DSP Repost

Date:- 25-05-20

Course:- Digital signal processing

Topic:- Day 1

Name:- G. Ravitej a

USN:- 4AL16EC101

Sem:- 6th 'B'

* Fourier transform and Wavelet's co-ordinate system transform:-

$u(x,y,t)$

$u_t = \alpha \nabla^2 u$

SVD = data-drive FFT

* FFT:- Fast fourier transform is used to process a Audio signal's, video, etc:- by compressing and Representing Efficiently by using FFT.

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos 2\pi kt + b_k \sin 2\pi kt)$$

Where $k \rightarrow$ Frequency

$a_k, b_k \rightarrow$ coefficients.

* It consists of cosine & sine components.

* Fourier transform:-

$$X_a(F) = \int_{-\infty}^{\infty} x(t) \cos 2\pi ft \, dt$$

$$X_b(F) = \int_{-\infty}^{\infty} x(t) \sin 2\pi ft \, dt.$$

$$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} \, dt$$

* continious fourier Transform:-

$$X(t) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} \, dt.$$

* Discrete fourier Transform:-

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$$

* Inner products in Hilbert space:-

$$\langle f(x), g(x) \rangle = \int_a^b f(x) g(x)\, dx.$$

* Fourier series using MathLab:-

```
clear all, close all, clc
figure
set(gcf, position, [1500, 200, 2000, 1200])
L=pi;
N=1024;
dx=2*L/(N-1);
f=0.*x;
f(N/4:N/2)=4*(1:N/4+1)/N;
f(N/2+1; 3*N/4)=1-4*(0:N/4-1)/N;
plot(x,f,'-k','linewidth',3.5), hold on
cc=jet(20);
Ao=sum(f.*ones(size(x))*dx/pi;
fFs=Ao/2
for k=1:20
    A(k)=sum(z.*cos(Pi*x/L)*dx/pi;
    B(k)=sum(z.*sin(Pi*k*x/L)*dx/pi;
    ZFs=fFs+A(k)*cos(k*Pi*x/L)+B(k)*sin
                              (k*Pi*x/2);
Plot(x,fFs,'_','colour',cc(k,:):linewidth:2)
 pause(.1)
end.
```

* Fourier series using Python :-

=) In [ ] : import numpy as np

```
import matplotlib. pyplot as plt
from matplotlib. cm import get_cmap
plt . rc params ['figure. fig size'] = [8,8]
plt. rc params . update [ {'font. size : 18)
dx = 0.001
L = np.pi
x= L* np. arrange ( 1+dx, 1+dx, dx)
n= len(x)
nq uart = int[np, floor (n/4)]
f= np. zero's _ like (x)
f [nquart:2 +nquart] = (4/n)* np . arrange (!nquart+1)
f (2* nquart :3 * nquart ]= np.ones (nquart)- (u/co)*(
   arrange (0. nquart +1)
fig . ax = plt. subplots()
ax. plot ( x, t, '_', colour = 'k', linewidth _2).
```

——————x——————