EENG 479 : Digital Signal Processing (DSP)

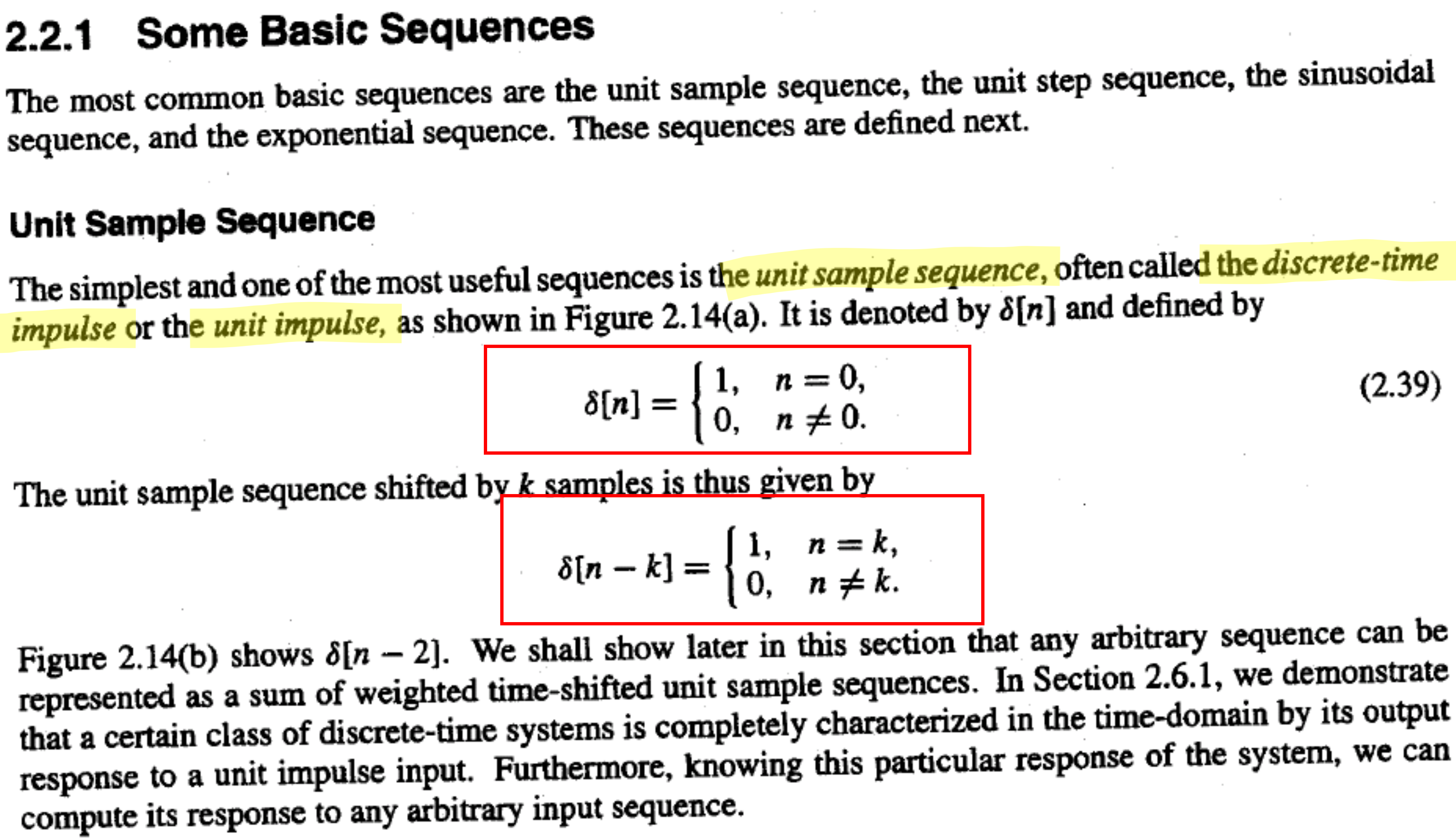
Lecture # 3

2.2 Typical sequences and sequence representation
2.3 The sampling Process

Prof. Mohab A. Mangoud

Professor of Wireless Communications (Networks, IoT and AI)
University of Bahrain, College of Engineering
Department of Electrical and Electronics Engineering
P.O.Box 32038- Kingdom of Bahrain
mmangoud@uob.edu.bh
http://mangoud.com

2.2 Typical sequences and sequence representation



2.2.1 Some Basic Sequences

The most common basic sequences are the unit sample sequence, the unit step sequence, the sinusoidal sequence, and the exponential sequence. These sequences are defined next.

Unit Sample Sequence

The simplest and one of the most useful sequences is the *unit sample sequence,* often called the *discrete-time impulse* or the *unit impulse,* as shown in Figure 2.14(a). It is denoted by $\delta[n]$ and defined by

$$\boxed{\delta[n] = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0. \end{cases}} \tag{2.39}$$

The unit sample sequence shifted by k samples is thus given by

$$\boxed{\delta[n-k] = \begin{cases} 1, & n = k, \\ 0, & n \neq k. \end{cases}}$$

Figure 2.14(b) shows $\delta[n-2]$. We shall show later in this section that any arbitrary sequence can be represented as a sum of weighted time-shifted unit sample sequences. In Section 2.6.1, we demonstrate that a certain class of discrete-time systems is completely characterized in the time-domain by its output response to a unit impulse input. Furthermore, knowing this particular response of the system, we can compute its response to any arbitrary input sequence.

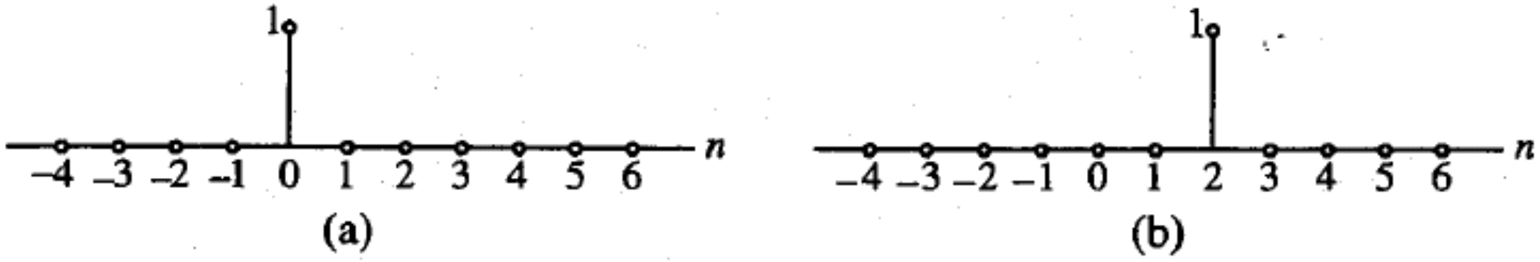


Figure 2.14: (a) The unit sample sequence $\{\delta[n]\}$ and (b) the shifted unit sample sequence $\{\delta[n-2]\}$.

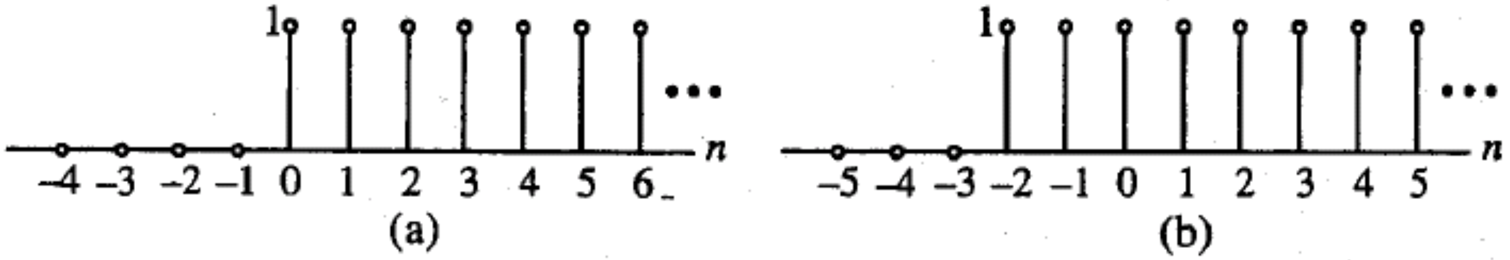


Figure 2.15: (a) The unit step sequence $\{\mu[n]\}$ and the shifted unit step sequence $\{\mu[n+2]\}$.

Unit Step Sequence

A second basic sequence is the *unit step sequence,* shown in Figure 2.15(a). It is denoted by $\mu[n]$ and is defined by

$$\mu[n] = \begin{cases} 1, & n \geq 0, \\ 0, & n < 0. \end{cases} \tag{2.40}$$

The unit step sequence shifted by k samples is thus given by

$$\mu[n-k] = \begin{cases} 1, & n \geq k, \\ 0, & n < k. \end{cases}$$

Figure 2.15(b) shows $\mu[n+2]$.

The unit sample and the unit step sequences are related as follows (Problem 2.3):

$$\mu[n] = \sum_{m=0}^{\infty} \delta[n-m] = \sum_{k=-\infty}^{n} \delta[k], \tag{2.41a}$$

$$\delta[n] = \mu[n] - \mu[n-1]. \tag{2.41b}$$

Sinusoidal and Exponential Sequences

A commonly encountered sequence is the *real sinusoidal sequence* with constant amplitude of the form

$$\boxed{x[n] = A\cos(\omega_o n + \phi), \qquad -\infty < n < \infty,} \tag{2.42}$$

where A, ω_o, and ϕ are real numbers. The parameters A, ω_o, and ϕ are called, respectively, the *amplitude*, the *angular frequency*, and the *phase* of the sinusoidal sequence $x[n]$.

Figure 2.16 shows different types of sinusoidal sequences. The real sinusoidal sequence of Eq. (2.42) can be written alternatively as

$$x[n] = x_i[n] + x_q[n], \tag{2.43}$$

where $x_i[n]$ and $x_q[n]$ are, respectively, the *in-phase* and the *quadrature components* of $x[n]$, and are given by

$$x_i[n] = A\cos\phi\cos(\omega_o n), \qquad x_q[n] = -A\sin\phi\sin(\omega_o n). \tag{2.44}$$

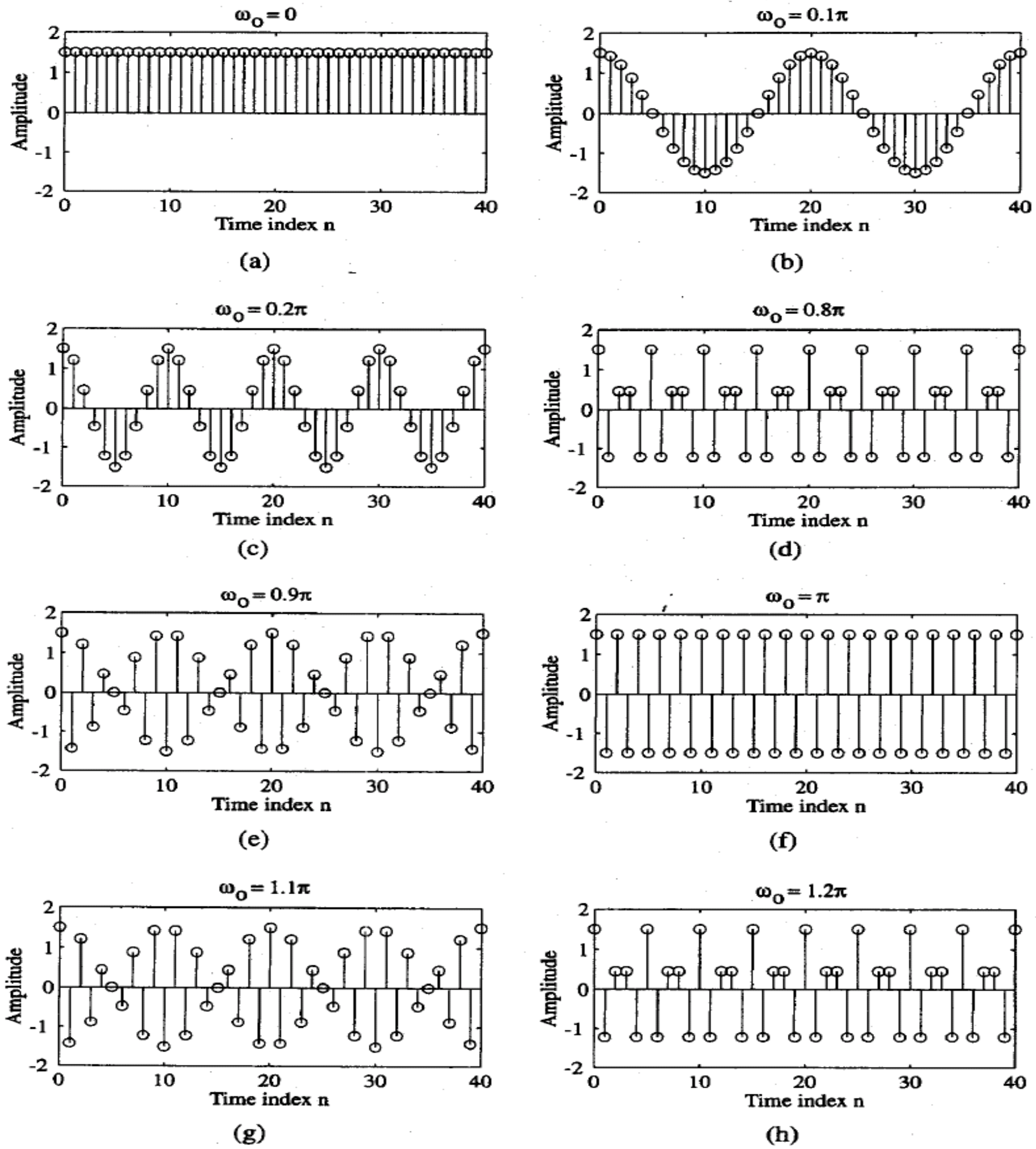


Figure 2.16: A family of sinusoidal sequences given by $x[n] = 1.5 \cos \omega_o n$: (a) $\omega_o = 0$, (b) $\omega_o = 0.1\pi$, (c) $\omega_o = 0.2\pi$, (d) $\omega_o = 0.8\pi$, (e) $\omega_o = 0.9\pi$, (f) $\omega_o = \pi$, (g) $\omega_o = 1.1\pi$, and (h) $\omega_o = 1.2\pi$.

Another set of basic sequences is formed by taking the nth sample value to be the nth power of a real or complex constant. Such sequences are termed *exponential sequences,* and their most general form is given by

$$\boxed{x[n] = A\alpha^n, \qquad -\infty < n < \infty,} \tag{2.45}$$

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

$$\boxed{\alpha = e^{(\sigma_o + j\omega_o)}, \qquad A = |A|\, e^{j\phi},}$$

we can rewrite Eq. (2.45) as

$$x[n] = Ae^{(\sigma_o + j\omega_o)n} = |A|\, e^{\sigma_o n} e^{j(\omega_o n + \phi)} \tag{2.46a}$$

$$= |A|\, e^{\sigma_o n} \cos(\omega_o n + \phi) + j\, |A|\, e^{\sigma_o n} \sin(\omega_o n + \phi), \tag{2.46b}$$

to arrive at an alternative general form of a *complex exponential sequence* where σ_o, ϕ, and ω_o are now real numbers. If we write $x[n] = x_{\text{re}}[n] + j x_{\text{im}}[n]$, then from Eq. (2.46b),

$$\boxed{\begin{aligned} x_{\text{re}}[n] &= |A|\, e^{\sigma_o n} \cos(\omega_o n + \phi), \\ x_{\text{im}}[n] &= |A|\, e^{\sigma_o n} \sin(\omega_o n + \phi). \end{aligned}}$$

Thus, the real and imaginary parts of a complex exponential sequence are real sinusoidal sequences with constant ($\sigma_o = 0$), growing ($\sigma_o > 0$), or decaying ($\sigma_o < 0$) amplitudes for $n > 0$. Figure 2.17 depicts a complex exponential sequence with a decaying amplitude. Note that in the display of a complex exponential sequence, its real and imaginary parts are shown separately.

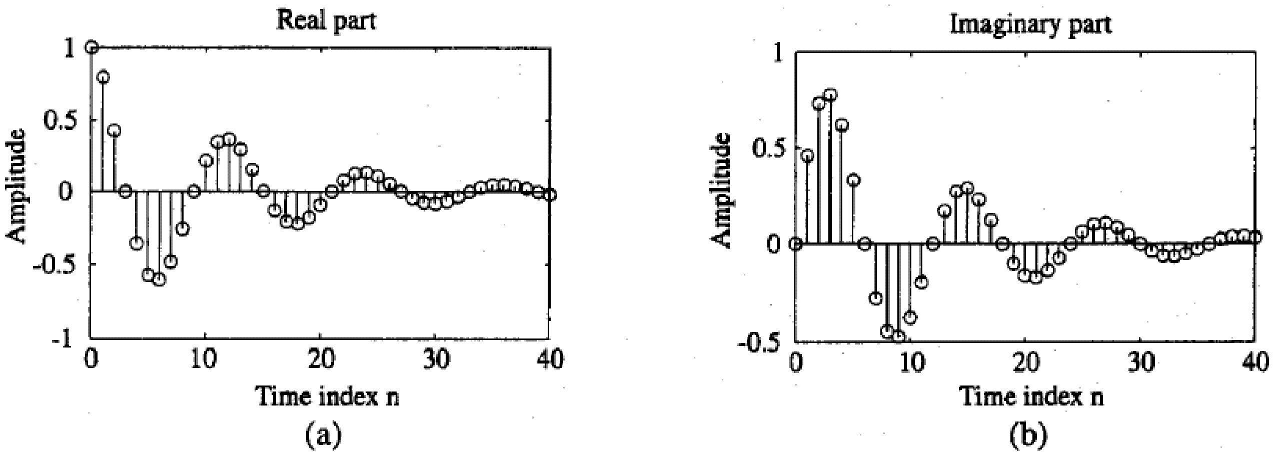


Figure 2.17: A complex exponential sequence $x[n] = e^{(-1/12+j\pi/6)n}$. (a) Real part and (b) imaginary part.

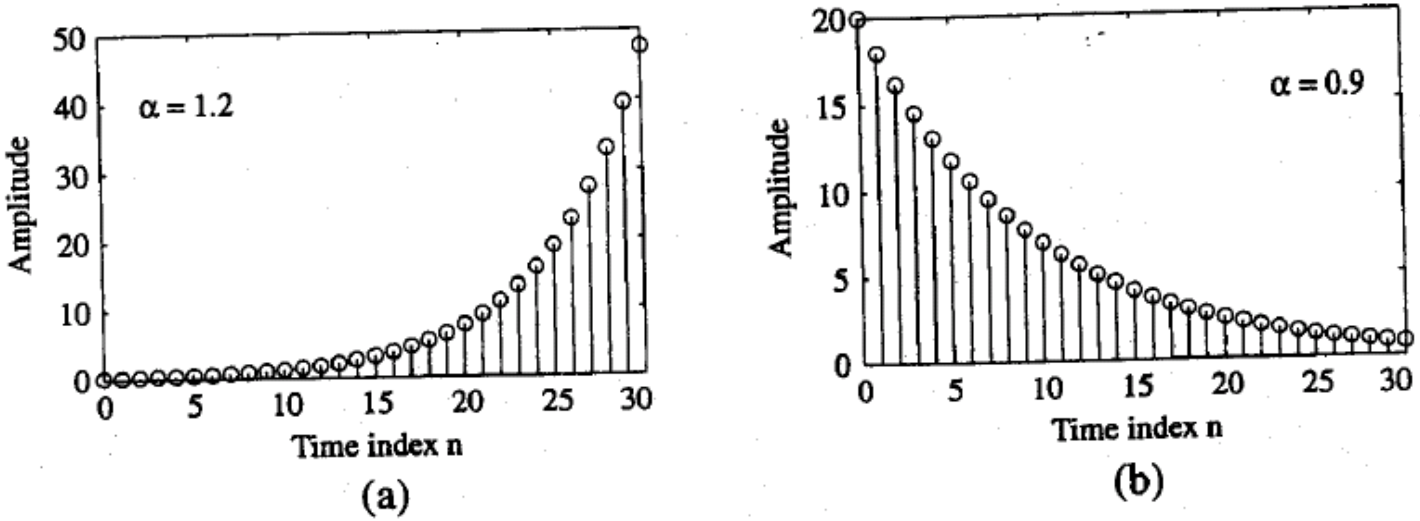


Figure 2.18: Examples of real exponential sequences: (a) $x[n] = 0.2(1.2)^n$, (b) $x[n] = 20(0.9)^n$.

Assignment (1)

Matlab Exercises

Example 2.8: Determination of the period of sinusoidal sequence

Example 2.9: Generation of a sequence wave sequence

Assignment (2)

page 115, M2.1,3,4

Due date next week

2.2.2 Sequence Generation Using MATLAB

MATLAB includes a number of functions that can be used for signal generation. Some of these functions of interest are

```
exp,    sin,    cos,    square,    sawtooth
```

For example, the code fragments to generate a length-N complex exponential sequence with an exponent $a + jb$ and of the form shown in Figure 2.17 is given by

```
n = 1:N;
x = K*exp((a + b*i)*n);
```

The complete code is given in Program 2.2. Likewise, the code fragments to generate a length-$N + 1$ real exponential sequence with an exponent a and of the form shown in Figure 2.18 is given by

```
n = 0:N;
x = K*a.^n;
```

The complete code is given in Program 2.3. Another type of sequence generation using MATLAB can be found earlier in Example 2.1.

```
% Program 2_2
% Generation of complex exponential sequence
%
a = input('Type in real exponent = ');
b = input('Type in imaginary exponent = ');
c = a + b*i;
K = input('Type in the gain constant = ');
N = input ('Type in length of sequence = ');
n = 1:N;
x = K*exp(c*n);                       %Generate the sequence

stem(n,real(x));                      %Plot the real part
xlabel('Time index n');
ylabel('Amplitude');
title('Real part');

disp('PRESS RETURN for imaginary part');
pause
stem(n,imag(x));%Plot the imaginary part
xlabel('Time index n');ylabel('Amplitude');
title('Imaginary part');
```

```
% Program 2_3
% Generation of real exponential sequence
%
a = input('Type in argument = ');
K = input('Type in the gain constant = ');
N = input ('Type in length of sequence = ');
n = 0:N;
x = K*a.^n;
stem(n,x);
xlabel('Time index n');ylabel('Amplitude');
title(['\alpha =  ',num2str(a)]);
```

2.2.3 Representation of an Arbitrary Sequence

An arbitrary sequence can be represented in the time-domain as a weighted sum of a basic sequence and its delayed versions. A commonly used basic sequence in the representation of an arbitrary sequence is the unit sample sequence. For example, the sequence $x[n]$ of Figure 2.21 can be expressed as

$$x[n] = 0.5\,\delta[n+2] + 1.5\,\delta[n-1] - \delta[n-2] + \delta[n-4] + 0.75\,\delta[n-6]. \tag{2.52}$$

An implication of this type of representation is considered later in Section 2.5.1, where we develop the general expression for calculating the output sequence of certain types of discrete-time systems for an arbitrary input sequence.

Since the unit step sequence and the unit sample sequence are simply related through Eq. (2.41a), it is also possible to represent an arbitrary sequence as a weighted combination of delayed unit step sequences (Problem 2.6).

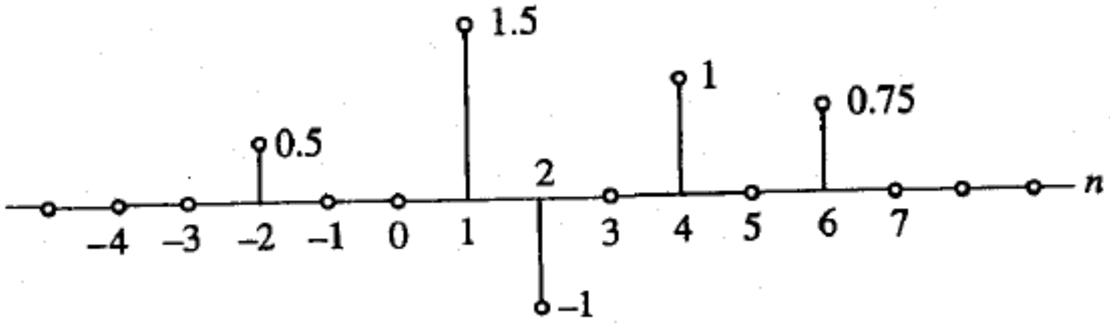
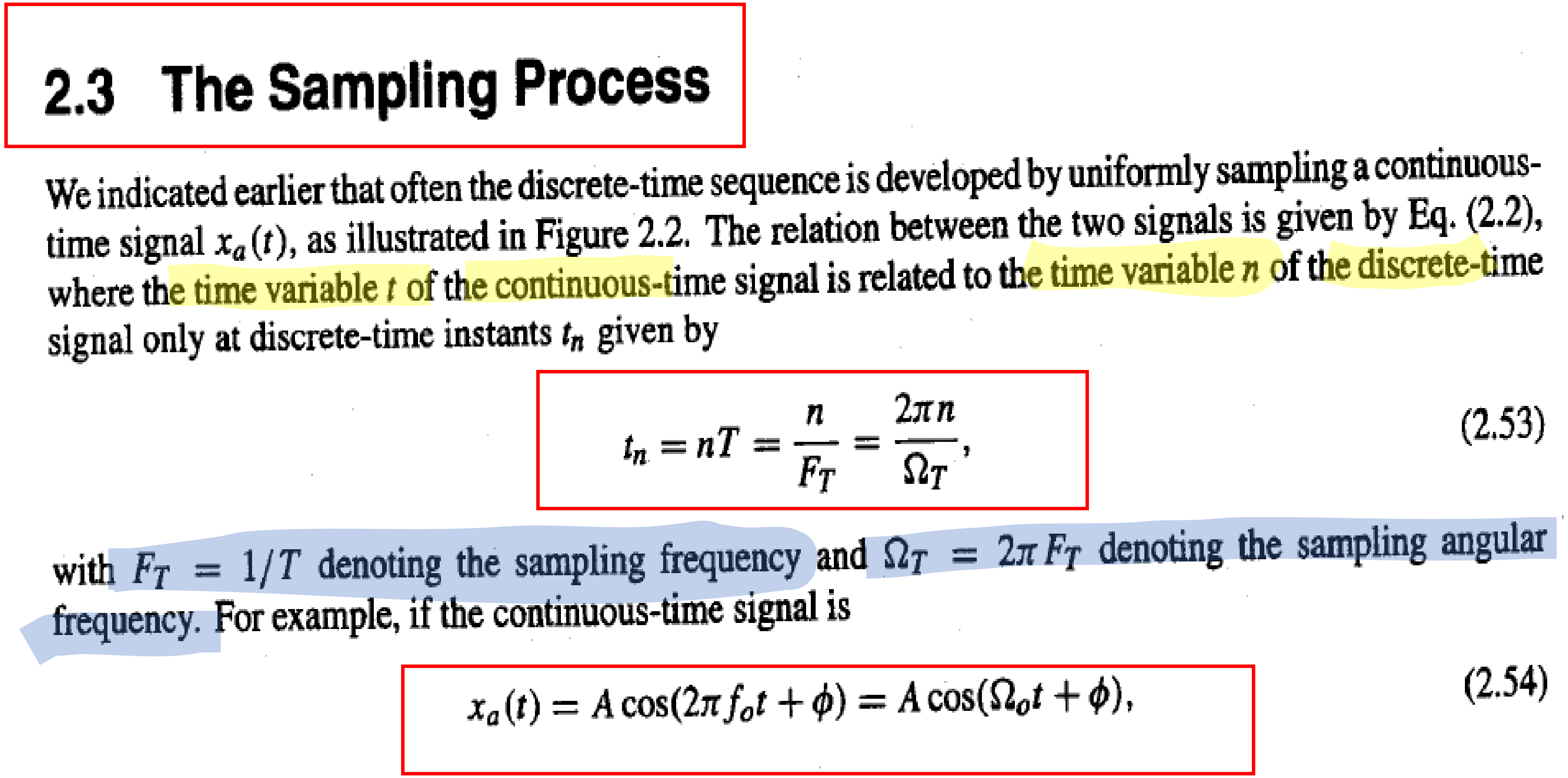


Figure 2.21: An arbitrary sequence $x[n]$.



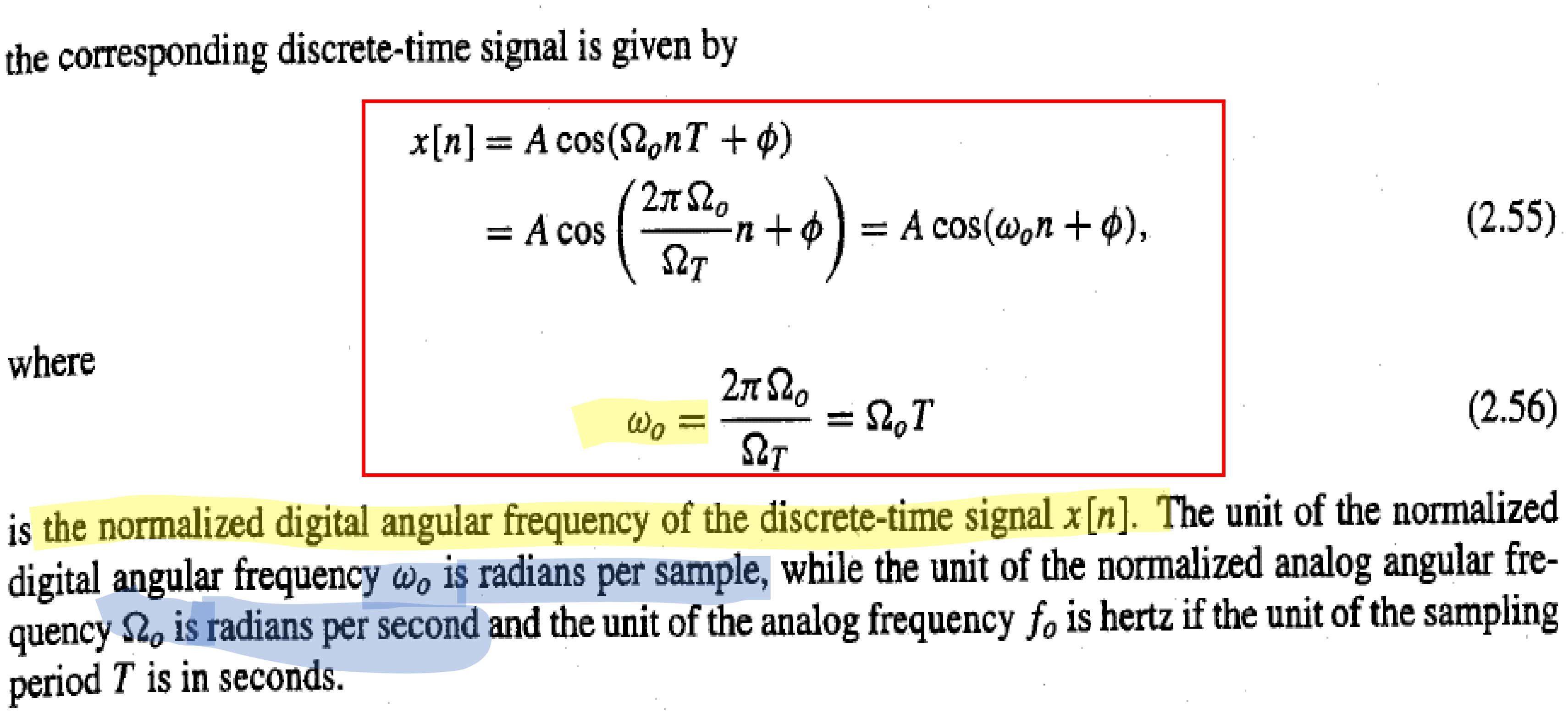
2.3 The Sampling Process

We indicated earlier that often the discrete-time sequence is developed by uniformly sampling a continuous-time signal $x_a(t)$, as illustrated in Figure 2.2. The relation between the two signals is given by Eq. (2.2), where the time variable t of the continuous-time signal is related to the time variable n of the discrete-time signal only at discrete-time instants t_n given by

$$\boxed{t_n = nT = \frac{n}{F_T} = \frac{2\pi n}{\Omega_T},} \tag{2.53}$$

with $F_T = 1/T$ denoting the sampling frequency and $\Omega_T = 2\pi F_T$ denoting the sampling angular frequency. For example, if the continuous-time signal is

$$\boxed{x_a(t) = A\cos(2\pi f_o t + \phi) = A\cos(\Omega_o t + \phi),} \tag{2.54}$$



the corresponding discrete-time signal is given by

$$x[n] = A\cos(\Omega_o nT + \phi) = A\cos\left(\frac{2\pi\Omega_o}{\Omega_T}n + \phi\right) = A\cos(\omega_o n + \phi), \tag{2.55}$$

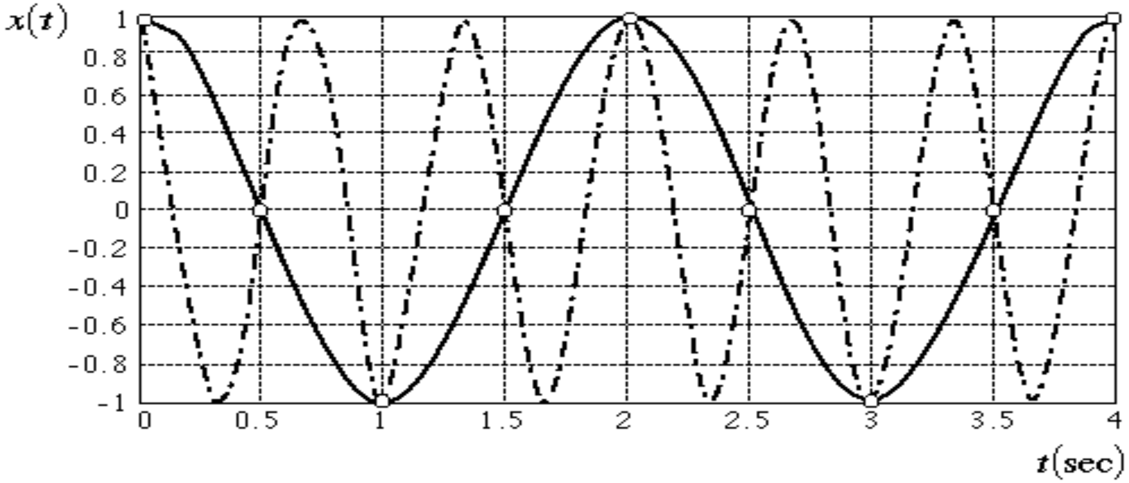
where

$$\omega_o = \frac{2\pi\Omega_o}{\Omega_T} = \Omega_o T \tag{2.56}$$

is the normalized digital angular frequency of the discrete-time signal $x[n]$. The unit of the normalized digital angular frequency ω_o is radians per sample, while the unit of the normalized analog angular frequency Ω_o is radians per second and the unit of the analog frequency f_o is hertz if the unit of the sampling period T is in seconds.

See Example 2.11

Aliasing



—— $x_1(t) = \cos 2\pi\ F_1 t, \quad F_1 = 0.5Hz$

— · — · — $x_2(t) = \cos 2\pi\ F_2 t, \quad F_2 = 1.5Hz$

$F_s = 2\ \ samples/\text{sec} \quad \Rightarrow \quad x_1(nTs) = x_2(nTs)$

Two sinusoidal signals are indistinguishable from their sampled versions whenever $F_s \pm F_1$ is a multiple of the sampling rate F_s

the solid line describes a 0.5Hz continuous-time sinusoidal signal and the dash-dot line describes a 1.5 Hz continuous time sinusoidal signal.

When both signals are sampled at the rate of *Fs equals two* samples/sec,
their samples coincide, as indicated by the circles in the figures. This means that *x one of n (T sub s)* is equal to *x two of n (T sub s)* and there is no way to distinguish the two signals apart from their sampled versions.

This phenomenon, known as aliasing, occurs whenever *F2 plus or minus F1* is a multiple of the sampling rate

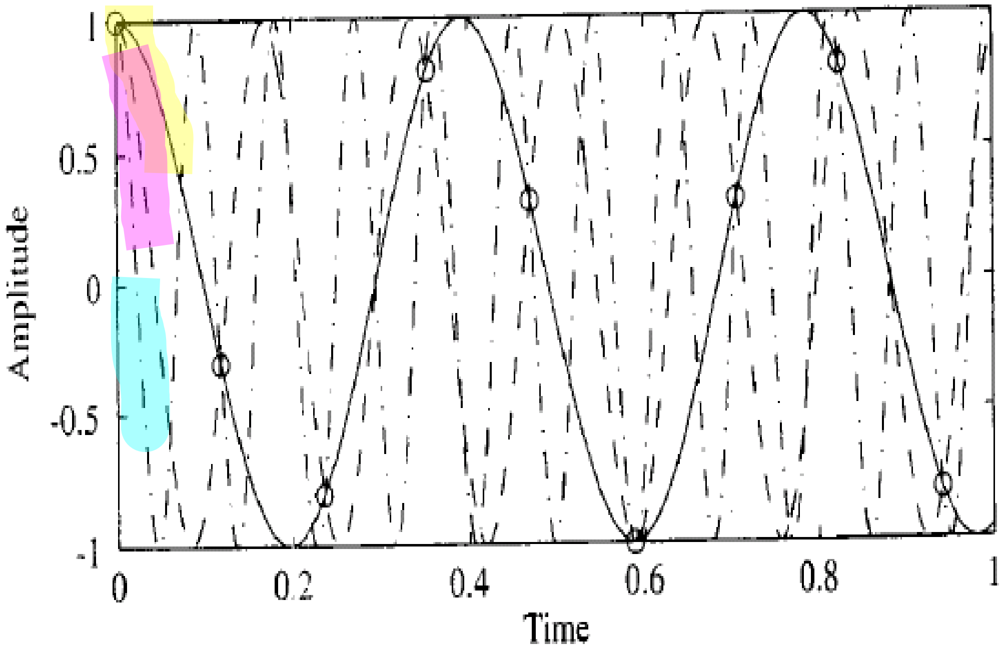
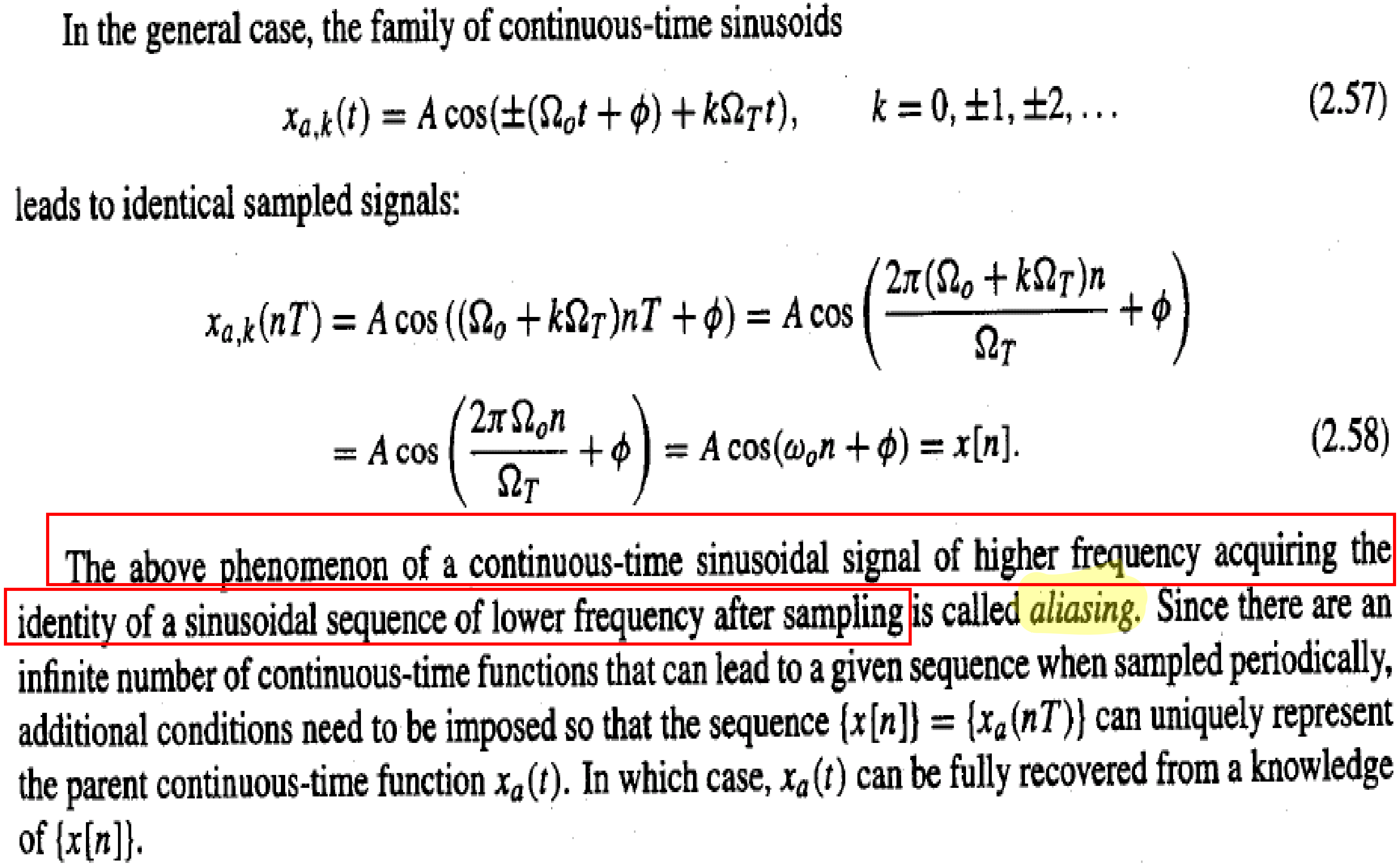


Figure 2.22: Ambiguity in the discrete-time representation of continuous-time signals. $g_1(t)$ is shown with the solid line, $g_2(t)$ is shown with the dashed line, $g_3(t)$ is shown with the dashed-dot line, and the sequence obtained by sampling is shown with circles.

Frequency = 3, 7, 13 Hz, sampling rate = 10 Hz with T=0.1 sec
g1[n]=cos(0.6 .pi. n), g2[n]=cos(1.4 .pi. n), g3[n]=cos(2.6 .pi. n)

As a result all three sequences above are identical and it is difficult to
Associate a unique continuous time function with any one of these sequences



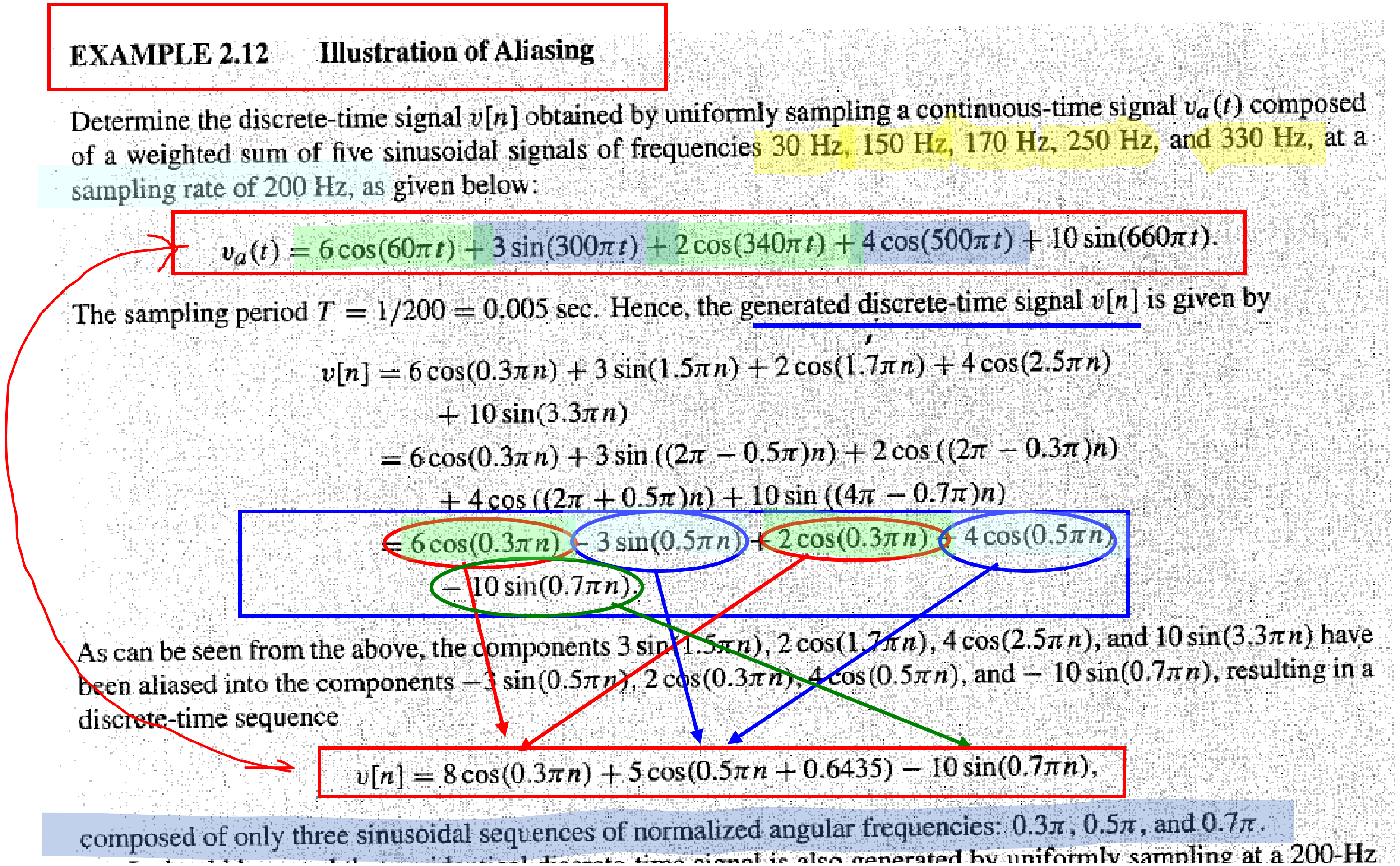
In the general case, the family of continuous-time sinusoids

$$x_{a,k}(t) = A\cos(\pm(\Omega_o t + \phi) + k\Omega_T t), \qquad k = 0, \pm 1, \pm 2, \ldots \tag{2.57}$$

leads to identical sampled signals:

$$\begin{aligned} x_{a,k}(nT) &= A\cos\left((\Omega_o + k\Omega_T)nT + \phi\right) = A\cos\left(\frac{2\pi(\Omega_o + k\Omega_T)n}{\Omega_T} + \phi\right) \\ &= A\cos\left(\frac{2\pi\Omega_o n}{\Omega_T} + \phi\right) = A\cos(\omega_o n + \phi) = x[n]. \end{aligned} \tag{2.58}$$

The above phenomenon of a continuous-time sinusoidal signal of higher frequency acquiring the identity of a sinusoidal sequence of lower frequency after sampling is called *aliasing*. Since there are an infinite number of continuous-time functions that can lead to a given sequence when sampled periodically, additional conditions need to be imposed so that the sequence $\{x[n]\} = \{x_a(nT)\}$ can uniquely represent the parent continuous-time function $x_a(t)$. In which case, $x_a(t)$ can be fully recovered from a knowledge of $\{x[n]\}$.



EXAMPLE 2.12 Illustration of Aliasing

Determine the discrete-time signal $v[n]$ obtained by uniformly sampling a continuous-time signal $v_a(t)$ composed of a weighted sum of five sinusoidal signals of frequencies 30 Hz, 150 Hz, 170 Hz, 250 Hz, and 330 Hz, at a sampling rate of 200 Hz, as given below:

$$v_a(t) = 6\cos(60\pi t) + 3\sin(300\pi t) + 2\cos(340\pi t) + 4\cos(500\pi t) + 10\sin(660\pi t).$$

The sampling period $T = 1/200 = 0.005$ sec. Hence, the generated discrete-time signal $v[n]$ is given by

$$\begin{aligned} v[n] &= 6\cos(0.3\pi n) + 3\sin(1.5\pi n) + 2\cos(1.7\pi n) + 4\cos(2.5\pi n) \\ &\quad + 10\sin(3.3\pi n) \\ &= 6\cos(0.3\pi n) + 3\sin((2\pi - 0.5\pi)n) + 2\cos((2\pi - 0.3\pi)n) \\ &\quad + 4\cos((2\pi + 0.5\pi)n) + 10\sin((4\pi - 0.7\pi)n) \\ &= 6\cos(0.3\pi n) - 3\sin(0.5\pi n) + 2\cos(0.3\pi n) + 4\cos(0.5\pi n) \\ &\quad - 10\sin(0.7\pi n). \end{aligned}$$

As can be seen from the above, the components $3\sin(1.5\pi n)$, $2\cos(1.7\pi n)$, $4\cos(2.5\pi n)$, and $10\sin(3.3\pi n)$ have been aliased into the components $-3\sin(0.5\pi n)$, $2\cos(0.3\pi n)$, $4\cos(0.5\pi n)$, and $-10\sin(0.7\pi n)$, resulting in a discrete-time sequence

$$v[n] = 8\cos(0.3\pi n) + 5\cos(0.5\pi n + 0.6435) - 10\sin(0.7\pi n),$$

composed of only three sinusoidal sequences of normalized angular frequencies: 0.3π, 0.5π, and 0.7π.

It should be noted that an identical discrete-time signal is also generated by uniformly sampling at a 200-Hz

Matlab Aliasing demo

Aliasing Demo (1)

The phenomenon of aliasing happens when the sampling frequency is less than twice the highest frequency of band-limited input signal.

In this example, the input signal is a sinusoidal signal of frequency 1.8KHz.

Three output sound signals are generated in sampling rates 8KHz, 4KHz and 2.6667KHz respectively.

Among these three outputs, we can observe that the aliasing arises only at sampling frequency of 2.6667KHz, which is less than twice of the highest input frequency 3.6KHz.

MATLAB file: aliasing.m