Quarto Relatorio PCO119

João Vitor Yukio Bordin Yamashita

September 22, 2022

1 Questão 1

Para ler o sinal foi feito:

```
    import scipy
    import matplotlib.pyplot as plt
    import scipy.io

    mat = scipy.io.loadmat('dados_fft.mat')
    datax = mat.get('x')
    datay = mat.get('y')

    #Visualizar os sinais:
    plt.plot(datax[0], color = '#1A4314')
    plt.title('Sinal X')

    plt.plot(datay[0])
    plt.title("Sinal Y")

```

Listing 1: Ler o sinal .mat

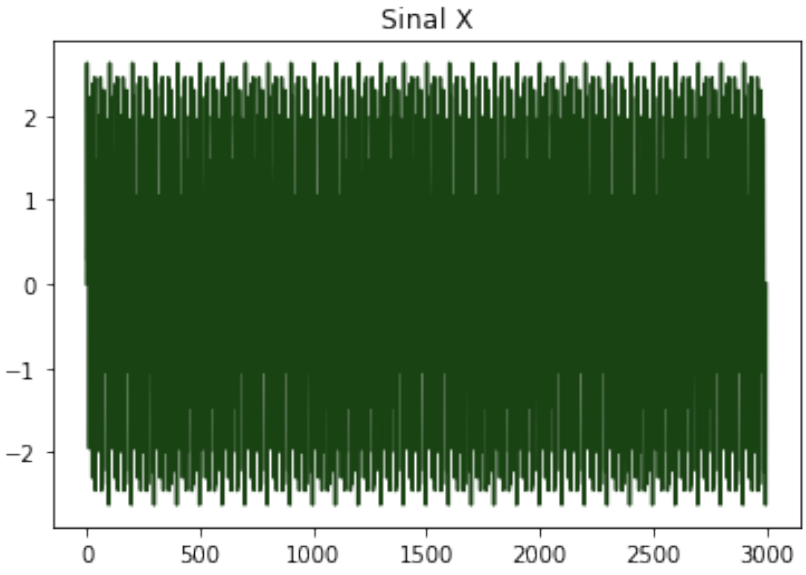

Figure 1: Sinal X

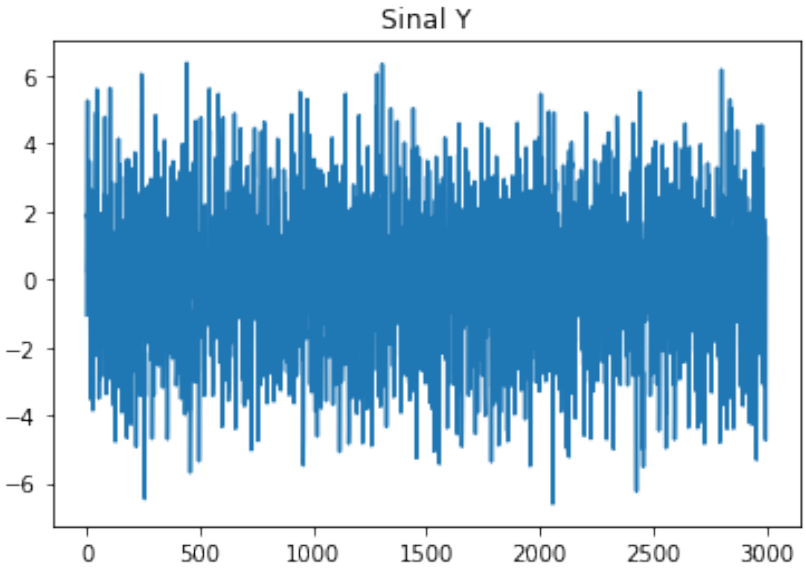

Figure 2: Sinal Y

Para calcular a FFT:

```
from scipy.fft import fft, fftfreq
import numpy as np

Fs = 1000
N = (Fs * 3) + 1  #Numero de amostras

#Sinal X
yf1 = fft(datax[0])/N #Normaliza
xf1 = fftfreq(N, 1/Fs)

plt.title("FFT do sinal X")
plt.plot(xf1, 2*np.abs(yf1), color = '#1A4314')


#Sinal Y
yf2 = fft(datay[0])/N
xf2 = fftfreq(N, 1/Fs)

plt.title("FFT do sinal Y")
plt.plot(xf2, 2*np.abs(yf2))


```

Listing 2: Calculo da FFT

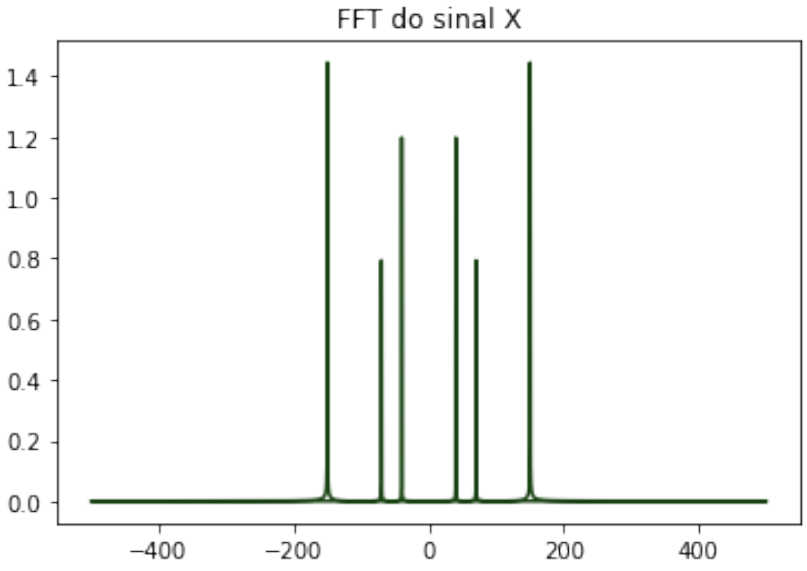

Figure 3: FFT do Sinal X

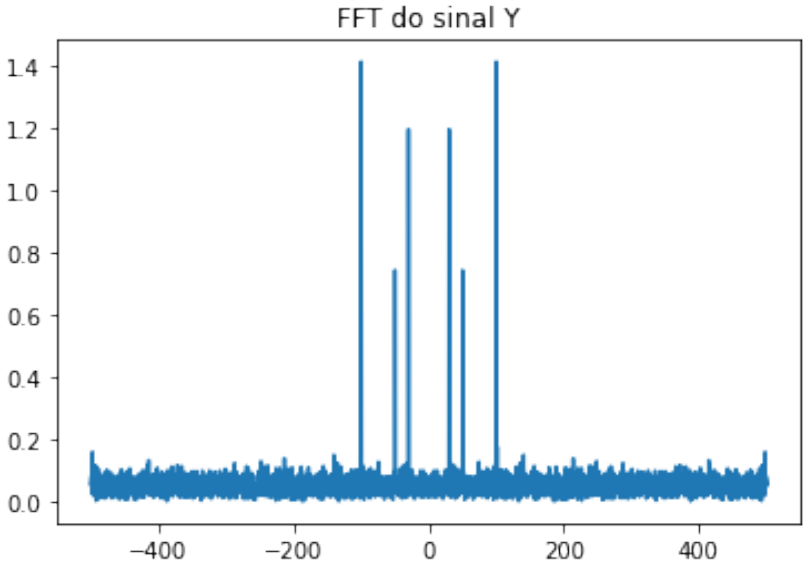

Figure 4: FFT do Sinal Y

Para observar melhor as frequências do sinal vamos fazer plots apenas de um lado:

```
    plt.title("FFT do sinal X")
    plt.plot(xf1[0:500], 2*np.abs(yf1[0:500]), color = '#1A4314')
    plt.grid()

    plt.title("FFT do sinal Y")
    plt.grid()
    plt.plot(xf2[0:500], 2*np.abs(yf2[0:500]))


```

Listing 3: Zoom na fft

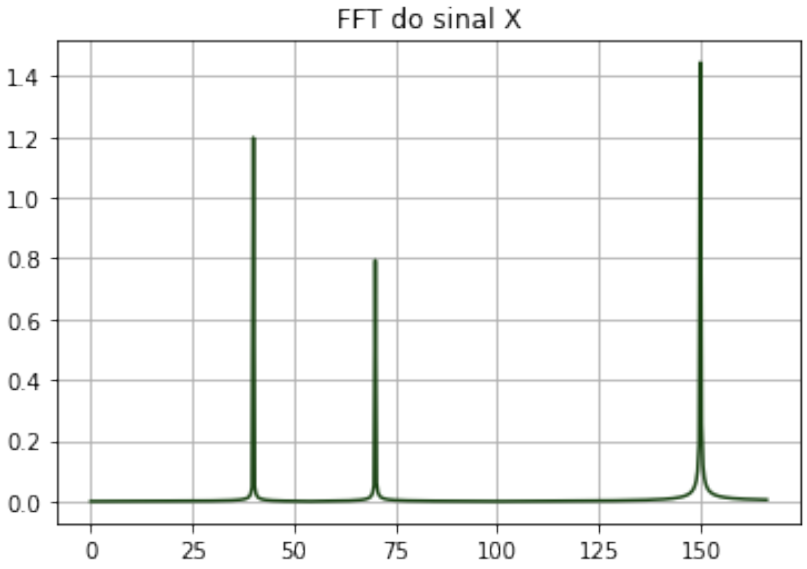

Figure 5: FFT do Sinal X

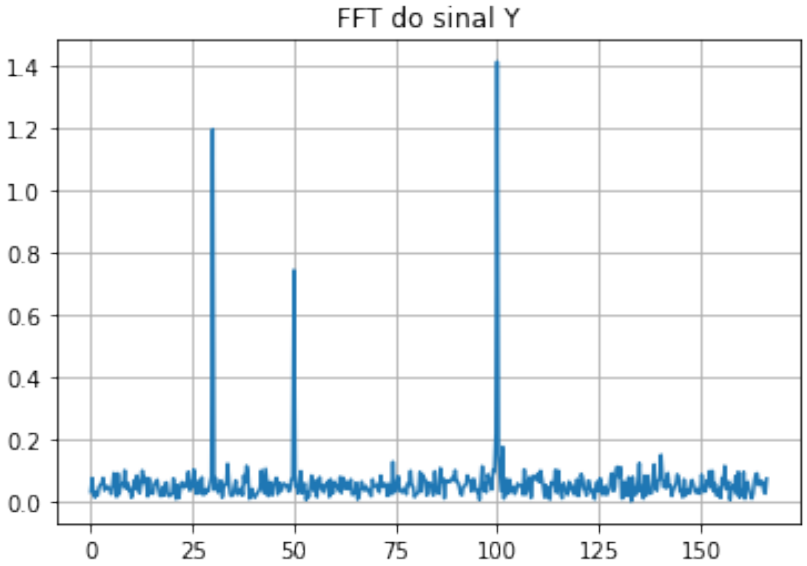

Figure 6: FFT do Sinal Y

Para o sinal Y podemos perceber que temos bem mais ruído quando comparado com o sinal X.

Podemos ver aproximadamente onde estão as frequências dos sinais, podemos fazer um pequeno script para encontrar onde estão as frequências mais importantes:

```
freqs = []#Para o sinal X
amps = []
for i in range(0,501):
    if(2*abs(yf1[i]) > 0.5):
        freqs.append(xf1[i])
        amps.append(2*abs(yf1[i]))
print(freqs)
print(amps)
Output:
    freqs = [39.98667110963013, 69.97667444185272, 149.95001666111298]
    amps = [1.1973547085131606, 0.7933060714831651, 1.4442537062534233]
```

```

```

Listing 4: Frequencias e amplitudes

Podemos perceber, que para o sinal X, as frequências que compõem esse sinal são, 40Hz, 70Hz e 150Hz, desta forma o sinal pode ser representado por:

$$X(t) = 1.2sin(2 \times \pi \times 40 \times t) + 0.8sin(2 \times \pi \times 70 \times t) + 1.44sin(2 \times \pi \times 150 \times t)$$

```
    freqs2 = [] #Para o sinal Y
    amps2 = []
    for i in range(0,501):
        if(2*abs(yf2[i]) > 0.5):
            freqs2.append(xf2[i])
            amps2.append(2*abs(yf2[i]))
    print(freqs2)
    print(amps2)
    Output:
        freqs2 = [29.990003332222596, 49.98333888703766, 99.96667777407532]
        amps2 = [1.1963052738127313, 0.743507592399777, 1.4131200443440317]

```

Listing 5: Frequencias e amplitudes

Podemos perceber, que para o sinal Y, as frequências que compõem esse sinal são, 30Hz, 50Hz e 100Hz, desta forma o sinal pode ser representado por:

$$Y(t) = 1.2sin(2 \times \pi \times 30 \times t) + 0.75sin(2 \times \pi \times 50 \times t) + 1.4sin(2 \times \pi \times 100 \times t)$$

2 Questão 2

Para simular o sinal fazemos:

```
    import math
    tsig = np.arange(0, 12, 0.1)

    from math import pi, cos

    sig = []
    for t in tsig:
        sinal = 1 - (8/pi**2) * (cos((pi*t)/2) + (1/9)*cos((3*pi*t)/2) + (1/25)*cos
    ((5*pi*t)/2) + (1/49)*cos((7*pi*t)/2) + (1/81)*cos((9*pi*t)/2))
        sig.append(sinal)
    plt.title("Sinal gerado")
    plt.grid()
    plt.plot(tsig, sig)

```

Listing 6: Simulacao do sinal

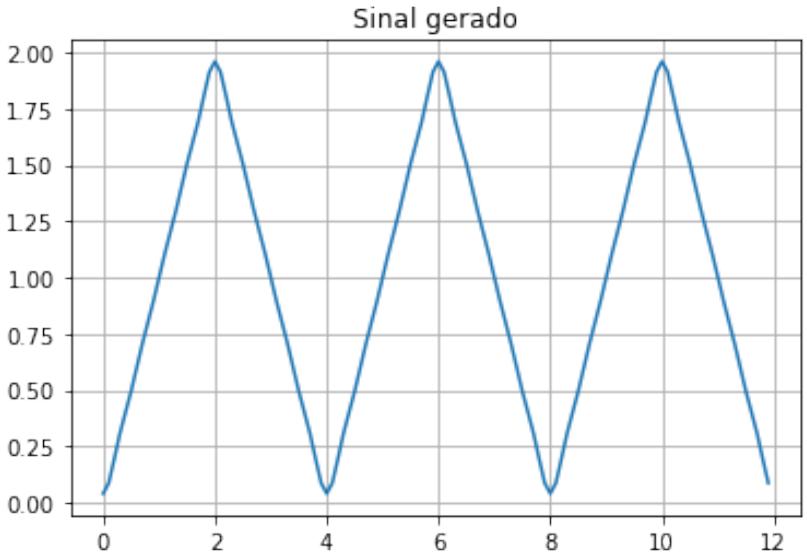

Figure 7: Saída da série

Podemos perceber que se trata da série de Fourier para uma onda triangular.

3 Questão 3

Para gerar o sinal fazemos:

```
    from scipy import signal

    Nsamp = 2000
    Fsamp = 100

    t = np.arange(0, Nsamp/Fsamp, 1/Fsamp)
    sig = 2.5*signal.square(2 * np.pi * 1 * t) + 2.5

    plt.plot(t, sig, color = 'purple')
    plt.title("Sinal quadrado")

```

Listing 7: Sinal quadrado

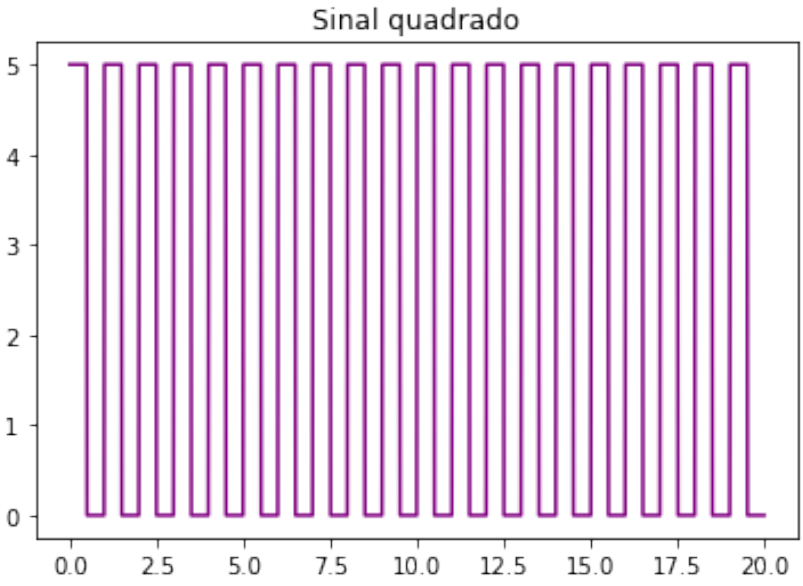

Figure 8: Sinal Quadrado

Para calcular a FFT fazemos:

```
    yf3 = fft(sig)/Nsamp
    xf3 = fftfreq(Nsamp, 1/Fsamp)

    plt.title("FFT do sinal quadrado")
    plt.plot(xf3, 2*np.abs(yf3), color = 'purple')

    #Dando um zoom
    plt.title("FFT do sinal quadrado")
    plt.grid()
    plt.plot(xf3[0:750], 2*np.abs(yf3[0:750]), color = 'purple')

```

Listing 8: FFT do Sinal quadrado

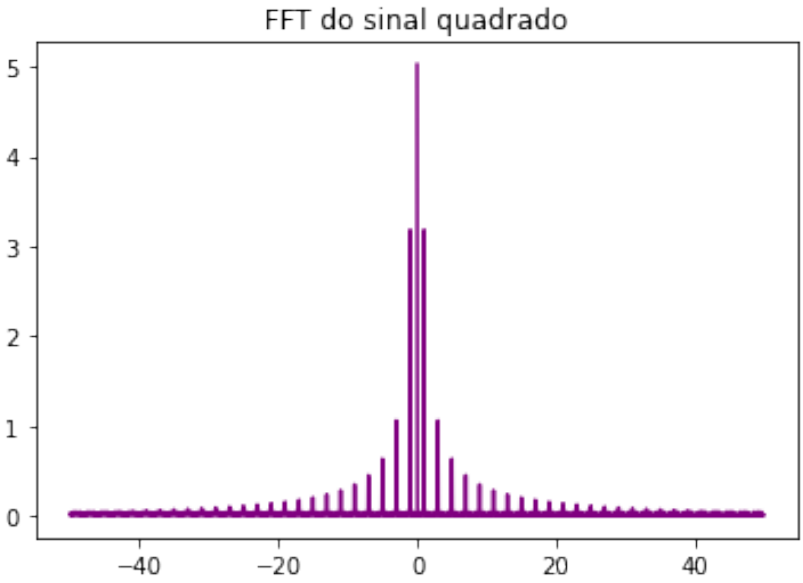

Figure 9: FFT Sinal Quadrado sem zoom

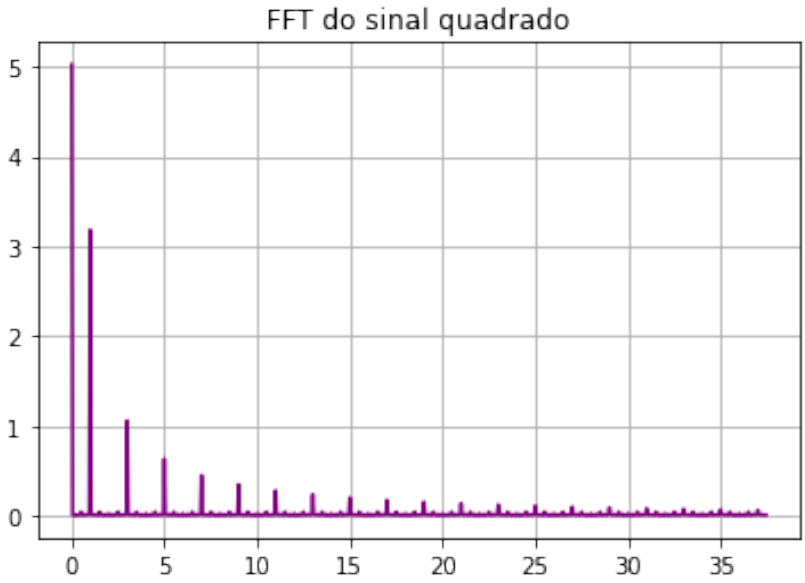

Figure 10: FFT do Sinal Quadrado com zoom

Podemos perceber que esse era o valor esperado, considerando que a representação de uma onda quadrada é uma soma de infinitas ondas senoidais, por meio desse exercício conseguimos perceber que uma onda quadrada é representada no domínio da frequência por infinitas frequências, onde temos uma frequência principal e várias harmonicas formando a onda.