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 1 Discrete-Time Signals in the time domain

1.1 Background Review

C1. Digital signal processing is concerned with the processing of a discrete-time signal, called the input signal, to develop another discrete-time signal, called the output signal, with more desirable properties.

C2. A discrete-time signal $x[n]$ is represented as a sequence of numbers, called **samples**.

C3. The discrete-time signal may be a finite length or an infinite length sequence. A finite length (also called **finite duration** or finite extent) sequence is defined only for a finite time interval:

$$N_1 \le n \le N_2 \tag{1.1}$$

where $-\infty < N_1$ and $N_2 < \infty$ with $N_1 \le N_2$. The length or duration N of the finite length sequence is

$$N = N_2 - N_1 + 1 \tag{1.2}$$

C4. A sequence $\tilde{x}[n]$ satisfying

$$\tilde{x}[n] = \tilde{x}[n + kN] \qquad \text{for all } n \tag{1.3}$$

is called a **periodic sequence** with a period N where N is appositive integer and k is any integer.

C5. The **energy** of a sequence $x[n]$ is defined by

$$E = \sum_{n=-\infty}^{\infty} \left| x[n] \right|^2 \tag{1.4}$$

The energy of a sequence over a finite interval $-K \le n \le K$ is defined by

$$E_K = \sum_{n=-K}^{K} \left| x[n] \right|^2 \tag{1.5}$$

C6. The **average power** of a periodic sequence $x[n]$ is defined by

$$P_{av} = \lim_{K \to \infty} \frac{1}{2k+1} E_k = \lim_{K \to \infty} \frac{1}{2k+1} \sum_{n=-K}^{K} \left| x[n] \right|^2 \tag{1.6}$$

The average power of a periodic sequence $\tilde{x}[n]$ with a period N is given by

$$P_{av} = \frac{1}{N} \sum_{n=0}^{N-1} \left| x[n] \right|^2 \tag{1.7}$$

C7. The **unit sample sequence**, often called the **discrete-time impulse** or **the unit impulse**, denoted by $\delta[n]$, is defined by

$$\delta[n] = \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \ne 0 \end{cases} \tag{1.8}$$

The **unit step sequence** denoted by $u[n]$, is defined by

$$u[n] = \begin{cases} 1, & \text{for } n \ge 0 \\ 0, & \text{for } n < 0 \end{cases} \tag{1.9}$$

C8. The **exponential sequence** is given by

$$x[n] = A\alpha^n \qquad (1.10)$$

where A and α are real or complex numbers. By expressing

$$\alpha = e^{(\sigma_0 + jw_0)}, \text{ and } A = |A|e^{j\phi},$$

the equation (1.10) can be rewritten as

$$x[n] = A\alpha^n = |A|e^{\sigma_0 n + j(w_0 n + \phi)} = |A|e^{\sigma_0 n}[\cos(w_0 n + \phi) + j\sin(w_0 n + \phi)] \qquad (1.11)$$

C9. The **real sinusoidal sequence** with a constant amplitude is of the form

$$x[n] = A\cos(w_0 n + \phi) \qquad (1.12)$$

where A, w_0 and ϕ are real numbers. The parameters A, w_0 and ϕ in Eqs. (1.11) and (1.12) are called, respectively, the amplitude, the angular frequency, and the initial phase of the sinusoidal sequence $x[n]$. $f_0 = w_0/2\pi$ is the frequency.

C10. The complex exponential sequence of Eq. (1.11) with $\sigma_0 = 0$ and the sinusoidal sequence of Eq. (1.12) are periodic sequences if $w_0 N$ is an integer multiple of 2π, that is,

$$w_0 N = 2\pi m \qquad (1.13)$$

where N is a positive integer and m is any integer. The smallest possible N satisfying this condition is the period of the sequence.

C11. Given two sequences $x[n]$ and $h[n]$ of length N

The **product** of $x[n]$ and $h[n]$ yields a sequence $y[n]$, also of length N, as given by

$$y[n] = x[n] \cdot h[n] \qquad (1.14)$$

The **addition** of $x[n]$ and $h[n]$ yields a sequence $y[n]$, also of length N, as given by

$$y[n] = x[n] + h[n] \qquad (1.15)$$

The **multiplication** of a sequence $x[n]$ by a scalar A results in a sequence $y[n]$ of length N, as given by

$$y[n] = A \cdot x[n] \qquad (1.16)$$

The **time-reversal** of a sequence $x[n]$ of infinite length results in a sequence $y[n]$ of infinite length, as defined by

$$y[n] = x[-n] \qquad (1.17)$$

The **delay** of a sequence $x[n]$ of infinite length by a positive integer M results in a sequence $y[n]$ of infinite length given by

$$y[n] = x[n - M] \qquad (1.18)$$

If M is a negative integer, the operation indicated in EQ. (1.18) results in an **advance** of the sequence $x[n]$.

A sequence $x[n]$ of length N can be appended by another sequence $g[n]$ of length M resulting in a longer sequence $y[n]$ of length N+M given by

$$\{y[n]\} = \{\{x[n]\}, \{g[n]\}\} \qquad (1.19)$$

1.2 Generation of Sequence

Project 1: Unit Sample and Unit Step Sequences

1. A unit sample sequence *u*[*n*] of length *N* can be generated using the Matlab command

   delta = [1 zeros(1,N-1)];

2. A unit sample sequence *ud*[*n*] of length *N* and delayed by M samples, where M < N, can be generated using the Matlab command

   delta_delay = [zeros(1,M) 1 zeros(1,N-M-1)];

3. A unit step sequence s[n] of length N can be generated using the Matlab command

   u = [ones(1,N)];

Program P1_1: To generate and plot a unit sample sequence

```
%Program P1_1
%Generation of a unit sample sequence
clf;
%Generate a vector from -10 to 20
n = -10:20;
%Generate the unit sample sequence
delta = [zeros(1,10) 1 zeros(1,20)];
%Plot the unit sample sequence
stem(n, delta);
xlabel('Time index n'); ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-10 20 0 1.2]);
```

Questions:

Q1 What are the purposes of the commands **clf, stem, axis, title, xlabel**, and **ylabel**?

Q2 Modify Program P1_1 to generate a delayed unit sample sequence delta_delay[n] with a delay of 11 samples. Run the modified program and display the result.

Q3 Modify Program P1_1 to generate a unit step sequence u[n]. Run the modified program and display the result.

Q4 Modify Program P1_1 to generate a delayed unit step sequence u_dealy[n] with an advance of 7 samples. Run the modified program and display the result.

Project 2: Exponential Signals

The exponential sequence can be generated using the Matlab operators **.^** and **exp.**

Program P1_2: To generate and plot a complex exponential sequence

```
%Program P1_2
%Generation of a complex exponential sequence
clf;
c = -(1/12) + (pi/6)*i;    %α = σ0 + jw0
```

```
k = 2;
n = 0:40;
x = k*exp(c*n);
subplot(2,1,1);
stem(n, real(x))
xlabel('Time index n'); ylabel('Amplitude');
title('Real part');
subplot(2,1,2);
stem(n, imag(x))
xlabel('Time index n'); ylabel('Amplitude');
title('Imaginary part');
```

Questions:

Q5 Which parameter controls the rate of growth or decay of this sequence? Which parameter controls the amplitude of this sequence?

Q6 What will happen if the parameter c is changed to (1/12) + (pi/6)*i?

Q7 What are the purposes of the operators **real** and **imag**?

Q8 What are the purposes of the command **subplot**?.

Program P1_3: To generate and plot a real-valued exponential sequence

```
%Program P1_3
%Generation of a real exponential sequence
clf;
n = 0:35; a=1.2; k = 0.2;
x = k*a.^n;
stem(n, x)
xlabel('Time index n'); ylabel('Amplitude');
title('Real exponential sequence');
```

Questions:

Q10 Which parameter controls the rate of growth or delay of this sequence? Which parameter controls the amplitude of this sequence?

Q11 What will happen if the parameter a is less than 1? Run Program P1_3 again with the parameter a changed to 0.75 and the parameter k changed to 10.

Q12 What is the length of this sequence and how can it be changed?

Q13 What is the difference between the arithmetic operators ^ and .^?.

Q14 To compute the energy of a real sequence u[n] stored as a vector u by using Matlab command **sum**(u.*u) Evaluate the energy of the real-valued exponential sequence x[n] generated in Questions Q11.

Project 3: Sinusoidal Sequences

The sinusoidal sequences can be generated in Matlab using the trigonometric operations cos and sin.

Program P1_4: To generate a sinusoidal signal

```
%Program P1_4
%Generation of a sinusoidal sequence
clf; % clear old graph
n = 0:40; f = 0.1; phase = 0; A = 1.5; % x[n] = Acos(2*pi*f*n – phase)
arg = 2*pi*f*n - phase;
x = A*cos(arg);
stem(n, x);
axis([0 40 -2 2]);
grid;
title('Sinusoidal sequence');
xlabel('Time index n'); ylabel('Amplitude');
axis;
```

Questions:

Q15 What are the purposes of the commands **axis** and **grid**?

Q16 Modify the above program to generate a sinusoidal sequence of length 50, frequency 0.08, amplitude 2.5, and phase shift 90 degrees and display it. What is the period of this sequence?

Q17 Replace the **stem** command in Program P1_4 with the **stairs** command or **plot** command. What the difference between the new plot and the old one?

Project 4: Random Signals

A random signal of length N with samples distributed in the interval (0,1) can be generated by using the Matlab command

1. **uniformly** distributed: **x = rand(1, N)**;
2. **normally** distributed with **zero mean** and **unity variance**: **x = randn(1, N)**;
3. **normally** distributed with ***a*** **mean** and **variance of *b***: **x = *a* + sqrt(*b*)*randn(1, N)**;

Questions:

Q18 Write a Matlab program to generate and display a random signal of length 100 whose elements are uniformly distributed in the interval [-2 2].

Q19 Write a Matlab program to generate and display a Gaussian random signal of length 75 whose elements are normally distributed with zero mean and a variance of 3.

Q20 Write a Matlab program to generate and display five sample sequences of a random sinusoidal signal of length 31

$$\{x[n]\} = \{A\cos(w_0 n + \phi)\}$$

where the amplitude A and the phase ϕ are statistically independent random variables with uniform probability distribution in the range $0 \le A \le 4$ for the amplitude and in the range $0 \le \phi \le 2\pi$ for the phase.

1.3 Simple Operations on Sequences

Let $s[n]$ be the signal corrupted by a random noise $d[n]$ resulting in the noisy signal $x[n] = s[n] + d[n]$. **The objective** is to operate on $x[n]$ to generate a signal $y[n]$ which is a reasonable approximation to $s[n]$. A simple approach is to generate an output sample by averaging a number of input samples around the sample at instant n.

Moving average: $y[n] = \frac{1}{M} \sum_{k=-\frac{M}{2}}^{\frac{M}{2}-1} x[n+k]$

Project 5: Signal Smoothing

Program P1_5: Implements the above algorithm

```
%Program P1_5
%Signal smoothing by averaging
clf; % clear old graph
R = 51; d = 0.8*(rand(R,1) – 0.5); %Generate random noise
m = 0:R-1;
s = 2*m.*(0.9.^m); %Generate uncorrupted signal
x = s + d'; %Generate noise corrupted signal
subplot(2,1,1);
plot(m,d', 'r-', m,s, 'g--', m,x', 'b-.');
xlabel('Time index n'); ylabel('Amplitude');
legend('d[n] ', 's[n] ', 'x[n] ');
x1 = [0 0 x]; x2 = [0 x 0]; x3 = [x 0 0];
y = (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1), 'r-', m,s, 'g--');
legend('y[n] ', 's[n] ');
xlabel('Time index n'); ylabel('Amplitude');
```

Questions:

Q21 What is the purposes of the legend commands?

Q22 What are the relations between the signals x1, x2, and x3, and the signal x?

Q23 Can you use the statement x = s + d to generate the noise-corrupted signal? If not, why not?

Project 6: Generation of Complex Signals

More complex signals can be generated by performing the basic operations on simple signals. For example, an amplitude modulated signal can be generated by modulating a high-frequency sinusoidal signal $x_H[n] = \cos(w_H n)$ with a low-frequency modulating signal $x_L[n] = \cos(w_L n)$. The resulting signal $y[n]$ is of the form

$$y[n] = A(1 + m \cdot x_L[n])x_H[n] = A(1 + m \cdot \cos(w_L n))\cos(w_H n)$$

where m, called the modulation index, is a number chosen to ensure that $(1 + m \cdot x_L[n])$ is positive for all *n*.

Program P1_6: To generate amplitude modulated sequence

```
%Program P1_6
%Generation of amplitude modulated sequence
clf;
n = 0:100;
m=0.4; fH = 0.1; fL = 0.01;
xH = sin(2*pi*fH*n);
xL = sin(2*pi*fL*n);
y = (1 + m*xL).*xH;
stem(n, y); grid;
title(' Amplitude modulated sequence ');
xlabel('Time index n'); ylabel('Amplitude');
```

Questions:

Q24 What is the difference between the arithmetic operators $*$ and $*.$?

As the frequency of a sinusoidal signal is the derivative of its phase with respect to time, to generate a swept-frequency sinusoidal signal whose frequency increases linearly with time, the argument of the sinusoidal signal must be a quadratic function of time. Assume that the argument is of the form $an^2 + bn$ (i.e. the angular frequency is $2an + b$). Solve for the values of *a* and *b* from the given conditions (minimum angular frequency and maximum angular frequency).

Program P1_7: To generate a swept frequency sinusoidal sequence

```
% Program P1_7
n=0:100;
a=pi/2/100;
b=0;
arg=a*n.*n+b*n;
x=cos(arg);
clf;
stem(n,x);
```

```
axis([0,100,-1.5,1.5]);
title('Swept-Frequency Sinusoidal Signal');
xlabel('Time index n');
ylabel('Amplitude');
grid:axis;
```

Questions:

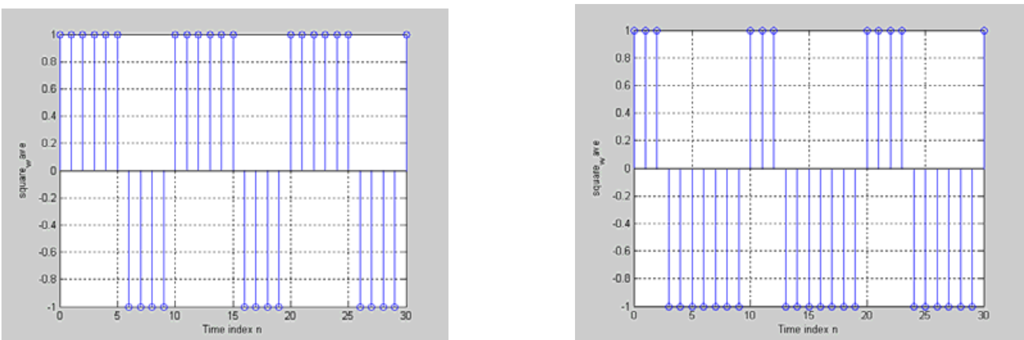
Q25 What are the minimum and maximum frequencies of this signal?

Q26 How can you modify the above program to generate a swept sinusoidal signal with a minimum frequency of 0.1 and maximum frequency of 0.3?

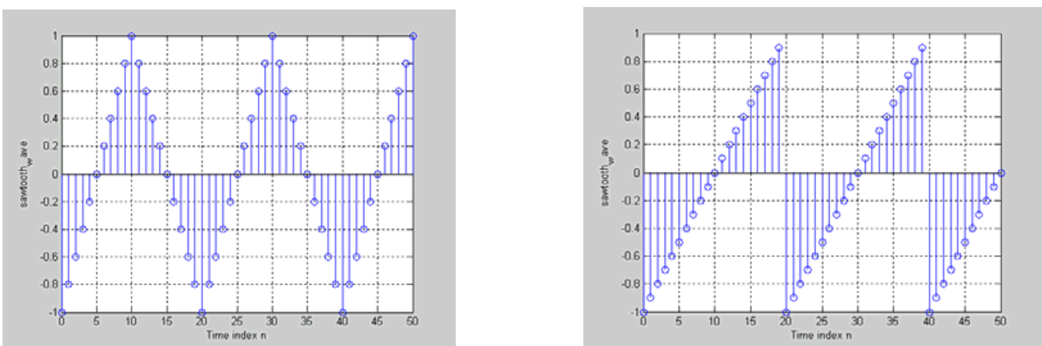
Project 7: Generation of Square-wave and Sawtooth Signals

Matlab functions square and sawtooth can be used to generate sequences of the types shown in Figures 1.1 and 1.2 respectively.

sawtooth(t,width) generates a modified triangle wave where width, a scalar parameter between 0 and 1, determines the point between 0 and 2 at which the maximum occurs. The function increases from -1 to 1 on the interval 0 to 2*width, then decreases linearly from 1 to -1 on the interval 2*width to 2. Thus a parameter of 0.5 specifies a standard triangle wave, symmetric about time instant with peak-to-peak amplitude of 1. sawtooth(t,1) is equivalent to sawtooth(t)



Figures 1.1 Square wave sequences



Figures 1.2 Sawtooth wave sequences