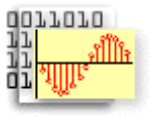


SECTION 1 - SIGNALS AND SYSTEMS

Convolution and Correlation

---

convol(x, y) **deconvol(z, x)**

correl(x, y) **covar(x, y)**

lcorr(x, y)

These functions compute, respectively, the discrete convolution, deconvolution, correlation, and covariance of sequences x and y. **lcorr** calculates a normalized covariance, assuming input vectors of equal length.

- **x**, **y**, real- or complex-valued arrays of at least two elements. For **lcorr**, **x** and **y** must be real-valued vectors.
- **z**, a real- or complex-valued vector of length such that length(z) − length(x) > 0

Results of **convol**, **correl**, and **covar** have length one less than the sum of the lengths of x and y. Results of **deconvol** have length z minus length x plus one. Results of **lcorr** have length x. All functions make use of the FFT algorithm and zero-padding, which may introduce small real or imaginary components for values near 0.

Convolution in time corresponds to multiplication in frequency, and vice versa, and is useful in digital filtering. Correlation is equivalent to convolution, with one sequence reversed in time, often used in finding the impulse response of a system. Covariance is useful for finding correlations on signals that have a bias.

Notes

- autocorrelations and autocovariances can be computed by using the same sample train for both arguments.

Function definition - convolution

Convolution is the process of "sliding" one function or sample train across another, multiplying elementwise, and summing the results. The sum for this, and following definitions, include only terms for which x and y are defined.

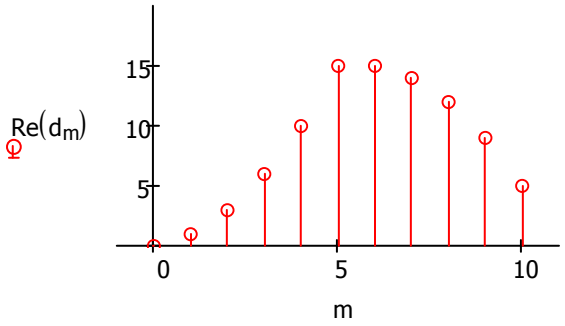
$$\text{convol}(x, y)_n = \sum_k \left(x_k \cdot y_{n-k}\right)$$

Define two short integer sequences:

$n := 0 .. 5 \qquad x_n := 1 \qquad y_n := n$

$d := \text{convol}(x, y) \qquad d^T = \left(1.987i \times 10^{-15} \quad 1 \quad 3 \quad 6 \quad 10 \quad 15 \quad 15 \quad 14 \quad 12 \quad 9 \quad 5\right)$

$M := \text{length}(x) + \text{length}(y) - 1 \qquad m := 0 .. M - 1$



Note that **convol** is not a circular convolution. Circular convolution is discussed at the end of this section. As the reversed sequence y is shifted relative to x, at first one term overlaps, then two, and so on. The maximum overlap is for index 5, where all 6 terms overlap and the value is

$$x \cdot \text{reverse}(y) = 15$$

As we continue shifting y, the amount of overlap decreases again, with the last term of the convolution again representing an overlap of one sample.

Example - deconvolution

A sequence can be recovered from its convolution by using **deconvol(z, x)**. This function assumes that z is the convolution of x with an unknown sequence y, and thus returns an answer with length equal to **length(z)** - **length(x)** + 1. (If this length isn't at least 1, you get an error telling you the arguments are invalid.)

$$Y := \text{deconvol}(d, x)$$

$$Y^T = \left(1.776 \times 10^{-15} + 3.134i \times 10^{-15} \quad 1 \quad 2 \quad 3 \quad 4 \quad 5\right)$$

All four of the functions discussed in this document use the **FFT**. The input sequences are first padded with zeros to prevent unwanted overlap and then transformed. To compute the convolution, we multiply the transforms element by element, and then take the inverse transform. For the deconvolution, we divide instead of multiplying. For the correlation and covariance, illustrated below, one transform is conjugated before multiplication.

The Fourier transform method is generally much faster than direct computation for long input sequences, however, it may introduce a small roundoff error. For example, in the answer above, notice that some elements of Y are not exactly integers. A number that's really 0 might show as something times 10^{-15}.

Function definition - correlation

The correlation shifts but does not reverse y.

$$\text{correl}(x, y)_i = \sum_k \left(x_k \cdot y_{i+k}\right)$$

At 0 shift, the first point in the correlation vector, the overlap of vectors is complete. Using the same example vectors from above,

$$C := \text{correl}(x, y)$$

$$C^T = \left(15 \quad 10 \quad 6 \quad 3 \quad 1 \quad 1.026 \times 10^{-15} - 2.051i \times 10^{-15} \quad 5 \quad 9 \quad 12 \quad 14 \quad 15\right)$$

The largest positive shift is represented by C_5 which is equal to

$$x_5 \cdot y_0 = 0$$

The next element represents the largest possible *negative* shift, which overlaps the last element of y with the first element of x:

$$x_0 \cdot y_5 = 5$$

You may prefer to look at a representation of the correlation in which the lag-0 element is in the middle, with negative shifts to the left and positive ones to the right. To reorder the result in this way, use the **recenter** function:

$$\text{recenter}(C)^T = \left(5 \quad 9 \quad 12 \quad 14 \quad 15 \quad 15 \quad 10 \quad 6 \quad 3 \quad 1 \quad 1.026 \times 10^{-15} - 2.051i \times 10^{-15}\right)$$

Function definition - covariance

The covariance carries out the same computations as the correlation, after normalizing each input to have mean 0.

$$\text{covar}(x, y)_i = \sum_k \left[(x_k - mx) \cdot (y_{i+k} - my) \right]$$

where mx and my are the means of the sequences x and y.

Finding the covariance of y with itself (the autocovariance):

$m := 0 .. 10$

$\text{covar}(y, y)_m =$

	0
0	17.5
1	8.75-2.3076i·10^-15
2	1+6.8701i·10^-17
3	-4.75-1.4414i·10^-16
4	-7.5-8.3789i·10^-16
5	-6.25+2.1284i·10^-16
6	-6.25-2.3076i·10^-15
7	...

$\text{correl}(y - \text{mean}(y), y - \text{mean}(y))_m =$

	0
0	17.5
1	8.75-2.3076i·10^-15
2	1+6.8701i·10^-17
3	-4.75-1.4414i·10^-16
4	-7.5-8.3789i·10^-16
5	-6.25+2.1284i·10^-16
6	-6.25-2.3076i·10^-15
7	-7.5-9.5688i·10^-16
8	-4.75+2.3125i·10^-16
9	1+1.7261i·10^-15
10	8.75+4.3152i·10^-15

For some further examples see the section on **FIR Filter Design** in Section 4, where **convol** is used to find the output from a filter with a finite impulse response.

Function definition - normalized covariance

This **Signal Processing Extension Pack** includes another correlation function, **lcorr**, which assumes that the two inputs have the same length. **lcorr** subtracts the means and normalizes the output, dividing by the sample length times the standard deviations of the two sequences. The lag convention in **lcorr** is opposite to **correl**, so that reversing **lcorr** corresponds to normalizing the first half of the recentered output of **covar**:

$$L := \text{recenter}\left(\frac{\text{covar}(y, y)}{\text{length}(y)\cdot\text{stdev}(y)\cdot\text{stdev}(y)}\right)$$

$$L^T = (-0.357 \quad -0.429 \quad -0.271 \quad 0.057 \quad 0.5 \quad 1 \quad 0.5 \quad 0.057 \quad -0.271 \quad -0.429 \quad -0.357)$$

$$\text{reverse}(\text{lcorr}(y, y))^T = (-0.3571 \quad -0.4286 \quad -0.2714 \quad 0.0571 \quad 0.5 \quad 1)$$

Circular Convolution

The functions presented here carry out convolution, correlation, etc., by padding the input arrays to make sure that the FFT calculation does not introduce wrapping effects. A circular convolution of two arrays of equal length N treats each array as one period of a periodic sequence and thus wraps the end of the moving array around as it is shifted. This is a situation requiring some caution - if your signals are not truly periodic, they must be zero padded so that aliasing does not occur.

Circular convolution is easy to implement directly using the FFT, but observe the aliasing effects when convolving two boxcars:

$N := 10$ $\quad n := 0 .. N - 1$

$a_n := \text{if}(0 \le n \le 6, 1, 0)$ $\quad b_n := \text{if}(0 \le n \le 6, 1, 0)$

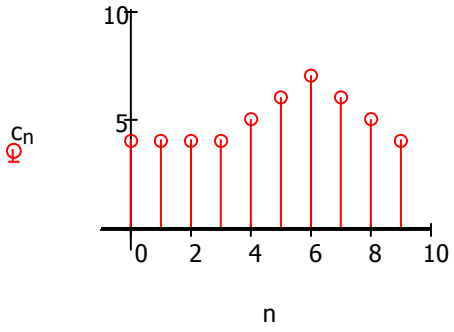
The circular convolution of a and b is

$c := \sqrt{N} \cdot \text{icfft}\left(\overrightarrow{\text{cfft}(a) \cdot \text{cfft}(b)}\right)$

$a^T = (1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0)$

$b^T = (1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0)$

$c^T = (4 \;\; 4 \;\; 4 \;\; 4 \;\; 5 \;\; 6 \;\; 7 \;\; 6 \;\; 5 \;\; 4)$



Note that the circular convolution always has the same length as the two arrays being convolved.