Processamento Digital de Sinais - PDS

LAB001 - Python em PDS:

Vamos usar a Linguagem Python para visualizar e compreender os conceitos de PDS

Bibliotecas de Python::

- Numpy e Scipy - https://scipy.org/
- Matplotlib - https://matplotlib.org/

Vamos representar os sinais no Python através de vetores (arrays):

Exemplo:

```
import numpy as np
n =  np.arange(0,11,1)
yD = np.sin(n*3.1415/8)
```

Cria um vetor *n* = [0,1,2,3,4,5,6,7,8,9,10] e um sinal senoidal discreto no tempo *yD*.

Para visualizar os sinais criados na forma de gráficos vamos usar a biblioteca Matplotlib.

Exemplo:

```
import numpy as np
from numpy import pi
import matplotlib.pyplot as plt

n = np.arange(11)
print(n)
yD1 = np.sin(n*pi/8)
yD2 = np.cos(n*pi/8)

#criando os gráficos
fig, ax = plt.subplots(2,1)

ax[0].stem(n, yD1, linefmt='b-')
ax[0].set_xlabel("Amostras")
ax[0].set_ylabel("Amplitude")
ax[0].grid(True)
ax[0].set_title('y[n] = sen[n*pi/8]')

ax[1].stem(n, yD2, linefmt='b-')
ax[1].set_xlabel("Amostras")
ax[1].set_ylabel("Amplitude")
ax[1].grid(True)
ax[1].set_title('y[n] = cos[n*pi/8]')

fig.tight_layout()
plt.show()
```

Podemos criar os sinais de outros modos:

```
import numpy as np
a = np.array( [2, 3, 4, 5.01, 7.304, 45, 334])
```

```
a
>> array([  2.   ,   3.   ,   4.   ,   5.01 ,   7.304,  45.   , 334.   ])
```

Exemplos para números complexos:

```
import numpy as np
a = numpy.array( [ [2,3], [4,5] ], dtype=complex)
c = numpy.array ( [2.+1.j, 2, 4 ] )
```

```
>> a
array([[2.+0.j, 3.+0.j],
 [4.+0.j, 5.+0.j]])
>> c
array([2.+1.j , 2.+0.j , 4.+0.j])
```

Exercícios:

1. Crie uma função em python que retorne a parte Par e a parte Impar de um sinal qualquer e teste nos sinais abaixo:
   a) s_1 = [2, 1, 0, 1, 2], n = [-2, -1, 0, 1, 2]
   b) s_2 = [-2,-1, 0, 1, 2], n = [-2, -1, 0, 1, 2]
   c) s_3 = [0, 0 ,0 , 2, 4], n = [-2, -1, 0, 1, 2]
   d) s_4 = [0 , -1 , -1, 3 , 2], n = [-2, -1, 0, 1, 2]
   e) s_5 = [0, 0 ,0 , 2, 4], n = [0, 1, 2, 3, 4]

   Modelo da função:
   xp,xi = separaParImpar(x,n)
   - Entrada:
     - x – sinal de entrada
     - n – vetor de tempo do sinal de entrada
   - Saída:
     - xp – parte par do sinal
     - xi – parte impar do sinal

Lembrando que

$$x[n]=x_{par}[n]+x_{ímpar}[n]$$

sendo:

$$x_{par}[n]=\frac{1}{2}\{x[n]+x[-n]\} \quad \text{e} \quad x_{ímpar}[n]=\frac{1}{2}\{x[n]-x[-n]\}$$

Processamento Digital de Sinais - PDS

2. Crie uma função para o sinal degrau:

   u = degrau[n,n_0]

   onde n_0 é o deslocamento no tempo da função

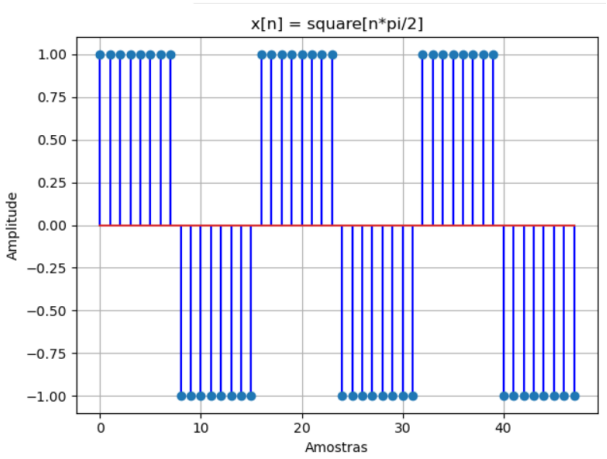
   e plote:

   a) $x[n] = u[n-1]$

   b) $x[n] = u[n]$

   c) $x[n] = u[n+1]$

3. Utilizando a função ***square*** da biblioteca "scipy.signal", crie um sinal onda quadrada igual da figura abaixo.



4. Crie e plote $x[n]$ no intervalo pedido:

   a) $x_1[n]=2\cos\left(\frac{\pi}{4}n\right)$, (intervalo de -10 a 10);

   b) $x[n]=0.5^n u(n)$, (intervalo de -10 a 10);

   c) $x[n]=2\left(0.8^{(n+2)}\right)u(n-2)$, (intervalo de -10 a 10);

   d) $x[n]=5*(-0.9)^n \cos\left(0.1\pi n\right)u(n)$, (intervalo de -10 a 10);

   e) $x[n]=\left(0.9e^{j\frac{\pi}{3}}\right)^n u(n)$, (intervalo de -10 a 10).