ESE 531: Homework 5

Noah Schwab

February 28, 2022

Problem solutions with figures are shown below. Work and code is shown in attachments at end of document.

5.10

$H_i(z)$ can be stable, but it cannot be causal. See attached work for plots and proof.

5.23

a) False. Counterexample: $H(e^{j\omega}) = 1 - \frac{1}{2}e^{-j\omega}$

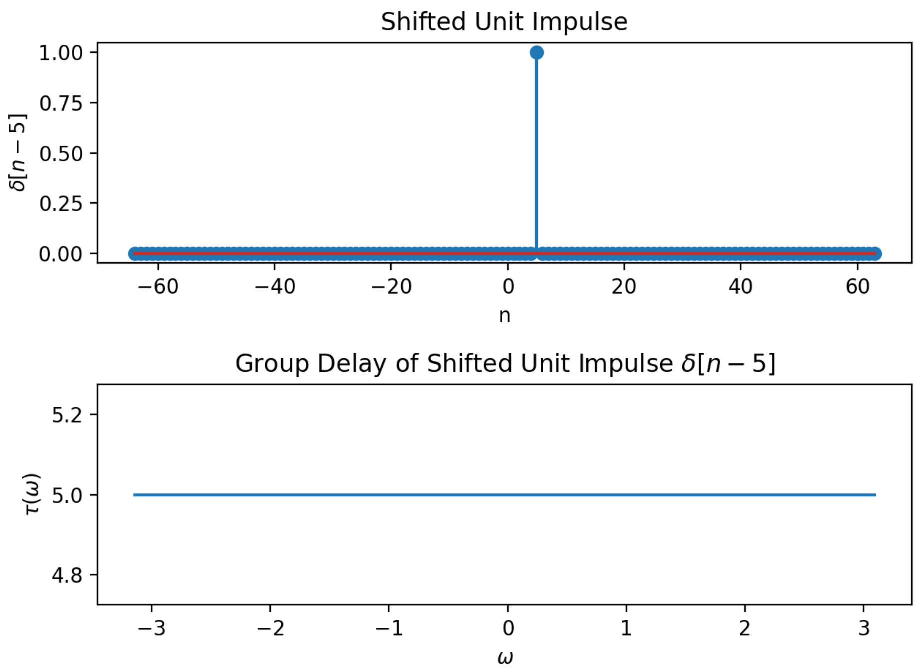
b) False. Counterexample: $H(z) = 1 + 2z^{-1} + 3z^{-2}$

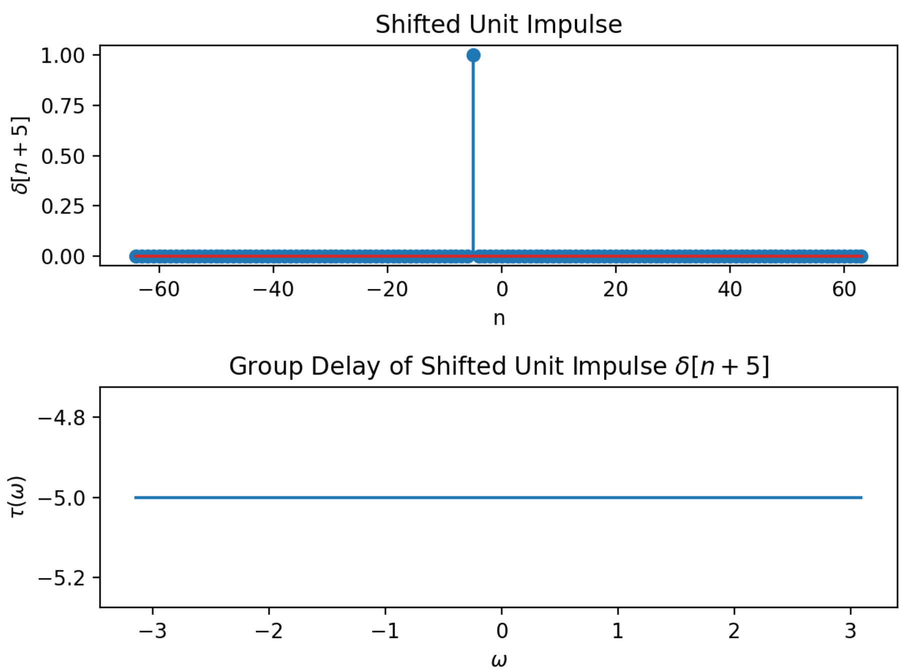
c) True.

5.34

The output of the system is $y_2[n]$.

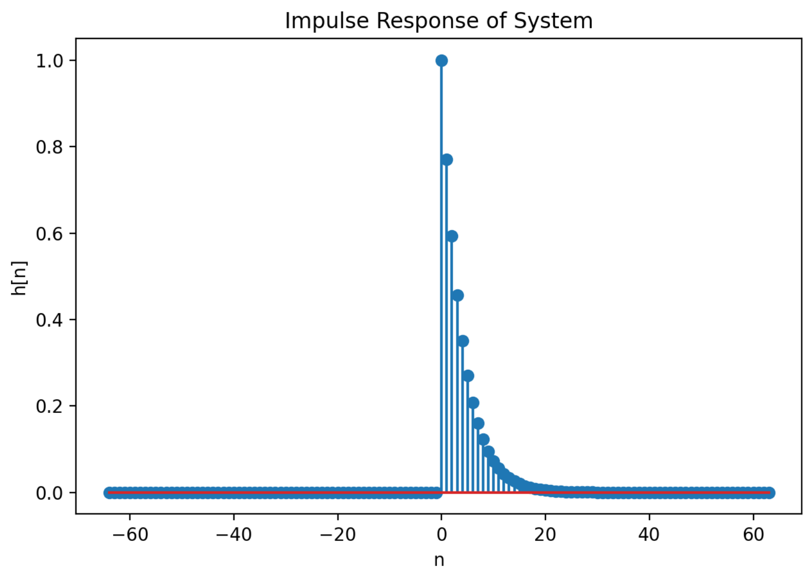
Matlab Problem 1: Group Delay



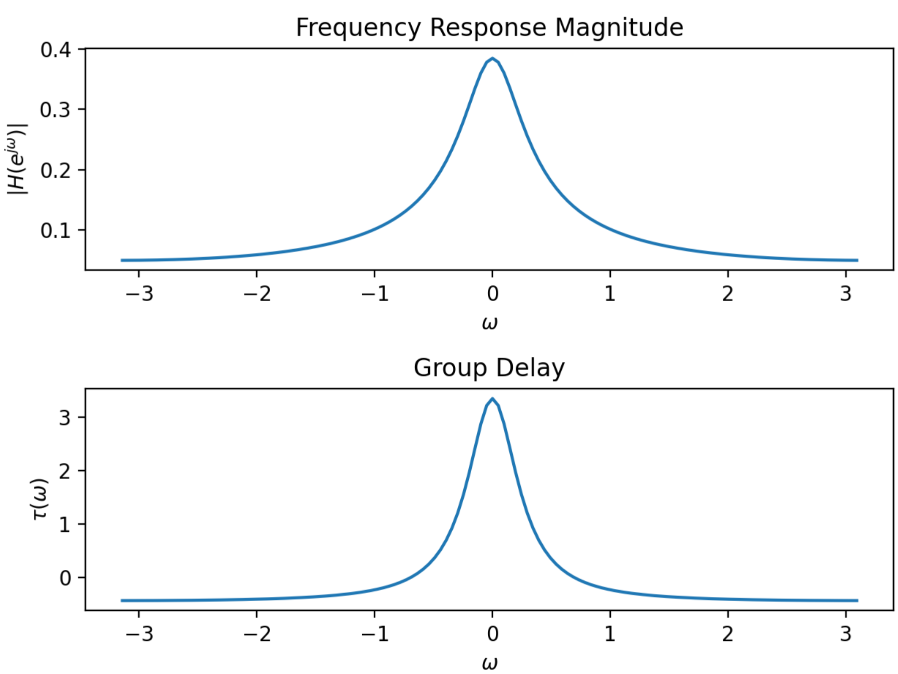


Matlab Problem 2: Causal First-Order System

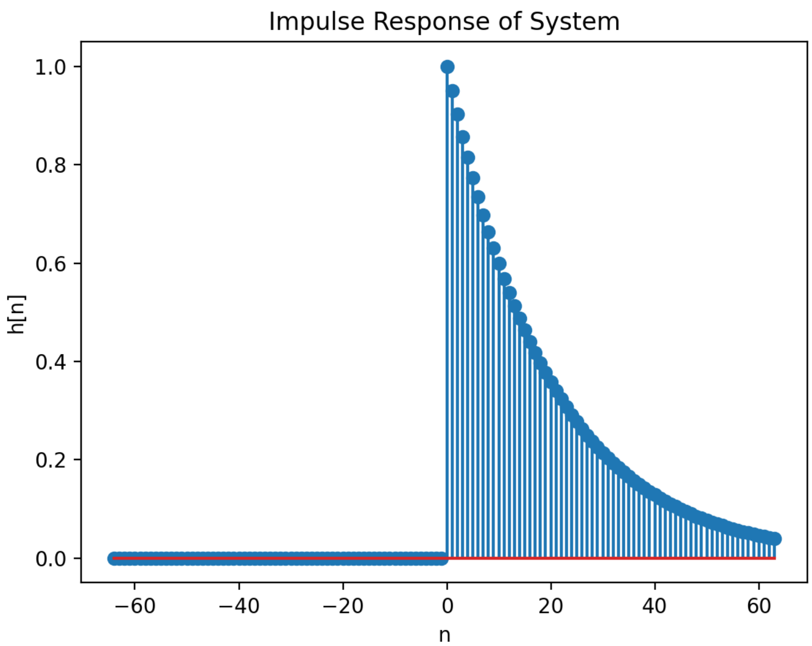
a) Impulse Response of Causal First-Order System

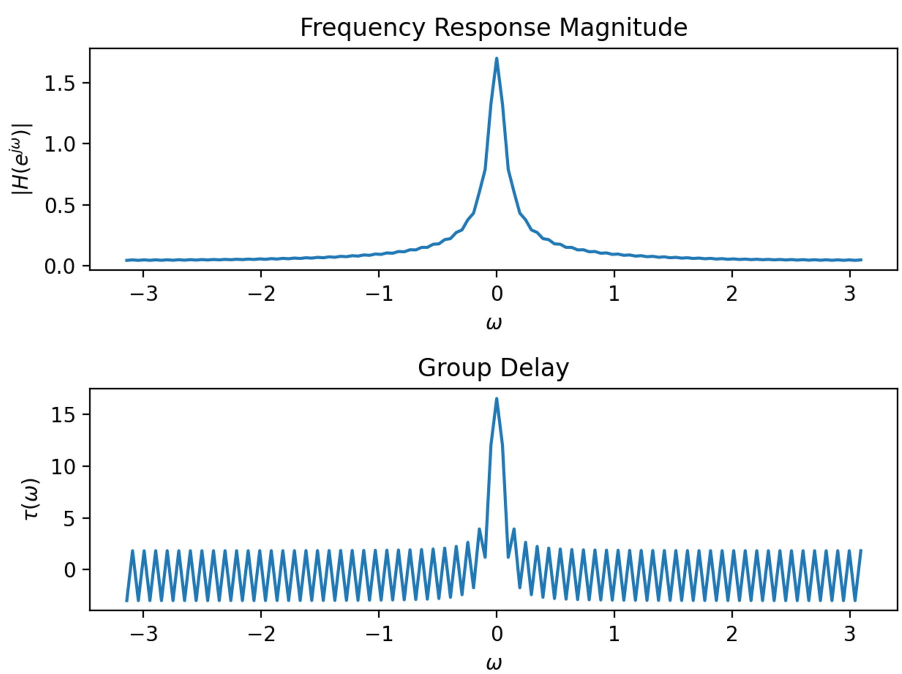


b) Frequency Response Magnitude and Group Delay of Causal First-Order System



c) Causal First-Order System with a Pole Closer to Unit Circle

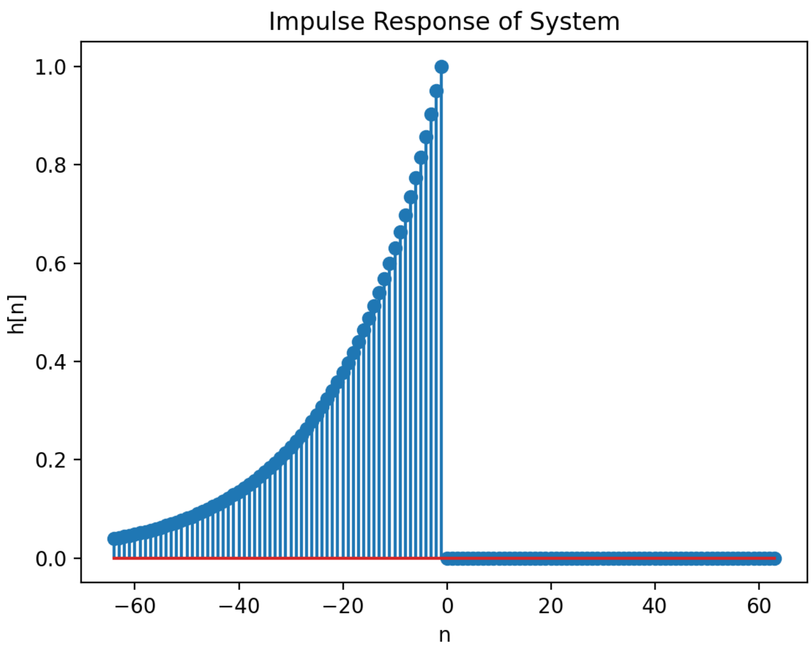




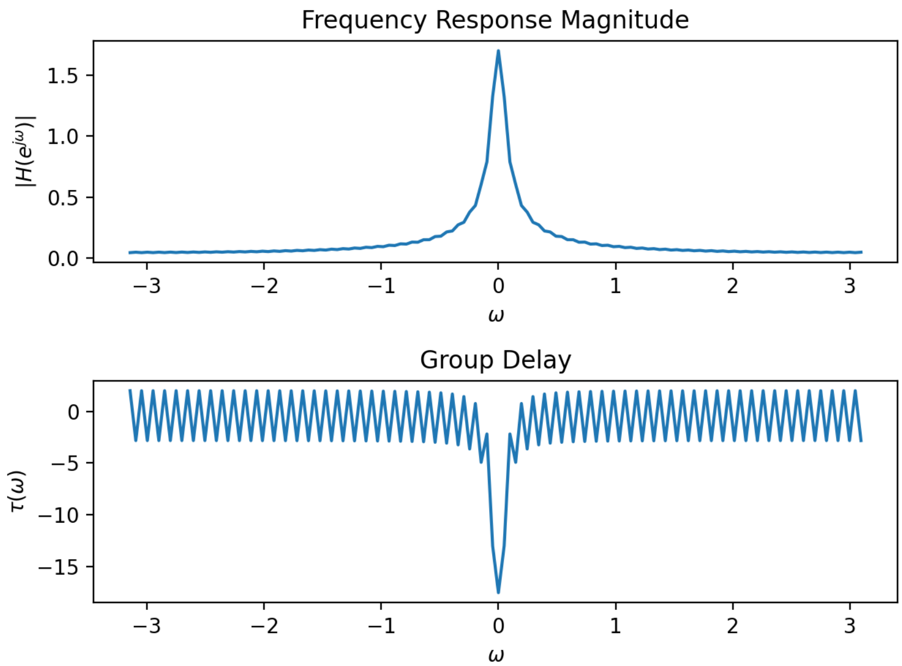
When the pole is moved closer to the unit circle, the impulse response expands in time and compresses in frequency. As we can see, for the pole located at 0.95, the impulse response has a density value that is spread out from n=0 to n=64, whereas the system with the pole located at 0.77 has larger values concentrated near n=0. Conversely, the system with a pole at 0.77 has a frequency response magnitude and group delay with values spread out over ω, whereas the system with a pole at 0.95 has a frequency response and group delay concetrated near $\omega = 0$.

Matlab Problem 3: Anticausal First-Order System

a) Impulse Response of Anticausal First-Order System

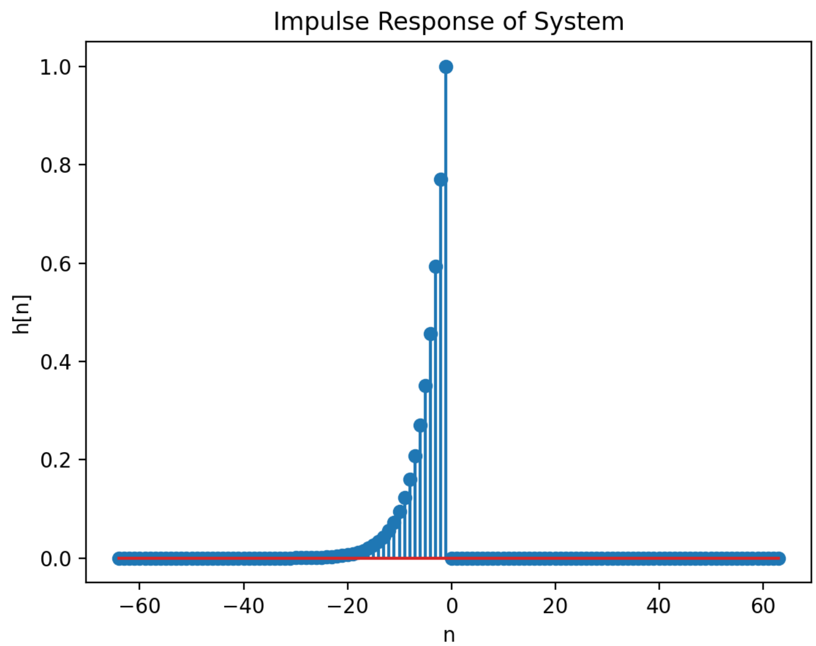


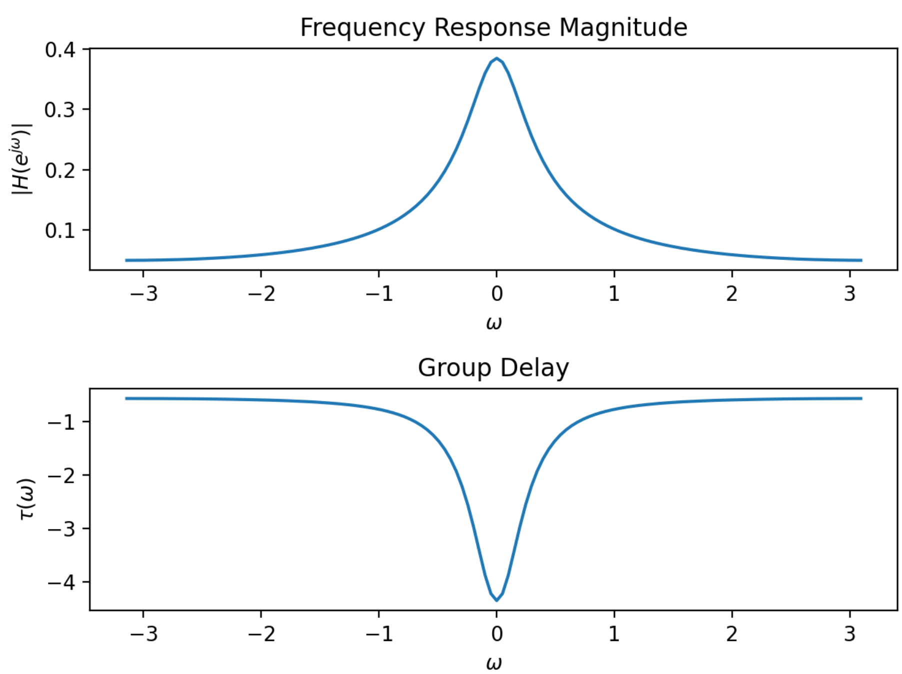
b) Frequency Response Magnitude and Group Delay of Anticausal First-Order System



c) There is a clear observable relationship between the representations of the causal filter from the previous problem and the anticausal filter. The impulse response is simply a mirror reflection about n=0. The frequency response magnitude does not change, since a flip in time does not affect the magnitude of the frequency response. The group delay, however, is a vertical flip about $\tau(\omega) = 0$, which makes sense given that group delay captures how much an input frequency will be delayed in the output, and flipping the impulse response about n=0 will change the sign of the delay at a given frequency.

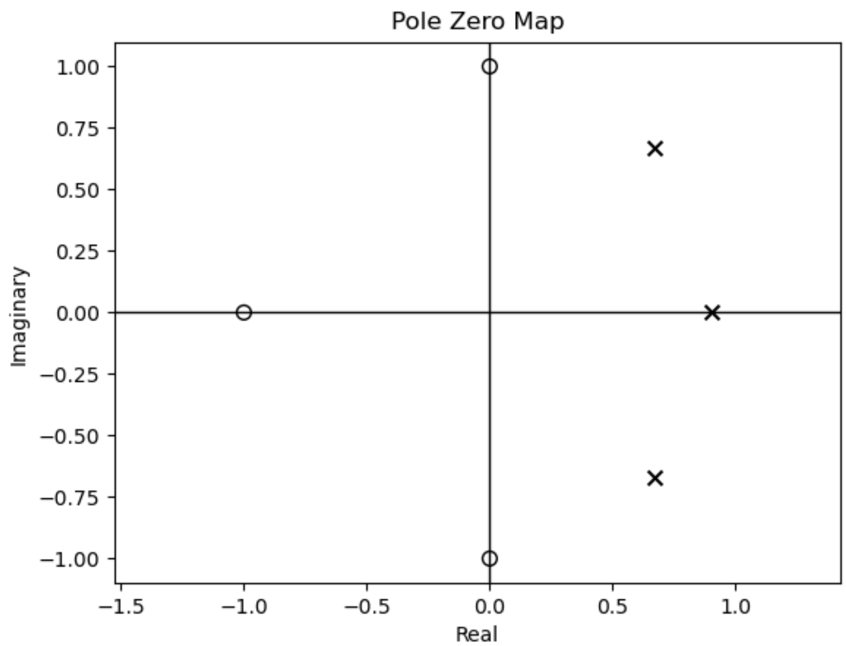
d) As we can see in the figures below, the relationship described in part c) also holds for the system with a pole at 0.77.





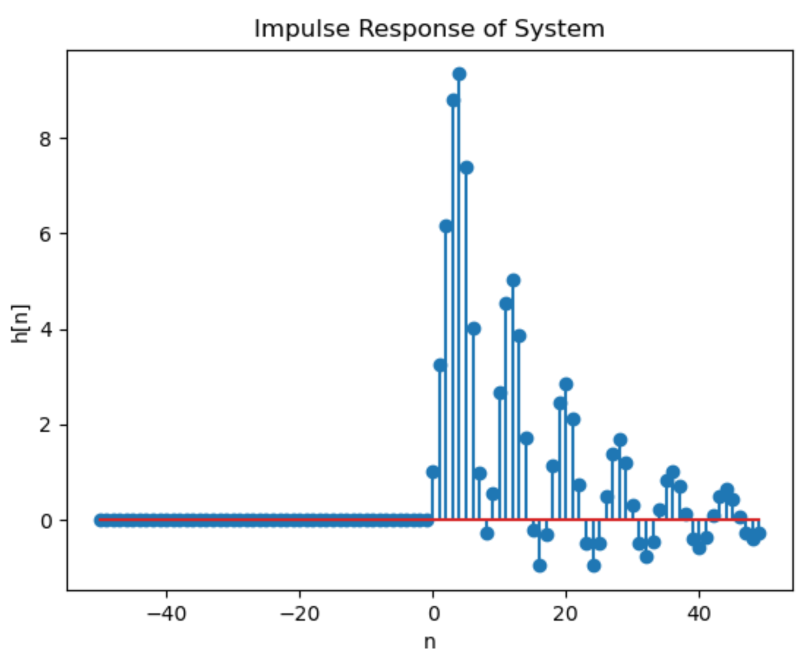
Matlab Problem 4: Higher-Order System

a) Pole-Zero Plot

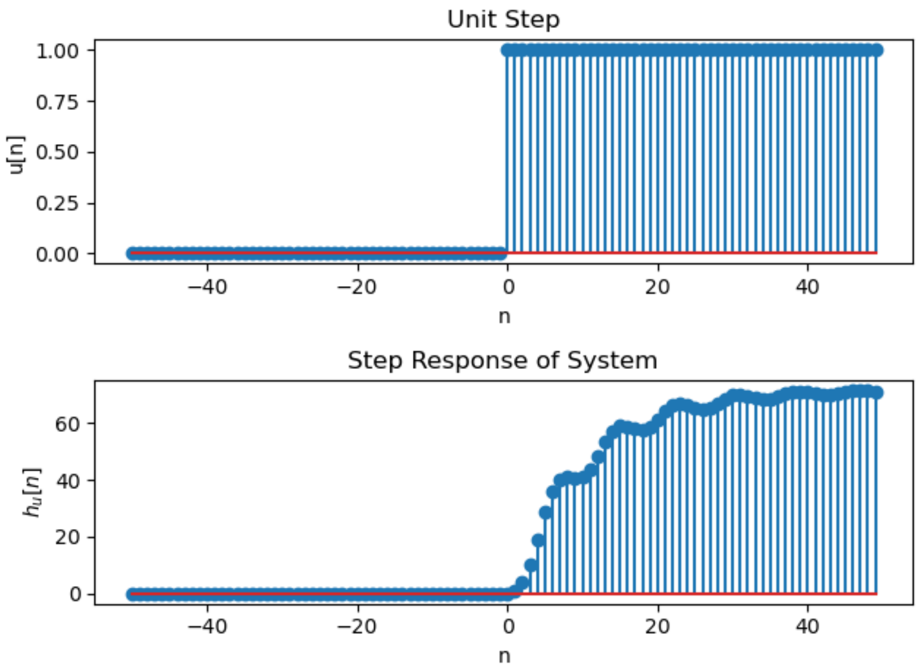


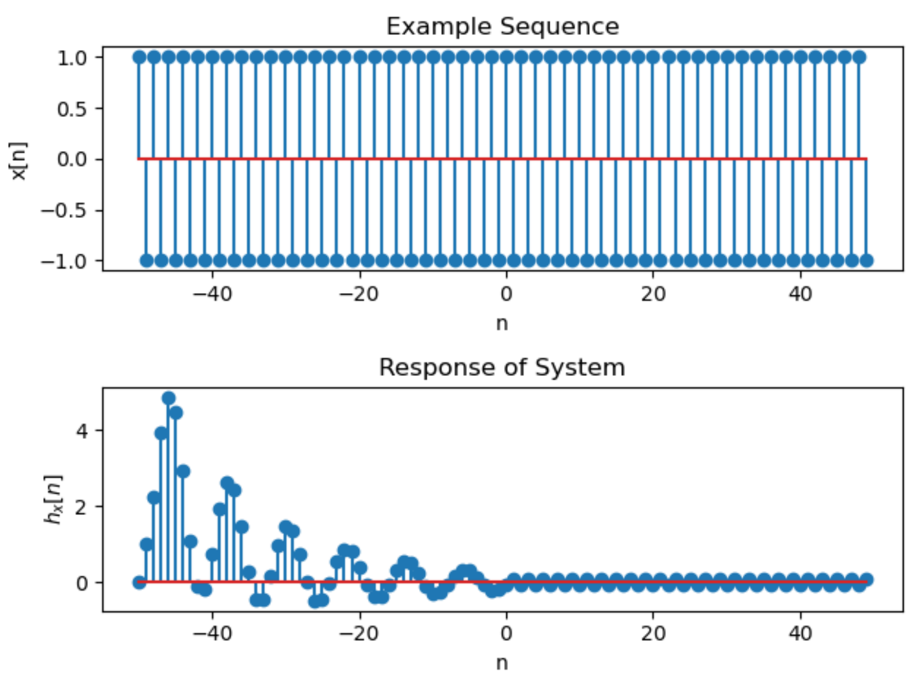
b) See attached python script ‘ESE531_HW5_P5.py’

c) Impulse Response

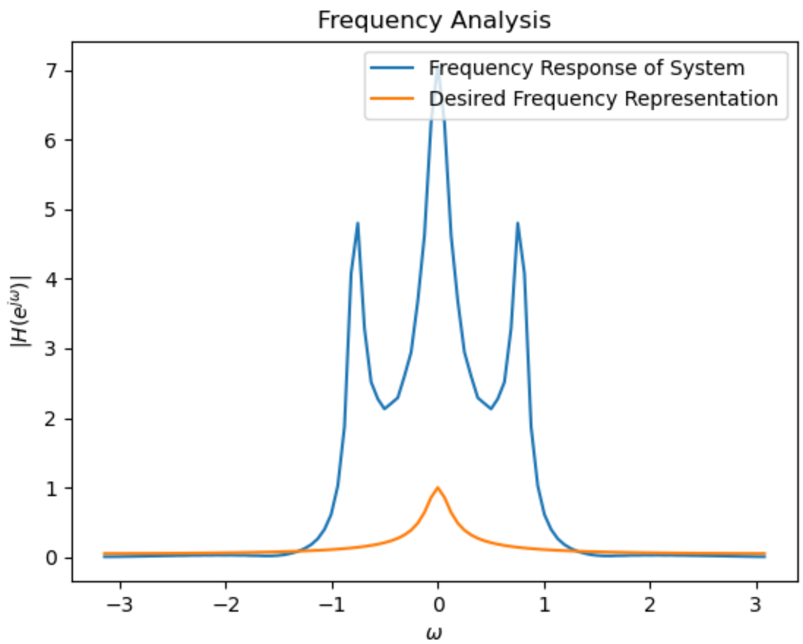


d) Example Responses of System

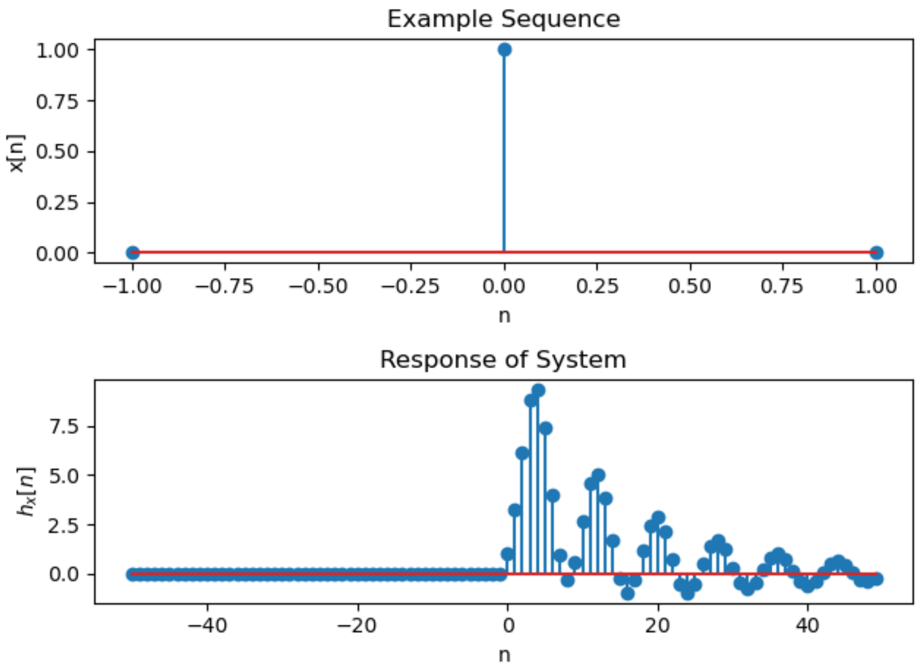




e) If we look at the frequency response of our higher order system and the frequency representation of a signal that is proportional to $(0.9)^n$, we can see that in order to achieve the desired output, we need to pass a signal that low-pass filters the impulse response of our system such that it obtains a frequency representation of the desired signal.



My attempt was to generate a sinc function as the input, since a sinc function in time corresponds to a square pulse in frequency, which would result in a sharp low-pass filter. However, we are restricted to a signal of length 3, which makes the implementation of a sharp low-pass filter difficult.

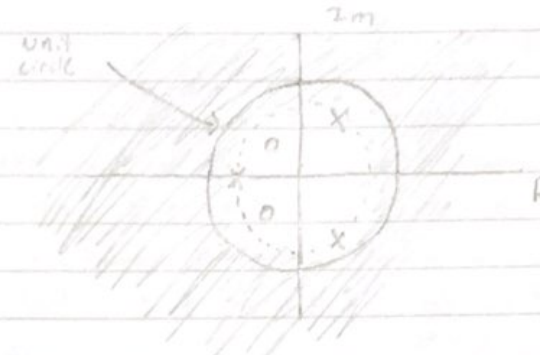

Example Sequence
x[n]
1.00
0.75
0.50
0.25
0.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
n
Response of System
$h_x[n]$
7.5
5.0
2.5
0.0
−40
−20
0
20
40
n

Noah Schwab
ESE 531 02/27/22

HW 5: Frequency Response LTI Systems

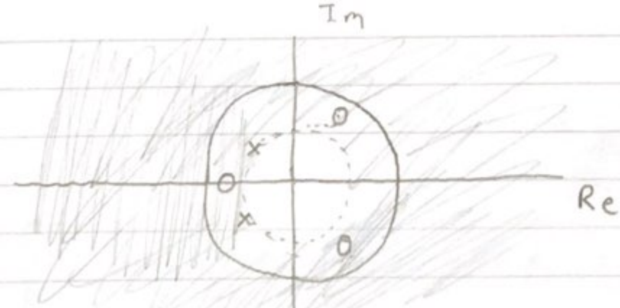
5.10

If $H(z)$ is causal, then it must have an ROC outside the outermost pole. In that case, the ROC of $H(z)$ is shown below



ROC $H(z)$

We see that the ROC includes the unit circle, so <u>$H(z)$ is both causal and stable</u>, but there must be an additional zero at $z=\infty$.
Consider the inverse system. $H_i(z)$ has an inverted pole-zero diagram as compared w/ $H(z)$ Furthermore, we know the ROC's of $H(z)$ and $H_i(z)$ must overlap



ROC $H_i(z)$

Since the ROC's must overlap, the only possible ROC for $H_i(z)$ is shown. The ROC includes the unit circle, so it is stable, but it has an upperbound at $z=\infty$, so it is <u>not causal</u>.

Alternatively, we could cite the definition of minimum-phase systems. $H(z)$ is stable and causal, but it has a zero at $z=\infty$, outside the unit-circle. Therefore, $H_i(z)$ cannot be stable and causal.

5.23

$$H(e^{jw}) = A(w)\, e^{j\theta(w)}$$

a) For the group delay to be nonnegative in the range $|w| < \pi$, then it must also be the case that the unwrapped phase is decreasing everywhere in that range. If the filter is causal, then $H(z)$ must have an ROC that extends to $z=\infty$. It is not guaranteed that a right-sided impulse response results in nonnegative group delay at $|w| < \pi$

Consider $\boxed{H(e^{jw}) = \dfrac{2e^{jw} - 1}{2e^{jw}} = 1 - \frac{1}{2}e^{-jw}}$

$H(z) = 1 - \frac{1}{2}z^{-1}$, which has one pole at $z=0$, and must include the unit-circle since it is defined for $H(e^{jw})$. Therefore, it is causal.

$$A(w) = 1$$
$$\theta(w) = \frac{e^{jw} - \frac{1}{2}}{e^{jw}}$$

For a first-order system, the group delay is given by

$$\text{grd}\left[1 - re^{j\theta}e^{-jw}\right] = \frac{r^2 - r\cos(w-\theta)}{|1 - re^{j\theta}e^{-jw}|^2}$$

In our case, $r = \frac{1}{2}$, $\theta = 0$

$$\text{grd}\left[1 - \tfrac{1}{2}e^{-jw}\right] = \frac{\frac{1}{4} - \frac{1}{2}\cos w}{|1 - e^{-jw}|^2}$$

Around $w = 0$, the group delay may have negative values. Therefore, this is False

b) False. While it is true that the simple integer delay has a group delay equal to the delay value, there are other systems, w/ a group delay that is a constant positive integer. Consider a zero-phase FIR filter that is delayed to become causal. The resulting filter has the same group phase as the delay system.

$$\boxed{H(z) = 1 + 2z^{-1} + 3z^{-2}}$$

c) True. If the filter is minimum phase and all poles and zeros are real, then all poles and zeros are between -1 and 1, since poles and zeros must be inside the unit circle

Consider the integral:

$$\int_0^{\pi} \tau(w)\,dw = -\theta(w)\Big|_0^{\pi} = \theta(0) - \theta(\pi)$$

Since all poles and zeros are between -1 and 1, there is always a phase contribution of 0. Therefore, $\theta(0) = \theta(\pi)$, which means

$$\boxed{\int_0^{\pi} \tau(w)\,dw = 0}$$

5.34

The highest frequency pulse at $w = 0.5$, which corresponds to the envelope at $n = 300$, will be eliminated due to shape of the frequency response magnitude.

The lowest frequency pulse at $w = 0.12$, which corresponds to the envelope at $n = 600$, will gain in amplitude by $\approx 1.9x$ and will be delayed by 40 samples

The mid frequency pulse at $w = 0.3$, which corresponds to the envelope at $n = 100$, will gain in amplitude by $\approx 1.5x$ and will be delayed by 80 samples

Therefore, the output of the system is $\boxed{y_2[n]}$

```python
# ESE 531: HW5 Problem 2

# Group Delay

# libraries
import numpy as np
import matplotlib.pyplot as plt

'''
function: gdel

arguments:

    x: signal x[n] at the times (n)
    n: vector of time indices
    Lfft: length of the FFT used

returns:

    gd: Group delay values on [pi, pi)
    w: list of frequencies over [-pi, pi)
'''
def gdel(x, n, Lfft):

    X = np.fft.fft(x, Lfft)
    dXdw = np.fft.fft(n * x, Lfft)
    gd = np.fft.fftshift(np.real(dXdw / X))
    w = (2 * np.pi / Lfft) * np.arange(0, Lfft) - np.pi

    return gd, w


if __name__ == "__main__":

    # generate a shifted unit impulse signal
    n = np.arange(-64, 64, 1)
    delta1 = np.zeros(128)
    delta2 = np.zeros(128)
    delta1[64 + 5] = 1
    delta2[64 - 5] = 1

    # generate and plot group delay
    gd1, w1 = gdel(delta1, n, 128)
    gd2, w2 = gdel(delta2, n, 128)

    # plot the delta functions with corresponding group delays

    fig, axs = plt.subplots(2)

    axs[0].stem(n, delta1)
    axs[0].set_title('Shifted Unit Impulse')
    axs[0].set_xlabel('n')
    axs[0].set_ylabel(r'$\delta[n - 5]$')

    axs[1].plot(w1, gd1)
    axs[1].set_title(r'Group Delay of Shifted Unit Impulse $\delta[n - 5]$')
    axs[1].set_xlabel(r'$\omega$')
    axs[1].set_ylabel(r'$\tau(\omega)$')

    fig.tight_layout()
    plt.show()

    fig, axs = plt.subplots(2)

    axs[0].stem(n, delta2)
    axs[0].set_title('Shifted Unit Impulse')
    axs[0].set_xlabel('n')
```

```
axs[0].set_ylabel(r'$\delta[n + 5]$')

axs[1].plot(w2, gd2)
axs[1].set_title(r'Group Delay of Shifted Unit Impulse $\delta[n + 5]$')
axs[1].set_xlabel(r'$\omega$')
axs[1].set_ylabel(r'$\tau(\omega)$')

fig.tight_layout()
plt.show()
```

```python
# ESE 531: HW5 Problem 4

# Anti-Causal First-Order System

# libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter
from ESE531_HW5_P2 import gdel

# part a

# generate impulse response by time reversing
h = np.zeros(128)
h[64] = 1
y = lfilter([1], [1, -0.95], h)
n = np.arange(-64, 64, 1)
y = np.flip(y)

# plot the impulse response
plt.stem(n, y)
plt.title('Impulse Response of System')
plt.xlabel('n')
plt.ylabel('h[n]')
plt.show()

# part b

# generate and plot the frequency response magnitude and group delay

# generate frequency response magnitude
f_mag = np.abs(np.fft.fft(y, 128, norm="ortho"))
f_mag = np.fft.fftshift(f_mag)

# generate group delay
gd, w = gdel(y, n, 128)

# plot the frequency response magnitude and group delay
fig, axs = plt.subplots(2)

axs[0].plot(w, f_mag)
axs[0].set_title('Frequency Response Magnitude')
axs[0].set_xlabel(r'$\omega$')
axs[0].set_ylabel(r'$|H(e^{j\omega})|$')

axs[1].plot(w, gd)
axs[1].set_title('Group Delay')
axs[1].set_xlabel(r'$\omega$')
axs[1].set_ylabel(r'$\tau(\omega)$')

fig.tight_layout()

plt.show()

# part d

# repeat parts a and b for when the pole is 1/0.77
h = np.zeros(128)
h[64] = 1
y2 = lfilter([1], [1, -0.77], h)
n2 = np.arange(-64, 64, 1)
y2 = np.flip(y2)

plt.stem(n2, y2)
plt.title('Impulse Response of System')
plt.xlabel('n')
plt.ylabel('h[n]')
```

```
plt.show()

f_mag2 = np.abs(np.fft.fft(y2, 128, norm="ortho"))
f_mag2 = np.fft.fftshift(f_mag2)

gd2, w2 = gdel(y2, n2, 128)

fig, axs = plt.subplots(2)

axs[0].plot(w2, f_mag2)
axs[0].set_title('Frequency Response Magnitude')
axs[0].set_xlabel(r'$\omega$')
axs[0].set_ylabel(r'$|H(e^{j\omega})|$')

axs[1].plot(w2, gd2)
axs[1].set_title('Group Delay')
axs[1].set_xlabel(r'$\omega$')
axs[1].set_ylabel(r'$\tau(\omega)$')

fig.tight_layout()

plt.show()
```

```python
# ESE 531: HW5 Problem 3

# Causal First-Order System

# libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter
from ESE531_HW5_P2 import gdel

# generate impulse response of given system function
h = np.zeros(128)
h[64] = 1
y = lfilter([1], [1, -0.77], h)
n = np.arange(-64, 64, 1)

# part a

# plot the impulse response
plt.stem(n, y)
plt.title('Impulse Response of System')
plt.xlabel('n')
plt.ylabel('h[n]')
plt.show()

# part b

# generate frequency response magnitude
f_mag = np.abs(np.fft.fft(y, 128, norm="ortho"))
f_mag = np.fft.fftshift(f_mag)

# generate group delay
gd, w = gdel(y, n, 128)

# plot the frequency response magnitude and group delay
fig1, axs1 = plt.subplots(2)

axs1[0].plot(w, f_mag)
axs1[0].set_title('Frequency Response Magnitude')
axs1[0].set_xlabel(r'$\omega$')
axs1[0].set_ylabel(r'$|H(e^{j\omega})|$')

axs1[1].plot(w, gd)
axs1[1].set_title('Group Delay')
axs1[1].set_xlabel(r'$\omega$')
axs1[1].set_ylabel(r'$\tau(\omega)$')

fig1.tight_layout()

plt.show()

# part c

# repeat parts a and b for a pole at 0.95
y2 = lfilter([1], [1, -0.95], h)
n2 = np.arange(-64, 64, 1)

plt.stem(n2, y2)
plt.title('Impulse Response of System')
plt.xlabel('n')
plt.ylabel('h[n]')
plt.show()

f_mag2 = np.abs(np.fft.fft(y2, 128, norm="ortho"))
f_mag2 = np.fft.fftshift(f_mag2)

gd2, w2 = gdel(y2, n2, 128)
```

```python
fig2, axs2 = plt.subplots(2)

axs2[0].plot(w2, f_mag2)
axs2[0].set_title('Frequency Response Magnitude')
axs2[0].set_xlabel(r'$\omega$')
axs2[0].set_ylabel(r'$|H(e^{j\omega})|$')

axs2[1].plot(w2, gd2)
axs2[1].set_title('Group Delay')
axs2[1].set_xlabel(r'$\omega$')
axs2[1].set_ylabel(r'$\tau(\omega)$')

fig2.tight_layout()

plt.show()
```

```python
# ESE 531: HW5 Problem 4

# Higher Order System

# libraries
import numpy as np
import matplotlib.pyplot as plt
import control
from scipy.signal import lfilter
from ESE531_HW5_P2 import gdel

# constants
p1 = 0.9
p2 = 0.6718 + 0.6718j
p3 = 0.6718 - 0.6718j
z1 = -1
z2 = 1j
z3 = -1j
b0 = 1 / 77

# part a

# polynomial coefficients derived by expanding pole, zero representation
sys = control.TransferFunction(np.poly((z1, z2, z3)), np.poly((p1, p2, p3)))
control.pzmap(sys, plot=True)
plt.show()

# part b
'''
function: zp2tf(p, z)

arguments:

    p: array_like of pole values
    z: array_like of zero values

return:

    Transfer Function description of impulse response as a rational function in orders of z^-1
'''

def zp2tf(z, p):

    # numerator is polynomial with zeros as roots
    num = np.poly(z)

    # denominator is polynomial with poles as roots
    den  = np.poly(p)

    return num, den

# part c

# use function zp2tf to compute impulse response
h = np.zeros(100)
h[50] = 1

z_list = [z1, z2, z3]
p_list = [p1, p2, p3]

b, a = zp2tf(z_list, p_list)
y = lfilter(b, a, h)
n = np.arange(-50, 50, 1)

plt.stem(n, y)
plt.title('Impulse Response of System')
plt.xlabel('n')
```

```python
plt.ylabel('h[n]')
plt.show()

# part d

# generate a 100 sample unit step signal
u_n = np.zeros(100)
u_n[50:] = 1

# step response of the system is the impulse convolved with the unit step
step_response = np.convolve(y, u_n, mode='same')

# plot the unit step along with the system step response
fig, axs = plt.subplots(2)

axs[0].stem(n, u_n)
axs[0].set_title('Unit Step')
axs[0].set_xlabel('n')
axs[0].set_ylabel('u[n]')

axs[1].stem(n, step_response)
axs[1].set_title('Step Response of System')
axs[1].set_xlabel('n')
axs[1].set_ylabel(r'$h_u[n]$')

fig.tight_layout()

plt.show()

# another example input sequence
x_n = np.ones(100)
for i in range(100):
    if i % 2 != 0:
        x_n[i] *= -1

x_response = np.convolve(y, x_n, mode='same')

fig, axs = plt.subplots(2)

axs[0].stem(n, x_n)
axs[0].set_title('Example Sequence')
axs[0].set_xlabel('n')
axs[0].set_ylabel('x[n]')

axs[1].stem(n, x_response)
axs[1].set_title('Response of System')
axs[1].set_xlabel('n')
axs[1].set_ylabel(r'$h_x[n]$')

fig.tight_layout()

plt.show()

# part e

x_3 = np.sinc(np.arange(-1, 2, 1))
x_3response = np.convolve(y, x_3, mode='same')

fig, axs = plt.subplots(2)

axs[0].stem(np.arange(-1, 2), x_3)
axs[0].set_title('Example Sequence')
axs[0].set_xlabel('n')
axs[0].set_ylabel('x[n]')

axs[1].stem(n, x_3response)
axs[1].set_title('Response of System')
```

```python
axs[1].set_xlabel('n')
axs[1].set_ylabel(r'$h_x[n]$')

fig.tight_layout()

plt.show()

# show frequency response and desired frequency representation
gd, w = gdel(y, n, 100)
plt.plot(w, np.fft.fftshift(np.abs(np.fft.fft(y, 100, norm="ortho"))), label='Frequency Response of System')

des_signal = [0.9**n for n in range(100)]
plt.plot(w, np.fft.fftshift(np.abs(np.fft.fft(des_signal, 100, norm="ortho"))), label='Desired Frequency Representation')
plt.title('Frequency Analysis')
plt.xlabel(r'$\omega$')
plt.ylabel(r'$|H(e^{j\omega})|$')
plt.legend()
plt.show()
```