Faculty of Applied Science and Technology

Laboratory 4 Introduction to Z-Transform

Student Name: Michael McCorkell Student Number: N01500049 Date: March 3rd 2025

Note 1: This is an individual lab. Please bring your laptop and install required software to complete this lab.
Note 2: For your lab report, please take necessary screenshots and proper captions and include them in the corresponding parts of the lab. Demonstrate your work during the lab time. Submit the report file and the related .m files on the course website.

1. Learning outcome:
1.1 Familiarize with symbolic z-transform in MATLAB.
1.2 Use of the z-transform to analyze a common digital signal process tool.
1.3 Convert transfer functions from the s-domain to the z-domain.

2. Background
2.1 z-transform in MATLAB:
The z-transform is the discrete-time equivalency to the Laplace transform. It expresses DT signals as linear combinations of DT complex exponentials.

The general format of the bilateral z-transform is:

$$F(z) = \mathcal{Z}\{f[n]\} = \sum_{n=-\infty}^{\infty} f[n]z^{-n}$$

The inverse Laplace transform is:

$$x[n] = \mathcal{Z}^{-1}\{F(z)\} = \frac{1}{2\pi j}\oint_{\Gamma} X(z)z^{n-1}dz$$

In MATLAB, the transforms are handled using ztrans() and iztrans().
The detailed description of the functions can be found in the document linked here.

2.2 z-domain and s-domain conversion
When DT sequence has been transformed by a z-transform, it is now in the z-domain. In our lecture, we have reviewed the connection between the s- and z-domain. It is connected by the relation below:

$$z = e^{sT_s} \quad \Leftrightarrow \quad s = \frac{1}{T_s}\ln(z)$$

Once the relationship between these two domains is established, in order to convert an analog transfer function into the digital domain. Please note that since this mapping relation is nonlinear, the zeros and poles of a sampled DT systems from its original CT system is also nonlinear.

3. Procedures

3.1 Mapping between the s-plane and the z-plane

Given a CT signal $x(t)$ and its Laplace transform $X(s)$, the poles of is $X(s)$ are:

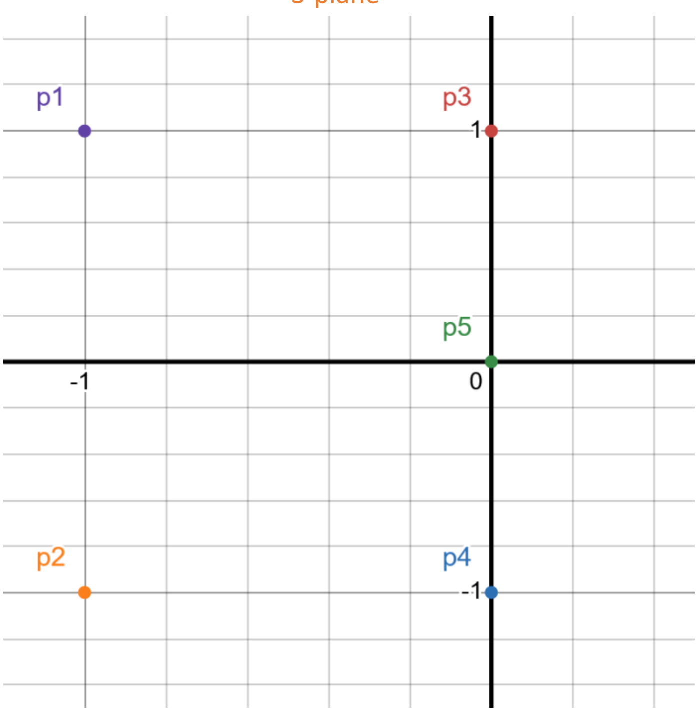
$$p_{1,2} = -1 \pm j$$
$$p_{3,4} = \pm j$$
$$p_5 = 0$$

Assuming we sampled this signal at $f_s = 1\ Hz$.

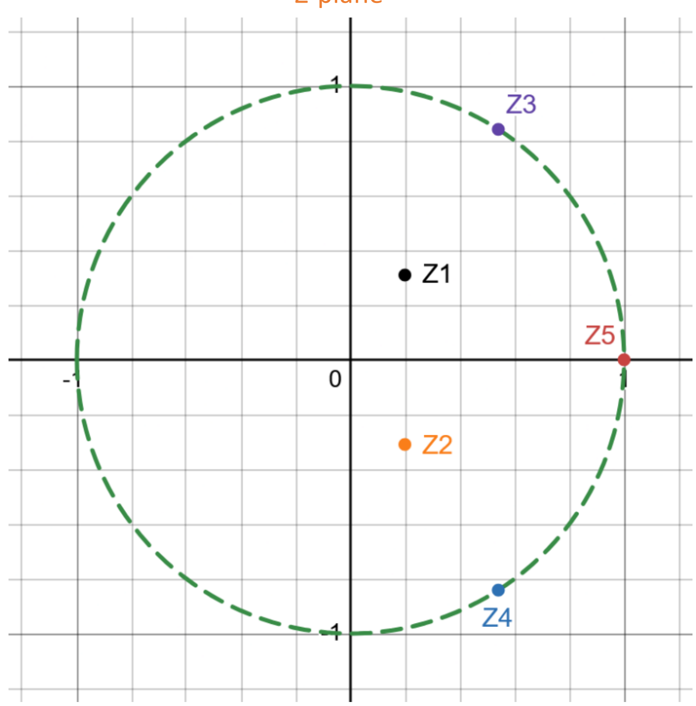
Question 1 (10%). Determine where the poles are mapped in the z-plane WITHOUT using MATLAB. Plot the poles in the s-plane and the z-plane respectively.

$e^{-1} \approx 0.3679;$ $e^{\pm j} = \cos(1) \pm j\sin(1)$ $\cos(1) \approx 0.5403, \sin(1) \approx 0.8415$ $z_{1,2} = 0.3679(0.5403 \pm j0.8415)$ $z_{1,2} \approx 0.1988 \pm j0.3097$	$e^{\pm j} = \cos(1) \pm j\sin(1)$ $z_{3,4} \approx 0.5403 \pm j0.8415$	$z_5 = e^{0T} = e^0 = 1$

S-plane

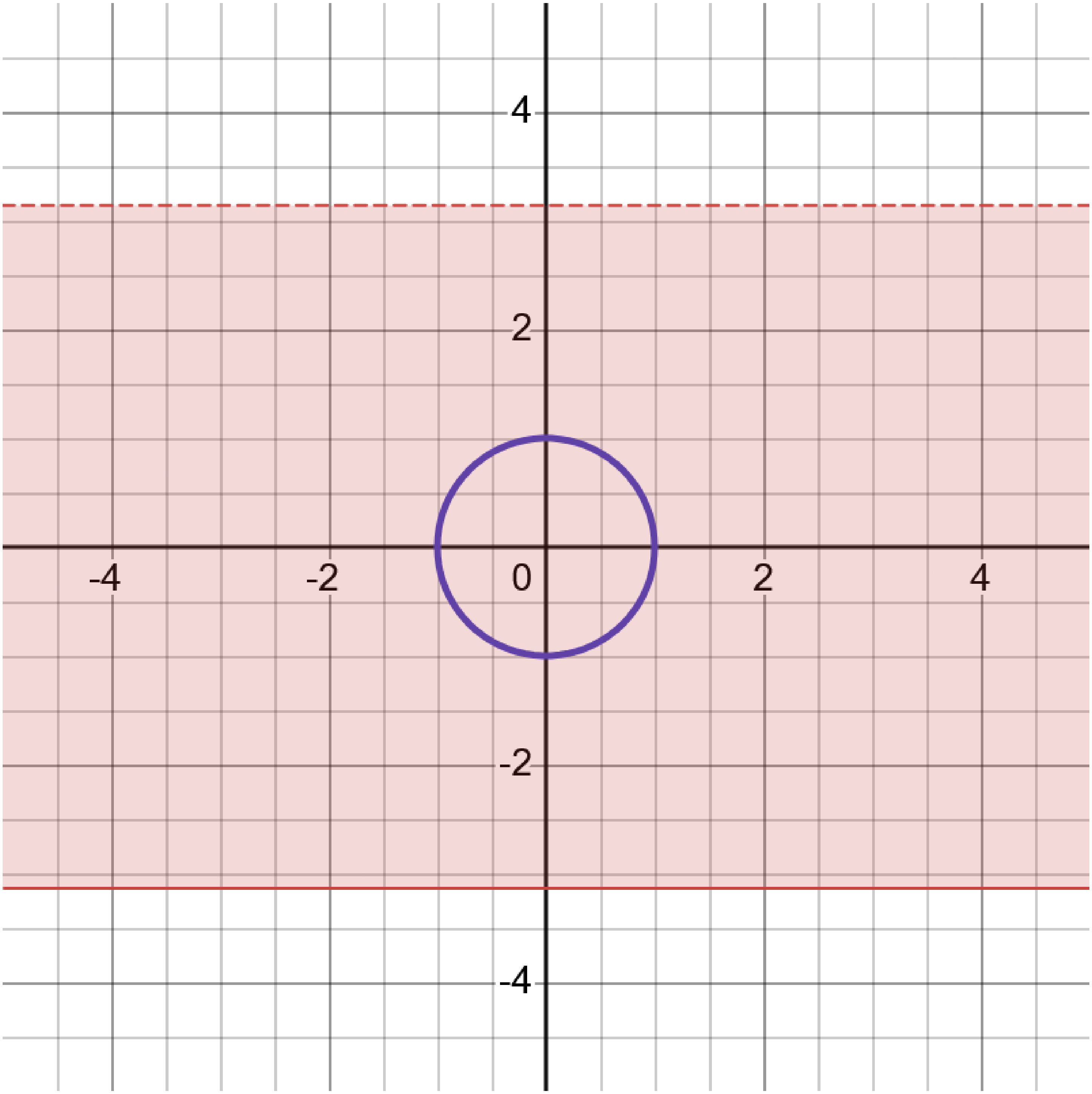


Z-plane



Question 2 (10%). Unit circle is a very important landmark in the z-plane. It is defined as $z = 1e^{j\omega}, -\pi \leq \omega < \pi$ in the z-plane. Determine where the mapping of the unit circle into the s-plane. Demonstrate it in the space below. Also, in the s-plane, indicate where the inside of the unit circle is mapped to and where the outside of the unit circle is mapped to.

$z = e^{iw} \rightarrow s = iw$



3.2 z-Transform

The z-transform is very useful in analyzing digital systems. Consider a DT LTI system with impulse response $h[n] = h[n] = \delta[n - n_0]$. Please use z-transform to find its system function $H(z)$.

Question 3 (10%). What is the system function $H(z)$? In practice, what does this system do?

$$H(z) = \sum_{n=-\infty}^{\infty} \delta[n - n_0] z^{-n} \rightarrow H(z) = z^{-n_0}$$

Shift operator/pure delay system.

Question 4 (10%). What is the system function $H(z)$ for a DT LTI system with impulse response $h[n] = \sum_{k=1}^{n_0} \delta[n - k]$? In practice, what does this system do?

$$h[n] = \sum_{k=1}^{n_0} \delta[n - k] \rightarrow H(z) = \sum_{k=1}^{n_0} z^{-k}$$

noise reduction

3.3 Simple Moving Average filter

Now we are going to use a simple DT system to explore the power of z-transform based analysis. This system takes N sample points at the input and computes the average of those N samples as a single output. It is commonly known as the N-tap Simple Moving Average (SMA) filter and can be used to reduce unwanted noises. A system that carries out smooth moving average can be described as below:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n - k]$$

Here N is the total number of samples used to compute the average, and is often referred to as the length of the filter or the number of taps.

The block diagram of an N-tap SMA filter is shown in Figure 1.

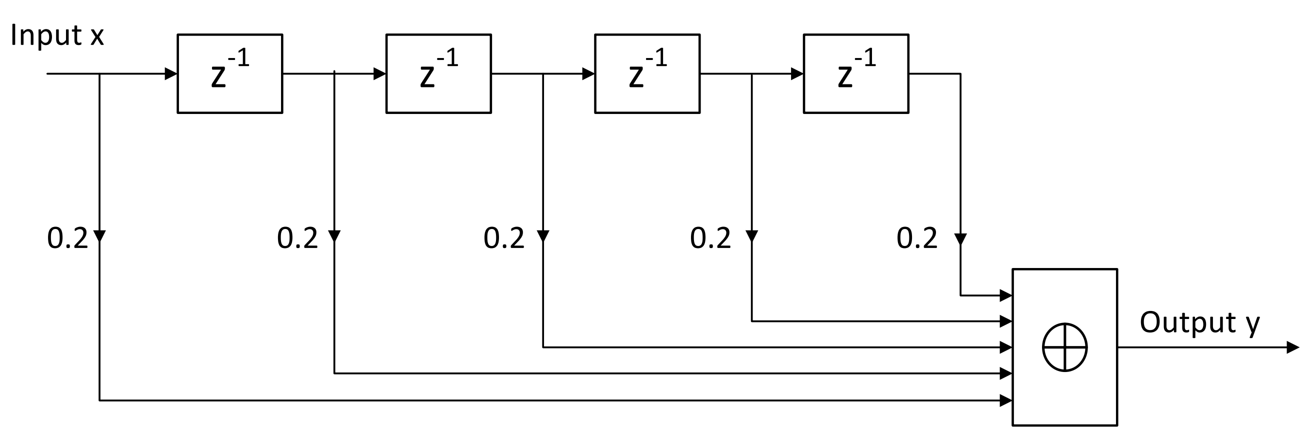


Figure 1. Block diagram of a SMA filter.

Question 5 (5%). What is the length or the number of taps of this filter?

5 taps (filter length).

Question 6 (5%). What is the purpose of including multiple blocks of z^{-1} in this design?

Storing past inputs- This system takes N sample points at the input and computes the average of those N samples as a single output. $\frac{1}{N}$

Question 7 (10%). Compute the transfer function $H(z) = \frac{Y(z)}{X(z)}$ of a SMA filter with length N=5.

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k] \rightarrow H(z) = \sum_{n=0}^{4} \frac{1}{5} z^{-n} \rightarrow H(z) = \frac{1}{5}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}) \rightarrow H(z) = \frac{1 - z^{-5}}{5(z-1)}$$

Question 8 (20%). Is this filter a low pass, high pass or band pass filter? Please provide your rationale without using MATLAB.

It's a low-pass filter. The filter smooths out abrupt signal fluctuations by averaging the previous five samples. Consequently, the high-frequency components are removed. As the amplitude of H(z) gets closer to zero, the signal is muffled at high frequencies (ω≈π). Low-frequency components are retained while high-frequency noise is eliminated since this filter is a low-pass filter.

In MATLAB, use digital filter function such as filter() to implement this SMA filter. y = filter(b,a,x) filters the input data x using a rational transfer function defined by the numerator and denominator coefficients b and a. The syntax of filter() can be found here. Use the following code snippet to create a sinusoidal signal that is contaminated by random noise:

```
t = linspace(-pi,pi,100);
rng default  %initialize random number generator
x = sin(t) + 0.25*rand(size(t));
```

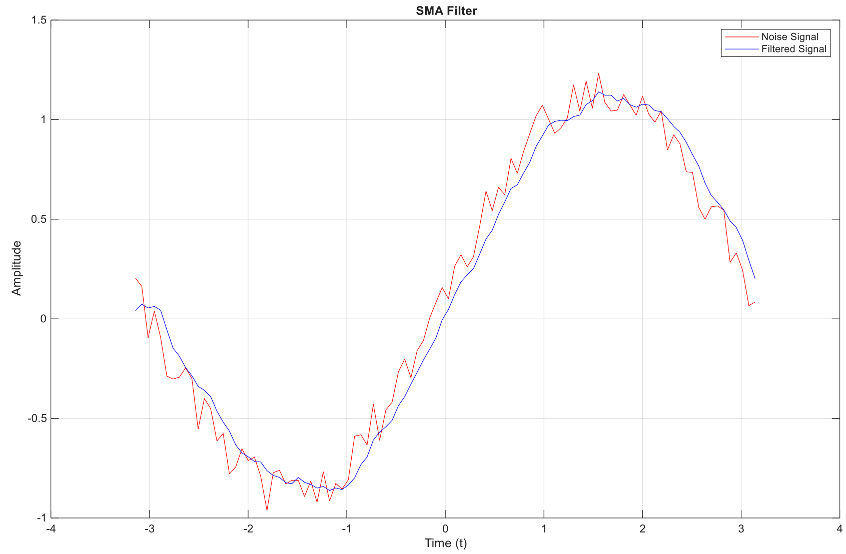
Task 1 (10%). Apply SMA filter using MATLAB filter() function and compare the before and after signal. Record your code and the comparison plot of x and y.

```
t = linspace(-pi, pi, 100);
rng default;
x=sin(t)+0.25*rand(size(t));

b=ones(1,5)/5;

y=filter(b,1,x);

figure;
plot(t, x, 'r'); hold on;
plot(t,y,'Color','b');
legend('Noise Signal', 'Filtered Signal');
xlabel('Time (t)'); ylabel('Amplitude');
title('SMA Filter'); grid on;
hold off;
```



Task 2 (10%). Based on the nature of this SMA filter, what will happen if you increase the filter length N. Explain your reasoning here based on the transfer function of the system.

The choice of N depends on whether you want to prioritize noise reduction or signal sharpness; a smaller N keeps crisp features while retaining noise, while a larger N provides more smoothing, reducing noise but possibly distorting the signal.