EXPERIMENT 10 – IIR FILTER DESIGN:

CODE:

"""

Created on Tue May 14 10:53:08 2024

@author: cb.en.u4ece22151

"""

import numpy as np

import matplotlib.pyplot as plt

import scipy.signal as signal

def butfilter(N,wc,analog=True):

x=np.arange(1,N+1)

o=(2\*x-1)\*np.pi/(2\*N)

pole=wc\*np.exp(1j\*o)

y=np.poly(pole.real)

x=np.prod(-pole)

c=[x.real]

d=np.real(y)

return c,d

wp=(np.pi)/2

ws=(3\*np.pi)/4

Ap=3.0116

As=13.9794

N=2 #from manual calculation

wc=1.9997 #analog CUTOFF

op=2 #ANALOG PASS

os=4.828 #ANALOG STOP

#computed from given inputs

Nc=np.log10((((10\*\*(0.1\*As))-1)/((10\*\*(0.1\*Ap))-1))\*\*0.5)

Nc2=(Nc/np.log(ws/wp))

N1=np.ceil(Nc2)

print("The order of the filter is:",N1)

Wc=op/((10\*\*(0.1\*Ap)-1)\*\*(1/(2\*N)))

print("The cutoff frequency of the LP filter is:",Wc)

c1,d1=signal.butter(N1,Wc,'low',analog="True")

c,d=butfilter(N1,Wc,analog="True")

plt.figure(1)

plt.grid()

plt.xlabel("s")

plt.ylabel("Amplitude")

plt.title("H(s) plot - Analog butterworth Butterworth filter")

WS,HS=signal.freqs(c,d)

plt.xticks(np.arange(0,11))

plt.plot(WS,(abs(HS)),color='r')

b,a=signal.bilinear(c,d,fs=1)

b1,a1=signal.bilinear(c1,d1,fs=1)

wz,hz=signal.freqz(b,a)

wz1,hz1=signal.freqz(b1,a1)

plt.figure(2)

plt.grid()

plt.xlabel("Frequency")

plt.ylabel("Amplitude")

plt.title("H(f) plot - Digital butterworth Butterworth filter by Bilinear ")

plt.semilogy(wz,(abs(hz)),label='Manual code filter response',color='r')

plt.semilogy(wz1,(abs(hz1)),label='Built in code filter response',color='b')

plt.axvline(wp,linestyle='--')

plt.axvline(ws,linestyle='--')

plt.axhline(0.707,linestyle='--')

plt.axhline(0.2,linestyle='--')

plt.legend()

plt.figure(3)

plt.grid()

plt.xlabel("Frequency")

plt.ylabel("Amplitude")

plt.title("H(f) plot - Digital butterworth Butterworth filter by Bilinear ")

plt.semilogx(wz,20\*np.log10(abs(hz)),label='Manual code filter response',color='r')

plt.semilogx(wz1,20\*np.log10(abs(hz1)),label='Built in code filter response',color='b')

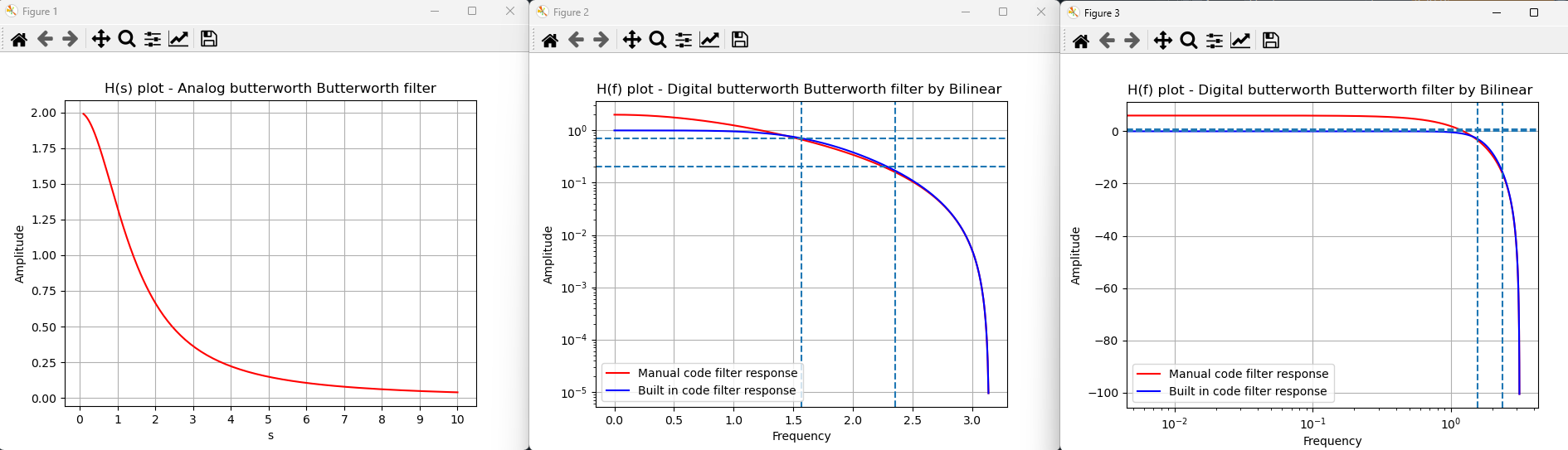
plt.axvline(wp,linestyle='--')

plt.axvline(ws,linestyle='--')

plt.axhline(0.707,linestyle='--')

plt.axhline(0.2,linestyle='--')

plt.legend()

OUTPUT:

