# SP Lab-3 Report

Name: Vedant & Sarvesh
Table number: 8
Roll number: 2023112012, 2023102039

---

In this lab, we will learn about Z-Transform and study different ROC's of the system.

## 3.1 : ROC & stability from pole positions of Z-Transform

We have given $X(z) = \frac{N(z)}{D(z)}$
Also for simplicity we assume N(z) = 1.

### (i) Concept of the Stability of the System

Poles and Region of Convergence (ROC) play a very important role for identifying if the system is stable or not. If the ROC of the system includes the unit circle in Z-plane then the system is said to be stable otherwise it is not stable.

The ROC of system always exists between the two consecutive poles of the system if the poles are arranged in the ascending order.

### (ii) Why can Poles never be a part of the ROC?

Poles, by definition, are the roots of the denominators of $X(z)$. Therefore, when z approaches the poles then $X(z) \rightarrow \infty$ so the z-transform doesn't converge at poles. This means that poles can't be part of the ROC.

a. Function code for calculating the given information : -

```
function [N, ROC, ROC_zero, S] = region_of_convergence(p)

p_sorted = sort(abs(p)); %this sorts the poles
p_unique = unique(abs(p)); %this stores the unique value of poles


N = length(p_unique) + 1;   %unique ROCs
ROC = zeros(N, 2);          %ROC matrix
S = zeros(N, 1);            %stability vector


%we have three cases that ROC > max(poles) , ROC btwn two poles and ROC is < min(poles).

%1. ROC i.e |Z| > max(poles)
    ROC(1, :) = [p_sorted(end), Inf];
    if p_sorted(end) > 1
        S(1) = 1;  % Stable if all poles are inside the unit circle
    end

%2. ROC i.e 2 < |Z| < N-1: between poles
    for i = 2:N-1
        ROC(i, :) = [p_sorted(i-1), p_sorted(i)];
        if p_sorted(i-1) < 1 && p_sorted(i) > 1
            S(i) = 1;  % Stable if the unit circle is in the ROC
        end
    end

%3. ROC i.e |Z| < min(poles)
    ROC(N, :) = [0, p_sorted(1)];
    if p_sorted(1) < 1
        S(N) = 1;  % Stable if all poles are outside the unit circle
    end

% To check if 0 is part of any ROC
ROC_zero = any(ROC(:, 1) == 0);

end
```

The main script :-

```
p = -1;
[N,ROC,ROC_zero,S] = region_of_convergence(p);
```

b. For each p we calculate the given :-

1. P = -1 :-

| Name ▲ | Value |
|---|---|
| N | 2 |
| p | -1 |
| ROC | [1,Inf;0,1] |
| ROC_zero | 1 |
| S | [0;0] |

2. P = 0 :-

| Name | Value |
|---|---|
| N | 2 |
| p | 0 |
| ROC | [0,Inf;0,0] |
| ROC_zero | 1 |
| S | [0;1] |

3. P = 0.3 :-

| Name | Value |
|---|---|
| N | 2 |
| p | 0.3000 |
| ROC | [0.3000,Inf;0,0.3000] |
| ROC_zero | 1 |
| S | [0;1] |

4. P = [0,0.65] :-

| Name | Value |
|---|---|
| N | 3 |
| p | [0,0.6500] |
| ROC | [0.6500,Inf;0,0.6500;0,0] |
| ROC_zero | 1 |
| S | [0;0;1] |

5. P = [1,-0.85] :-

| Name | Value |
|---|---|
| N | 3 |
| p | [1,-0.8500] |
| ROC | [1,Inf;0.8500,1;0,0.8500] |
| ROC_zero | 1 |
| S | [0;0;1] |

6. P = [0.45,-0.45] :-

| Name | Value |
|---|---|
| N | 2 |
| p | [0.4500,-0.4500] |
| ROC | [0.4500,Inf;0,0.4500] |
| ROC_zero | 1 |
| S | [0;1] |

7. P = [2,2,2] :-

| Name | Value |
|---|---|
| N | 2 |
| p | [2,2,2] |
| ROC | [2,Inf;0,2] |
| ROC_zero | 1 |
| S | [1;0] |

8. P = [0,1,2] :-

| Name | Value |
|---|---|
| N | 4 |
| p | [0,1,2] |
| ROC | [2,Inf;0,1;1,2;0,0] |
| ROC_zero | 1 |
| S | [1;0;0;1] |

9. P =[ -0.5,1i] :-

| Name | Value |
|---|---|
| N | 3 |
| p | [-0.5000 + 0.0000i,0.0... |
| ROC | [1,Inf;0.5000,1;0,0.5000] |
| ROC_zero | 1 |
| S | [0;0;1] |

10. P = [0,1i,-1i] :-

| | |
|---|---|
| N | 3 |
| p | [0.0000 + 0.0000i,0.00... |
| ROC | [1,Inf;0,1;0,0] |
| ROC_zero | 1 |
| S | [0;0;1] |

11. P = [1,-1,2+1i,2-1i] :-

| Name | Value |
|---|---|
| N | 3 |
| p | [1.0000 + 0.0000i,-1.0... |
| ROC | [2.2361,Inf;1,1;0,1] |
| ROC_zero | 1 |
| S | [1;0;0] |

12. P = [1+i, 1+2i, 1+3i, 2-i] :-

| Name | Value |
|---|---|
| N | 4 |
| p | [1.0000 + 1.0000i,1.00... |
| ROC | [3.1623,Inf;1.4142,2.2... |
| ROC_zero | 1 |
| S | [1;0;0;0] |

ROC

4x2 double

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 3.1623 | Inf | |
| 2 | 1.4142 | 2.2361 | |
| 3 | 2.2361 | 2.2361 | |
| 4 | 0 | 1.4142 | |
| 5 | | | |

13. P = [1,-1,1i,-1i] :-

| Name | Value |
|---|---|
| N | 2 |
| p | [1.0000 + 0.0000i,-1.0... |
| ROC | [1,Inf;0,1] |
| ROC_zero | 1 |
| S | [0;0] |

Explanation :-

### a. For p = [0,1,2] :-

The given poles are on the real axis of the z-plane.
The following are the possible ROC's for the given poles :-

- 0<|z|<1
- 1<|z|<2
- |z|>2

**ROC and Stability:**

- |z|<1:

This ROC results in a causal and unstable system.

- 1<|z|<2:

This ROC results in a causal and unstable system.

- $|z|>2$:

 This ROC corresponds to a stable system because it includes the unit circle $|z|=1$, which ensures that all poles are inside the unit circle.

**b. For p = [1, -1, j, -j] :-**

These poles are symmetric in the z-plane and lie on the unit circle.

 The following are the possible ROC's for the given poles :-

- $|z|<1$
- $|z|>1$

**ROC and Stability:**

- $|z|<1$:

 This ROC would include the area inside the unit circle, leading to an unstable system because the poles lie on the unit circle.

- $|z|>1$ :

This ROC would exclude the unit circle and the poles, leading to an unstable system.

# 3.2 : Visualise Z-Transform using 3D Plot

While plotting the Z-Transform in the matlab, we create plot of the log-magnitude of Z-transform i.e. $log\ |X(z)|$ using the function meshgrid() and plot it using mesh().

## (i) Why take log-magnitude instead of directly ploting?

- Taking a log helps in handling a wide range of values so that we can visualise a large set of data at once.
- Many signals and systems have responses that decay exponentially. The logarithm of an exponential decay/rise turns it into a linear decay/rise, which is easier to analyse.
- It helps in distinguishing Poles and Zeros in the 3D plot. Because log takes negative values from 0 to 1 and positive values for greater than 1, therefore Zero shows the dip while Poles show the rise.

## (ii) About meshgrid() and mesh() function

meshgrid(x,y) returns 2-D grid coordinates based on the coordinates contained in vectors x and y. X is a matrix where each row is a copy of x, and Y is a matrix where each column is a copy of y.

mesh(X,Y,Z) creates a mesh plot, which is a three-dimensional surface that has solid edge colours and no face colours. The function plots the values in matrix Z as heights above a grid in the x-y plane defined by X and Y. The edge colours vary according to the heights specified by Z.

## (iii) zplane() function

zplane(b,a) where b and a are row vectors, first uses roots to find the zeros and poles of the transfer function represented by the numerator coefficients b and the denominator coefficients a.

'b' and 'a' are coefficients array of the polynomial in $z^{-1}$ with increasing power.

## (iv) impz() function

impz(b,a,n) returns the impulse response of the digital filter with numerator coefficients b and denominator coefficients a with ‘n’ number of samples and returns the response coefficients in h and the sample times in t.

NOTE: By default, impz function plots the impulse response for which the system is causal.

## (v) Matlab Script

## For (a)

```
% From the documentation:
% meshgrid(x,y) returns 2-D grid coordinates based on the coordinates contained in vectors x and y. X is a matrix where each row is a copy of x, and Y is a matrix where each column is a copy of y.
% Defining the axis x and y
x = -1.5:0.01:1.5;
y = -1.5:0.01:1.5;
% 3*3 plot with center at origin can be defined using the above axis

% Generate the grid using meshgrid
[X, Y] = meshgrid(x, y);

% Defining the complex z variable
z = X + 1i * Y;

p= 0.8;
% change the value of m from here

X_z= z ./ (z - p);
log_X_z = log(abs(X_z));

% Display the grid
figure;
mesh(X, Y, log_X_z);
xlabel('X');
ylabel('Y');
zlabel('Z');
title('2-D Grid on the Z-Plane');
```

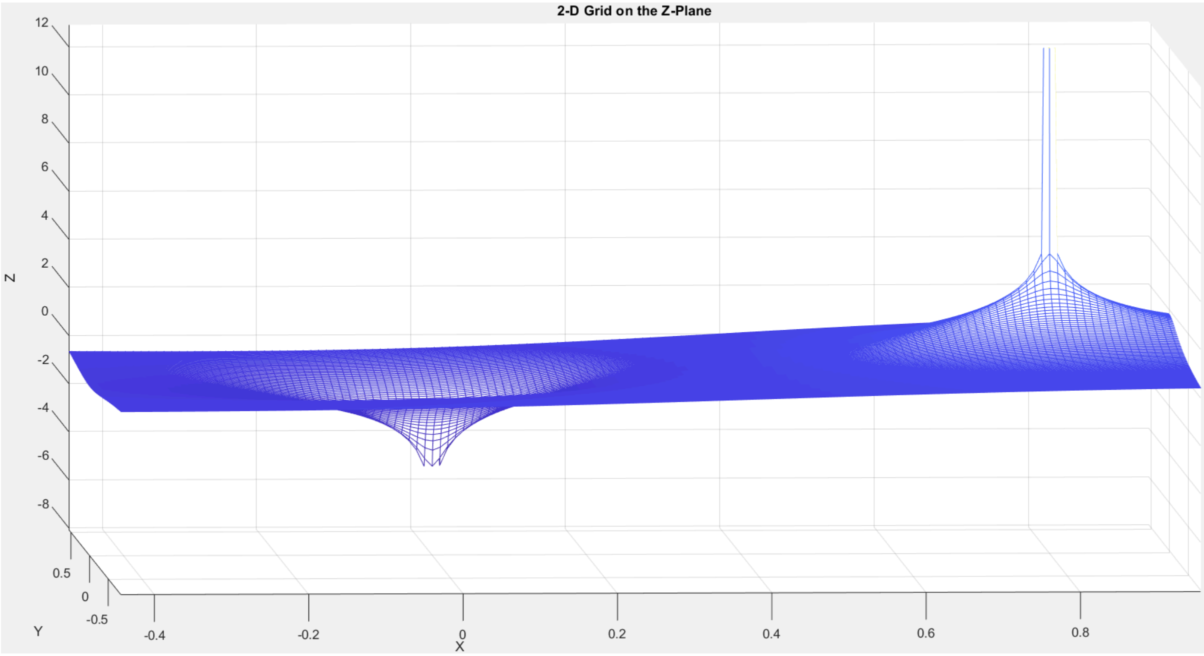
## For (c)

```
p = 0.8;
b = [1];
a = [1,-p];
n = 51;
title('Pole-Zero Plot');
impz(b,a,n)
% The pole-zero plot is displayed using the impz function. The function takes the numerator and denominator coefficients of the transfer function as input arguments. The number of points to be plotted is also specified. The plot shows the poles and zeros of the transfer function in the complex plane.
```

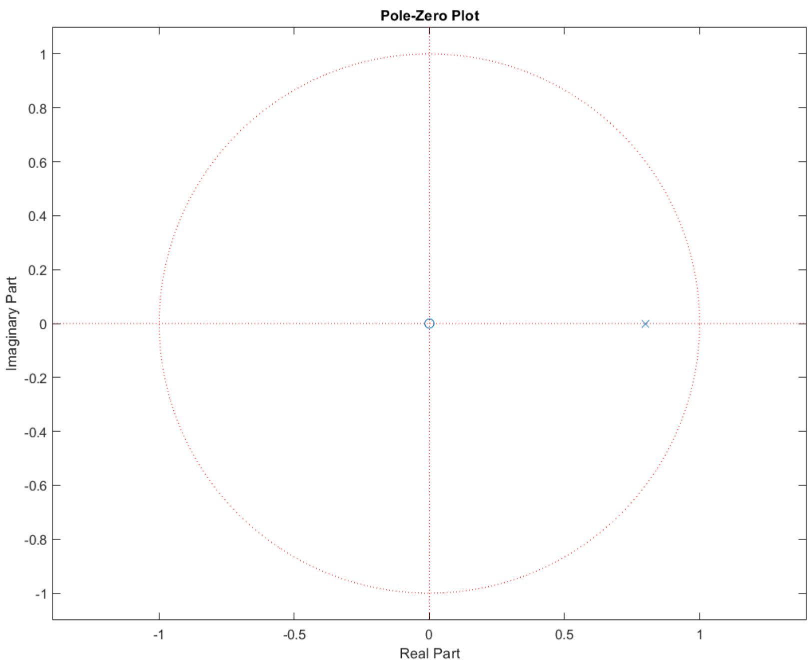
Explanation of the code is given with the code using comments.

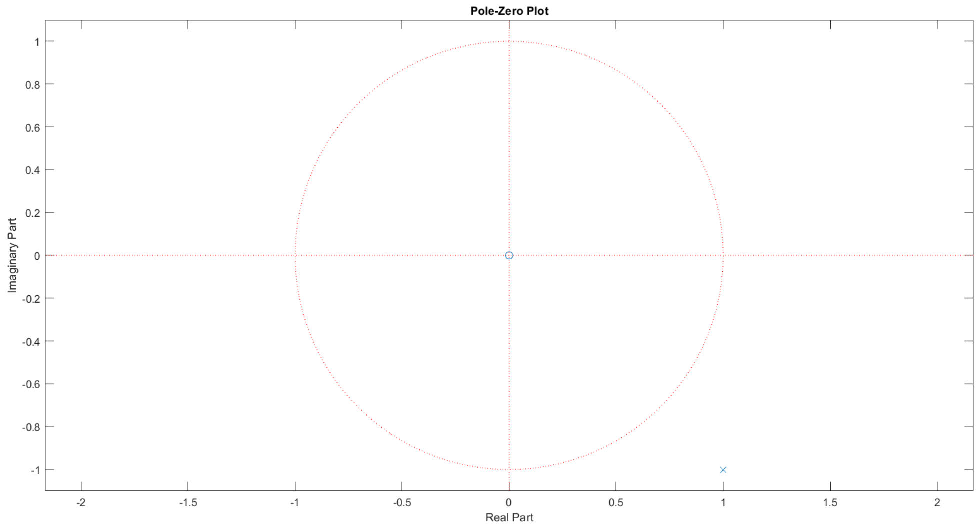
## (vi) Plots

### For (a):

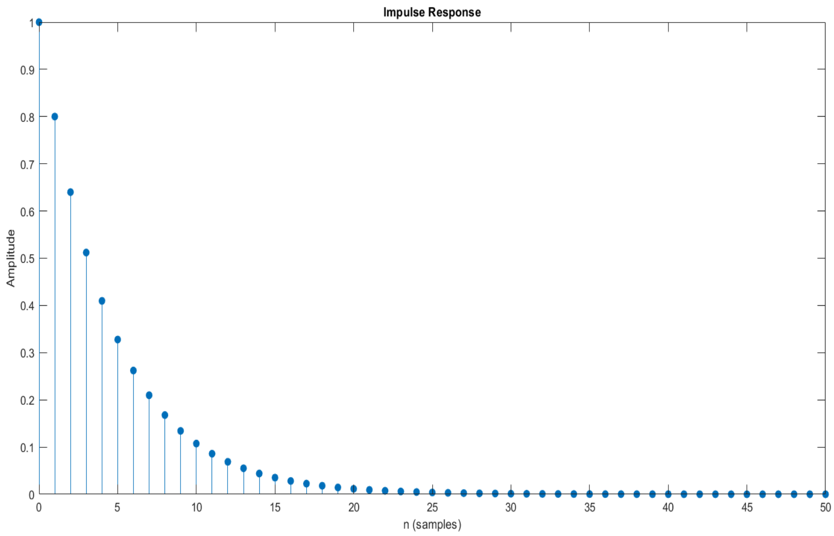


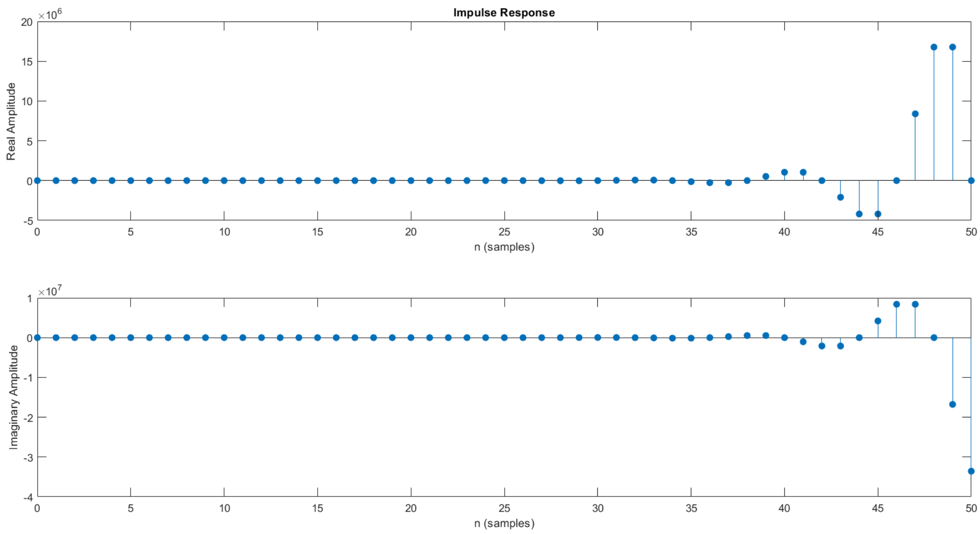

### For (b):




Pole-Zero Plot
Imaginary Part
Real Part

For (c):


Impulse Response
Amplitude
n (samples)


Impulse Response
Real Amplitude
n (samples)
Imaginary Amplitude
n (samples)

### (iv) Observation from the Plots

From the pole-zero plot, we can see that if the pole is inside the unit circle therefore the signal is stable and hence the impulse response is decaying.
If the pole is outside the unit circle then the signal is not stable, hence impulse response is also not converging.

From the 3D plot, zero can be spotted on the bulged down portion and pole is present in the bulged up region.

## 3.3 : System with multiple poles

We have given Transfer function of the system in Z-domain as following:

$$X(z) = \frac{z^2 - (2\cos\theta)\, z + 1}{z^2 - (2r\cos\theta)z + r^2}, \qquad r \in (0,1), \theta \in [0, \pi]$$

### (i) Finding the Poles and zeros

By definition,
Poles are the roots of the denominators and zeros are the roots of the numerators.
By factoring numerator and denominator, we get the following expression :-

$$X(z) = \frac{(z - e^{j\theta})(z - e^{-j\theta})}{(z - re^{j\theta})(z - re^{-j\theta})}$$

So we get :-

Zeros = $e^{j\theta}, e^{-j\theta}$

Poles = $re^{j\theta}, re^{-j\theta}$

As $\theta$ changes, the angles of the poles and zeros with respect to the positive real axis changes. For $\theta = 0$ both poles and zeros lie on the real axis.

As $\theta$ increases from 0 to π , the poles and zeros rotate counterclockwise around the origin, sweeping from the positive real axis to the negative real axis.

r controls the distance of the poles from the origin. The zeros remain on the unit circle since their magnitude is always 1.

As r decreases from 1 to 0, the poles move closer to the origin. When r = 1, the poles are on the unit circle at the same locations as the zeros. As r approaches 0, the poles collapse toward the origin.

As r < 1 and if we have ROC |z| > r, then we can achieve a stable system !

### (ii) Intuition for the Impulse Response

Because it is given that r<1 therefore the magnitude of poles are always less than 1. Also, the impz() function plots the causal system, that is it takes the ROC outside the outermost pole. Therefore, it means that the system will be always stable whatever be the value of r or $\theta$. Hence, all the impulse responses will converge and die out to zero.

### (iii) Pole-Zero Plot for $r = 0.95$ and $\theta = 60^o$

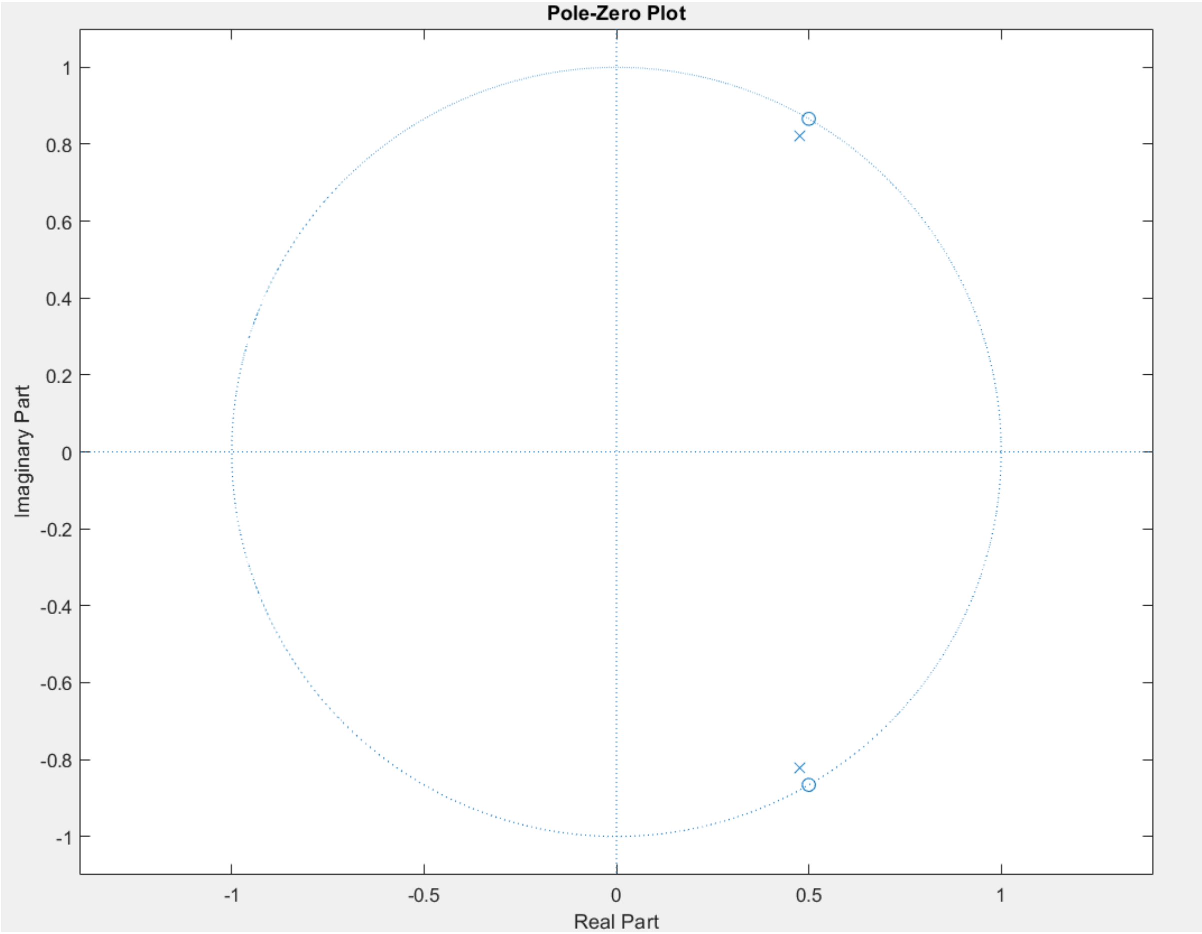
For plotting in matlab, We have the following code :-

```
n = pi/3;
r = 0.95;

b = [1,-2*cos(n),1];
a = [1,-2*r*cos(n),r^2];

figure;
zplane(b,a);
```

We see that we have conjugate poles and zeros.  Following is the plot for it :-




We can observe that we get conjugate poles and zeros .

## (iii) Relation with Notch Filter

Reference:- Book PM 339
Notch Filter is a filter that allows all the frequency to pass except one frequency which is equal to $\theta$ in the equation.
Generally, we keep the value of ‘r’ such that the poles are very near to the zeros to reduce the bandwidth of the filter for only attenuating a particular frequency.

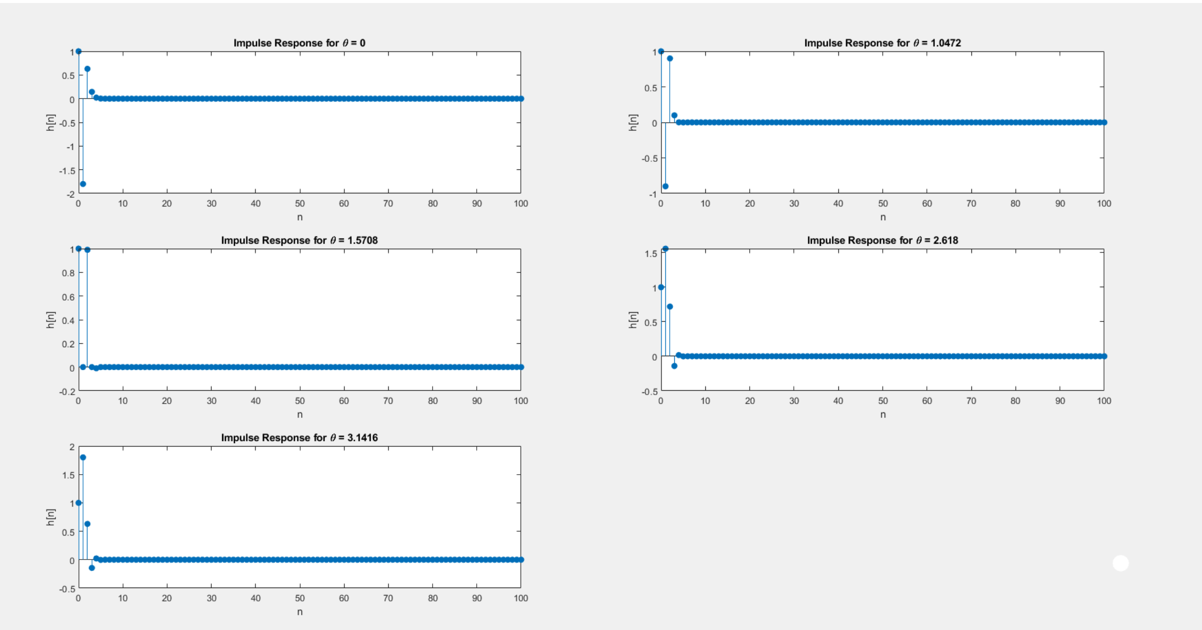
We know that r controls the decay or growth of the system's impulse response . When r is near 1 there is a slower decay and as soon as r is less than 1 , there is a faster decay which means that the system is stable as it has its ROC inside the unit circle plane so it will decay !

θ controls the frequency of the oscillations. As θ increases the system becomes more oscillatory .

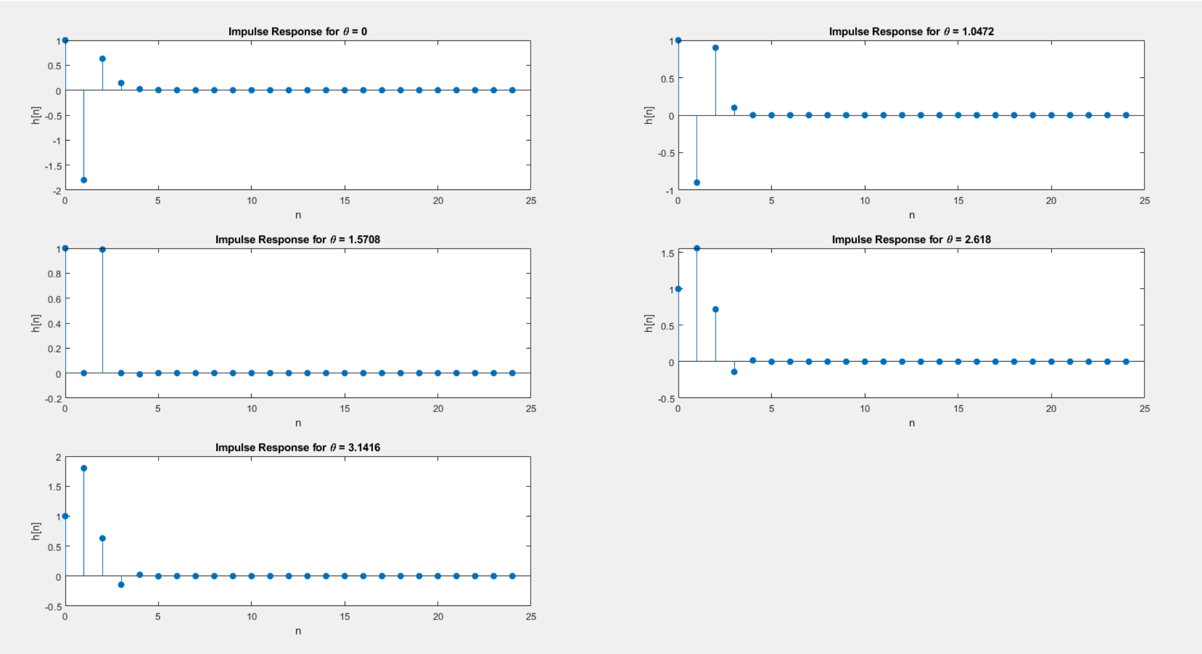
For each θ and r plots and there analysis :-

Close up view for n = 25

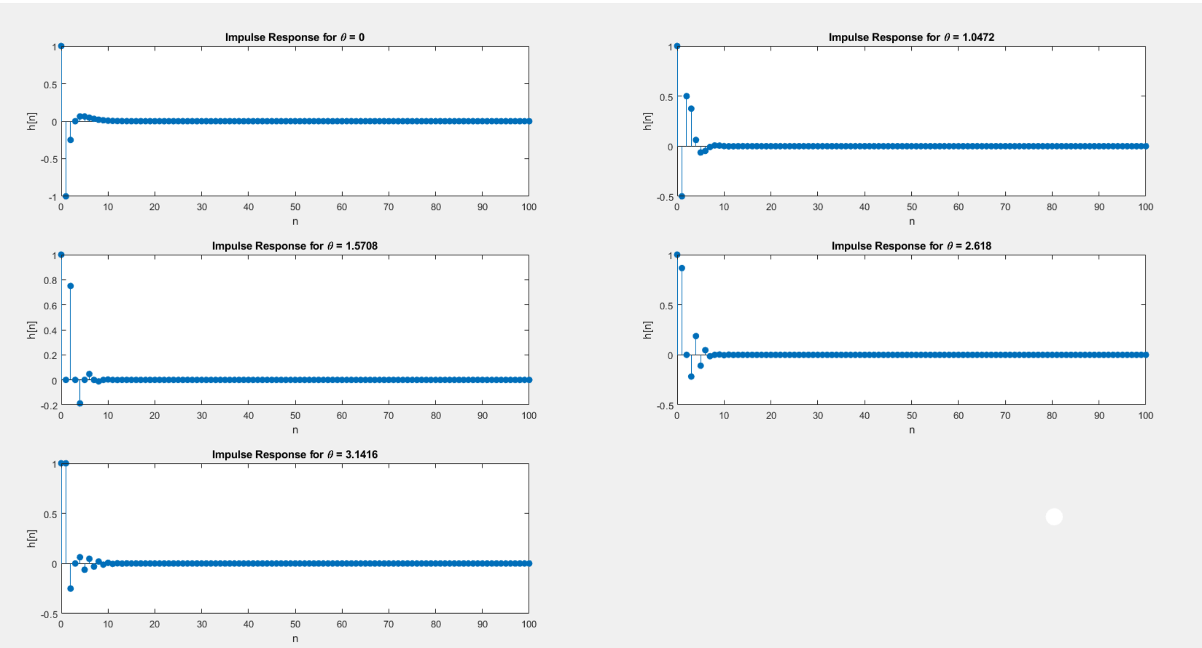
I. r = 0.1 :-



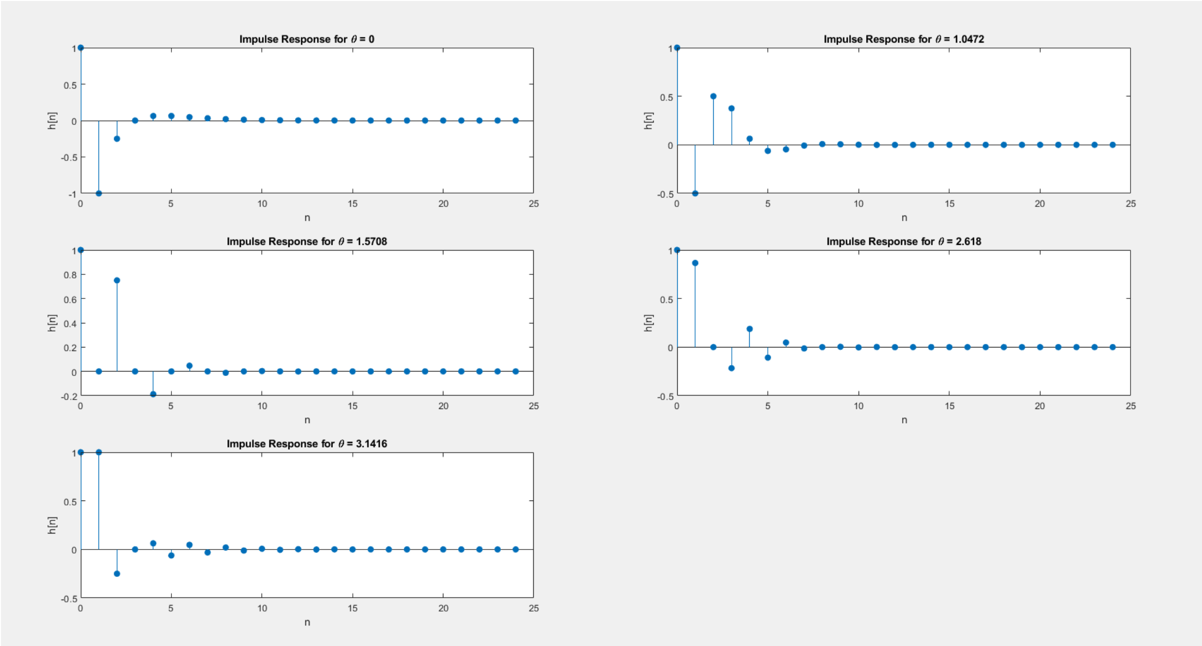

Close up view :-



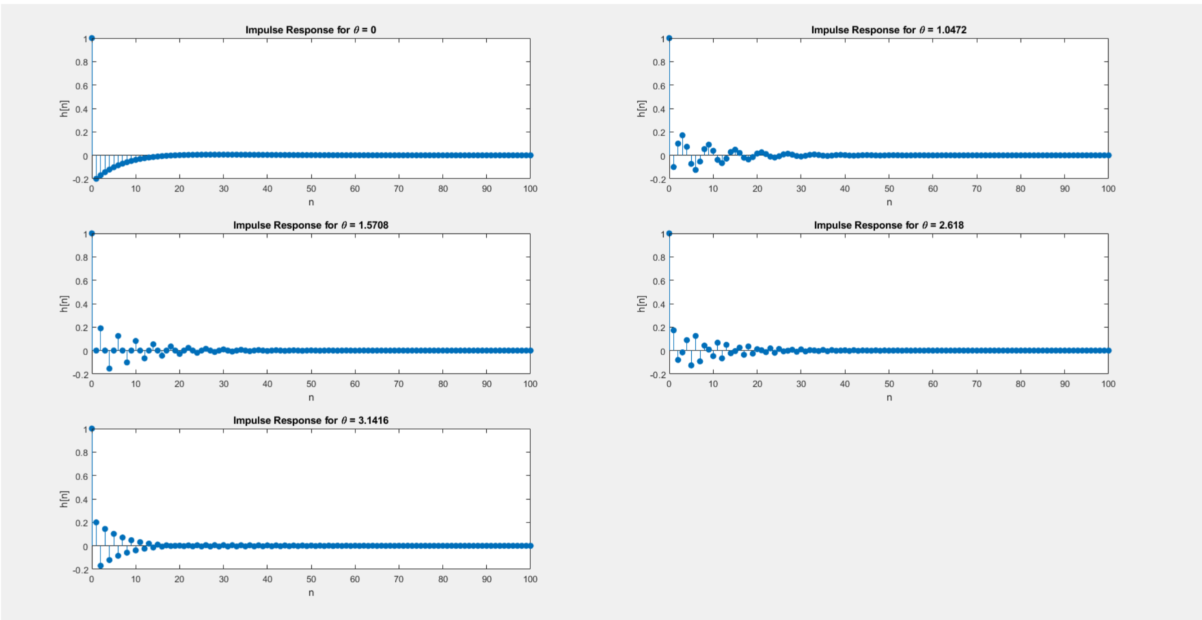

## II. r = 0.5 :-



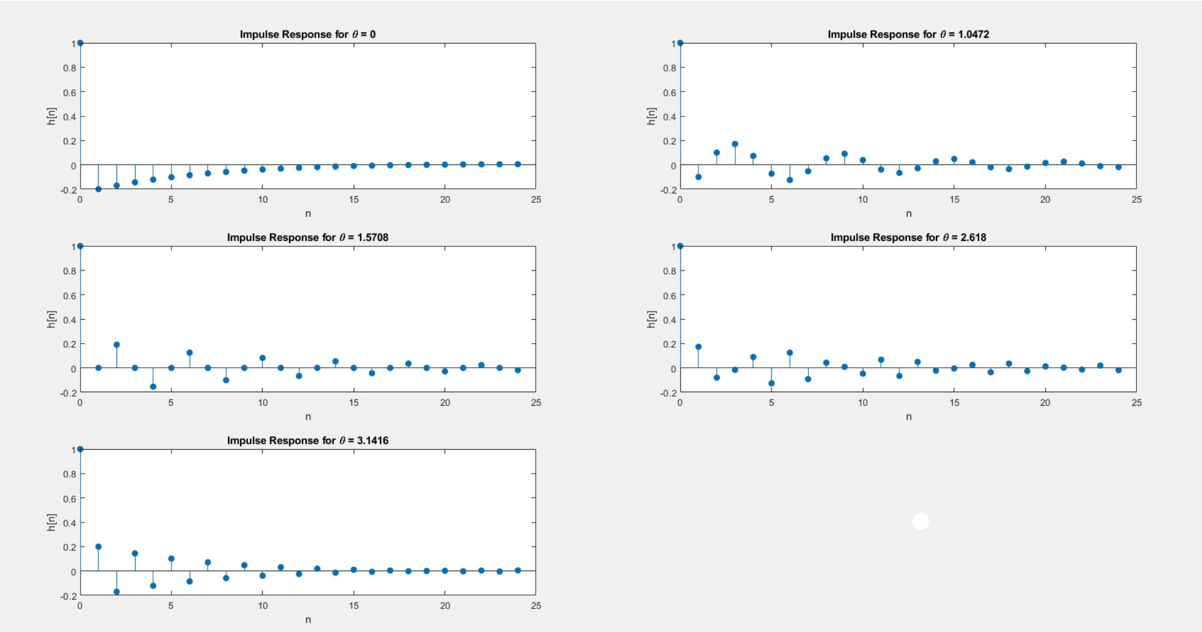

## Close up view :-



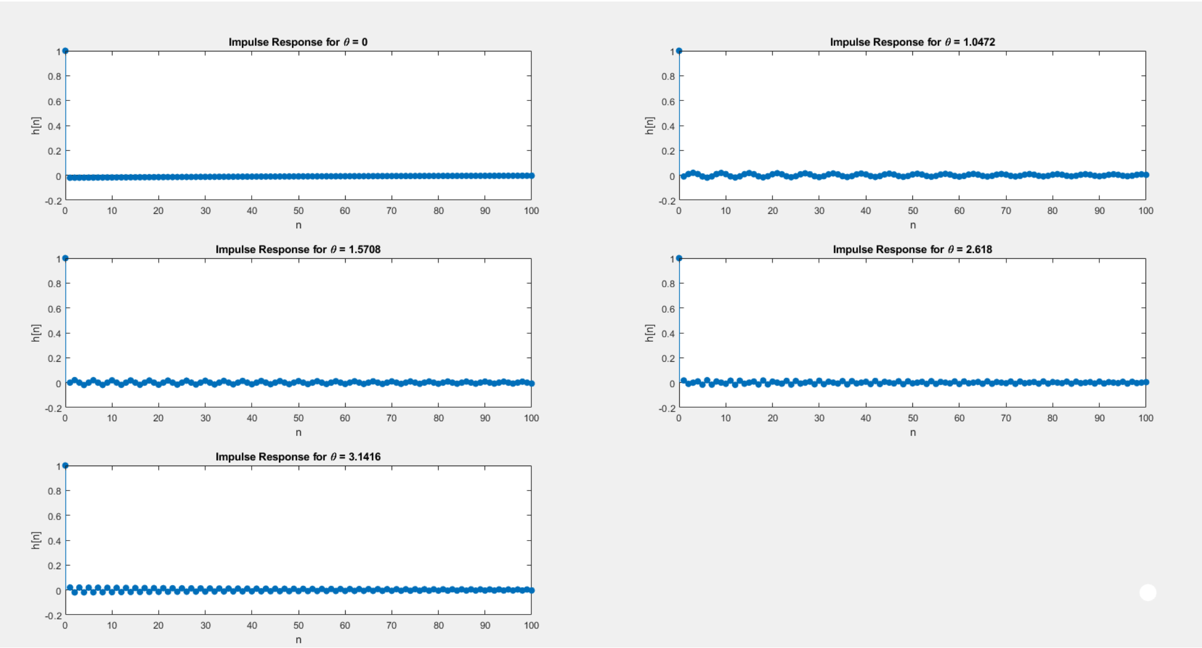

## III. r = 0.9 :-



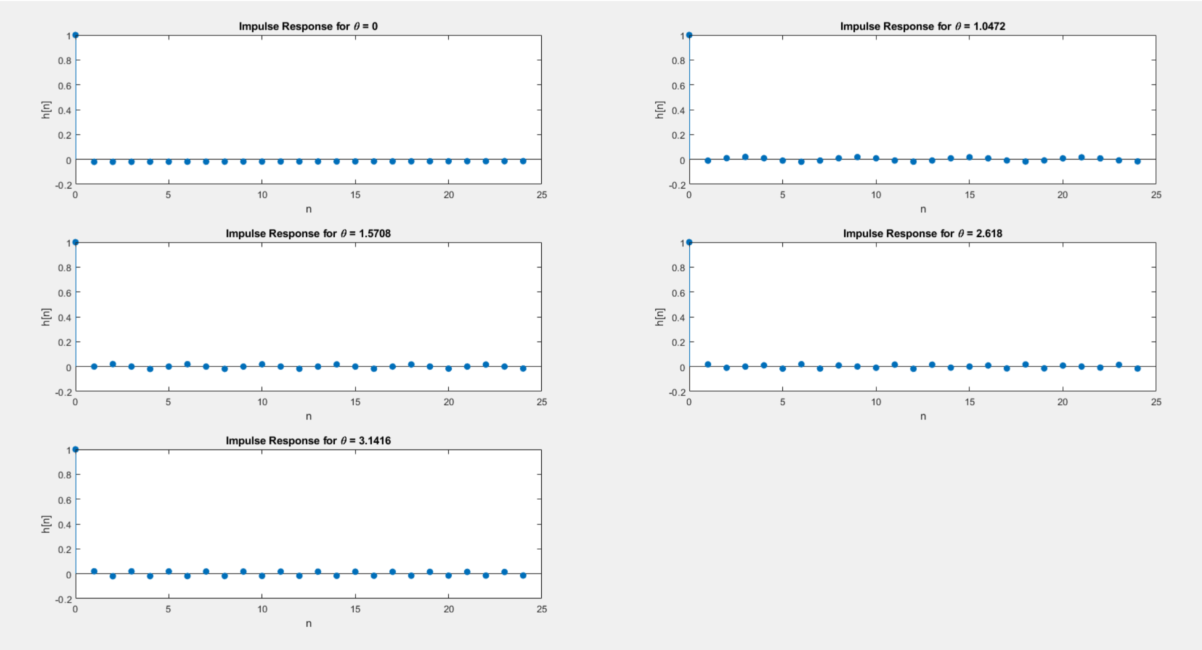

## Close up view :-




# IV. r = 0.99 :-




# Close up view:-


Impulse Response for θ = 0
Impulse Response for θ = 1.0472
Impulse Response for θ = 1.5708
Impulse Response for θ = 2.618
Impulse Response for θ = 3.1416
h[n]
n