Laboratory

10

FIR Filters and Z-Transform

Objective

The goal of this lab is to learn how to implement FIR filters in MATLAB, and then study the response of FIR filters to various signals. In addition, we will use FIR filters to study the convolution operation and properties such as linearity and time-invariance. In the second of the experiment, we will be familiar in converting discrete time signals to Z domain and plotting the poles and zeros of the system to determine the properties of the system such as causality and stability.

Introduction

- **Finite Impulse Response Filter Definition**

  In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying).
  FIR filters can be discrete-time or continuous-time, and digital or analog.

  As we know that the output a linear time invariant system is determined by convolving its input signal with its impulse response.

  For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence y[n] in terms of its input sequence x[n] (see Figure 1):

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \\ &= \sum_{i=0}^{N} b_i x[n-i] \end{aligned}$$

  where :
  - **x[n]** is the input signal,

- **y[n]** is the output signal,
- **b_i** are the filter coefficients, also known as tap weights, that make up the impulse response.

- **N** is the filter order; an **Nth-order** filter has **(N+1)** terms on the right-hand side. The **x[n -i]** in these terms are commonly referred to as taps, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a **5th order/6-tap** filter.

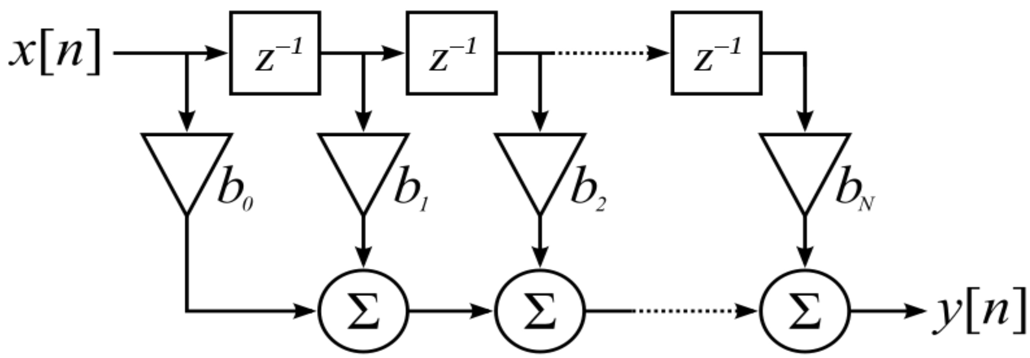


Figure 1 : A discrete-time FIR filter of order N. The top part is an N-stage delay line with N + 1 taps. Each unit delay is a z−1 operator in Z-transform notation.

- ## Finite Impulse Response Filter Properties

An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter. FIR filters:

1. **Require no feedback**. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.
2. **Inherently stable**. This is due to the fact that, because there is no required feedback, all the poles are located at the origin and thus are located within the unit circle (the required condition for stability in a Z transformed system).

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity.

- ## Impulse Response

The impulse response h[n] can be calculated if we set x[n]=δ[n] in the above relation, where δ[n] is the delta impulse. The impulse response for an FIR filter then becomes the set of coefficients b_n, as follows :

$$h[n] = \sum_{i=0}^{N} b_i \delta[n-i] = b_n$$

for n=0 to N.

- **Filter Design**

To design a filter means to select the coefficients such that the system has specific characteristics. The required characteristics are stated in filter specifications. Most of the time filter specifications refer to the frequency response of the filter. There are different methods to find the coefficients from frequency specifications:

1. Window design method.
2. Frequency Sampling method.
3. Weighted least squares design.

- **Z-Transform**

The Z-transform converts a discrete time-domain signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.

The Z-transform, like many integral transforms, can be defined as either a one-sided or two-sided transform.

Bilateral Z-transform

The bilateral or two-sided Z-transform of a discrete-time signal x[n] is the formal power series X(z) defined as

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

Unilateral Z-transform

Alternatively, in cases where x[n] is defined only for n ≥ 0, the single-sided or unilateral Z-transform is defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n}.$$

- **Region of convergence**

The region of convergence (ROC) is the set of points in the complex plane for which the Z-transform summation converges.

$$ROC = \left\{ z : \left| \sum_{n=-\infty}^{\infty} x[n] z^{-n} \right| < \infty \right\}$$

Example 1 (causal ROC)

let $x[n] = 0.5^n u[n]. Expanding\ x[n] on\ interval\ (-\infty, \infty)\ it\ becomes$

$x[n] = \{\ldots \ldots, 0,0,0,0,0,1,0.5,0.25,0.5^3, \ldots \ldots.\}$

$$\sum_{n=-\infty}^{\infty} x[n]z^{-n} = \sum_{n=0}^{\infty} 0.5^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{0.5}{z}\right)^n = \frac{1}{1 - 0.5z^{-1}}.$$

The last equality arises from the infinite geometric series and the equality only holds if $|0.5Z^{-1}|$ which can be rewritten in terms of Z as$|Z| > 0.5$. Thus, the ROC is $|Z| > 0.5$. In this case the ROC is the complex plane with a disc of radius 0.5 at the origin "punched out" (see Figure 1).

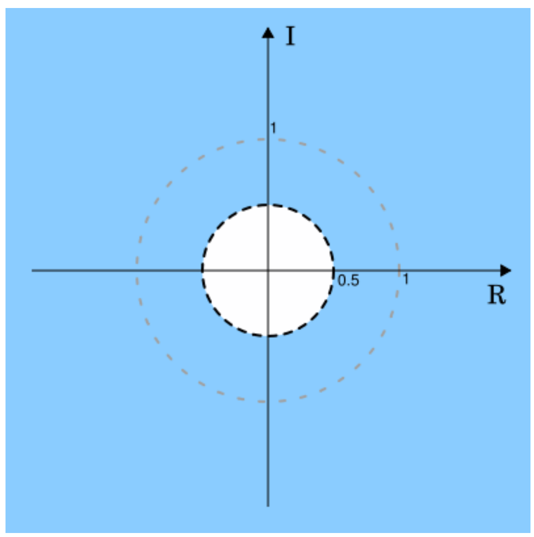


Figure 1 : ROC shown in blue, the unit circle as a dotted grey circle (appears reddish to the eye) and the circle $|Z| > 0.5$ is shown as a dashed black circle

Example 2 (anticausal ROC)

let $x[n] = -0.5^n u[-n-1]. Expanding\ x[n] on\ interval\ (-\infty, \infty)\ it\ becomes$

$x[n] = \{\ldots \ldots, -(0.5)^{-3}, -(0.5)^{-2}, -(0.5)^{-1}, 0,0,0,0,0, \ldots \ldots.\}$

$$\sum_{n=-\infty}^{\infty} x[n]z^{-n} = -\sum_{n=-\infty}^{-1} 0.5^n z^{-n} = -\sum_{n=-\infty}^{-1} \left(\frac{z}{0.5}\right)^{-n}$$

$$= -\sum_{m=1}^{\infty}\left(\frac{z}{0.5}\right)^m = 1 - \frac{1}{1 - 0.5^{-1}z}$$

.

The equality only holds if $|0.5^{-1}Z\ \ |$which can be rewritten in terms of Z as $|Z| < 0.5$. Thus, the ROC is $|Z| < 0.5$. In this case the ROC is a disc centered at the origin and of radius 0.5 (see Figure 2).

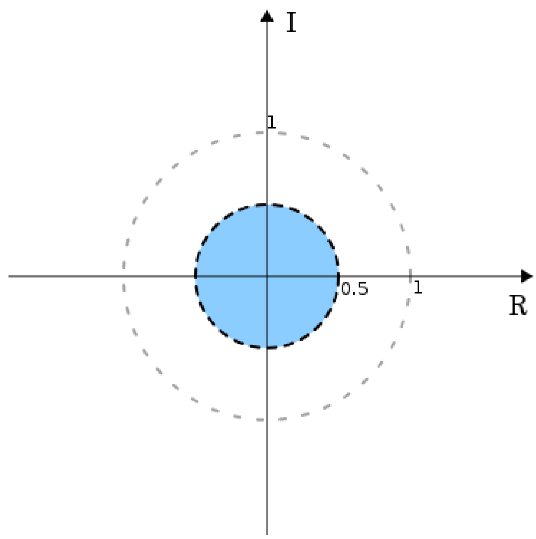


Figure 2: ROC shown in blue, the unit circle as a dotted grey circle and the circle $|Z| = 0.5$ is shown as a dashed black circle

- ## Poles and Zeros

The **zeros** of a z-transform X(z) are the values of z where X(z) = 0.
The **poles** of a z-transform X(z) are the values of z where X(z) = ∞. (cf. mesh plot of X(z))

A **pole-zero** plot is a graphic description of rational X(z), up to the scale factor. Use o for zeros and × for poles. Multiple poles or zeros indicated with adjacent number.

By definition, the ROC will not contain any poles.

- ## Analysis of LTI Systems by Z-Transform

Due to its convolution property, the z-transform is a powerful tool to analyze LTI systems

$$y[n] = h[n] * x[n] \xrightarrow{Z} Y(z) = H(z)X(z)$$

Causality of LTI systems

The LTI system is causal *iff* its impulse response h[n] is 0 for n < 0.

For a causal system, the system function (assuming it exists) has a series expansion that involves only non-positive powers of *Z*:

$$H(z) = \sum_{n=0}^{\infty} h[n]\, z^{-n} = h[0] + h[1]\, z^{-1} + h[2]\, z^{-2} + \cdots .$$

An LTI system with impulse response h[n] is **causal iff** the ROC of the system function is the exterior of a circle of radius $r < \infty$ ***including*** z = ∞, i.e., ROC = $\{r < \lvert Z\rvert \leq \infty\}$, or, in the trivial case of a memoryless system, ROC = $\{0 \leq \lvert Z\rvert \leq \infty\}$

Stability of LTI systems

The LTI system is BIBO stable ***iff*** $\sum_{n=-\infty}^{\infty} \lvert h[n]\rvert < \infty$.

An LTI system is BIBO **stable *iff*** the ROC of its system function includes the unit circle.

Procedure

In this section , you will be able to define and apply **FIR** filters as a discrete-time system that converts **x[n]** into output signal **y[n]** using built-in MATLAB functions. After that, you will find the impulse response **h[n]** for a given FIR filter in order to use it to generate output signal **y[n]** by convoluting **h[n]** with **x[n].** Also, you should be able to design a FIR filters for a given specifications.

- **Defining and Applying FIR Filters**

Consider the system for which the outputs are given by

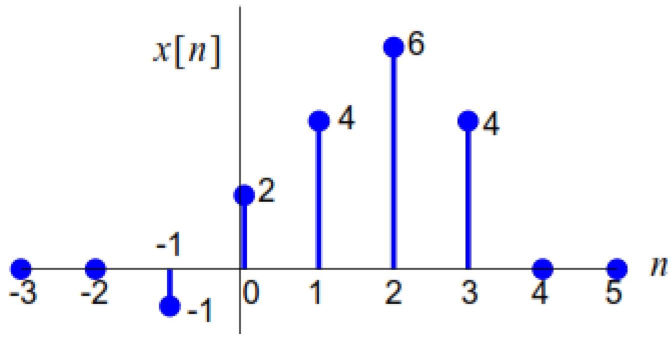
$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

The filter coefficients $\{b_k\}$ are constants that define the filter behavior, which are :

$b_0 = \frac{1}{3}$, $b_1 = \frac{1}{3}$, $b_2 = \frac{1}{3}$

The applied filter is one of the simplest **FIR** filters which is called **Moving Average Filter** .

Consider a *finite-length* input sequence **x[n]** having (non-zero values) over the interval -1≤n≤3 , where the values of x[n] as follows:



The following MATLAB code shows how the given **FIR** filter can be applied on the given sequence of inputs **x[n]** and plotted the output signal the output signal **y[n]** (see Figure 2) using built-in MATLAB functions.

```
n = -3:5;
x=[ 0 0 -1 2 4 6 4 0 0];
B=1/3*[1 1 1];
y=filter(B,1,x);
stem(n,y,'filled');
```

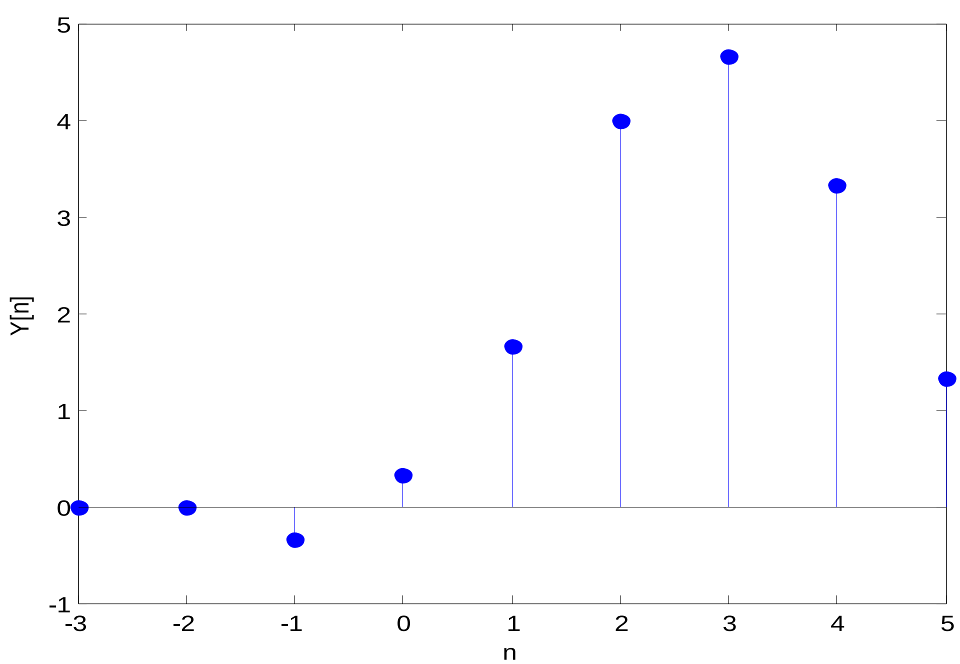


Figure 2 : Output signal y[n] after applying FIR filter on x[n]

Notes:

The action of the moving average filter has resulted in the output being smoother than the input.

Since only past and present values of the input are being used to calculate the present output, this filtering operation can operate in real-time.

- **Filtering a Sinusoidal Sequence with a Moving Average Filter**

Consider the system for which the outputs are given by

$$y[n] = h[n] * x[n],\ where$$

$$h[n] = \frac{1}{4}(\delta[n] + \delta[n-1] + \delta[n-2] + \ \delta[n-3])$$

$$x[n] = \cos(\ w_o * n)\ \ for\ w_o\ =\ \pi/8$$

The following MATLAB code shows how the given **FIR** filter, which is a **h[n]** in this example, can be applied on an input signal **x[n]** and then plotting the output signal **y[n]** (see Figure 3) using built-in MATLAB functions.

```
w0=pi/8;
n= 0:100;
h=[1 1 1 1]/4;
x=cos(w0*n);
y=filter(h1,1,x1);
stem(y);
```

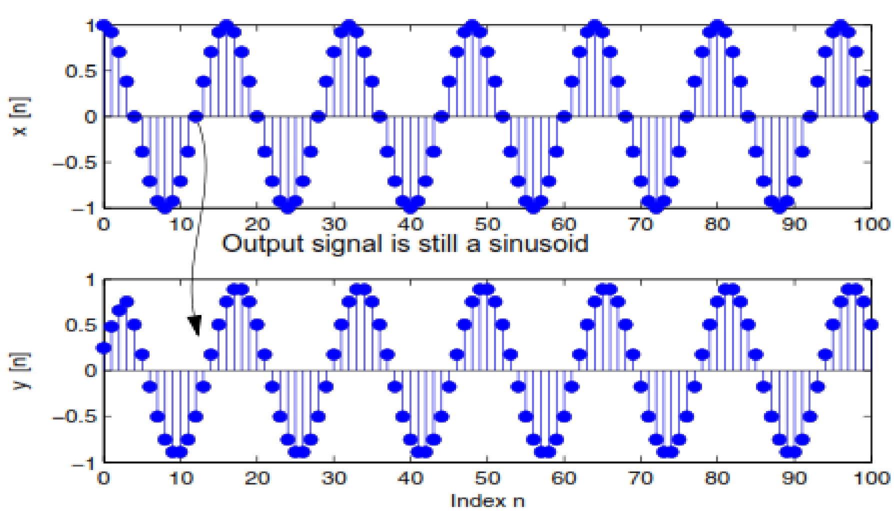


Figure 3 : Output signal y[n] after applying 4-tap moving average filter on input signal x[n]

Notes:

The output signal at $w_o = \pi/8$ is still a sinusoid, but the amplitude is reduced and there is small amount of phase shift.

- **Finding Z-Transform and Plotting poles and zeros**

Now, you will be able to find the Z-transform for a given input signal **x[n]** and impulse response **h[n]** in order to find the output **y[n]** by inversing the **Y(Z)** to discrete time domain using built-in function. In addition, you will study the properties of the LTI system by plotting poles and zeros for **H(Z).**

Consider the impulse response **h[n]** and the input signal **x[n]** as follows:

$$h[n] = n * u(n) \; and \; x[n] = \left(\frac{1}{2}\right)^n$$

Use the following Matlab code to find the output function **Y(Z)= H(Z)*X(Z)** and then find **y[n]** by inversing **Y(Z)**.

```
syms n z;
h=n*heaviside(n);
x=(0.5)^n;
Xz= ztrans(x,n,z)
Hz= ztrans(h,n,z)
Yz=Hz*Xz
y=iztrans(Yz,z,n)

Xz =
z/(z - 1/2)

Hz =
z/(z^2 - 2*z + 1)

Yz =
z^2/((z - 1/2)*(z^2 - 2*z + 1))

y =
2*n + 2*(1/2)^n - 2
```

From the above code, the Z-Transform has been calculated using **ztrans** function for **x[n]** and **h[n]**. After that , the output was calculated by multiply **H(Z)** with **X(Z)** and then **Y(Z)** was inversed to y[n] using **iztrans** function. Figure 4 shows the output **y[n],** which was plotted by using **ezplot** function. But, the output signal isn't in discrete time because the defined symbol cannot be stemmed.

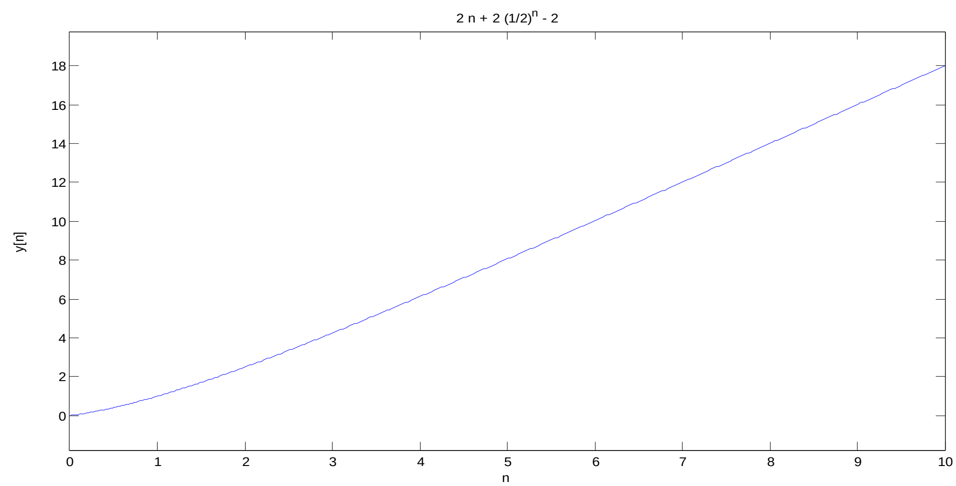


Figure 4: Output function Y[n] for continue value of n

Now, you will study the LTI system properties by plotting zeros and poles for the rational function H(Z) and specify for which values of Z that satisfy the region of convergence conditions , and then determine the casualty and stability of the system.

Plot the zeros and poles of the system using **zplane** function. The output should be as Figure 5 for the calculated H(Z)= $\frac{Z}{Z^2-2Z+1}$.

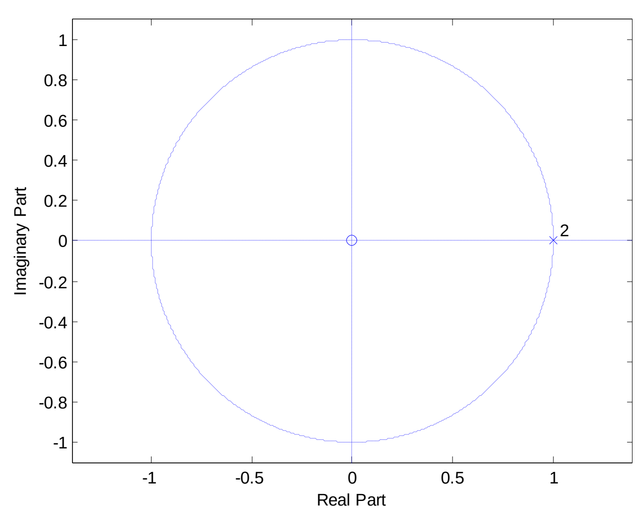


Figure 5: Zeros and Poles of the system H(Z)

From the plotted poles and zeros in Z-Plane , answer the following questions by justifying your answers:

1. Determine the region of convergence for H(Z).
2. Is the system causal or anti-casual ?
3. Is the system memoryless or not ?
4. Is the system BIBO stable or not ?

TODO:

Consider the impulse response is **h[n] =** $\boldsymbol{\alpha^n u[n]}$ **,** and the input signal **x[n]=** $\boldsymbol{n.u[n]}$. Plot the zeros and poles for three different values of $\alpha = 0.9\ ,1\ ,1.1$. Then, plot the output **y[n]** for the mentioned values of α. Determine the values of α that make the system BIBO stable overtime, justify your answers.