UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI KEJURUTERAAN
ELEKTRIK DAN ELEKTRONIK
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DIGITAL SIGNAL PROCESSING

BEET3373	SEMESTER 1	SESI 2021/2022

LAB 2: Z-TRANSFORM

NO.	STUDENTS' NAME	MATRIC. NO.
1.	AHMAD IRFAN BIN HARMAN	B081910068
2.	MOHAMED HAZEEM BIN HASHAINI	B081910350
3.	MUHAMAD AZIM HAMZI BIN AZAHA	B081910086
4.	RAHMAN KAZI ASHIKUR	B081910450

PROGRAMME	3 BEEC
SECTION / GROUP	3BEEC S1/1
DATE	14/11/2021
NAME OF INSTRUCTOR(S)	1. DR.jamil
	2.

EXAMINER'S COMMENT(S)	TOTAL MARKS

LAB 2: Z-TRANSFORM

1.0 OBJECTIVES

Student able:

1. To understand basic digital signal frequency characteristics and analysis
2. To understand and verify the basic concepts of Z-transform and the relationship between the system transfer function to the frequency response
3. To understand the effects of zeros and poles in the system transfer function and the frequency response

2.0 EQUIPMENT

Hardware	Type/Version
1. Workstation (Computer)	Windows 7
2. MATLAB	R2013

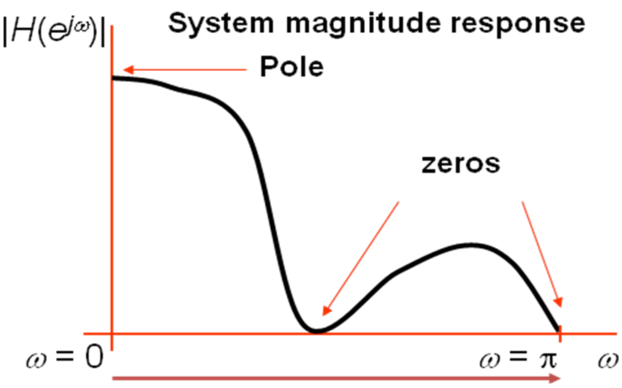
3.0 SYNOPSIS & THEORY

The Z-transform for discrete-time signals is the counterpart for Laplace transform for continuous-time signals. Fourier transform does not converge for all sequences; however, the Z-transform converges for a broader class of signals and the Z-transform notation is easier to handle analytically. For any given sequence, the set of values of *z* for which the Z-transform converges is called the region of convergence (ROC). The most useful form is when *X(z)* is a rational function inside the ROC.

$$X(z) = \frac{P(z)}{Q(z)}$$

, where *P(z)* and *Q(z)* are polynomials in *z*.

The values of *z* for which ***X(z) = 0*** are called the **zeros** of *X(z),* whereas the values of *z* for which *X(z)* is **infinite** are called the **poles** of *X(z)*. The poles of *X(z)* for finite values of *z* are the roots of the denominator polynomial. The poles and zeros play an important role in a system filtering effect. A digital filter can be designed by creating zeros and poles at appropriate positions on the z-plane.



Place zeros at positions (close to or on the unit circle) where you wish to have low or zero values for the system transfer function. This method can be applied when designing a **low pass filter**. For a **high pass filter**, poles can be placed at locations (close to but not on the unit circle) where you wish to have high values for the system transfer function. Each of the zeros or poles created must be placed at positions mirroring one of the other along the real axis of the z-plane.

4.0 PROCEDURE

1. Z-Transform, System Transfer Function, and Frequency Response

1. The systems with input-output relationships described below are to be analyzed.

System 1: $y_1[n] = \frac{1}{M} \cdot \sum_{k=0}^{M-1} x[n-k]$ where $M = 5$

System 2: $y_2[n] = \frac{1}{M} \cdot \sum_{k=0}^{M-1} (-1)^k \, x[n-k]$ where $M = 5$

System 3: $y_3[n] = x[n] + a \cdot y_3[n-1]$ where $a = -0.5$

Exercises

a. Determine the **transfer functions** of the three systems by performing the Z-transform on their impulse responses. Assume that all systems have zero initial condition while performing the Z-transform.

$$H_1(z) = \sum_{n=-\infty}^{\infty} h_1[n] z^{-n} = 0.2 + 0.2z^{-1} + 0.2z^{-2} + 0.2z^{-3} + 0.2z^{-4}$$

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n} = 0.2 - 0.2z^{-1} + 0.2z^{-2} - 0.2z^{-3} + 0.2z^{-4}$$

$$H_3(z) = \frac{1}{1 + 0.5z^{-1}}$$

b. Find the **roots** of the transfer functions above. You may want to use the built-in mathematical functions in MATLAB such as the **`roots` function**. Write the system transfer functions in the form of factors based on the roots determined.

   (Hint: You may refer to Appendix for information about the `roots` function.) Code example: **`zpart1.m`**

c. What would be the effects on the system transfer functions when we set z^{-1} to one of the corresponding roots determined in the answer above? Categorize the roots. You can try to use `tf2zpk` function to check the values of the zeros and poles.

```
r1 =
  0.3090 + 0.9511i
  0.3090 - 0.9511i
 -0.8090 + 0.5878i
 -0.8090 - 0.5878i

p1 =
    0
    0
    0
    0

k1 =
   0.2000

sys =

 0.2 z^4 + 0.2 z^3 + 0.2 z^2 + 0.2 z + 0.2
 ---------------------------------------
                  z^4
```

```
Sample time: 0.1 seconds
Discrete-time transfer function.

r2 =
 -0.3090 + 0.9511i
 -0.3090 - 0.9511i
  0.8090 + 0.5878i
  0.8090 - 0.5878i

p2 =
    0
    0
    0
    0

k2 =
   0.2000

sys =

 0.2 z^4 - 0.2 z^3 + 0.2 z^2 - 0.2 z + 0.2
 -----------------------------------------
                   z^4

Sample time: 0.1 seconds
Discrete-time transfer function.

r3 =
    0

p3 =
  -0.5000

k3 =
    1

sys =

    z
 -------
 z + 0.5

Sample time: 0.1 seconds
```

2. Effects of Zeros and Poles and Their Relationship to the System Filtering Effect

1. Plot the positions of the zeros and poles of Systems 1, 2, and 3 from Section 2.
   (Hint: You may refer to Appendix for the use of the `zplane` function to plot the zeros and poles.)
   Code example: **`zpart2.m`**

2. Open file named **`zpart3.m`** to generate and plot the frequency responses (magnitude) of Systems 1, 2, and 3. The `abs()` functions is to find the magnitudes of the zeros. What can you conclude about the filtering effects of Systems 1, 2, and 3?

   (Hint: Observe the locations of the poles and zeros.)

   In signal processing, a **finite impulse response** (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time.
   Infinite impulse response (**IIR**) filter does not become exactly zero past a certain point but continues indefinitely.
   FIR (Finite Impulse response) filters contain only zeros and no poles (except at origin). If a filter contains poles, it is IIR (Infinite Impulse response). IIR filters are indeed afflicted with stability issues and must be handled with care.

   so Filtering effects of system 1,2,3 are as follows:
   1) Observing the frequency plot of system 1,2,3
   System 1 is a low pass filter as lower frequencies have high magnitude response and at higher frequencies filter has low magnitude response, whereas system 2,3 are high pass filters.
   2) system 1,2 are FIR filters, on the other hand system 3 is an IIR filter and a stable filter as pole is inside the unit circle with negative real part.

Exercises

a. For Systems 1 and 2, determine the angles and magnitudes associated to the zeros. Compare the angles with the spectral points in the magnitude responses at which the magnitudes are of zero value. (Hint: You may need to use the `angle()` and `abs()` functions to find the angles and magnitudes of the zeros respectively. Normalize the angles you find by π for the comparison of spectral points in the magnitude responses). Refer again code example: **`zpart3.m`**

b. Compare and analyze the impulse responses and frequency responses of System 1 and System 2. What can you conclude about the relationship between the impulse responses of the low pass and high pass filters, both having magnitude responses that are mirror image to each other at the normalized frequency of 0.5? How can we make use of such a relationship in filter design?

   (Hint: Impulse responses are directly proportional to the filter's effect.)

   system 1,2 are FIR filters, System 1 is a low pass filter as lower frequencies have high magnitude response and at higher frequencies filter has low magnitude response; where as system 2,3 are high pass filters.

c. Based on the understanding of the effects of zeros and poles on the frequency response of a system, design the following filters.

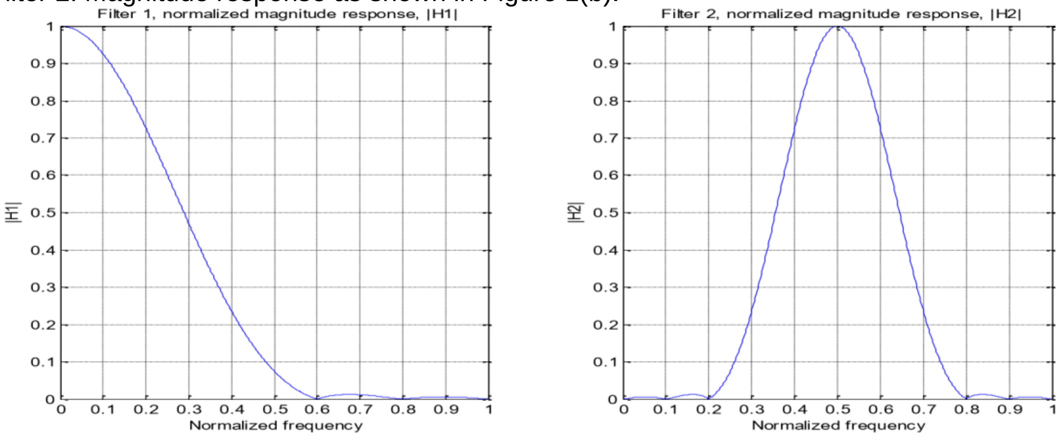
   You must design your filter with the minimum number of zeros/poles. Start your design by observing the zero-level spectral points and de-normalize them with π. (Hint: You may need to apply **Euler's formula cosθ + jsinθ** and the **MATLAB function `poly()`** to design the filters.)

   Code example: **`zpart4.m`**

Determine the **zeros/poles** of the filters. Plot the **magnitude responses** and **z-plane** on the same figure.

Filter 1: Magnitude response as shown in Figure 1(a).

Filter 2: Magnitude response as shown in Figure 1(b).



(a) (b)

Figure 1 – Frequency Response

5.0 EXPERIMENT RESULT

(Please attached all the graphs and codings obtained from the Procedure)

Part 1.a

Calculation

S1

S1

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k], \quad M = 5$$

$$y(n) = \frac{1}{5}[x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4)]$$

$$Y(z) = \frac{1}{5}[x(z) + x(x^{-1}) + x(z^{-2}) + x(z^{-4})]$$

$$\frac{Y(z)}{X(z)} = \frac{1}{5}[1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}]$$

S2

$$\underline{\underline{S_2}}$$

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} (-1)^k x[1-k], \quad M = 5$$

$$y(n) = \frac{1}{5} \sum_{k=0}^{4} (-1)^k x[n-k], \quad M = 5$$

$$y(n) = \frac{1}{5} \left[x(n) - x(n-1) + x(n-2) - x(n-3) + x(n-4) \right]$$

$$y(z) = \frac{1}{5} \left[x(z) - x(z^{-1}) + x(z^{-2}) - x(z^{-3}) + x(z^{-4}) \right]$$

$$\frac{y(z)}{x(z)} = \frac{1}{5} \left[1 - z^{-1} + z^{-2} - z^{3} + z^{-4} \right]$$

S3

$$\underline{\underline{S_3}}$$

$$y(n) = x(n) + a \cdot y(n-1), \quad a = 5$$

$$y(n) = x(n) - 0.5 y(n-1)$$

$$y(n) + 0.5 y(n-1) = x(n)$$

$$y(z) + 0.5 y(z^{-1}) = x(z)$$

$$\frac{y(z)}{x(z)} = \frac{1}{1 + 0.5 z^{-1}}$$

1.b

Coding with results.

S1

System 1

```
b1=[0.2 0.2 0.2 0.2 0.2];     %Coefficients (Num.) of transfer function System 1
a1=[1 0 0 0 0];       %Coefficients (Den.) of transfer function System 1
root1_num= roots(b1)         %Use roots function root1_den= roots(a1)
root1_Den= roots(a1)
```

output

```
root1_num = 4×1 complex
     0.3090 + 0.9511i
     0.3090 - 0.9511i
    -0.8090 + 0.5878i
    -0.8090 - 0.5878i

root1_Den = 4×1
     0
     0
     0
     0
```

S2

System 2

```
b2=[0.2 -0.2 0.2 -0.2 0.2];    %Coefficients (Num.) of transfer function System 1
a2=[1 0 0 0 0];       %Coefficients (Den.) of transfer function System 1
root2_num= roots(b2)         %Use roots function root1_den= roots(a1)
root2_Den= roots(a2)
```

output

```
root2_num = 4×1 complex
    -0.3090 + 0.9511i
    -0.3090 - 0.9511i
     0.8090 + 0.5878i
     0.8090 - 0.5878i

root2_Den = 4×1
     0
     0
     0
     0
```

S3

System 3

```
b3=[1 0];                    %Coefficients (Num.) of transfer function System 2
a3=[1 0.5];                    %Coefficients (Den.) of transfer function System 2
root3_num=roots(b3)            %Use roots function root2_Num=roots(b3)
 root3_Den=roots(a3)            % %Use roots function  root2_den=roots(a3)
```

output

```
root3_num = 0
root3_Den = -0.5000
```

1.c

```
 %System 1
 b1=[0.2 0.2 0.2 0.2 0.2];
a1=[1 0 0 0 0];
root1_num= roots(b1);
root1_Den= roots(a1);

%system2
b2=[0.2 -0.2 0.2 -0.2 0.2];
a2=[1 0 0 0 0];
root2_num= roots(b2) ;
root2_Den= roots(a2);

%System 3
b3=[1 0];
a3=[1 0.5];
root3_num=roots(b3) ;
 root3_Den=roots(a3);

 [r1,p1,k1]=tf2zpk(b1,a1)
 [r2,p2,k2]=tf2zpk(b2,a2)
 [r3,p3,k3]=tf2zpk(b3,a3)
```

Output

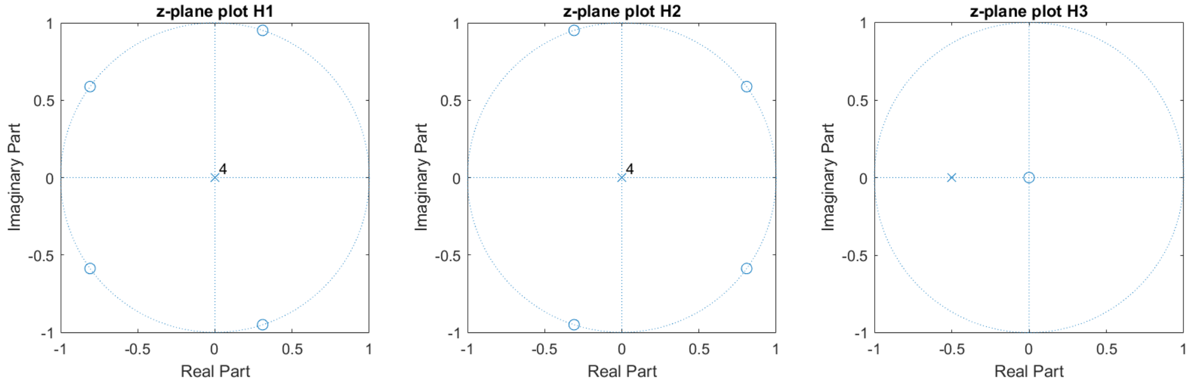
```
r1 = 4×1 complex
      0.3090 + 0.9511i
      0.3090 - 0.9511i
     -0.8090 + 0.5878i
     -0.8090 - 0.5878i

p1 = 4×1
        0
        0
        0
        0

k1 = 0.2000
r2 = 4×1 complex
     -0.3090 + 0.9511i
     -0.3090 - 0.9511i
      0.8090 + 0.5878i
      0.8090 - 0.5878i

p2 = 4×1
        0
        0
        0
        0

k2 = 0.2000
r3 = 0
p3 = -0.5000
k3 = 1
```

PART 2.1

Coding

```
% zpart2.m
%System 1
 b1=[0.2 0.2 0.2 0.2 0.2];
a1=[1 0 0 0 0];
root1_num= roots(b1);
root1_Den= roots(a1);

%system2
b2=[0.2 -0.2 0.2 -0.2 0.2];
a2=[1 0 0 0 0];
root2_num= roots(b2) ;
root2_Den= roots(a2);

%System 3
b3=[1 0];
a3=[1 0.5];
root3_num=roots(b3) ;
 root3_Den=roots(a3);

 %%Plotting   of zeros and poles at z-plane
 %System 1
 subplot(1,3,1);
 zplane(root1_num,root1_Den);
 title('z-plane plot H1');
 axis ([-1 1 -1 1]);
  %System 2
 subplot(1,3,2);
 zplane(root2_num,root2_Den);
 title('z-plane plot H2');
 axis ([-1 1 -1 1]);
  %System 3
 subplot(1,3,3);
 zplane(root3_num,root3_Den);
 title('z-plane plot H3');
 axis ([-1 1 -1 1]);
```

Output



2.2
Coding

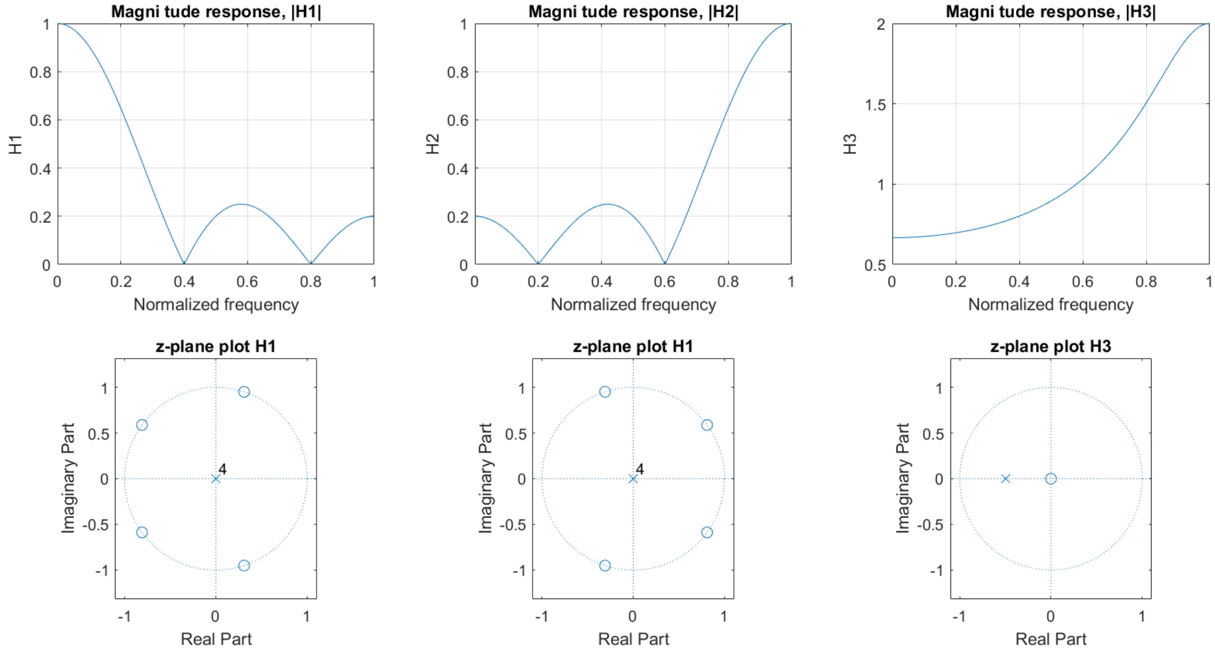
```
N=1024;
%System 1
 b1=[0.2 0.2 0.2 0.2 0.2];  % coefficients (Num.) of transfer function system 1
a1=[1 0 0 0 0];                % Coefficients (Den.) of transfer function System
1
root1_num= roots(b1);                 % Using roots function
root1_Den= roots(a1);           % Using roots function
[H1,W1]=freqz(b1,a1,N);           % Use freqz function: [H, W] = FREQZ (B, A, N) ;
with N=1024
```

```
%system2
b2=[0.2 -0.2 0.2 -0.2 0.2];
a2=[1 0 0 0 0];
root2_num= roots(b2) ;
root2_Den= roots(a2);
[H2,W2]=freqz(b2,a2,N);
```

```
%System 3
b3=[1 0];
a3=[1 0.5];
root3_num=roots(b3) ;
 root3_Den=roots(a3);
 [H3,W3]=freqz(b3,a3,N);


 %% z-plane and magnitude response plots

 %System 1
 subplot(2,3,1)
 plot (W1/pi, abs(H1))
```

```
; %use subplot
 title ('Magni tude response, |H1|');
 xlabel ('Normalized frequency');
 ylabel ('H1');
 grid on;
  subplot(2,3,4)
 zplane(root1_num,root1_Den); %use subplot
 title ('z-plane plot H1');

 %system2
 subplot(2,3,2)
 plot (W2/pi, abs(H2)); %use subplot
 title ('Magni tude response, |H2|');
 xlabel ('Normalized frequency');
 ylabel ('H2');
 grid on;
  subplot(2,3,5)
 zplane(root2_num,root2_Den); %use subplot
 title ('z-plane plot H1');

  %system3
 subplot(2,3,3)
 plot (W3/pi, abs(H3)); %use subplot
 title ('Magni tude response, |H3|');
 xlabel ('Normalized frequency');
 ylabel ('H3');
 grid on;
  subplot(2,3,6)
 zplane(root3_num,root3_Den); %use subplot
 title ('z-plane plot H3');
```

Output



2 C

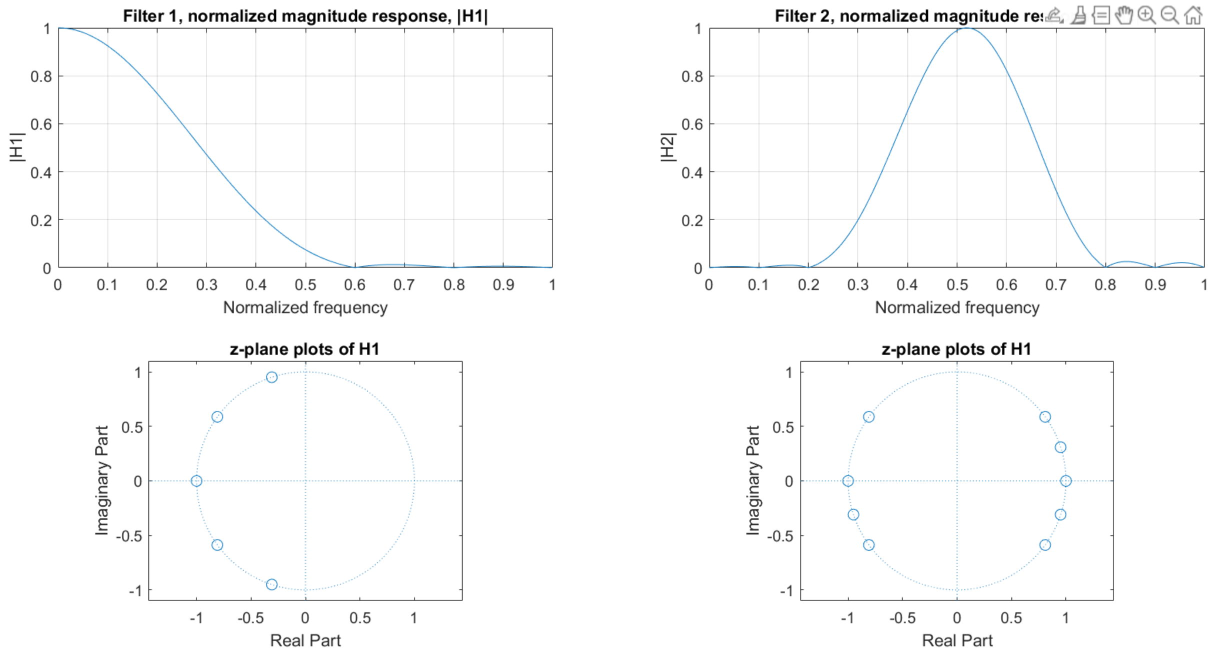
```
% zpart4.m clear; close all;
N = 1024;
%Filter 1
Z1_ang = [-1 0.8 -0.8 0.6 -0.6]*pi; %Denormalized with pi
Z1=cos(Z1_ang)+j*sin(Z1_ang);% Z1 = ??? %Use formula cosɵ + jsinɵ
h1 = poly(Z1); %Use poly function %To find the coefficients of the characteristic
polynomial of a matrix Z1
[H1, W1] = freqz(h1, N);
H1 = H1 / max(abs(H1));
n1 = [0:length(h1)-1];
```

```
%Filter 1
Z2_ang=[0 1 0.1 -0.1 0.9 0.2 -0.2 0.8 -0.8]*pi;
Z2=cos(Z2_ang)+j*sin(Z2_ang);
h2 = poly(Z2);
[H2, W2] = freqz(h2, N);
H2 = H2 / max(abs(H2));
n2 = [0:length(h2)-1];
```

```
%Plottings figure;
% %Filter 1
subplot(2,2,1);
plot(W1/pi, abs(H1));
set(gca,'xtick',[0:0.1:1]);
grid on;
title('Filter 1, normalized magnitude response, |H1|');
xlabel('Normalized frequency');
ylabel('|H1|');
subplot(2,2,3);
zplane(Z1', []);
title('z-plane plots of H1');
```

```
% %Filter 1
subplot(2,2,2);
plot(W2/pi, abs(H2));
set(gca,'xtick',[0:0.1:1]);
grid on;
title('Filter 2, normalized magnitude response, |H2|');
xlabel('Normalized frequency');
ylabel('|H2|');
subplot(2,2,4);
zplane(Z2', []);
title('z-plane plots of H2');
```

Output



6.0 QUESTION & DISCUSSION

Answer all Questions in Procedure (Exercise) and discuss each task based on Experiment Data and Experiment Result.

1.

d. What would be the effects on the system transfer functions when we set z^{-1} to one of the corresponding roots determined in the answer above? Categorize the roots. You can try to use `tf2zpk` function to check the values of the zeros and poles.

```
r1 =
  0.3090 + 0.9511i
  0.3090 - 0.9511i
 -0.8090 + 0.5878i
 -0.8090 - 0.5878i

p1 =
    0
    0
    0
    0

k1 =
   0.2000

sys =

 0.2 z^4 + 0.2 z^3 + 0.2 z^2 + 0.2 z + 0.2
 -----------------------------------------
```

```
                    z^4

Sample time: 0.1 seconds
Discrete-time transfer function.

r2 =
 -0.3090 + 0.9511i
 -0.3090 - 0.9511i
  0.8090 + 0.5878i
  0.8090 - 0.5878i

p2 =
    0
    0
    0
    0

k2 =
   0.2000

sys =

 0.2 z^4 - 0.2 z^3 + 0.2 z^2 - 0.2 z + 0.2
 -----------------------------------------
                    z^4

Sample time: 0.1 seconds
Discrete-time transfer function.

r3 =
    0

p3 =
  -0.5000

k3 =
    1

sys =

    z
 -------
 z + 0.5

Sample time: 0.1 seconds
```

2.2 Open file named zpart3.m to generate and plot the frequency responses (magnitude) of Systems 1, 2, and 3. The abs() functions is to find the magnitudes of the zeros. What can you conclude about the filtering effects of Systems 1, 2, and 3? (Hint: Observe the locations of the poles and zeros.

Its Pole-Zero Plot displays the filter's transfer function's Z-domain poles and zeros. Poles are denoted by "X," while zeros are denoted by "O." Real values are represented along the horizontal axis and imaginary values are represented along the vertical axis in the Z-plane. The "unit circle" is represented by a circle with a diameter of 1.0.

System 1:
System 1 can be considered a low pass filter. The pole angle dictates filter frequency, and the pole radius dictates Q. Pole at zero means that the system is marginally stable. The two zeroes lie in the negative value of the real part indicates that it is pulling down the response from the highest frequency and two zeroes lies on the positive part and close to origin means that the response increases from zero frequency.

System 2:
I believe system 2 is the opposite of system 1. It can be a high pass filter. Pole at zero means that the system is marginally stable. Two zeroes lie in the negative part means there is a fluctuation of response which in low frequency and two zeroes lies in the positive value of the real part means that it is pulling up the response from lowest frequency.

System 3:
System 3 is a high pass filter. A zero at the origin (corresponding to a pure differentiation) implies that the system has zero gain at zero frequency. A pole lies in the negative part indicates that it is pulling up the response from low frequency to the highest frequency.

The location of zeros corresponds to the frequencies that will be attenuated while the location of poles corresponds to the frequencies that will be amplified.

For System 1, the zeros are in the middle to high frequencies (0.4 and 0.8 normalized frequency), which means that the middle and high frequencies will be attenuated and filtered. The poles will only have a minimal effect because they are in the origin. Thus, System 1 is a lowpass filter.

For System 2, the zeros are located in the low to middle frequencies (0.2 and 0.6 normalized frequency), which means that the low and middle frequencies will be attenuated and filtered. The poles will only have a minimal effect because they are in the origin. Thus, System 2 is a high pass filter.

For System 3, the poles are in the high frequency (normalized frequency of 1), which means that the high frequencies will be amplified. The zeros in this case will only have a minimal effect because they are in the origin. Thus, System 3 is a high frequency amplifier.

System 1:

System 1 is a moving average filter with length $M = 5$ i.e., the output at any time instance t is the average of the previous five input instances i.e., $y(t) = [x(t) + x(t-1) + x(t-2) + x(t-3) + x(t-4)]/5$. This kind of filtering smooths down the ripples in the inputs and thus acts like a low pass filter.

System 2:

System 2 is just like system 1 but here the previous input instances have different weights in the average and thus this is known as weighted moving average filter. In the given system, the weights are either -1 or 1 i.e., the odd time instanced inputs are given lower weights (-1) and even instanced inputs are given more weights (1). This type of filter intentionally introduces a lot of ripples and thus acts like a high pass filter.

System 3:

System 3 is a first order Auto regressive filter. These kinds of filters require memory for physical implementation and the output at any time instance is dependent on the previous output. This creates a regression like effect in the output and thus gives meaning to its name. These kinds of filters have a special importance in digital signal processing as they implement infinite impulse response filters which have a non-zero pole. Further, this filter in particular will be a high pass filter, but it will be different from system 2 because it will have a very smooth frequency response whereas system 2 will have ripples in its frequency response.

a. Compare and analyse the impulse responses and frequency responses of System 1 and System 2. What can you conclude about the relationship between the impulse responses of the low pass and high pass filters, both having magnitude responses that are mirror image to each other at the normalized frequency of 0.5? How can we make use of such a relationship in filter design?

   (Hint: Impulse responses are directly proportional to the filter's effect.)

System 2 is just like system 1 but here the previous input instances have different weights in the average and thus this is known as weighted moving average filter. In the given system, the weights are either -1 or 1 i.e., the odd time instanced inputs are given lower weights (-1) and even instanced inputs are given more weights (1). This type of filter intentionally introduces a lot of ripples and thus acts like a high pass filter.

system 1,2 are FIR filters, System 1 is a low pass filter as lower frequencies have high magnitude response and at higher frequencies filter has low magnitude response, whereas system 2,3 are high pass filters.

So, we can convert high pass signal from low pass like System 2

7.0 CONCLUSION

<u>In this lab we able to understand basic digital signal frequency characteristics and analysis As well as understand and verify the basic concepts of Z-transform and the relationship between the system transfer function to the frequency response.</u>

Also, we understand the effects of zeros and poles in the system transfer function and the frequency response

Appendix

Z-Transform, Plotting of Roots in MATLAB

Assuming that we have a transfer function, which is in the *z*-domain as given below:

$$X(z) = \frac{z^{-1}}{1 - 0.25z^{-1} - 0.375z^{-2}} = \frac{z}{z^2 - 0.25z - 0.375}$$

Perform a partial fraction on the transfer function below:

$$X(z) = \frac{z}{(z-0.75)(z+0.5)} = \frac{c_1 z}{(z-0.75)} + \frac{c_2 z}{(z+0.5)} = \frac{0.8z}{(z-0.75)} + \frac{-0.8z}{(z+0.5)}$$

The partial fraction of the transfer function above can be found using the `residuez` function in MATLAB.

```
>> a = [0 1];
b=[1 -0.25 -0.375]; >>
[r,p,k] = residuez(a,b)
 r =
    0.8000
-0.8000

 p =
0.7500
-0.5000
 k =
      []
```

From the calculation in MATLAB, we find that the following partial fraction of the transfer function is:

$$X(z) = \frac{z^{-1}}{1 - 0.25z^{-1} - 0.375z^{-2}} = \frac{0.8z^{-1}}{1 - 0.75z^{-1}} + \frac{-0.8z^{-1}}{1 + 0.5z^{-1}} = \frac{0.8z}{(z-0.75)} + \frac{-0.8z}{(z+0.5)}$$

The `residuez` function returns the poles in column vector *p*, the residues corresponding to the poles in column vector *r*, and any improper part of the original transfer function in row vector *k*.

The coefficients are determined in the ascending order of z^{-1} . After the poles and zeros have been determined, we can plot them either with:

i) zeros and poles in **column** vectors, or

```
>> zero =[0]
zero =
0

>> pole = [0.75; -0.5]
pole =
0.7500
-0.5000
>> zplane(zero,pole)%Plot zeros and poles on the Z-plane
```

ii) numerator and denominator coefficients in **row** vectors.

This method is used to find the roots of the numerator and denominator polynomial using the built-in MATLAB function, `roots`.

```
>> num = [0 1 0];
>> den = [1 -0.25 -0.375];
>> zero = roots(num) %Find the root of the numerator
>> pole = roots(den) %Find the root of the denominator
 zero =
0

pole =
0.7500
-0.5000

>> zplane(zero,pole)%Plot zeros and poles on the Z-plane
```

Both methods will plot the same output at the *z*-plane shown in the following figure.

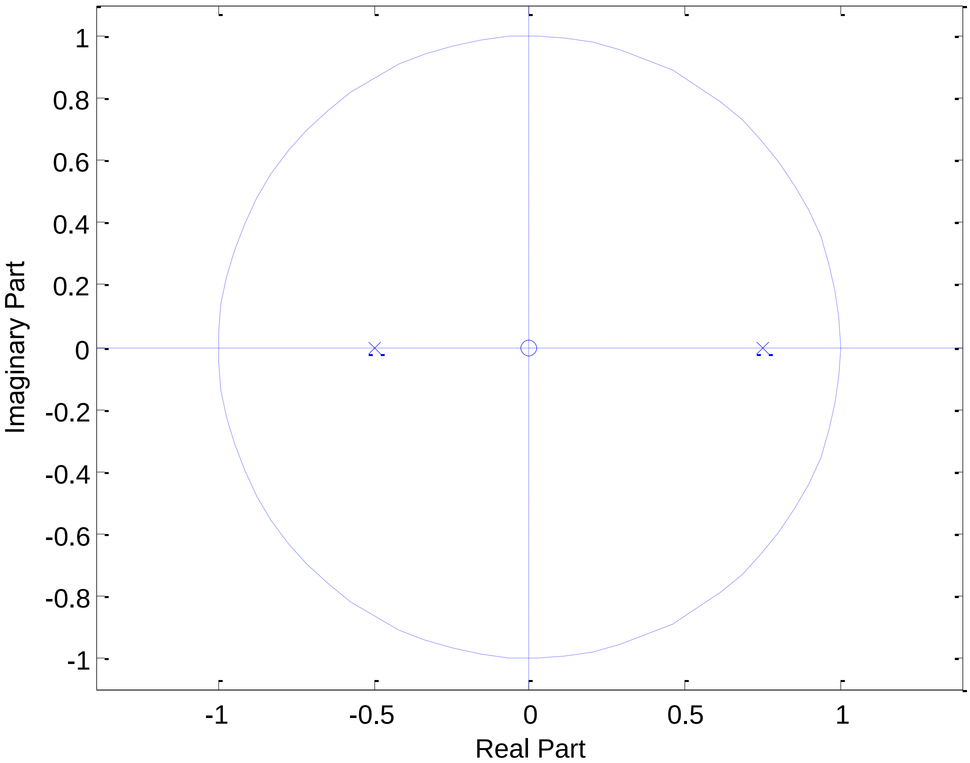


Figure 2 – Zeros and Poles on the Z-Plane

CODE EXAMPLE:

```
% zpart1.m
%% System 1
b1=[0.2 0.2 0.2 0.2 0.2]     %Coefficients (Num.) of transfer function System 1
a1=[1 0 0 0 0]               %Coefficients (Den.) of transfer function System 1
root1_num=???                %Use roots function
root1_den=???                %Use roots function
%% System 2
b2=???                       %Coefficients (Num.) of transfer function System 2
a2=???                       %Coefficients (Den.) of transfer function System 2
root2_num=???                %Use roots function
root2_den=???                %Use roots function
%% System 3
b3=???                       %Coefficients (Num.) of transfer function System 3
a3=???                       %Coefficients (Den.) of transfer function System 3
root3_num=???                %Use roots function
root3_den=???                %Use roots function
```

```
--------------------------------------------------------------------------------
% zpart2.m
%% System 1
???
%% System 2
???
%% System 3
???
%% Plotting of zeros and poles at z-plane
%System 1
subplot(1,3,1); zplane(root1, []);
title('z-plane plot H1'); axis ([-1 1 -1 1]);
%System 2
???
%System 3
???

--------------------------------------------------------------------------------
% zpart3.m
N=1024; %%
System 1
b1=[0.2 0.2 0.2 0.2 0.2]    %Coefficients (Num.) of transfer function System 1
a1=[1 0 0 0 0]              %Coefficients (Den.) of transfer function System 1
root1_num=???               %Use roots function
root1_den=???               %Use roots function
[H1, W1]= ???         %Use freqz function: [H,W] = FREQZ(B,A,N);with N=1024
%% System 2
b2=???                     %Coefficients (Num.) of transfer function System 2

a2=???                     %Coefficients (Den.) of transfer function System 2

root2_num=???

root2_den=???

[H2, W2] =???              %Use freqz function: [H,W] = FREQZ(B,A,N);with N=1024

%% System 3

b3=???                     %Coefficients (Num.) of transfer function System 3

a3=???                     %Coefficients (Den.) of transfer function System 3

root3_num=???

root3_den=???

[H3, W3] =???              %Use freqz function: [H,W] = FREQZ(B,A,N);with N=1024

%% z-plane and magnitude response plots
%System 1
plot(W1/pi, abs(H1)); % use subplot
title('Magnitude response, |H1|');  xlabel
('Normalized frequency'); ylabel('|H1|');
zplane(root1, []); % use subplot title('z-plane
plot H1');  %System 2
???
%System 3
??? --------------------------------------------------------------------------------
---

% zpart4.m clear;
close all;
N = 1024;
%Filter 1
Z1_ang = [-1 0.8 -0.8 0.6 -0.6]*pi; %Denormalized with pi
 Z1 = ??? %Use formula cose + jsine
```

```
h1 = ??? %Use poly function %To find the coefficients of the characteristic polynomial of
a matrix Z1
[H1, W1] = freqz(h1, N);
H1 = H1 / max(abs(H1)); n1
= [0:length(h1)-1];
  %Filter
2
Z2_ang = ???
Z2 = ??? h2
= ???
[H2, W2] = ???
H2 = ??? n2 = ???
%Plottings figure;
%Filter 1
subplot(2,2,1);
plot(W1/pi, abs(H1));
set(gca,'xtick',[0:0.1:1]); grid on;
title('Filter 1, normalized magnitude response, |H1|');
xlabel('Normalized frequency');  ylabel('|H1|');
subplot(2,2,3);  zplane(Z1', []);
title('z-plane plots of H1');

%Filter 2
???
```