University of Bahrain
College of Information Technology
Department of Computer Engineering

Experiment 5

Z transform

Prepared By

Name : Ali Redha Ali

ID: 20195330

Sec: 01

Course Number: ITCE340/272

Lab Objectives:

Intriduce the students to the following topics:

a- Z-transfor in MATLAB
b- How to solve difference equation using z- transform
c- How to implement IIR and FIR filter in MATLAB

Part 1: Z-Transform Description

ztrans(f)finds the Z-Transform of f. By default, the independent variable is nand the transformation variable is z. If fdoes not contain n, ztransuses symvar.

ztrans(f,transVar)uses the transformation variable transVarinstead of z.

ztrans(f,var,transVar)uses the independent variable varand transformation variable transVarinstead of nand z, respectively.

Examples

collapse all

Z-Transform of Symbolic Expression

Compute the Z-transform of sin(n). By default, the transform is in terms of z.

```
% First example

syms n
f =sin(n);
ztrans(f)`
ans =
(z*sin(1))/(z^2 - 2*cos(1)*z + 1)

```

Specify Independent Variable and Transformation Variable

Compute the Z-transform of `exp(m+n)`. By default, the independent variable is `n` and the transformation variable is `z`.

```
syms m n
f =exp(m+n);
ztrans(f);
```

```
ans =

(z*sin(1))/(z^2 - 2*cos(1)*z + 1)
```

Specify the transformation variable as `y`. If you specify only one variable, that variable is the transformation variable. The independent variable is still `n`.

)

```
ans =
(y*exp(m))/(y - exp(1))
```

Specify both the independent and transformation variables as m and y in the second and third arguments, respectively.

```
ztrans(f,m,y)

ans =
(y*exp(n))/(y - exp(1))
```

Z-Transforms Involving Dirac and Heaviside Functions

Compute the Z-transform of the Heaviside function and the binomial coefficient.

```
syms n z ztrans(heaviside(n-
3),n,z)

ans =
(1/(z - 1) + 1/2)/z^3

ztrans(nchoosek(n,2))

ans =
z/(z - 1)^3
```

Z-Transform of Array Inputs

Find the Z-transform of the matrix M. Specify the independent and transformation variables for each matrix entry by using matrices of the same size. When the arguments are nonscalars, ztrans acts on them element-wise.

```
syms a b c d w x y z
M = [exp(x) 1; sin(y) i*z]; vars
= [w x; y z]; transVars = [a b;
c d];
ztrans(M,vars,transVars)

ans =
[                             (a*exp(x))/(a - 1),                  b/(b - 1)]
[ (c*sin(1))/(c^2 - 2*cos(1)*c + 1), (d*1i)/(d - 1)^2]
```

If ztrans is called with both scalar and nonscalar arguments, then it expands the scalars to match the nonscalars by using scalar expansion. Nonscalar arguments must be the same size.

```
syms w x y z a b c d
ztrans(x,vars,transVars)

ans =
[ (a*x)/(a - 1),             b/(b - 1)^2]
[ (c*x)/(c - 1), (d*x)/(d - 1)]
```

Z-Transform of Symbolic Function

Compute the Z-transform of symbolic functions. When the first argument contains symbolic functions, then the second argument must be a scalar.

```
syms f1(x) f2(x) a b
f1(x) = exp(x); f2(x) =
x;
ztrans([f1 f2],x,[a b])
```

```
ans =
[ a/(a - exp(1)), b/(b - 1)^2]
```

If Z-Transform Cannot Be Found

If ztrans cannot transform the input then it returns an unevaluated call.

```
syms f(n)
f(n) = 1/n;
F = ztrans(f,n,z)
```

```
F =
ztrans(1/n, n, z)
```

Return the original expression by

using iztrans. iztrans(F,z,n)

```
ans =
1/n
```

Part 2: Solving Difference equation

A **discrete-time LTI system** can be represented as a (first order) **difference equation** of the form:

$$a_1 y[n] + a_2 y[n-1] = b_1 x[n] + b_2 x[n-1]$$

$$y[n] = (-a_2 y[n-1] + b_1 x[n] + b_2 x[n-1]) / a_1$$

This is hard to solve analytically, and we'd like to be able to perform some form of analogous manipulation like continuous time transfer functions, i.e.

$$Y(s) = H(s)X(s)$$

Example:

Calculate the output of a first order difference equation of a input signal $x[n] = 0.5^n u[n]$

$$0.5^n u[n] \overset{z}{\leftrightarrow} X(z) = \frac{z}{z-0.5}$$

System transfer function (z-transform of the impulse response)

$$y[n] - 0.8y[n-1] = x[n]$$

$$H(z) = \frac{1}{1-0.8z^{-1}} = \frac{z}{z-0.8}$$

The (z-transform of the) output is therefore:

$$Y(z) = \frac{z^2}{(z-0.5)(z-0.8)}$$

$$= \frac{1}{0.3}\left(\frac{0.8z}{(z-0.5)} - \frac{0.5z}{(z-0.8)}\right)$$

$$y[n] = (0.8 * 0.5^n u[n] - 0.5 * 0.8^n u[n]) / 0.3$$

Excrsize: Sove the above example using MATLAB

```
syms n

f = (0.8 * 0.5^n) - (0.5 * 0.8^n) / 0.3;
ztrans(f)
```

```
ans =

(4*z)/(5*(z - 1/2)) - (5*z)/(3*(z - 4/5))
```

Part 3: IIR and FIR Digital Filter

("finite impulse response") and IIR ("infinite impulse response") filtering.

IIR filters are the most efficient type of filter to implement in DSP (digital signal processing). Difficult to control, can be unstable, has analog History

An FIR filter requires more computation time on the DSP and more memory. Easy to control, Always stable, No analog history

A discrete time LTI filter can be expressed as a difference equation

$$a(1)y[n] = b(1)x[n] + b(2)x[n-1] + \cdots + b(n_b+1)x[n-n_b]$$

$$-a(2)y[n-1] - \cdots - a(n_a+1)y[n-n_a]$$

If all a(i)'s are zero except a(1) we call it FIR,

otherwise it's IIR MATLAB:

z = filter(b,a,x)

Creates filtered data y by processing the data in vector x with the filter described by vectors a and b.

Example:

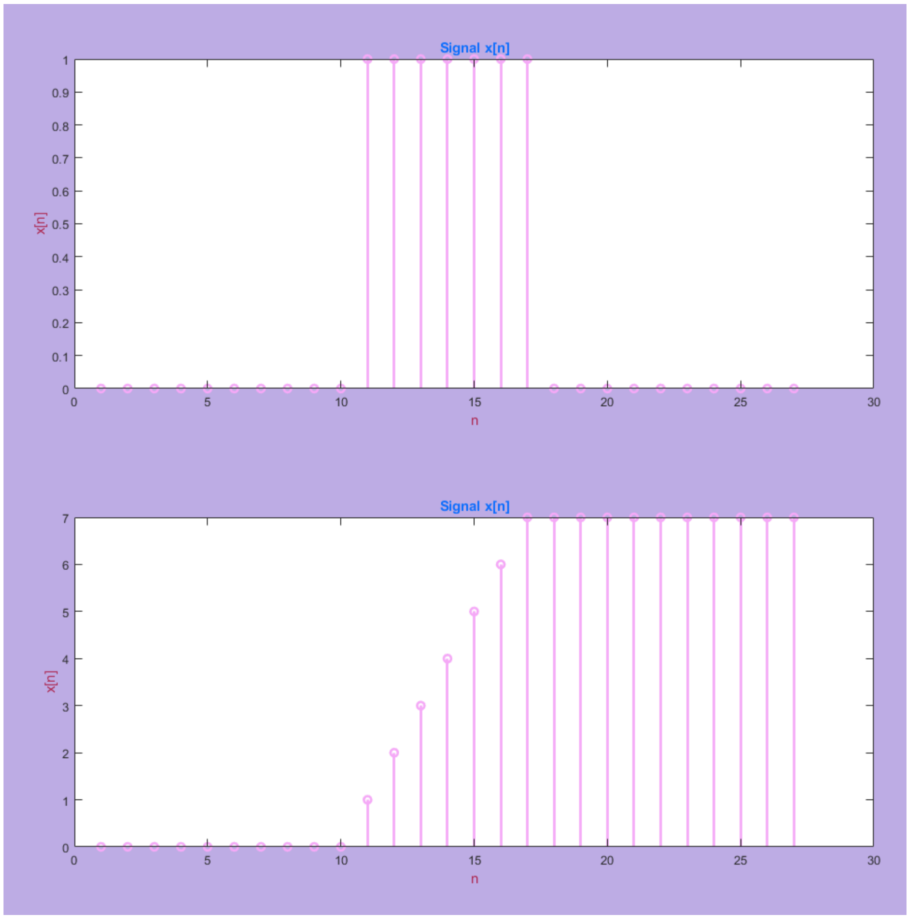
- Input: x[n] is the rectangular signal
- The difference equation describing the filter is

$$y[n] - y[n-1] = x[n]$$

MATLAB code:

```
x = [zeros(1, 10) ones(1, 7) zeros(1, 10)];
a = [1, -1];
b = 1;
z = filter (b, a, x);
f14 = figure("Name", 'Signals');
set(f14, 'color', '#BDACE4');
title('{\bf Method 1 :Residue }', 'color', ...
    '#0d6efd','FontSize', ...
     14,'FontName' ...
    ,'TimeNewRoman'); grid on
subplot(2, 1, 1)
stem(x, 'color', '#F5A9F7', 'LineWidth', 2)
subplot(2, 1, 2)
stem(z, 'color', '#F5A9F7', 'LineWidth', 2)

```


Signal x[n]
x[n]
n
Signal x[n]
x[n]
n

Exercise

1.Write a MATLAB code to implement the following filter expressed by the following equation

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

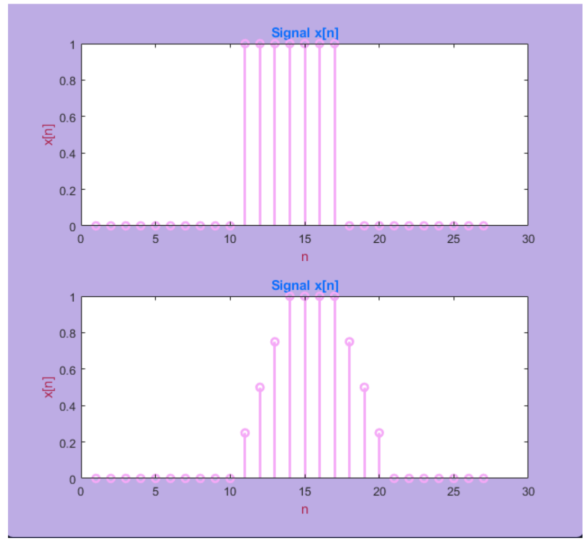
```
x = [zeros(1, 10) ones(1, 7) zeros(1, 10)];
a = [1];
b = [1/4 1/4 1/4 1/4];
z = filter (b, a, x);
f14 = figure("Name", 'Signals');
set(f14, 'color', '#BDACE4');

subplot(2, 1, 1)

stem(x, 'color', '#F5A9F7', 'LineWidth', 2)

xlabel('n', 'color', '#ad2750')
ylabel('x[n]', 'color', '#ad2750')
title('Signal x[n]', 'color', '#0d6efd')
subplot(2, 1, 2)

stem(z, 'color', '#F5A9F7', 'LineWidth', 2)
title('Signal x[n]', 'color', '#0d6efd')
xlabel('n', 'color', '#ad2750')
ylabel('x[n]', 'color', '#ad2750')
```



2. what is the type of the filter

Finite impulse response

3. use the filter to

For applications requiring linear phase and reasonable memory and computational resources, FIR filters (finite impulse response) are generally selected. Signal enhancement applications in audio and biomedicine have widely deployed them