

1o Trabalho de PSD (notebook para preencher com os resultados)

Número do Grupo : inserir o número do grupo aqui

```
In [1]: import sk_dsp_comm.sigsys as ss
import numpy as np
from matplotlib import pylab
from pylab import *
import scipy.signal as signal
from IPython.display import Audio, display
from IPython.display import Image, SVG
```

```
In [2]: pylab.rcParams['savefig.dpi'] = 100 # default 72
#pylab.rcParams['figure.figsize'] = (6.0, 4.0) # default (6,4)
%config InlineBackend.figure_formats=['png'] # default for inline viewing
#config InlineBackend.figure_formats=['svg'] # SVG inline viewing
%config InlineBackend.figure_formats=['pdf'] # render pdf figs for LaTeX
```

Problema 1

Usando a função stem() em Python calcule e plote as seguintes funções

$$x_1[n] = \sin\left(\frac{\pi}{5}n\right), 0 \leq n \leq 15$$

$$x_2[n] = \cos\left(\frac{3\pi}{5}n\right), 0 \leq n \leq 15$$

$$x_3[n] = \sin\left(\frac{\pi}{4}n\right) \cos\left(\frac{\pi}{4}n\right), 0 \leq n \leq 32$$

Qual é o período da fundamental de cada sinal?

Parte a - sinal $x_1[n]$

O período da fundamental, N , é

adicione o resultado aqui e explicação de como ele foi determinado

In [3]: *# escreva o seu código aqui*

Parte b - sinal $x_2[n]$

O período da fundamental, N , é

adicione o resultado aqui e explicação de como ele foi determinado

In [4]: *# escreva o seu código aqui*

Parte c - sinal $x_3[n]$

Para encontrar o período, primeiro expanda o produto usando uma relação trigonométrica apropriada:

$$x_3[n] = \frac{1}{2} \left[\sin(0 \cdot n) + \sin\left(\frac{\pi}{2}n\right) \right], 0 \leq n \leq 32$$

Você deve escolher N tal que ...

adicione o resultado aqui e explicação de como ele foi determinado

In [5]: *# escreva o seu código aqui*

Problema 2

Usando a função `stem()` calcule e plote o sinal

$$x[n] = \begin{cases} 2n + 1, & -3 \leq n \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

no intervalo $-8 \leq n \leq 8$.

OBS: Criar a função $x[n]$ sem o uso de loop!! Observe ainda que o módulo python “sk_dsp_comm.sigsys” possui a função `direct()` que serve para criar uma função janela - estude o exemplo python `ex2.py` fornecido na aba “Material para realização dos projetos”.

In [6]: `# escreva o seu código aqui`

Problema 3

Usando a função `stem()` calcule e plote o sinal

$$x[n] = u[n]u[8 - n] + u[n - 2] - u[n - 5]$$

no intervalo $-2 \leq n \leq 16$. Observe que o módulo `sigsys` (aliased como `ss`) possui a função `dstep()` definida, a qual pode tornar esse problema bastante simples.

OBS: Estude os exemplos `ex1.py` e `ex2.py` fornecidos na aba “Material para realização dos projetos”.

In [7]: `# escreva o seu código aqui`

Problema 4

Considere o sinal

$$y[n] = \begin{cases} 2, & n = 0 \\ 1, & n = 1 \\ -1, & n = 3 \\ 3, & n = 4 \\ 0, & \text{caso contrário} \end{cases}$$

Escreva uma função python ($y[n]$) como indicado abaixo para que você possa responder as questões que seguem.

```
In [8]: def y(n):  
        """  
        Use a função dimpulse do módulo sigsys para diretamente definir  
        o sinal  $y[n]$   
        """  
        # # insira seu código aqui  
        return yy
```

Parte a

Usando a função `stem()` plote $z_1[n] = y[n - 2]$.

```
In [9]: # escreva o seu código aqui
```

Parte b

Usando a função `stem()` plote $z_2[n] = y[n + 2]$.

```
In [10]: # escreva o seu código aqui
```

Parte c

Usando a função `stem()` plote $z_3[n] = y[-n]$.

```
In [11]: # escreva o seu código aqui
```

Problema 5

Dado os sinais abaixo:

$$x_1[n] = \{1, 2, -3, 2, 1, 1\} \quad h_1[n] = \{1, 2, -3\}$$

$$x_2[n] = \{0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, -1, -1, -1, -1, 0, 0, 0, 0\} \quad h_2[n] = \{1, 4, 10, 16, 19, 16, 10, 4, 1\}$$

Parte a

Escrever uma função na linguagem python ("**sem usar funções prontas de qualquer biblioteca**") para fazer a **convolução** entre dois sinais unidimensionais (1D).

```
In [12]: def conv(x,h):  
         # insira seu código aqui  
         return y
```

Parte b

Faça a convolução entre os sinais $x_1 * h_1$ e $x_2 * h_2$ e plote usando a função `stem` os sinais originais e os resultados das convoluções.

```
In [13]: # escreva o seu código aqui
```

Parte c

De maneira similar à "Parte b" do exercício, usando **o mesmo código da função convolução**, faça a **correlação** entre os sinais e plote os resultados.

Dica, lembre-se que:

Convolução: $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k]$

Correlação: $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n + k]$

```
In [14]: # escreva o seu código aqui
```

Parte d

Utilize agora a função da biblioteca numpy (numpy.convolve) para realizar as mesmas convoluções da "Parte a". Use a função numpy.convolve - mode{'full'} e discuta os resultados.

```
In [15]: # escreva o seu código aqui
```