

MIT OpenCourseWare
<http://ocw.mit.edu>

HST.582J / 6.555J / 16.456J Biomedical Signal and Image Processing
Spring 2007

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

Chapter 2 - DIGITAL FILTERS

©Bertrand Delgutte and Julie Greenberg, 1999

Introduction

Digital filtering is one of the most useful type of processing for discrete-time signals. Digital filters arise naturally in the modeling of physical systems, and have been used for numerical calculations at least since the invention of calculus in the seventeenth century. They are used for separating signals from noise and for frequency analysis, an operation which often reveals important features in the signal. The theory of digital filters is so well developed that it is usually easy to design filters that meet even complicated sets of specifications. Much of this theory directly parallels the theory of analog filters which, historically, was developed first. Indeed, even in the present time, digital filters are often intended to simulate the operation of continuous-time physical systems. In these notes, we will emphasize digital filters because of their importance in the laboratory exercises, and because the underlying mathematical concepts are easier to understand with discrete-time signals than with continuous-time signals.

The name “filter” is used because these signal-processing elements typically “pass” or amplify certain frequency components of the signal, while they “stop” or attenuate others. Such a “frequency domain” analysis of digital filters will be treated in Chapter 3. The goal of this chapter is to define digital filters and describe their basic properties.

2.1 Filters defined by linear difference equations

A discrete-time system is any mathematical transformation that maps a discrete-time input signal $x[n]$ into an output signal $y[n]$. We will begin by considering discrete-time systems defined by a *linear, constant-coefficient difference equation (LCCDE)*, which constitute an important class of digital filters:

$$y[n] = \sum_{k=1}^K a_k y[n-k] + \sum_{m=0}^M b_m x[n-m] \quad (2.1)$$

Eq. 2.1 can be used to compute the output $y[n]$ at time n from a finite number of previous values of the input ($x[n]$) and the output. The maximum of the numbers M and K is called the *order* of the filter. If the input signal is only defined after a certain time, say for $n \geq n_0$, then values of both the input and output for a short time prior to n_0 must be known in order to initialize the difference equation. Specifically, $y[n]$ must be known for $n_0 - K \leq n \leq n_0 - 1$, and $x[n]$ for $n_0 - M \leq n \leq n_0 - 1$. In many applications, it is justified to assume that these values are zero

if the system has not received any input for a long time before n_0 , so that its response to any previous inputs has decayed to zero.

2.1.1 Examples of digital filters defined by LCCDEs

1. Simple gain, or amplifier:

$$y[n] = Gx[n]$$

2. Delay of n_0 samples:

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

These first two examples are essential in that any digital filter is a combination of gains and delays.

3. Two-point moving average:

$$y[n] = \frac{1}{2}(x[n] + x[n - 1])$$

4. Euler's formula for approximating the derivative of a continuous-time function:

$$y[n] = \frac{x[n] - x[n - 1]}{T_s}$$

where T_s is the sampling interval.

5. Averaging over N consecutive epochs of duration L :

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n - kL]$$

6. Trapezoidal integration formula:

$$y[n] = \frac{y[n - 1] + (x[n] + x[n - 1])T_s}{2}$$

7. Digital "leaky integrator", or first-order lowpass filter:

$$y[n] = ay[n - 1] + x[n] \quad \text{with } 0 < a < 1$$

This filter is the digital equivalent of an RC analog circuit. It is a useful "building block" for designing complex filters.

8. Digital resonator:

$$y[n] = a_1y[n - 1] + a_2y[n - 2] + bx[n] \quad \text{with } a_1^2 + 4a_2 < 0$$

This is the digital equivalent of the harmonic oscillator.

2.1.2 Finite-impulse response and infinite-impulse response filters

Digital filters can be classified on the basis of the coefficients in Eq. 2.1.

1. **FIR filters.** If all the a_k coefficients in Eq. 2.1 are zero, then the output depends only on a finite number of values of the input. Such filters are called *finite impulse response (FIR), all-zero, or moving average (MA) filters*. The term “moving average” is clear because the output is a weighted sum (or average) of samples of the input. Examples 1–5 above are FIR filters.
2. **IIR filters.** If at least one of the a_k coefficients in Eq. 2.1 is nonzero, the filter is said to have an *infinite impulse response (IIR)*. IIR filters include both autoregressive (AR) and autoregressive, moving average (ARMA) filters, defined as follows:
 - (a) **Autoregressive (AR) filters.** If all of the b_m coefficients except b_0 are zero, the output depends only on the current value of the input and a finite number of past values of the output. Such filters are called *all-pole, purely recursive, or autoregressive (AR) filters*. The term “autoregressive” means that the output is approximately a sum of its own past values. Examples 7 and 8 above are AR filters.
 - (b) **Autoregressive, moving-average (ARMA) filters.** In the general case, both a_k and b_m in Eq. 2.1 are nonzero, with $K > 1$ and $M > 0$. Such filters are called *pole-zero or autoregressive, moving-average (ARMA) filters*. Example 6 above is an ARMA filter.

The origin of the terms *finite impulse response* and *infinite impulse response* is explained in Sec. 2.2.2. The meaning of the terms *all-zero*, *all-pole*, and *pole-zero* will become apparent in the discussion of Z-transforms in Chapter 6.

The distinction between FIR and IIR filters is an important one because signal-processing properties, design techniques, and implementation issues are quite different for these two types of filters.

2.1.3 Recursively computable difference equations

For IIR filters, a difference equation of the form of Eq. 2.1 does not define a unique digital filter, but $K + 1$ distinct filters. Consider for example the second-order difference equation:

$$y[n] = a_1y[n - 1] + a_2y[n - 2] + x[n] \quad (2.2)$$

This equation can be rewritten as both

$$y[n - 1] = 1/a_1(y[n] - a_2y[n - 2] - x[n]) \quad (2.3)$$

and

$$y[n - 2] = 1/a_2(y[n] - a_1y[n - 1] - x[n]). \quad (2.4)$$

Each of these three forms of the difference equation correspond to a distinct digital filter. Of these three filters, the one described by Eq. 2.2 is of the most practical significance, because

the present value of $y[n]$ is computed recursively as a function of the past values $y[n-1]$ and $y[n-2]$. This filter is said to be *causal*. Causality is discussed further in Sec. 2.3.1. The third filter also corresponds to a recursively computable formula for $y[n-2]$ as a function of future values $y[n-1]$ and $y[n]$. This is called an *anticausal* filter. The second filter is of the least practical significance, because it does not lead to a recursively computable formula; computing the value of $y[n-1]$ requires having already computed both the past value $y[n-2]$ and the future value $y[n]$. Therefore it cannot be implemented on a digital computer.

Generalizing, among the $K+1$ filters that can be theoretically defined from a difference equation of order K , only the causal filter and the anticausal filter are recursively computable. These two forms can always be derived from each other by simply reversing the time axis, so there is no loss of generality in restricting our attention to causal IIR filters.

2.2 LTI systems

An important category of discrete-time systems are those which have the properties of linearity and time-invariance.

1. **Linearity.** Linearity consists of both superposition and scaling.
 - (a) **Superposition.** If the response of a discrete-time system to $x_1[n]$ is $y_1[n]$, and the response to $x_2[n]$ is $y_2[n]$, then the response to $x_1[n] + x_2[n]$ is $y_1[n] + y_2[n]$.
 - (b) **Scaling.** If the response of a discrete-time system to $x[n]$ is $y[n]$, then the response to $cx[n]$ is $cy[n]$, where c is a real or complex constant.
2. **Time invariance.** If the response of a discrete-time system to $x[n]$ is $y[n]$, then the response to $x[n-n_0]$ (the input $x[n]$ delayed by n_0 samples) is $y[n-n_0]$ (the original response delayed by n_0 samples).

Systems that have both of these properties are referred to as linear time-invariant (LTI) systems.

It can be verified by inspection from Eq. 2.1 that all digital filters (both FIR and IIR) defined by a linear, constant-coefficient difference equation are LTI. If, however, the filter coefficients a_k and b_m in Eq. 2.1 are functions of sample number n , the system will be linear, but not time-invariant. This is the case, for example, in linear amplitude modulation: $y[n] = ax[n] \cos(2\pi f_c n + \theta)$.

Although a strict definition of digital filters might only include those that are LTI, in practice the term filter often used to refer to nonlinear operations that resemble LTI systems in some respects. For example, *median filters* give as output the median of a finite number of samples of the input. They resemble moving-average filters, except that the linear average is replaced by the nonlinear median operation. Median filters are nonlinear, but time-invariant. *Adaptive filters* are discrete-time systems for which the filter coefficients a_k and b_m vary with time sufficiently slowly that, over short times, they behave like linear, time-invariant systems, even though over long times such systems are neither linear, nor time-invariant. Usually, the coefficients of adaptive

filters are adjusted as a function of the input and output signals to meet certain performance criteria such as providing the best signal-to-noise ratio.

2.2.1 Response of LTI systems to arbitrary inputs

We will now consider general properties of linear, time-invariant systems, keeping in mind that digital filters defined by LCCDEs are an important class of LTI systems. In this section, we will show that **the response of an LTI system to an arbitrary signal can be completely characterized by its response to one particular signal, the unit sample**. The *unit sample* $\delta[n]$ is also called the discrete-time impulse and is defined by:

$$\delta[n] \triangleq \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0. \end{cases}$$

The key to proving this property is to write the input signal $x[n]$ as a weighted sum of delayed unit samples:

$$x[n] = \sum_{m=-\infty}^{\infty} x[m]\delta[n-m]$$

For example, Fig. 2.1 shows the decomposition of a triangular signal into a sum of unit samples.

Now, let $h[n]$ be the response of an LTI system to the unit sample $\delta[n]$. By the time-invariance property of LTI systems, the response to the delayed unit sample $\delta[n-m]$ must be the delayed unit-sample response $h[n-m]$. By the scaling property, the response to $x[m]\delta[n-m]$ is $x[m]h[n-m]$. Note that $x[m]$ is considered to be a constant weighting factor for the delayed unit sample because it does not depend on the time index n . Finally, by the superposition principle, the response to $x[n]$ can be written as a weighted sum of the $h[n-m]$:

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \triangleq x[n] * h[n] \quad (2.5)$$

This expression is by definition the *discrete convolution* of $x[n]$ with $h[n]$, denoted $x[n] * h[n]$.

Therefore, if we know the response of an LTI system to a unit impulse (denoted $h[n]$), then we can determine the response of that system to any arbitrary input. This is a powerful property of LTI systems, and does not hold for systems that are nonlinear or time-varying.

2.2.2 Determining the impulse response for digital filters described by LCCDEs

Equation 2.5 shows that the response of an LTI system to any signal can be computed if the system's *unit-sample response*, or *impulse response*, $h[n]$ is known. Of course this property applies to digital filters defined by a linear constant-coefficient difference equation (Eq. 2.1), a special case of LTI systems.

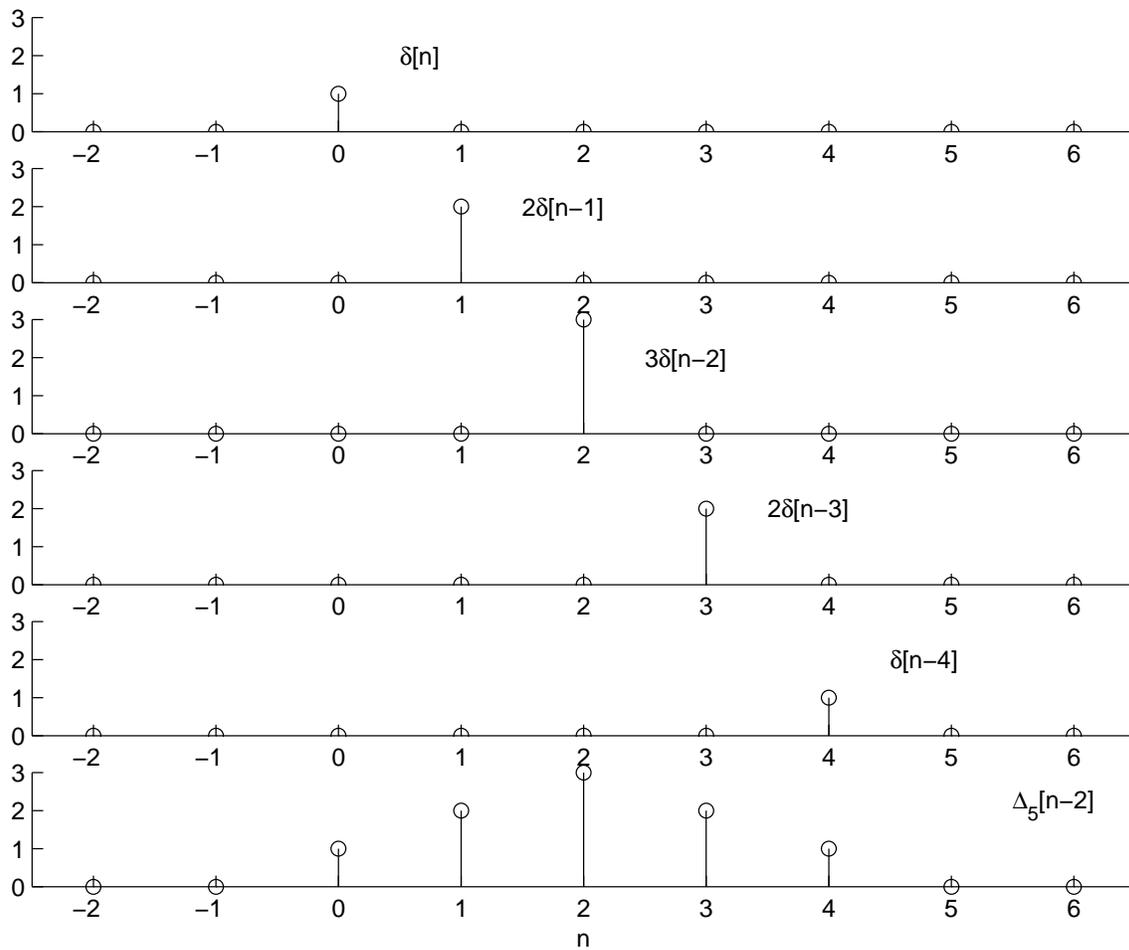


Figure 2.1: Decomposition of the triangular signal $\Delta_5[n-2]$ into a weighted sum of unit samples.

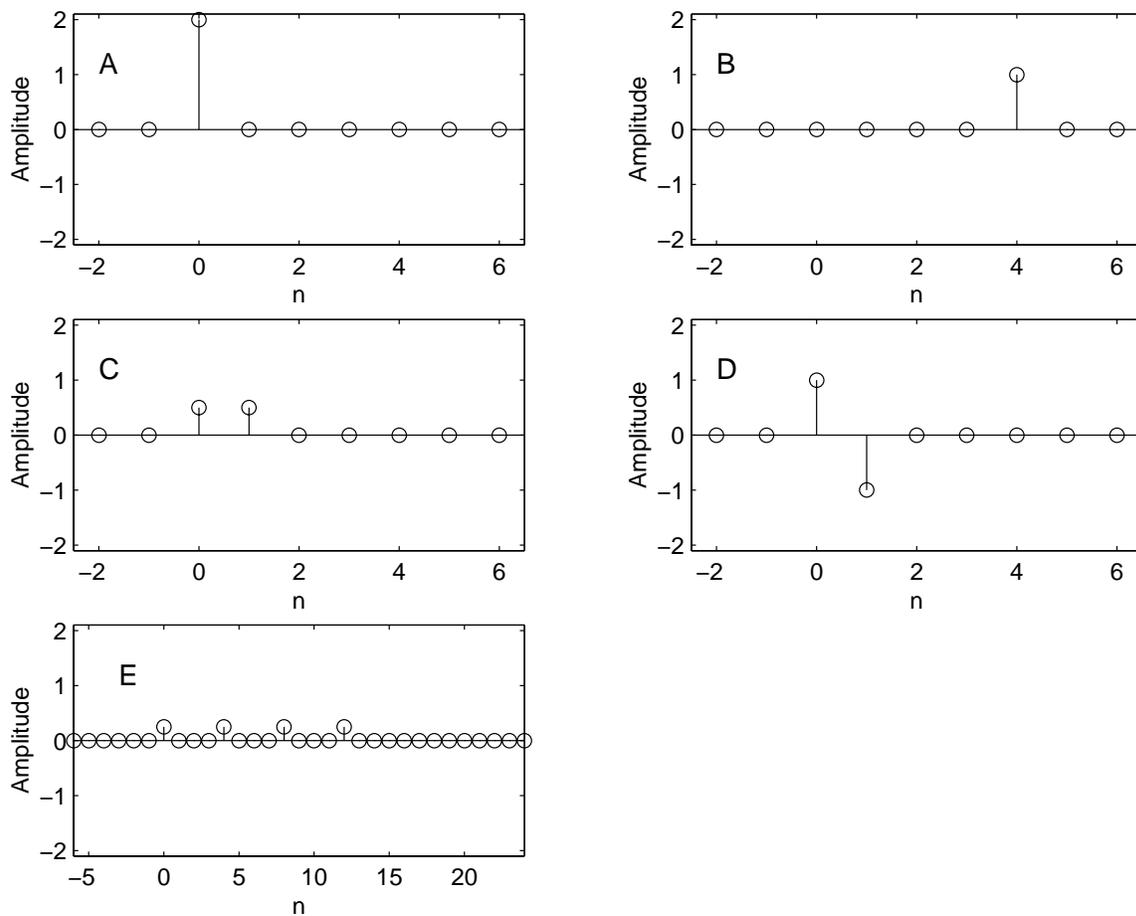


Figure 2.2: Unit sample responses of simple FIR filters (A) Gain with $G = 2$. (B) Delay with $n_0 = 4$. (C) Two-point moving average. (D) Euler's approximation to the derivative with $T_s = 1$. (E) Averager with $N = 4$ and $L = 6$.

For FIR filters, the unit-sample response can be found by inspection from the b_m coefficients:

$$h[m] = \begin{cases} b_m & \text{if } 0 \leq m \leq M \\ 0 & \text{otherwise} \end{cases}$$

The observation that $h[n]$ is zero except for the finite set of times $[0, M]$ justifies the term *finite-impulse response* filter. Figure 2.2 shows the unit-sample responses of the FIR filter examples from Sec. 2.1.1. The corresponding mathematical expressions are:

1. Gain:

$$h[n] = G\delta[n]$$

2. Delay:

$$h[n] = \delta[n - n_0]$$

3. Two-point moving average:

$$h[n] = \frac{1}{2}\delta[n] + \frac{1}{2}\delta[n - 1]$$

4. Euler's approximation to the derivative:

$$h[n] = (\delta[n] - \delta[n - 1])/T_s$$

5. Averager:

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} \delta[n - kL]$$

Note that the impulse response of any FIR filter can be represented as a superposition of impulse responses consisting of simple gains and delays.

For IIR filters, the unit-sample response cannot, in general, be found by inspection from Eq. 2.1, because it is of infinite duration (hence its name). One method suitable for digital computers is to recursively apply the difference equation (Eq. 2.1) with a unit sample as the input. For example, it is straightforward to verify by induction on n that the unit-sample response of the first-order lowpass filter is $h[n] = a^n u[n]$, where $u[n]$ is the unit step function defined in Eq. 2.7. We will give a general analytic method for finding the unit-sample response of IIR filters in Chapter 6.

2.2.3 Properties of convolution/Combinations of filters

By making the substitution of variable $l = n - m$, it is easy to verify that:

$$x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n - m] = \sum_{l=-\infty}^{\infty} h[l]x[n - l] = h[n] * x[n] \quad (2.6)$$

Therefore, convolution is a *commutative* operation.

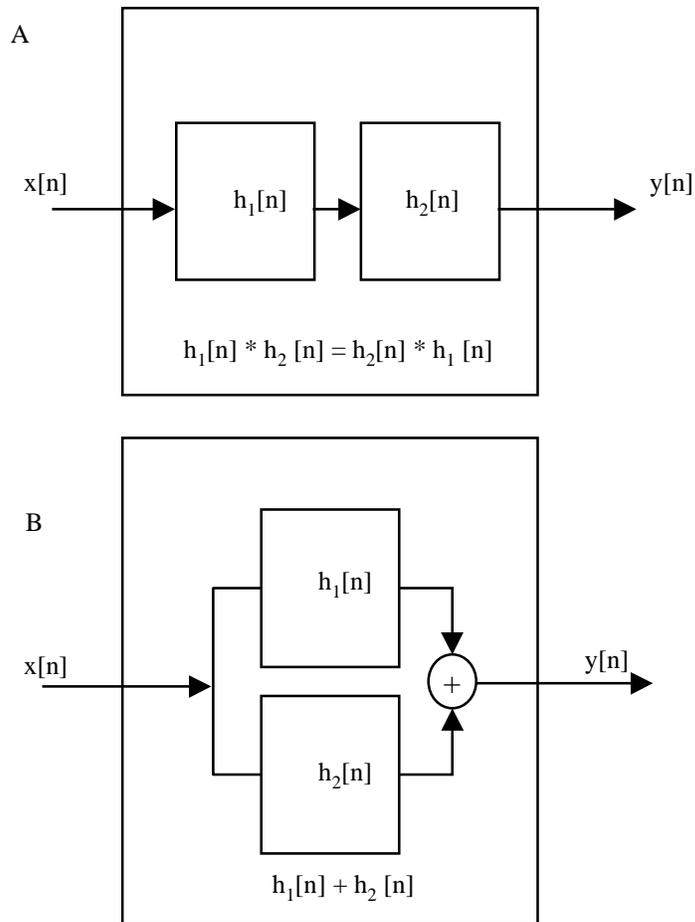


Figure 2.3: Combination of filters in cascade (A) and in parallel (B).

Using another change of variables, we can also verify the following:

$$x[n] * (h_1[n] * h_2[n]) = (x[n] * h_1[n]) * h_2[n]$$

In other words, convolution is an *associative* operation.

When the output of a filter with unit-sample response $h_1[n]$ is used as input to another filter with unit-sample response $h_2[n]$, the two filters are said to be placed in *cascade* (Fig. 2.3(A)). From the associative property of convolution, we can see that the unit-sample response of cascaded filters is *simply the convolution of their unit-sample responses*. Because convolution is commutative (i.e. $h_1[n] * h_2[n] = h_2[n] * h_1[n]$), this result also implies that *the output cascade of LTI filters does not change if the order of the filters is changed*.

For example, Fig. 2.4 shows that any IIR filter defined by a difference equation of the form Eq. 2.1 can be considered as the cascade of an FIR filter and a purely recursive filter. The order in which these two filters are placed does not change the final result, which leads to an efficient implementation of the filter requiring only a number of storage and delay elements equal to the filter order.

Simple manipulations also reveal that

$$(x[n] * h_1[n]) + (x[n] * h_2[n]) = x[n] * (h_1[n] + h_2[n]),$$

indicating that convolution is *distributive over addition*.

This gives a useful result for *parallel* combinations of LTI filters, in which one forms the sum of the outputs of two (or more) filters (Fig. 2.3(B)). As a result of the distributivity of convolution over addition, the unit-sample response of a parallel combination of filters is the sum of their unit-sample responses. Decomposition of a high-order filter into a parallel combination of simpler ones is a very common technique for analysis and design of filters. In fact, it is the basis for the method of finding the unit-sample response of IIR filters given in Chapter 6.

2.2.4 Convolution example

To illustrate the computation of a convolution, we will consider the response of the first-order lowpass filter to a rectangular pulse of duration N :

$$x[n] = u[n] - u[n - N] = \begin{cases} 1 & \text{if } 0 \leq n \leq N - 1 \\ 0 & \text{otherwise} \end{cases}$$

where $u[n]$ is the unit step, defined by

$$u[n] \triangleq \begin{cases} 0 & \text{if } n < 0 \\ 1 & \text{if } n \geq 0 \end{cases} \quad (2.7)$$

The exponential unit-sample response of the filter $h[n] = a^n u[n]$ and the input $x[n]$ are shown in Figs. 2.5(A) and (B) respectively. To compute the output, we will use the second form of convolution in Eq. 2.6:

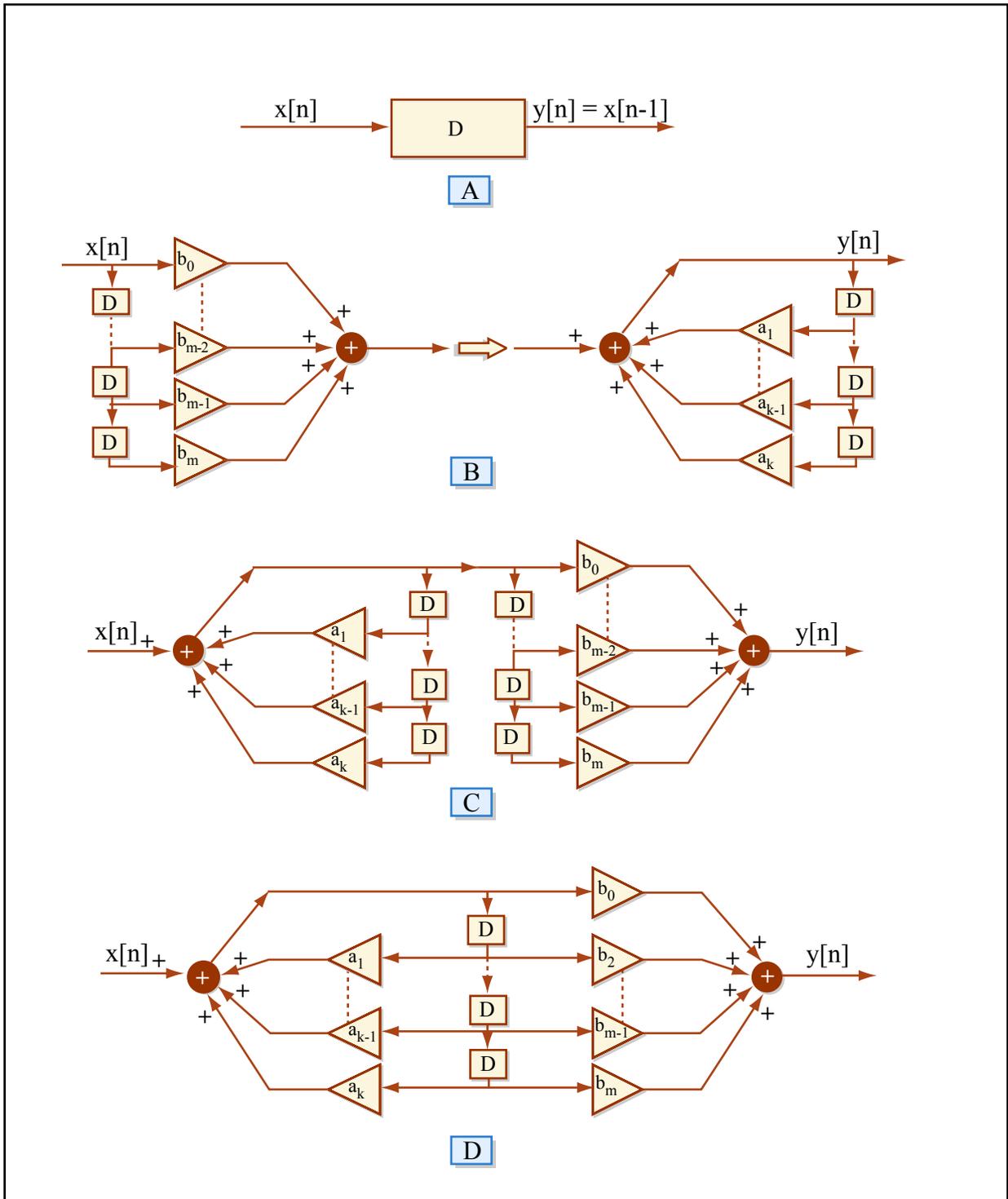


Figure by MIT OpenCourseWare.

Figure 2.4: (A) Block diagram for a unit delay. (B) Decomposition of a digital filter into the cascade of an FIR filter and a purely recursive filter. (C) Interchanging the order of the cascaded filters in B. (D) Combining the two filters in C for efficient implementation.

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} a^m u[m] x[n-m] = \sum_{m=0}^{\infty} a^m x[n-m] \quad (2.8)$$

Three regions must be distinguished:

1. For $n < 0$, $x[n-m]$ is equal to zero for $m \geq 0$, so that $y[n] = 0$. This is generally true if both $x[n]$ and $h[n]$ are zero for negative times.
2. For $0 \leq n \leq N-1$, the sum in Eq. 2.8 is from $m=0$ to n because $x[n-m]$ is zero for $m > n$. Therefore:

$$y[n] = \sum_{m=0}^n a^m = \frac{1-a^{n+1}}{1-a}$$

In this range, $y[n]$ exponentially approaches the asymptote $\frac{1}{1-a}$.

3. For $n \geq N$, $x[n-m]$ is zero outside of the interval $n-N+1 \leq m \leq n$, so that Eq. 2.8 becomes:

$$y[n] = \sum_{m=n-N+1}^n a^m = a^n - N + 1 \frac{1-a^N}{1-a} = \frac{1-a^{-N}}{1-a^{-1}} a^n$$

Thus, $y[n]$ exponentially decays to zero.

The output signal $y[n]$ is shown in Fig. 2.5(C).

2.3 Causality and stability

2.3.1 Causality

A discrete-time system is said to be *causal* if its response at time n_0 depends only on the input for times $n \leq n_0$. For LTI filters, it is easy to see from the convolution formula (Eq. 2.5) that this will be the case if and only if the unit-sample response $h[n]$ is zero for $n < 0$. In Sec. 2.1.3 we claimed that one interpretation of a digital filter defined by a linear, constant-coefficient difference equation naturally leads to a causal filter because the output sample $y[n]$ is recursively computable from previous samples of the output signal, and past and present values of input. Note that previous samples of output signal are, in turn, computed from even older values of the input and output signals, which ultimately depend only on past values of the input and initial conditions.

Causality is necessary for processing signals in real time, for example in control applications. On the other hand, when the signal has been recorded and stored prior to processing, the notions of “past” and “future” become largely a matter of convention, and it is possible to use non-causal filters. Causality is of little relevance for signals where the independent variable is not time, such as digital images. The non-causal FIR filters described by

$$y[n] = \sum_{m=-M}^M b_m x[n-m]$$

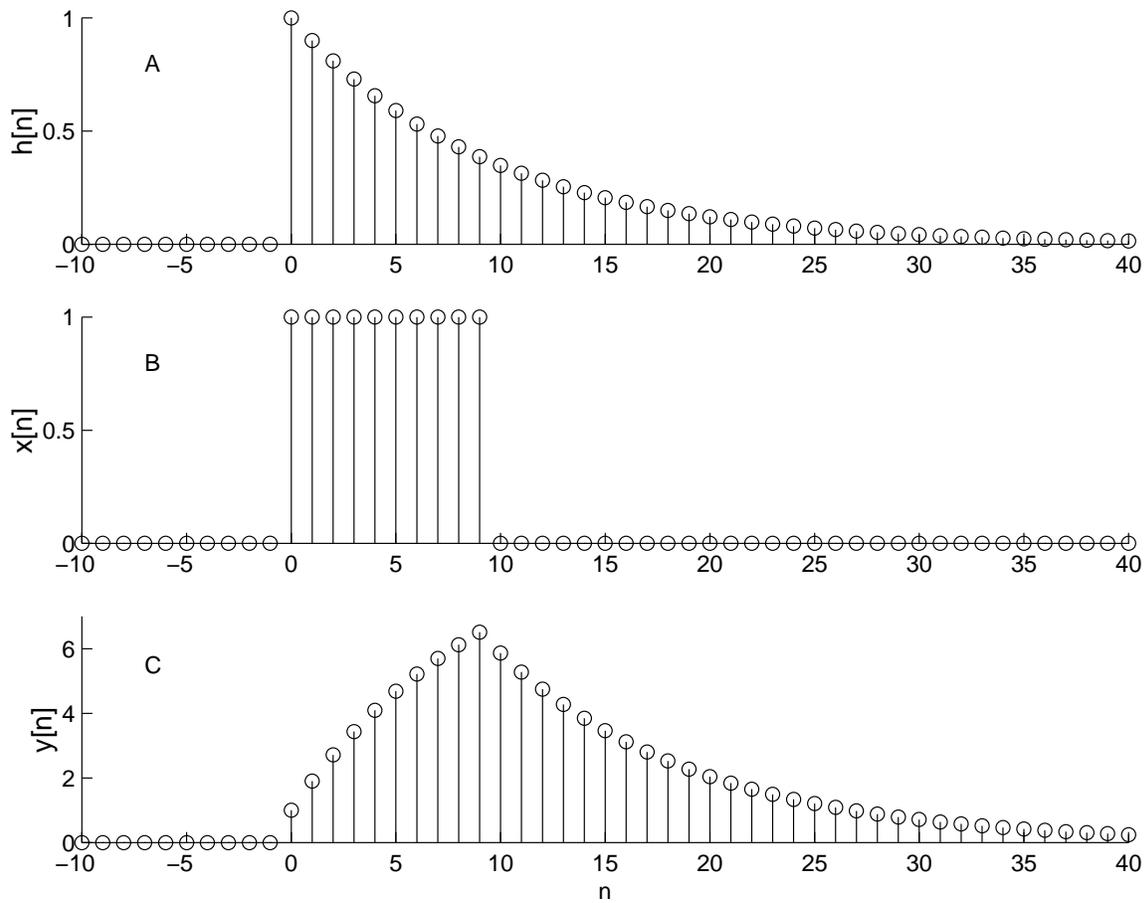


Figure 2.5: Convolution example: (A) Unit-sample response of the first-order low-pass filter, $h[n] = a^n u[n]$, with $a = 0.9$. (B) Input signal, $x[n] = u[n] - u[n - N]$, with $N = 10$. (C) Output signal.

is an example of a class of LTI digital filters which are not described by a linear, constant-coefficient difference equation of the form of Eq. 2.1. FIR filters of this form whose unit-sample response is symmetric around the origin (i.e. $b_m = b_{-m}$), are referred to as *zero-phase*, since they introduce no delay in the processing. One can always change a zero-phase FIR filter into a causal filter by shifting its unit-sample response by half the duration of the unit-sample response.

2.3.2 Stability

A system is said to be *stable* if a bounded input gives a bounded output. For an LTI system to be stable, it is necessary and sufficient that its unit-sample response be absolutely summable:

$$\sum_{n=-\infty}^{\infty} |h[n]| = C < \infty \quad (2.9)$$

To show that this condition is sufficient, let $x[n]$ be a bounded signal, i.e. $|x[n]| \leq B$ for all n . Then, the response of the system $y[n]$ can be shown to be bounded:

$$|y[n]| = \left| \sum_{m=-\infty}^{\infty} h[m]x[n-m] \right| \leq \sum_{m=-\infty}^{\infty} |x[n-m]| |h[m]| \leq B \sum_{m=-\infty}^{\infty} |h[m]| = BC < \infty$$

If $h[n]$ is of finite duration, then it must be absolutely summable because the infinite sum in Eq. 2.9 reduces to a finite sum. Therefore, *FIR filters are always stable*. IIR filters are not necessarily stable; for example, the first-order lowpass filter is unstable if $|a| \geq 1$ because its unit-sample response $a^n u[n]$ is not absolutely summable. The Z-transform, considered in Chapter 6, provides a general stability condition for IIR filters.

2.4 Continuous-time (“analog”) filters (optional)

2.4.1 Definition and properties

An important class of analog filters are continuous-time system defined by a linear, constant-coefficient differential equation of the form

$$\sum_{k=0}^K a_k \frac{d^k y(t)}{dt^k} = \sum_{m=0}^M b_m \frac{d^m x(t)}{dt^m} \quad (2.10)$$

This type of differential equation is obtained when solving input-output relations for electrical networks consisting of linear elements such as resistors, capacitors, inductors and amplifiers, or for mechanical networks consisting of masses, linear springs and damping elements. Properties of analog filters closely resemble those of digital filters, so that we will give results without proofs.

Like digital filters defined by a linear, constant-coefficient difference equation, these analog filters are linear, time-invariant (LTI) systems. LTI systems are entirely specified by their response to

the *unit impulse* $\delta(t)$. (See below for an introduction to the unit impulse). This is shown by writing an arbitrary input $x(t)$ as a superposition of (infinitely many) unit impulses:

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t - \tau)d\tau \quad (2.11)$$

This equation means that $x(t)$ can be considered as the limit of a weighted sum of rectangular pulses with infinitesimally small widths. If we define $h(t)$ as the response of an LTI system to the unit impulse, and apply the linearity and time-invariance properties to Eq. 2.11, we can write the filter response $y(t)$ to $x(t)$:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau \triangleq x(t) * h(t) \quad (2.12)$$

where $*$ now denotes the operation of *continuous convolution*. The convolution integral (Eq. 2.12) can be considered as a limit of the convolution sum (Eq. 2.5) when the sampling frequency becomes very high.

Because continuous convolution has the same associativity, commutativity and distributivity properties as discrete convolution, rules for combining analog filters in cascade and in parallel are the same as for digital filters.

A necessary and sufficient condition for an analog filter to be stable is that its impulse response be absolutely integrable:

$$\int_{-\infty}^{\infty} |h(t)|dt < \infty \quad (2.13)$$

All filters realized by connecting linear resistors, capacitors, and inductors are guaranteed to be both stable and causal. Such filters are said to be “physically realizable”.

2.4.2 Singularity functions as operators on functions

A singularity function is the limit in a special sense of a series of functions. One of the most useful singularity function is *the unit impulse*, or *Dirac’s δ function*. The unit impulse is the “limit” of a brief pulse with unit area when its width becomes very small. It is clear that the limit cannot be taken in the usual sense because, if the pulse width tends to zero while its area remains constant, the value of the δ function must become infinite. To make the term “limit” more precise, consider the integral

$$f(x) \cdot \phi(x) \triangleq \int_{-\infty}^{\infty} f(x)\phi(x)dx$$

where $f(x)$ is an “operator function”, $\phi(x)$ is a “test function”, and $f(x) \cdot \phi(x)$ denotes the result of the operation, a real number. If we keep the operator $f(x)$ constant, and vary the test function, we will in general obtain different results for different test functions. If two operators give the same results for all possible test functions, then they can be considered to be equal. Consider specifically the operator $\Pi_{\Delta}(x)$, which is a rectangular pulse with unit area and width Δ

$$\Pi_{\Delta}(x) = \begin{cases} 1/\Delta & \text{if } -\Delta/2 \leq x \leq \Delta/2 \\ 0 & \text{otherwise} \end{cases}$$

Let us apply this operator to a test function $\phi(x)$

$$\Pi_{\Delta}(x) \cdot \phi(x) = \int_{-\infty}^{\infty} \Pi_{\Delta}(x)\phi(x)dx = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} \phi(x)dx$$

When the pulse width Δ tends to zero, this expression tends to $\phi(0)$:

$$\lim_{\Delta \rightarrow 0} \Pi_{\Delta}(x) \cdot \phi(x) = \phi(0)$$

providing the $\phi(t)$ is continuous at the origin. We define the operator “unit impulse” $\delta(x)$ as the limit of the operator $\Pi_{\Delta}(x)$ when Δ goes to zero. This means that the result of applying $\delta(x)$ to an arbitrary test function is by definition:

$$\int_{-\infty}^{\infty} \delta(x)\phi(x)dx \triangleq \phi(0) \quad (2.14)$$

Strictly speaking, $\delta(x)$ is not a function, but, in the sense of operators, i.e. in the sense of what $\delta(x)$ “does” to test functions, it is the limit of the series of functions $\Pi_{\Delta}(x)$.

2.4.3 Properties of Dirac’s unit impulse

A special case of Eq. 2.14 in which $\phi(x)$ is the unity function yields

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \quad (2.15)$$

This relation justifies the term “unit impulse”. Similarly, we define the delayed impulse $\delta(x - x_0)$ by

$$\int_{-\infty}^{\infty} \delta(x - x_0)\phi(x)dx \triangleq \int_{-\infty}^{\infty} \delta(x)\phi(x + x_0)dx = \phi(x_0) \quad (2.16)$$

This definition is the basis for writing an arbitrary, continuous function $f(x)$ as a “sum” of unit impulses:

$$f(x) = \int_{-\infty}^{\infty} \delta(x - \xi)f(\xi)d\xi = f(x) * \delta(x)$$

Always in the sense of operators, one has

$$f(x)\delta(x - x_0) = f(x_0)\delta(x - x_0)$$

because, for every test function,

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)\phi(x)dx = f(x_0)\phi(x_0) = \int_{-\infty}^{\infty} f(x_0)\delta(x - x_0)\phi(x)dx$$

This definition makes sense only if $f(x)$ is continuous for $x = x_0$. Thus, terms of the form $u(x)\delta(x)$ or $\delta(x)^2$ are mathematically meaningless.

The unit impulse can be considered in an operational sense as the derivative of the unit step function $u(x)$. To see this, consider the integral:

$$I \triangleq \int_{-\infty}^{\infty} u(x)\phi'(x)dx \quad (2.17)$$

where $\phi(x)$ is a test function that vanishes at infinity, and $'$ denotes the derivative. Substituting the definition of the unit step $u(x)$ into Eq. 2.17, we obtain

$$I = \int_0^{\infty} \phi'(x) dx = \phi(\infty) - \phi(0) = -\phi(0) = -\int_{-\infty}^{\infty} \delta(x)\phi(x) dx \quad (2.18)$$

The last equality comes from the definition of the unit impulse in Eq. 2.14. We can also integrate Eq. 2.17 by parts by introducing the formal derivative of the unit step $u(x)$

$$I = \int_{-\infty}^{\infty} u(x)\phi'(x) dx = -\int_{-\infty}^{\infty} u'(x)\phi(x) dx \quad (2.19)$$

Comparing Eqs. 2.18 and 2.19, which hold for all test functions, we conclude that, in an operational sense, $\delta(x)$ is the derivative of $u(x)$.

By a similar line of reasoning, we define the derivative of the unit impulse $\delta'(x)$ by the expression

$$\int_{-\infty}^{\infty} \delta'(x)\phi(x) dx \triangleq -\phi'(0)$$

That this definition makes sense can again be seen by integrating by parts. More generally, the delayed δ' “function” is defined by:

$$\int_{-\infty}^{\infty} \delta'(x - x_0)\phi(x) dx \triangleq -\phi'(x_0)$$

This operator is useful for writing a derivative in the form of a convolution:

$$\phi'(x) = \int_{-\infty}^{\infty} \delta'(x - \xi)\phi(\xi) d\xi = \phi(x) * \delta'(x)$$

2.5 Summary

An important class of digital filters are discrete-time systems defined by linear, constant-coefficient difference equations. Such digital filters are part of the broader class of linear, time-invariant (LTI) systems. LTI systems are completely characterized by their unit-sample response $h[n]$, in the sense that their response $y[n]$ to an arbitrary signal $x[n]$ can be computed by means of the discrete convolution:

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[m]x[n - m]$$

If only a finite number of samples of the unit-sample response are non-zero, the filter is said to have a finite impulse response (FIR). FIR filters are always stable, and their unit-sample response can be found by inspection from the difference equation.

2.6 Further reading

- *Oppenheim and Schaffer*, Chapter 2, Sections 1–5

- *Siebert*, Chapter 7, Chapter 9
- *Karu*, Chapters 8, 10, 11, and 14
- *Oppenheim, Willsky and Nawab*, Chapters 1 and 2