Convolution - Discrete time*

Anders Gjendemsjø

Abstract

Time discrete convolution

1 Introduction

The idea of **discrete-time convolution** is exactly the same as that of continuous-time convolution[1]. For this reason, it may be useful to look at both versions to help your understanding of this extremely important concept. Convolution is a very powerful tool in determining a system's output from knowledge of an arbitrary input and the system's impulse response.

It also helpful to see convolution graphically, i.e. by using transparencies or Java Applets. Johns Hopkins University[2] has an excellent Discrete time convolution[3] applet. Using this resource will help understanding this crucial concept.

2 Derivation of the convolution sum

We know that any discrete-time signal can be represented by a summation of scaled and shifted discrete-time impulses, see here[4]. Since we are assuming the system to be linear and time-invariant, it would seem to reason that an input signal comprised of the sum of scaled and shifted impulses would give rise to an output comprised of a sum of scaled and shifted impulse responses. This is exactly what occurs in **convolution**. Below we present a more rigorous and mathematical look at the derivation:

Letting $\mathcal{H}$ be a discrete time LTI system, we start with the folowing equation and work our way down the the convoluation sum.

$$\begin{aligned} y(n) &= \mathcal{H}(x(n)) \\ &= \mathcal{H}\left(\sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)\right) \\ &= \sum_{k=-\infty}^{\infty} \mathcal{H}(x(k)\,\delta(n-k)) \\ &= \sum_{k=-\infty}^{\infty} x(k)\,\mathcal{H}(\delta(n-k)) \\ &= \sum_{k=-\infty}^{\infty} x(k)\,h(n-k) \end{aligned} \tag{1}$$

Let us take a quick look at the steps taken in the above derivation. After our initial equation we rewrite the function $x(n)$ as a sum of the function times the unit impulse. Next, we can move around the $\mathcal{H}$ operator

*Version 1.4: Nov 6, 2003 2:31 pm -0600

†http://creativecommons.org/licenses/by/1.0

[1]"Continuous Time Convolution" <http://cnx.org/content/m10085/latest/>

[2]http://www.jhu.edu

[3]http://www.jhu.edu/signals

[4]"Discrete time signals": Section The unit sample <http://cnx.org/content/m11476/latest/#s3s1>

and the summation because $\mathcal{H}(\cdot)$ is a linear, DT system. Because of this linearity and the fact that $x(k)$ is a constant, we pull the constant out and simply multiply it by $\mathcal{H}(\cdot)$. Finally, we use the fact that $\mathcal{H}(\cdot)$ is time invariant in order to reach our final state - the convolution sum!

Above the summation is taken over all integers. Howerer, in many practical cases either $x(n)$ or $h(n)$ or both are finite, for which case the summations will be limited. The convolution equations are simple tools which, in principle, can be used for all input signals. Following is an example to demonstrate convolution; how it is calculated and how it is interpreted.

2.1 Graphical illustration of convolution properties

A quick graphical example may help in demonstrating why convolution works.

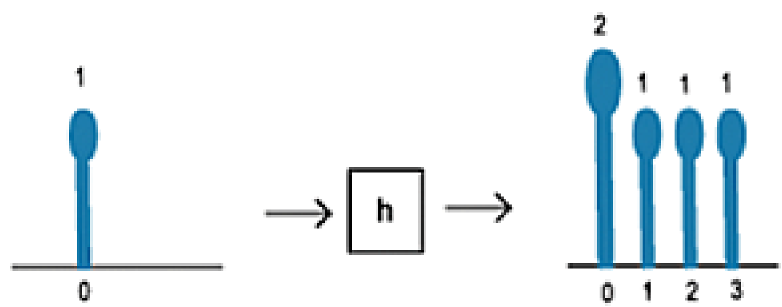

Figure 1: A single impulse input yields the system's impulse response.

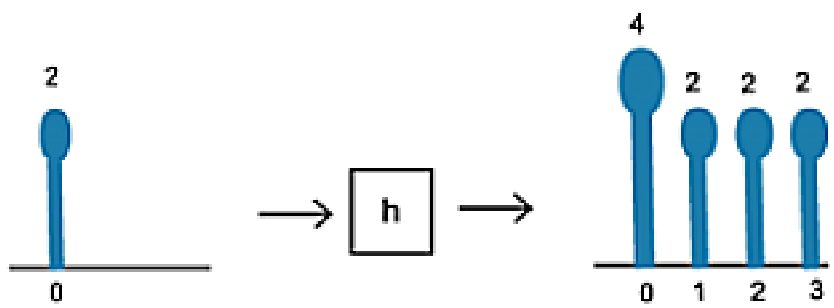

Figure 2: A scaled impulse input yields a scaled response, due to the scaling property of the system's linearity.

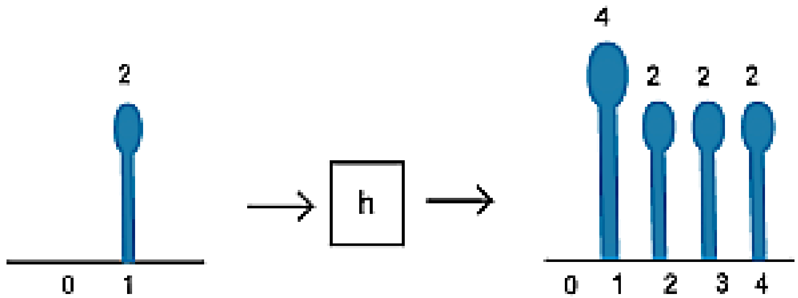

Figure 3: We now use the time-invariance property of the system to show that a delayed input results in an output of the same shape, only delayed by the same amount as the input.

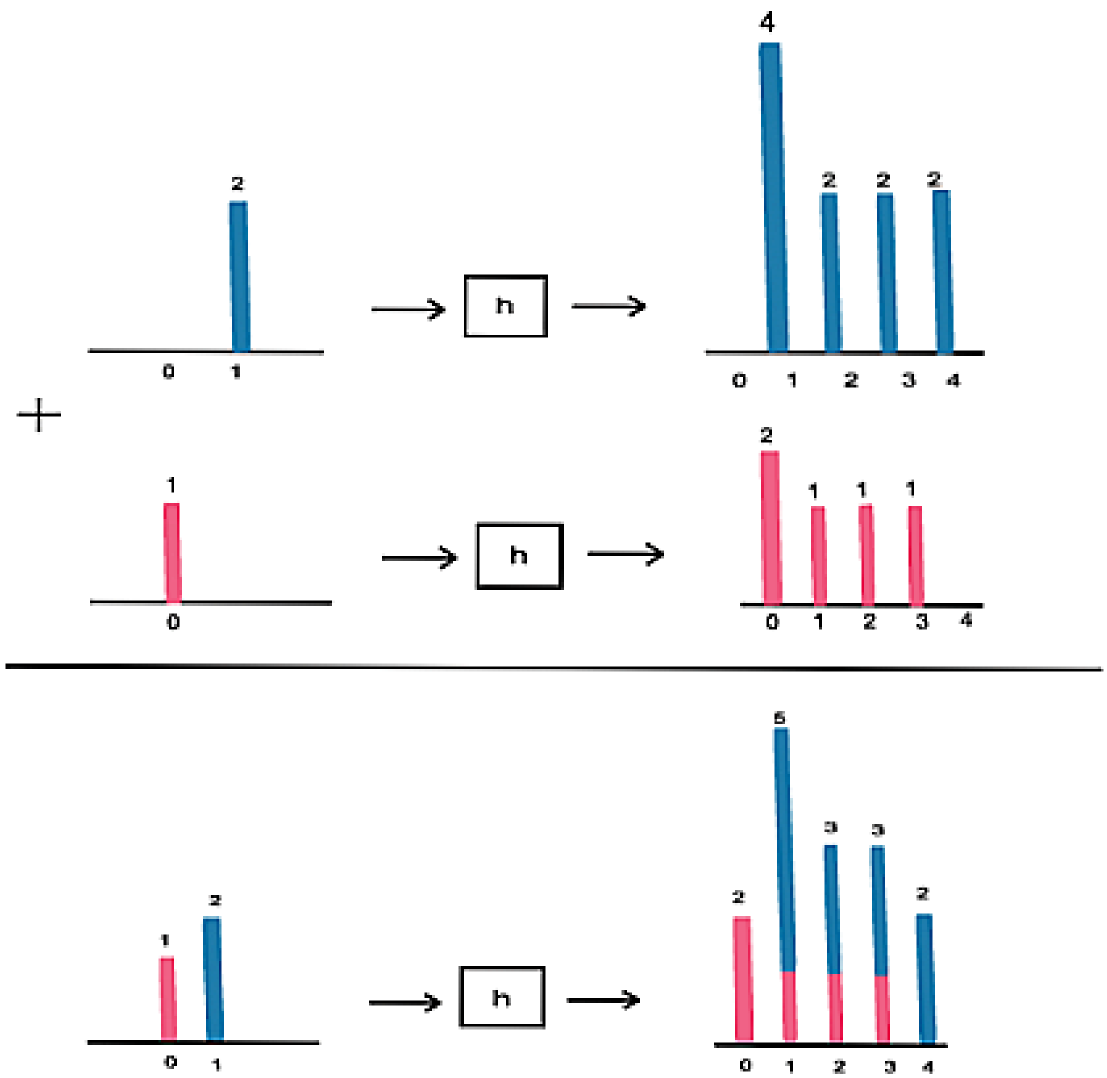

Figure 4: We now use the additivity portion of the linearity property of the system to complete the picture. Since any discrete-time signal is just a sum of scaled and shifted discrete-time impulses, we can find the output from knowing the input and the impulse response.

3 Convolution Sum

As mentioned above, the convolution sum provides a concise, mathematical way to express the output of an LTI system based on an arbitrary discrete-time input signal and the system's response. The **convolution sum** is expressed as

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) \tag{2}$$

As with continuous-time, convolution is represented by the symbol *, and can be written as

$$y(n) = x(n) * h(n) \tag{3}$$

By making a simple change of variables into the convolution sum, $k = n - k$, we can easily show that convolution is **commutative**:

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= h(n) * x(n) \end{aligned} \tag{4}$$

From (4) we get a convolution sum that is equivivalent to the sum in (2):

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)\, x(n-k) \tag{5}$$

For more information on the characteristics of convolution, read about the Properties of Convolution[5].

4 Convolution Through Time (A Graphical Approach)

In this section we will develop a second graphical interpretation of discrete-time convolution. We will begin this by writing the convolution sum allowing x to be a causal, length-m signal and h to be a causal, length-k, LTI system. This gives us the finite summation,

$$y(n) = \sum_{l=0}^{m-1} x(l)\, h(n-l) \tag{6}$$

Notice that for any given n we have a sum of the m products of $x(l)$ and a time-delayed $h(n-l)$. This is to say that we multiply the terms of x by the terms of a time-reversed h and add them up.

Going back to the previous example:

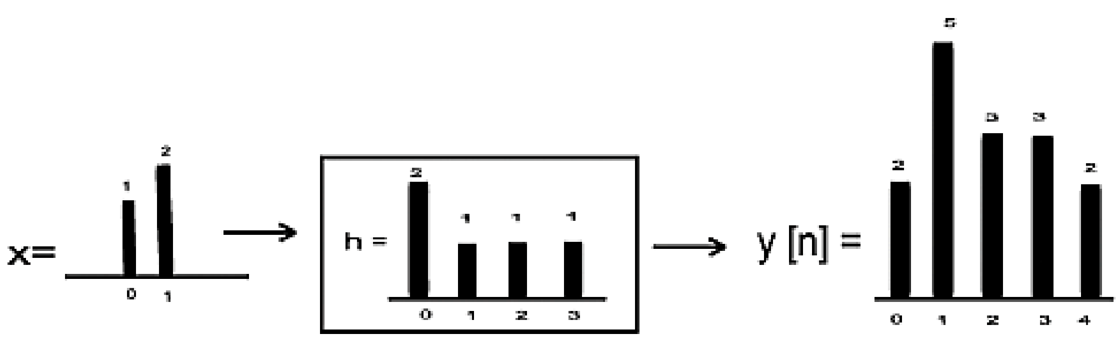

Figure 5: This is the end result that we are looking to find.

[5]"Properties of Continuous Time Convolution" <http://cnx.org/content/m10088/latest/>

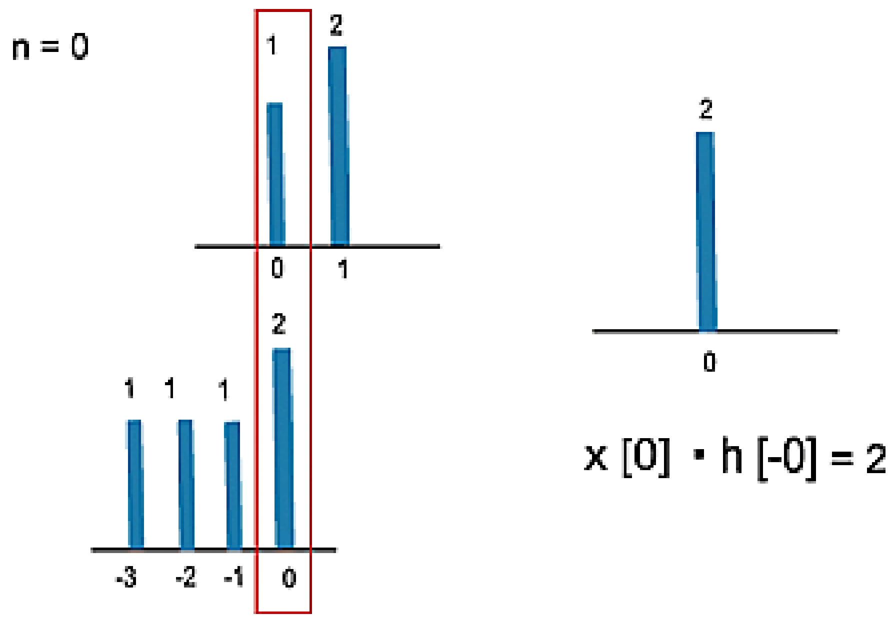

Figure 6: Here we reverse the impulse response, h , and begin its traverse at time 0.

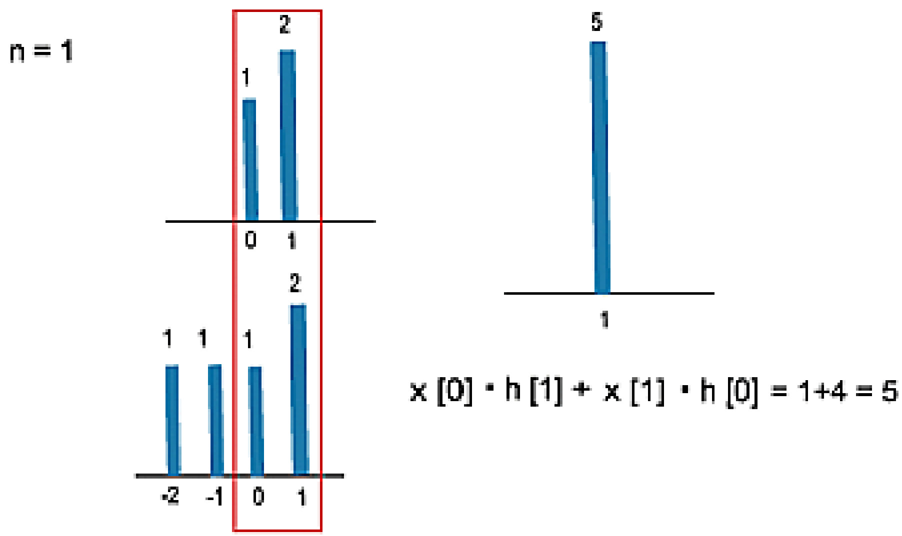

Figure 7: We continue the traverse. See that at time 1, we are multiplying two elements of the input signal by two elements of the impulse respone.

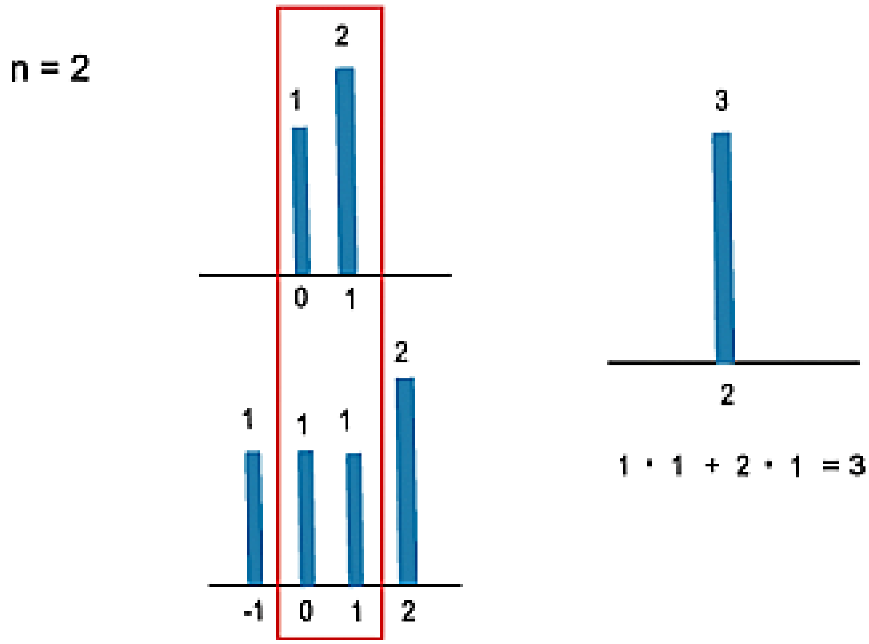

Figure 8

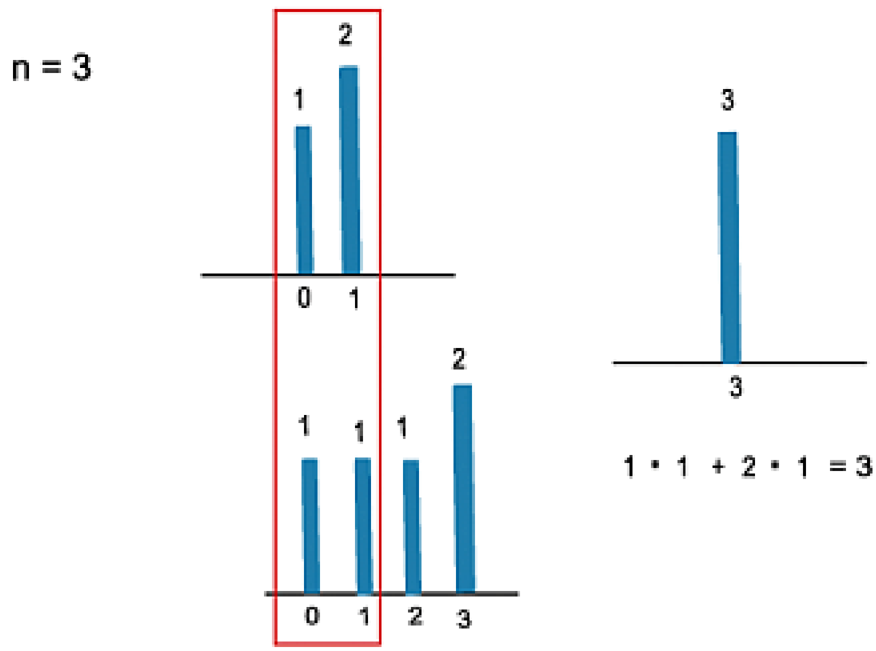

Figure 9: If we follow this through to one more step, $n = 4$, then we can see that we produce the same output as we saw in the intial example.

What we are doing in the above demonstration is reversing the impulse response in time and "walking it across" the input signal. Clearly, this yields the same result as scaling, shifting and summing impulse responses.

This approach of time-reversing, and sliding across is a common approach to presenting convolution, since it demonstrates how convolution builds up an output through time.