Discrete Time Convolution

Søren Bøgeskov Nørgaard

Started: September 18, 2013
Last update: October 10, 2014

1 Convolution Sum and Needed Values

The discrete time convolution is defined as:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \qquad (1)$$

where $x[n]$ and $h[n]$ are the two signals to be convoluted (see example below).

To start convolution, first find number of n's that we should calculate Equation 1 for to get the resulting signal $y[n]$. This is given as

$$N_y = N_x + N_h - 1$$

where N_y is the length (number of n's) of the resulting signal, N_x is the length of the $x[n]$ signal, and N_h is the length of the $h[n]$ signal.

Next find the first n to calculate for. This is

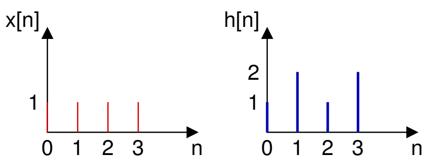
$$n_{y,\text{start}} = h[n]_{\text{start}}$$

that is, the starting n in $h[n]$ (0 in the example below)[1].

Now we hold $h[n]$ in place, flip $h[n]$ so the first "pin" comes last, and let it slide over it from left to right, incrementing n by one for each step. For every pin that overlaps the two overlapping pins are multiplied. The result for a given n, is the sum of all the "overlap products".

2 Worked Example

We wish to convolute the two signals below:



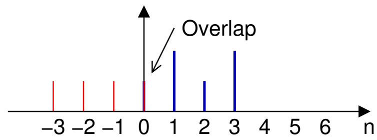
First we determine, that

$$N_y = 4 + 4 - 1 = 7$$
$$n_{y,\text{start}} = h[n]_{\text{start}} = 0$$

so the resulting signal, $y[n]$ starts at $n = 0$ and ends at $n = 6$.

[1]Note, that $x[n]$ and $h[n]$ can be swapped around so the convolution becomes $x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

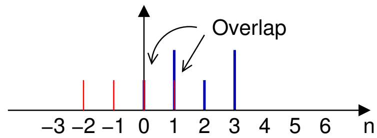
2.1 $n = 0$



We keep $h[n]$ still, flip $x[n]$, and move it in from the left ($x[n]$ does not change when it's flipped because it's symmetrical). Now the last value of $x[n]$ overlaps with first value of $h[n]$, so they are multiplied. There are no other overlaps.

$$y[0] = 1 \cdot 1 = 1$$

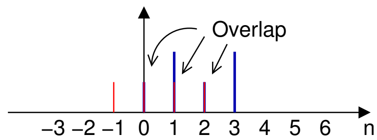
2.2 $n = 1$



Now the last two of $x[n]$ overlaps with the first two of $h[n]$. These are individually multiplied, and added together.

$$y[1] = 1 \cdot 1 + 1 \cdot 2 = 3$$

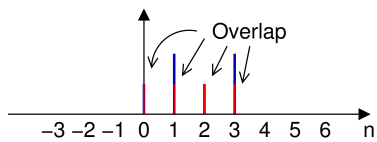
2.3 $n = 2$



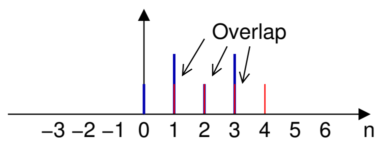
Same procedure goes here but now with three overlaps.

$$y[2] = 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 1 = 4$$

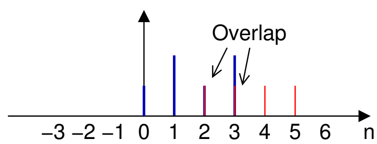
2.4 $n = 3, n = 4, n = 5$



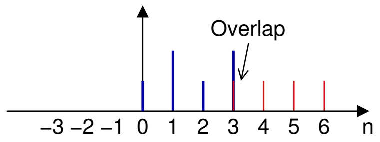
$$y[3] = 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 1 + 1 \cdot 2 = 6$$



$$y[4] = 1 \cdot 2 + 1 \cdot 1 + 1 \cdot 2 = 5$$



$$y[5] = 1 \cdot 1 + 1 \cdot 2 = 3$$



$$y[6] = 1 \cdot 2 = 2$$

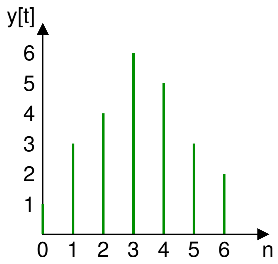
and from here, there are no more overlaps.

2.5 Resulting Signal

The resulting signal, $y[n]$, has now been found:

$$y[n] = \{1 \quad 3 \quad 4 \quad 6 \quad 5 \quad 3 \quad 2\} \tag{2}$$

which is shown below:



3 Matlab/Octave/Scilab Check

To check your results (`:-)`), use the following command in either Matlab, Octave, or Scilab (for the example above)

```
conv([1 1 1 1], [1 2 1 2])
```

and you get the result as in Equation 2

```
ans =

     1     3     4     6     5     3     2
```

4 Solving as Polynomials

Another way to solve the convolution is by treating the signals as polynomials and multiplying them together. If the two signals from above are written as coefficients of a polynomial, it will look like this (using t as the variable)

$$x[n] = \{1 \quad 1 \quad 1 \quad 1\} \Rightarrow p_x = 1 + 1t + 1t^2 + 1t^3$$
$$h[n] = \{1 \quad 2 \quad 1 \quad 2\} \Rightarrow p_h = 1 + 2t + 1t^2 + 2t^3$$

Now we multiply these two together

$$\begin{aligned} p_x p_h &= (1 + t + t^2 + t^3)(1 + 2t + t^2 + 2t^3) \\ &= \left[1 + 2t + t^2 + 2t^3\right] + \left[t + 2t^2 + t^3 + 2t^4\right] \\ &+ \left[t^2 + 2t^3 + t^4 + 2t^5\right] + \left[t^3 + 2t^4 + t^5 + 2t^6\right] \end{aligned}$$

Collecting all the terms yields

$$\begin{aligned} p_x p_h &= 1 + 3t + 4t^2 + 6t^3 + 5t^4 + 3t^5 + 2t^6 \\ \text{coefs}(p_x p_h) &= \{1 \quad 3 \quad 4 \quad 6 \quad 5 \quad 3 \quad 2\} \end{aligned}$$

which is the same result as before.