# Relatório – ST7: T7: Projeto de Filtro FIR Passa-Baixas com Janela de Hamming

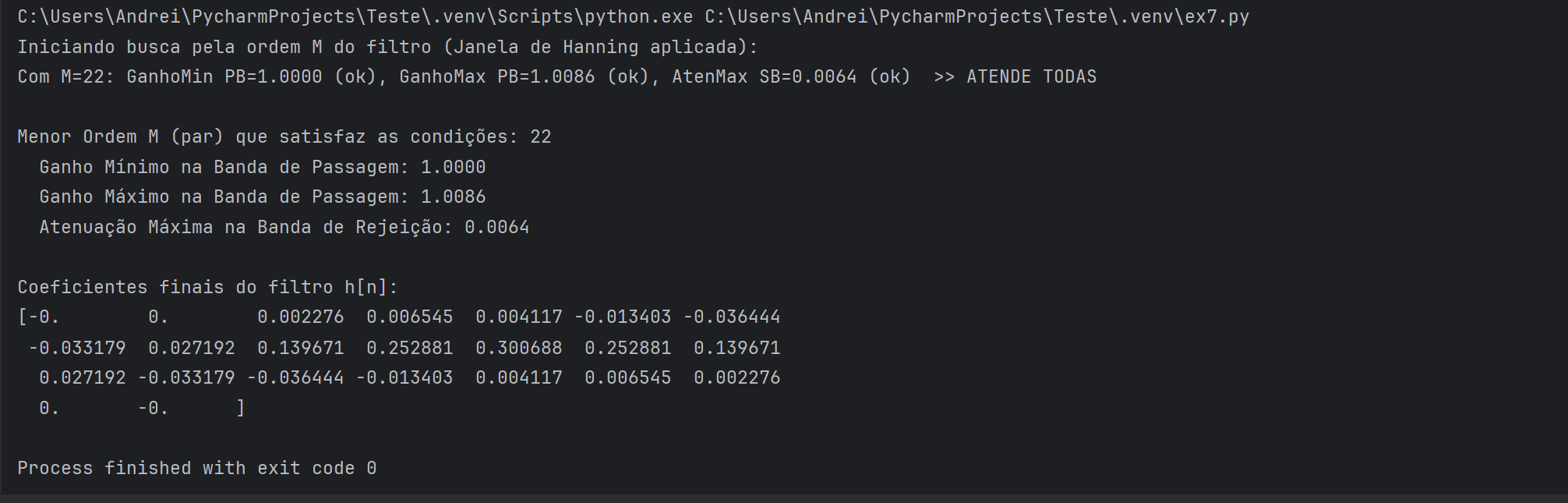
## 1. Enunciado

## 

## 2. Código Python

import numpy as np  
import matplotlib.pyplot as plt  
  
freq\_pass\_lim = 0.15 \* np.pi  
freq\_stop\_lim = 0.45 \* np.pi  
transition\_width\_rad = freq\_stop\_lim - freq\_pass\_lim  
num\_freq\_points = 1024  
angular\_frequencies = np.linspace(0, np.pi, num\_freq\_points)  
  
  
def criar\_filtro\_janelado\_hanning(ordem\_filtro):  
 indices\_tempo = np.arange(ordem\_filtro + 1)  
 ponto\_central = ordem\_filtro / 2  
 freq\_corte\_ideal = (freq\_pass\_lim + freq\_stop\_lim) / 2  
 sinc\_ideal = np.sinc((freq\_corte\_ideal / np.pi) \* (indices\_tempo - ponto\_central))  
 janela\_hanning = 0.5 - 0.5 \* np.cos(2 \* np.pi \* indices\_tempo / ordem\_filtro)  
 coeficientes\_janelados = sinc\_ideal \* janela\_hanning  
 coeficientes\_normalizados = coeficientes\_janelados / np.sum(coeficientes\_janelados)  
 return coeficientes\_normalizados, indices\_tempo  
  
  
def verificar\_especificacoes\_filtro(coefs\_filtro, indices\_n\_filtro):  
 resposta\_freq\_H = np.array(  
 [np.sum(coefs\_filtro \* np.exp(-1j \* omega\_i \* indices\_n\_filtro)) for omega\_i in angular\_frequencies])  
 magnitude\_H = np.abs(resposta\_freq\_H)  
 ganho\_min\_passabanda = magnitude\_H[angular\_frequencies <= freq\_pass\_lim].min()  
 ganho\_max\_passabanda = magnitude\_H[angular\_frequencies <= freq\_pass\_lim].max()  
 atenuacao\_max\_rejeicao = magnitude\_H[angular\_frequencies >= freq\_stop\_lim].max()  
 return ganho\_min\_passabanda, atenuacao\_max\_rejeicao, ganho\_max\_passabanda  
  
  
ordem\_atual\_M = int(np.ceil(3.1 \* 2 \* np.pi / transition\_width\_rad))  
if ordem\_atual\_M % 2:  
 ordem\_atual\_M += 1  
  
print("Iniciando busca pela ordem M do filtro (Janela de Hanning aplicada):")  
while True:  
 coeficientes\_finais, indices\_n = criar\_filtro\_janelado\_hanning(ordem\_atual\_M)  
 min\_pb, max\_sb, max\_pb = verificar\_especificacoes\_filtro(coeficientes\_finais, indices\_n)  
  
 atende\_min\_pb = min\_pb >= 0.99  
 atende\_max\_pb = max\_pb <= 1.01  
 atende\_max\_sb = max\_sb <= 0.06  
 todas\_atendidas = atende\_min\_pb and atende\_max\_pb and atende\_max\_sb  
  
 print(f"Com M={ordem\_atual\_M}: GanhoMin PB={min\_pb:.4f} ({'ok' if atende\_min\_pb else 'X'}), "  
 f"GanhoMax PB={max\_pb:.4f} ({'ok' if atende\_max\_pb else 'X'}), "  
 f"AtenMax SB={max\_sb:.4f} ({'ok' if atende\_max\_sb else 'X'}) "  
 f">> {'ATENDE TODAS' if todas\_atendidas else 'NÃO ATENDE'}")  
  
 if todas\_atendidas:  
 print(f"\nMenor Ordem M (par) que satisfaz as condições: {ordem\_atual\_M}")  
 print(f" Ganho Mínimo na Banda de Passagem: {min\_pb:.4f}")  
 print(f" Ganho Máximo na Banda de Passagem: {max\_pb:.4f}")  
 print(f" Atenuação Máxima na Banda de Rejeição: {max\_sb:.4f}")  
 break  
 ordem\_atual\_M += 2  
  
print("\nCoeficientes finais do filtro h[n]:")  
print(np.round(coeficientes\_finais, 6))  
  
resposta\_H\_final = np.array(  
 [np.sum(coeficientes\_finais \* np.exp(-1j \* omega\_i \* indices\_n)) for omega\_i in angular\_frequencies])  
frequencias\_plot = angular\_frequencies / np.pi  
  
plt.figure(figsize=(10, 9))  
  
plt.subplot(3, 1, 1)  
plt.stem(indices\_n, coeficientes\_finais, basefmt=" ")  
plt.title(f"Coeficientes do Filtro FIR (M={ordem\_atual\_M}, Janela de Hanning)")  
plt.xlabel("Índice n")  
plt.ylabel("h[n]")  
plt.grid(True)  
  
plt.subplot(3, 1, 2)  
plt.plot(frequencias\_plot, np.abs(resposta\_H\_final), color='dodgerblue')  
plt.axvline(freq\_pass\_lim / np.pi, color='green', linestyle='--', label='Borda Passa-Faixa')  
plt.axvline(freq\_stop\_lim / np.pi, color='red', linestyle='--', label='Borda Rejeita-Faixa')  
plt.hlines([0.99, 1.01], 0, freq\_pass\_lim / np.pi, color='lightgreen', linestyle=':')  
plt.hlines([0.06], freq\_stop\_lim / np.pi, 1, color='lightcoral', linestyle=':')  
plt.title("Resposta em Magnitude |H(e^{jω})|")  
plt.xlabel("Frequência Normalizada (×π rad/amostra)")  
plt.ylabel("Magnitude |H|")  
plt.ylim(-0.05, 1.15)  
plt.legend()  
plt.grid(True)  
  
plt.subplot(3, 1, 3)  
plt.plot(frequencias\_plot, 20 \* np.log10(np.abs(resposta\_H\_final) + 1e-9), color='darkorange')  
plt.axvline(freq\_pass\_lim / np.pi, color='green', linestyle='--', label='Borda Passa-Faixa')  
plt.axvline(freq\_stop\_lim / np.pi, color='red', linestyle='--', label='Borda Rejeita-Faixa')  
plt.hlines([20 \* np.log10(0.99), 20 \* np.log10(1.01)], 0, freq\_pass\_lim / np.pi, color='lightgreen', linestyle=':',  
 label='Tolerância Passa-Faixa')  
plt.hlines([20 \* np.log10(0.06)], freq\_stop\_lim / np.pi, 1, color='lightcoral', linestyle=':',  
 label='Tolerância Rejeita-Faixa')  
plt.title("Resposta em Magnitude em dB |H(e^{jω})|")  
plt.xlabel("Frequência Normalizada (×π rad/amostra)")  
plt.ylabel("Magnitude |H| (dB)")  
plt.ylim(-80, 5)  
plt.legend()  
plt.grid(True)  
  
plt.tight\_layout()  
plt.show()

## 3. Resultato



## 4. Gráfico da Resposta em Frequência

