| Course Name: | Digital Signal Processing Laboratory | Semester: | VI |
|---|---|---|---|
| Date of Performance: | 13 / 03 / 2025 | Batch No.: | B - 2 |
| Faculty Name: | Dr. Om Goswami | Roll No.: | 16014022050 |
| Faculty Sign & Date: | | Grade / Marks: | |

# Experiment No: 4
# Title: Z Transform

**Aim and Objective of the Experiment:**

To find the Z transform, plot its pole-zero plot and ROC using MATLAB:

a) Left-sided exponential sequence
b) Right-sided exponential sequence
c) Two-sided exponential sequence

**COs to be achieved:**

**CO3:** Design digital FIR filters

**Theory:**

The Z-transform is a mathematical tool used in Digital Signal Processing (DSP) for analyzing discrete-time signals and systems. It is particularly useful in the analysis and design of digital filters and the study of system stability. The Z-transform converts a discrete-time signal from the time domain into the complex frequency domain, allowing easier manipulation and analysis.

The Z-transform of a discrete-time signal is defined as:

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n}$$

Where z is a complex variable given by $z = Re^{j\omega}$, with R being the magnitude and the $\omega$ phase.

**Region of Convergence (ROC)**

The Region of Convergence (ROC) of the Z-transform is the set of values of for which the Z-transform series converges. The ROC is crucial in determining the system's stability and causality.

***Properties of ROC:***

- The ROC is always a circular region in the complex plane (either inside, outside, or between circles).

- The ROC cannot contain any poles.
- If *x[n]* is a right-sided sequence, the ROC extends outward from the outermost pole.
- If *x[n]* is a left-sided sequence, the ROC extends inward up to the innermost pole.
- If *x[n]* is a two-sided sequence, the ROC lies between poles.
- The ROC must be included for stability.

**Pole-Zero Plot**

A **Pole-Zero Plot** is a graphical representation of the poles and zeros of a system in the Z-plane. It provides insight into system behavior and stability.

- **Poles:** Values of *z* where *x(z)*becomes infinite.
- **Zeros:** Values *z* of where *x(z)* becomes zero.

The poles and zeros are plotted in the complex plane, where the x-axis represents the real part and the y-axis represents the imaginary part.

**Stability of a System**

A discrete-time system is **stable** if the ROC includes the unit circle. This ensures that the system's impulse response is absolutely summable:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

***Stability Conditions:***

- For **causal systems**, all poles must be inside the unit circle $|z|<1$.
- For **anti-causal systems**, all poles must be outside the unit circle $|z|>1$ .
- For **two-sided systems**, the ROC must include $|z|=1$for stability.

Stepwise-Procedure / Algorithm:
1. Click on the MATLAB Icon on the desktop.
2. MATLAB window open.
3. Click on the ‘FILE’ Menu on the menu bar.
4. Click on NEW M-File from the file Menu.
5. An editor window opens, start typing commands for the given aim.
6. Now SAVE the file in the directory.
7. Then Click on DEBUG from Menu bar and Click Run

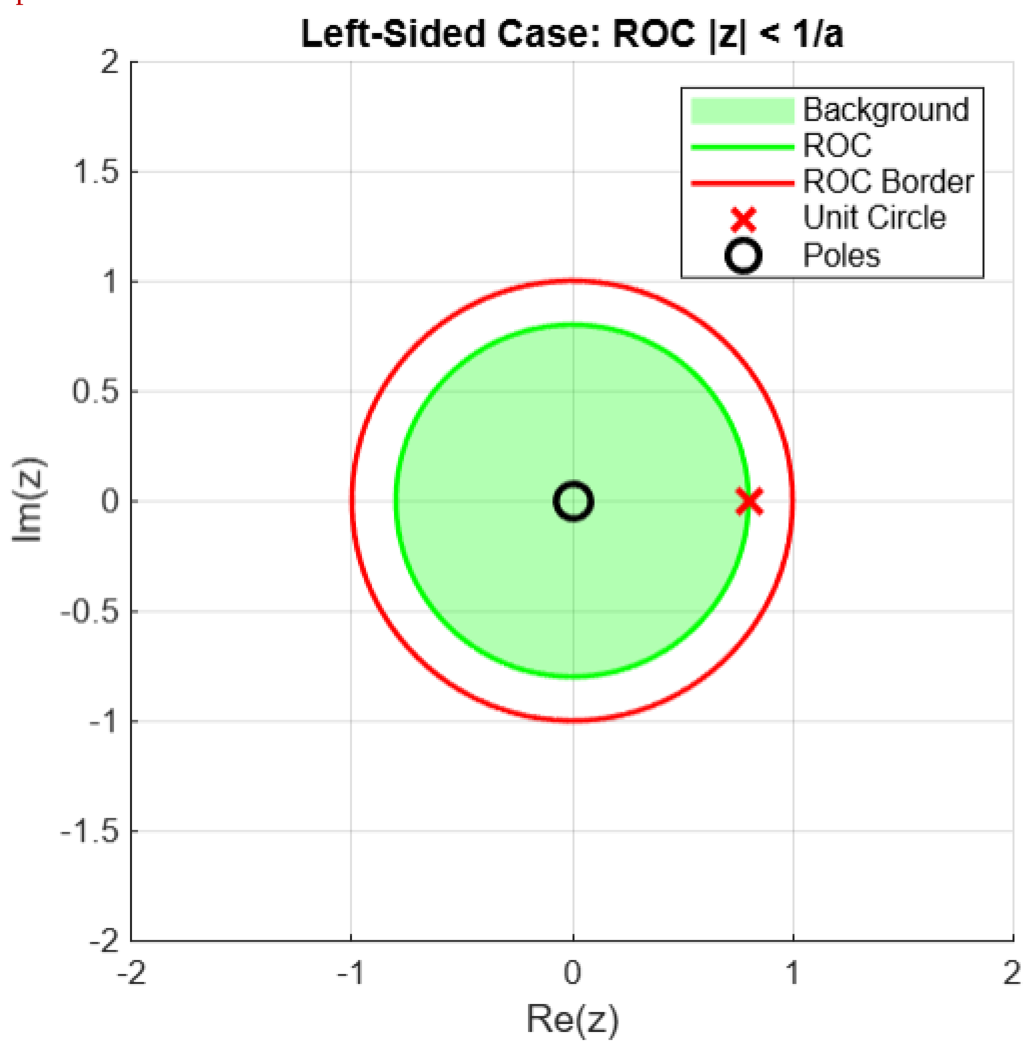
Code for Left-sided exponential sequence.

a) Code for Right-sided exponential sequence

```
clc; clear; close all;
syms n z;
a_val = 0.8;

x = (a_val^n) * heaviside(-n-1);

X = z / (z - a_val);

[num, den] = numden(X);

% Solve for poles and zeros
poles = double(solve(den, z));  % Solving the denominator for poles
zeros = double(solve(num, z));  % Solving the numerator for zeros

% Define the unit circle and the ROC circle
theta = linspace(0, 2*pi, 100);
unit_circle_x = cos(theta);
unit_circle_y = sin(theta);
roc_circle_x = (a_val) * cos(theta);
roc_circle_y = (a_val) * sin(theta);

large_circle_x = 5*cos(theta);
large_circle_y = 5*sin(theta);

% Plot
figure; hold on;

% If |z| < 1/a_val (Inside ROC circle)
fill(roc_circle_x, roc_circle_y, 'g', 'FaceAlpha', 0.3, 'EdgeColor', 'none'); % ROC
region (inside the circle with radius 1/a_val)
plot(roc_circle_x, roc_circle_y, 'g', 'LineWidth', 1.5); % ROC circle boundary
(radius 1/a_val)
% If |z| >= 1/a_val (Outside ROC circle)
% fill(large_circle_x, large_circle_y, 'g', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
% Fill large circle
% fill(roc_circle_x, roc_circle_y, 'w', 'FaceAlpha', 1, 'EdgeColor', 'none'); %
Remove ROC circle fill by overlaying white
% plot(roc_circle_x, roc_circle_y, 'g', 'LineWidth', 1.5); % ROC circle boundary
(radius 1/a_val)

% Plot the unit circle (radius = 1)
plot(unit_circle_x, unit_circle_y, 'r', 'LineWidth', 1.5); % Unit circle boundary

% Plot poles and zeros
plot(real(poles), imag(poles), 'rx', 'MarkerSize', 10, 'LineWidth', 2); % Plot poles
plot(real(zeros), imag(zeros), 'ok', 'MarkerSize', 10, 'LineWidth', 2); % Plot zeros

title('Left-Sided Case: ROC |z| < 1/a');
xlabel('Re(z)'); ylabel('Im(z)');
```

```
grid on; axis equal;
legend('Background', 'ROC', 'ROC Border', 'Unit Circle', 'Poles', 'Zeroes');
xlim([-2 2]);
ylim([-2 2]);
```

Output

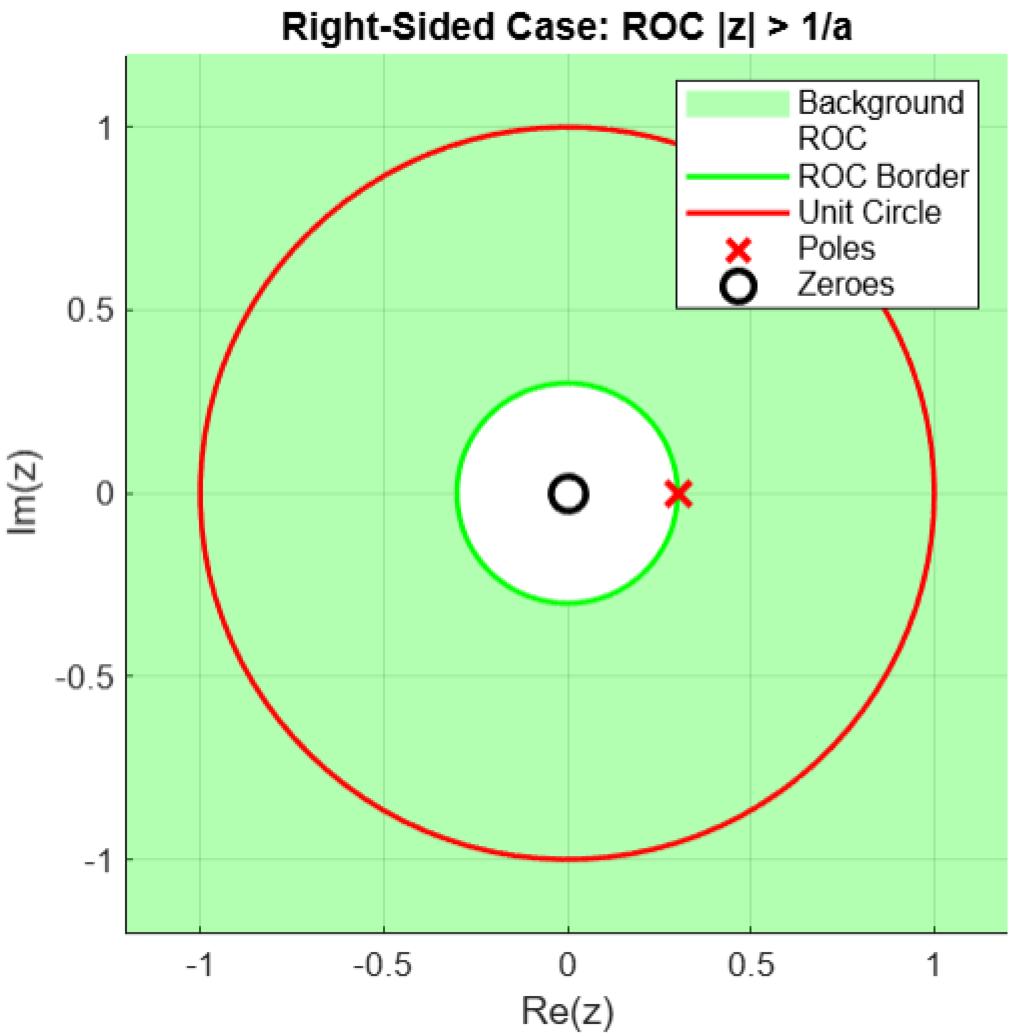


b) Code for Left-sided exponential sequence

```
clc; clear; close all;
syms n z;
a_val = 0.3;

x = (a_val^n) * heaviside(n);

X = z / (z - a_val);

[num, den] = numden(X);

% Solve for poles and zeros
poles = double(solve(den, z));  % Solving the denominator for poles
zeros = double(solve(num, z));  % Solving the numerator for zeros

% Define the unit circle and the ROC circle
theta = linspace(0, 2*pi, 100);
unit_circle_x = cos(theta);
unit_circle_y = sin(theta);
roc_circle_x = (a_val) * cos(theta);
roc_circle_y = (a_val) * sin(theta);

large_circle_x = 5*cos(theta);
large_circle_y = 5*sin(theta);

% Plot
figure; hold on;

% If |z| < 1/a_val (Inside ROC circle)
% fill(roc_circle_x, roc_circle_y, 'g', 'FaceAlpha', 0.3, 'EdgeColor', 'none'); %
ROC region (inside the circle with radius 1/a_val)
% plot(roc_circle_x, roc_circle_y, 'g', 'LineWidth', 1.5); % ROC circle boundary
(radius 1/a_val)
% If |z| >= 1/a_val (Outside ROC circle)
fill(large_circle_x, large_circle_y, 'g', 'FaceAlpha', 0.3, 'EdgeColor', 'none'); %
Fill large circle
fill(roc_circle_x, roc_circle_y, 'w', 'FaceAlpha', 1, 'EdgeColor', 'none'); % Remove
ROC circle fill by overlaying white
plot(roc_circle_x, roc_circle_y, 'g', 'LineWidth', 1.5); % ROC circle boundary
(radius 1/a_val)

% Plot the unit circle (radius = 1)
plot(unit_circle_x, unit_circle_y, 'r', 'LineWidth', 1.5); % Unit circle boundary

% Plot poles and zeros
plot(real(poles), imag(poles), 'rx', 'MarkerSize', 10, 'LineWidth', 2); % Plot poles
plot(real(zeros), imag(zeros), 'ok', 'MarkerSize', 10, 'LineWidth', 2); % Plot zeros

title('Right-Sided Case: ROC |z| > 1/a');
xlabel('Re(z)'); ylabel('Im(z)');
grid on; axis equal;
```

```
legend('Background', 'ROC', 'ROC Border', 'Unit Circle', 'Poles', 'Zeroes');
xlim([-1.2 1.2]);
ylim([-1.2 1.2]);
```

Output

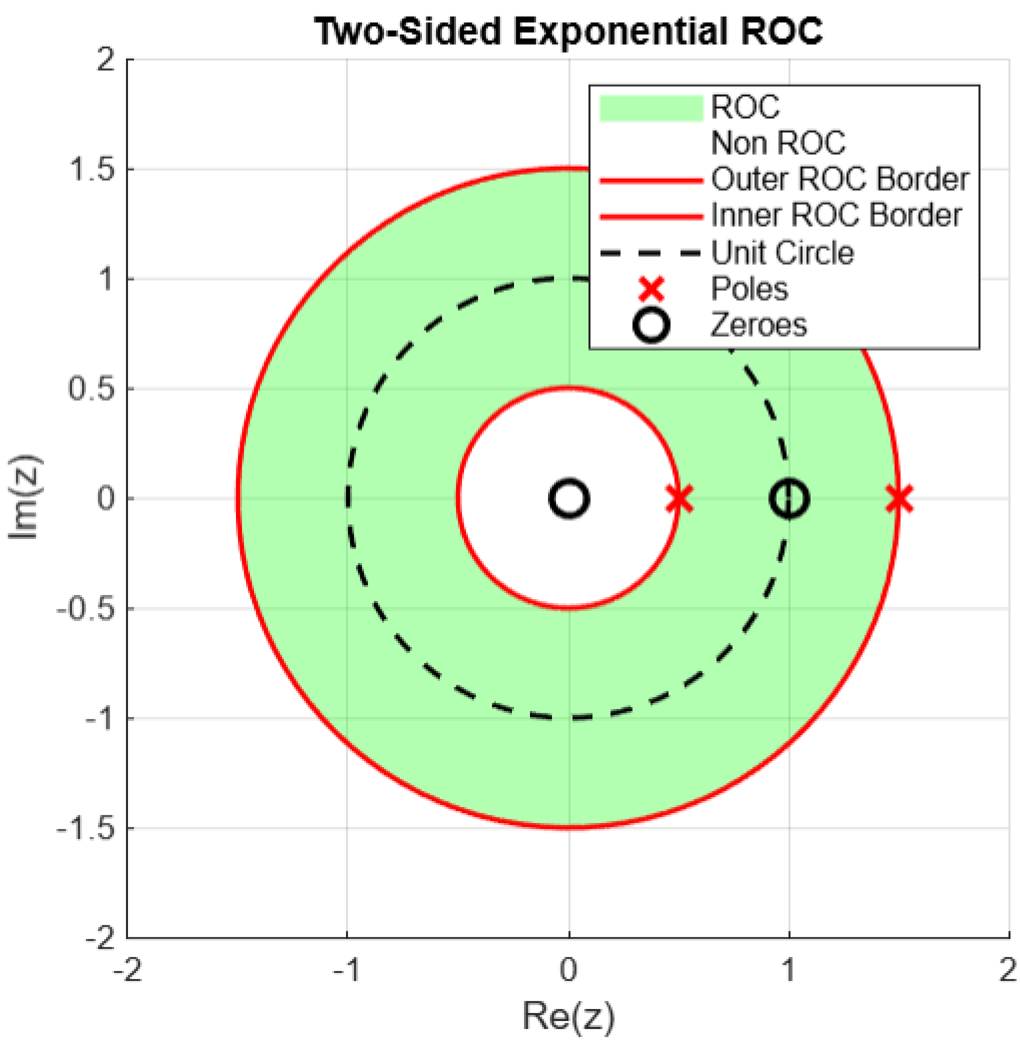



c) Code for Two-sided exponential sequence

```
syms n z;

% Parameters for the two-sided exponential ROC
a_val = 0.5;  % Pole a (inside circle, |z| > |a_val|)
b_val = 1.5;  % Pole b (outside circle, |z| < |b_val|)

X = (z / (z - a_val)) + (z / (z - b_val));

[num, den] = numden(X);
```

```
poles = double(solve(den, z));  % Solving the denominator for poles
zeros = double(solve(num, z));  % Solving the numerator for zeros

% Define the unit circle and the ROC circles
theta = linspace(0, 2*pi, 100);
unit_circle_x = cos(theta);
unit_circle_y = sin(theta);

% Define circles for the poles' regions
roc_circle_in_x = a_val * cos(theta);  % ROC for |z| > |a_val|
roc_circle_in_y = a_val * sin(theta);

roc_circle_out_x = b_val * cos(theta);  % ROC for |z| < |b_val|
roc_circle_out_y = b_val * sin(theta);

% Plot
figure; hold on;

% Fill the region between the two circles (ROC region)
fill(roc_circle_out_x, roc_circle_out_y, 'g', 'FaceAlpha', 0.3, 'EdgeColor',
'none'); % Fill outer region
fill(roc_circle_in_x, roc_circle_in_y, 'w', 'FaceAlpha', 1, 'EdgeColor', 'none'); %
Remove inner circle fill by overlaying white

% Plot the ROC circle boundaries
plot(roc_circle_in_x, roc_circle_in_y, 'r', 'LineWidth', 1.5);  % Inner ROC
boundary
plot(roc_circle_out_x, roc_circle_out_y, 'r', 'LineWidth', 1.5); % Outer ROC
boundary

% Add the unit circle for reference
plot(unit_circle_x, unit_circle_y, 'k--', 'LineWidth', 1.5);  % Unit circle

plot(real(poles), imag(poles), 'rx', 'MarkerSize', 10, 'LineWidth', 2); % Plot
poles
plot(real(zeros), imag(zeros), 'ok', 'MarkerSize', 10, 'LineWidth', 2); % Plot
zeros

% Title and axis details
title('Two-Sided Exponential ROC');
xlabel('Re(z)');
ylabel('Im(z)');
grid on;
axis equal;
xlim([-2 2]);
ylim([-2 2]);
legend('ROC', "Non ROC", 'Outer ROC Border', 'Inner ROC Border', 'Unit Circle',
'Poles', 'Zeroes');
```

Output




Conclusion:

This experiment demonstrates how to generate and plot left-sided, right-sided, and two-sided exponential sequences using MATLAB. It helps in understanding the behavior of discrete-time signals in different regions of the Z-domain.

**Post Lab Subjective/Objective type Questions:**

1. **What is the Region of Convergence (ROC), and how does it affect the stability and causality of a system?**
   The Region of Convergence (ROC) in the Z-transform is the range of values for the complex variable z where the Z-transform of a given sequence converges. In simpler terms, it's the region in the complex plane where the infinite summation of the Z-transform results in a finite value. The ROC plays a crucial role in determining two key properties of a system: stability

and causality. For stability, the ROC must include the unit circle |z|=1. This ensures that the system's impulse response is absolutely summable, meaning the system is bounded-input, bounded output (BIBO) stable. For causality, if a system is right-sided (exists for n≥0), the ROC lies outside the outermost pole. If a system is left-sided (exists for n<0), the ROC lies inside the innermost pole. So, the location of the ROC directly tells us whether the system is causal, anti-causal, or two-sided.

2. **How do you determine whether a system is stable using its ROC?**
   A system is stable if its ROC includes the unit circle |z|=1. After finding the poles of the system, we identify where the ROC lies relative to those poles. If the ROC surrounds the unit circle, the system is stable because it guarantees the system's response will not grow unbounded. If the unit circle is not part of the ROC, then the system is unstable because it means the system's response could potentially blow up, making it impossible to maintain a bounded output for a bounded input.

3. **If a system has a pole outside the unit circle, what does that imply about its stability?**
   If a system has a pole located outside the unit circle (meaning its magnitude is greater than 1), it implies that the system is unstable. For a causal system to be stable, all of its poles must be located inside the unit circle |z|<1. Having a pole outside this boundary suggests that the system's response will grow exponentially without bounds, leading to instability. So, one or more poles outside the unit circle automatically mean the system cannot be BIBO stable.

4. **How does the Z-transform compare to the Laplace transform? Discuss their similarities and differences.**
   The Z-transform and Laplace transform are both mathematical tools used to analyze signals and systems, but they are used in different contexts. The Z-transform is used for discretetime signals and systems, while the Laplace transform deals with continuous-time signals and systems. Both transforms convert time-domain signals into the complex frequency domain, which simplifies the analysis of system behavior, especially for stability and frequency response.

   In terms of similarities, both transforms make use of poles and zeros to describe system behavior and both require a Region of Convergence (ROC) for the transform to be valid. They also both relate to their respective Fourier transforms: the Z-transform becomes the Discrete-Time Fourier Transform (DTFT) when evaluated on the unit circle |z|=1, and the Laplace transform becomes the Continuous-Time Fourier Transform (CTFT) when evaluated along the jω-axis in the s-plane.

   As for differences, the Z-transform works with the complex variable z, whereas the Laplace transform uses s. The ROC for the Z-transform is typically a ring or annulus in the complex plane, while the ROC for the Laplace transform is a half-plane. Also, the Z-transform is mainly used in digital signal processing (DSP) and digital filter design, whereas the Laplace transform is common in analog signal processing and control systems.

**Signature of faculty in-charge with Date:**