MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Lab Report

Lab Report No : 03

Lab Report on : Study of Z-transform, inverse Z-transform, and residue method using MATLAB.

Course Title : Digital Signal Processing Lab

Course Code : ICT-3206

Submitted By	Submitted To
Name: Kuldip Saha Mugdha ID: IT-22018 3rd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	Dr. Mst. Nargis Akter Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Date of Performance: 16/11/25 Date of Submission: 23/11/25

Introduction:

The Z-transform is an important tool in Digital Signal Processing for analyzing discrete-time signals and systems. It converts time-domain sequences into the complex frequency domain, making system analysis easier. Inverse Z-transform is used to recover the original signal. Residue and polynomial division methods are commonly used techniques to find inverse Z-transforms. This experiment demonstrates these concepts using MATLAB.

Objective:

- To find the Z-transform and inverse Z-transform of discrete-time signals.
- To determine residues of a rational Z-transform function.
- To obtain inverse Z-transform using the polynomial division method.

Theory:

The **Z-transform** of a discrete-time signal $x(n)$is defined as

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

The **inverse Z-transform** converts $X(z)$back into $x(n)$.

The **residue method** is used to express a rational Z-transform as a sum of partial fractions.

The **polynomial division method** expands the Z-transform into a power series to obtain the time-domain sequence.

MATLAB Code:

Program 1: Z-transform and Inverse Z-transform

```
clc; clear all; close all;
syms n w0 z

a = n + 1;
disp('The input equation is:')
disp(a)

b = ztrans(a, n, z);
disp('The z-transform is:')
disp(b)

c = iztrans(b, z, n);
disp('The inverse z-transform is:')
disp(c)

% ---- Plot first signal ----
n_vals = 0:10;
a_vals = double(subs(a, n, n_vals));

figure;
stem(n_vals, a_vals, 'filled');
title('Signal 1: a[n] = n + 1');
xlabel('n');
ylabel('a[n]');
```

```
a1 = cos(w0*n);
disp('The input equation is:')
disp(a1)

b1 = ztrans(a1, n, z);
disp('The z-transform is:')
disp(b1)

c1 = iztrans(b1, z, n);
disp('The inverse z-transform is:')
disp(c1)

w0_val = pi/4;          % choose any w0 (e.g., π/4)
a1_vals = double(subs(a1, [n, w0], [n_vals; w0_val*ones(size(n_vals))]))

figure;
stem(n_vals, a1_vals, 'filled');
title('Signal 2: a_1[n] = cos(\omega_0 n)');
xlabel('n');
ylabel('a_1[n]');
```

Output:

```
The input equation is:
n + 1

The z-transform is:
z/(z - 1) + z/(z - 1)^2

The inverse z-transform is:
n + 1

The input equation is:
cos(n*w0)

The z-transform is:
(z*(z - cos(w0)))/(z^2 - 2*cos(w0)*z + 1)

The inverse z-transform is:
cos(n*w0)
```

Program 2: Residue Calculation

```
% Finding the residues of  Z^3 / ((z-0.5)*(z-0.75)*(z-1))

clc; clear all; close all;
syms z

d = (z - 0.5) * (z - 0.75) * (z - 1);   % The denominator of F(z)
a1 = collect(d);
den = sym2poly(a1);

num = [0 1 0 0];    % Numerator Z^3

[num1, den1] = residue(num, den);

fprintf('r1 = %4.2f \t', num1(1)); fprintf('p1 = %4.2f \t', den1(1));
fprintf('\nr2 = %4.2f \t', num1(2)); fprintf('p2 = %4.2f \t', den1(2));
fprintf('\nr3 = %4.2f \t', num1(3)); fprintf('p3 = %4.2f \t', den1(3));
fprintf('\n');
```

Output:

```
r1 = 8.00          p1 = 1.00
r2 = -9.00         p2 = 0.75
r3 = 2.00          p3 = 0.50
```

Program 3: Inverse Z-transform Using Polynomial Division

```
% Inverse Z-transform by the polynomial division method
% x(z) = (1 + 2*z^(-1) + z^(-2)) / (1 - z^(-1) + 0.3561*z^(-2))

clc; clear all; close all;

b = [1 2 1];
a = [1 -1 0.3561];

n = 5;   % number of power series points
b = [b zeros(1, n-1)];

[h, om] = deconv(b, a);

disp('The terms of inverse z-transforms')
disp(h)
```

Output:

```
The terms of inverse z-transforms
    1.0000    3.0000    3.6439    2.5756    1.2780
```

Result:

- Z-transform and inverse Z-transform of given signals were obtained successfully.
- Residues and poles of the given Z-domain function were calculated correctly.
- Inverse Z-transform using polynomial division was successfully determined.

Conclusion:

This experiment demonstrated the application of Z-transform and inverse Z-transform techniques using MATLAB. Residue and polynomial division methods were effectively used to analyze discrete-time systems.