MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO
INSTITUTE OF INFORMATION & COMMUNICATION TECHNOLOGIES

ADVANCED DIGITAL SIGNAL PROCESSING

Lab 5: Z-Transform

Objective

The objective of this lab is to check and evaluate the stability as well as the inverse z-transform of the system.

1 Introduction

When LTI systems are described by a difference equation,

$$y(n) + \sum_{k=1}^{N} a_k y(n-k) = \sum_{l=0}^{M} b_l x(n-l) \tag{1}$$

the system function $H(z)$ can easily be computed. Taking the z-transform of both sides, and using properties of the z-transform,

$$Y(z) + \sum_{k=1}^{N} a_k z^{-k} Y(z) = \sum_{l=0}^{M} b_l z^{-l} X(z) \tag{2}$$

or

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)} &= \frac{\sum_{l=0}^{M} b_l z^{-l}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \\ &= \frac{b_0 z^{-M}\left(z^M + \cdots + \dfrac{b_M}{b_0}\right)}{z^{-N}\left(z^N + \cdots + a_N\right)} \end{aligned}$$

After factorization, we obtain,

$$H(z) = b_0 z^{N-M} \frac{\prod_{l=1}^{N}(z - z_l)}{\prod_{k=1}^{N}(z - p_k)} \tag{3}$$

Where z_l represent the system *zeros* and p_k are the system *poles*. Thus $H(z)$ (and hence an LTI system) can also be represented in the z-domain using a pole-zero plot. This fact is useful in designing simple filters by proper placement of poles and zeros.

z-transform is an extremely useful mathematical tool for the analysis and design of discrete time systems. The z-transfer function, also known as the *Pulse Transfer Function H(z)* is defined as the ratio of the z-transform of the output to the z-transform of the input when all initial conditions are zero.

Mathematically,

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{4}$$

The roots of the denominator of $H(z)$ are called "*poles*" and those of the numerator are called "*zeros*". The above equation has M zeros and N poles.

A system is said to be stable if all of its poles lie inside of a unit circle on the z-plane. The following example illustrates the use of MATLAB to compute zero and poles of a transfer function and to plot them onto the z-plane.

1.1 Example

Given a causal system, $y(n) = 0.9y(n-1) + x(n)$. Determine $H(z)$ and sketch its pole-zero plot.

The difference equation can be put in the form,
$y(n) - 0.9y(n-1) = x(n)$
From equation (3), we get,

$$H(z) = \frac{1}{1 - 0.9z^{-1}} \qquad |z| > 0.9$$

Since the system is causal. There is one pole at 0.9 and one zero at the origin. We will use MATLAB to illustrate the use of the `zplane` function.

```
clc, clear all, close all
num = [1, 0]; % numerator of H(z)
den = [1, -0.9]; % denominator of H(z)
zplane(num,den)
```

Note that we specified $num = [1, 0]$ instead of $num = 1$ because the `zplane` function assumes that scalars are *zeros* or *poles*.

Lets do one more example and find the poles and zeros of the following pulse transfer function and plot them onto the z-plane.

$$H(z) = \frac{2 - z^{-1}}{1 - 0.1z^{-1} - 0.02z^{-2}}$$

The pole-zero plot of a rational z-transform $H(z)$ can be readily obtained using the function `zplane`. If the z-transform is given in the form of a rational function as in Eq. (4), the command to use is `zplane(num,den)` where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $H(z)$ in ascending powers of z^{-1}.

The function `tf2zp` can be used to determine the *zeros* and *poles* of a rational z-transform $H(z)$. The program statement to use is `[z,p,k] = tf2zp(num,den)` where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $H(z)$ in ascending powers of z^{-1} and the output file contains the gain constant `k` and the computed *zeros* and *poles* given as column vectors `z` and `p`, respectively.

```
clc, clear all, close all
num = [2 -1]; % numerator of H(z)
den = [1 -0.1 -0.02]; % denominator of H(z)
zplane(num,den)
[z,p,k] = tf2zp(num,den);
m = abs(p); % to find the distance from origin
    of poles
disp(['Zeros = [', num2str(z') ']'])
disp(['Poles = [', num2str(p') ']'])
disp(['Gain = [', num2str(k') ']'])
disp(['Radius of poles = [', num2str(m') ']'])
```

On the z-plane, "×" indicate the poles and "○" indicate the zeros. Looking at the outcome, is this system stable? if so, why?

2 Inverse z-transform

As we know, there are two methods to find inverse z-transform,

(i) Partial fraction method

(ii) Power series method

MATLAB has build-in functions to find the inverse z-transform using both of the above methods.

2.1 Partial fraction method

The inverse z-transform computation requires an evaluation of a complex contour integral that, in general, is a complicated procedure. The most practical approach is to use the partial fraction expansion method. The z-transform, however, must be a rational function.

A closed-form expression for the inverse of a rational z-transform can be obtained by first performing a partial-fraction expansion using the function `residuez` and then determining the inverse of each term in the expansion by looking up a table of z-transforms. The function `residuez` can also be used to convert a z-transform given in the form of a partial-fraction expansion to a ratio of polynomials in z^{-1}.

2.1.1 Example

Determine the partial fraction expansion of the following pulse transfer function.

$$H(z) = \frac{z}{3z^2 - 4z + 1}$$

First rearrange $H(z)$ so that it is a function in ascending powers of z^{-1}

$$H(z) = \frac{z^{-1}}{3 - 4z^{-1} + z^{-2}} = \frac{0 + z^{-1}}{3 - 4z^{-1} + z^{-2}}$$

Now using the MATLAB script,

```
num = [0 1];
den = [3 -4 1];
[r,p,k] = residuez(num,den)
```

The result of the program is shown below,

```
r =
    0.5000
   -0.5000
p =
    1.0000
    0.3333
k =
      []
```

Hence, we obtain,

$$H(z) = \frac{1/2}{1 - z^{-1}} - \frac{1/2}{1 - 1/3z^{-1}}$$

$H(z)$ has two poles: $z_1 = 1$ and $z_2 = 1/3$. Further, time-domain representation can be done using z-transform table. For instance,
$h(n) = \frac{1}{2}u(n) - \frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$
Similarly, to convert back to the rational function form, we use the following MATLAB command.

```
[num,den] = residuez(r,p,k)
```

Results,

```
num =
   -0.0000     0.3333
den =
    1.0000    -1.3333     0.3333
```

Hence,

$$H(z) = \frac{0 + 1/3z^{-1}}{1 - 4/3z^{-1} + 1/3z^{-2}} = \frac{z^{-1}}{3 - 4z^{-1} + z^{-2}}$$
$$= \frac{z}{3z^2 - 4z + 1}$$

2.2 Power series method

To find inverse z-transform by power series, one may use the built-in function `impz`. It provides the samples of the *time-domain* sequence, which is assumed to be causal. There are three versions of this function: `[g,t] = impz(num,den)`, `[g,t] = impz(num,den,L)`, and `[g,t] = impz(num,den,L,FT)`, where `num` and `den` are row vectors containing the coefficients of the numerator and denominator polynomials of $H(z)$ in ascending powers of z^{-1}, `L` is the desired number of the samples of the inverse transform, `g` is the vector containing the samples of the inverse transform starting with the sample at `n = 0`, `t` is the length of `g`, and `FT` is the specified sampling frequency in Hz with default value of unity.

Use on-line help facility of MATLAB to find-out more about this built-in function. You may also use MATLAB function `filter` for the same purpose.

2.2.1 Example

Determine the inverse z-transform of the following pulse transfer function.

$$H(z) = \frac{1 + 2z^{-1}}{1 + 0.4z^{-1} - 0.12z^{-2}}$$

```
clc, clear all, close all
L = 20; % length of output vector
num = [1 2];
```

```
den = [1 0.4 -0.12];
[g,t] = impz(num,den,L); % impulse response
disp(['Coefficients of the power series
    expansion = [', num2str(g') ']'])
% Plot the impulse response
stem(g);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

We can use built-in function `filter` to achieve the same outcome.

```
clc, clear all, close all
L = 20; % length of output vector
num = [1 2];
den = [1 0.4 -0.12];
x = [1 zeros(1,L-1)];
y = filter(num,den,x);
disp(['Coefficients of the power series
    expansion = [', num2str(y) ']'])
```

3 Impulse and step response of discrete time systems

The impulse response and step response of an LTI discrete time system can easily be plotted using the built-in functions `dimpulse` and `dstep`.

To compute the step response for $n = 0$ to $n = 10$, type the following commands.

```
clc, clear all, close all
n = 0:10;
num = [1 2];
den = [1 0.4 -0.12];
y = dstep(num,den,length(n));
```

The response can then be plotted using the `stem` plot. To compute the `impulse` response, simply replace `dstep` with `dimpulse` in the above commands.

Note that the impulse response of a stable system always settles to zero and the step response of a stable system always settles to a constant value. For unstable systems, on the other hand, these responses grow without bound. Marginally stable systems produce cycling or oscillating behavior.

4 Exercise

1. Find poles and zeros of the following pulse transfer function and plot them onto the z-plane.

$$H(z) = \frac{2.25 - 2.1z^{-1} - 3.95z^{-2} - 1.6z^{-3} - 0.2z^{-4}}{4 - 2.96z{-}1 + 0.8z^{-2} - 0.1184z^{-3} - 0.0064z{-}4}$$

2. Determine the partial fraction expansion of the following pulse transfer function.

$$H(z) = \frac{18z^3}{18z^3 + 3z^2 - 4z - 1}$$

3. Find the inverse z-transform of the transfer function given in example 2.1.1 using the power series method.

4. Plot impulse response and step response of the transfer function given in example 1.1 and 2.1.1. Also try the built-in function `filter` to plot the same responses.