National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering and Computer Science

Faculty Member: Dr. Ahmad Salman Dated: 03/03/2023

Semester: 6th Section: BEE 12C

EE-330 Digital Signal Processing

Lab 4: Frequency Response and Pole Zero Plots

Group Members

Name	Reg. No	PLO4 - CLO4		PLO5 - CLO5	PLO8 - CLO6	PLO9 - CLO7
		Viva / Quiz / Lab Performance	Analysis of data in Lab Report	Modern Tool Usage	Ethics and Safety	Individual and Teamwork
		5 Marks	5 Marks	5 Marks	5 Marks	5 Marks
Danial Ahmad	331388					
Muhammad Umer	345834					
Tariq Umar	334943					

1 Table of Contents

2 Frequency Response and Pole Zero Plots of FIR/IIR Filters

2.1 Objectives

In this lab, you will use PeZ to create filters with complex conjugate poles and zeros. These are called second-order filters because the denominator polynomial is a quadratic with two roots.

2.2 Introduction

The purpose of this lab is to explore the design and implementation of second-order filters using PeZ, a software tool commonly used for filter design. In this lab, we will focus on creating filters with complex conjugate poles and zeros, which are essential components in many applications, including signal processing, audio processing, and communications.

Second-order filters are named so because the denominator polynomial is a quadratic with two roots. These filters have two poles and two zeros, which can be placed anywhere in the complex plane to achieve the desired frequency response. By manipulating the location of these poles and zeros, we can create different types of filters such as high-pass, low-pass, band-pass, and band-stop filters.

2.3 Software

MATLAB is a high-level programming language and numerical computing environment. Developed by MathWorks, it provides an interactive environment for numerical computation, visualization, and programming. MATLAB is widely used in various fields, including engineering, science, and finance, due to its capabilities for matrix and vector operations, implementation of algorithms, and creation of graphical representations of data. The objective of this lab is to provide a hands-on experience with the A-to-D sampling and the D-to-A reconstruction processes that are essential for digital image processing. We will also demonstrate a commonly used method of image zooming (reconstruction) that produces "poor" results, which will help illustrate the importance of understanding the underlying principles of digital image processing.

2.4 Lab Report Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

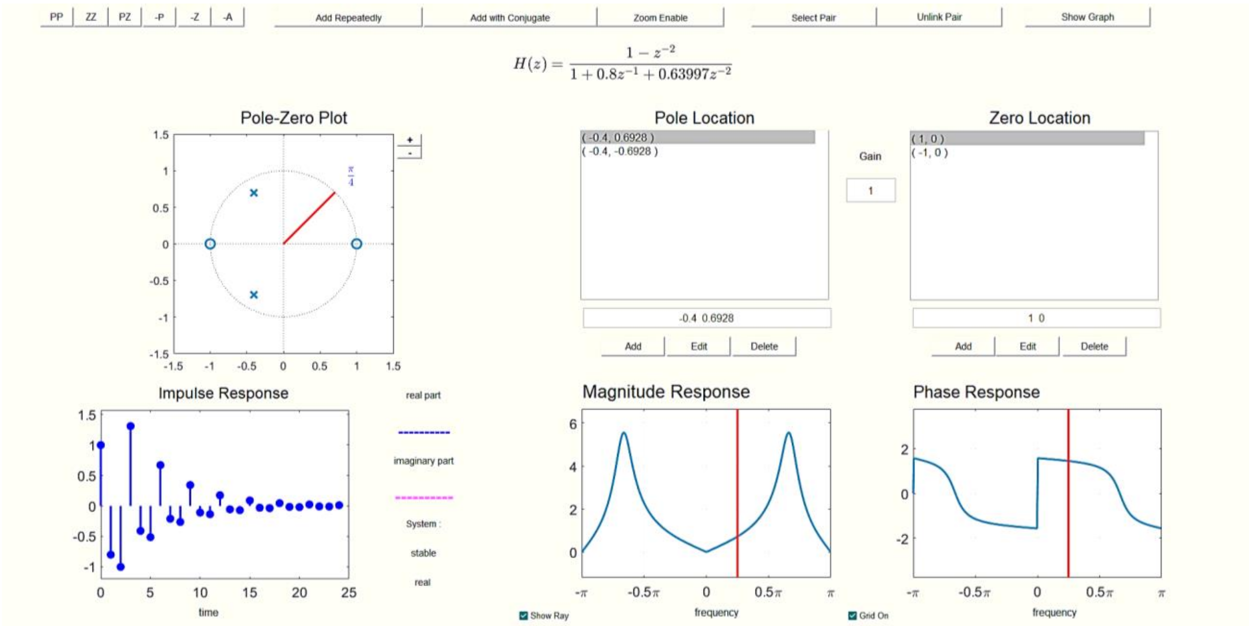
- Lab objectives
- MATLAB codes
- Results (graphs/tables) duly commented and discussed
- Conclusion

3 Lab Procedure

3.1 Lab Task 1: Create an IIR Filter with PeZ

a) Use the PeZ interface to implement the following second-order system by determining where the two poles and two zeros are located and then placing the poles and zeros at the correct locations in the z-plane. Look at the frequency response and determine what kind of filter you have?

```
%% Task 1.a
zeros = roots([1 0 -1]);
display(zeros)
poles = roots([1 0.8 0.64]);
display(poles)

                                    Output

zeros =
-1
1

poles =
-0.4000 + 0.6928i
-0.4000 - 0.6928i
```



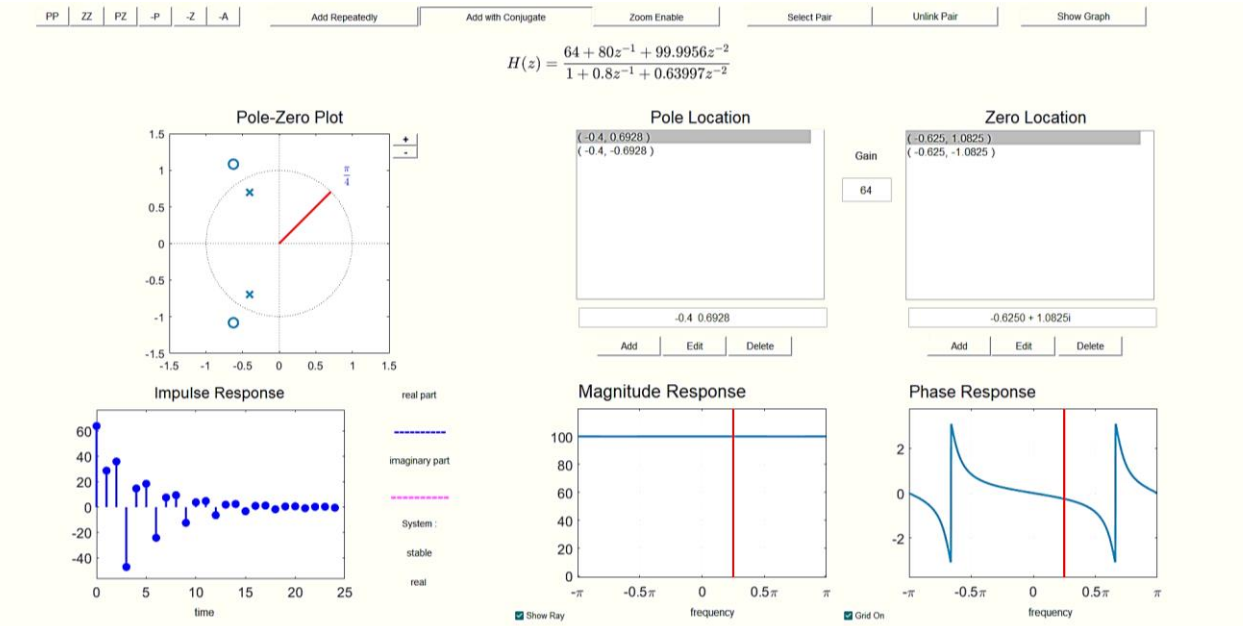
Answer: From the frequency response, we infer that we have created a Band Stop filter.

It is tempting to think that with two poles the frequency response ends up always having a peak, but there are two interesting cases where that doesn't happen: (1) all-pass filters where $|H(e^{j\omega})| =$ constant, and (2) IIR notch filters that null out one frequency but are relatively flat across the rest of the frequency band.

b) Implement the following second-order system by determining where the two poles and two zeros are located and then placing the poles and zeros at the correct locations in the z -plane. Look at the frequency response and determine what kind of filter you have. Describe the type of filter that you have now created.

```
%% Tassk 1.b
zeros = roots([64 80 100]);
display(zeros)
poles = roots([1 0.8 0.64]);
display(poles)

                                        Output

zeros =
-0.6250 + 1.0825i
-0.6250 - 1.0825i

poles =
-0.4000 + 0.6928i
-0.4000 - 0.6928i
```

$$H(z) = \frac{64 + 80z^{-1} + 99.9956z^{-2}}{1 + 0.8z^{-1} + 0.63997z^{-2}}$$



Answer: From the frequency response, we infer that we have created an All-Pass filter.

3.2 Lab Task 2: Bandpass Filter Design for IIR

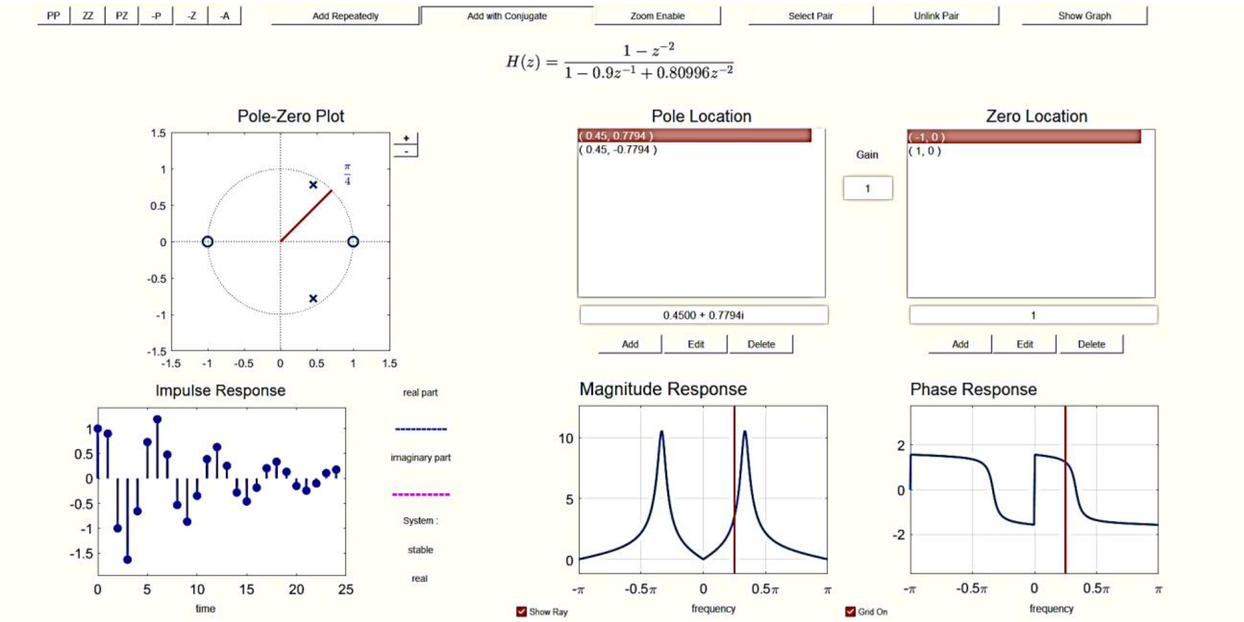
It is easy to design a narrow passband IIR filter by putting a complex pole-pair near the unit circle.

Complex Poles: The first exercise is to move one pole-pair around and obtain formulas for how the frequency response changes as a function of the pole-pair radius and angle.

a) Place following single pole-pairs:

$$z = 0.9e^{\pm j\pi/3}\text{, and zeros at } z = \pm 1.$$

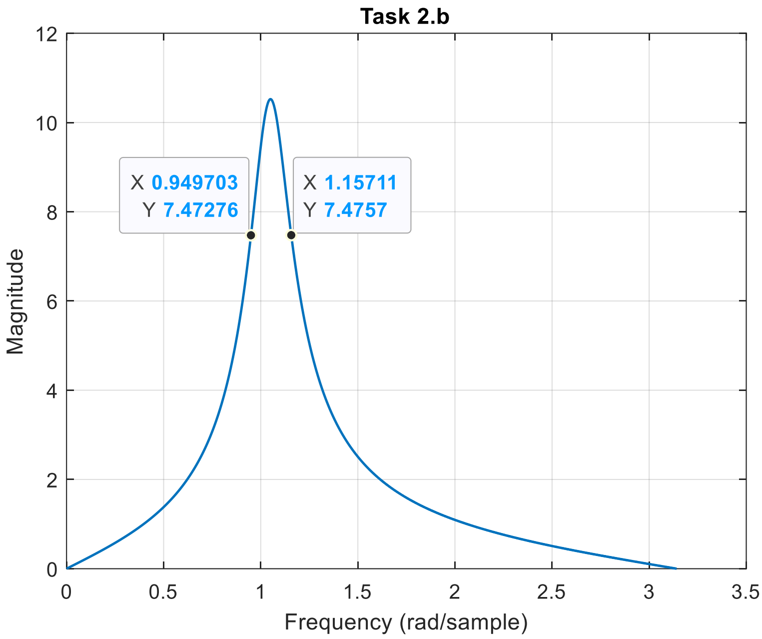
```
%% Task 2.a
pp = 0.9 * exp(1i * pi / 3);
poles = [pp; conj(pp)];
display(poles)
zeros = [1; -1];
display(zeros)

                                   Output

poles =
0.4500 + 0.7794i
0.4500 - 0.7794i

zeros =
1
-1
```



b) Make a plot of the frequency response (magnitude only) with freqz and measure the width of the peak versus frequency. This presents a problem because we must define how to measure width. The usual definition is to measure the width at the “3-dB level.” To do this, the measurement must be made with respect to the peak value of the frequency response. If the peak value is H_{max}, then the “3-dB level” is at 0.707 H_{max}.

```
%% Task 2.b
b = [1 0 -1];
a = [1 -0.9 0.80996];
[h, w] = freqz(b, a, 8096);

figure
plot(w, abs(h), 'LineWidth', 1.15)
```

```
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
title('Task 2.b')
grid

db3_level = max(abs(h) * 0.707);
display(db3_level)
                                        Output

db3_level =
7.4405
```

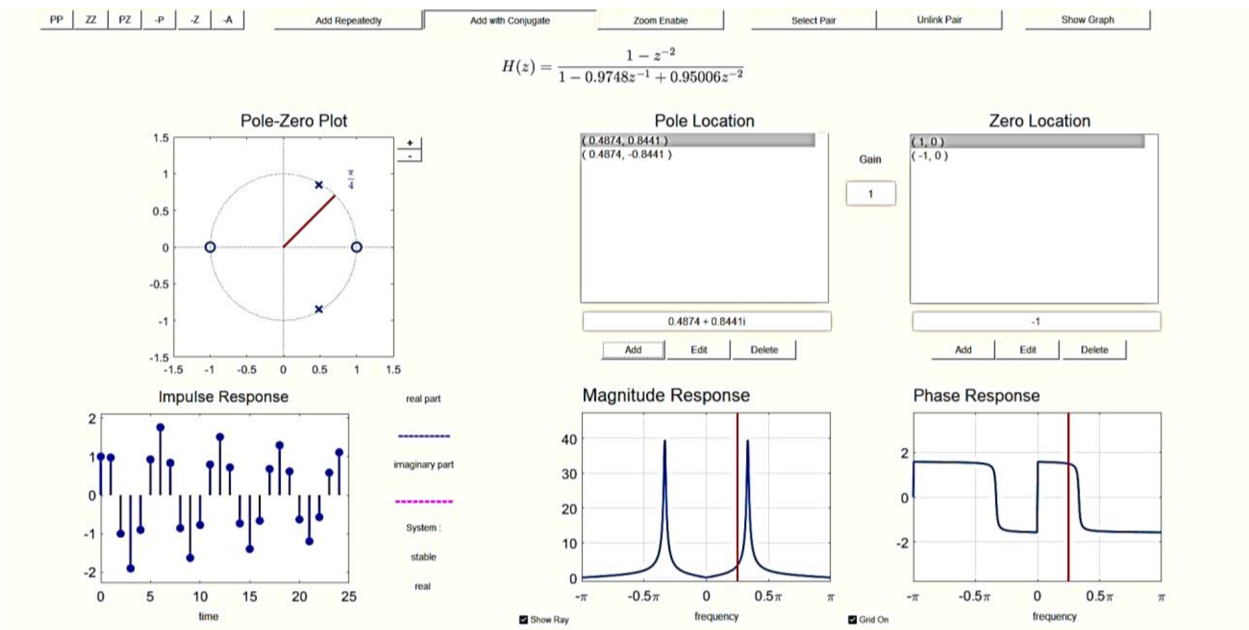


3-dB Level Width = X2 – X1 = 1.15711 – 0.949703 = 0.207407

c) Move the pole-pair so that the angles remain fixed at $\pm\pi/3$, but the radius is $r = 0.95$ and $r = 0.975$. In each case, measure the 3-db width of the peak. Using these measured values, create a formula for the width that is proportional to $(1 - r)$.

```
%% Task 2.c
pp = 0.975 * exp(1i * pi / 3);
poles = [pp; conj(pp)];
display(poles)

b = [1 0 -1];
a = [1 -0.9748 0.95006];
[h, w] = freqz(b, a, 8096);

figure
plot(w, abs(h), 'LineWidth', 1.15)
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
title('Task 2.c')
grid
```

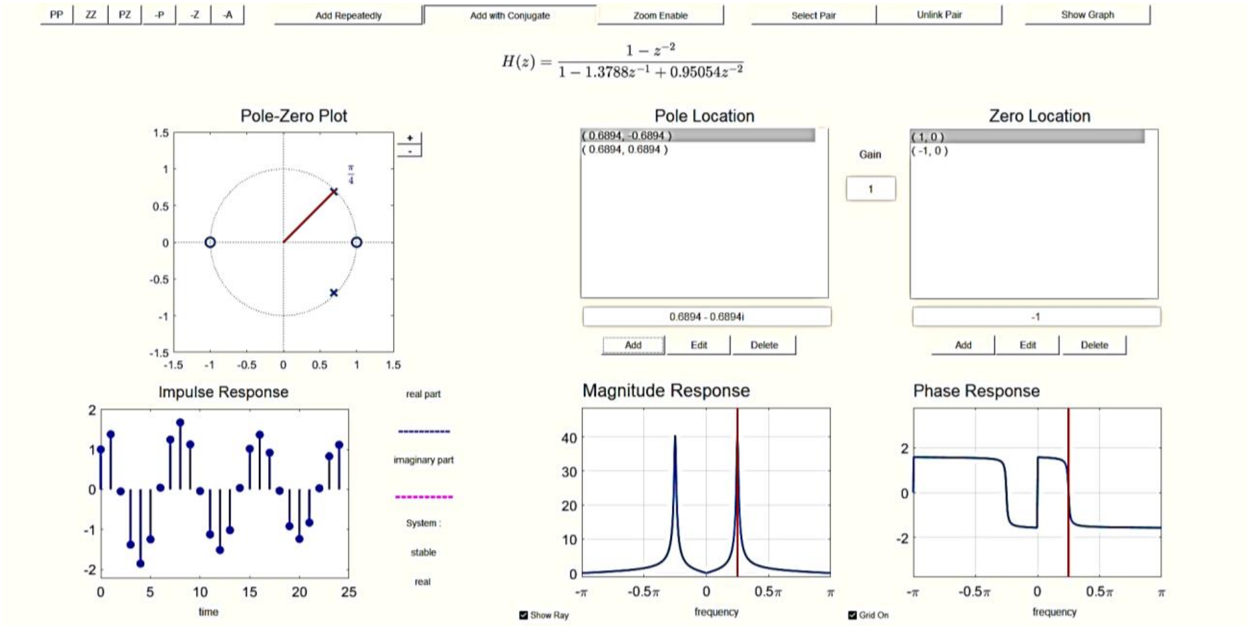
```
db3_level = max(abs(h) * 0.707);
display(db3_level)

                                Output
poles =
0.4875 + 0.8444i
0.4875 - 0.8444i

db3_level =
28.3140
```

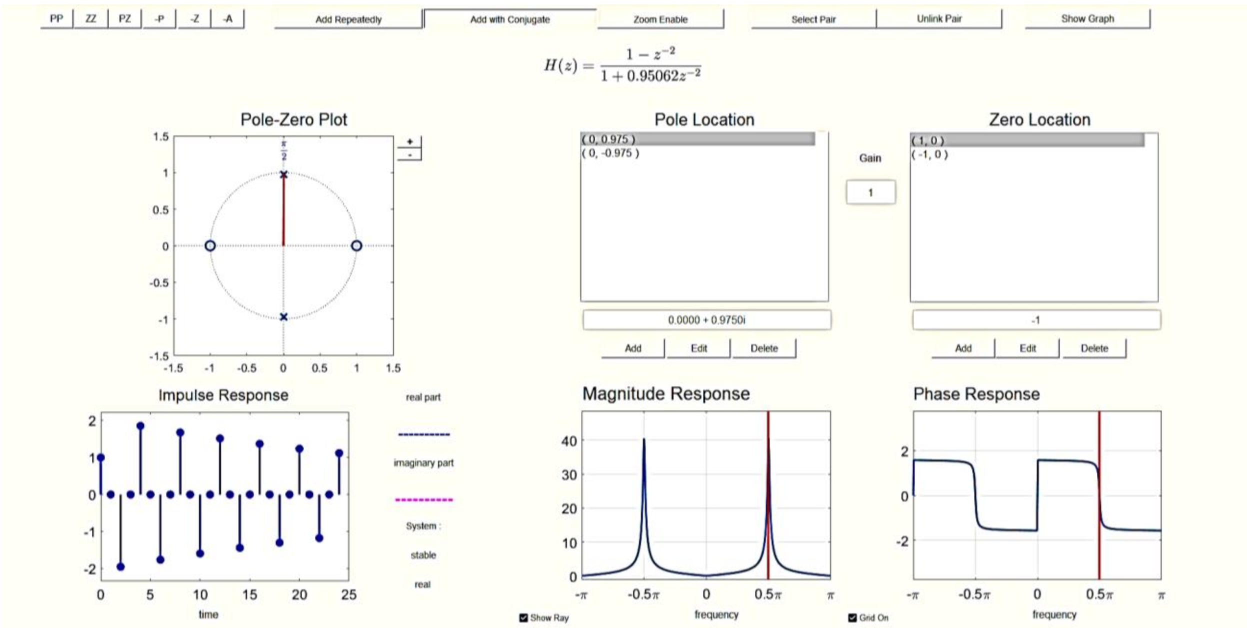


3-dB Level Width = X2 – X1 = 1.15711 – 0.949703 = 0.207407
Peak Width ≈ $K\,\frac{1-r}{\sqrt{r}}$, where K is a constant of proportionality; **[K = 1.967]**

d) Move the pole-pair so that its radius remains fixed, and the angles change from ±π/3 to ±π/4 and then to ±π/2. State a formula for the peak location as a function of the pole location.

```
%% Task 2.d
pp = 0.975 * exp(1i * pi / 4);
poles_a = [pp; conj(pp)];
display(poles_a)

pp = 0.975 * exp(1i * pi / 2);
poles_b = [pp; conj(pp)];
display(poles_b)

                                Output
poles_a =
0.6894 + 0.6894i
0.6894 - 0.6894i

poles_b =
0.0000 + 0.9750i
0.0000 - 0.9750i
```



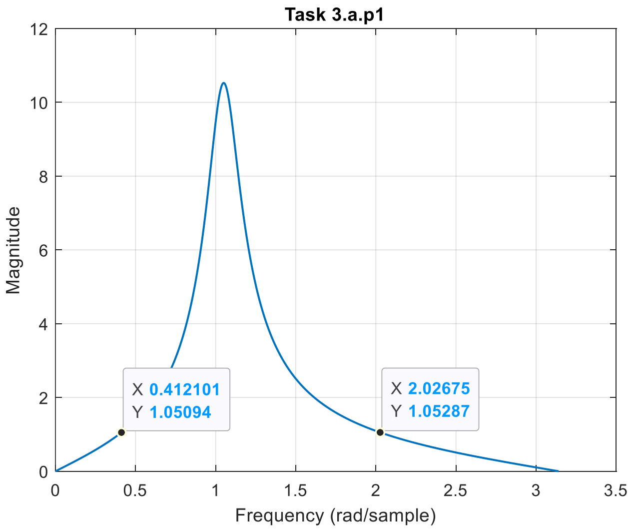


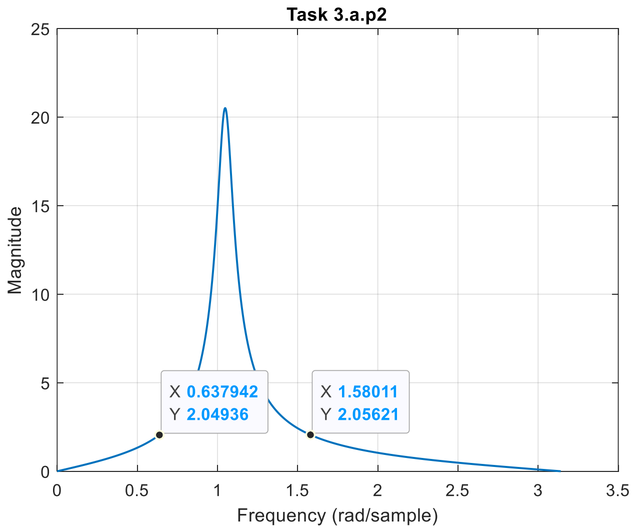
Peak Location = $\pm \tan^{-1}(\frac{y}{x})$; where y and x are the imaginary and real axis points respectively.

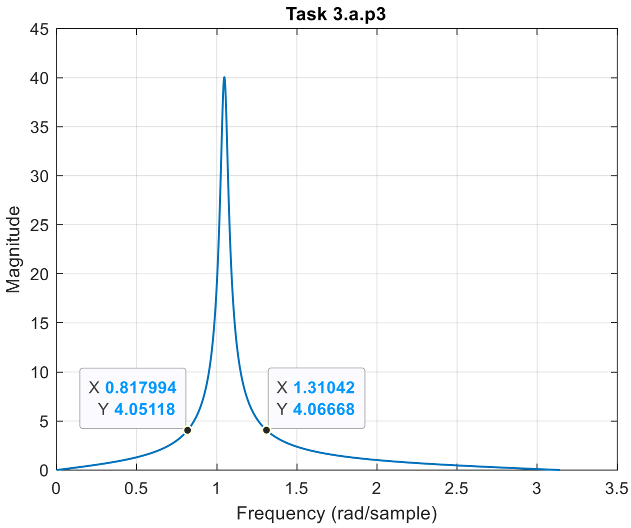
Passband and Stopband: We can characterize general bandpass filters if we define the passband width to be equal to the 3-dB width (as in the previous section). We also need a definition for the stopband, and we will arbitrarily define the stopbands of a BPF to be those regions where the frequency response (magnitude) is below −20 dB, which is equivalent to 10% of the peak value.

a) Determine the stopband regions for three of the filters designed in the previous section. Use the cases where the pole angles are $\pm\pi/3$ and the radii are r = 0.9, 0.95 and 0.975. In each case, measure the frequency regions of the two stopbands.

```
%% Task 3.a.p1
b = [1 0 -1];
a = [1 -0.9 0.80996];
[h, w] = freqz(b, a, 8096);

figure
plot(w, abs(h), 'LineWidth', 1.15)
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
title('Task 3.a.p1')
grid

%% Task 3.a.p2
b = [1 0 -1];
a = [1 -0.95 0.90246];
[h, w] = freqz(b, a, 8096);

figure
plot(w, abs(h), 'LineWidth', 1.15)
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
title('Task 3.a.p2')
grid

%% Task 3.a.p3
b = [1 0 -1];
a = [1 -0.9748 0.95006];
[h, w] = freqz(b, a, 8096);

figure
plot(w, abs(h), 'LineWidth', 1.15)
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
title('Task 3.a.p3')
grid
```







b) For the same three filters, record the passband edges. The passband will be the peak width at the 3-dB level, so it will occupy a region such as $\omega_1 \leq \omega \leq \omega_2$, where ω_1 and ω_2 are the band edges.

Answer: Already done in Lab Task 1.

c) Usually, filter design becomes difficult when we want the passband and stopband edges to be very close to one another. The difference between neighboring passband and stopband edges is called the Transition Width. Therefore, summarize the measurements of the previous two parts in a table that lists the two transition widths for each filter versus r. Does it depend on r?

Yes, it depends on r. More specifically,

$$r \propto transition\ width$$

4 Conclusion

In this lab, we used PeZ to design and implement second-order filters with complex conjugate poles and zeros. Through this exercise, we gained a better understanding of the behavior of these types of filters and how they can be used to shape the frequency response of a signal. Through these exercises, we saw firsthand how the placement of poles and zeros can significantly impact the frequency response of a filter. We also learned how to use PeZ to design and implement these filters, which is a valuable skill in signal processing and other related fields.