Discrete-Time Systems, LTI Systems, and Discrete-Time Convolution

1. Introduction

In this set of notes, we begin our mathematical treatment of *discrete-time systems*. As shown in Figure 1, a discrete-time system operates or transforms some input sequence $x[n]$ to produce an output sequence $y[n]$. Examples of such systems include audio filters and feedback control systems. Unlike the off-line filtering example in the previous set of notes, these systems produce an output in *real-time* (on-line) as input is fed into the system.

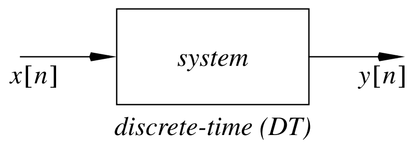


Figure 1

In their most general form, discrete-time systems produce an output $y[n]$ at time index n that is a function of previous output values, as well as past, current and future values of the input:

$$y[n] = f(y[n-1], \ldots, y[n-N], x[n+M], \ldots, x[n+1], x[n], x[n-1], \ldots, x[n-M]), \; N, M > 0 . \quad (1)$$

Below, we first classify discrete-time systems by three important properties: (1) *causality*, (2) *time-invariance* and (3) *linearity*. We then define an important class of discrete-time systems — namely, *Linear, Time-Invariant (LTI)* systems. Next, we define the concept of the *unit-impulse response of a system*, and classify LTI systems into *Finite Impulse Response* (*FIR*) and *Infinite Impulse Response* (*IIR*) systems. Finally, we introduce the *discrete-time convolution* operation, and show that a discrete-time system is completely characterized by its unit-impulse response.

2. Important system characterizations

A. Causality

Definition: A system is said to be *causal* if and only if its output depends only on previous values of the output and current and/or previous values of the input. If a system is not causal, it is said to be *noncausal*.

Examples: Consider the system below:

$$y[n] = \frac{1}{3}(x[n] + x[n+1] + x[n+2]) \quad (2)$$

The system in equation (2) is *noncausal* because the output $y[n]$ depends on future values of the input $x[n]$. For example, the output at $n = 0$ is dependent on the input for $n \in \{0, 1, 2\}$. In Figure 2, we plot a sample output sequence corresponding to a sample input sequence for system (2). Note that for this noncausal system, the output precedes the input. To compute the value of $y[n]$ for any specific n for the plotted input sequence, we simply apply equation (2) above. For example, $y[-1]$ is given by,

$$y[-1] = \frac{1}{3}(x[-1] + x[0] + x[1]) = \frac{1}{3}(0 + 2 + 4) = 2 . \quad (3)$$

Now, consider a second system below:

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \quad (4)$$

The system in equation (4) is *causal* because the output $y[n]$ depends only on present and past values of the input $x[n]$. For example, the output at $n = 0$ is dependent on the input for $n \in \{0, -1, -2\}$. In Figure 3, we plot a sample output sequence corresponding to a sample input sequence for system (4). Note that for this causal system, the output follows the input. To compute the value of $y[n]$ for any specific n for the plotted input sequence, we simply apply equation (4) above. For example, $y[2]$ is given by,

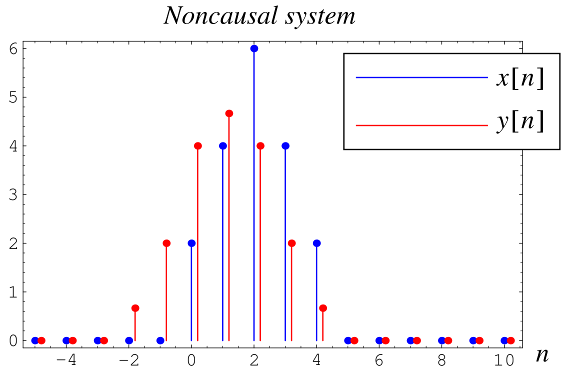


Figure 2

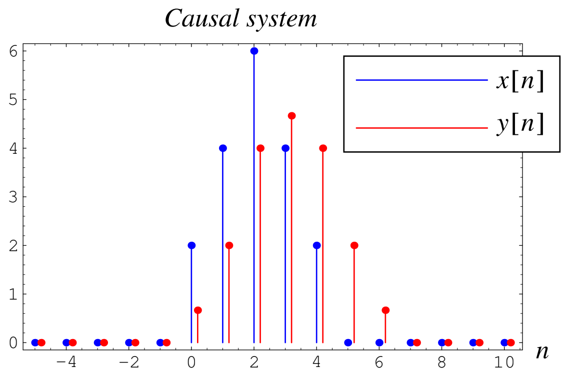


Figure 3

$$y[2] = \frac{1}{3}(x[2] + x[1] + x[0]) = \frac{1}{3}(6 + 4 + 2) = 4 . \tag{5}$$

In general, the class of *causal* discrete-time systems is given by,

$$y[n] = f(y[n-1], \ldots, y[n-N], x[n], x[n-1], \ldots, x[n-M]) \text{ , } N, M > 0 . \tag{6}$$

B. Time invariance

<u>Definition</u>: A system is said to be *time-invariant* if and only if,

$$x[n] \Rightarrow y[n] \text{ implies } x[n-n_0] \Rightarrow y[n-n_0] . \tag{7}$$

That is, if a system produces output $y[n]$ for input $x[n]$, a time-invariant system will produce the time-delayed output $y[n-n_0]$ for the time-delayed input $x[n-n_0]$. If a system is not time-invariant, it is said to be *time-variant*.

Another way to look at time-invariance is depicted in Figure 4 below. A time-invariant system will produce the same output regardless of whether a time delay precedes or follows the system. In Figure 4, that means that for a time-invariant system,

$$w[n] = y[n-n_0] . \tag{8}$$

<u>Examples</u>: Consider the system below:

$$y[n] = x[n]^2 \tag{9}$$

The system in equation (9) is *time-invariant*. We will show this by applying the procedure depicted in Figure 4. First, we follow the top path in the diagram:

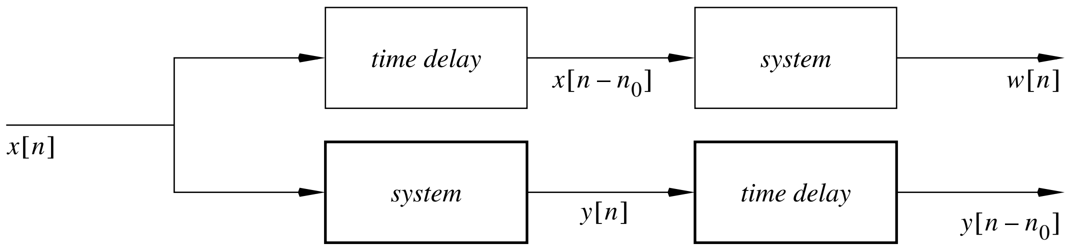


Figure 4

$$w[n] = x[n-n_0]^2 \tag{10}$$

Note that we simply replaced $x[n]$ with $x[n-n_0]$ in equation (9) to produce $w[n]$. Next, we follow the bottom path in the diagram:

$$y[n-n_0] = x[n-n_0]^2 \tag{11}$$

Note that in this case, we first compute $y[n]$ [equation (9)] and then replace n with $n-n_0$. Since (10) and (11) are equivalent, system (9) is *time-invariant*.

Next, consider the system below:

$$y[n] = n\,x[n] \tag{12}$$

Let us again compute $w[n]$ and $y[n-n_0]$:

$$w[n] = n\,x[n-n_0] \tag{13}$$

$$y[n-n_0] = (n-n_0)x[n-n_0] \tag{14}$$

To generate $w[n]$, we substituted $x[n-n_0]$ for $x[n]$ in equation (12); to generate $y[n-n_0]$ we substituted $n-n_0$ for n. Note that since $w[n]$ and $y[n-n_0]$ produce different results, system (12) is *time-variant*.

Next, consider the system below:

$$y[n] = x[-n] \tag{15}$$

Let us again compute $w[n]$ and $y[n-n_0]$:

$$w[n] = x[-n-n_0] \tag{16}$$

$$y[n-n_0] = x[-(n-n_0)] = x[-n+n_0] \tag{17}$$

To generate $w[n]$, we substituted $x[-n-n_0]$ for $x[-n]$ in equation (15); to generate $y[n-n_0]$ we substituted $n-n_0$ for n. Note that since $w[n]$ and $y[n-n_0]$ produce different results, system (15) is *time-variant*.

Finally, consider the system below:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{18}$$

Let us again compute $w[n]$ and $y[n-n_0]$:

$$w[n] = \sum_{k=0}^{M} b_k x[n-n_0-k] \tag{19}$$

$$y[n-n_0] = \sum_{k=0}^{M} b_k x[n-n_0-k] \tag{20}$$

To generate $w[n]$, we substituted $x[n-n_0-k]$ for $x[n-k]$ in equation (18); to generate $y[n-n_0]$ we substituted $n-n_0$ for n. Note that since $w[n]$ and $y[n-n_0]$ produce the same outcome, system (18) is *time-invariant*. The table below summarizes the above results and gives two more examples.

system	*time invariant?*
$y[n] = x[n]^2$	*yes*
$y[n] = n\,x[n]$	*no*
$y[n] = x[-n]$	*no*
$y[n] = \sum_{k=0}^{M} b_k x[n-k]$	*yes*
$y[n] = \cos(x[n]-x[n-1])$	*yes*
$y[n] = 2^n x[n-1]$	*no*

C. Linearity

Definition: A system is said to be *linear* if and only if,

$$x_1[n] \Rightarrow y_1[n] \text{ and } x_2[n] \Rightarrow y_2[n] \text{ implies } \alpha x_1[n] + \beta x_2[n] \Rightarrow \alpha y_1[n] + \beta y_2[n] \tag{21}$$

for arbitrary scalars α and β.

That is, if a system produces output $y_1[n]$ for input $x_1[n]$, and output $y_2[n]$ for input $x_2[n]$, a linear system will produce the output $\alpha y_1[n] + \beta y_2[n]$ for the input $\alpha x_1[n] + \beta x_2[n]$. If a system is not linear, it is said to be *nonlinear*.

Another way to look at linearity is depicted in Figure 5 below. A linear system will produce the same output regardless of whether the inputs or outputs are summed and scaled. In Figure 5, that means that for a linear system,

$$w[n] = y[n]. \tag{22}$$

Examples: Below, we first consider the same systems as we did for time invariance and then show two additional examples. First, let us consider:

$$y[n] = x[n]^2 \tag{23}$$

The system in equation (23) is *nonlinear*. We will show this by applying the procedure depicted in Figure 5. First, we apply the top diagram in Figure 5:

$$y_1[n] = x_1[n]^2,\ y_2[n] = x_2[n]^2 \tag{24}$$

$$w[n] = \alpha y_1[n] + \beta y_2[n] = \alpha x_1[n]^2 + \beta x_2[n]^2 \tag{25}$$

Next, we follow the bottom diagram in Figure 5:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{26}$$

$$y[n] = x[n]^2 = (\alpha x_1[n] + \beta x_2[n])^2 = \alpha^2 x_1[n]^2 + 2\alpha\beta x_1[n] x_2[n] + \beta x_2[n]^2 \tag{27}$$

Since the results in equations (25) and (27) are different, system (23) is *nonlinear*.

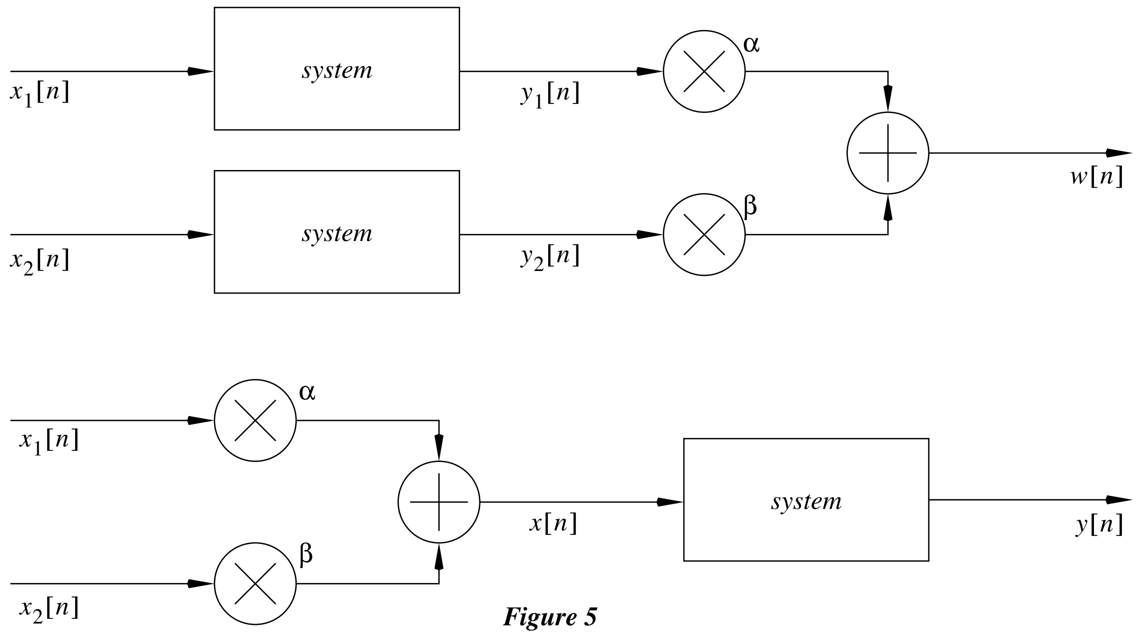


Figure 5

Next, consider the system below:

$$y[n] = n\,x[n] \tag{28}$$

Let us follow the same procedures as in the previous example:

$$y_1[n] = n\,x_1[n],\ y_2[n] = n\,x_2[n] \tag{29}$$

$$w[n] = \alpha y_1[n] + \beta y_2[n] = \alpha n x_1[n] + \beta n x_2[n] \tag{30}$$

Next:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{31}$$

$$y[n] = n\,x[n] = n(\alpha x_1[n] + \beta x_2[n]) = \alpha n x_1[n] + \beta n x_2[n] \tag{32}$$

Since the results in equations (30) and (32) are the same, system (28) is *linear.*

Next, consider the system below:

$$y[n] = x[-n] \tag{33}$$

Let us follow the same procedures as in the previous example:

$$y_1[n] = x_1[-n],\ y_2[n] = x_2[-n] \tag{34}$$

$$w[n] = \alpha y_1[n] + \beta y_2[n] = \alpha x_1[-n] + \beta x_2[-n] \tag{35}$$

Next:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{36}$$

$$y[n] = x[-n] = \alpha x_1[-n] + \beta x_2[-n] \tag{37}$$

Since the results in equations (35) and (37) are the same, system (33) is *linear.*

Next, consider the system below:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{38}$$

Let us follow the same procedures as in the previous example:

$$y_1[n] = \sum_{k=0}^{M} b_k x_1[n-k]\,,\ y_2[n] = \sum_{k=0}^{M} b_k x_2[n-k] \tag{39}$$

$$\begin{aligned} w[n] &= \alpha y_1[n] + \beta y_2[n] = \alpha \sum_{k=0}^{M} b_k x_1[n-k] + \beta \sum_{k=0}^{M} b_k x_2[n-k] \\ &= \sum_{k=0}^{M} b_k(\alpha x_1[n-k] + \beta x_2[n-k]) \end{aligned} \tag{40}$$

Next:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{41}$$

$$y[n] = \sum_{k=0}^{M} b_k(\alpha x_1[n-k] + \beta x_2[n-k]) \tag{42}$$

Since the results in equations (40) and (42) are the same, system (38) is *linear*.

Below, we consider two additional systems. First, let us consider:

$$y[n] = 2x[n] + (1/2)^n x[n-1] + 3x[n-5] \tag{43}$$

Again, we follow the same procedure as before:

$$y_1[n] = 2x_1[n] + (1/2)^n x_1[n-1] + 3x_1[n-5] \tag{44}$$

$$y_2[n] = 2x_2[n] + (1/2)^n x_2[n-1] + 3x_2[n-5] \tag{45}$$

$$\begin{aligned} w[n] &= \alpha y_1[n] + \beta y_2[n] \\ &= \alpha(2x_1[n] + (1/2)^n x_1[n-1] + 3x_1[n-5]) + \beta(2x_2[n] + (1/2)^n x_2[n-1] + 3x_2[n-5]) \end{aligned} \tag{46}$$

Next:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{47}$$

$$\begin{aligned} y[n] &= 2x[n] + (1/2)^n x[n-1] + 3x[n-5] \\ &= 2(\alpha x_1[n] + \beta x_2[n]) + (1/2)^n(\alpha x_1[n-1] + \beta x_2[n-1]) + 3(\alpha x_1[n-5] + \beta x_2[n-5]) \\ &= \alpha(2x_1[n] + (1/2)^n x_1[n-1] + 3x_1[n-5]) + \beta(2x_2[n] + (1/2)^n x_2[n-1] + 3x_2[n-5]) \end{aligned} \tag{48}$$

Since the results in equations (46) and (48) are the same, system (43) is *linear*.

Finally, we consider the system below:

$$y[n] = 3x[n] + 1 \tag{49}$$

Again, we follow the same procedure as before:

$$y_1[n] = 3x_1[n] + 1 \tag{50}$$

$$y_2[n] = 3x_2[n] + 1 \tag{51}$$

$$\begin{aligned} w[n] &= \alpha y_1[n] + \beta y_2[n] \\ &= \alpha(3x_1[n] + 1) + \beta(3x_2[n] + 1) \\ &= 3\alpha x_1[n] + 3\beta x_2[n] + (\alpha + \beta) \end{aligned} \tag{52}$$

Next:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \tag{53}$$

$$\begin{aligned} y[n] &= 3x[n] + 1 \\ &= 3(\alpha x_1[n] + \beta x_2[n]) + 1 \\ &= 3\alpha x_1[n] + 3\beta x_2[n] + 1 \end{aligned} \tag{54}$$

Since the results in equations (52) and (54) are different, system (49) is *nonlinear.* Note that in this case, although the system looks linear, the constant term makes the system nonlinear. The table below summarizes the above examples.

system	*linear?*
$y[n] = x[n]^2$	*no*
$y[n] = n\,x[n]$	*yes*
$y[n] = x[-n]$	*yes*
$y[n] = \sum_{k=0}^{M} b_k x[n-k]$	*yes*
$y[n] = 2x[n] + (1/2)^n x[n-1] + 3x[n-5]$	*yes*
$y[n] = 3x[n] + 1$	*no*

3. Linear, Time-Invariant systems

The class of *causal* discrete-time *Linear, Time-Invariant (LTI)* systems can be described by the following difference equation:

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k] \tag{55}$$

for constant coefficients a_l and b_k.

In this course, we will confine our study to LTI systems. There are at least two important reasons for this. First, LTI systems lend themselves to analysis that more general systems do not. We will see an extremely important case of this in the next section, where we will characterize an LTI system entirely by its impulse response; this same analysis is not possible, in general, for more general discrete-time systems. Second, all discrete-time systems can be approximated as LTI systems over short time constants. Nevertheless, it is important to realize that many systems in real life may not be, strictly speaking, LTI systems, but can only be approximated as such.

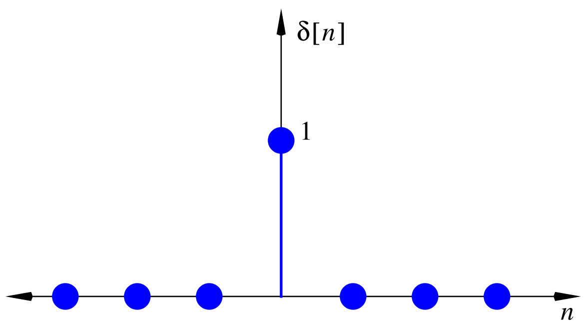
4. Impulse response and discrete-time convolution

A. Impulse response

The unit impulse response of a system is defined as the output $h[n]$ of a system in response to a unit impulse $\delta[n]$, where,

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (56)$$

is plotted in Figure 6 below.



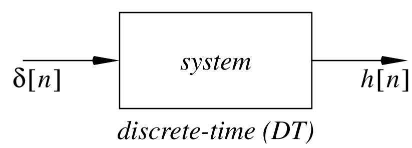


Figure 6

Note that for causal systems,

$$h[n] = 0, \; n < 0. \quad (57)$$

That is, a causal system is at rest prior to the arrival of an impulse at $n = 0$.

B. Classification of LTI systems by impulse response

Previously, we have classified causal discrete-time LTI systems into two broad categories: (1) *non-recursive* and (2) *recursive*. Recall that non-recursive systems are given by the following difference equation:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]. \quad (58)$$

Note that non-recursive systems do not refer to previous values of the output, while recursive systems,

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k] \quad (59)$$

are not only a function of the input to the system, but previous outputs as well. Given our definition of the unit-impulse response $h[n]$ above, we will now label these two categories of LTI systems by their impulse response characteristics. Non-recursive systems in (58) are also known as *Finite Impulse Response* (*FIR*) systems, while recursive systems in (59) are also known as *Infinite Impulse Response* (*IIR*) systems. These names indicate that non-recursive systems have an impulse response that is finite in length, while recursive systems have an impulse response that is infinite in length. Let us consider one specific example of each type of system. In Figure 7, we plot the impulse responses of the following two systems:

$$y[n] = \frac{1}{3}x[n] + \frac{1}{3}x[n-1] + \frac{1}{3}x[n-2] \text{ (FIR example)} \quad (60)$$

$$y[n] = (9/10)y[n-1] + x[n] \text{ (IIR example).} \quad (61)$$

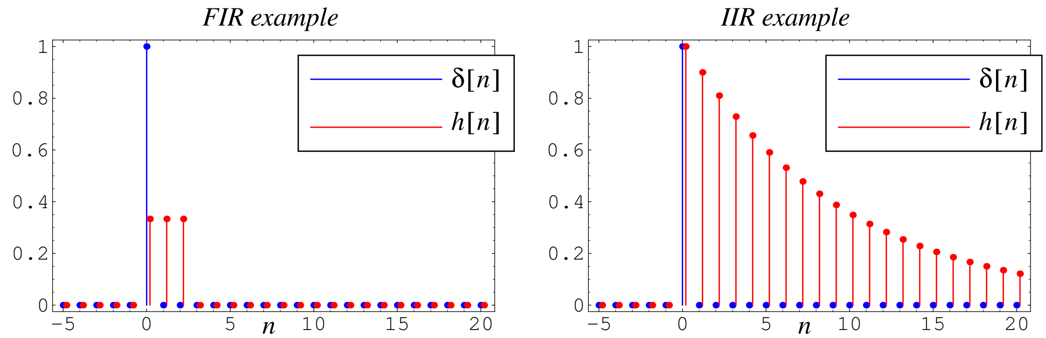


Figure 7

Note that the FIR system's impulse response $h[n]$ is limited in time, while the IIR system's impulse response $h[n]$ only goes to zero as $n \to \infty$; that is the IIR system's impulse response is infinite in length.

C. Discrete-time convolution

A truly remarkable fact that applies to LTI systems is that such systems can be *completely characterized* by their impulse response. That is, once we know $h[n]$ for a given system, we can compute the output $y[n]$ for that system for *any* input sequence $x[n]$. This extremely important fact follows from the fact that we can express any input sequence $x[n]$ as the sum of weighted and time-shifted impulse functions:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{62}$$

For example, the discrete-time signal $x[n]$ in Figure 8 below can be written as:

$$x[n] = \delta[n+1] + 2\delta[n] + 2\delta[n-1] - \delta[n-2]. \tag{63}$$

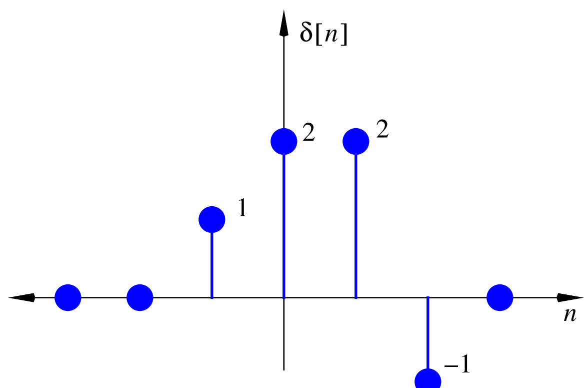


Figure 8

Now, let us assume that the response of an LTI system to a unit impulse $\delta[n]$ is given by $h[n]$:

$$\delta[n] \Rightarrow h[n] \tag{64}$$

By time-invariance of an LTI system, we know that:

$$\delta[n-k] \Rightarrow h[n-k]. \tag{65}$$

That is, a time-shifted impulse $\delta[n-k]$ at the input will result in a time-shifted impulse response $h[n-k]$. By linearity of an LTI system, we know that:

$$x[k]\delta[n-k] \Rightarrow x[k]h[n-k] \tag{66}$$

That is, a weighted, time-shifted impulse results in a weighted, time-shifted impulse response. Finally, the sum of weighted, time-shifted impulses results in the sum of weighted, time-shifted impulse responses:

$$\sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \Rightarrow \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{67}$$

Hence, the output $y[n]$ corresponding to an input $x[n]$ for an LTI system can be written as:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{68}$$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{69}$$

Equation (69) represents the *discrete-time convolution* of the input sequence $x[n]$ with the impulse response $h[n]$. Note that the discrete-time convolution is entirely defined by the impulse response of the system and the input sequence to the system. The concept of discrete-time convolution is so important that we have a special notation for it:

$$y[n] = x[n] * h[n] \tag{70}$$

where the $*$ operator denotes the convolution of $x[n]$ and $h[n]$. Note that equation (70) is short-hand notation for equation (69).

5. Important convolution properties

In this section, we list and prove some of the important properties of the convolution operator.

A. Commutative property

For any two discrete-time sequences $x[n]$ and $y[n]$,[1] the *commutative* property holds:

$$x[n] * y[n] = y[n] * x[n] \tag{71}$$

Proof:

$$x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[k]y[n-k] \tag{72}$$

$$y[n] * x[n] = \sum_{k=-\infty}^{\infty} y[k]x[n-k] \tag{73}$$

In equation (73), let us make the substitution $k = n - l$ and sum over the new variable l

$$y[n] * x[n] = \sum_{l=-\infty}^{\infty} y[n-l]x[l] = \sum_{l=-\infty}^{\infty} x[l]y[n-l] \tag{74}$$

Changing the index in equation (74) from l back to k:

$$y[n] * x[n] = \sum_{k=-\infty}^{\infty} x[k]y[n-k] = x[n] * y[n] \tag{75}$$

and hence the proof is complete.

1. *Note that here we use $x[n]$ and $y[n]$ to denote any discrete-time sequences, not necessarily an input and output sequence, respectively.*

B. Associative property

For any three discrete-time sequences $x[n]$, $y[n]$ and $z[n]$, the *associative* property holds:

$$(x[n] * y[n]) * z[n] = x[n] * (y[n] * z[n]) \tag{76}$$

Proof:

$$(x[n] * y[n]) * z[n] = \sum_{l=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[l-k]\right)z[n-l] \tag{77}$$

$$x[n] * (y[n] * z[n]) = \sum_{l=-\infty}^{\infty} x[l]\left(\sum_{k=-\infty}^{\infty} y[k]z[n-l-k]\right) \tag{78}$$

In equation (78), let us make the substitution $q = l + k$ and sum over the new variable q:

$$\begin{aligned} x[n] * (y[n] * z[n]) &= \sum_{l=-\infty}^{\infty} x[l]\left(\sum_{q=-\infty}^{\infty} y[q-l]z[n-q]\right) \\ &= \sum_{l=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} x[l]y[q-l]z[n-q] \\ &= \sum_{q=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} x[l]y[q-l]z[n-q] \\ &= \sum_{q=-\infty}^{\infty}\left(\sum_{l=-\infty}^{\infty} x[l]y[q-l]\right)z[n-q] \end{aligned} \tag{79}$$

Changing the indices in equation (79) from q to l, and from l to k:

$$x[n] * (y[n] * z[n]) = \sum_{l=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]y[l-k]\right)z[n-l] = (x[n] * y[n]) * z[n] \tag{80}$$

and hence the proof is complete.

C. Multiple systems

Here we consider the impulse response of two LTI systems connected in series, as depicted in Figure 9 below:

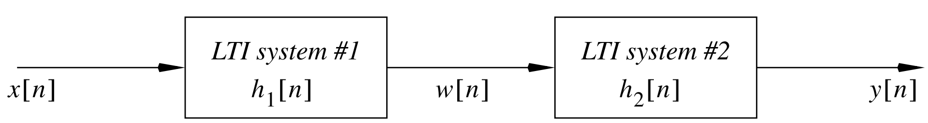


Figure 9

Note that the two LTI systems are assumed to have the impulse responses $h_1[n]$ and $h_2[n]$, respectively. Using the associative property of the convolution operator,

$$w[n] = x[n] * h_1[n] \tag{81}$$

$$y[n] = w[n] * h_2[n] = (x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n]) \tag{82}$$

Letting,

$$h[n] = h_1[n] * h_2[n], \tag{83}$$

we can express the output $y[n]$ as,

$$y[n] = x[n] * h[n]. \quad (84)$$

Thus, the impulse response of two LTI systems connected in series is given by the convolution of the individual impulse responses.

D. Convolution with an impulse

For any discrete-time sequence $x[n]$, the following property holds:

$$x[n] * \delta[n-n_0] = x[n-n_0] \quad (85)$$

Proof:

$$x[n] * \delta[n-n_0] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-n_0-k] \quad (86)$$

Note that $\delta[n-n_0-k]$ is nonzero only for $k = n-n_0$. Therefore, we can rewrite equation (86) as:

$$x[n] * \delta[n-n_0] = x[n-n_0] \quad (87)$$

and hence the proof is complete.

6. Simple convolution example

Below we show two approaches to computing the convolution sum for the sample input sequence $x[n]$ and finite impulse response function $h[n]$ plotted in Figure 10 below.

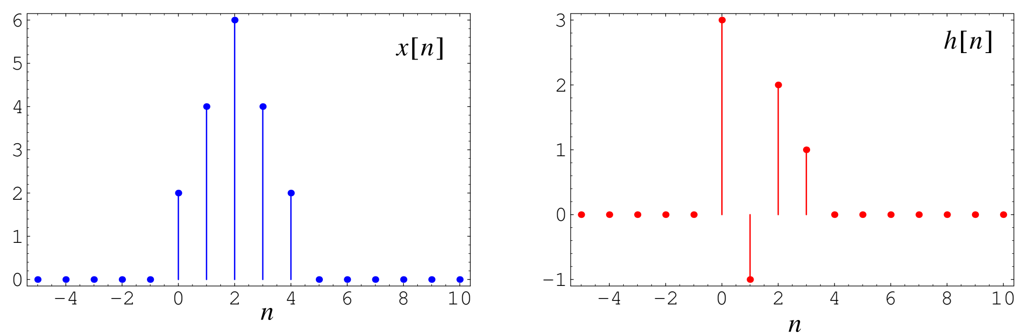


Figure 10

A. Approach #1

In this first approach, we use equation (69) to compute the output $y[n]$:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \quad (88)$$

Let us denote $[3, -1, 2, 1]$ as the *convolution mask* (vector), consisting of the nonzero terms of the impulse response. In order to compute $y[n]$, we first reverse the order of the convolution mask to give the vector $[1, 2, -1, 3]$. We then slide this reversed convolution mask along the input sequence $x[n]$ and take the dot product of the reversed convolution mask and that part of the input sequence $x[n]$ overlapping the reversed convolution mask. This process is illustrated in Table 1 for $n \in \{0, 1, \ldots, 7, 8\}$.Note that the dot product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is given by,

$$\mathbf{a} = [a_1, a_2, \ldots, a_l] \quad (89)$$

n	k	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
	$x[k]$	0	0	0	0	2	4	6	4	2	0	0	0	0
0	$h[0-k]$		1	2	-1	3	$y[0] = [0, 0, 0, 2] \cdot [1, 2, -1, 3] = 6$							
1	$h[1-k]$			1	2	-1	3	$y[1] = [0, 0, 2, 4] \cdot [1, 2, -1, 3] = 10$						
2	$h[2-k]$				1	2	-1	3	$y[2] = [0, 2, 4, 6] \cdot [1, 2, -1, 3] = 18$					
3	$h[3-k]$					1	2	-1	3	$y[3] = 16$ [a]				
4	$h[4-k]$	$y[4] = 18$ [b]					1	2	-1	3				
5	$h[5-k]$	$y[5] = [6, 4, 2, 0] \cdot [1, 2, -1, 3] = 12$						1	2	-1	3			
6	$h[6-k]$	$y[6] = [4, 2, 0, 0] \cdot [1, 2, -1, 3] = 8$							1	2	-1	3		
7	$h[7-k]$	$y[7] = [2, 0, 0, 0] \cdot [1, 2, -1, 3] = 2$								1	2	-1	3	
8	$h[8-k]$	$y[8] = [0, 0, 0, 0] \cdot [1, 2, -1, 3] = 0$									1	2	-1	3

a. Dot product terms not shown for space reasons ($[2, 4, 6, 4] \cdot [1, 2, -1, 3]$ *).*

b. Dot product terms not shown for space reasons ($[4, 6, 4, 2] \cdot [1, 2, -1, 3]$ *).*

Table 1

$$\mathbf{b} = [b_1, b_2, \ldots, b_l] \tag{90}$$

$$\mathbf{a} \cdot \mathbf{b} = [a_1, a_2, \ldots, a_l] \cdot [b_1, b_2, \ldots, b_l] = \sum_{k=1}^{l} a_k b_k \tag{91}$$

Figure 11below plots the convolution of $x[n]$ and $h[n]$ from Figure 10 above.

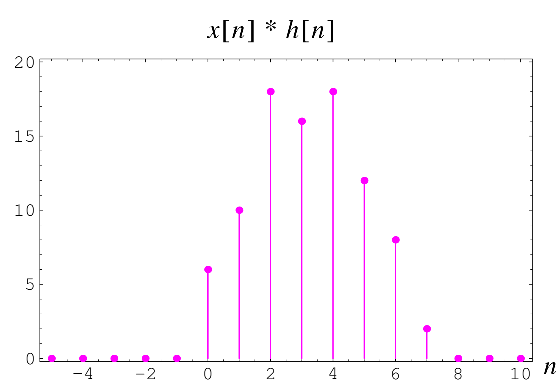


Figure 11

B. Approach #2

In the second approach, we use the commutative property of convolution to express the convolution sum as follows:

$$y[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{92}$$

Table 2 below illustrates how we can use equation (92) to compute the convolution sum. We first write down $x[n]$ and $h[n]$. Then, we multiply each nonzero element of $h[k]$ by $x[n]$, and shift the resulting vectors to the right by k spaces. For example, $k = 0$ corresponds to a zero shift, while $k = 2$ corresponds to a shift of two time units to the right. Finally, to compute $y[n]$, we add the resulting columns. Note that this procedure results in exactly the same result as the previous approach.

n	0	1	2	3	4	5	6	7	8
$x[n]$	2	4	6	4	2				
$h[n]$	3	-1	2	1					
$h[0]x[n-0]$	6	12	18	12	6				
$h[1]x[n-1]$		-2	-4	-6	-4	-2			
$h[2]x[n-2]$			4	8	12	8	4		
$h[3]x[n-3]$				2	4	6	4	2	
$y[n]$	6	10	18	16	18	12	8	2	0

Table 2

7. Conclusion

In this set of notes, we first introduced important properties of discrete-time systems, including causality, time-invariance and linearity. We then defined the class of Linear, Time-Invariant (LTI) systems, and showed how the impulse response completely characterized such systems through the discrete-time convolution operator. Next, we showed important properties of the convolution operator, such as the commutative and associative properties, and illustrated two approaches for computing the convolution sum by hand for finite-impulse response systems.