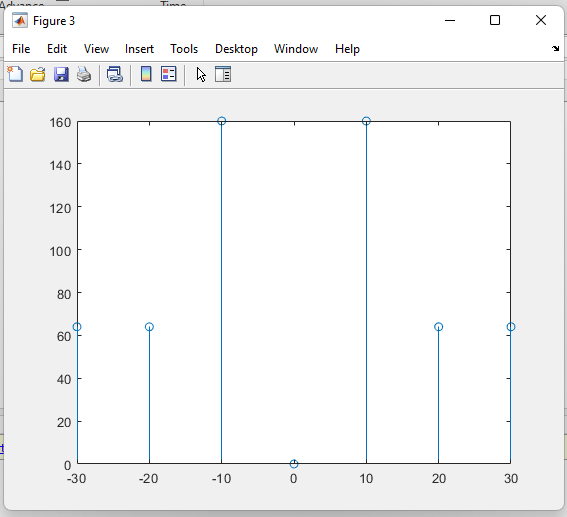
f=Fs\*(-N/2:(N/2)-1)/N;

stem(f,abs(b));

axis([ -30 30 0 160]);







***MATLAB code of DFT:***

Problem:

As an example of computing a DFT using the matrix form, suppose the input samples are as follows.

Code:

clc;

close all;

clear all;

%DFT of a sequence

N=input('Enter the length of sequence=');

x=input('Enter the sequence=');

n=[0:1:N-1];

k=[0:1:N-1];

wN=exp(-j\*2\*pi/N);

nk=n'\*k;

wNnk=wN.^nk;

Xk=x\*wNnk;

disp('Xk=');

disp(Xk);

mag=abs(Xk)

subplot(2,1,1);

stem(k,mag);

grid on;

xlabel('.......>k');

title('MAGNITUDE OF FOURIER TRANSFORM');

ylabel('Magnitude');

phase=angle(Xk)

subplot(2,1,2);

stem(k,phase);

grid on;

xlabel('.........>k');

title('PHASE OF FOURIER TRANSFORM');

ylabel('Phase');

***Output:***

Input from Keyboard:

Enter the length of sequence= 4

Enter the sequence= [3, -1, 0, 2]

***Output:***

Xk= 4.0000 3.0000 + 3.0000i 2.0000 - 0.0000i 3.0000 - 3.0000i

mag = 4.0000 4.2426 2.0000 4.2426

phase = 0 0.7854 -0.0000 -0.7854



**MATLAB code of IDFT:**

Problem: As a numerical check, suppose we compute the IDFT of the results from before example.

clc;

close all;

clear all;

%IDFT of a sequence

Xk=input('Enter X(K)=');

[N,M]=size(Xk);

if M~=1;

Xk=Xk.';

N=M;

end;

xn=zeros(N,1);

k=0:N-1;

for n=0:N-1;

xn(n+1)=exp(j\*2\*pi\*k\*n/N)\*Xk;

end;

xn=xn/N;

disp('x(n)=');

disp(xn);

plot(xn);

grid on;

plot(xn);

stem(k,xn);

xlabel('....>n');

ylabel('........>Magnitude');

title('IDEF OF A SEQUENCE');

***Input:***

Enter X(K)=[4 ,3+3j, 2, 3-3j]

Output:

x(n)= 3.0000

-1.0000 + 0.0000i

0.0000 + 0.0000i

2.0000 - 0.0000i



