3. Convolution

1. Convolution
2. Correlation

Textbook: [Smith, ch 6 and 7]

1. Convolution

Using the strategy of **impulse decomposition**, systems are described by a signal called the ***impulse response***.

Convolution relates the three signals of interest:

- the **input signal**,
- the **output signal**, and
- the **impulse response**.

Two different viewpoints of convolution:

- the **input side** algorithm and
- the **output side** algorithm.

1.1 The Delta Function and Impulse Response

An **impulse** is a signal composed of all zeros, except a single nonzero point.

Impulse decomposition :

- provides a way to analyze signals one sample at a time.
- the system's procedure can then be described by a **convolution**.

The **delta function**, $\delta[n]$, is a ***normalized*** **impulse**, that is, sample number zero has a value of one, while all other samples have a value of zero.
For this reason, the delta function is frequently called the **unit impulse**.

The **impulse response** is the signal that exits a system when a delta function (unit impulse) is the input.

If two systems are different in any way, they will have different impulse responses (denoted as $h[n]$).

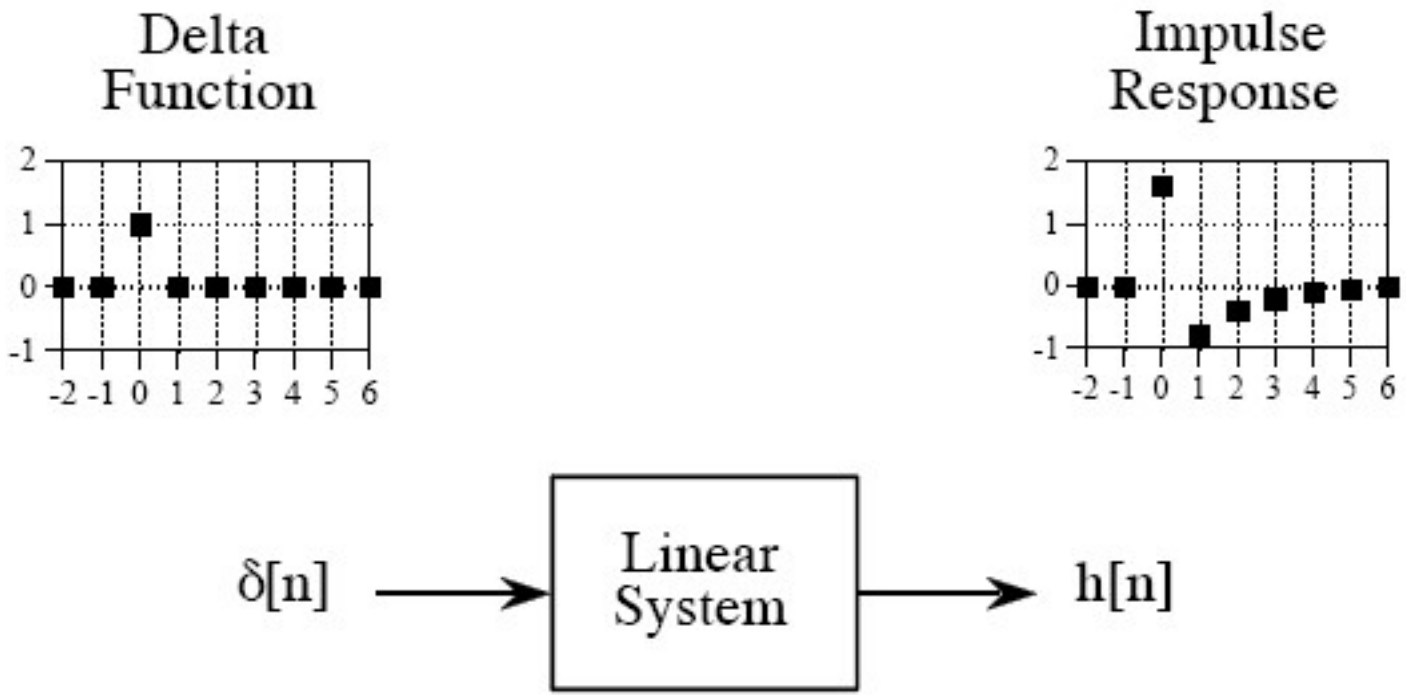

Fig. 1. Definition of ***delta*** *function* and ***impulse response***. The *impulse response* of a linear system, denoted by $h[n]$, is the output of the system when the input is a ***delta*** *function* (normalized impulse)

Any **impulse** can be represented as a ***shifted*** and ***scaled*** **delta** function.

Example

If the input to a LTI system is an impulse, such as

$$a[n] = -3\delta[n-8],$$

then

if $\delta[n]$ results in $h[n]$,

it follows that $-3\delta[n-8]$ results in $-3h[n-8]$.

For an LTI system if we know a system's **impulse response**, you immediately know how it will react to ***any*** impulse.

1.2 Convolution

In linear systems, **convolution** is used to describe the relationship between three signals:

1. the input signal,
2. the **impulse response**, and
3. the output signal.

1. The input signal: a set of impulses - each a scaled and shifted delta function;
2. The output for each impulse - a scaled and shifted version of the impulse response;
3. The overall output signal: by adding these scaled and shifted impulse responses.

The input signal **convolved** with the impulse response is equal to the output signal:

$$y[n] = h[n] \otimes x[n]$$

$$x[n] * h[n] = y[n]$$

Fig. 2. The output signal from a linear system is equal to the input signal *convolved* with the system's impulse response.

If we know a system's impulse response, then we can calculate what the output will be for ***any possible*** **input** signal.

If the system being considered is a ***filter***, the impulse response is called the **filter kernel**, the **convolution kernel**, or simply, the **kernel**.

1) Convolution is used for **low-pass** and **high-pass** filtering

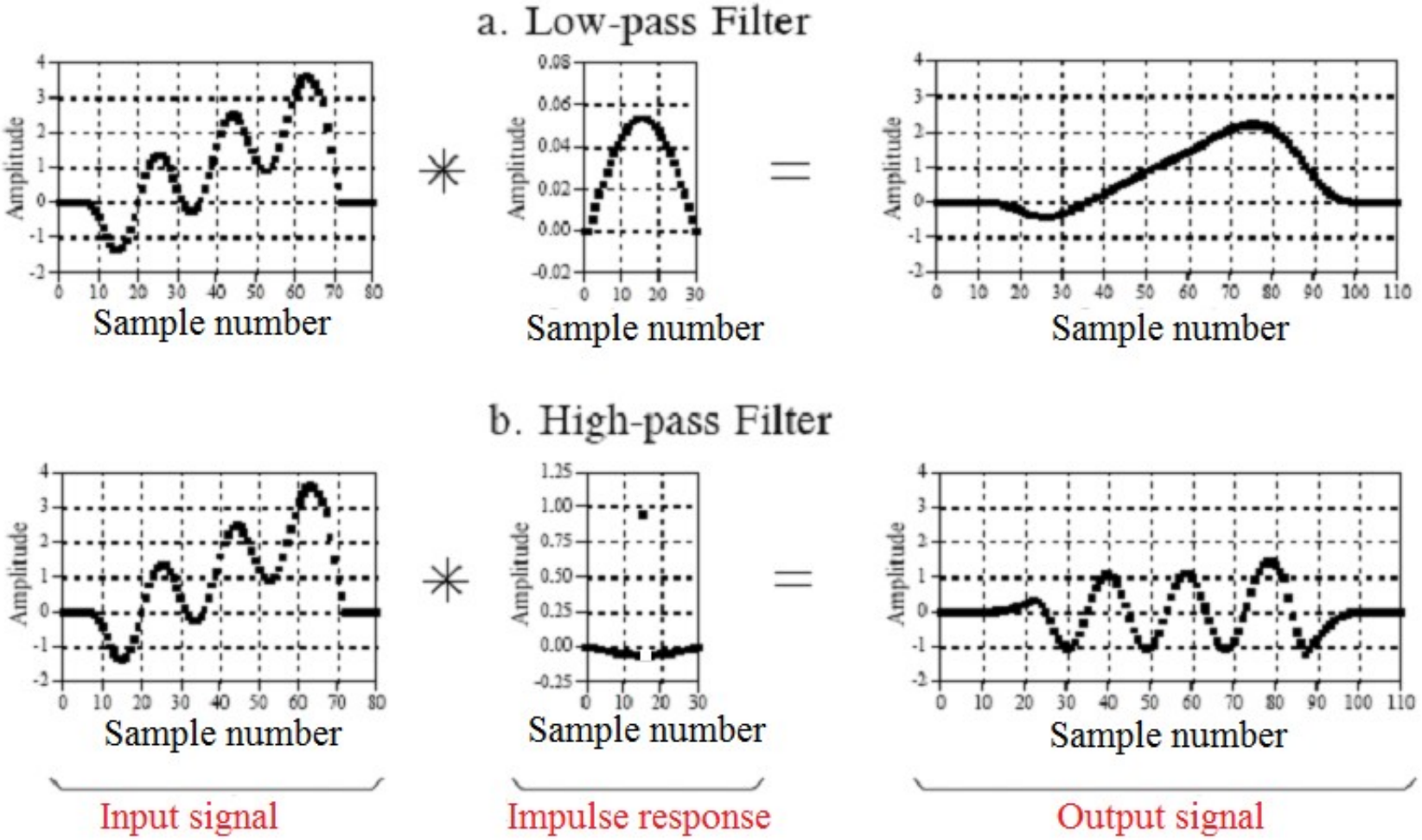

Fig. 3. Examples of low-pass and high-pass filtering using convolution. The input signal is a few cycles of a sine wave plus a slowly rising ramp. These two components are separated by using properly selected impulse responses.

2) **Other** examples of using **convolution**:

- The **inverting attenuator** flips the signal top-for-bottom, and reduces its amplitude (fig. 4.a).
- The **discrete derivative** (the first difference) results in an output signal related to the ***slope*** of the input signal (fig. 4.b).

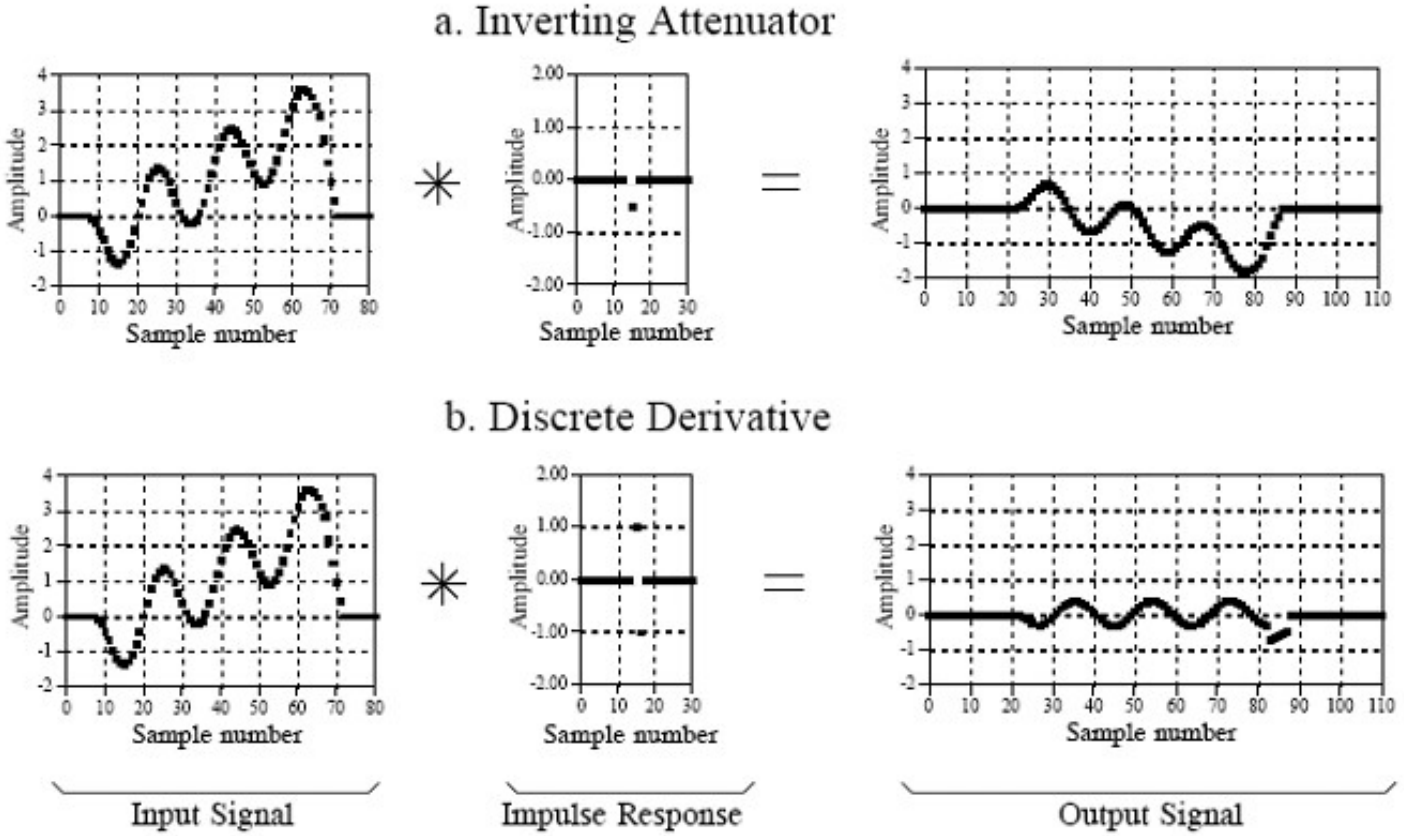

Fig. 4.

3) **Length** of signal

The input signal can be hundreds, thousands, or even millions of samples in length.
The impulse response is usually much shorter: a few points to a few hundred points.

Convolution doesn't restrict **how long** these signals are. It does, however, specify **the length** of the **output signal**.

The **length** of the **output** signal is equal to the length of the input signal, plus the length of the impulse response, minus one.

Example (fig. 5). Signal $x[n]$ is **convolved** with $h[n]$ to produce $y[n]$:

- a 9 point input signal, $x[n]$,
- is passed through a system with a 4 point impulse response, $h[n]$,
- resulting in a (9 + 4 - 1 =) 12 point output signal, $y[n]$.

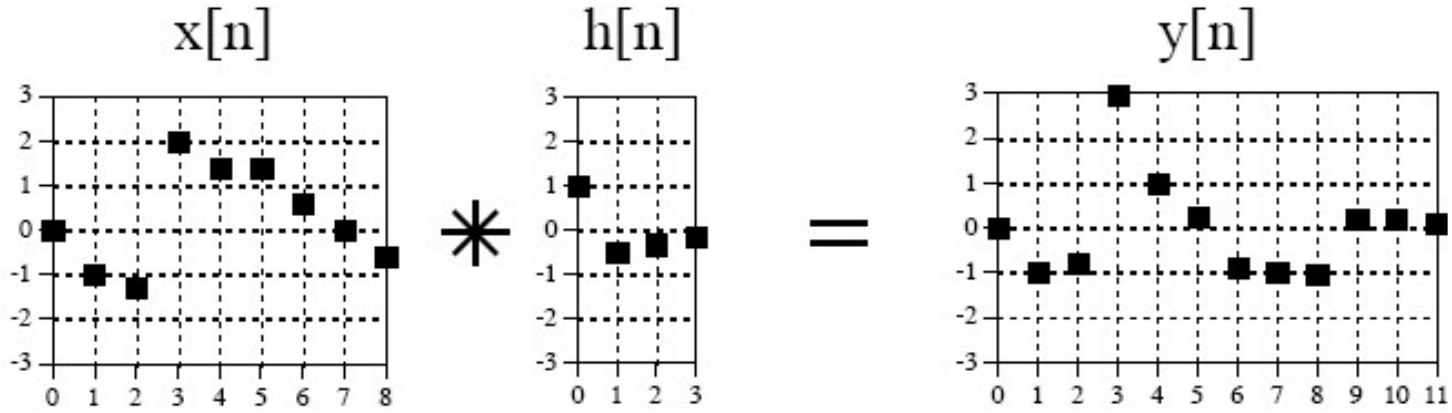

Fig. 5. A 9 point input signal, convolved with a 4 point impulse response, results in a 12 point output signal.

Two algorithms for convolution

Convolution can be understood in two separate ways:

1. From the **viewpoint of the input signal** - analyzing how each sample in the input signal *contributes* to many points in the output signal.
2. From the **viewpoint of the output signal** – examining how each sample in the output signal has *received* information from many points in the input signal.

1.3 The Input Side Algorithm

Example (fig. 6)

Observe the convolution process, given in fig. 5:

- The 9 points of the input signal contribute to nine signals, which are next added to produce the output signal, $y[n]$.
- The processing of a single input point results in a scaled and shifted impulse response that is represented by square markers. The remaining data points in the 9 signals, that are represented by diamonds, are zeros added.

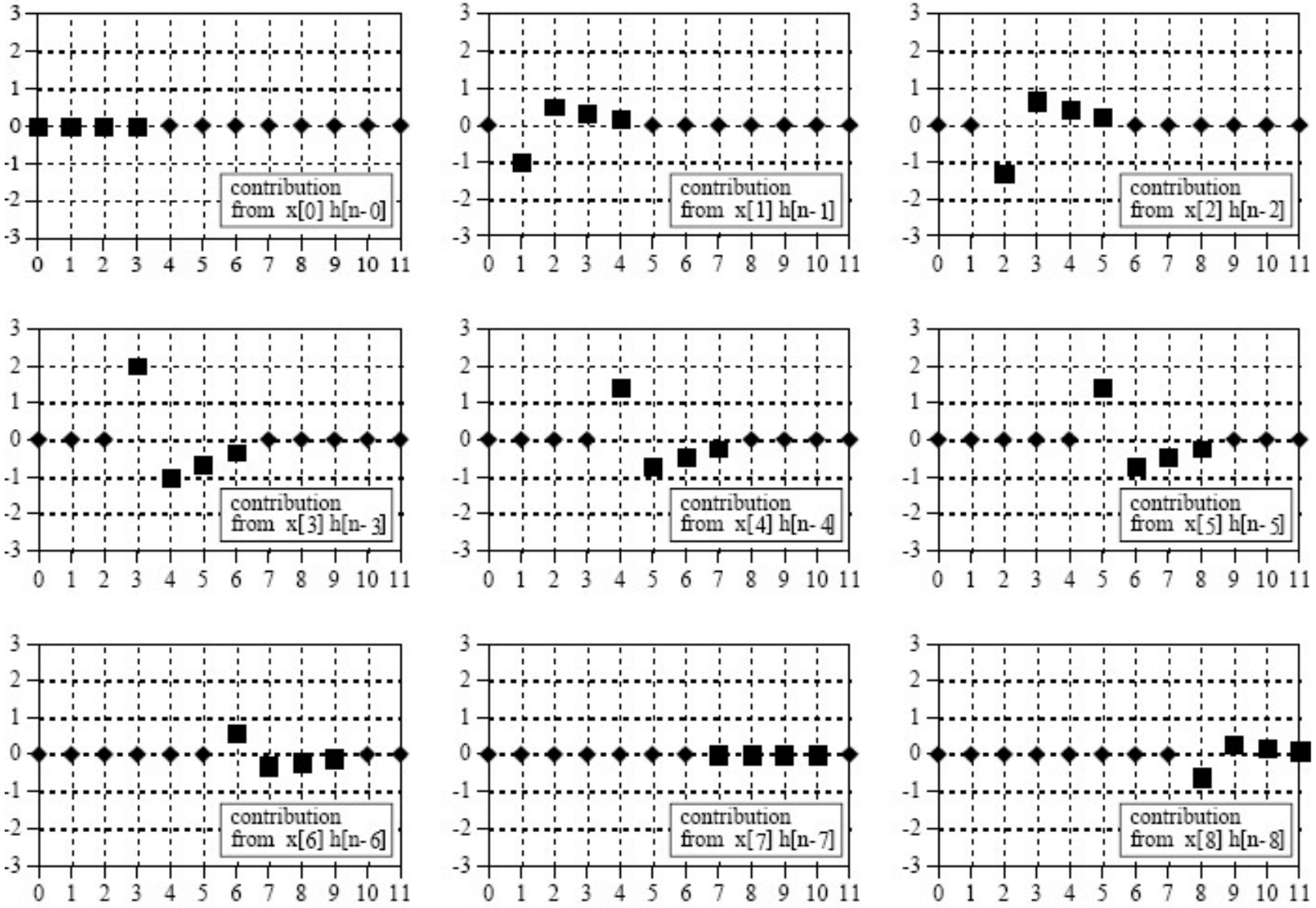

Fig. 6. 9 signals – contributions of single input points

Commutativity

Convolution is *commutative*: $a[n] \otimes b[n] = b[n] \otimes a[n]$.

- The mathematics does not care which is the input signal and which is the impulse response, only that *two signals are convolved with each other*.
- But exchanging the two signals has no physical meaning in system theory. The input signal and impulse response are two totally different things and exchanging them doesn't make sense.

Example (fig. 7)

Swap the signals given in fig. 5: we make $x[n]$ a four point signal and $h[n]$ a nine point signal. The four samples in $x[n]$ result in four shifted and scaled versions of the nine point impulse response.

The output signal in Fig. 7 is ***identical*** to the output signal in Fig. 5.

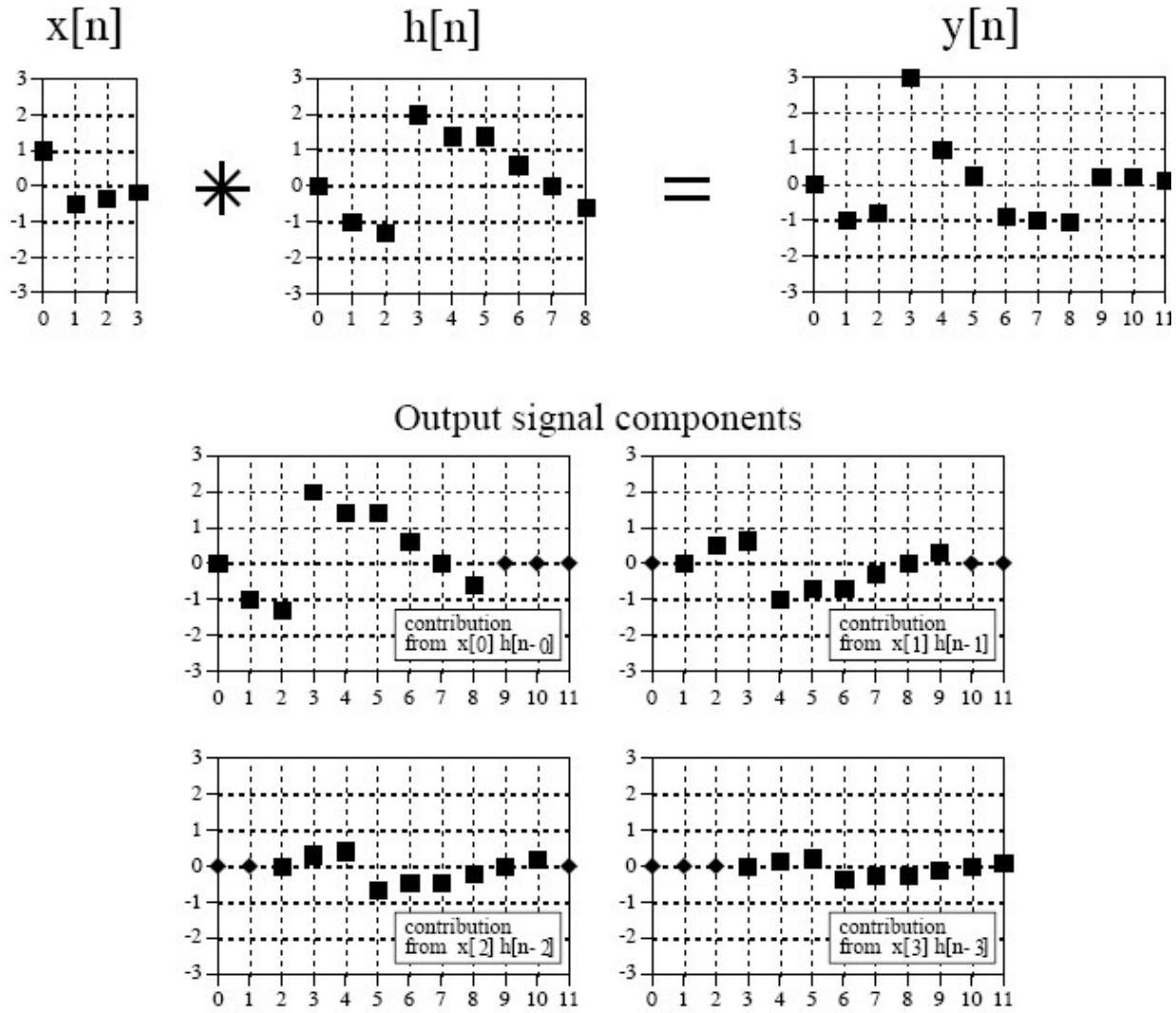

Fig. 7. The waveforms for the input signal and impulse response are exchanged from the example of Fig. 5. Since convolution is commutative, the output signals for the two examples are identical.

1.4 The Output Side Algorithm

Now we are looking at individual samples in the ***output signal***, to find the contributing points from the input.

Example (fig. 8)

The point $y[6]$ in Fig. 5 is equal to the sum of all the sixth points in the nine signal components, shown in Fig. 6.

Five of them have *added* zeros (the diamond markers) at the sixth sample, and can therefore be ignored. Only four of the output components can eventually have a nonzero value in the sixth position. These are the output components generated from the input samples:

$$x[3],\ x[4],\ x[5],\ \text{and}\ x[6].$$

$y[6]$ is determined as:

$$y[6] = x[3]h[3] + x[4]h[2] + x[5]h[1] + x[6]h[0]$$

$$y[n] = \sum_{k=0}^{k=3} x[n-k] \cdot h[k]$$

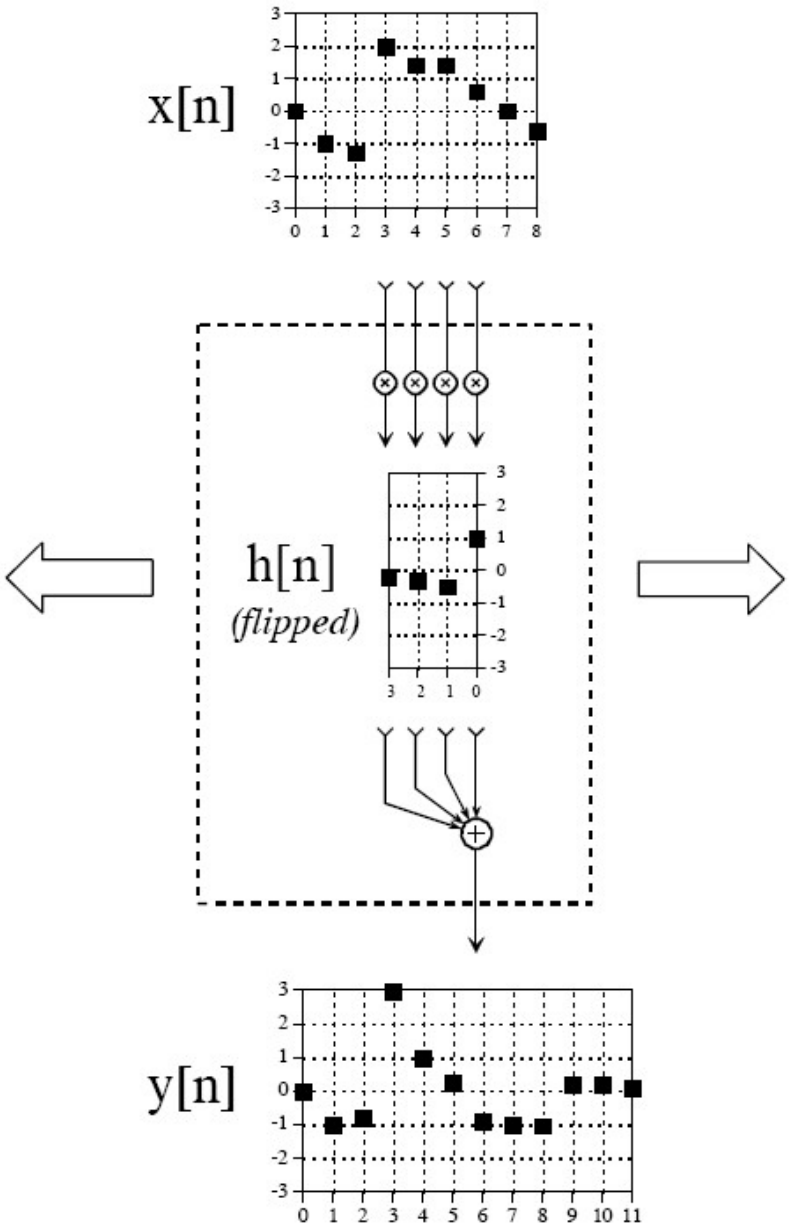

Fig. 8. The convolution machine. In particular it shows how is the output sample six computed.

The **convolution machine**:

- The impulse response is ***flipped left-for-right***. This places sample number zero on the right, and increasingly positive sample numbers running to the left.
- Each point in the output signal is affected by points in the input signal weighted by a ***flipped*** **impulse response**.

Zero-padding of the signal – a way to handle the problem of ***nonexisting*** samples by ***inventing*** the nonexistent samples, i.e. adding zero-valued samples to the ends of the input signal.

Example (fig. 9) In Fig. 9(a), the convolution machine is located fully to the left at $y[0]$, trying to receive input from samples: $x[-3]$, $x[-2]$, $x[-1]$, x[0], but three of these samples: $x[-3]$, $x[-2]$, and $x[-1]$, do not exist. This same dilemma arises in (d) for y[11], with undefined input signal points $x[9]$, $x[10]$, and $x[11]$.

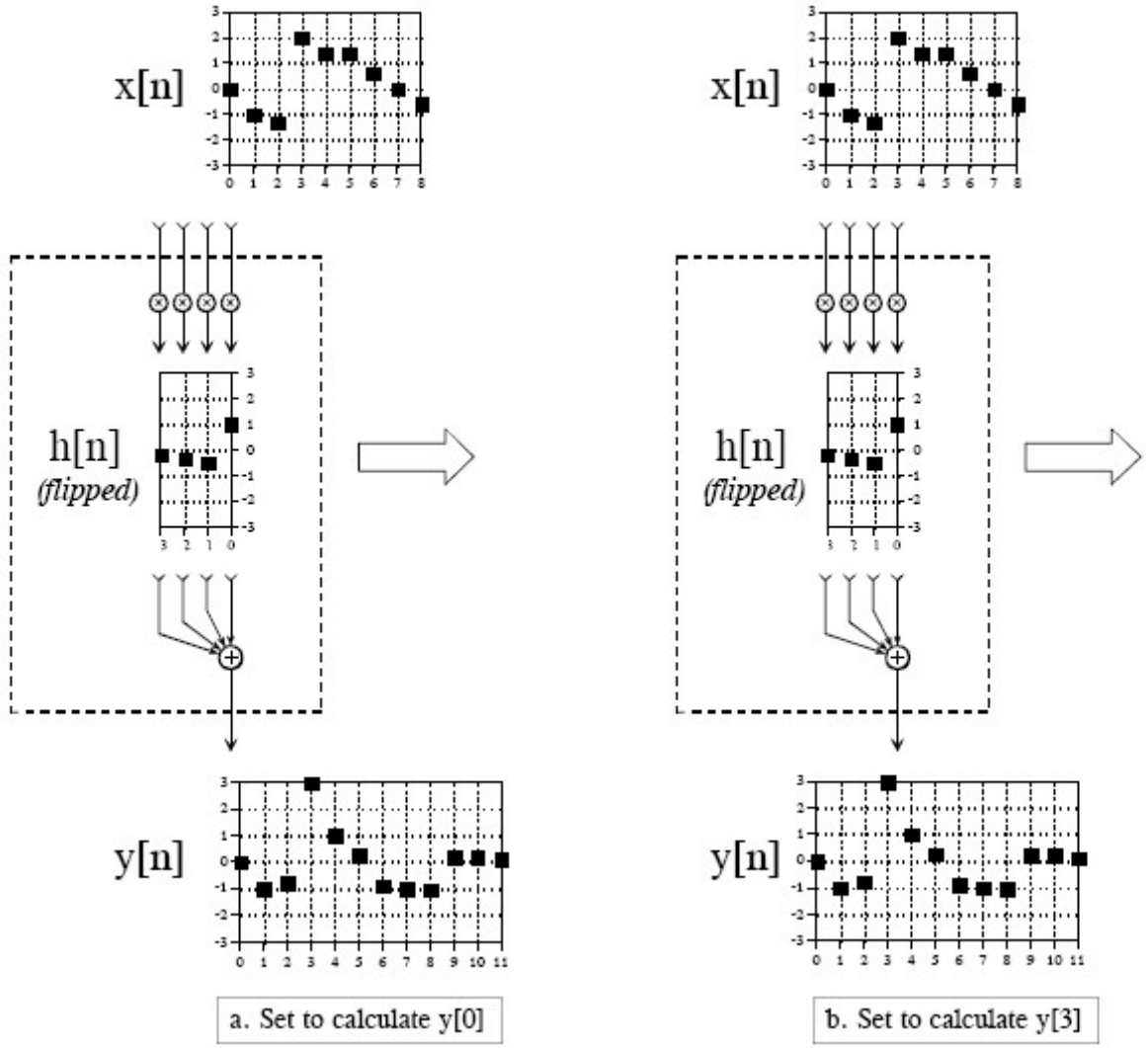

Fig. 9. The convolution machine: (a) through (d) show the convolution machine set to calculate four different output signal samples, y[0], y[3], y[8], and y[11].

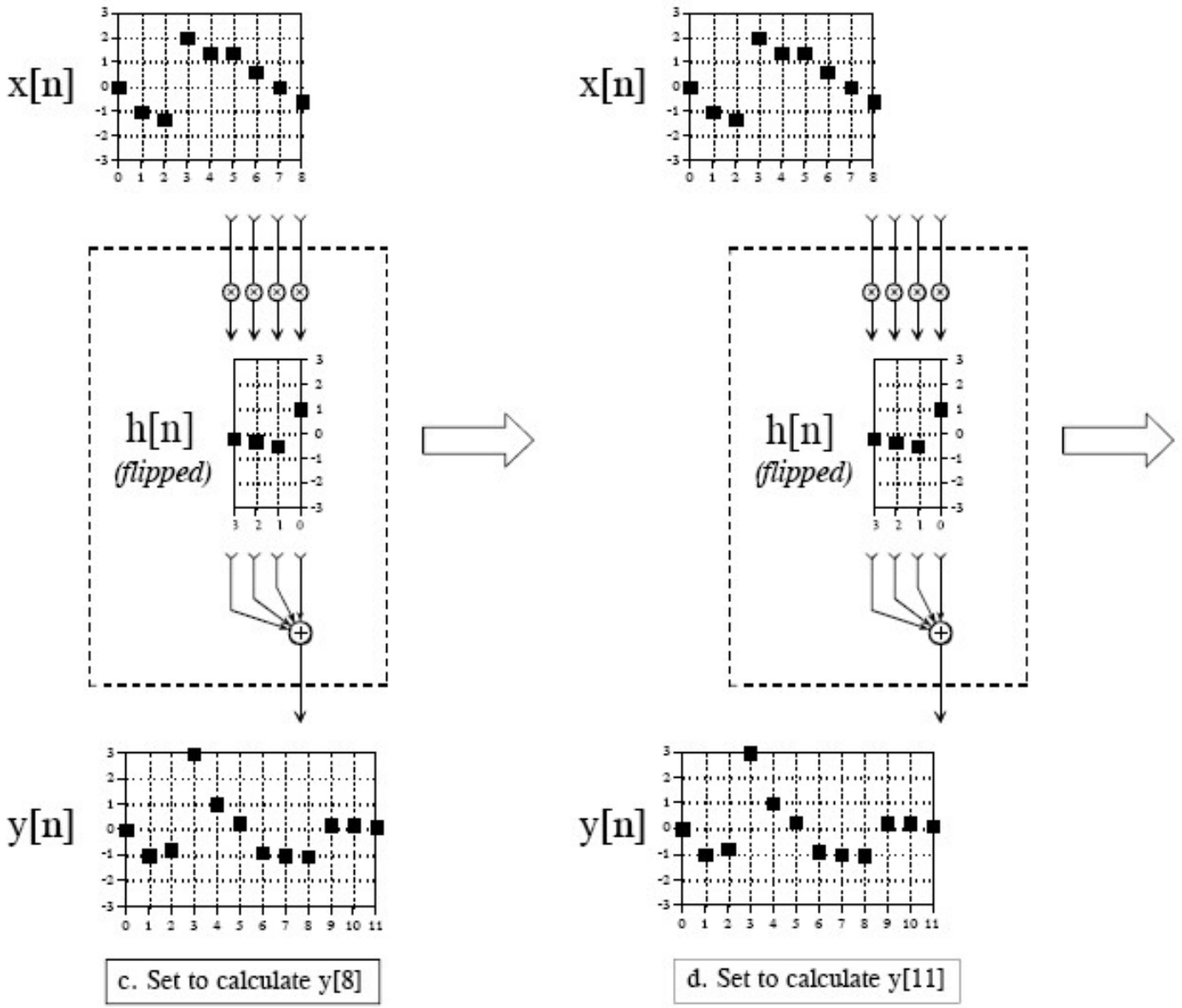

Fig. 9 (continued)

The far left and far right samples in the output signal are based on *incomplete* information, i.e. **the impulse response is not fully explored.**
If the impulse response is *M* points in length, the first *and* last *M*-1 samples in the output signal are based on incomplete information.

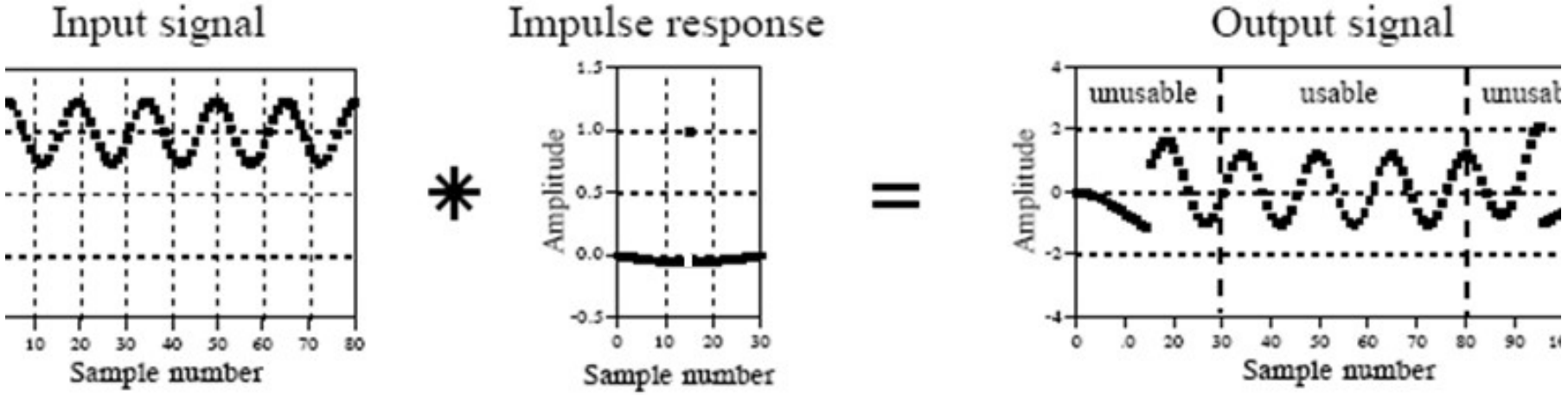

Fig. 10 End effects in convolution. For an *M* point impulse response, the first and last *M*-1 points in the output signal may not be usable (e.g. a high-pass filter used to remove the DC component from the input signal).

As a general rule: the **beginning** and **ending** samples in processed signals will be quite useless.

The standard equation for convolution

If $x[n]$ is an N point signal running from 0 to N-1, and $h[n]$ is an M point signal running from 0 to M-1, the convolution of the two:

$$y[n] = x[n] \otimes h[n],$$

is an $N+M$-1 point signal running from 0 to $N+M$-2, given by:

$$y[i] = \sum_{j=0}^{M-1} h[j]\, x[i-j]$$

This equation is called the **convolution sum**:

- Each point in the output signal is calculated independently of all other points in the output signal.
- The index, *i*, determines which sample in the output signal is being calculated.
- Index *i* corresponds to the left-right position of the convolution machine.
- The index, *j*, is used *inside* of the convolution machine.

1.5 The Sum of Weighted Inputs

Think of the impulse response as a set of **weighting coefficients:**

- each sample in the output signal is equal to a sum of weighted inputs,
- the weighting coefficients do not **need to be restricted** to the ***left side*** of the output sample being calculated.
- Viewing the convolution as a sum of weighted inputs, the weighting coefficients could be chosen ***symmetrically*** around the output sample.

Example

In convolution $y[6]$ is being calculated from: $x[3]$, $x[4]$, $x[5]$, and $x[6]$. For example, $y[6]$ might receive contributions from: $x[4]$, $x[5]$, $x[6]$, $x[7]$, and $x[8]$. The weighing coefficients for these five inputs would be held in:

$$h[2],\ h[1],\ h[0],\ h[-1],\ \text{and}\ h[-2].$$

i.e., the impulse response that corresponds to our selection of symmetrical weighing coefficients requires the use of ***negative indexes***.

Mathematically, there is only one concept here - **convolution**.

1.6 Common Impulse Responses

Delta Function

The simplest impulse response is nothing more that a **delta function**:

- An impulse on the input produces an identical impulse on the output;
- *All* signals are passed through the system ***without change***;
- Convolving any signal with a delta function results in the same signal.

$$x[n] \otimes \delta[n] = x[n]$$

The **delta function** is the **identity for convolution**.
Any signal convolved with a delta function **is left unchanged**.

Scaled delta

A system that **amplifies** or **attenuates** has a *scaled* delta function for an impulse response:

$$x[n] \otimes k\delta[n] = kx[n]$$

Amplification (if $k > 1$) or attenuation (if $k < 1$).

Shifted delta

A **relative shift** between the input and output signals corresponds to an impulse response that is a *shifted* delta function. The variable, s, determines the amount of shift in this equation:

$$x[n] \otimes \delta[n+s] = x[n+s]$$

This could be described as a signal **delay**, or a signal **advance**, depending on the direction of the shift.

Echo

An impulse response composed of a delta function plus a shifted and scaled delta function leads to the output of this system being the input signal plus a delayed version of the input signal, i.e., an ***echo***.

a) Identity. Convolving a signal with the delta function leaves the signal unchanged. This is the goal of systems that transmit or store signals.

b) Amplification & Attenuation. Increasing or decreasing the amplitude of the delta function.

c) Shift. Shifting the delta function produces a corresponding shift between the input and output signals (a *delay* or an *advance*). This impulse response delays the signal by four samples.

d) Echo. A delta function plus a shifted and scaled delta function results in an *echo* being added to the original signal. Here the echo is delayed by four samples and has an amplitude of 60% of the original signal.

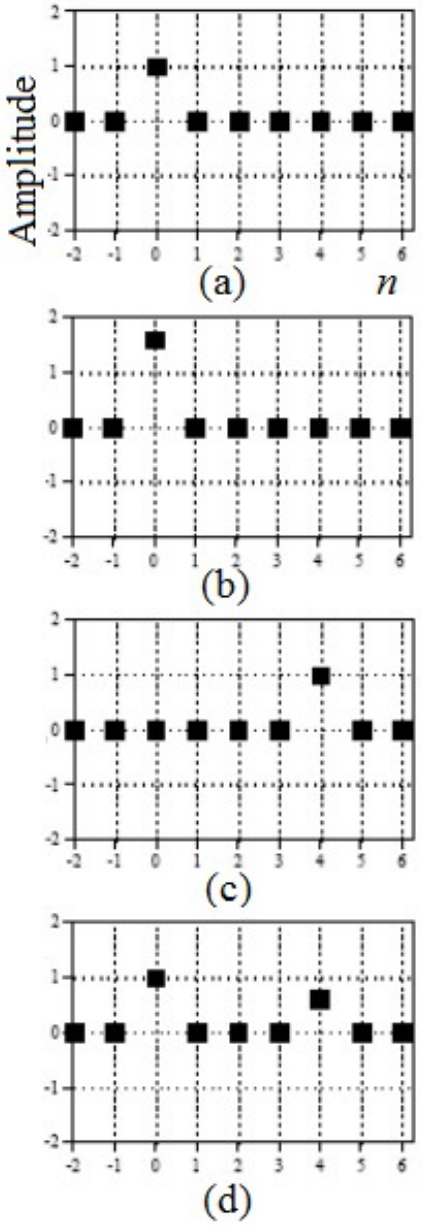

Fig. 11. Simple impulse responses using shifted and scaled delta functions.

Integration and differentiation operations

Convolution can change discrete signals in ways that resemble **integration** and **differentiation**:

- The discrete operation that mimics the ***first derivative*** is called the **first difference**.
- The discrete form of the ***integral*** is called the **running sum**.

It is also common to call them: **discrete derivative** and **discrete integral**.

a) First Difference. This is the discrete version of the *first derivative*: the output signal is the ***slope*** of the input signal.

b) Running Sum. Each sample in the output signal is equal to the sum of all samples in the input signal to the *left*.

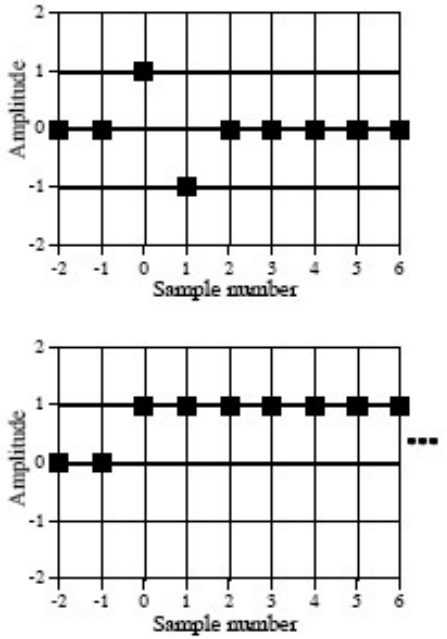

Fig. 12. Impulse responses that mimic calculus operations.

The **running sum** is the inverse operation of the **first difference**.

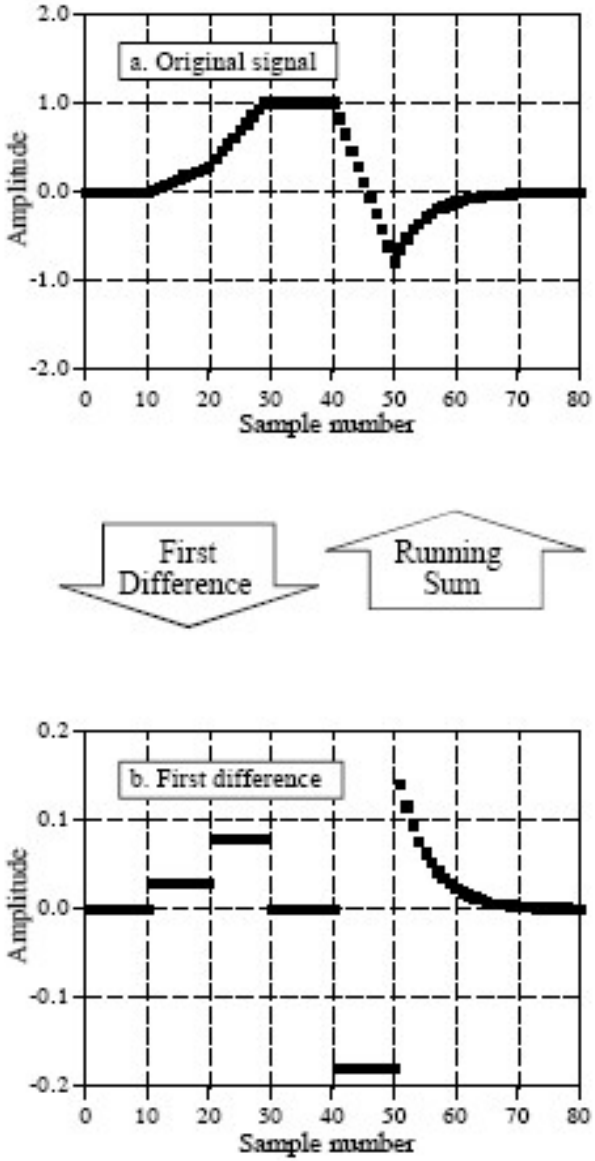

Fig. 13.
Example of calculus-like operations. The signal in (b) is the *first difference* of the signal in (a). The signal in (a) is the *running sum* of the signal in (b).

These impulse responses are simple enough that a **full convolution** program is usually **not needed** to implement them.

Each sample in the output signal is a ***sum of weighted samples*** *from the input*.

Calculation of first difference

- The **first difference** can be calculated:

$$y[n] = x[n] - x[n - 1]$$

In this relation, $x[n]$ is the original signal, and $y[n]$ is the first difference.

- Another common method is to define the slope symmetrically around the point being examined, such as:

$$y[n] = (x[n+1] - x[n-1])/2.$$

Calculation of running sum

To calculate each sample in the **running sum** we shall use the recursion equation:

$$y[n] = x[n] + y[n - 1]$$

In this relation, $x[n]$ is the original signal, and $y[n]$ is the running sum.

2. Correlation

Example

In a radar system the received signal will consist of two parts:
(1) a shifted and scaled version of the transmitted pulse, and
(2) random noise, resulting from interfering radio waves, thermal noise in the electronics, etc.

The **shift** between the transmitted and received pulse is a direct measure of the distance to the object being detected.

The problem solved by **correlation** is:

given a signal of some ***known shape***, what is the best way to determine where (or if) the signal occurs in ***another*** **signal**.

Correlation is a mathematical operation that is very similar to convolution.

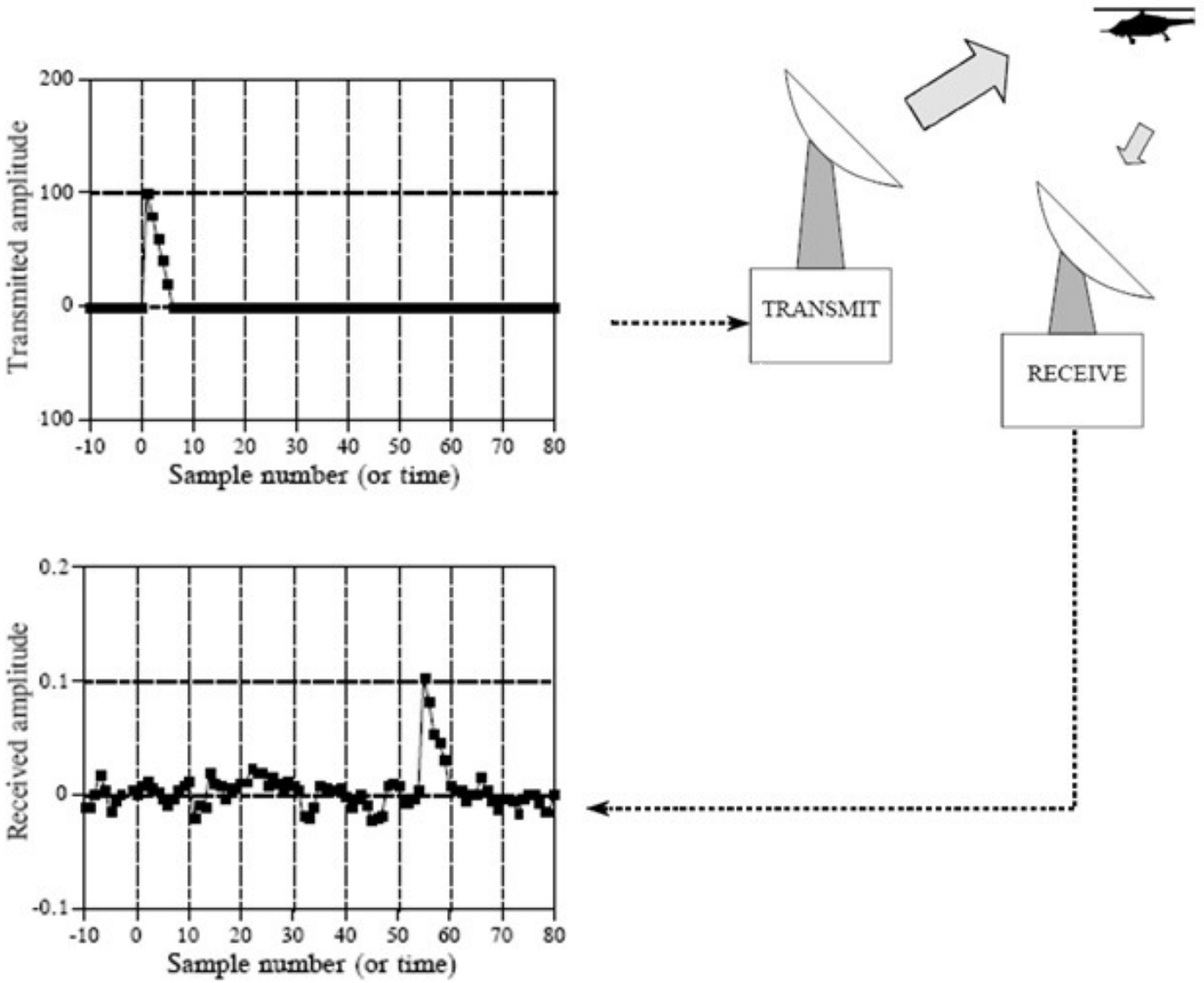

Fig. 23. Correlation, a key element of a radar system. Detection of a known waveform in a noisy signal is the fundamental problem in echo location.

Correlation uses two signals, $x[n]$ and $t[n]$, to produce a third signal $y[n]$:

- The third signal is called the **cross-correlation** of the two input signals.
- If a signal is correlated with *itself*, the resulting signal is instead called the **autocorrelation**.

- The received signal, $x[n]$, is known. The waveform we are looking for, $t[n]$, i.e. the **target** signal, is contained *within* the **correlation machine**.
- $y[n]$ is calculated by moving the correlation machine left or right. The amplitude of each output sample is a measure of how much the received signal *resembles* the target signal, *at that location*.
- A peak will occur in the cross-correlation signal for every target signal that is present in the received signal.

Correlation is the *optimal* technique for detecting a known waveform in *random white noise* - the peak is higher above the noise using correlation than can be produced by any other linear system.

The **correlation** machine and **convolution** machine are **identical**, except that the signal inside of the convolution machine is *flipped* left-for-right, whereas in the correlation machine this flip doesn't take place.

It is possible to represent *correlation* using the same mathematics as *convolution*.
This requires *pre-flipping* one of the two signals being correlated, so that the left-for-right flip inherent in convolution is canceled.

For instance, when $a[n]$ and $b[n]$, are convolved to produce $c[n]$, then:

$$a[n] \otimes b[n] = c\,[n]\ .$$

In comparison, the cross-correlation of $a[n]$ and $b[n]$ can be written:

$$a[n] \otimes b[-n] = c\,[n]\ .$$

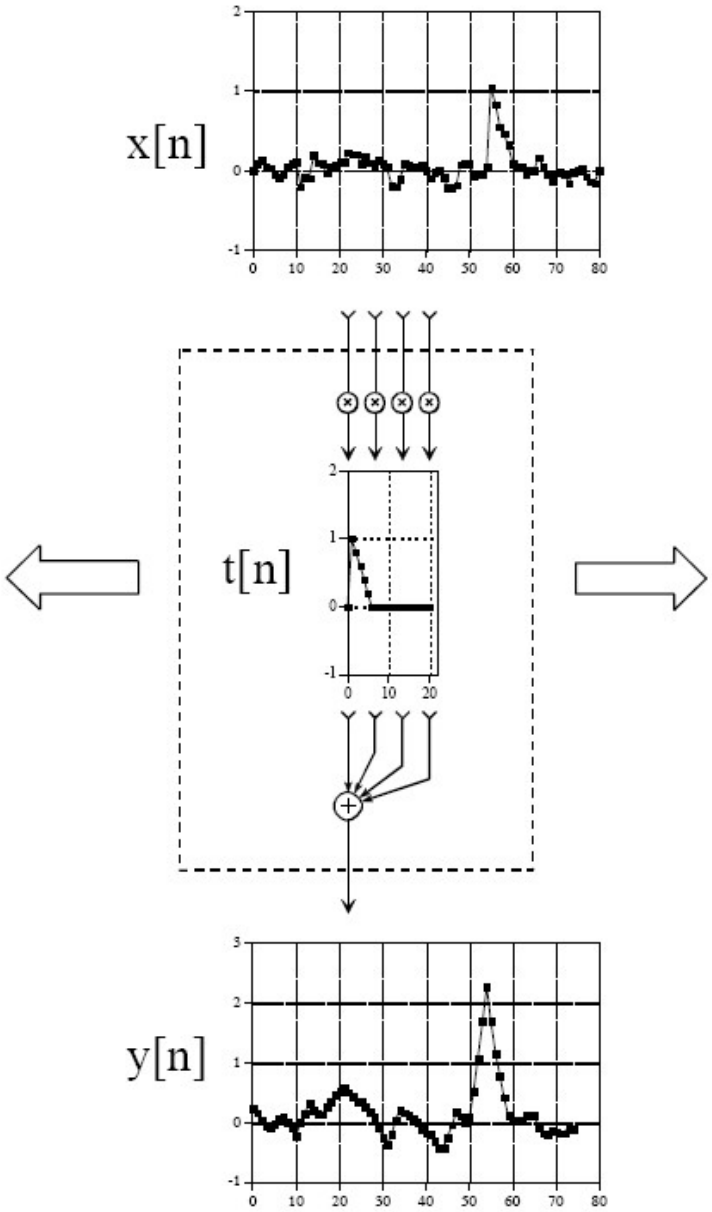

Fig. 24. The correlation machine - $y[n]$ is the cross-correlation of $x[n]$ and $t[n]$.

The indicated samples from $x[n]$ are multiplied by the corresponding samples in $t[n]$, and the products added.

Remark

Convolution and correlation represent **very different** DSP procedures:

- Convolution is the relationship between a system's input signal, output signal, and impulse response.
- Correlation is a way to detect a known waveform in a noisy background.
- The similar mathematics is only a convenient coincidence.

Exercise 3-1

Illustrate the signal convolution operation:

a) from viewpoint of the input signal,

b) from viewpoint of the output signal,

for the following signals:

- input signal

n	0	1	2	3	4	5
$\boldsymbol{x}[n]$	10	4	1	0	-1	-5

- impulse response

n	0	1	2
$\boldsymbol{h}[n]$	3	2	1

Explain the performed decompositions.