Atalho - Atualizado



Open in Colab

# Título: Execução dos Exercícios. Semestre 2024.1

## Entrege até

10/06/2024, 10:30

### Aluno(s):

 Vinícius José

### Professor:

 Diego Rátiva

# Base do Experimeto

 Livro

# Documentos - Resultado

 [Vídeo](Upar a playlist) - Atualizar  Repo

 Documento

 Colab - Atualizar! (feito)  Apresentação

# Sugestões do professor

Avaliação:

1. Compartilhe o código com [diego.rativa@ecomp.poli.br](mailto:diego.rativa@ecomp.poli.br)
2. Salve um vídeo por cada exercício de no máximo 2min explicando o exercício.
3. Salve o PDF do Colab e adicione na entrega desta tarefa.
4. Serão sorteados estudantes para apresentar os exercícios durante o horário da aula.

Explicação.

1. Não foque no código, (a disciplina não é de programação) e sim na metodología e os resultados na hora de explicar os fundamentos.
2. Não esqueça de fundamentar-se no livro.

# Execução

In [72]: *# Ambiente*

**!**pip install numpy

**!**pip install matplotlib

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

Requirement already satisfied: numpy in c:\users\vjfr\appdata\local\packages\pythonsoftware foundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (1.2

5.0)

Requirement already satisfied: matplotlib in c:\users\vjfr\appdata\local\packages\pythonsof twarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (3.8.2)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\vjfr\appdata\local\packages\pyt honsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-pa ckages (from matplotlib) (1.2.0)

Requirement already satisfied: cycler>=0.10 in c:\users\vjfr\appdata\local\packages\pythons oftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packag es (from matplotlib) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\vjfr\appdata\local\packages\py thonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-p ackages (from matplotlib) (4.45.0)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\vjfr\appdata\local\packages\py thonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-p ackages (from matplotlib) (1.4.5)

Requirement already satisfied: numpy<2,>=1.21 in c:\users\vjfr\appdata\local\packages\pytho nsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-pack ages (from matplotlib) (1.25.0)

Requirement already satisfied: packaging>=20.0 in c:\users\vjfr\appdata\local\packages\pyth onsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-pac kages (from matplotlib) (23.1)

Requirement already satisfied: pillow>=8 in c:\users\vjfr\appdata\local\packages\pythonsoft warefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from matplotlib) (9.5.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\vjfr\appdata\local\packages\pyt honsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-pa ckages (from matplotlib) (3.1.1)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\vjfr\appdata\local\packages

\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\sit e-packages (from matplotlib) (2.8.2)

Requirement already satisfied: six>=1.5 in c:\users\vjfr\appdata\local\packages\pythonsoftw arefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages

(from python-dateutil>=2.7->matplotlib) (1.16.0)

## Exercício 1

### Base Matemática

Utilização da Transformada de Fourier.

*N*−1

*X*[*f*] = ∑ *x*[*t*] ⋅ *e*

*N*

*n*=0

−*j* 2*π nk*

In [73]:

*# Execução: Parâmetros*

Ts **=** 1**/**64 T0 **=** 4

N0 **=** int(T0 **/** Ts)

*# Vetor de tempo*

t **=** np**.**arange(0, Ts **\*** N0, Ts)

*# Sinal g*

g **=** Ts **\*** np**.**exp(**-**2 **\*** t) g[0] **=** Ts **\*** 0.5

*# Transformada de Fourier*

G **=** np**.**fft**.**fft(g)

*# Conversão para coordenadas polares*

Gp **=** np**.**angle(G) Gm **=** np**.**abs(G)

*# Vetor k e frequência angular*

k **=** np**.**arange(N0)

w **=** 2 **\*** np**.**pi **\*** k **/** T0

In [74]:

*# Plotagem*

plt**.**figure(figsize**=**(10, 6))

*# Magnitude*

plt**.**subplot(211)

plt**.**stem(w[:32], Gm[:32]) plt**.**title('Magnitude')

plt**.**xlabel('Frequency (rad/s)')

plt**.**ylabel('Magnitude')

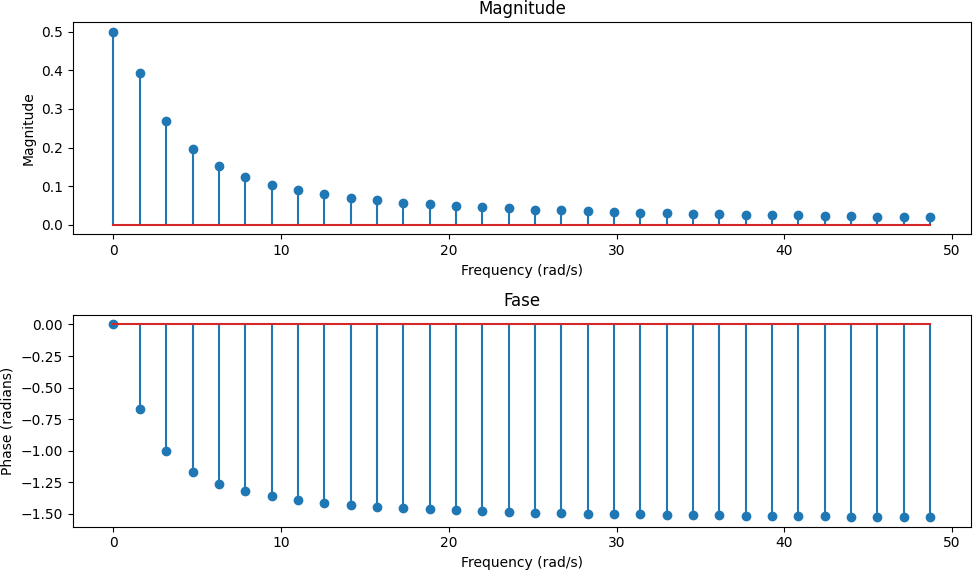
*# Fase*

plt**.**subplot(212)

plt**.**stem(w[:32], Gp[:32]) plt**.**title('Fase')

plt**.**xlabel('Frequency (rad/s)') plt**.**ylabel('Phase (radians)')

plt**.**tight\_layout() plt**.**show()



## Exercício 2

### Base Matemática

Utilização de conceitos de amostragem e DFT para calcular a DFT de um sinal discreto definido por partes.

*N*−1

*G*[*k*] = ∑ *g*[*n*] ⋅ *e*

*N*

*n*=0

−*j* 2*π nk*

In [75]: *# Execução: Parâmetros*

B **=** 4

f0 **=** 1**/**4

Ts **=** 1 **/** (2 **\*** B) T0 **=** 1 **/** f0

N0 **=** int(T0 **/** Ts)

*# Vetor k*

k **=** np**.**arange(N0 **+** 1)

*# Inicializar gk com zeros*

gk **=** np**.**zeros\_like(k)

**for** m **in** range(len(k)):

**if** 0 **<=** k[m] **<=** 3:

gk[m] **=** 1

**if** k[m] **==** 4 **or** k[m] **==** 28: gk[m] **=** 0.5

**if** 5 **<=** k[m] **<=** 27:

gk[m] **=** 0

**if** 29 **<=** k[m] **<=** 31:

gk[m] **=** 1

*# Transformada de Fourier*

Gr **=** np**.**fft**.**fft(gk)

In [76]:

*# Plotagem*

plt**.**figure(figsize**=**(10, 6))

*# Gráfico de gk* plt**.**subplot(211) plt**.**stem(k, gk) plt**.**title('gk')

plt**.**xlabel('k')

plt**.**ylabel('gk')

*# Gráfico da magnitude da FFT*

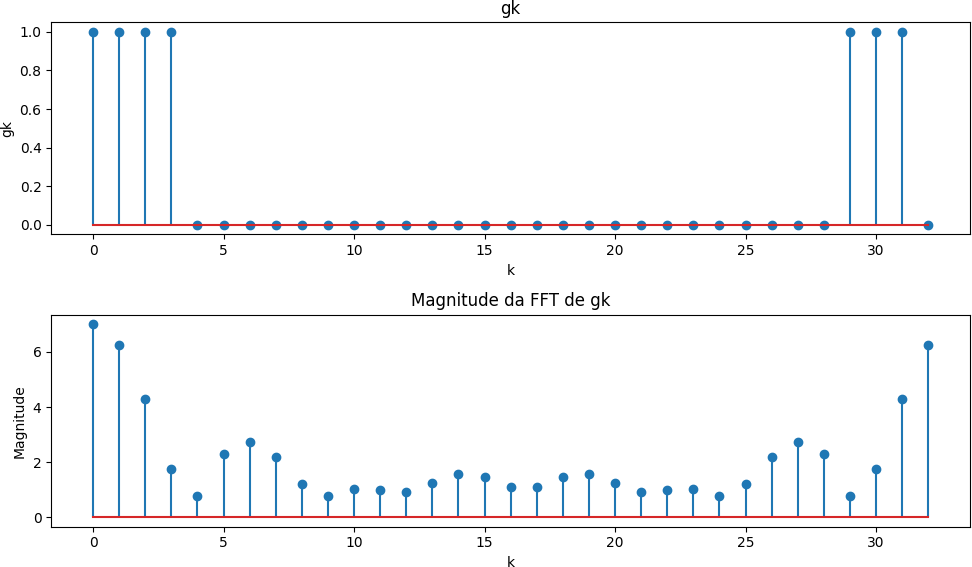
plt**.**subplot(212)

plt**.**stem(k, np**.**abs(Gr))

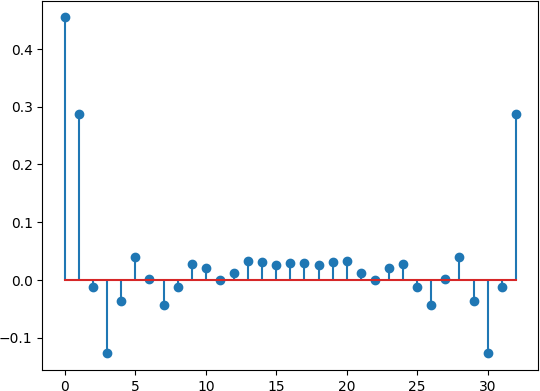
plt**.**title('Magnitude da FFT de gk') plt**.**xlabel('k')

plt**.**ylabel('Magnitude')

plt**.**tight\_layout() plt**.**show()



## Exercício 3



### Base matemática

Aplicação de conceitos de filtros no domínio da frequência.

*Y* [*k*] = *H*[*k*] ⋅ *G*[*k*]

In [77]:

*# Execução: Definições*

q **=** np**.**arange(33)

Hq **=** np**.**zeros\_like(q) *# Inicializar Hq com zeros*

**for** m **in** range(len(q)):

**if** 0 **<=** q[m] **<=** 7:

Hq[m] **=** 1

**if** 25 **<=** q[m] **<=** 31:

Hq[m] **=** 1

**if** 9 **<=** q[m] **<=** 23:

Hq[m] **=** 0

**if** q[m] **==** 8 **or** q[m] **==** 24: Hq[m] **=** 0.5

Gq **=** np**.**ones\_like(Hq) *# Inicializar Gq como exemplo*

Yq **=** Gq **\*** Hq

yk **=** np**.**fft**.**ifft(Yq)

In [78]:

*# Plot final*

plt**.**figure()

plt**.**stem(q, yk**.**real) *# Plota apenas a parte real de yk*

plt**.**show()