# INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

# CURSO DE GRADUAÇÃO EM ELETRÔNICA INDUSTRIAL

# DISCIPLINA: PROCESSADORES DE SINAIS DIGITAIS APLICADOS A ÁUDIO E VÍDEO

# Atividade Prática 04

# Aluno

# Marcelo Brancalhão Gaspar

# Professor

# Fernando Santana Pacheco

# Florianópolis, SC - 30 de abril de 2022

A. No Python (Numpy/Scipy), implemente com a função scipy.signal.lfilter um sistema média móvel de duas amostras. Considere a entrada x=[10,3,−3,3,8,1,−2]

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

print(y)

plt.plot(y,'-',y,'ok',x,'r',x,'\*k')

plt.show()

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

B. Verifique se a saída está correta (faça o cálculo manualmente). Foi por esse motivo que usamos um sinal x(n)x(n) simples, para calcular a saída manualmente e ter certeza que a implementação está correta.

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

y(0) = 0.5(x(0)+x(-1))= 0.5(10+0) = 5

y(1) = 0.5(x(1)+x(0)) = 0.5(3+10) = 6.5

y(2) = 0.5(x(2)+x(1)) = 0.5(-3+3) = 0

y(3) = 0.5(x(3)+x(2)) = 0.5( 3-3) = 0

y(4) = 0.5(x(4)+x(3)) = 0.5( 8+3) = 5.5

y(5) = 0.5(x(5)+x(4)) = 0.5( 8+1) = 4.5

y(6) = 0.5(x(6)+x(5)) = 0.5(-2+1) = -0.5

C. Verifique com a função scipy.signal.freqz a resposta em frequência do sistema. Que tipo de filtro é o sistema média móvel? Qual a sua frequência de corte, em Hz, considerando uma frequência de amostragem de 44100 Hz?

Confira se o resultado está correto fazendo também de forma analítica. Anexe uma foto com seu cálculo. Para isso, lembre que a frequência de corte de um filtro é aquela em que a magnitude de H(z)H(z) é 12√12. Você já tem a resposta em frequência (olhe no vídeo). Obtenha a magnitude, depois iguale a 12√12 e isole a frequência.

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

f2w = 2\*np.pi

fs,h = signal.freqz(b, fs=f2w\*44100)

fig, ax = plt.subplots()

ax.set\_title('Resposta em frequência')

ax.plot(fs/(f2w), 20\*np.log10(abs(h)))

ax.set\_ylabel('Amplitude [dB]')

ax.set\_xlabel('Frequency [Hz]')

plt.grid()

plt.show()

Texto, Carta

Descrição gerada automaticamente

Gráfico, Gráfico de linhas, Gráfico de dispersão

Descrição gerada automaticamente

D. Um tom senoidal puro de 17 kHz, amplitude de pico de 1,5 V, amostrado a 44,1 kHz, é aplicado à entrada do sistema média móvel de 2 amostras. Qual o sinal de saída (tipo, frequência e amplitude)? Mostre.

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

f2w = 2\*np.pi

fs,h = signal.freqz(b, fs=f2w\*44100)

k = np.linspace(0,1,44100)

tom = 1.5\*(np.sin(f2w\*17000\*k))

plt.xlim(0,0.0001)

plt.plot(k,tom,'\*r')

plt.plot(k,tom,'--k')

plt.xlabel('Angle [rad]')

plt.ylabel('sin(k)')

plt.show()

y = signal.lfilter(b,a,tom)

ffty = np.fft.fft(y)

n = ffty.size

freq = np.fft.fftfreq(n, 1/44100)

plt.xlim(0,20000)

plt.plot(freq,abs(ffty)\*2/44100)

plt.show()

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente

E. Com zplane (baixar de <https://www.dsprelated.com/showcode/244.php>), onde estão localizadas as singularidades (polos e zeros) no domínio Z do sistema média móvel de 2 amostras?

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

f2w = 2\*np.pi

fs,h = signal.freqz(b, fs=f2w\*44100)

k = np.linspace(0,1,44100)

tom = 1.5\*(np.sin(f2w\*17000\*k))

y = signal.lfilter(b,a,tom)

ffty = np.fft.fft(y)

n = ffty.size

freq = np.fft.fftfreq(n, 1/44100)

plt.xlim(0,20000)

plt.plot(freq,abs(ffty)\*2/44100)

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import patches

from matplotlib.figure import Figure

from matplotlib import rcParams

def zplane(b,a,filename=None):

"""Plot the complex z-plane given a transfer function.

"""

# get a figure/plot

ax = plt.subplot(111)

# create the unit circle

uc = patches.Circle((0,0), radius=1, fill=False,

color='black', ls='dashed')

ax.add\_patch(uc)

# The coefficients are less than 1, normalize the coeficients

if np.max(b) > 1:

kn = np.max(b)

b = b/float(kn)

else:

kn = 1

if np.max(a) > 1:

kd = np.max(a)

a = a/float(kd)

else:

kd =1

# Get the poles and zeros

p = np.roots(a)

z = np.roots(b)

k = kn/float(kd)

# Plot the zeros and set marker properties

t1 = plt.plot(z.real, z.imag, 'go', ms=10)

plt.setp( t1, markersize=10.0, markeredgewidth=1.0,

markeredgecolor='k', markerfacecolor='g')

# Plot the poles and set marker properties

t2 = plt.plot(p.real, p.imag, 'rx', ms=10)

plt.setp( t2, markersize=12.0, markeredgewidth=3.0,

markeredgecolor='r', markerfacecolor='r')

ax.spines['left'].set\_position('center')

ax.spines['bottom'].set\_position('center')

ax.spines['right'].set\_visible(False)

ax.spines['top'].set\_visible(False)

# set the ticks

r = 1.5; plt.axis('scaled'); plt.axis([-r, r, -r, r])

ticks = [-1, -.5, .5, 1]; plt.xticks(ticks); plt.yticks(ticks)

if filename is None:

plt.show()

else:

plt.savefig(filename)

return z, p, k

zplane(b,a)

Diagrama

Descrição gerada automaticamente

F. Ainda no Python, implemente uma média móvel com quatro amostras. Há um efeito de suavização na saída?

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

print('x = ', array)

b4 = [0.25,0.25,0.25,0.25]

a4 = [1]

y4 = signal.lfilter(b4,a4,array)

print('y2 = ', y)

print('y4 = ', y4)

plt.plot(x,'r',x,'or',y4,'--k',y4,'ok')

plt.show()

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

G. Verifique a resposta em frequência do sistema com freqz. Qual a diferença em relação ao sistema de 2 amostras?

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

import matplotlib.pyplot as plt

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

f2w = 2\*np.pi

fs,h = signal.freqz(b, fs=f2w\*44100)

fig, ax = plt.subplots()

ax.set\_title('Resposta em frequência')

ax.plot(fs/(f2w), 20\*np.log10(abs(h)))

ax.set\_ylabel('Amplitude [dB]')

ax.set\_xlabel('Frequency [Hz]')

plt.grid()

print('x = ', array)

b4 = [0.25,0.25,0.25,0.25]

a4 = [1]

y4 = signal.lfilter(b4,a4,array)

print('y2 = ', y)

print('y4 = ', y4)

plt.plot(x,'r',x,'or',y4,'--k',y4,'ok')

f2w=2\*np.pi

fs4,h4 = signal.freqz(b4, fs=f2w\*44100)

fig, ax = plt.subplots()

ax.set\_title('Resposta em frequência')

ax.plot(fs4/(f2w), 20 \* np.log10(abs(h4)))

ax.plot(fs/(f2w),20\*np.log10(abs(h)))

ax.set\_ylabel('Amplitude [dB]')

ax.set\_xlabel('Frequency [Hz]')

plt.grid()

plt.show()

Gráfico

Descrição gerada automaticamente com confiança média

Gráfico, Gráfico de linhas, Gráfico de dispersão

Descrição gerada automaticamente

H. Gere um sinal sweep no Ocenaudio (ou gere diretamente no Python) e grave em formato wav. Leia o arquivo no Python e use como entrada da média móvel com 4 amostras. Grave a saída, abra no Ocenaudio e observe a saída no tempo e no domínio da frequência. O resultado é condizente com o esperado no item G? Mostre.

# Código e resultados;

######################################################

## Created on: Abril 26, 2022

## Author: Marcelo Brancalhão Gaspar

## Instituto Federal de Santa Catarina

## DSP 2 - Fernando Santana Pacheco

##

######################################################

import numpy as np

from scipy import signal

from scipy.io import wavfile

import matplotlib.pyplot as plt

from scipy.signal import chirp

x = [10,3,-3,3,8,1,-2]

array = np.array(x)

a = [1]

b = [0.5,0.5]

y = signal.lfilter(b, a, array)

print('x = ', array)

b4 = [0.25,0.25,0.25,0.25]

a4 = [1]

y4 = signal.lfilter(b4,a4,array)

print('y2 = ', y)

print('y4 = ', y4)

fs = 44100

t = np.linspace(0,1,fs)

sweep = chirp(t,100,1,1000)

plt.plot(t,sweep)

plt.show()

wavfile.write("sweep.wav",fs , sweep)

filtered=signal.lfilter(b4,a4,sweep)

wavfile.write("sweepfilter.wav",fs , filtered)

Gráfico

Descrição gerada automaticamente