Linear Time Invariant System

Lab Exercises on January 16, 2022

Department of Electronics and Computer Engineering

Pulchowk Campus, Lalitpur

Ashlesh Pandey

PUL074BEX007

Contents

List of Figures

Listings

1 Objectives

- Familiarization with linear system transformations and response visualization.
- Familiarization with second order section representation of LTI system.

2 Background Theory

2.1 Linear Time Invariant (LTI) System

$$x[n] \longrightarrow \boxed{H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}} \longrightarrow y[n]$$

Figure 1: LTI system with transfer function $H(z)$

A linear time invariant (LTI)system is characterized by the transfer function $H(z) = \dfrac{Y(z)}{X(z)}$, where $X(z)$ and $Y(z)$ are the Z-transforms of the sequences $x[n]$ and $y[n]$ respectively. When the inverse Z-transform of the transfer function $H(z)$ is taken, the resulting difference equation can be written as:

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots + b_Nx[n-N] - a_1y[n-1] - a_2y[n-2] - \cdots - a_Ny[n-N]$$

The digital signal processing basically deals with the methods of implementation of the above difference equation. The equation can be solved using both hardware and software. It can be implemented using microprocessor based designs in which the assembly language programming plays the vital role. For the fast processing of the signals, considering the improved system performance the digital signal processing chips are preferred. The DSP chips are based on the Harvard architecture rather than the Von-Neumann's architecture, usually found in most of the personal computers.

If the transfer function $H(z)$ is known for the given LTI system the MATLAB signal processing toolbox functions can be used to plot the frequency response of the system. For this there is a function;

```
[H, W]= freqz(b, a, w)
```

which gives the complex values in amplitude `H` and angle `W` radians versus w points frequency. Here `a` and `b` are the vector sequences representing the numerator and denominator coefficients of $H(z)$.

2.2 Linear Systems Transformation

In discrete time systems the transfer function in the Z domain plays the key role in determining the nature of the system. The nature of the system is determined from the number and locations of the poles and zeros in the Z-plane. In lower order systems the locations of the poles and zeros can be easily determined from the transfer function of the system. However the higher order systems possess transfer functions with numerator and denominator polynomials of greater degree. As a result the process of determining the poles and zeros of the system becomes complex and tedious. Besides this in most of the higher order discrete time systems, the transfer function is specified in the form of second order sections. If the transfer function of the system consists of a large number of such second order sections either in cascade or parallel form, the determination of the poles and zeros of the system becomes even harder.

MATLAB signal processing toolbox provides a number of functions for transforming the discrete time linear systems from one form to another. The name of the functions and their purpose are listed as follows:

Function	**Purpose**
`sos2zp`	Transforms second order sections into zeros and poles.
`sos2tf`	Performs second order sections to transfer function conversion.

`tf2zp`	Transfer function to pole zero conversion.
`zp2sos`	Zero-poles to second order sections.
`zplane`	Plots the pole-zero diagram in Z-plane.
`freqz`	Determines the magnitude and phase of the transfer function.

Note: Similar functions are available in the `Scipy` library in Python.

3 Lab Exercises

Problem 1

In the given LTI system of fig above, if the coefficients b and a are specified as,

$b_0 = 0.0663, \quad b_1 = 0.1989, \quad b_2 = 0.1989, \quad b_3 = 0.0663$

$a_0 = 1, \quad a_1 = -0.9349, \quad a_2 = 0.5668, \quad a_3 = -0.1015$

then the order of the system is 3 i.e. $N = 3$.

a. Plot the frequency response of the system.

b. From the magnitude response of the system, find out the cut-off frequency.

c. Identify the nature of the system analyzing its frequency response.

```python
import numpy as np
from scipy import signal
from matplotlib import pyplot as plt

b = np.array([0.0663, 0.1989, 0.1989, 0.0663])
a = np.array([1, -0.9349, 0.5668, -0.1015])
w, h = signal.freqz(b, a)
h_mag = 20 * np.log10(abs(h))
h_phase = np.unwrap(np.arctan2(np.imag(h), np.real(h)))
cutoffMag = max(h_mag)-3
cutoffFreq = np.interp(cutoffMag, np.flipud(h_mag), np.flipud(w))/np.pi
cutOffMagDisplay='$K_c=$ {:.3f}'.format(cutoffMag)
cutOffFreqDisplay='$\omega_c=$
```

```
{:.5f}'.format(cutoffFreq)
fig=plt.figure(constrained_layout=True)
(subfig1, subfig2)=fig.subfigures(1,2)
(axleft1,axleft2)=subfig1.subplots(2,1,sharex=True)
axleft1.plot(w/np.pi, h_mag)
axleft1.grid()
axleft1.set_title('Magnitude plot')
axleft1.set_ylabel('Magnitude (dB)',)
axleft1.axhline(cutoffMag, color='red', linestyle='--', linewidth=1)
axleft1.text(cutoffFreq, cutoffMag, cutOffMagDisplay, horizontalalignment='right',
        verticalalignment='top',rotation=60,color='red')
axleft1.axvline(cutoffFreq, color='green', linestyle='--', linewidth=1)
axleft1.text(cutoffFreq, 0, cutOffFreqDisplay, horizontalalignment='left',
        verticalalignment='bottom',rotation=60,transform=axleft1.get_xaxis_transform(),color='green')
axleft2.plot(w/np.pi, h_phase*180/np.pi)
axleft2.grid()
axleft2.set_title('Phase plot')
axleft2.set_ylabel('Phase (degree)')
axRight=subfig2.subplots(1,1)
axRight.plot(w/np.pi,abs(h))
axRight.grid()
axRight.set_title('Linearized form')
axRight.set_ylabel('Gain')
fig.suptitle('Frequency Response of the system\n(PUL074BEX007)')
fig.supxlabel('Normalized Frequency ($\\times \pi$ rad/sample)')
plt.savefig('../Figures/lab_4_1_py.pdf', format='pdf')
plt.show()
```

Listing 1: Python script for frequency response and linearized form visualization for given coefficients

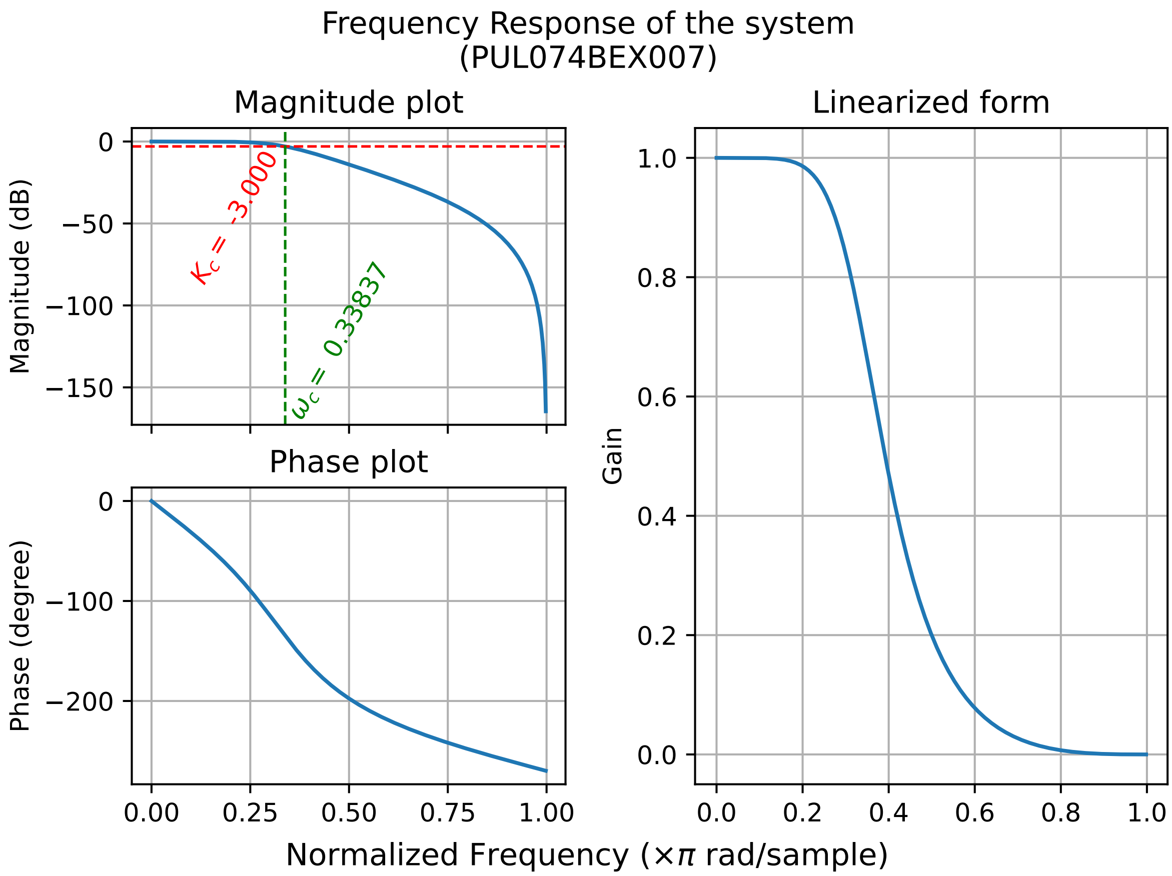

Figure 2: Observation of frequency response and linearized form for given coefficients

The frequency response for given set of coefficients is shown in Figure 2. The magnitude response has annotations added at cut-off frequency and cut-off magnitude for better result and visualization. The noted value of cut-off frequency is $\omega_c = 0.33837$, which is actually in normalized form, i.e. ($\times\pi$ rad/sample). Similarly, the linearized form plot suggests that the nature of the system is low pass.

Problem 2

The transfer function of the fourth-order discrete time system is given as,

$$H(z) = \frac{0.0018 + 0.0073z^{-1} + 0.011z^{-2} + 0.007z^{-3} + 0.008z^{-4}}{1 - 3.0544z^{-1} + 3.8291z^{-2} - 2.2925z^{-3} + 0.55072z^{-4}}$$

a. Find out the poles and zeros of the system and plot them in the z- plane.

b. Use them to determine the second order sections in the cascaded form.

c. Plot the frequency response of the system and comment on the nature of the system.

d. After knowing the numerator and denominator coefficients of each second order section, draw the signal flow graph to represent the cascaded structure.

```
import numpy as np
from scipy import signal
from plot_zplane import zplane
from matplotlib import pyplot as plt
b = np.array([0.0018,0.0073,0.011,
              0.007,0.008])
a = np.array([1,-3.0544,3.8291,
              -2.2925,0.55072])
zeros, poles, gain= signal.tf2zpk(b,a)
sos=signal.tf2sos(b,a)
print(sos)
z1,p1,g1=signal.tf2zpk(sos[0,0:3],sos[0,3:6])
z2,p2,g2=signal.tf2zpk(sos[1,0:3],sos[1,3:6])
w, h = signal.freqz(b, a)
fig=plt.figure(constrained_layout=True)
(subfig1, subfig2)=fig.subfigures(1,2,width_ratios=[1,3])
subfig1.suptitle('Pole and zero plot')
(axleft1,axleft2,axleft3)=subfig1.subplots(3,1)
zplane(zeros,poles,axleft1)
axleft1.set_title('System')
axleft1.set_ylabel('Imaginary part')
axleft1.set_xlabel('Real part')
zplane(z1,p1,axleft2)
axleft2.set_title('1st SOS')
axleft2.set_ylabel('Imaginary part')
axleft2.set_xlabel('Real part')
zplane(z2,p2,axleft3)
axleft3.set_title('2nd SOS')
axleft3.set_ylabel('Imaginary part')
axleft3.set_xlabel('Real part')
axRight=subfig2.subplots(1,1)
axRight.plot(w/np.pi,abs(h))
axRight.grid()
axRight.set_title('Linearized form')
```

```python
axRight.set_ylabel('Gain')
axRight.set_xlabel('Normalized Frequency ($\\times \pi$ rad/sample)')
fig.suptitle('Zero-pole plot and frequency response of the system\n(PUL074BEX007)')
plt.savefig('../Figures/lab_4_2_py.pdf', format='pdf')
plt.show()
```

Listing 2: Python script for determination of SOS, zero-pole plot, and linearized form visualization for given transfer function

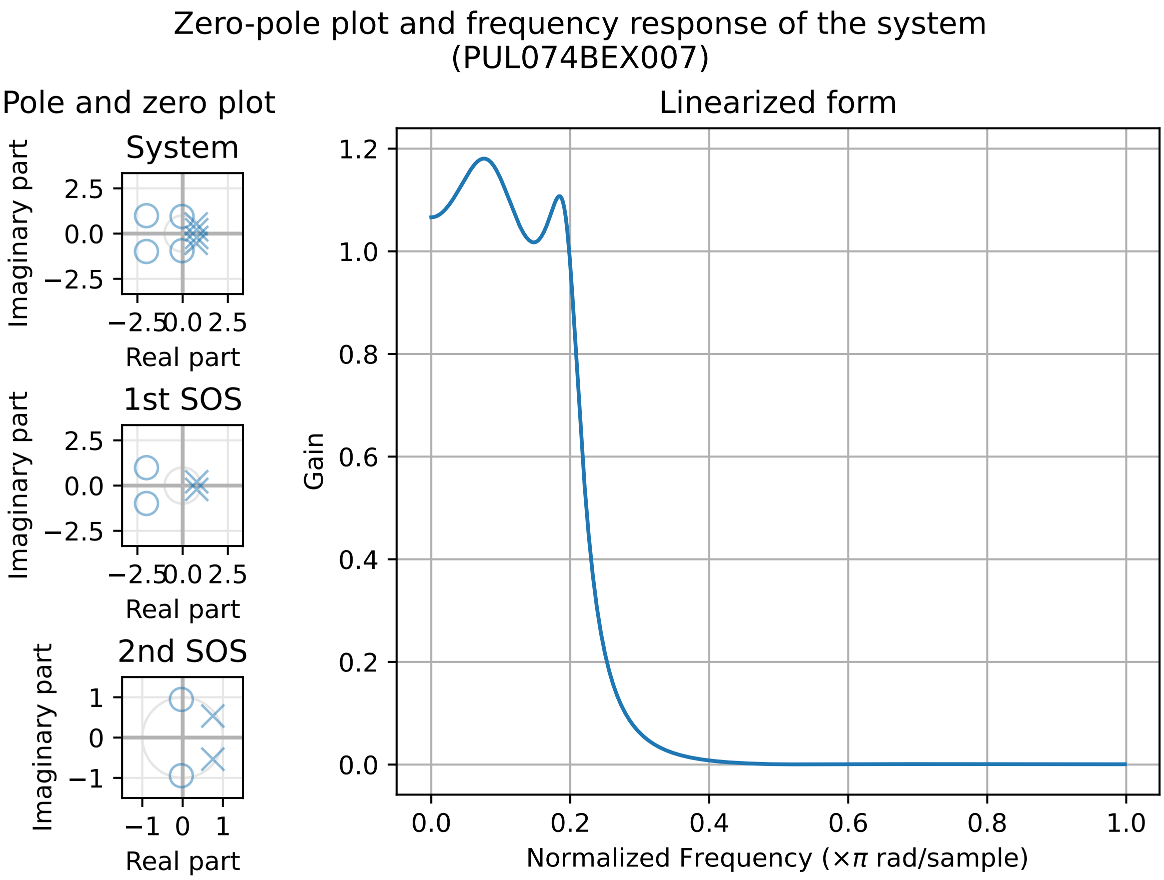

Figure 3: Observation of zero-pole plot and linearized form for given transfer function

The zero-pole plot in the Z-plane for the system, the first and second sections of the SOS are observed in Figure 3. Similarly, the linearized form plot suggests that the nature of the system is low pass. The second order sections for the given transfer function are displayed in the terminal as,

```
[[ 0.0018  0.00718618  0.00893373  1.  -1.55475193  0.64915547]
[ 1.  0.06323235  0.89548237  1.  -1.49964807  0.8483638]]
```

Upon review of the return value from `tf2sos`, each row contains one section of the SOS. Likewise, the first three values in each row represent the numerator coefficients and the remaining three values represent the denominator coefficients of that section. Hence, the two second order sections in cascade structure can be represented as,

$$H_1(z) = \frac{0.0018 + 0.00718618z^{-1} + 0.00893373z^{-2}}{1 - 1.55475193z^{-1} + 0.64915547z^{-2}}$$

$$H_2(z) = \frac{1 + 0.06323235z^{-1} + 0.89548237z^{-2}}{1 - 1.49964807z^{-1} + 0.8483638z^{-2}}$$

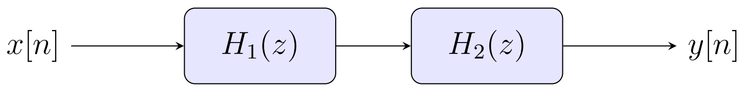

Figure 4: LTI system represented as cascade of two SOS

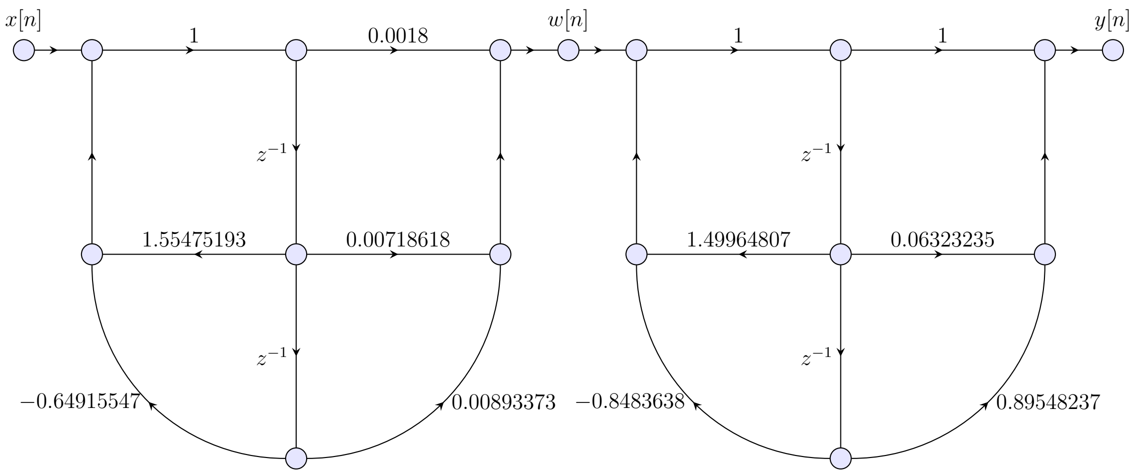

Figure 5: Signal flow graph to represent the cascade structure in Figure 4

Problem 3

Let a discrete time system be implemented by cascading of the following three second order sections:

Section 1: $H_1(z) = \dfrac{0.0007378(1+2z^{-1}+z^{-2})}{1-1.2686z^{-1}+0.7051z^{-2}}$

Section 2: $H_2(z) = \dfrac{1+2z^{-1}+z^{-2}}{1-1.0106z^{-1}+0.3583z^{-2}}$

Section 3: $H_3(z) = \dfrac{1+2z^{-1}+z^{-2}}{1-0.9044z^{-1}+0.2155z^{-2}}$

a. Using above three second order sections in cascaded form determine the poles and zeros of the system and plot them in z-plane.

b. Determine the transfer function of the system, formed by cascading of the above three sections. Determine the poles and zeros from this transfer function and plot them in z-plane. Your result should match with that from previous one.

c. Draw the direct form structures-I and II of the system.

```python
import numpy as np
from scipy import signal
from plot_zplane import zplane
from matplotlib import pyplot as plt
b = np.array([1,2,1])
a1 = np.array([1,-1.2686,0.7051])
a2 = np.array([1,-1.0106,0.3583])
a3 = np.array([1,-0.9044,0.2155])
sos=np.vstack((np.hstack((0.0007378*b,a1)),np.hstack((b,a2)),np.hstack((b,a3))))
z1, p1, k1=signal.sos2zpk(sos)
b_tf,a_tf=signal.sos2tf(sos)
print(b_tf)
print(a_tf)
z2,p2,k2=signal.tf2zpk(b_tf,a_tf)
fig, (ax1, ax2) = plt.subplots(1,2,constrained_layout=True)
zplane(z1,p1,ax1)
ax1.set_title('From cascade sections')
ax1.set_ylabel('Imaginary part')
ax1.set_xlabel('Real part')
zplane(z2,p2,ax2)
ax2.set_title('From transfer function')
ax2.set_ylabel('Imaginary part')
ax2.set_xlabel('Real part')
fig.suptitle('Poles and zeros plot of system\n(PUL074BEX007)',y=0.9)
```

```python
plt.savefig('../Figures/lab_4_3_py.pdf', format='pdf')
plt.show()
```

Listing 3: Python script for zero-pole plot visualization for given SOS cascade

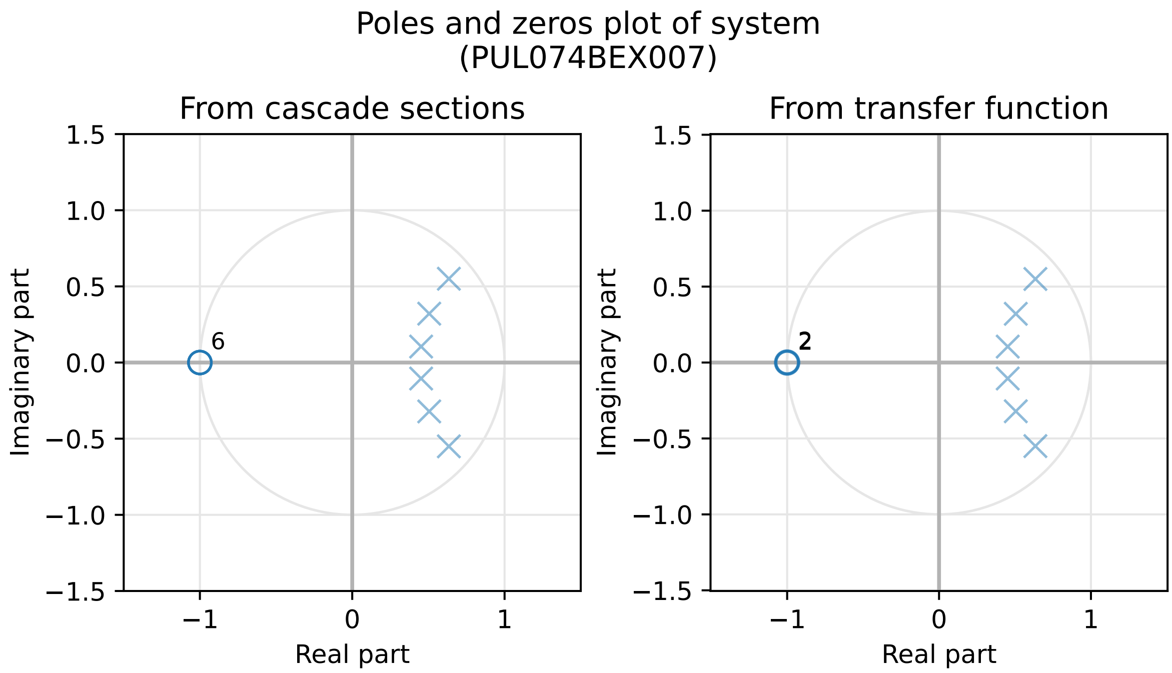

Figure 6: Observation of zero-pole plot for given SOS cascade

The zero-pole plot for cascade of three sections of SOS and the zero-pole plot from the determined transfer function is same as seen in Figure 6. Slight deviation is seen as there are 6 zeros as same co-ordinate for the plot from cascade of sections where as there are 3 set of 2 poles each in same co-ordinate for the plot from transfer function, which is due to computational accuracy. The coefficients of numerator and denominator of the transfer function, b and a for the given cascade of three SOS are displayed in the terminal as,

```
[0.0007378  0.0044268  0.011067  0.014756  0.011067  0.0044268
0.0007378]
[ 1.  -3.1836  4.62225564  -3.77950345  1.81361859  -0.47999815
0.05444334]
```

Hence the transfer function can be written as,

$$H(z) = \frac{\begin{array}{r}0.0007378 + 0.0044268z^{-1} + 0.011067z^{-2} + 0.014756z^{-3} \\ + 0.011067z^{-4} + 0.0044268z^{-5} + 0.0007378z^{-6}\end{array}}{\begin{array}{r}1 - 3.1836z^{-1} + 4.62225564z^{-2} - 3.77950345z^{-3} \\ + 1.81361859z^{-4} - 0.47999815z^{-5} + 0.05444334z^{-6}\end{array}} \tag{1}$$

Direct form I

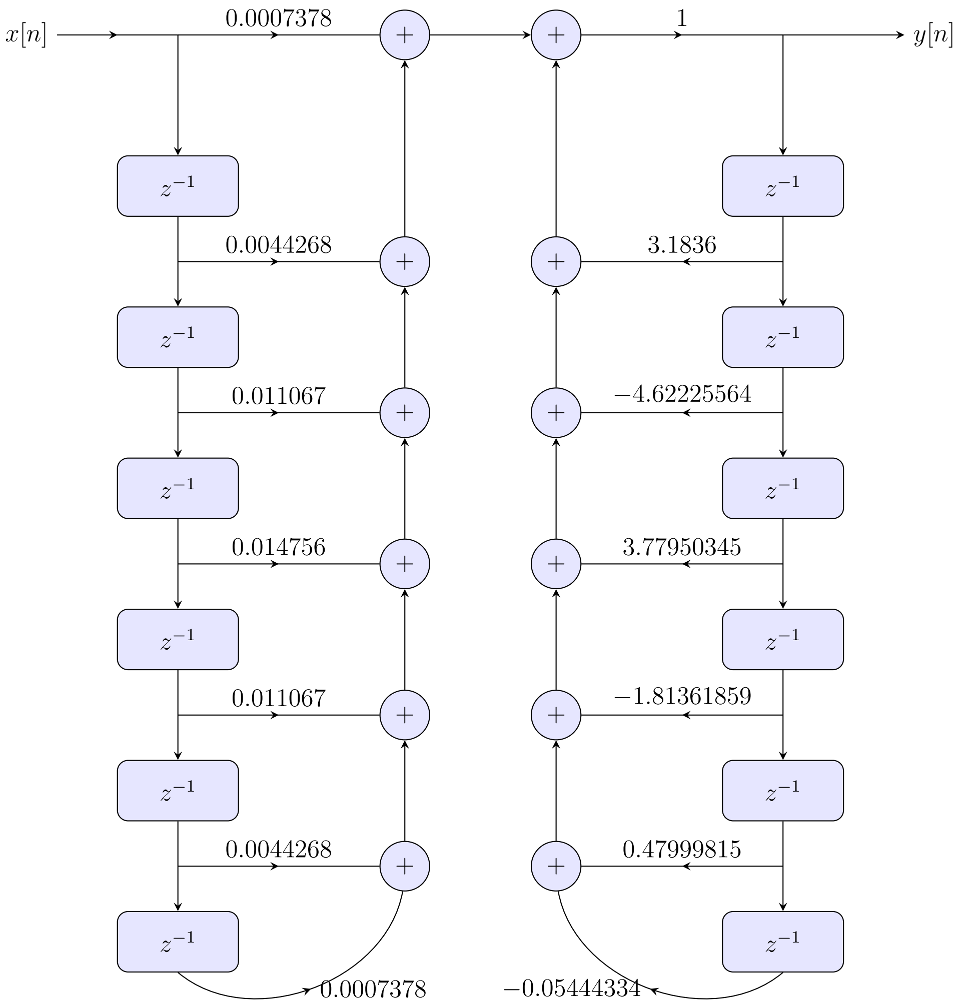

Figure 7: Direct form I to represent the determined transfer function in Equation 1

Direct form II

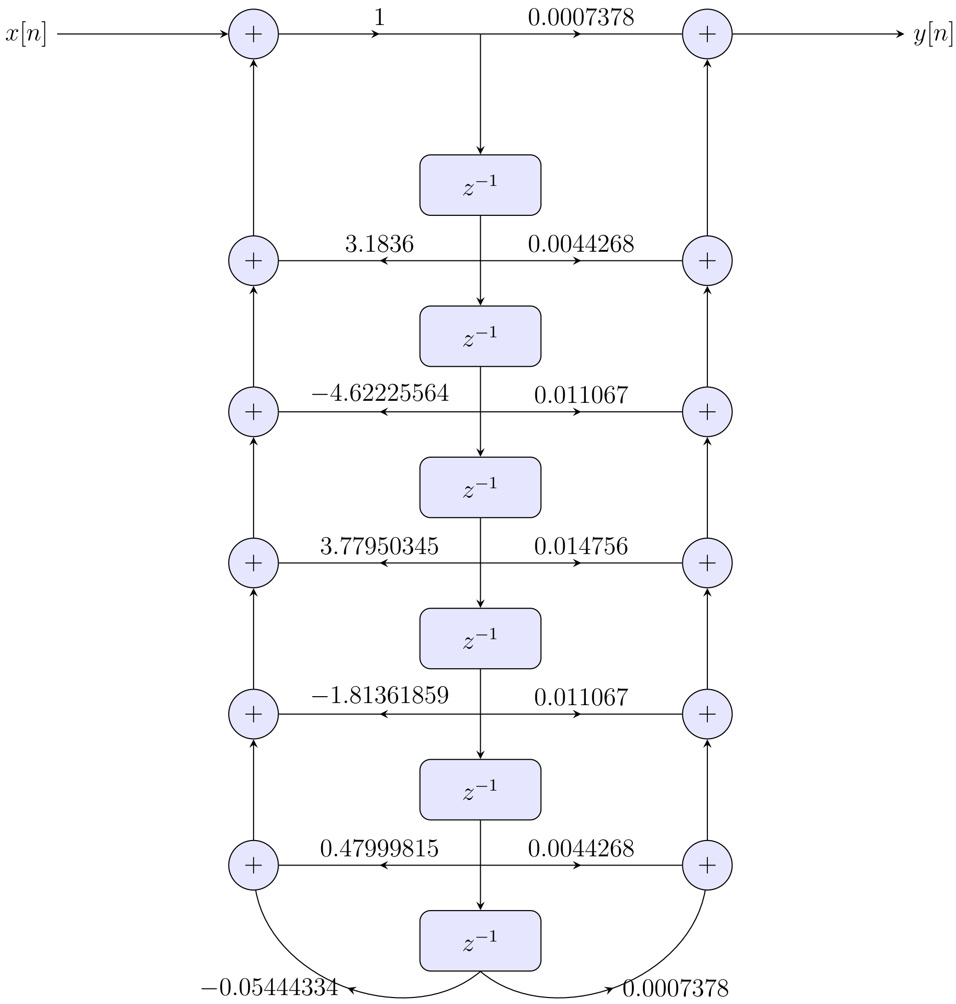

Figure 8: Direct form II to represent the determined transfer function in Equation 1

4 Discussion and Conclusion

In this lab experiment, we performed different visualizations and calculations for linear time invariant (LTI) systems. Initially, with the first problem the given coefficients of numerator and denominator were used to plot the frequency response of the system. Cut-off annotation were also added to determine the cut-off frequency. Similarly, the linearized form of the

response was used to determine the nature of the response, which was low-pass. Another problem with the fourth order transfer function was plotted in the z-plane along with the two second order sections (SOS) zero-pole plot. Similarly, the linearized form of the response was used to determine the nature of the system which was also low-pass. Furthermore, a signal flow graph to represent the cascade structure of the two SOS was drawn as shown in Figure 5. Lastly, with the given three SOS, the transfer function was determined. The zero-pole plots drawn directly from the cascaded sections and the transfer function were compared. Finally, the transfer function obtained in Equation 1 was represented in direct form I and II in Figure 7 and Figure 8 respectively.
Hence, the objectives of the lab experiment were fulfilled.