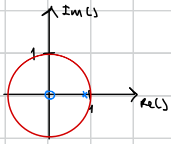
Assignment 4

Johnny Ngo Nguyen

P1 Given $H(z) = \frac{1}{1-az^{-1}}$

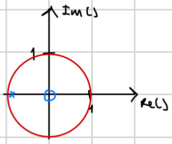
(a) $a = 0.9$: $H(z) = \frac{1}{1-0.9z^{-1}}$, $p = 0.9$, $z = 0$



→ LP as pole close to 1

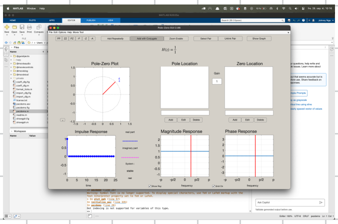
zero "close" to $z = -1$

$a = -0.9$: $H(z) = \frac{1}{1+0.9z^{-1}}$, $p = -0.9$, $z = 0$



→ HP as pole close to -1

zero close (!!) to $z = 0$

(b) 

Result when running pezdemo.

P2 Given causal filter with $H(z) = \frac{1}{(1-\frac{1}{2}z^{-1})(1+\frac{1}{2}z^{-1})}$

(a) Want to find $H_I(z)$ such that $H(z)\,H_I(z) = 1$

$H_I(z) = \frac{1}{H(z)} = (1-\frac{1}{2}z^{-1})(1+\frac{1}{2}z^{-1}) = \underline{\underline{1 - \frac{1}{4}z^{-2}}}$

(b) We have $p_1 = p_2 = 0$. Causal filter → ROC exterior, $|z| > |p_{max}|$ → unit circle included in ROC, $\underline{H_I(z) \text{ stable}}$

Can also see we have FIR filter as transfer function is on the form $H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}$ (finite length).

While IIR are often on the form $H(z) = \frac{B(z)}{A(z)}$.

<u>FIR filters are stable</u>

(c) A system is called minimum phase if all zeros and poles are inside unit circle.

⇒ a stable pole-zero system that is minimum phase has a stable invers that is also minimum phase.

→ $H(z)$ has poles and zeros in unit circle, thus it is minimum phase. Stable as ROC contains unit circle.

→ $H_I(z)$ has poles and zeros in unit circ, — " — . Stable as it is FIR filter.

(d) Linear phase filters are on the form $H(\omega) = |H(\omega)|e^{j\phi(\omega)}$, $\phi(\omega) = \alpha - D\omega$.

Linear phase filter have zeros occur in reciprocal pairs – for a zero in z_k, there is a zero in $\frac{1}{z_k}$ (comes from symmetry / anti-sym. of impulse response).

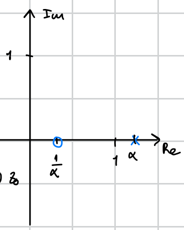
For $H_I(z)$: $z_1 = \frac{1}{2}$ and $z_2 = -\frac{1}{2}$, but no $z_3 = 2$ and $z_4 = -2$.

→ <u>$H_I(z)$ does not have linear phase characteristics</u>

P3 Low-freq. shelving filter, $A(z) = \frac{\alpha - z^{-1}}{1-\alpha z^{-1}}$, used to alter characteristics of sound at different frequencies.

(a) $z = \frac{1}{\alpha}$ and $p = \alpha$

→ pole and zero reciprocal



S. 4.44: $H(z) = z^{-N}\frac{A(z^{-1})}{A(z)}$

is an all-pass, reciprocal pole/zero &

→ <u>All-pass filter</u>

Could also calculate $|A(z)| = \left|\frac{\alpha - z^{-1}}{1-\alpha z^{-1}}\right|$ and see that it equals 1. $|A(z)| = 1$.

(b) $H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \ldots b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}}$, in our case $b_n = \{\alpha, -1\}$ and $a_n = \{1, -\alpha\}$

We have two branch that sums to $H(z) = H_u(z) + H_u(z)$ where $H_u = \frac{1}{2}(1 + A(z))$ and $H_u = \frac{K}{2}(1 - A(z))$.

$$- H_u(z) = \frac{1}{2}\left(1 + \frac{\alpha - z^{-1}}{1-\alpha z^{-1}}\right)$$

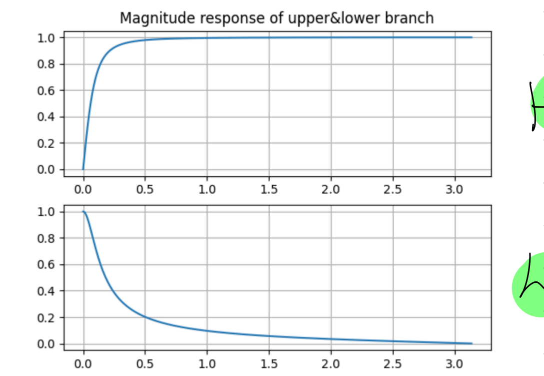
$$= \frac{1}{2}\left(\frac{1 - \alpha z^{-1} + \alpha - z^{-1}}{1 - \alpha z^{-1}}\right), \quad \alpha = 0.9$$

$$= \underline{\frac{1.9 - 1.9z^{-1}}{2 - 1.8z^{-1}}}$$

$$-H_L(z) = \frac{K}{2}\left(1 - \frac{\alpha - z^{-1}}{1-\alpha z^{-1}}\right)$$
$$= \frac{K}{2}\left(\frac{1-\alpha z^{-1} - \alpha + z^{-1}}{1-\alpha z^{-1}}\right), \quad \alpha = 0.9$$
$$= \frac{K}{2}\left(\frac{0.1+0.1z^{-1}}{1-0.9z^{-1}}\right), \quad K = 1$$
$$= \frac{0.1+0.1z^{-1}}{2-1.8z^{-1}}$$

A4P3.py > ...

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp

alpha = 0.9
K = 1
# H(z) = B(z)/A(z)
b_u = [1.9, -1.9]
a_u = [2, -1.8]
b_l = [0.1, 0.1]
a_l = [2, -1.8]

# scipy.freqz calulates freq. response of digital filter given coefficients
# returns w (discrete sample freqs. from [0,pi])
# and h (freq.resp. as complex numbers, evaluated with freqs. from w)
w_u, h_u = sp.signal.freqz(b_u, a_u)
w_l, h_l = sp.signal.freqz(b_l, a_l)
# print(f"w = {w}")
# print(f"h = {h}")

fig, (ax1, ax2) = plt.subplots(2)

ax1.set_title("Magnitude response of upper&lower branch")
ax1.plot(w_u, abs(h_u))
ax1.grid()

ax2.plot(w_l, abs(h_l))
ax2.grid()

plt.show()
```



HP

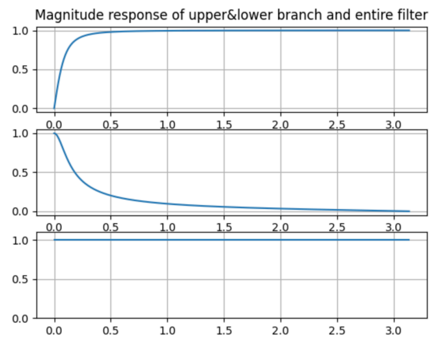
LP

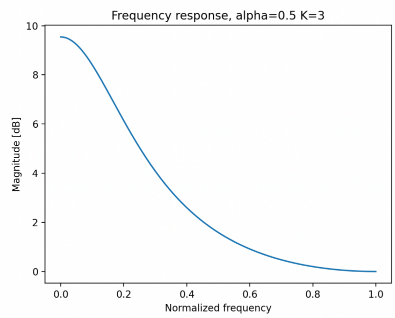
c

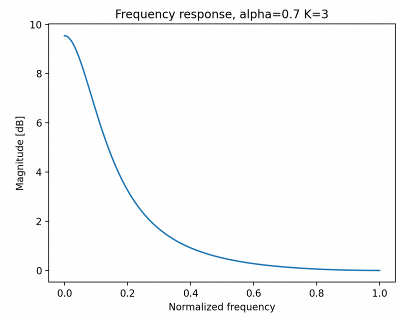
LFshelving.py M | A1.ipynb | A4P3.py 1, M

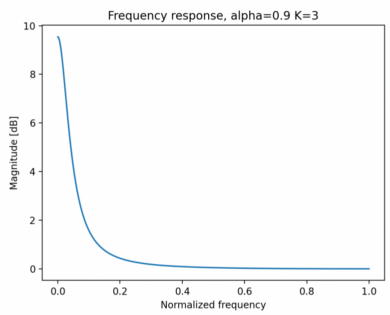
A4P3.py > ...

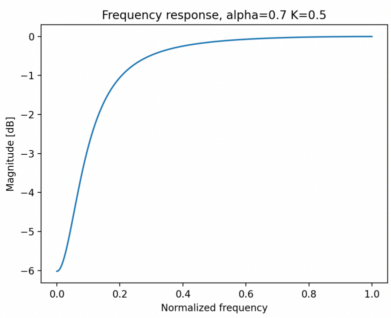
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp

alpha = 0.9
K = 1
# H(z) = B(z)/A(z)
b_u = [1.9, -1.9]
a_u = [2, -1.8]
b_l = [0.1, 0.1]
a_l = [2, -1.8]

# scipy.freqz calulates freq. response of digital filter given coefficients
# returns w (discrete sample freqs. from [0,pi])
# and h (freq.resp. as complex numbers, evaluated with freqs. from w)
w_u, h_u = sp.signal.freqz(b_u, a_u)
w_l, h_l = sp.signal.freqz(b_l, a_l)
w = w_u
h = h_u + h_l
# print(f"w = {w}")
# print(f"h = {h}")

fig, (ax1, ax2, ax3) = plt.subplots(3)

ax1.set_title("Magnitude response of upper&lower branch and entire filter")
ax1.plot(w_u, abs(h_u))
ax1.grid()

ax2.plot(w_l, abs(h_l))
ax2.grid()

ax3.plot(w, abs(h))
ax3.set_ylim(0, 1.1)
ax3.grid()

plt.savefig("magnitude_responses.png")
plt.show()
```

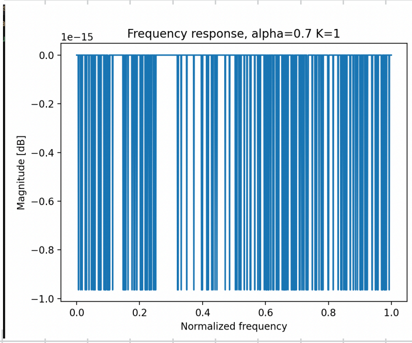


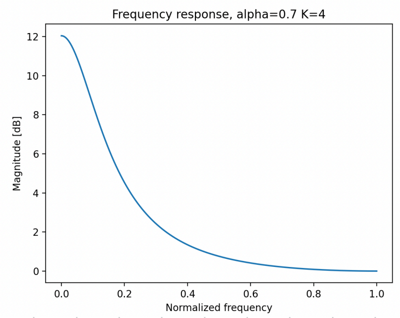












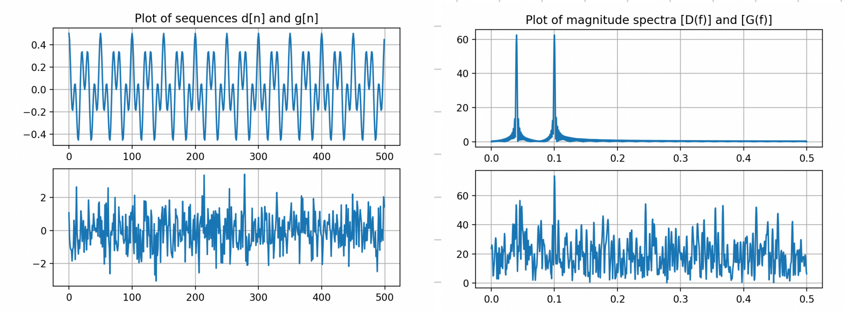
K controls how much contribution we get from LP-filter. Higher K-value results in more contribution.
α controls the bandwidth

P4 $d[n] = A_x \cos(2\pi f_x n) + A_y \cos(2\pi f_y n), \quad 0 \le n \le L-1$

$A_x = A_y = 0.25, \quad f_x = 0.10, \quad L = 500$

Noise $e[n]$: $g[n] = d[n] + e[n]$.

(a)



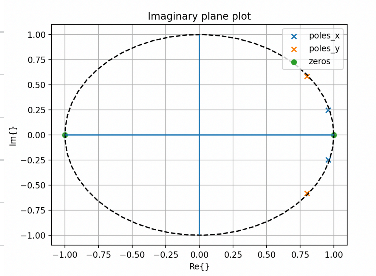
(b) The poles should be as close to the unit circle as possible so $|r| \to 1$. in $H(z) = \dfrac{b_0}{(1 - re^{j\omega_0}z^{-1})(1 - re^{-j\omega_0}z^{-1})}$, let $r = 0.99$

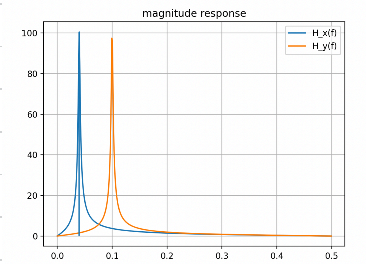
Resonant peak $\omega_r = \arccos\left(\frac{1+r^2}{2r}\cos(\omega_0)\right)$

$- H_x(z) = \dfrac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j\omega_{x0}}z^{-1})(1-0.99e^{-j\omega_{x0}}z^{-1})}, \quad \omega_{x0} = 2\pi f_{x0}, \quad$ den.: $[1,\ 1 - 0.99e^{-()} - 0.99e^{()},\ 0.99^2 e^0]$

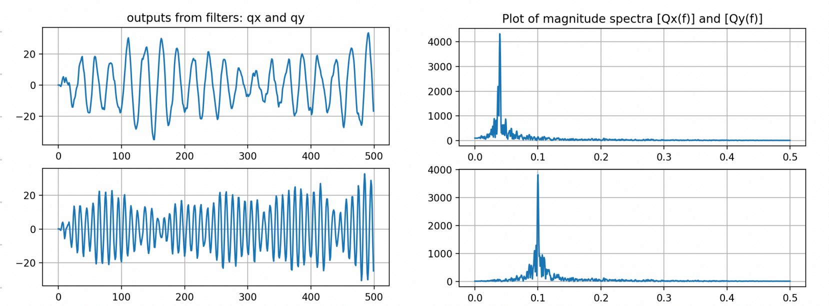
$- H_y(z) = \dfrac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j\omega_{y0}}z^{-1})(1-0.99e^{-j\omega_{y0}}z^{-1})}, \quad \omega_{y0} = 2\pi f_{y0}$

```python
# PROBLEM 4C

b = [-1, 0, 1]
a_x = [1, -0.99*np.exp(-1j*2*np.pi*fx)-0.99*np.exp(1j*2*np.pi*fx), 0.99*0.99]
a_y = [1, -0.99*np.exp(-1j*2*np.pi*fy)-0.99*np.exp(1j*2*np.pi*fy), 0.99*0.99]

zeros = np.roots(b)
poles_x = np.roots(a_x)
poles_y = np.roots(a_y)
theta = np.linspace(0, 2*np.pi)

fig, ax = plt.subplots()
ax.plot(np.cos(theta), np.sin(theta), '--k')
ax.scatter(np.real(poles_x), np.imag(poles_x), marker='x', label='poles_x')
ax.scatter(np.real(poles_y), np.imag(poles_y), marker='x', label='poles_y')
ax.scatter(np.real(zeros), np.imag(zeros), marker='o', label='zeros')
ax.set_title("Imaginary plane plot")
ax.set_ylabel("Im{}")
ax.set_xlabel("Re{}")
ax.hlines(0, -1, 1)
ax.vlines(0, -1, 1)
ax.legend(loc='upper right')
ax.grid()

w_x, h_x = sp.signal.freqz(b, a_x)
w_y, h_y = sp.signal.freqz(b, a_y)

fig, ax1 = plt.subplots()
ax1.plot(w_x, abs(h_x), label=r'H_x(f)')
ax1.plot(w_y, abs(h_y), label=r'H_y(f)')
ax1.set_title("magnitude response")
ax1.legend()
ax1.grid()

plt.show()
```





(c)



Comparing to earlier plots, alot of noise have been removed.

Magnitude response confirms that filters are implemented correctly.

(d) Combining filters to: $H(z) = H_x(z) + H_y(z) = \dfrac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j\omega_{x0}}z^{-1})(1-0.99e^{-j\omega_{x0}}z^{-1})} + \dfrac{(1-z^{-1})(1+z^{-1})}{(1-0.99e^{j\omega_{y0}}z^{-1})(1-0.99e^{-j\omega_{y0}}z^{-1})}$

$$= \frac{(1-z^{-1})(1+z^{-1})(1-0.99e^{j\omega_{y0}}z^{-1})(1-0.99e^{-j\omega_{y0}}z^{-1}) + (1-z^{-1})(1+z^{-1})(1-0.99e^{j\omega_{x0}}z^{-1})(1-0.99e^{-j\omega_{x0}}z^{-1})}{(1-0.99e^{j\omega_{x0}}z^{-1})(1-0.99e^{-j\omega_{x0}}z^{-1})(1-0.99e^{j\omega_{y0}}z^{-1})(1-0.99e^{-j\omega_{y0}}z^{-1})}$$