Signal Processing Systems (521279S)
Part 5(a) : Correlation and convolution - Processing in frequency domain

Version 1.1

November 25, 2025

Goals of the study. In the exercise, we take a look at some techniques which allow efficient implementation of correlation and convolution for long data sequences. Especially, we consider processing in the frequency-domain.

Contents

History

V1.0	11.10.2019	.
V1.1	25.11.2025	Subsections reorganized.

Correlation of two signals can be used for measuring the degree of interdependence of one process upon another, or establishing similarity between two data sets. We might be interested in evaluating the strength of correlation or finding those portions of two signals where large similarity between them exist.

For a system having a certain impulse response, the process of *convolution* gives its output for a certain input. Correlation and convolution have different purposes, but they are related operations, if we consider their computation. Therefore, they are often discussed in the same context as done in the following.

1 Definitions

1.1 Basic discrete correlation and convolution

1.1.1 Cross-correlation

The discrete cross-correlation between two data sequences $x_1(n)$ and $x_2(n)$, when the evaluation window consists of N data elements, can be defined as

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) x_2(n+j) \tag{1}$$

where the integer j represents the amount of lag between the data windows (Fig. 1).

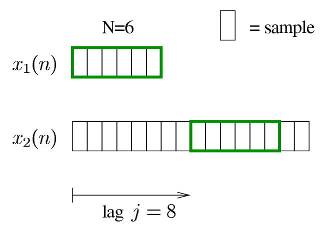


Figure 1: An example of data windows involved in computation of cross-correlation. $x_1(n)$ determines a pattern which we seek for in $x_2(n)$.

To measure similarity, the cross-correlation is typically evaluated for several lags as the phase relationship of the signals is not known. Then, the maximum value of $r_{12}(j)$ provides the degree of correlation. In many applications, the primary interest is not $r_{12}(j)$ itself but the lag j which maximizes $r_{12}(j)$.

In many cases, the lag j takes values from the set $\{0, 1, ..., M\}$ where M is the maximum lag considered. However, the range of j depends on the application and it can also take negative values.

Example. Let the sampling frequency of a signal $x_2(t)$ be 50 kHz. A signal pattern resembling a sequence $x_1(n)$, $n = 0, .., 99$ is expected to begin at a time t between 0.1 and 0.7 ms. Corresponding minimum and maximum lags are $0.0001 \times 50000 = 5$ and $0.0007 \times 50000 = 35$, and the cross-correlation should be evaluated for this range. To evaluate it for the maximum lag, we use the portion of $x_2(n)$ that begins at $n = 35$ and finishes at $n = 134$. ■

If all $x_2(n + j)$ are not available for particular j we can still evaluate the cross-correlation measure by using zeros in the place of those $x_2(n + j)$. However, there is a linear decrease in the values of $r_{12}(j)$ which is known as the *end effect*.

1.1.2 Autocorrelation

Autocorrelation $r_{11}(j)$ is a specific case of the cross-correlation, where $x_1(n) = x_2(n)$. Note that $r_{11}(0) = (1/N) \sum_{n=0}^{N-1} x_1^2(n)$ corresponds to the average power of the signal over the N samples used in evaluation. In fact, if a power signal repeats itself every Nth sample the DFT of its autocorrelation evaluated for the lags $j \in \{0, 1, ..., N - 1\}$ gives the power spectrum of the signal. The autocorrelation is useful for identifying hidden periodicities in the signal as illustrated in Fig. 2.

1.1.3 Convolution

Discrete-time convolution of two real-valued signals $h(n)$ and $u(n)$ can be defined as

$$y(k) = \sum_{n=0}^{N-1} h(n)u(k - n) \tag{2}$$

Here, $h(n)$ can be associated with the impulse response of some system, $u(n)$ with its input signal and $y(n)$ with its output. The similarity between this definition and the cross-correlation defined in (1) implies that similar computational structures can be used for solving both kinds of problems (see Sec. 2).[1]

[1] For complex-valued signals $x_1(n)$ and $x_2(n)$ the definition of the cross-correlation is

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1^*(n) x_2(n + j)$$

where * denotes complex conjugation. Taking the conjugate of $x_1(n)$ ensures that aligned lumps with imaginary components will contribute positively to the real part of the sum. The definition of convolution for complex-valued signals $h(n)$ and $u(n)$ is

$$y(k) = \sum_{n=0}^{N-1} h(n)u(k - n).$$

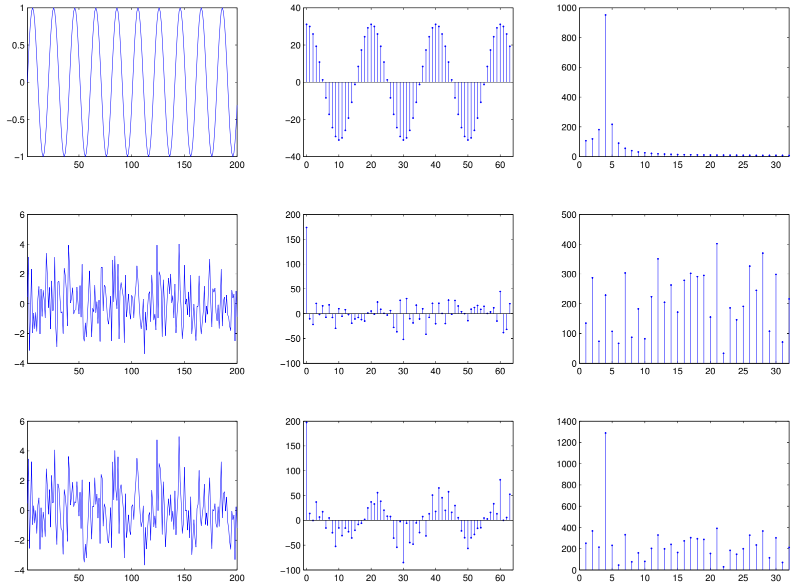


Figure 2: Detecting hidden periodities using autocorrelation. Rows: top - a sine signal $s(t)$, middle - a noise signal $n(t)$, bottom - s(t)+n(t). Columns: left - signal plot, middle - autocorrelation $r_{12}(j)$, right - magnitude of the DFT of $r_{12}(j)$.

1.2 Other correlation measures

The cross-correlation defined in (1) is not always an appropriate measure for correlatedness or similarity of two signals. Therefore, variants of it are used in many cases.

For example, (1) depends on the mean values of the sequences in the data windows. Therefore, if the correlatedness of *shapes* of the data sequences (waveforms of signals) were considered as a more appropriate measure, one would subtract those mean values

$$\begin{aligned}
\bar{x}_1 &= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n) \\
\bar{x}_2(j) &= \frac{1}{N}\sum_{n=0}^{N-1} x_2(n+j)
\end{aligned}$$

before multiplications, which gives a measure

$$c_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} (x_1(n) - \bar{x}_1)(x_2(n+j) - \bar{x}_2(j)). \tag{3}$$

Another correction that is used many times in practice is to normalize by powers of the signals. Combining these methods, we obtain a correlation measure called the *correlation coefficient*:

$$c_{12}(j) = \frac{\sum_{n=0}^{N-1} (x_1(n) - \bar{x}_1)(x_2(n+j) - \bar{x}_2(j))}{\sqrt{\sum_{n=0}^{N-1} (x_1(n) - \bar{x}_1)^2} \sqrt{\sum_{n=0}^{N-1} (x_2(n+j) - \bar{x}_2(j))^2}}. \tag{4}$$

This measure gives values in the range $[-1, +1]$.[2] Weak point of the correlation coefficient is its computational complexity which may preclude the use of it in real-time systems.

Example. In image analysis, a common task is to find correspondences between two images. 2-D cross-correlation can be used to find good matches between blocks of pixels. However, due to changes in imaging conditions, there can be brightness changes. If one does not or cannot compensate for those changes by prefiltering of images (some form of band-pass/high-pass filtering), the performance of the cross-correlation is poor. Therefore, 2-D correlation coefficient is used in many applications as a robust block matching measure. ■

The cross-correlation measures the similarity of two signals. Alternatively, distance measures based on differences $x_1(n) - x_2(n+j)$ can be used. For example, motion estimation in hybrid video codecs is typically based on the evaluation of the sums of absolute differences (SAD). The particular advantage of this measure is that no multiplications are needed to evaluate it.

[2] If the signals are zero-mean over the evaluation window, $\bar{x}_1 = 0$ and$\bar{x}_2(j) = 0$, and if their average energies are constant, $(1/N) \sum x_1^2(n) = 1$, $\bar{x}x_2(j) = (1/N) \sum = 1$.

2 Frequency-domain implementation

When the length of h in Eq. 2, N, is small, the computation of convolution can be implemented directly in the time-domain (recall MAC unit-based FIR filter implementation). This is also true for correlation. However, when N is large, it becomes interesting to consider the implementation in the frequency domain, which we consider next.

The fast Fourier transform (FFT) provides a very efficient way for computing discrete Fourier transforms (DFT). On the other hand, DFT provides a way for computing specific types of convolutions, so-called *cyclic convolutions*. In the following, it is shown how it is possible to compute *linear convolutions* and correlations from cyclic convolutions.

2.1 Cyclic convolution and correlation from DFT

If $u(n)$ is a *periodic* signal with the period length N and the duration of $h(n)$ is N then the convolution theorem states that the convolution can be computed via multiplication in the frequency domain:

$$y(k) = \sum_{n=0}^{N-1} h(n)u(k-n) = F_D^{-1}[\, H(l)U(l)\,], \quad k = 0, 1, \ldots, N-1 \tag{5}$$

where $H(l)$ and $U(l)$ are the N-point DFTs of $h(n)$ and $u(n)$, $n = 0, 1, \ldots, N-1$, and F_D^{-1} denotes the inverse DFT operation. The convolution is cyclic which means that $y(k+iN)$ is equal to $y(k)$ as $u(n)$ is periodic.

Similarly, the correlation theorem says that the cross-correlation can be computed as

$$r_{12}(j) = \frac{1}{N} F_D^{-1}[\, X_1^\star(l) X_2(l)\,], \tag{6}$$

where $X_1^\star(l)$ denotes the N-point DFT of $x_1(-m)$ which is the complex conjugate of $X_1(k)$, and $X_2(k)$ is the DFT of $x_2(n)$. Presumptions about the periodicity and finiteness of the functions are the same as in the case of convolution.

Theorems (5) and (6) make restricting assumptions about associated functions. Assumed periodicity of $u(n)$ in (5) and $x_2(n)$ in (6) quarantees that the results are meaningful, that is, the results provide correct information about the linear convolution or correlation. However, signals are typically not periodic, and additional techniques are needed before the FFT can be used. Zero padding and sectioning allow us to do that.

2.2 Zero padding

Zero padding is used to make application of the FFT possible For example, let $N = 3$, and let $x_1(n)$ be (a_0, a_1, a_2), and one cycle of $x_2(n)$ be (b_0, b_1, b_2). Then the cross-correlation of the signals is cyclic:

a_0	a_1	a_2	**lag** j	**cyclic correlation** $(\times 3)$
b_0	b_1	b_2	0	$a_0b_0 + a_1b_1 + a_2b_2$
b_1	b_2	b_0	1	$a_0b_1 + a_1b_2 + a_2b_0$
b_2	b_0	b_1	2	$a_0b_2 + a_1b_0 + a_2b_1$
b_0	b_1	b_2	3	starts to repeat ..

Equation (6) can be used to calculate this result.

If $x_2(n)$ is $(b_0, b_1, b_2, 0, 0, 0, \dots)$ the cross-correlation is

a_0	a_1	a_2	**lag** j	**linear correlation** $(\times 3)$
b_0	b_1	b_2	0	$a_0b_0 + a_1b_1 + a_2b_2$
b_1	b_2	0	1	$a_0b_1 + a_1b_2$
b_2	0	0	2	a_0b_2
0	0	0	3	all zero after this ..

Note the difference in the results above. Now, if we construct signals $x'_1(n)$ and $x'_2(n)$ so that $x'_1(n)$ is $(a_0, a_1, a_2, 0, 0)$, and $x'_2(n)$ is periodic with a period $(b_0, b_1, b_2, 0, 0)$ we get

a_0	a_1	a_2	0	0	**lag** j	**cyclic correlation** $(\times 5)$
b_0	b_1	b_2	0	0	0	$a_0b_0 + a_1b_1 + a_2b_2$
b_1	b_2	0	0	b_0	1	$a_0b_1 + a_1b_2$
b_2	0	0	b_0	b_1	2	a_0b_2
0	0	b_0	b_1	b_2	3	a_2b_0
0	b_0	b_1	b_2	0	4	$a_1b_0 + a_2b_1$
b_0	b_1	b_2	0	0	5	starts to repeat ..

When you compare this result to the linear correlation computed above you can see that the values for the lags $j = 0, 1, 2$ are the same. As we know that the linear correlation must be zero for other lags the cyclic correlation of the zero-padded sequences gave the values we needed (up to a scaling factor).

In general, if the length of $x_1(n)$ is N_1 and the length of $x_2(n)$ is N_2 the minimum amount of zero padding makes their length $N_1 + N_2 - 1$. Then cyclicity of the DFT does not affect the result *at all*.[3]

In order to apply the FFT the length should be a power of two. If this is not the case, we can add zeros until this requirement is fulfilled. Alternative implementations of fast correlation are illustrated in Fig. 3.

[3] If the effects of cyclicity are allowed in some part of the result, less zero padding may be sufficient.

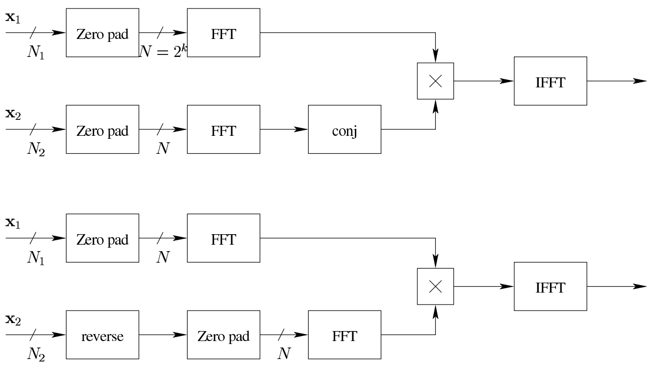


Figure 3: Two ways to implement fast FFT based correlation: either time reversal $x_1(-m)$ or complex conjugation $X_1^*(l)$ can be used. The values of $r_{12}(j)$ are extracted from the output of the IFFT operation.

Example. The method described applies also to convolution. Consider FIR filtering of a signal with a 200-tap filter. Due to the large number of taps, an FFT based implementation can give computational advantage[4]. Let us assume that for some reason we partition the input $u(n)$ to sections having length of 250 elements and the overlap-add method (see Sec. 2.3) is used. With this strategy, cyclic effects are not allowed and the minimum requirement for using DFT is to have padded sections of length $200 + 250 - 1 = 449$ elements. The next power-of-two ≥ 449 is 512, so we add extra 63 zeros to use the FFT[5]. ■

2.3 Sectioning

In many cases, very long sequences have to be processed (consider a typical signal filtering application). Then, it becomes necessary to be able to process signals in smaller parts which is called *sectioning*. There are two main strategies for doing it.

In the *overlap-add* strategy, the idea is to split the data $u(n)$ into sections of length N_u. If the system's impulse response $h(n)$ has the length N_h, both the $h(n)$ and the section are padded by zeros so that they both have length equal to $N_u + N_h + N_f - 1$ (length of the cyclic convolution). The parameter $N_f \geq 0$ represents an extra padding, which can be used to obtain specific cyclic convolution length required by some FFT algorithm.

[4]The rule of thumb is that computations in the frequency domain is beneficial when N is over 128; see (Ifeachor & Jervis 2002, Table 5.1). The exact limit depends on the implementation architecture.

[5]It might be interesting to know, that zero padding here just interpolates the spectrum.

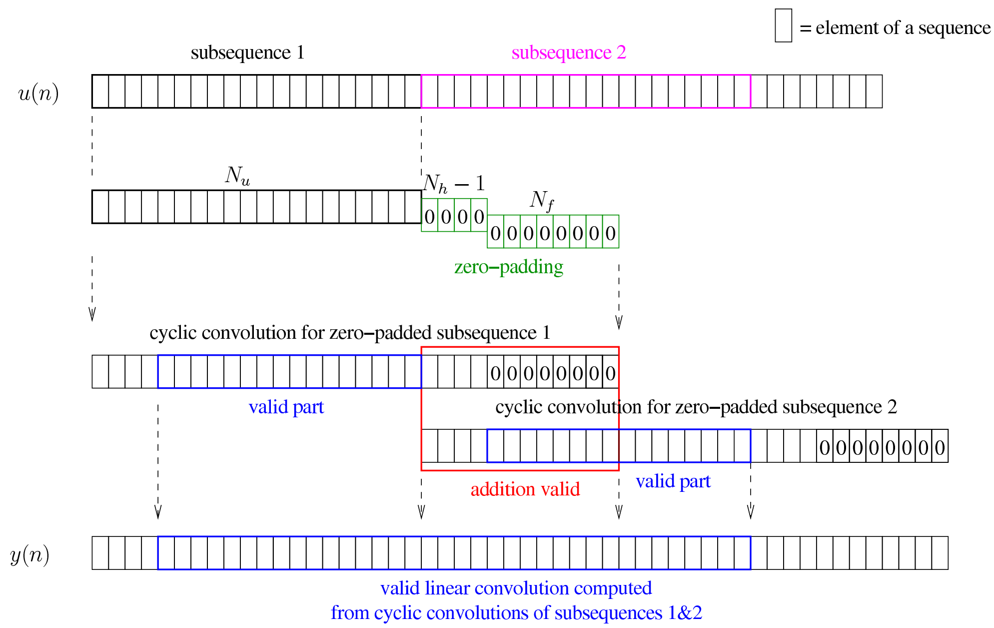


Figure 4: Overlap-add based sectioning ($N_u = 20$, $N_h = 5$, $N_f = 8$). Choice of N_f allows use of 32-point FFT. The first $N_h - 1$ and last $N_h - 1 + N_f$ values of cyclic convolution of each section do not correspond to linear convolution.

As the cyclic convolution is calculated the first $N_h - 1$ and last $N_h + N_f - 1$ values do not correspond to the linear convolution. However, if sections of $u(n)$ are overlapped by $N_h + N_f - 1$ elements and corresponding portions of the cyclic convolution are added the correct linear convolution is obtained as a result. The approach is illustrated in Fig. 4.

An alternative to the overlap-add strategy is the *overlap-save* method. The idea is to save only the valid parts of convolution for a section and discard the rest. To do this, the input signal is split into sections, whose length is *the length of the cyclic convolution implementation* (e.g. the length of FFT). The length of the system's impulse response, N_h, must be less than N_u. Input signal sections used as an input to FFT must overlap by $N_h - 1$ elements. Once the convolution of the input section and zero-padded impulse response has been calculated, the first $N_h - 1$ elements of the result are discarded, and the last elements form a part of the convoluted signal (Fig. 5).

See also the numeric examples of the sectioning methods on the last page of this document (Fig. 8).

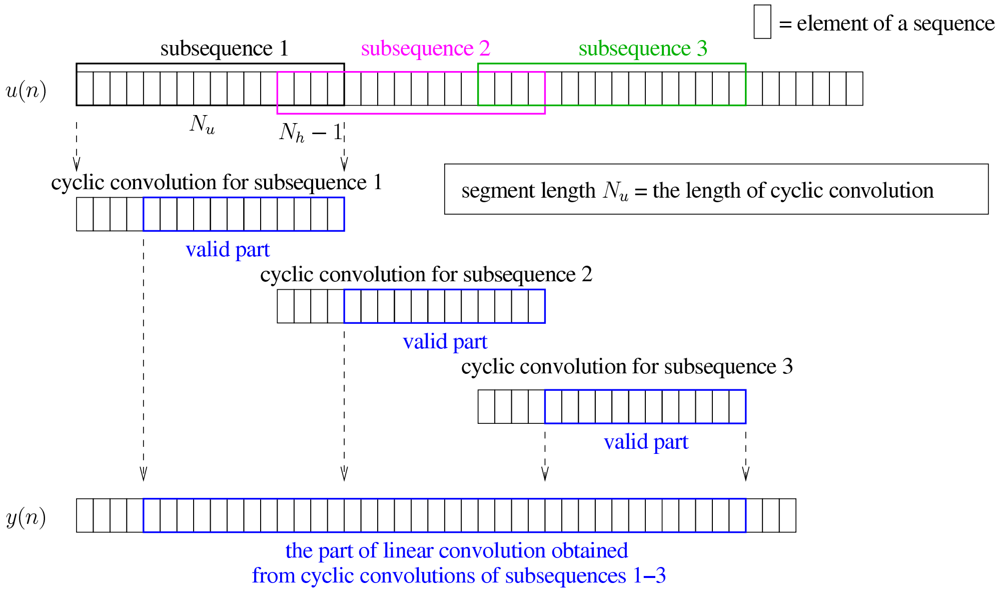


Figure 5: Overlap-save based sectioning ($N_u = 16$, $N_h = 5$). The length of subsequences (here 16) is equal to the number of points used in FFT.

2.4 Evaluating correlation via sectioning for convolution

It was outlined in Eq. 6 and Fig. 3, how FFT can be used to implement correlation. However, it is not clear at this point how we implement processing in the case, where the other signal (x_2) is very long. The sectioning techniques were presented above for convolution, and we can use them also for correlation. To do that, we express the correlation (Def. 1 in Sec. 1.1.1) in terms of convolution (Def. 2) first.

We get

$$
\begin{aligned}
r_{12}(j) &= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n+j) && (7)\\
&\propto \sum_{n=0}^{N-1} x_1(n)x_2(n+j) && (8)\\
&\stackrel{n=-n'}{=} \sum_{-n'=0}^{N-1} x_1(-n')x_2(j-n') && (9)\\
&\stackrel{x_1(-n')=s(n')}{=} \sum_{n'=1-N}^{0} s(n')x_2(j-n') && (10)\\
&\stackrel{m=n'+N-1}{=} \sum_{m=0}^{N-1} s(m-N+1)x_2(j-(m-N+1)) && (11)\\
&\stackrel{h(m)=s(m-N+1)}{=} \sum_{m=0}^{N-1} h(m)x_2(j-m+N-1) && (12)
\end{aligned}
$$

$$
\begin{aligned}
&\stackrel{k=j+N-1}{=} \sum_{m=0}^{N-1} h(m)x_2(k-m) && (13)\\
&\stackrel{x_2(m)=u(m)}{=} \underbrace{\sum_{m=0}^{N-1} h(m)u(k-m)}_{\text{convolution } y(k)}, && (14)
\end{aligned}
$$

where the equalities shown over '=' signs show the action taken. We read out the following:

1. Based on actions of (12) and (10), $h(m)$ in convolution is set according to

$$h(m) = s(m-N+1) = x_1(-(m-N+1)) = x_1((N-1)-m),$$

   that is, we reverse x_1.

2. According to the action of (14), we set $u(m) = x_2(m)$.

3. Finally, according to (13), the cross correlation for the lag j can be obtained from the convolution at the point $k = j + N - 1$.

For example, correlation for the lag $j = 0$ corresponds to the convolution at $k = N-1$. The derived solution is illustrated in Fig. 6. The signal x_1 can be considered as a finite-size template, that we are looking for in x_2, which can be very long.

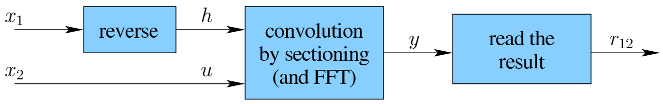


Figure 6: Setup for evaluating correlation via convolution.

3 Correlation for specific lag: recursive implementation

The cross-correlation *for a specific lag* j can be implemented recursively in on-line applications. At the time instant k, the cross-correlation can be defined as

$$r_{12}(j,k) = \frac{1}{N}\sum_{n=k}^{k+N-1} x_1(n)x_2(n+j). \tag{15}$$

Note that direct evaluation of this sum requires N multiplications, $N-1$ additions and one division. However, we have the following dependence:

$$\begin{aligned} r_{12}(j,k+1) &= \frac{1}{N}\sum_{n=k+1}^{k+N} x_1(n)x_2(n+j) & (16)\\ &= r_{12}(j,k) + \frac{1}{N}x_1(k+N)x_2(k+N+j) - \frac{1}{N}x_1(k)x_2(k+j) & (17) \end{aligned}$$

So, four operations are needed to update the value of autocorrelation for the lag j: one multiplication and one division (or two multiplications) to obtain the second term, and one addition and one subtraction. A first-in-first-out (FIFO) buffer of size N must be used to make the third term available (see Fig. 7).

Finally, note that calculation of the mean values required by some correlation definitions (see Sec. 1.2) can also be implemented recursively.

4 Summary

Two common operations in signal processing are convolution and correlation whose definitions resemble each other. Therefore, they can be implemented with similar techniques. The purpose of the correlation is to evaluate the similarity of signals, and there are several related measures that can be used for this purpose. The choice depends on the application characteristics where performance of a measure and its computational complexity must be taken into account.

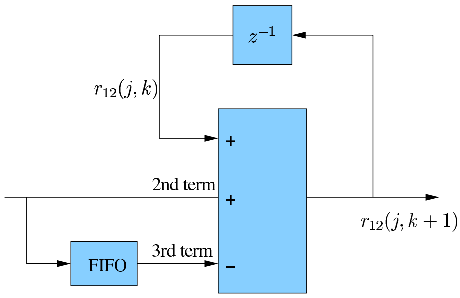
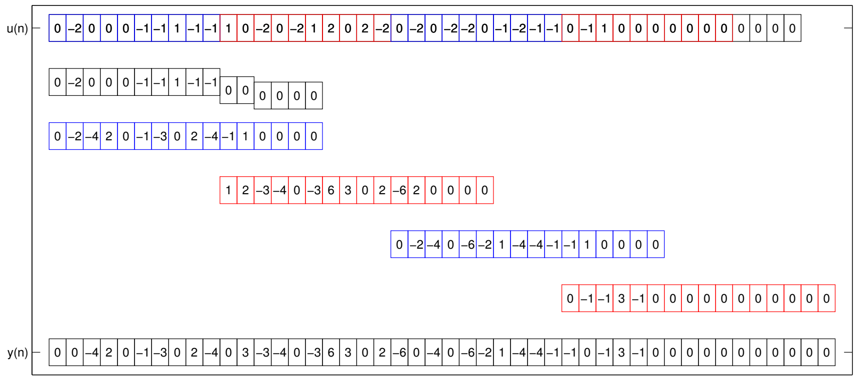


Figure 7: Recursive implementation of the cross-correlation.

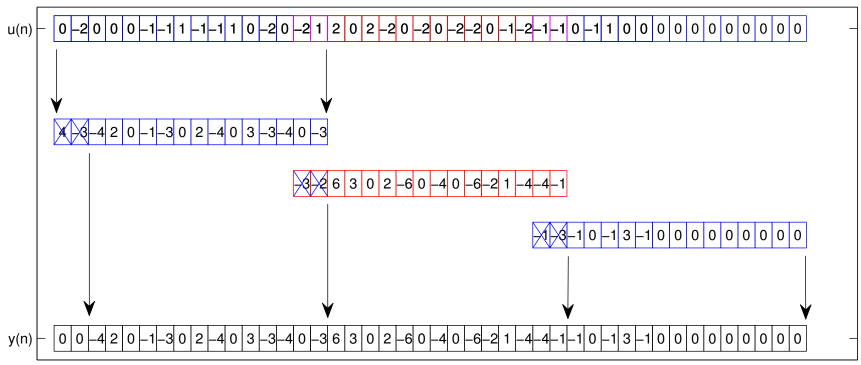
Convolution and correlation can be implemented in the frequency domain (using FFT). However, assumptions behind the theorems must be taken into account to have correct implementations. To process long sequences, one may use processing in sections (Sec. 2.1.3) or recursive formulations (Sec. 2.2). The last part of the course considers adaptive filtering techniques which find correlations in measured signals using different kind of recursion.

References

Ifeachor, E. C. & Jervis, B. W. (2002) Digital Signal Processing: a Practical Approach. 2nd edition. Pearson Education Ltd., Harlow, England.



(a) Overlap-add. Row 1: input $u(n)$, non-overlapping sections in different color ($N_u = 10$). Row 2: First section with zero padding (6 zeros added as $N = 16$). Rows 3-6: outputs of the cyclic convolution for each input section. Row 7: output $y(n)$.



(b) Overlap-save. Row 1: input $u(n)$, sections illustrated by blue and red, their overlaps with magenta color ($N_u = N = 16$). Rows 2-4: outputs of the cyclic convolution for each input section. Discarded elements crossed. Row 5: output $y(n)$.

Figure 8: Numeric illustration of the sectioning techniques. The input sequences are here convolved by $h(n) = (1, 2, -1)$

.