Introduction to DSP — Time Domain Representation of Signals and Systems

Dr. Waleed Al-Hanafy
waleed_alhanafy@yahoo.com
Faculty of Electronic Engineering, Menoufia Univ., Egypt

Digital Signal Processing (ECE407) — Lecture no. 1

July 6, 2011

Overview

References:

[1] S. M. Kuo, B. H. Lee, and W. Tian, Real-Time Digital Signal Processing Implementations and Applications, 2nd ed. John Wiley & Sons Ltd, 2006.

[2] S. Haykin, Adaptive Filter Theory, 3rd ed. Prentice Hall, 1996.

- Signals can be classified as *deterministic* or *random*.

Deterministic signals are used for test purposes and can be described mathematically while random signals are information-bearing signals such as speech.

- A digital signal is a sequence of numbers $x(n), -\infty < n < \infty$, where n is the time index.
- Examples:
  - The *unit-impulse* sequence, with only one nonzero value at $n = 0$, is defined as:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}, \tag{1}$$

    where $\delta(n)$ is also called the *Kronecker* delta function. This unit-impulse sequence is very useful for testing and analysing the characteristics of DSP systems.
  - The *unit-step* sequence is defined as:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}. \tag{2}$$

- *Sinusoidal* signals (sinusoids, tones, or sinewaves) can be expressed in a simple mathematical formula. An analog sinewave can be expressed as:

$$x(t) = A\sin(\Omega t + \phi) = A\ \sin(2\pi f t + \phi), \tag{3}$$

where A is the amplitude of the sinewave.

$$\Omega = 2\pi f$$

is the frequency in radians per second (rad/s), f is the frequency in cycles per second (Hz), and ϕ is the phase in radians. The digital signal corresponding to the analog sinewave defined in (3) can be expressed as:

$$x(n) = A\sin(\Omega nT + \phi) = A\ \sin(2\pi f nT + \phi), \tag{4}$$

where T is the sampling period in seconds. This digital sequence can also be expressed as

$$x(n) = A\sin(\omega n + \phi) = A\ \sin(\pi F n + \phi), \tag{5}$$

where

$$\omega = \Omega T = 2\pi f T = \frac{2\pi f}{f_s} \quad (6)$$

is the digital frequency in radians per sample and

$$F = \frac{\omega}{\pi} = \frac{f}{(f_s/2)} \quad (7)$$

is the normalised digital frequency in cycles per sample.

The units, relationships, and ranges of these analog and digital frequency variables are summarised in Table below

Variables	Unit	Relationship	Range
Ω	Radians per second	$\Omega = 2\pi f$	$-\infty < \Omega < \infty$
f	Cycles per second (Hz)	$f = \frac{F f_s}{2} = \frac{\omega}{2\pi T}$	$-\infty < f < \infty$
ω	Radians per sample	$\omega = \Omega T = \pi F$	$-\pi \le \omega \le \pi$
F	Cycles per sample	$F = \frac{f}{(f_s/2)}$	$-1 \le F \le 1$

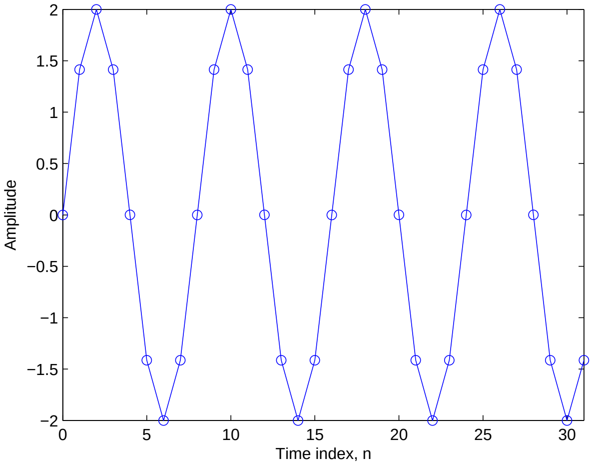
A MATLAB Example

Example: Generate 32 samples of a sinewave with $A = 2$, $f = 1000$ Hz, and $f_s = 8$ kHz using MATLAB program.

Solution: Since $F = \frac{f}{(f_s/2)} = 0.25$, we have $\omega = \pi F = 0.25\,\pi$. From (5), we can express the generated sinewave as $x(n) = 2\sin(\omega n)$, $n = 0, 1, \cdots, 31$. The generated sinewave samples are plotted below

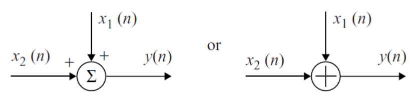
```
n = [0:31];                   % Time index n
omega = 0.25*pi;              % Digital frequency
xn = 2*sin(omega*n);          % Sinewave generation
plot(n, xn, '-o');            % Samples are marked by 'o'
xlabel('Time index, n');
ylabel('Amplitude');
axis([0 31 -2 2]);
```

MATLAB Example (cont'd)



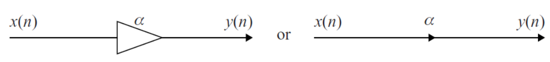
Block Diagram Representation of Digital Systems

- Block diagram of an adder:



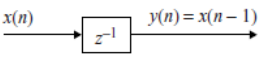
$\longrightarrow y(n) = x_1(n) + x_2(n)$

- Block diagram of a multiplier:



$\longrightarrow y(n) = \alpha x(n)$

- Block diagram of a unit delay:



$\longrightarrow y(n) = x(n-1)$

Implementation Example

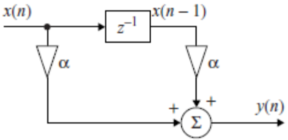
Example: Consider a simple DSP system described by the difference equation

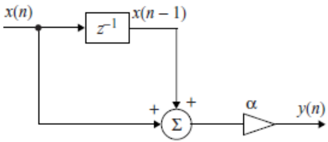
$$y(n) = \alpha x(n) + \alpha x(n-1) \tag{8}$$

Solution: y(n) can be computed using two multiplications and one addition. A simple algebraic simplification of (8) may be used to reduce computational requirements to one multiplication and one addition as:

$$y(n) = \alpha[x(n) + x(n-1)], \tag{9}$$

Both schemes are represented below





Linear Time-Invariant Systems (LTI)

If the input signal to an LTI system is the unit-impulse sequence $\delta(n)$ defined in (1), then the output signal is called the impulse response of the system, $h(n)$.

Example: Consider a digital system with the I/O equation

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) \quad (10)$$

Applying the unit-impulse sequence $\delta(n)$ to the input of the system, the outputs are the impulse response coefficients and can be computed as follows:

$$\begin{aligned} h(0) &= y(0) = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 0 = b_0 \\ h(1) &= y(1) = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 = b_1 \\ h(2) &= y(2) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 1 = b_2 \\ h(3) &= y(3) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 = 0 \\ &\vdots \end{aligned}$$

Therefore, the impulse response of the system defined in (10) is $\{b_0, b_1, b_2, 0, 0, \cdots\}$.

LTI (cont'd)

The I/O equation given in (10) can be generalised with L coefficients, expressed as:

$$\begin{aligned} y(n) &= b_0x(n) + b_1x(n-1) + \cdots + b_{L-1}x(n-L+1) \\ &= \sum_{l=0}^{L-1} b_l x(n-l). \end{aligned} \tag{11}$$

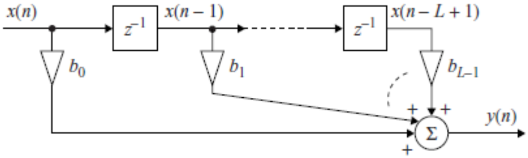
Substituting $x(n) = \delta(n)$ into (11), the output is the impulse response expressed as:

$$\begin{aligned} h(n) &= \sum_{l=0}^{L-1} b_l \delta(n-l) \\ &= \begin{cases} b_n, & n = 0, 1, \cdots, L-1 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Therefore, the length of the impulse response is L for the system defined in (11). Such a system is called a finite impulse response (FIR) filter.

LTI (cont'd)

- The coefficients, $b_l, l = 0, 1, \cdots, L-1$, are called filter coefficients (also called as weights or taps).
- For FIR filters, the filter coefficients are identical to the impulse response coefficients.
- The signal-flow diagram of the system described by the I/O equation in (11) is illustrated below. The string of z^{-1} units is called a tapped-delay line.
- The parameter, L, is the length of the FIR filter. Note that the order of filter is $L-1$ for the FIR filter with length L since they are $L-1$ zeros.



A Causal System

The output of an LTI system in the time-domain as illustrated below can be expressed as

$$\begin{aligned} y(n) &= x(n) \star h(n) = h(n) \star x(n) \\ &= \sum_{l=-\infty}^{\infty} x(l)h(n-l) = \sum_{l=-\infty}^{\infty} h(l)x(n-l), \end{aligned} \qquad (12)$$

where $\star$ denotes the linear convolution.

A digital system is called a causal system if and only if

$$h(n) = 0, \quad n < 0.$$

Causality (cont'd)

A causal system does not provide a zero-state response prior to input application; that is, the output depends only on the present and previous samples of the input.

This is an obvious property for real-time DSP systems since we simply do not have future data.

For a causal system, the limits on the summation of (12) can be modified to reflect this restriction as

$$y(n) = \sum_{l=0}^{\infty} h(l)x(n-l)$$

IIR filter

Exercise (IIR system): Consider a digital system with the I/O equation

$$y(n) = bx(n) - ay(n-1),$$

assuming that the system is causal, i.e., $y(n) = 0$ for $n < 0$ and let $x(n) = \delta(n)$, prove that: $y(n) = (-1)^n a^n b, \quad n = 0, 1, 2, \cdots, \infty$. This system has infinite impulse response $h(n)$ if the coefficients a and b are nonzero. A digital filter can be classified as either an FIR filter or an infinite impulse response (IIR) filter, depending on whether or not the impulse response of the filter is of finite or infinite duration. The I/O equation of the IIR system can be generalised as:

$$y(n) = \sum_{l=0}^{L-1} b_l x(n-l) + \sum_{m=1}^{M} a_m y(n-m) \tag{13}$$

This IIR system is represented by a set of feedforward coefficients $\{b_l, l = 0, 1, \cdots, L-1\}$ and a set of feedback coefficients $\{a_m, m = 1, 2, \cdots, M\}$. Since the outputs are fed back and combined with the weighted inputs, IIR systems are feedback systems. Note that when all a_m are zero, (13) is identical to (11).

IIR (cont'd)

Therefore, an FIR filter is a special case of an IIR filter without feedback coefficients.

Exercise (IIR system): Using MATLAB to implement both FIR and IIR filters.

Answer:

For FIR, yn=filter(b, 1, xn),

and for IIR, yn=filter(b, a, xn)

Conclusion

Concluding remarks

- Digital signals are firstly introduced with the main deterministic examples.
- Digital systems are then modelled with the emphasis of LTI system.
- System causality is defined for such systems.
- Examples of both finite and infinite impulse response systems or filters (FIR & IIR) are studied.
- Graphical representations of these systems are illustrated.
- MATLAB examples of both digital signals and systems are given.