Digital Signal Processing

中華技術學院電子系/所

林盈灝

Reference: Digital Signal Processing Laboratory Using Matlab
Author: Sanjit K. Mitra

Chapter 2 Discrete-Time Systems in the Time-Domain

2.1 Introduction

A discrete-time system processes an input signal in the time-domain to generate an output signal with more desirable properties by applying an algorithm composed of simple operations on the input signal and its delayed versions.

2.2 Background Review

C1. For a **linear** discrete-time system, if $y_1[n]$ and $y_2[n]$ are the responses to the input sequences $x_1[n]$ and $x_2[n]$, respectively, then for an input

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{2.1}$$

the response is given by

$$y[n] = \alpha y_1[n] + \beta y_2[n] \tag{2.2}$$

The superposition property of Eq. (2.2) must hold for any arbitrary constant α and β and for all possible inputs $x_1[n]$ and $x_2[n]$.

C2. For a **time-invariant** discrete-time system, if $y_1[n]$ is the responses to an input $x_1[n]$, then the response to an input

$$x[n] = x_1[n - n_0]$$

is simply

$$y[n] = y_1[n - n_0]$$

where n_0 is any positive or negative integer.

C3. A **linear time-invariant (LTI)** discrete-time system satisfies both the linearity and the time-invariance properties.

C4. If $y_1[n]$ and $y_2[n]$ are the responses of a **causal** discrete-time system to the inputs $u_1[n]$ and $u_2[n]$, respectively, then

$$u_1[n] = u_2[n] \quad \text{for } n < N$$

implies also that

$$y_1[n] = y_2[n] \quad \text{for } n < N$$

C5. A discrete-time system is said to be bounded-input, bounded-output (BIBO) stable if, for any bounded input sequence $x[n]$, the corresponding output $y[n]$ is also a bounded sequence, that is, if

$$\|x[n]\| \le B_x < \infty \Rightarrow \|y[n]\| \le B_y < \infty \quad \text{for all values of } n$$

C6. The response of a digital filter to a **unit sample sequence** $\{\delta[n]\}$ is called the unit sample response or simply, the **impulse response**, and denoted as $\{h[n]\}$. Correspondingly, the response of a discrete-time system to a **unit step sequence** $\{u[n]\}$, denoted as $\{s[n]\}$, is its unit step response or, simply the **step response**.

C7. The response $y[n]$ of an LTI discrete-time system characterized by an impulse response $h[n]$ to an input signal $x[n]$ is given by

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{2.3}$$

which can be alternately

$$y[n] = \sum_{k=-\infty}^{\infty} h[n-k]x[k] \tag{2.4}$$

The sum in Eqs. (2.3) and (2.4) is called the convolution sum of the sequences $x[n]$ and $h[n]$, and is represented compactly as:

$$y[n] = h[n] * x[n] = x[n] * h[n] \tag{2.5}$$

C8. The overall impulse response by $h[n]$ of the LTI discrete-time system obtained by a cascade connection of two LTI discrete-time systems with impulse responses $h_1[n]$ and $h_2[n]$, respectively, and as shown in Figure 2.1, is given by

$$h[n] = h_1[n] * h_2[n] \tag{2.6}$$

If the two LTI systems in the cascade connection of Figure 2.1 are such that

$$h_1[n] * h_2[n] = \delta[n] \tag{2.7}$$

then the system $h_2[n]$ is said to be the **inverse** of the LTI system $h_1[n]$ and vice verse.

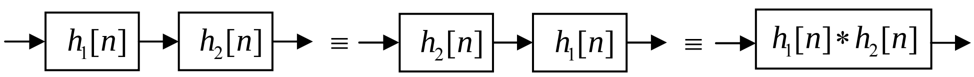

Figure 2.1 The cascade connection

C9. An LTI discrete-time system is **BIBO stable** if and only if its impulse response sequence $\{h[n]\}$ is absolutely summable, that is

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty \tag{2.8}$$

C10. An LTI discrete-time system is **causal** if and only if its impulse response sequence $\{h[n]\}$ satisfies the condition

$$h[k] = 0 \quad \text{for } k < 0 \tag{2.9}$$

C11. The class of LTI discrete-time systems with which we shall be mostly concerned in this lecture is characterized by a **linear constant coefficient difference equation** of the form

$$\sum_{k=0}^{N} d_k y[n-k] = \sum_{k=0}^{M} p_k x[n-k] \tag{2.10}$$

where $x[n]$ and $y[n]$ are, respectively, the input and the output of the system, and $\{d_k\}$ and $\{p_k\}$ are constants. The order of the discrete-time system is $\max(N, M)$, which is the order of the difference equation characterizing the system. If we assume the system to be causal, then we can rewrite Eq. (2.9) to express $y[n]$ explicitly as a function of $x[n]$:

$$y[n] = -\sum_{k=1}^{N} \frac{d_k}{d_0} y[n-k] + \sum_{k=0}^{M} \frac{p_k}{d_0} x[n-k] \tag{2.11}$$

provided $d_0 \neq 0$. The output $y[n]$ can be computed using Eq. (2.10) for all $n \geq n_0$ knowing $x[n]$ and the initial conditions $y[n_0-1], y[n_0-2], \cdots, y[n_0-N]$

C12. A discrete-time system is called a **finite impulse response (FIR)** system if its impulse response $h[n]$ is of finite length. Otherwise, it is an **infinite impulse response (IIR)** system. The causal system of Eq. (2.11) represents an FIR system if $d_k = 0$ for $k > 0$. Otherwise, it is IIR system.

2.3 Simulation of Discrete-time Systems

For the simulation of causal LTI discrete-time systems described by Eq. (2.10), the command **filter** can be used. If we denote

$$num = [p_0 \;\; p_1 \;\; \cdots \;\; p_M]$$
$$den = [d_0 \;\; d_1 \;\; \cdots \;\; d_N]$$

then y = filter(num, den, x) generates an output vector y of the same length as the specified input vector x with zero initial conditions i.e., $y[-1] = y[-2] = \cdots = y[-N] = 0$. The output can also be computed using y = filter(num, den, x, ic) where ic is the vector of initial conditions. i.e., ic $= [y[-1], y[-2], \cdots, y[-N]]$. Access to final conditions is obtained using [y, fc] = filter(num, den, x, ic)

Project 1: The Moving Average System

A causal M-point smoothing FIR filter is defined as

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \tag{2.12}$$

The system of Eq. (2.12) is also known as a **moving average filter**. We illustrate its use in filtering high-frequency components from a signal composed of a sum of several sinusoidal signals.

Program P2_1:

```
% Program P2_1
% Simulation of an M-point Moving Average Filter
% Generate the input signal
n = 0:100;
s1 = cos(2*pi*0.05*n); % A low-frequency sinusoid
s2 = cos(2*pi*0.47*n); % A high frequency sinusoid
x = s1+s2;
% Implementation of the moving average filter
M = input('Desired length of the filter = ');
num = ones(1,M);
```

```
y = filter(num,1,x)/M;
% Display the input and output signals
clf;
subplot(2,2,1);
plot(n, s1);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #1');
subplot(2,2,2);
plot(n, s2);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Signal #2');
subplot(2,2,3);
plot(n, x);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Input Signal');
subplot(2,2,4);
plot(n, y);
axis([0, 100, -2, 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Signal');
axis;
```

Questions:

Q1 Run the above program for M = 2 to generate the output signal with $x[n] = s_1[n] + s_2[n]$ as the input. Which component of the input $x[n]$ is suppressed by the discrete-time system simulated by this program?

Q2 If the linear time-invariant system is changed from $y[n] = 0.5(x[n] + x[n-1])$ to $y[n] = 0.5(x[n] - x[n-1])$, what is its effect on the input $x[n] = s_1[n] + s_2[n]$?

Q3 Run Program P2_1 for other values of filter length M, and various values of the frequencies of the sinusoidal signals $s_1[n]$ and $s_2[n]$. Comment on your results..

Project 2: A Simple Nonlinear Discrete-time System

Given the nonlinear system as follows:

$$y[n] = x[n]^2 - x[n-1]x[n+1] \tag{2.13}$$

The following program is used to generate the output signal $y[n]$ for different types of the input $x[n]$.

Program P2_2:

```
%Program P2_2
clf;
n = 0:200;
x = cos(2*pi*0.05*n);
% Compute the output signal
x1 = [x 0 0]; % x1[n] = x[n+1]
x2 = [0 x 0]; % x2[n] = x[n]
x3 = [0 0 x]; % x3[n] = x[n-1]
y = x2.*x2-x1.*x3;
y = y(2:202);
% Plot the input and output signals
subplot(2,1,1)
plot(n, x)
xlabel('Time index n');ylabel('Amplitude');
title('Input Signal')
subplot(2,1,2)
plot(n,y)
xlabel('Time index n');ylabel('Amplitude');
title('Output signal');
```

Questions:

Q4 Use sinusoidal signals with different frequencies as the input signals and compute the output signal for each input. How do the output signals depend on the frequencies of the input signals? Verify your observation mathematically.

Q5 Use sinusoidal signals of the form $x[n] = \sin(\omega_0 n) + k$ as the input signal and compute the output signal? How does the output signal $y[n]$ depend on the DC value k

Project 3: Linear and Nonlinear Systems

We now investigate the linearity property of a causal system of the type described by Eq. (2.10). Consider the system given by

$$y[n] - 0.4y[n-1] + 0.75y[n-2] = 2.2403x[n] + 2.4908x[n-1] + 2.2403x[n-2] \qquad (2.14)$$

Program P2_3: To generate three different input sequences $x_1[n]$, $x_2[n]$ and $x[n] = ax_1[n] + bx_2[n]$, and to compute and plot the corresponding output sequences $y_1[n]$, $y_2[n]$ and $y[n]$.

```
% Program P2_3
% Generate the input sequences
clf;
```

```
n = 0:40;
a = 2;b = -3;
x1 = cos(2*pi*0.1*n);
x2 = cos(2*pi*0.4*n);
x = a*x1 + b*x2;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; % Set zero initial conditions
y1 = filter(num,den,x1,ic); % Compute the output y1[n]
y2 = filter(num,den,x2,ic); % Compute the output y2[n]
y = filter(num,den,x,ic); % Compute the output y[n]
yt = a*y1 + b*y2;
d = y - yt; % Compute the difference output d[n]
% Plot the outputs and the difference signal
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
title('Output Due to Weighted Input: a \cdot x_{1}[n] + b \cdot x_{2}[n]');
subplot(3,1,2)
stem(n,yt);
ylabel('Amplitude');
title('Weighted Output: a \cdot y_{1}[n] + b \cdot y_{2}[n]');
subplot(3,1,3)
stem(n,d);
xlabel('Time index n');ylabel('Amplitude');
title('Difference Signal');
```

Questions:

Q6 Run Program P2_3 and compare $y[n]$ obtained with weighted input with $yt[n]$ obtained by combining the two outputs $y_1[n]$ and $y_2[n]$ with the same weights. Are these two sequences equal? Is this system linear?

Q7 Repeat Q6 with nonzero initial condition.

Q8 Consider another system described by:

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

Modify Program P2_3 to compute $y[n]$, $y_1[n]$ and $y_2[n]$ of the above system. Compare $y[n]$ with $yt[n]$. Are these two sequences equal? Is this system linear?

Project 4: Time-Invariant and Time-Varying Systems

Program P2_4 is used to simulate the system Eq. (2-14) to generate two different input sequences $x[n]$ and $x[n-d]$, and to compute and plot the corresponding output sequences $y_1[n]$ and $y_2[n]$, and the difference $y_1[n]-y_2[n+d]$.

Program P2_4:

```
% Program P2_4
% Generate the input sequences
clf;
n = 0:40; D = 10;a = 3.0;b = -2;
x = a*cos(2*pi*0.1*n) + b*cos(2*pi*0.4*n);
xd = [zeros(1,D) x];
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; % Set initial conditions
% Compute the output y[n]
y = filter(num,den,x,ic);
% Compute the output yd[n]
yd = filter(num,den,xd,ic);
% Compute the difference output d[n]
d = y - yd(1+D:41+D);
% Plot the outputs
subplot(3,1,1)
stem(n,y);
ylabel('Amplitude');
title('Output y[n]'); grid;
subplot(3,1,2)
stem(n,yd(1:41));
ylabel('Amplitude');
title(['Output due to Delayed Input x[n ‘, num2str(D),']']); grid;
subplot(3,1,3)
stem(n,d);
xlabel('Time index n'); ylabel('Amplitude');
title('Difference Signal'); grid;
```

Questions:

Q9 Run Program P2_4 and compare the output sequences $y[n]$ and $yd[n-10]$. What is the relation between these two sequences? Is this system time-invariant?

Q10 Repeat Q9 for nonzero initial conditions. Is this system time-invariant?

Q11 Consider another system described by:

$$y[n] = nx[n] + x[n-1] \qquad (2\text{-}15)$$

Modify Program P2_4 to simulate the above system and determine whether this system is time-invariant or not?

Q12 Modify Program P2_3 to test the linearity of the system of Eq. (2-15)?

Project 5: Computation of Impulse Responses of LTI Systems

The Matlab command $y = impz(num, den, N)$ can be used to compute the first N samples of the impulse response of the causal LTI discrete-time of Eq. (2-11)

Program 2_5

```
% Program P2_5
% Compute the impulse response y
clf;
N = 40;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
y = impz(num,den,N);
% Plot the impulse response
stem(y);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```

Questions:

Q13 Run Program P2_5 and generate the impulse response of the discrete-time system of Eq. (2-14).

Q14 Modify Program P2_5 to generate the first 45 samples of the impulse reponse of the following causal LTI system

$$\begin{aligned} & y[n]+0.71y[n-1]-0.46y[n-2]-0.63y[n-3] \\ & =0.9x[n]-0.45x[n-1]+0.35x[n-2]+0.002x[n-3] \end{aligned} \quad (2.16)$$

Project 6: Cascade of LTI Systems

In practice a causal LTI discrete-time system of higher order is implemented as a cascade of lower order causal LTI discrete-time systems. For example, the fourth-order discrete-time system given below

$$\begin{aligned} & y[n]+1.6y[n-1]+2.28y[n-2]+1.325y[n-3]+0.68y[n-4] \\ & =0.06x[n]-0.19x[n-1]+0.27x[n-2]-0.26x[n-3]+0.12x[n-4] \end{aligned} \quad (2.17)$$

can be realized as a cascade of two second-order discrete-time systems:

Stage No. 1

$$y_1[n]+0.9y_1[n-1]+0.8y_1[n-2]=0.3x[n]-0.3x[n-1]+0.4x[n-2] \quad (2.18)$$

Stage No. 1

$$y_2[n]+0.7y_2[n-1]+0.85y_2[n-2]=0.2y_1[n]-0.5y_1[n-1]+0.3y_1[n-2] \quad (2.19)$$

Program 2_6 simulates the fourth-order system of Eq. (2.17), and cascade system of Eqs. (2.18) and (2.19). Its first generates a sequence $x[n]$, and then uses it as the input of the fourth-order system, generating the output $y[n]$. It then applies the same input $x[n]$ to Stage No. 1 and finds its output sequence $y_1[n]$. Next, it uses $y_1[n]$ as the input of Stage No. 2 and finds its output $y_2[n]$. Finally, the difference between the two overall outputs $y[n]$ and $y_2[n]$ are formed.

Program 2_6

```
% Program P2_6
% Cascade Realization
clf;
x = [1 zeros(1,40)]; % Generate the input
n = 0:40;
% Coefficients of 4th order system
den = [1 1.6 2.28 1.325 0.68];
num = [0.06 -0.19 0.27 -0.26 0.12];
% Compute the output of 4th order system
y = filter(num,den,x);
% Coefficients of the two 2nd order systems
num1 = [0.3 -0.2 0.4];den1 = [1 0.9 0.8];
num2 = [0.2 -0.5 0.3];den2 = [1 0.7 0.85];
% Output y1[n] of the first stage in the cascade
y1 = filter(num1,den1,x);
% Output y2[n] of the second stage in the cascade
y2 = filter(num2,den2,y1);
% Difference between y[n] and y2[n]
d = y - y2;
% Plot output and difference signals
subplot(3,1,1);
stem(n,y);
ylabel('Amplitude');
title('Output of 4th order Realization'); grid;
subplot(3,1,2);
stem(n,y2)
ylabel('Amplitude');
title('Output of Cascade Realization'); grid;
subplot(3,1,3);
stem(n,d)
xlabel('Time index n');ylabel('Amplitude');
```

```
title('Difference Signal'); grid;
```

Questions:

Q15 Run Program P2_6 to compute the output sequences $y[n]$ and $y2[n]$ and the difference signal $d[n]$. Is $y[n]$ the same as $y2[n]$?

Q16 Repeat Q15 with the input changed to a sinusoidal sequence.

Q17 Repeat Q15 with arbitrary nonzero initial condition vectors ic, ic1, and ic2.

Project 7: Convolution

The convolution operation is implemented in MATLAB by the command conv, provided the two sequences to be convolved are of finite length.

Program 2_7

```
% Program P2_7
clf;
h = [3 2 1 -2 1 0 -4 0 3];    % impulse response
x = [1 -2 3 -4 3 2 1];        % input sequence
y = conv(h,x);
n = 0:14;
subplot(2,1,1);
stem(n,y);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Obtained by Convolution'); grid;
x1 = [x zeros(1,8)];
y1 = filter(h,1,x1);
subplot(2,1,2);
stem(n,y1);
xlabel('Time index n'); ylabel('Amplitude');
title('Output Generated by Filtering'); grid;
```

Questions:

Q18 Run Program P2_7 to generate $y[n]$ obtained by the convolution of the sequences $h[n]$ and $x[n]$, and to generate $y1[n]$ obtained by the filtering the input $x[n]$ by the FIR filter $h[n]$. Is there any difference between $y[n]$ and $y1[n]$. What is the reason for using $x1[n]$ obtained by zero-padding $x[n]$ as the input for generating $y1[n]$.

Project 8: Stability of LTI Systems

An LTI discrete-time system is BIBO stable if its impulse response is absolutely summable. Program P2_8 is a simple program used to compute the sum of the absolute values of the impulse response samples of a causal IIR LTI system. Its compute N samples of the impulse response sequence, evaluates

$$S(K)=\sum_{n=0}^{K}|h[n]|$$

for increasing *K*, and checks the value of $|h[K]|$ at each iteration step. If the value of $|h[K]|$ is smaller than 10^{-6}, then it is assumed that the sum $S(K)$ has converged and is very close to $S(\infty)$.

Program 2_8

```
% Program P2_8
% Stability test based on the sum of the absolute
% values of the impulse response samples
clf;
num = [1 -0.8]; den = [1 1.5    0.9];
N = 200;
h = impz(num,den,N+1);
parsum = 0;
for k = 1:N+1;
parsum = parsum + abs(h(k));
if abs(h(k)) < 10^(-6), break, end
end
% Plot the impulse response
n = 0:N;
stem(n,h)
xlabel('Time index n'); ylabel('Amplitude');
% Print the value of abs(h(k))
disp('Value =');disp(abs(h(k)));
```

Questions:

Q19 What are the purposes of the commands for, end and break?

Q20 What is the discrete-time system whose impulse response is being determined by Program P2_8? Run Program P2_8 to generate the impulse response. Is this system stable? If $|h[K]|$ is not smaller than 10^{-6} but the plot shows a decaying impulse response, run Program P2_8 again with a larger value of N.

Q21 Consider the following discrete-time system characterized by the difference equation:

$$y[n]=x[n]-4x[n-1]+3x[n-2]+1.7y[n-1]-y[n-2]$$

Modify Program P2_8 to compute and plot the impulse response of the above system. Is this system stable?

Project 9: Illustration of the Filtering Concept

Consider the following two discrete-time systems characterized by the difference equations:

System No. 1

$$y[n] = 0.5x[n] + 0.27x[n-1] + 0.77x[n-2]$$

System No. 2

$$y[n] = 0.45x[n] + 0.5x[n-1] + 0.45x[n-2] + 0.53y[n-1] - 0.46y[n-2]$$

Program P2_9 is used to compute the outputs of the above two systems for an input $x[n] = \cos\left(\frac{20\pi n}{256}\right) + \cos\left(\frac{200\pi n}{256}\right)$, with $0 \le n < 299$.

Program 2_9

```
% Program P2_9
% Generate the input sequence
clf;
n = 0:299;
x1 = cos(2*pi*10*n/256);
x2 = cos(2*pi*100*n/256);
x = x1+x2;
% Compute the output sequences
num1 = [0.5 0.27 0.77];
y1 = filter(num1,1,x); % Output of System #1
den2 = [1 -0.53 0.46];
num2 = [0.45 0.5 0.45];
y2 = filter(num2,den2,x); % Output of System #2
% Plot the output sequences
subplot(2,1,1);
plot(n,y1);axis([0 300 -2 2]);
ylabel('Amplitude');
title('Output of System #1'); grid;
subplot(2,1,2);
plot(n,y2);axis([0 300 -2 2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Output of System #2'); grid;
```

Questions:

Q22 Run Program P2_9. In this question both filters are lowpass filters but with different attenuation in the stopband, especially at the frequencies of the input signal. Which filter has better characteristics for suppression of the high-frequency component of the input signal $x[n]$?

Q23 Modify Program P2_9 by changing the input sequence to a swept sinusoidal sequence (length 301, minimum frequency 0, and maximum frequency 0.5). Which filter has better characteristics for suppression of the high-frequency component?