

## TPN°2- Transformada Discreta de Fourier

**Autor:** Pablo D. Folino

**Link del repositorio:** [https://github.com/PabloFolino/MSE\\_PSF\\_TP2.git](https://github.com/PabloFolino/MSE_PSF_TP2.git)

### **Enunciado:**

Dado el archivo `clases/tp2/resolucion_espectral.txt` que contiene 100 valores reales sampleados a  $F_s=200\text{Hz}$ , indique: 1) Resolución espectral. 2) Espectro en frecuencia de la señal. 3) A simple inspección que frecuencia(s) distingue. 4) Aplique alguna técnica que le permita mejorar la resolución espectral y tome nuevamente el espectro. 5) Indique si ahora los resultados difieren del punto 3 y argumente su respuesta. 6) Pegue el link a un pdf con los códigos, gráficos y comentarios..

A simple inspección aparece una frecuencia predominante a 50Hz.

Se realiza un programa(***resolucion\_espectral\_v1.py***) en donde se lee el archivo de referencia, se gráfica la señal de entrada su espectro y características. Luego aplicando la técnica de zero padding se procesa nuevamente.

El programa posee una sección de definiciones de los valores a probar:

```
4  import numpy as np
5  import matplotlib.pyplot as plt
6  import scipy.signal as sc
7  import os
8
9  # Valores del enunciado
10 Fs = 200      # frecuencia de muestreo
11 Ts = 1/Fs
12 N = 100
13
14 # Se utiliza la técnica de zero padding, se agregan M1 veces N
15 # la nueva resolución espectral es  $F_s/(M1+1) \cdot N$ 
16 M1 = 9
```

Se abre el archivo y se revisa el contenido:

```

19 # Se lee el archivo
20 with open("../Informe/Archivos de enunciados/resolucion_espectral.txt","r") as ins:
21     cont = ins.read() # Esto devuelve el contenido completo, no linea por linea
22     arr = eval(cont)
23     señal=arr
24
25 os.system("clear")
26
27 # Se verifica que se leyeron los N elementos
28 if(len(señal)==N):
29     print("\tSe leyeron N={} elementos del archivo".format(N))
30     print("\tEl tipo de elemento es{}".format(type(señal)))
31     print("\tEl rimer elemento del archivo es{}".format(señal[1]))
32     print("\tEl tipo del primer elemento es{}".format(type(señal[1])))
33 else:
34     print("\t ERROR-No se pudieron leer los N={} elementos del archivo".format(N))

```

Se calcula la transformada de la señal de entrada:

```

36 #-----Señal original-----
37 #-----Discreto-----
38 ts = Ts*np.arange(0, N,1)
39 #-----Tranformada-----
40 fft = np.fft.fft(señal)/N
41 M_fft = np.fft.fftshift(abs(fft))
42 F=np.fft.fftfreq(int(Fs/2),Ts)
43 F=np.fft.fftshift(F)
44 #-----Potencia promedio-----
45 pot_promedio=np.sum(M_fft**2)

```

Se calcula la transformada usando el método de zero padding:

```

47 #-----Señal con Zero Padding-----
48 N_zp=(M1+1)*N
49 señal_zp=np.zeros(N_zp)
50 for i in range(0, len(señal)):
51     señal_zp[i]=señal[i]
52
53 #-----Discreto-----
54 ts_zp = Ts*np.arange(0,N_zp,1)
55 #-----Tranformada-----
56 fft_zp = np.fft.fft(señal_zp)/N_zp
57 M_fft_zp = np.fft.fftshift(abs(fft_zp))
58 F_zp=np.fft.fftfreq(int(Fs*(M1+1)/2),Ts)
59 F_zp=np.fft.fftshift(F_zp)
60 #-----Potencia promedio-----
61 pot_promedio_zp=np.sum(M_fft_zp**2)

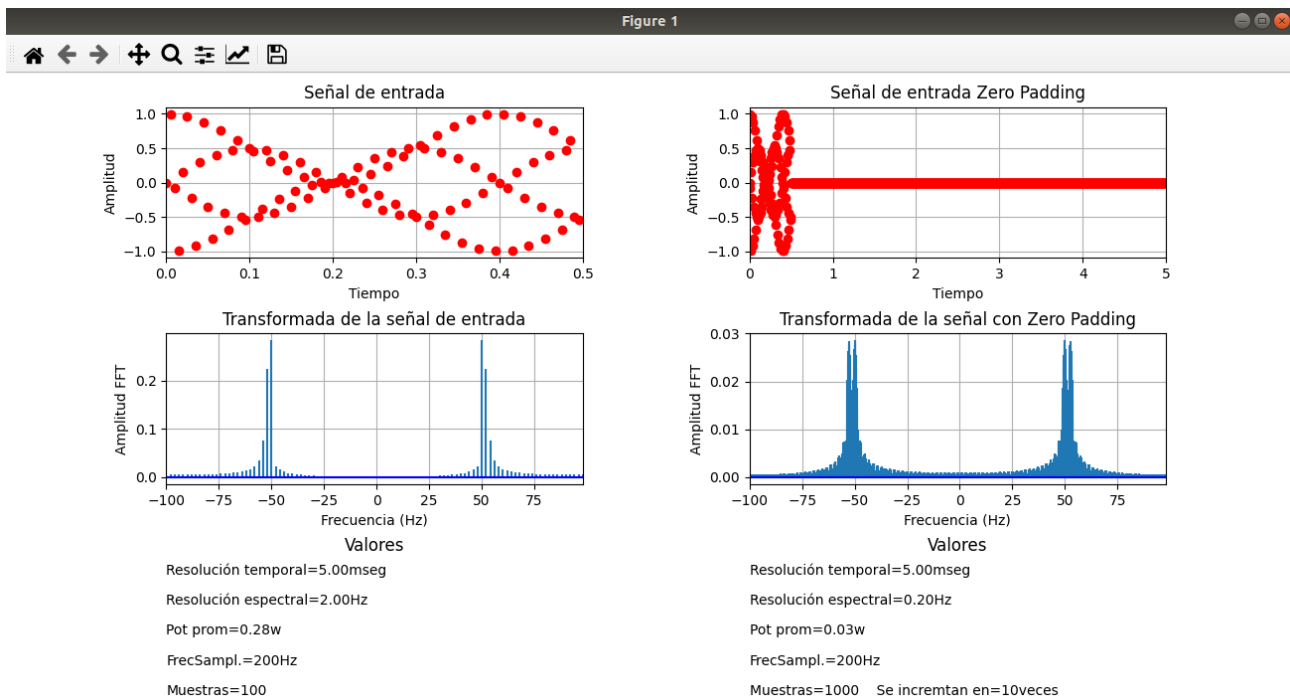
```

Para graficar se utiliza el siguiente código:

```
71 #-----Señal original-----
72 s1 = fig.add_subplot(3,2,1)
73 plt.title("Señal de entrada")
74 plt.xlabel("Tiempo")
75 plt.ylabel("Amplitud")
76 s1.grid(True)
77 plt.xlim(0,N/Fs)
78 s1.plot(ts,señal,'ro')
79
80 s5 = fig.add_subplot(3,2,3)
81 plt.title("Transformada de la señal de entrada")
82 plt.xlabel("Frecuencia (Hz)")
83 plt.ylabel("Amplitud FFT")
84 s5.grid(True)
85 plt.xlim(-Fs/2,Fs/2-Fs/N)
86 #s5.plot(F, M_fft)
87 s5.stem(F, M_fft, markerfmt=" ", basefmt="-b")
88
89
90 s9 = fig.add_subplot(3,2,5)
91 plt.title("Valores")
92 plt.xlim(0,10)
93 plt.ylim(0,10)
94 plt.axis('off')
95 s9.spines['right'].set_visible(False)
96 s9.spines['top'].set_visible(False)
97 s9.spines['bottom'].set_visible(False)
98 s9.spines['left'].set_visible(False)
99 plt.text(0,9,"Resolución temporal="+str(f'{Ts*1000:.{2}f}')+"mseg",fontsize=10)
100 plt.text(0,7,"Resolución espectral="+str(f'{Fs/N:.{2}f}')+"Hz",fontsize=10)
101 plt.text(0,5,"Pot prom="+str(f'{pot_promedio:.{2}f}')+"w",fontsize=10)
102 plt.text(0,3,"FrecSampl.="+str(Fs)+"Hz",fontsize=10)
103 plt.text(0,1,"Muestras="+str(N),fontsize=10)
104
```

El código para graficar la señal con zero padding es muy similar.

Y luego se grafican las señales:



Se observa que al aumentar la cantidad de muestras se tiene mejor resolución espectral pero la potencia baja proporcionalmente.