Digital Signal Processing Laboratory Report - Lab2

Haodong Zhang, 12113010

***Abstract*—A discrete-time system is characterized by its ability to take a discrete-time signal as input and generate a discrete-time signal as output. In this experimental study, we delve into the intricate world of discrete-time systems while examining their fundamental properties. Typically, discrete-time systems are represented in two primary forms: difference equations and block diagrams. Throughout this report, we employ these two tools to express the essence of discrete-time systems. Our investigation centers on the formulation of discrete-time systems as difference equations, enabling us to transform certain equations from the realm of continuous-time systems into discrete-time counterparts. We provide numerous illustrative examples to test and scrutinize the behavior of these discrete-time systems. Additionally, we explore operations between systems, including cascading and parallel configurations, and delve into the concept of inverse systems that correspond to discrete-time systems. To gain deeper insights into system properties, particularly linearity and time-invariance, we employ "black-box" systems (utilizing Matlab files bbox4.p, bbox5.p, and bbox6.p) to assess whether these systems meet these critical criteria. Furthermore, this report extends its exploration to real-life scenarios. We first employ an audio filter, subjecting an audio signal to two discrete-time systems to perceptually assess the impact of these systems on the sound by analyzing changes before and after processing. This auditory exploration offers an intuitive understanding of their effects. Subsequently, we investigate the application of filter functions in the context of the stock market, evaluating their strengths and weaknesses. All experiments are conducted using Matlab, and the accompanying report provides the corresponding code implementations and their outcomes.**

***Index Terms*—Discrete-Time Systems, Difference Equation, Matlab**

I. Introduction

THIS report is focused on the study of discrete-time systems, which are systems that take a discrete-time signal as input and produce a discrete-time signal as output. Discrete-time systems can generally be described in two primary ways: through a difference equation or using a block diagram representation. Typically, we use "x[n]" to denote the input signal at index "n" and "y[n]" to denote the output signal at index "n" making the difference equation a combination of these two signals. On the other hand, the block diagram representation portrays the discrete-time system as a flowchart composed of various symbols. It typically features a box labeled "D" to signify a unit of delay, which shifts "n" to "n-1," a circular symbol denoted as "+" representing signal addition, and a triangular symbol for multiplication. Coefficient values for these multiplicative operations are often displayed adjacent to the symbols, and arrows connect them to form the flowchart. Mathematically, we use the notation "y = S(x)" to represent a discrete-time system "S" with input signal "x[n]" and output signal "y[n]". To gain a comprehensive understanding of discrete-time systems, it is crucial to first explore their fundamental properties, including linearity, time-invariance, stability, and others. A system is considered linear if it exhibits the properties of additivity and scaling (homogeneity). Additionally, a system is deemed time-invariant if its behavior remains consistent regardless of the choice of "n=0", ensuring that identical experiments yield the same outcomes, irrespective of the starting time. Systems meeting both the linearity and time-invariance criteria are collectively referred to as Linear Time-Invariant (LTI) systems. This property holds significant importance, as LTI systems simplify analysis and possess a plethora of advantageous characteristics.

This is a lab report of the Digital Signal Processing (EE323) in the Southern University of Science and Technology (SUSTech).

Haodong Zhang is a undergraduate student majoring in communication engineering, Department of Electronic and Electrical Engineering, Southern University of Science and Technology (SUSTech), Shenzhen 518055, China (email: 12113010@mail.sustech.edu.cn).

II. Background Exercises

A. Example Discrete-time Systems

Firstly, we can use some examples to have a basic perceive to the Discrete-time digital systems and express it with difference equation and block diagram.

Discrete-time digital systems are often used in place of analog processing systems. With the development of computer and digital technology, many signals have become discrete and digital, and many corresponding devices have become discrete-time digital systems, such as digital cameras, computers, etc. Therefore, traditional continuous-time systems can be transformed into discrete-time digital systems for analysis. These digital systems can provide higher quality and lower cost through the use of standardized, high-volume digital processors.

For example, the following two continuous-time systems are commonly used in electrical engineering:

$$differentiator : y(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t) \tag{1}$$

$$integrator : y(t) = \int_{-\infty}^{t} x(\tau)d\tau \tag{2}$$

These systems can be replaced by the discrete-time digital systems form to approximate them. The concrete way of realization is to replace the continuous-form signal $y(t)$ and $x(t)$ with the discrete-form signal $y[n]$, $x[n]$, and then $x[n] - x[n-1]$ and $\sum\limits_{k=-\infty}^{n} x[k]$ substitute for the derivation operator $\frac{\mathrm{d}}{\mathrm{d}t}x(t)$ and integration operator $\int_{-\infty}^{t} x(\tau)d\tau$ respectively. Therefore, the difference equation can be written as below:

$$differentiator : y[n] = x[n] - x[n-1] \tag{3}$$

$$integrator : y[n] = \sum_{k=-\infty}^{n} x[k] \tag{4}$$

The equation (4) can be changed to the closed form as below, i.e. no summations in equation:

$$integrator : y[n] - y[n-1] = x[n] \tag{5}$$

These discrete-time systems can also be represented as a block diagram below:

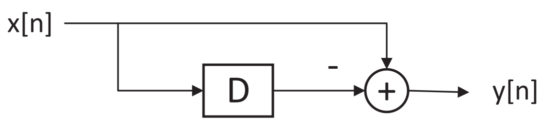

Fig. 1: block diagram of the discrete-time system for (3)

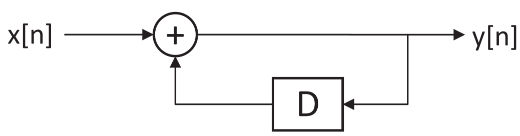

Fig. 2: block diagram of the discrete-time system for (5)

B. Stock Market Example

After the above example, we already know how to use discrete time systems to represent continuous systems, and know how to use difference equations and block diagrams to represent a discrete time system. In terms of practical application, we can take the stock market as an example to understand preliminarily the representation and application of discrete-time systems in this aspect.

The digital signal processing (DSP) techniques are so powerful that it can be used for very different kinds of signals. As long as the signal can be written as a discrete sequence and a reasonable discrete expression can be constructed to represent the system and fed into a computer, the signal can be processed. Therefore DSP has been used in a very wide range of applications. For example, it can be used in the stock market analysis. If a stockbroker wants to see whether the average value of a certain stock is increasing or decreasing, he must use some methods to calculate the average value of each day. There are three common and possible methods as below:

$$\begin{aligned}1 : avgvalue[today] =& \frac{1}{3}(value[today] + value[yesterday] \\ &+ value[2daysago])\end{aligned}$$

$$\begin{aligned}2 : avgvalue[today] =& 0.8 * avgvalue[yesterday] \\ &+ 0.2 * value[today]\end{aligned}$$

$$\begin{aligned}3 : avgvalue[today] =& avgvalue[yesterday] \\ &+ \frac{1}{3}(value[today] - value[3daysago])\end{aligned}$$

For each methods, we can express it by difference equation as follows:

$$1 : y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \tag{6a}$$

$$2 : y[n] = 0.8y[n-1] + 0.2x[n] \tag{6b}$$

$$3 : y[n] = y[n-1] + \frac{1}{3}(x[n] - x[n-3]) \tag{6c}$$

These equations also can be expressed by the system diagram as follows:

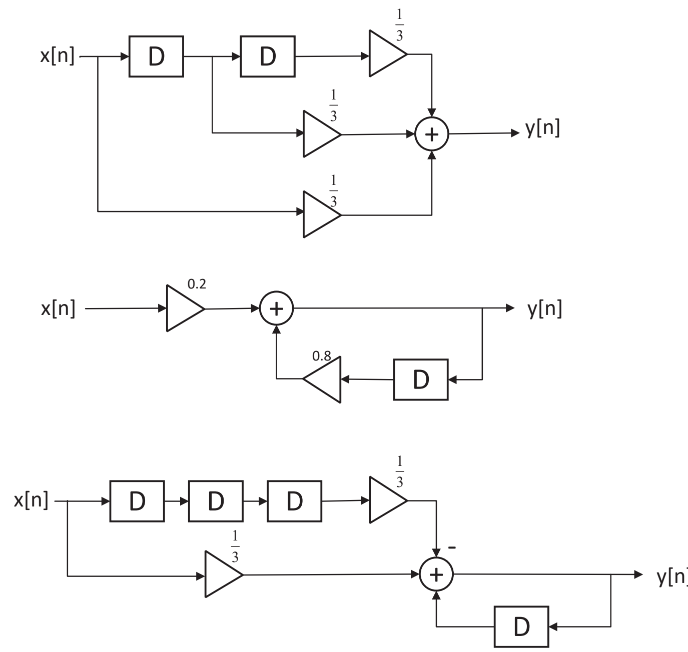

Fig. 3: block diagram for (6a), (6b) and (6c)

We can also calculate the impulse response of each system by letting $x[n] = \delta[n]$ as input. For (6a), the impulse response is $h[n] = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2])$ that is equal to:

$$h[n] = \{\ldots, 0, 0, \underset{\uparrow}{\frac{1}{3}}, \frac{1}{3}, \frac{1}{3}, 0, 0, \ldots\}$$

For (6b), the impulse response is $h[n] = 0.8 * h[n-1] + 0.2 * \delta[n] = \sum_{k=0}^{\infty} 0.2 * 0.8^k \delta[n-k]$. For (6c), the impulse response is $h[n] = h[n-1] + \frac{1}{3}(\delta[n] - \delta[n-3]) = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2])$ that is equal to:

$$h[n] = \{\ldots, 0, 0, \underset{\uparrow}{\frac{1}{3}}, \frac{1}{3}, \frac{1}{3}, 0, 0, \ldots\}$$

It is worth noting that the (6a) and (6c) are known as moving averages. That is because that the value of their current moment is related to the current and previous average, which is the meaning of "average", and as the current moment moves, the average moment will also move, which is the meaning of "moving". For (6a), the value of the current moment is the average of the input of the current moment and the previous two moments. For (6c), the increment of the current time relative to the previous time is the input of the current time minus the input of three moments before, and then is weighted, which also has an average meaning in it.

III. Example Discrete-Time Systems

After the background exercises, we can further analyze the discrete-time systems by using codes and figures. We can write two Matlab functions to apply the differentiator and integrator systems, designed in (3) and (5), to arbitrary input signals. The codes of two Matlab functions are as follows:

$differentiator.m$:

```
function output = differentiator(input,n)
l = length(n);
output = zeros(1,l+1);
output(1) = input(1);
for i = 1:l-1
    output(i+1) = input(i+1)-input(i);
end
output(l+1) = -input(l);
```

$integrator.m$:

```
function output = integrator(input,n)
l = length(n);
output = zeros(1,l);
sum = 0;
for i = 1:l
    sum = input(i)+sum;
    output(i) = sum;
end
```

In the above code, we think that the rest of the values of the output signal of the input signal outside the given range of n should be 0.

Then we can apply the functions above to the following two signals for -10 $\leq$ n $\leq$ 20.

$$\delta[n] - \delta[n-5] \tag{7a}$$

$$u[n] - u[n-(N+1)] \;\; with\; N = 10 \tag{7b}$$

Then the results of the input and output are shown in Fig. 4 and Fig. 5.

The corresponding Matlab codes are as follows:

```
n = -10:20;
n2 = -10:21;
x1 = ((n-0)==0) - ((n-5)==0);
x2 = (n>=0) - (n>=11);
d1 = differentiator(x1,n);
d2 = differentiator(x2,n);
i1 = integrator(x1,n);
i2 = integrator(x2,n);
figure;
subplot(3,1,1);
stem(n,x1);
xlabel('n');
title('input:\delta[n] - \delta[n-5]');
subplot(3,1,2);
stem(n2,d1);
xlim([-10,21]);
xlabel('n');
title('output of differentiator');
subplot(3,1,3);
stem(n,i1);
xlabel('n');
title('output of integrator');
figure;
subplot(3,1,1);
stem(n,x2);
xlabel('n');
title('input:u[n] - u[n-(11)]');
subplot(3,1,2);
stem(n2,d2);
xlim([-10,21]);
xlabel('n');
title('output of differentiator');
subplot(3,1,3);
stem(n,i2);
xlabel('n');
title('output of integrator');
```

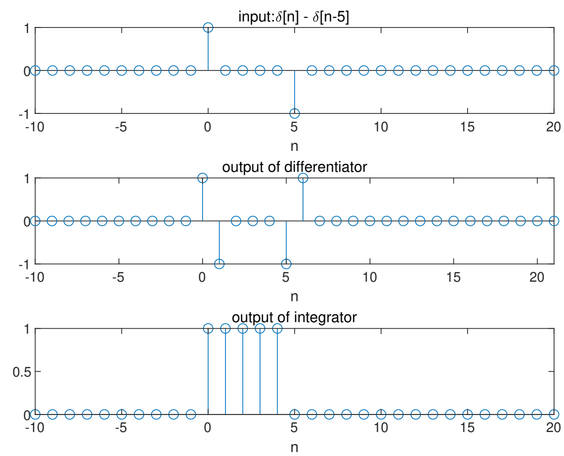

Fig. 4: the input and output of signal (7a)

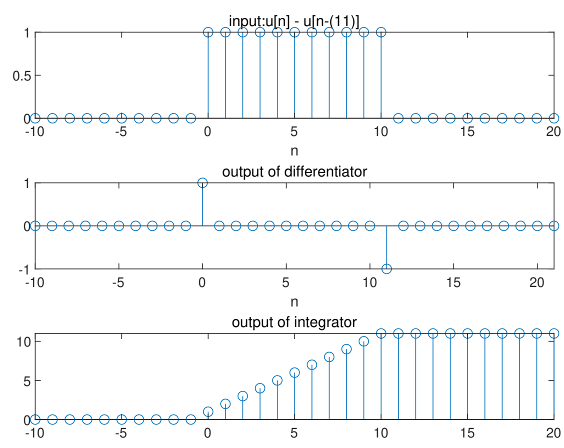

Fig. 5: the input and output of signal (7b)

Then we can analyze the stability of these systems. A stable system means that small input leads to response that does not diverge. Commonly, we also use the BIBO stable system which means that if the input to a stable system is bounded, the output must also be bounded to represent the stability of the system. For LTI system, we can use the impulse response h[n] to represent the stability, i.e. if $\sum_{k=-\infty}^{\infty}|h[k]| < \infty$ the system is stable. For differentiator system, $h[n] = \delta[n] - \delta[n-1]$, so $\sum_{k=-\infty}^{\infty}|h[k]| = \sum_{k=-\infty}^{\infty}(\delta[k] - \delta[k-1]) = 2 < \infty$.Thus, this system is stable system. For integrator system, h[n]=u[n], so $\sum_{k=-\infty}^{\infty}|h[k]| = \sum_{k=-\infty}^{\infty}u[k] = \infty$. Thus, this system is not stable system. Therefore, differentiator system is stable system, integrator system is non-stable system.

IV. Difference Equations

In this section, we can use Matlab function to express the difference equation and further analyze the operation between the systems. For example, there are two system expressed by difference equation:

$$S1 : y[n] = x[n] - x[n-1] \tag{8a}$$

$$S2 : y[n] = \frac{1}{2}y[n-1] + x[n] \tag{8b}$$

Then we can write Matlab functions to express them: $filter\_1.m$ for S1:

```
function output = filter_1(input,n)
l = length(n);
output = zeros(1,l+1);
output(1) = input(1);
for i = 1:l-1
    output(i+1) = input(i+1)-input(i);
end
output(l+1) = -input(l);
```

$filter\_2.m$ for S2:

```
function output = filter_2(input,n)
l = length(n);
output = zeros(1,l);
sum = 0;
for i = 1:l
    sum = input(i)+0.5*sum;
    output(i) = sum;
end
```

For analysis of the operation among the systems, we can use these functions to calculate and analyze the each of following 5 systems: S1, S2, S1(S2) (i.e., the series connection with S1 following S2), S2(S1) (i.e., the series connection with S2 following S1) and S1+S2. Firstly, we can draw the system diagram as follows:

Then, we can calculate the impulse response of each of them, the results are shown in Fig. 7.

From the results, we can find that the impulse response of S1(S2) is equal to the response of S2(S1) which means that for LTI systems, the sequence of passing through two systems is interchangeable, that is, the effect of passing through system 1 and then system 2 is the same as that of passing through system 2 and then system 1. This is because mathematically, passing through a system is convolution with the impulse response of the system, and the convolution operation satisfies the commutative law. So the order of the systems is interchangeable. And we also can find that the impulse response of S1+S2 is equal to the simple addition between the impulse response of S1 and the impulse response of S2. This is because mathematically, convolution operations satisfy distributive law.

The corresponding Matlab codes are as follows:

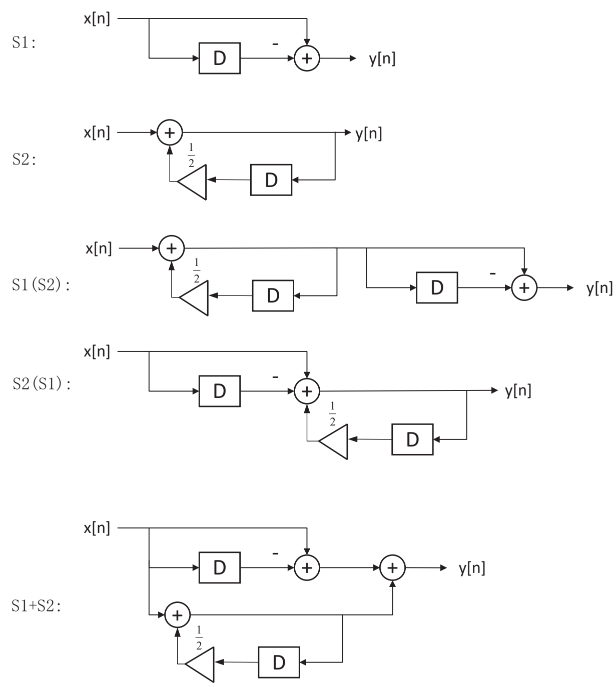

Fig. 6: the block diagram of 5 systems

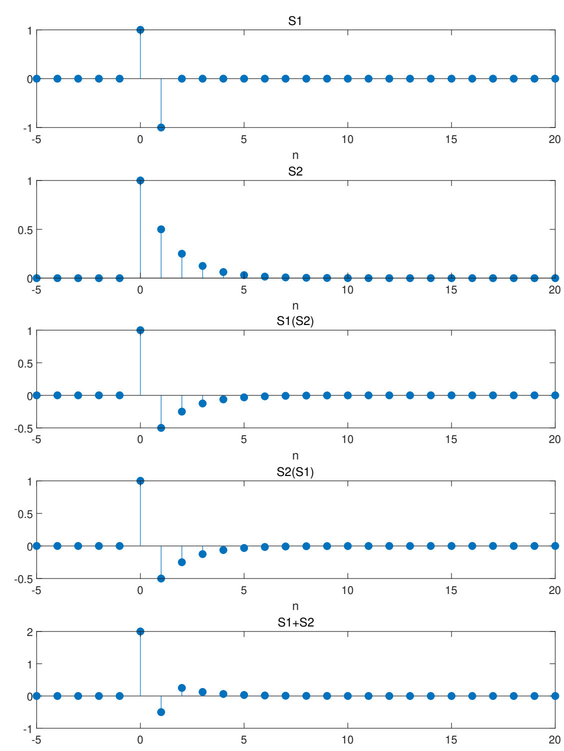

Fig. 7: the impulse response of 5 systems

```
figure;
n = -5:20;
delta = ((n-0)==0);
y1 = filter_1(delta,n);
y1 = y1(1:length(n));
y2 = filter_2(delta,n);
y3 = filter_1(y2,n);
y3 = y3(1:length(n));
y4 = filter_2(y1,n);
y5 = y1+y2;
subplot(5,1,1);
stem(n,y1,'filled');
xlabel('n');
title('S1');
subplot(5,1,2);
stem(n,y2,'filled');
xlabel('n');
title('S2');
subplot(5,1,3);
stem(n,y3,'filled');
xlabel('n');
title('S1(S2)');
subplot(5,1,4);
stem(n,y4,'filled');
xlabel('n');
title('S2(S1)');
subplot(5,1,5);
stem(n,y5,'filled');
xlabel('n');
title('S1+S2');
```

V. Audio Filtering

After having a basic understanding about the discrete-time systems and difference equation expression. In this section, we can use a practical case to visually and intuitively feel the effect of the discrete system on the signal. We can write some Matlab codes to load a music and listen to the signal, and then filter the audio signal with each of the two systems S1 and S2, the load and sound codes are as follows:

```
y = audioread("music.au")
s1 = filter_1(y,y);
s2 = filter_2(y,y);
sound(y);
sound(s1);
sound(s2);
```

Then we can listen the original signal and the filtered signal after each system and compare these signals. We can find that compared with the original signal, the music signal after S1 has become lighter, less a lot of musical elements, especially the background music of the music signal, as well as a large part of the sound bottom noise has been reduced or cleared, a more obvious feature is that compared with the original music signal, the drum sound in the signal through S1 has been greatly reduced, and only the weaker drum sound can be heard. This may be because S1 is a differential system, the common part of the signal of the previous time and the current time is eliminated, leaving only the changed part, so the less changed part such as drums, noise, background sound are eliminated or weakened, and the changed part is still left, so the overall sound and melody are not changed. For S2, the music signal is much thicker than the original signal, the sound volume is much louder than before, and more elements of the music can be heard, and even the previous echoes can be heard. Take a more obvious feature to analyze - drum sound, it can be found that the sound of the drum is louder than the sound of the original signal, and the sound of the background music and the background noise is also increased. This may be because the system is an accumulation system, adding the elements of the previous moment to the current moment, so the sound heard in the current moment contains the echoes of the previous, such as the drumbeat of musical elements that do not change much, after continuous accumulation, it is obvious that the sound has become louder. However, the previous sound does not completely drown out the current sound, because the cumulative amount of the previous time in the system function is multiplied by a weight of 0.5 to reduce the influence of the previous sound, so we can still hear the overall melody has not changed much.

VI. Inverse Systems

In this section, we want to find a LTI system S3: $y = S_3(x)$ such that $x = S_3(S_2(x))$ for any discrete-time signal x. We can say that the system S3 and S2 are inverse system because they cancel out the effects of each other. In fact the system S3 can be described by the difference equation: $y[n] = ax[n] + bx[n-1]$ where a and b are constants. In order to obtain the constants a and b, we can write the expression $y_{output} = S_3(S_2(x)) = ay_2[n] + by_2[n-1] = \frac{1}{2}ay_2[n-1] + ax[n] + by_2[n-1] = x[n]$ so $\frac{1}{2}a + b = 0$ and $a = 1$, thus, $a = 1, b = -\frac{1}{2}$. Therefore the expression of S3 is as follows:

$$S3 : y[n] = x[n] - \frac{1}{2}x[n-1] \tag{9}$$

Then we can use the system diagram to express it:

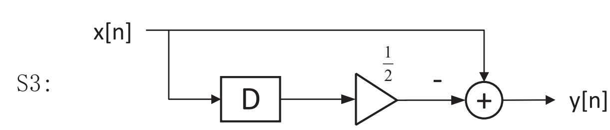

Fig. 8: the block diagram for system S3

Then we can write Matlab function to express S3: $filter\_3.m$ for S3:

```
function output = filter_3(input,n)
l = length(n);
output = zeros(1,l+1);
output(1) = input(1);
for i = 1:l-1
    output(i+1) = input(i+1)-0.5*input(i);
end
output(l+1) = -0.5*input(l);
```

Next we can obtain the impulse response of both S3 and S3(S2). The results are shown in Fig. 9.

The Matlab codes are as follows:

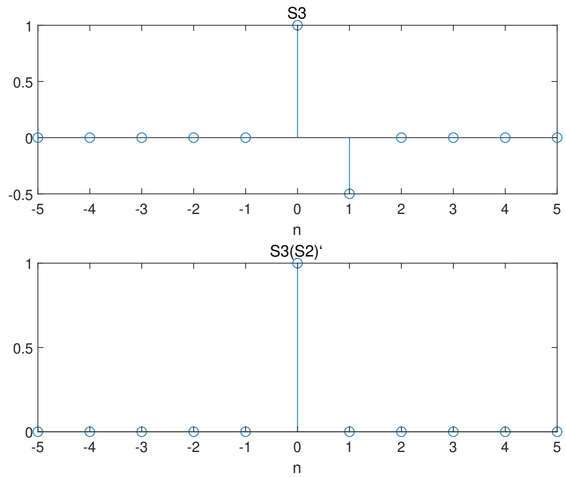

Fig. 9: the impulse response of both S3 and S3(S2)

```
a = 1;
b = -0.5;
n = -5:5;
delta = ((n-0)==0);
y3 = filter_3(delta,n);
y3 = y3(1:length(n));
y2 = filter_2(delta,n);
y32 = filter_3(y2,n);
y32 = y32(1:length(n));
figure;
subplot(2,1,1);
stem(n,y3);
xlabel('n');
title('S3');
subplot(2,1,2);
stem(n,y32);
xlabel('n');
title('S3(S2)');
```

VII. System Tests

Linear and time-invariant are important properties of systems. If a system is linear or time-invariant, it will greatly convenient for us to analyze the system and calculate the output signal. Therefore, it is necessary to determine if a system is linear and/or time-invariant. In this section, we will try to realize this procedure by inputting several different signals to test whether several systems satisfy the properties of linear and time-invariant correspondence. In this experiment, the systems are several black box systems in the files bbox4.p, bbox5.p, bbox6.p. The principle of the test is simple, as long as we try to input multiple signals until the output does not meet the relationship of the corresponding property which can be explained that the system does not meet the property, on the contrary, although it is not sure that it definitely meets the property, but it can be stated that meets the property with large probability.

To test these systems, we firstly designed the following testing input signals:

$$x1[n] = sin(\pi n)$$
$$x2[n] = 0.5^n$$
$$x3[n] = random\ signal$$

The Matlab codes are as follows:

```
n = -10:40;
s10 = sin(pi*n);
s1 = [zeros(1,5),s10,zeros(1,5)];
s20 = 0.5.^n;
s2 = [zeros(1,5),s20,zeros(1,5)];
s30 = randn(1,51);
s3 = [zeros(1,5),s30,zeros(1,5)];
```

In the above code, we add 5 zeros to the front and back of the original signal, so that we can later shift the signal left and right to explore whether the system is time invariant.

Now, we can test and explore the property of the systems. For linearity, we firstly calculate the $bbox(a*x1[n]+b*x2[n])$ and then calculate the $a*bbox(x1[n])+b*bbox(x2[n])$, if they are equal, then it is linear and vice versa. a and b arbitrary constant, we can set as 1 and 2 here. For time-invariant, let $y[n] = bbox(x[n])$, if $y[n-n0]$ is equal to $bbox(x[n-n0])$, it is time-invariant and vice versa.

For bbox4, we can firstly test the linearity, the codes are as follows:

```
y1 = bbox4(s1);
y2 = bbox4(s2);
y3 = bbox4(s3);
z41 = bbox4(s1+2*s3);
z42 = y1+2*y3;
figure;
plot(z41-z42);
```

The difference value are shown in Fig. 10.

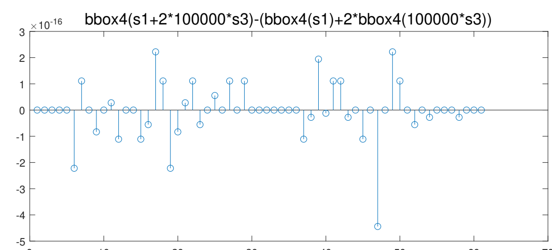

Fig. 10: the difference value between bbox4(s1+2*s3) and bbox4(s1)+2*bbox4(s3)

It can be seen from the figure that the difference is very small with the order of 10 to the -16 power, which is probably caused by the accuracy problem. In order to verify whether it is a precision problem, we can increase the input signal s3 by 100,000 times, and then calculate the difference, as shown in Fig. 11.

Therefore, bbox4 is a linear system. Then we test if bbox4 is time-invariant. It is possible to draw the output diagram of the original signal, the output diagram after the original signal translation, the diagram after the translation of the above two signals and the difference diagram after the alignment of the two signals. The codes are as follows:

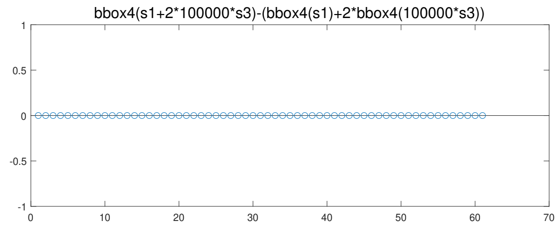

Fig. 11: the difference value between bbox4(s1+2*100000*s3) and bbox4(s1)+2*bbox4(100000*s3)

```
n2 = -15:45;
y3 = bbox4(s3);
s3_sf = [s30, zeros(1,10)];
z41_sf = bbox4(s3_sf);
z42_sf = y3(6:length(y2));
figure;
subplot(4,1,1)
stem(n2,y3,'filled');
title('bbox4(s3[n])')
subplot(4,1,2)
stem(n2,z41_sf,'filled');
title('bbox4(s3[n+5])')
subplot(4,1,3)
stem(z41_sf(1:length(y1)-5),'filled');
hold on
stem(z42_sf,'filled')
title('aligned two signal')
subplot(4,1,4)
stem(z41_sf(1:length(y1)-5)-z42_sf,'filled'
    );
title('difference value')
```

The results are shown in Fig. 12. As can be seen from this figure, the signals after alignment will be different, so the system does not meet the property of time invariance. Therefore, bbox4 is linear and time-varying.

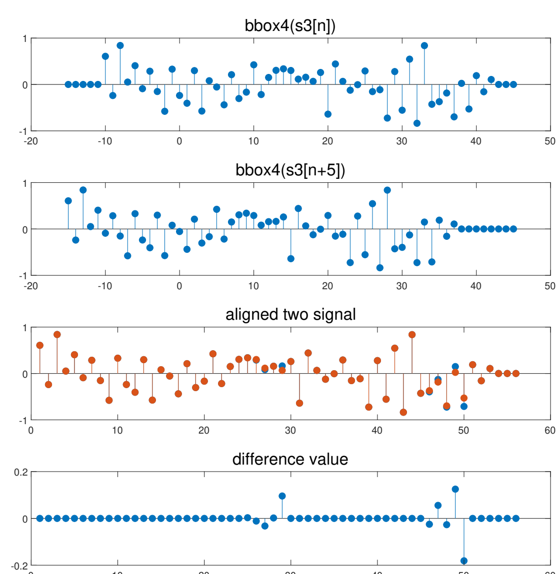

Fig. 12: testing results of bbox4

Similarly, we can also use the signals as input to analyze the linearity and time-invariance for bbox5 and bbox6, and plot the similar figures as that in analysis of bbox4. For bbox5, the testing results are shown in Fig. 13.

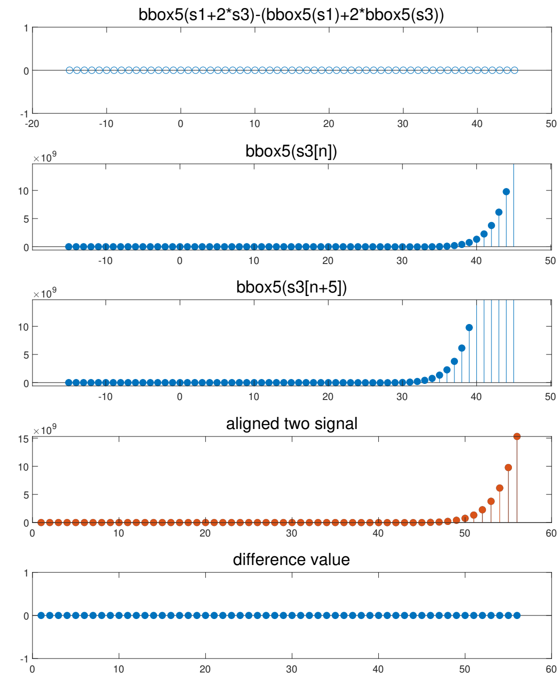

Fig. 13: testing results of bbox5

Therefore, bbox5 is linear and time-invariant. For bbox6, the results are shown in Fig. 14.

Therefore, bbox6 is non-linear and time-invariant.

Their codes are similar, we can take the codes of testing for bbox5 as example to show here:

```
figure;
y1 = bbox5(s1);
y2 = bbox5(s2);
y3 = bbox5(s3);
y3_scale = bbox5(100000*s3);
z41 = bbox5(s1+2*100000*s3);
z42 = y1+2*y3_scale;
subplot(5,1,1)
stem(n2,z41-z42);
title('bbox5(s1+2*s3)-(bbox5(s1)+2*bbox5(s3
    ))','FontSize',15);
s3_sf = [s30,zeros(1,10)];
z41_sf = bbox5(s3_sf);
z42_sf = y3(6:length(y2));
n2 = -15:45;
subplot(5,1,2)
stem(n2,y3,'filled');
ylim([0 15.3*10^9]);
title('bbox5(s3[n])','FontSize',15)
subplot(5,1,3)
stem(n2,z41_sf,'filled');
ylim([0 15.3*10^9]);
title('bbox5(s3[n+5])','FontSize',15)
subplot(5,1,4)
```

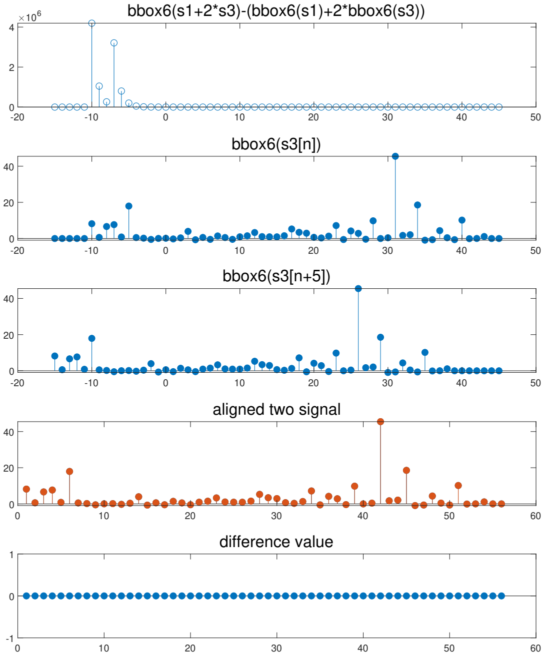

Fig. 14: testing results of bbox6

```
stem(z41_sf(1:length(y1)-5),'filled');
hold on
stem(z42_sf,'filled')
title('aligned two signal','FontSize',15)
subplot(5,1,5)
stem(z41_sf(1:length(y1)-5)-z42_sf,'filled'
    );
title('difference value','FontSize',15)
```

VIII. Stock Market Example

In this section, we will analyze the daily stock market exchange rates for a publicly traded stock. We will apply the filters (6b) and (6c) to smooth the stock values. Firstly, we should write Matlab functions to express the filter (6b) and (6c):

$filter\_4.m$ for (6b)

```
function output = filter_4(input,n)
l = length(n);
output = zeros(1,l);
sum = 0;
for i = 1:l
    sum = 0.2*input(i)+0.8*sum;
    output(i) = sum;
end
```

$filter\_5.m$ for (6c)

```
function output = filter_5(input,n)
l = length(n);
output = zeros(1,l);
sum = 0;
for i = 1:l
    if i<4
        sum = 1/3*(input(i))+sum;
    else
        sum = 1/3*(input(i)-input(i-3))+sum
            ;
    end
    output(i) = sum;
end
```

The corresponding Matlab codes are as follows:

```
load stockrates.mat;
y4 = filter_4(rate,rate);
y5 = filter_5(rate,rate);
figure;
subplot(3,1,1);
plot(rate);
title('the original stock values');
subplot(3,1,2);
plot(y4);
title('the values after filter 6(b)');
subplot(3,1,3);
plot(y5);
title('the values after filter 6(c)');
```

In these codes, we use an initial value of 0, and set the initial values of the rate vector to 0 (for the days prior to the start of data collection).

Then the original stock value and the result after each filters are shown in Fig. 15.

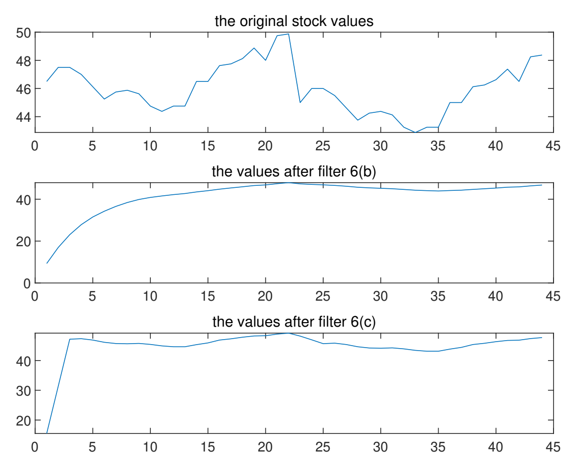

Fig. 15: the original stock value and the result after each filters

In order to see the relationship between several curves more intuitively and more clearly, we can draw the above curves into a graph for analysis as shown in Fig. 16.

According to the results obtained above, we can analyze the advantage and disadvantages of the two filters. For the filter (6b), the advantages are that its curve is very smooth, which eliminates the fluctuation of the original value well, and its response speed to the current value is small, that is, it will not change greatly with the drastic fluctuation of the current value, and the overall change is relatively stable. This is because the proportion of yesterday accounted for 0.8, which is much higher than today's proportion, so it is relatively less affected by the current value. Therefore, for the stock market,

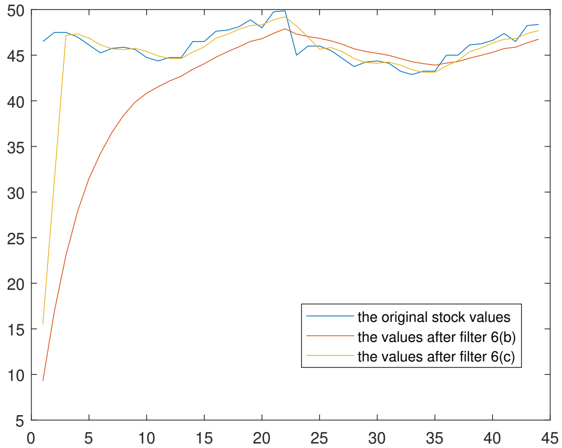

Fig. 16: the original stock value and the result after each filters

it will not be too much affected by today's stock, the overall trend is conservative, the reliability of the estimate is high, and the judgment of the trend will not be affected by the drastic changes of the stock in an individual day. However, it also has some disadvantages. Firstly, in the front part of the curve, it follows the original data slowly. As can be seen from the figure, the filtered estimated curve follows the changes of the original data better after more than ten days. In other words, it is greatly affected by the initial value, and if the initial value is not selected well, it will take a long time to eliminate this effect. Secondly, it shows a certain degree of hysteresis, that is, it will not change in time with the initial value change, there will be a certain hysteresis, in other words, when the original value has begun to rise, the filtered curve will not rise immediately, but after a few days there is an upward trend. Moreover, the amplitude of rise and fall is relatively small and gentle compared with the original value, which can not well reflect the degree of amplitude changing.

For the filter (6c), the advantages are that first of all, in the initial stage, it can quickly rise to keep up with the original data, as can be seen from the figure, only about 3 values, the two curves can be joined, and the following speed is very fast. Secondly, it follows the original data curve well and can well reflect the rise and fall trend of the original data. In addition, the response speed is also very fast, when the original data suddenly rises and falls, the filtered curve can quickly follow. However, there are some disadvantages in this filtered value. It can be seen that the filtered curve still has great fluctuations, and is not very smooth, because it is greatly affected by the current value, and the response speed is fast, so if the original data has great fluctuations, even when the singular value, the filter function will not smooth it off, or will be greatly affected by the singular data.

As can be seen from the above curve, the initial value has a great impact on the effect of the filter. As can be seen from the figure above, if the initial value is 0, the effect of the initial part is not good, and it takes a long time to return to normal. So we should choose a better way to initialize the output of the filter, a better way is to set the initial value directly to the true value, so that the initial part does not deviate too much.

IX. Conclusion

In this experiment, we deepened our understanding and cognition of discrete-time system. Through many examples, we learned how to use difference equation and block diagram to express a discrete time system. We also used examples from real life, including audio filtering and stock market, to further intuitively and deeply feel the role and impact of discrete system. In addition, we also analyzed the inverse system of discrete system. And we used Matlab to test whether several black boxes meet the linear and time-invariant properties, and concluded that bbox4 is time-varying and bbox6 is nonlinear. Through this experiment, my most profound experience is that difference equation plays a huge role in the description of discrete time systems. The use of difference equation can not only represent discrete systems well, but also facilitate the realization in computer systems and programming. Moreover, it also facilitates the calculation and inverse of operations between systems. And it can be well applied to practice for analysis. In addition, I have a better understanding of how to draw pictures to represent signals and the relationship between signals, and my ability of drawing and programming has improved. So far, through experiments, we have enough cognition and basic knowledge of discrete-time system, and then we can further analyze the discrete system in the following labs.

Acknowledgments

Thanks to the teachers and teaching assistants for their guidance. Thank the teacher for teaching us theoretical knowledge and providing the lab manual for us, which raised many meaningful questions for us and guided and inspired us to think deeply.