Experimental Methods Problem Sheet 1,13

25 October 2023 15:26

13) a) i) $X \sim \text{Poisson}(\lambda = 13)$

$$P(X=10) = \frac{13^{10}}{10!}e^{-13} = \underline{0.086}$$

ii) $X' \sim \text{Poisson}\left(\frac{13}{2}\right)$

$$P(X'=3) = \frac{(13/2)^3}{3!}e^{-\frac{13}{2}} = \underline{0.069}$$

iii) $P(X'<4) = \sum_{i=1}^{3} P(X'=i)$

$$= \frac{(13/2)}{1!}e^{-\frac{13}{2}} + \frac{(13/2)^2}{2!}e^{-\frac{13}{2}} + \frac{(13/2)^3}{3!}e^{-\frac{13}{2}}$$

$$= \underline{0.11}$$

b) i) $X \sim \text{Geometric}\left(p = \frac{1}{100}\right)$

Geometric distribution since waiting on first success, nothing after matters

$$P(X \leq 60) = 1 - P(X \geq 61)$$

Probability of first success within 60 cars

$$1 - P(X \geq 61) = 1 - (1-p)^{60}$$

$= 1 -$ P(60 fails initially)

$$\therefore P(X \leq 60) = 1 - \left(1 - \frac{1}{100}\right)^{60}$$
$$= \underline{0.453}$$

$$\therefore P(X \geq 61) = \underline{0.547}$$

ii)

Rate of pickups in 1 hour $= \frac{\text{pickups}}{\text{car}} \times \frac{\text{cars}}{\text{hour}}$

iii) Rate of pickups in 1 hour $= \frac{\text{pickups}}{\text{car}} \times \frac{\text{cars}}{\text{hour}}$

$\therefore \lambda = \frac{1}{100} \times 60 = 0.6$

$\therefore X \sim \text{Poisson}(\lambda = 0.6)$

$1 - P(X=0) = 1 - \frac{0.6^0}{0!} e^{-0.6}$

$= \underline{0.451}$

c) i) $Z \sim \text{Gaussian}(\mu = 0, \sigma = 1)$

$P(-1.4 < Z < 1.4) = 0.838$

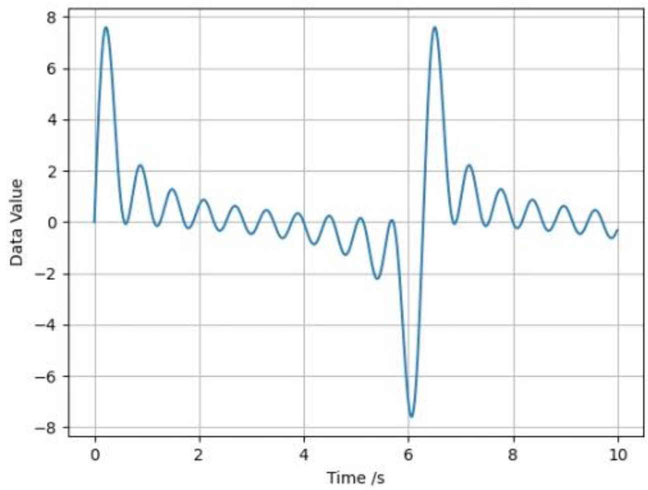
ii) $P(Z > 1.4) = 0.0808$

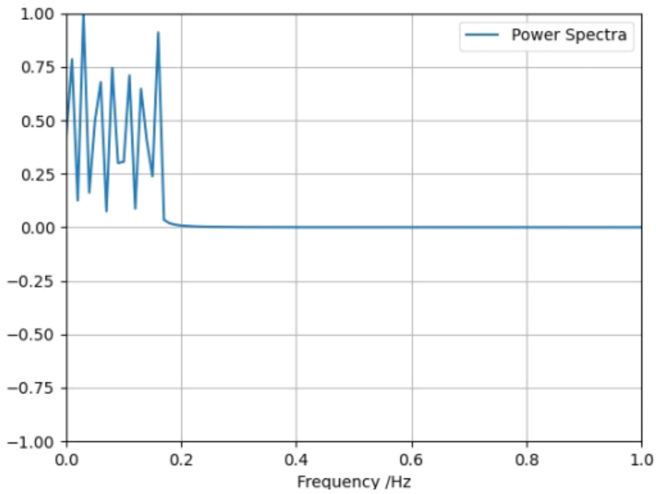
1)

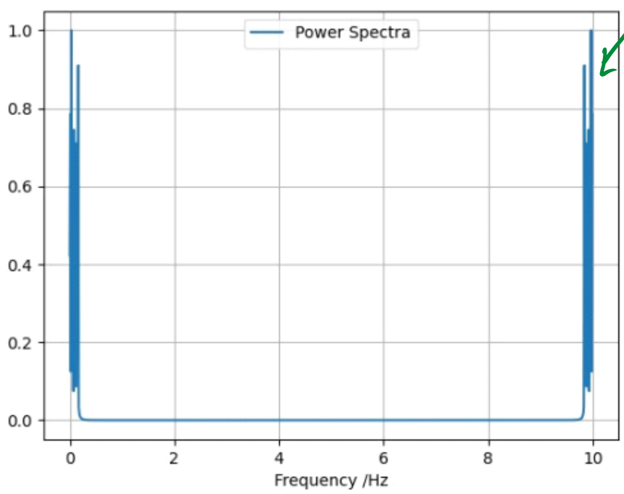
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft, ifft

dt=0.01
MaxTime=10
time=np.arange(0,10,dt)

def noise(t):
    return 0.5*((np.sin(10*(t-0.03))+0.01)*(0.01+np.cos(15*t)))+0.3*((np.sin(13.5*(t-0.3))-0.1)*(0.1+np.cos(7.5*t+4)))

n=int(input("n:    "))

data=[0]*(int(round(MaxTime/dt,0)))

for i in range(0,n):
    data += np.sin((i+1)*time)

data[i] +=noise(time[i])

plt.plot(time,data)
plt.xlabel('Time /s')
plt.ylabel('Data Value')
plt.grid(True)
plt.savefig('EM_Q1.png')
print("Saved plot as: EM_Q1_Data.png")
plt.clf()

FFTdata=fft(data)

#plt.plot(time,(np.real(FFTdata))/np.max(np.real(FFTdata)), label="Real part")
#plt.plot(time,(np.imag(FFTdata))/np.max(np.imag(FFTdata)), label="Imaginary part")
plt.plot(time,(np.absolute(FFTdata)**2)/np.max(np.absolute(FFTdata)**2), label="Power Spectra")
plt.legend(loc="best")
plt.xlabel('Frequency /Hz')
plt.axis((0,1,-1,1))
plt.grid(True)
plt.savefig('EM_Q1_FFT.png')
print("Saved plot as: EM_Q1_FFT_Data.png")
plt.clf()
```

```python
plt.axis((0,1,-1,1))
plt.grid(True)
plt.savefig('EM_Q1_FFT.png')
print("Saved plot as: EM_Q1_FFT_Data.png")
plt.clf()
```

Screen clipping taken: 30/10/2023 17:06







Not sure why this region appears at 10Hz?

14) $I_{avg} = \frac{ne}{\Delta t}$

Follows poisson like distribution with the mean of n electrons, and $\sigma = \sqrt{\lambda} = \sqrt{\mu}$

$\therefore \Delta I = \sqrt{n}\frac{e}{\Delta t}$

mean number of electrons contributing to a single pixel: $\frac{n}{N}$

Average current to single pixel: $\frac{I}{N}$

Current to a single pixel: $\frac{I}{N}$, $I = \frac{ne}{\Delta t}$, $\therefore I_{avg} = \frac{ne}{\Delta t N}$

Expected electrons in time Δt: $\frac{n}{N}$

$\therefore$ Expected current fluctuation: $\Delta I = \sqrt{\frac{n}{N}} \times \frac{e}{\Delta t}$

Consider: $\frac{\sqrt{I_{avg}}}{\Delta I}$

Had to lose Δt for it to work?

$$\therefore \frac{\sqrt{I_{avg}}}{\Delta I} = \frac{\sqrt{\frac{ne}{N}}}{e\sqrt{\frac{n}{N}}} = \frac{1}{\sqrt{e}}$$

$$\therefore \frac{I_{avg}}{\Delta I} = \frac{\sqrt{I_{avg}}}{\sqrt{e}} = \sqrt{\frac{I}{N}} \times \frac{1}{\sqrt{e}} = \sqrt{\frac{I}{Ne}}$$ as given