Correlation & convolution. Frequency domain implementation.

Signal Processing Systems Fall 2025

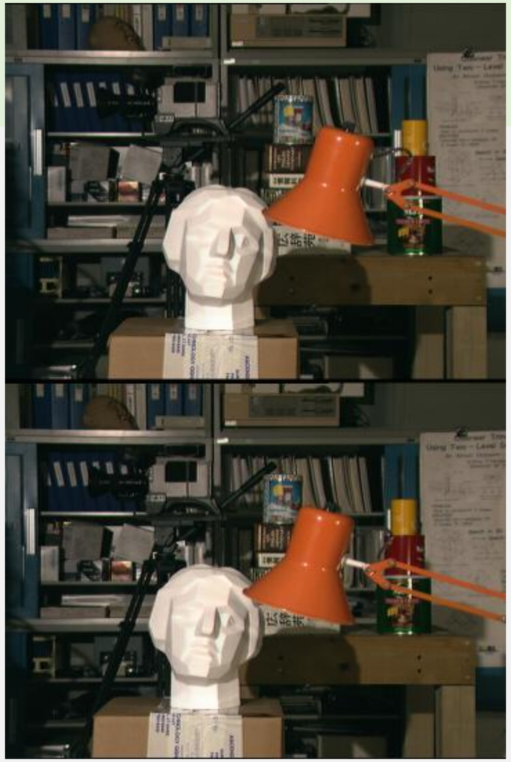
Lecture 10 (Friday 28.11.)

Outline

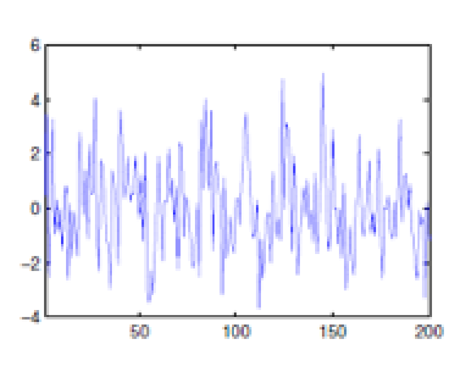
- Definitions
  - Cross-correlation, autocorrelation, convolution
  - Other correlation measures
- Implementating convolution and cross-correlation
  - In frequency domain
  - Cyclic convolution, zero padding, sectioning
- Recursive implementation of cross-correlation

1. Definitions

- Cross-correlation
  - Measuring **similarity** of two signals
  - Searching for a known feature from a signal
  - Determine relative lag of two signals
- Autocorrelation
  - Detecting repeated patterns in signals (e.g. periodic events obscured by noise)
  - Finding missing fundamental frequency based on harmonic components
- Convolution (linear)
  - The output of a linear time-invariant (LTI) system is obtained through convolution of the input with the system's impulse response
  - FIR filtering – so far, we have considered implementations in time domain.
- Cross-correlation and convolution are related operations
  - Same techniques can be used for implementation



Stereo vision & depth estimation



Periodicity here?

Linear convolution

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

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

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.

This definition specifies so-called linear convolution. It corresponds to the definition of the FIR filtering.

So far, time-domain implementations has been considered in this course, for example, use of MAC unit (DT2) and multirate implementation (DT4).

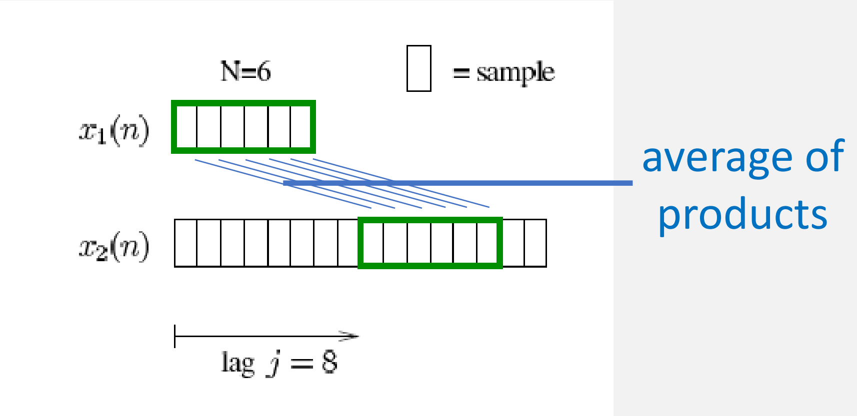
In some applications, however, **frequency-domain** may provide a better alternative.

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

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

where the integer j represents the amount of lag between the data windows

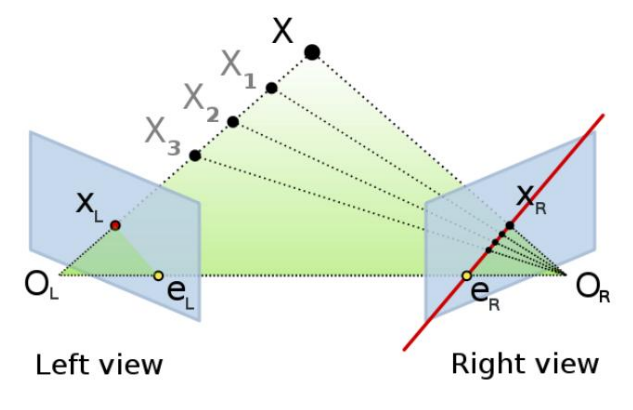


A set of lag values is evaluated in this way. The interest may be in

- the maximum value = strength of signal correlation
- the lag maximizing the correlation = localization of a pattern

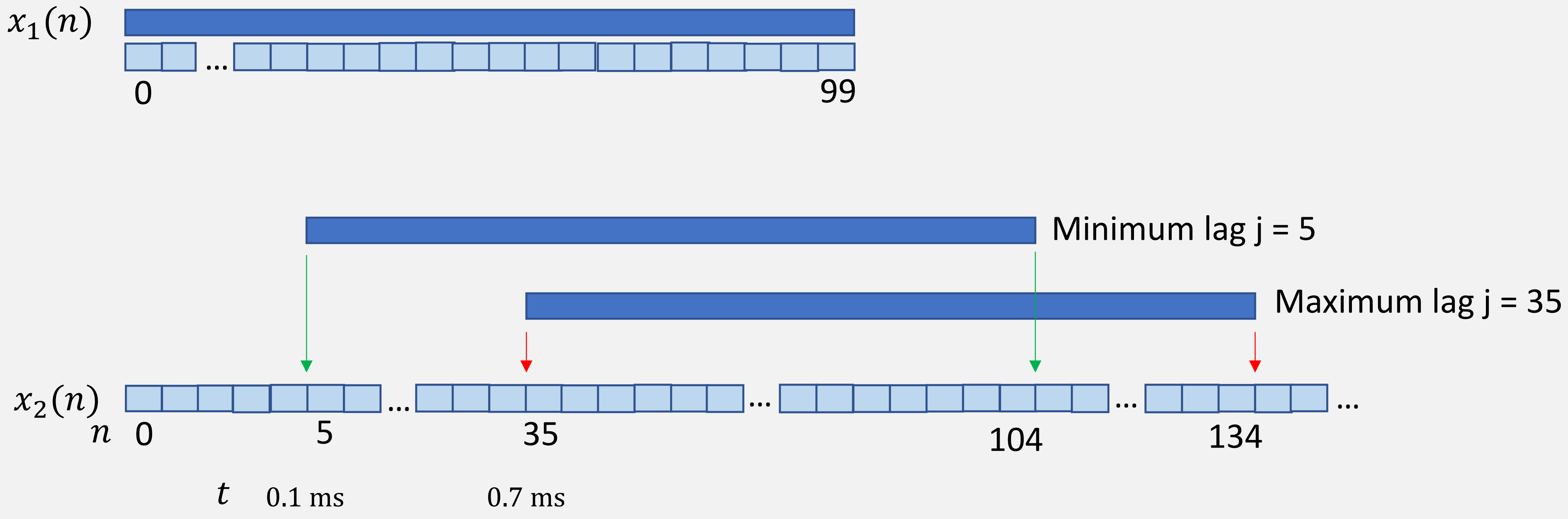
The set of lags depends on the case considered and may also take negative values.

Example. Stereo matching in computer vision



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 $n = 134$. ∎

Template sequence, length = 100 samples



Autocorrelation

Autocorrelation is a special case of cross-correlation, where $x_1(n) = x_2(n)$.

$r_{11}(0) = (1/N)\sum_{n=0}^{N-1} x_1^2(n)$ corresponds to the **average power** of the signal over N samples.

Autocorrelation is useful for identifying hidden periodities in the signal.

signal samples

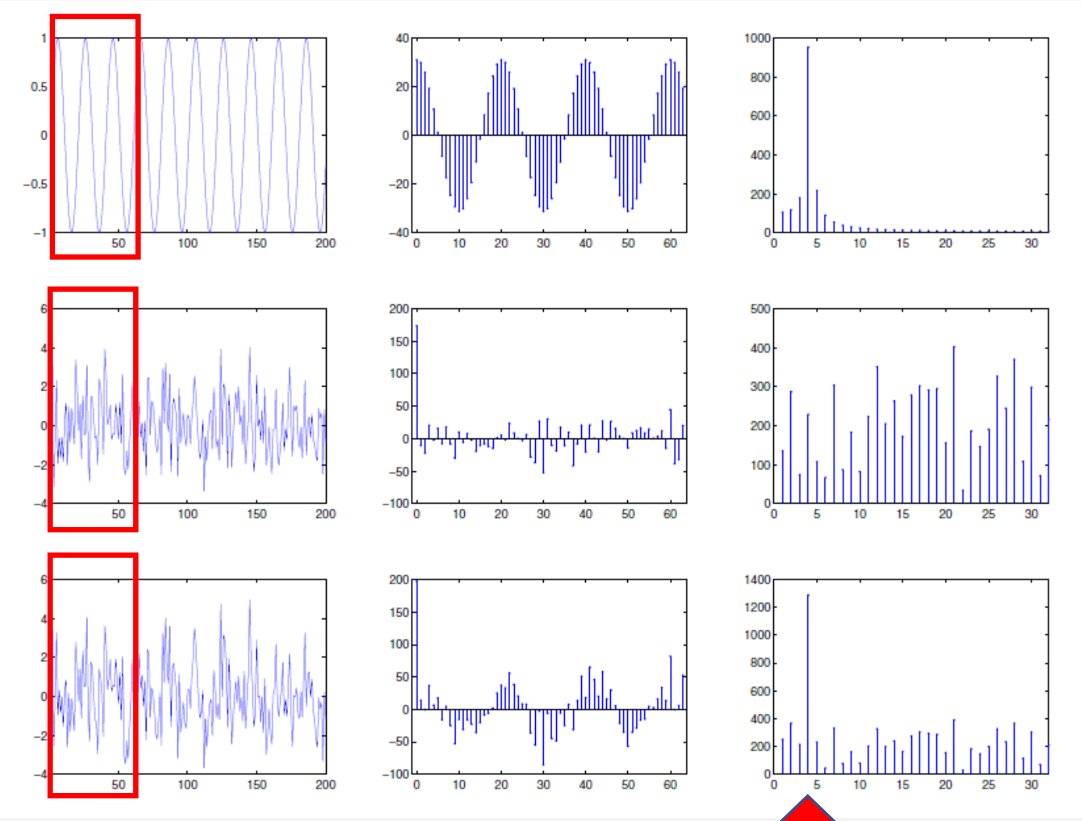
Use some time window to establish **template**: auto-correlation for different lags *j*

auto-correlation's DFT magnitude

signal

noise

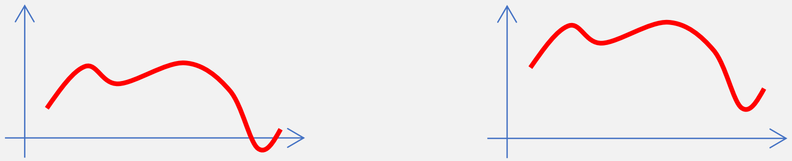
signal +noise



PERIOD

Examples of other correlation measures

- **Appropriate form of matching measure depends on application. Some examples:**
- Measuring cross-correlation of **shapes**
    - Subtract means from inputs before evaluating the cross-correlation
- Correlation coefficient
    - Normalize by powers of signals
    - Example: **robust** block matching in processing of pairs of images (as there are illumination variation problems)
- Sum of absolute differences (SAD)
    - Advantage: no multiplications
    - Example use: motion estimation by block matching in hybrid video encoders (2-D SAD)
    - Low SAD indicates small residuals => low number of bits for encoded residuals



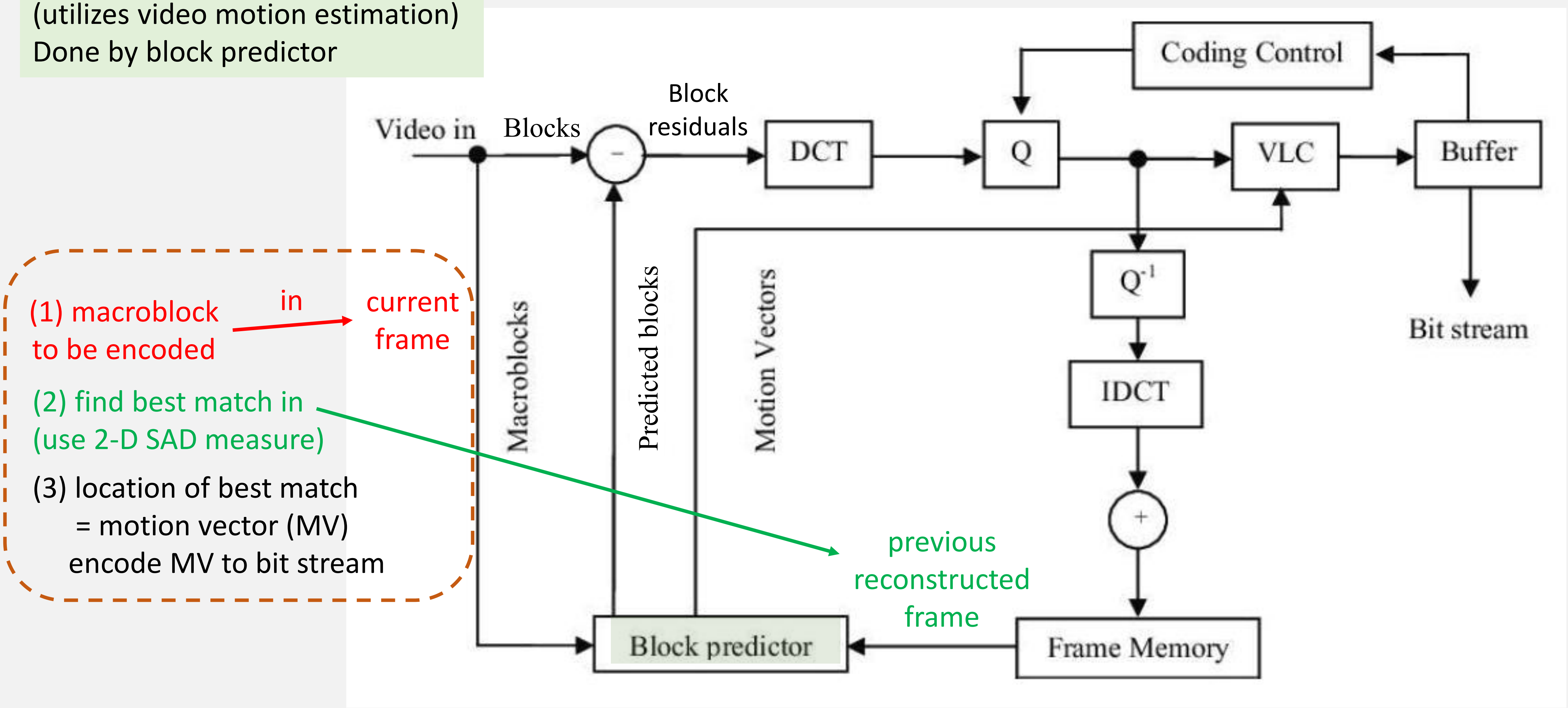
$$c_{12}(j) = \frac{1}{N}\sum_{n=0}^{N-1}(x_1(n) - \bar{x}_1)(x_2(n+j) - \bar{x}_2(j))$$

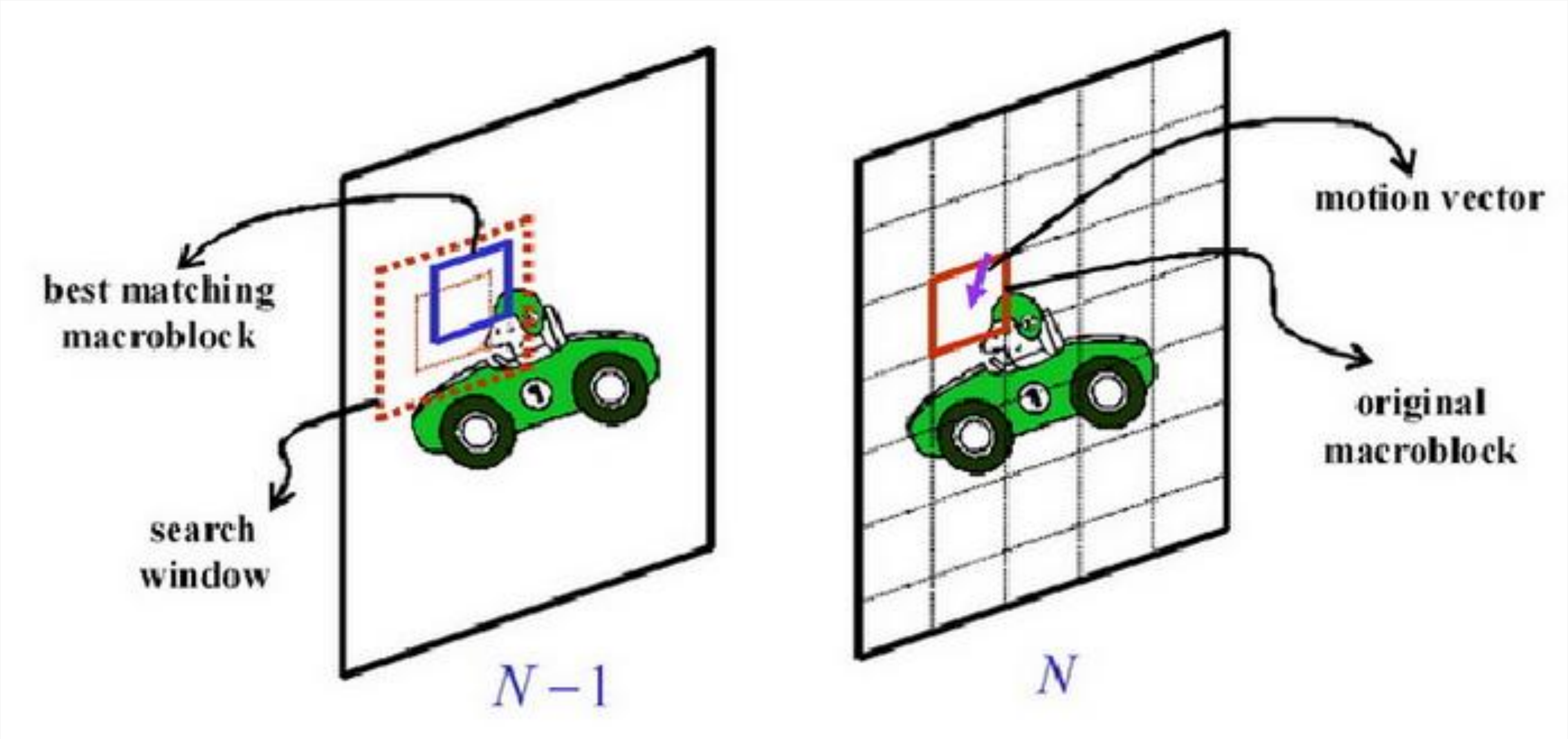
$$c_{12}(j) = \frac{\sum_{n=0}^{N-1}(x_1(n) - \bar{x}_1)(x_2(n+j) - \bar{x}_2(j))}{\sqrt{\sum_{n=0}^{N-1}(x_1(n) - \bar{x}_1)^2}\sqrt{\sum_{n=0}^{N-1}(x_2(n+j) - \bar{x}_2(j))^2}}$$

$$\mathrm{SAD}(j) = \sum_{n=0}^{N-1}|x_1(n) - x_2(n+j)| \qquad \text{(1-D)}$$

Transform based motion compensated video encoder

Prediction of block content
(utilizes video motion estimation)
Done by block predictor





previous
reconstructed
frame

current
frame

Note. Case of complex inputs

The definition of linear convolution

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

holds also for linear convolution of **complex-valued** signals.

The definition of cross-correlation of complex-valued signals is

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

where * denotes complex conjugate.

Without conjugation:

$$\underbrace{(a+jb)}_{x_1}\underbrace{(c+jd)}_{x_2} = (ac-bd) + j(ad+bc)$$

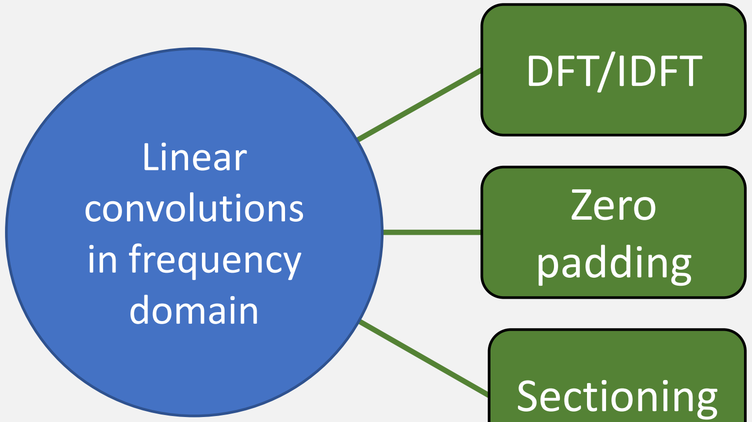
With conjugation:

$$\underbrace{(a-jb)}_{x_1^*}\underbrace{(c+jd)}_{x_2} = (\boldsymbol{ac} + \boldsymbol{bd}) + j(ad+bc)$$

Taking the conjugate ensures that in the case of good match imaginary components will contribute positively to the **real part** of the sum. Real part can then be used as a measure of correlation strength.

2. Implementing convolution & cross-correlation

- We can use time-domain techniques considered for FIR filtering (e.g. multirate)
- When the length of the filter is large, it becomes interesting to use the <u>frequency domain</u>
  - **Convolution theorem** provides a method for computing cyclic convolutions using DFT and IDFT
  - **Zero padding** provides a way for applying cyclic convolution to compute linear convolution
  - **Sectioning** provides a way to process long data sequences
- Techniques are also applicable for implementing cross-correlation in the case of long templates



2.1. Convolution theorem

Linear convolution

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

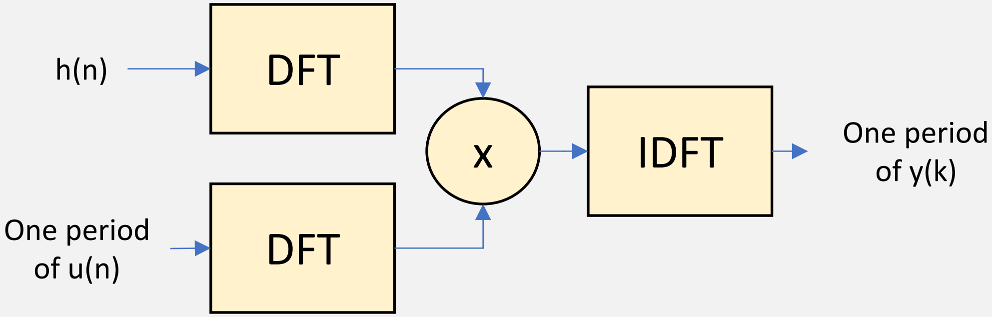
in the case of **periodic signal**

THEOREM

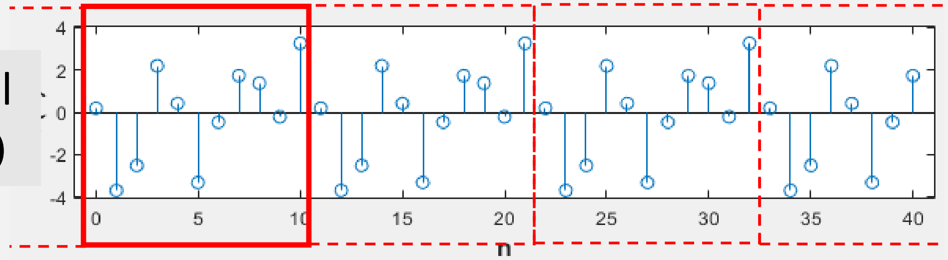
If $u(n)$ is a *periodic* signal with the period length N and the duration of $h(n)$ is N then the convolution theorem states that the convolution can be computed via multiplication in the frequency domain:

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

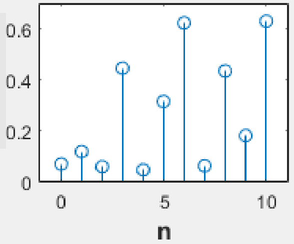
where $H(l)$ and $U(l)$ are the N-point DFTs of $h(n)$ and $u(n)$, $n = 0, 1, \ldots, N-1$, and F_D^{-1} denotes the inverse DFT operation. The convolution is cyclic which means that $y(k+iN)$ is equal to $y(k)$ as $u(n)$ is periodic.



Signal $u(n)$

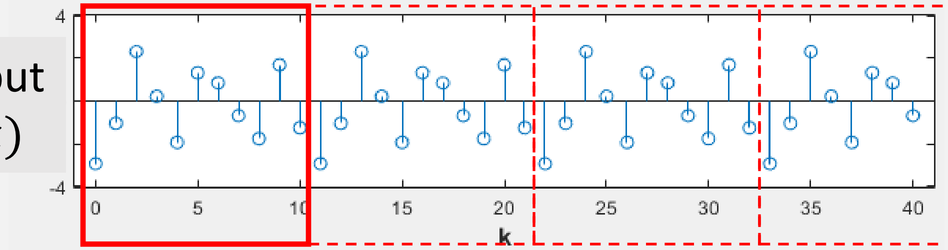


Filter $h(n)$



Example N = 11

Output $y(k)$



Correlation theorem

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

Definition of cross-correlation

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.

Similarity of correlation and convolution => same approaches can be applied to both.

But given assumptions (periodicity, filter/template length) do not hold in general => we cannot use DFT/IDFT directly

Two techniques needed:

(1) Zero padding makes cyclic correlation/convolution to output values of linear correlation/convolution.

(2) Sectioning allows one to compute linear correlation/convolution for long sequences.

2.2 Zero padding

Case of no padding

Consider computation of the cross-correlation (linear) of

$x_1(n)$ $\boxed{a_0 \quad a_1 \quad a_2}$

$x_2(n)$ non-periodic (b_0, b_1, b_2, 0, 0, 0, …)

The result wanted:

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 ..

Applying correlation theorem to (a_0, a_1, a_2) **and** (b_0, b_1, b_2)

$x_2(n)$ considered periodic (b_0, b_1, b_2, b_0, b_1, b_2, b_0, …)

$x_1(n)$ a_0	a_1	a_2	**lag** j	**cyclic correlation** $(\times 3)$
b_0	b_1	b_2	0	$\boxed{a_0b_0 + a_1b_1 + a_2b_2}$
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 ..

Only for lag j = 0, the cyclic correlation on the right correponds to the linear correlation on the left.

2.2 Zero padding

Pad two zeros to the template

Consider computation of the cross-correlation (linear) of

$x_1(n)$ $\boxed{a_0 \quad a_1 \quad a_2}$ $\boxed{0 \ \ 0}$

$x_2(n)$ non-periodic (b_0, b_1, b_2, 0, 0, 0, …)

Zeros do not change the result of linear cross-correlation (only the division factor 1/N is changed)

					lag j	linear correlation ($\times 5$)
a_0	a_1	a_2	0	0		
b_0	b_1	b_2	0	0	0	$a_0b_0 + a_1b_1 + a_2b_2$
b_1	b_2	0	0	0	1	$a_0b_1 + a_1b_2$
b_2	0	0	0	0	2	a_0b_2
0	0	0	0	0	3	all zero after this ..

Applying correlation theorem to (a_0, a_1, a_2, 0, 0) **and** (b_0, b_1, b_2, 0, 0)

$x_2(n)$ considered periodic (b_0, b_1, b_2, 0, 0, b_0, b_1, b_2, b_0, …)

						lag j	cyclic correlation ($\times 5$)
$x_1(n)$	a_0	a_1	a_2	0	0		
	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 ..

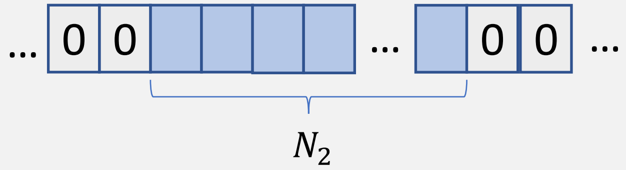
With zero padding, the values for lags 0, 1 and 2 match.

Now **for lags j = 0, 1, and 2**, the cyclic correlation on the right correponds to the linear correlation on the left.
Zero padding can be used to control, how cyclic correlation provides linear correlation values. Same holds for convolution.

Convolution with filter h : $y(k) = \sum_{n=0}^{N_1-1} h(n)u(k-n)$

N_1 is the filter length

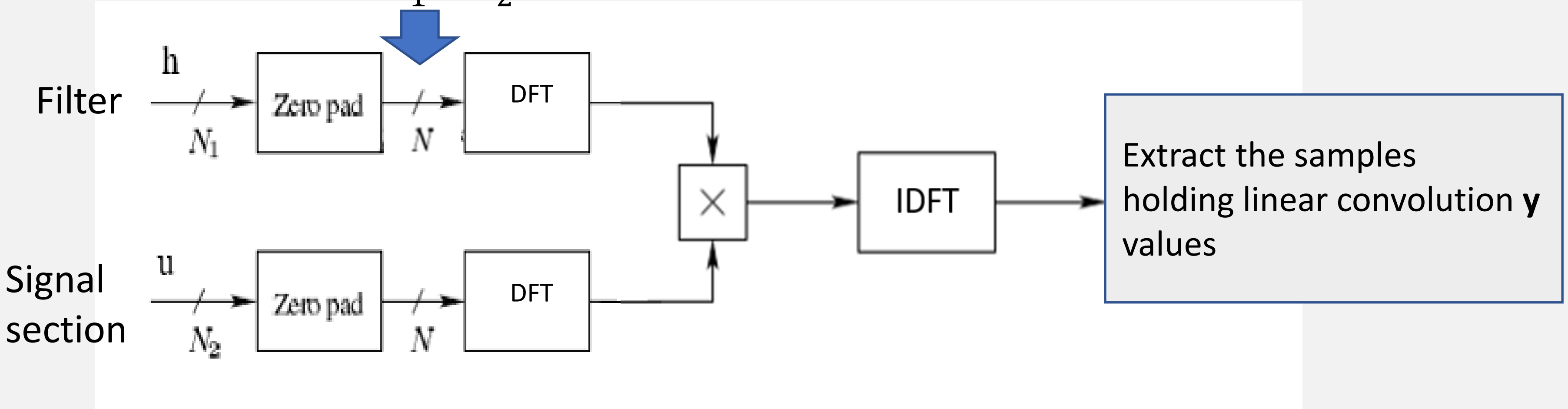
N_2 samples in $u(n)$, assume **surrounding values zero**



Signal section

Then the structure which provides **all** linear convolution values

$N \geq N_1 + N_2 - 1$



If $N < N_1 + N_2 - 1$, some linear convolution values are missing from the result.

Note. In the sectioning methods introduced in the following**:**

* **overlap-add** uses zero padding according to $N \geq N_1 + N_2 - 1$.
Addition compensates for the fact that surrounding values are non-zero.
* **overlap-save** uses $N = N_2$. Bad values from IDFT are neglected.

2.3. Sectioning

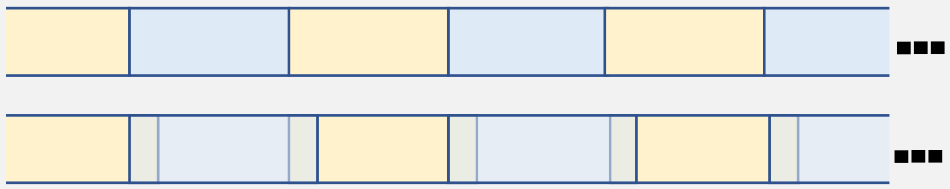
In many cases, **very long sequences** must be processed (consider a typical signal filtering application). If we consider application of frequency domain techniques, which process blocks of data, we must split the signal into smaller parts (sections).

There are two main strategies for doing sectioning: overlap-add and overlap-save.

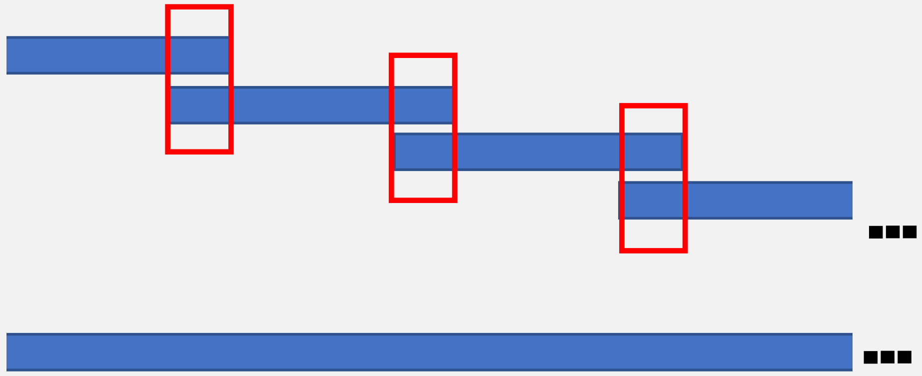
In both techniques, sections of the signal u(n) are extracted and used for evaluating cyclic convolutions. The final result is based on extracting valid linear convolution values. In overlap-add some addition operations are used to get linear convolution values.

Note: DFT of the zero-padded system response h(n) can be precomputed in both strategies. So, one DFT (for signal section) and one IDFT are needed at run-time.

Input sections: non-overlapping / overlapping



Cyclic convolution outputs:



Overlap-add strategy

The input signal u(n) is split into non-overlapping sections of length N_u.

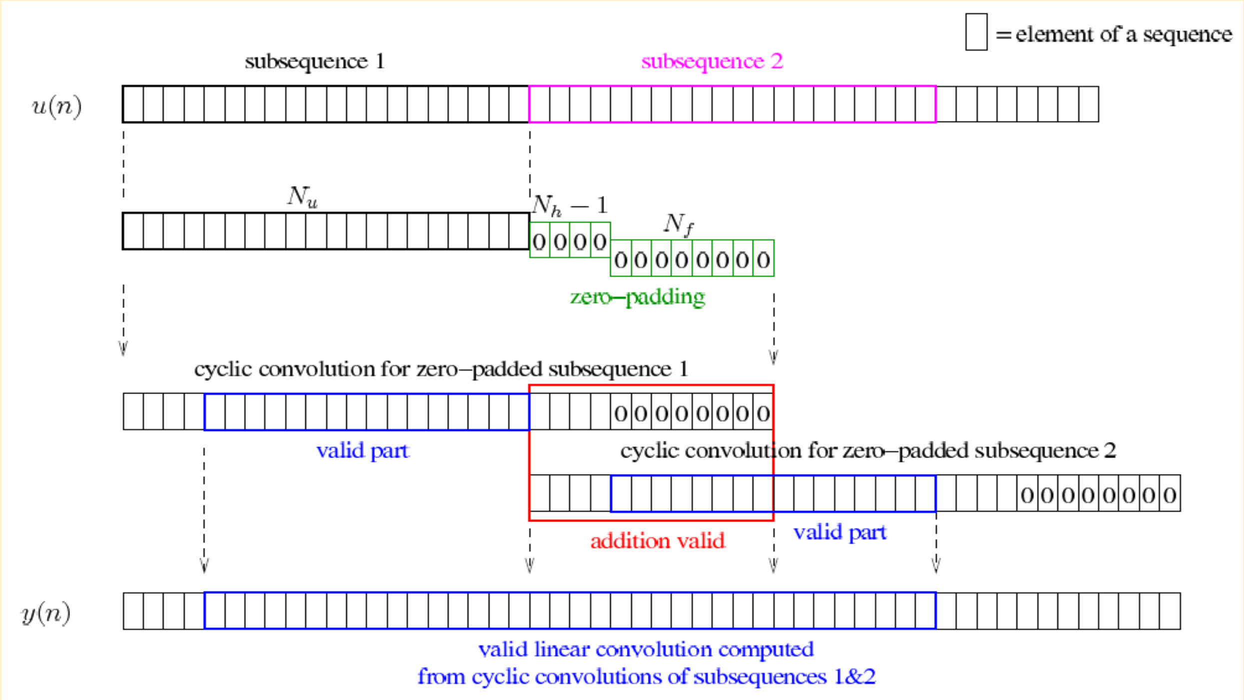
For cyclic convolution, a section and the system's impulse response h(n) are both padded by zeros so that they both have length equal to $N_u + N_h + N_f - 1$, where N_h is the length of h(n). The parameter $N_f \geq 0$ is an extra padding, which matches the cyclic convolution with the length required for applying an FFT algorithm.

The result of cyclic convolution is such that the samples correspond to the linear convolution except for the first $N_h - 1$ and last $N_h + N_f - 1$ samples.

However, if we overlap the cyclic convolution outputs by $N_h + N_f - 1$ samples and compute the sum of those overlapped portions we get a valid linear convolution.

Note: the last N_f samples of the cyclic convolution are zero, so actually addition is not needed for that part.

Sketch for N_u=20, N_h=5, N_f=8 :



Overlap-save strategy

The input signal u(n) is split into overlapping sections of length N_u which is equal to the length used for cyclic convolution.

The sections are overlapped by N_h -1 samples, where N_h is the length of system's impulse response h(n).

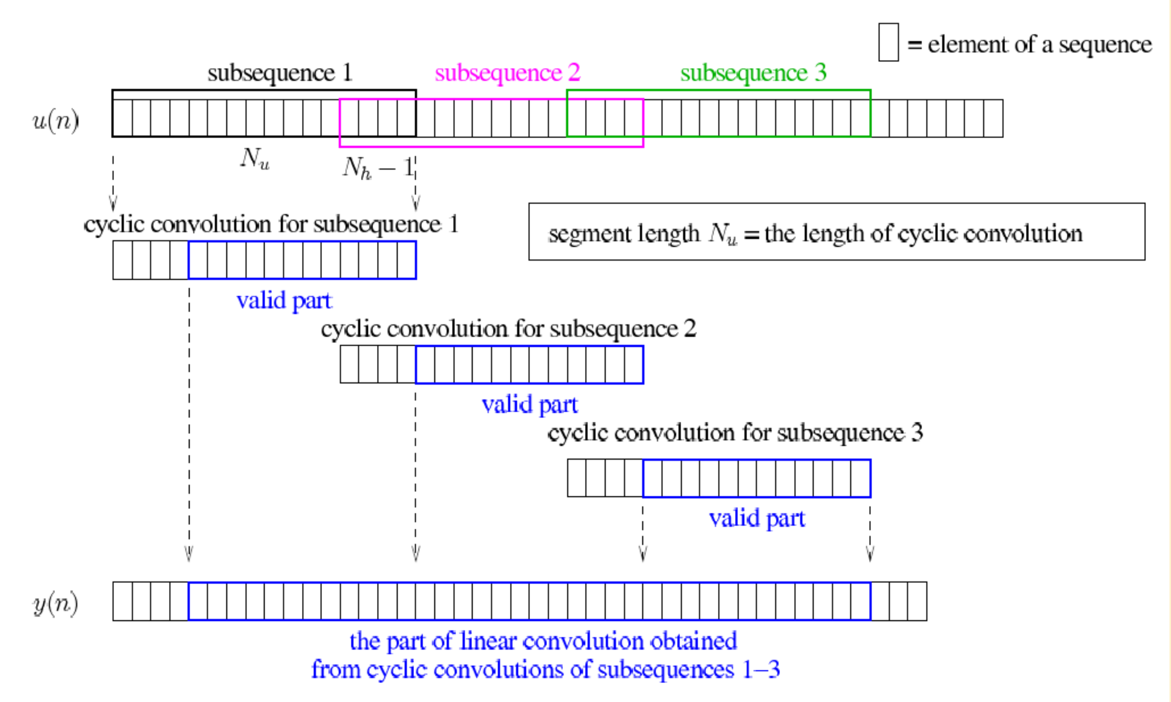
Only h(n) is zero-padded for cyclic convolution, by N_u-N_h zeros.

The result of cyclic convolution is such that the samples correspond to the linear convolution except for the first N_h -1 samples.

So, the last N_u-(N_h-1) samples of the cyclic convolution are included in the output stream ('saved'). As the input sections were overlapped by N_h -1 samples, we get a continuous stream of valid linear convolution outputs.

Note: overlap-save does not involve any extra addition operations, all computations are in the cyclic convolution implementation.

Sketch for N_u=16, N_h=5:



Using sectioning to evaluate cross-correlation

We can use the previous techniques by expressing the cross-correlation in terms of convolution.

Derivation of the relationship shown on the right.

From the actions done we note that :

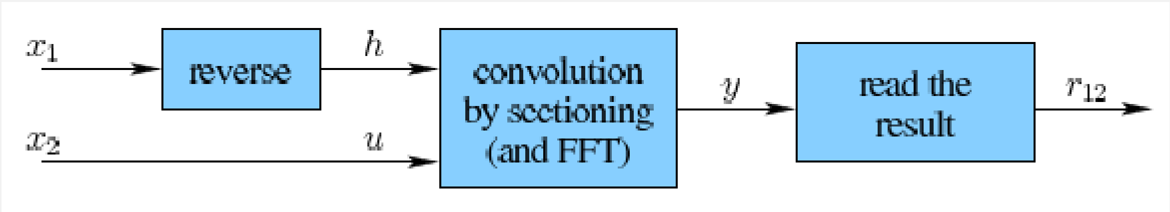
1. Convolution's $h()$ is set up by reversing $x_1()$ as

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

2. Convolution's $u()$ is set equal to $x_2()$

3. The cross-correlation for lag j is obtained from convolution output $y(k)$ at $k=j+N-1$:

$$r_{12}(j) = \frac{1}{N} y(j+N-1)$$



$$\begin{aligned}
r_{12}(j) \quad &= \quad \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n+j) \\
&\propto \quad \sum_{n=0}^{N-1} x_1(n)x_2(n+j) \\
&\overset{n=-n'}{=} \quad \sum_{-n'=0}^{N-1} x_1(-n')x_2(j-n') \\
1 \quad &\overset{x_1(-n')=s(n')}{=} \quad \sum_{n'=1-N}^{0} s(n')x_2(j-n') \\
&\overset{m=n'+N-1}{=} \quad \sum_{m=0}^{N-1} s(m-N+1)x_2(j-(m-N+1)) \\
1 \quad &\overset{h(m)=s(m-N+1)}{=} \quad \sum_{m=0}^{N-1} h(m)x_2(j-m+N-1) \\
3 \quad &\overset{k=j+N-1}{=} \quad \sum_{m=0}^{N-1} h(m)x_2(k-m) \\
2 \quad &\overset{x_2(m)=u(m)}{=} \quad \underbrace{\sum_{m=0}^{N-1} h(m)u(k-m),}_{\text{convolution } y(k)}
\end{aligned}$$

Using FFT in evaluations

FFT is available for powers of 2. **Zero padding used so that $N = 2^k$ reached.**

Convolution:

System: $h(n)$

Signal (section): $u(n)$

Cross correlation:

Template: $x_1(n)$

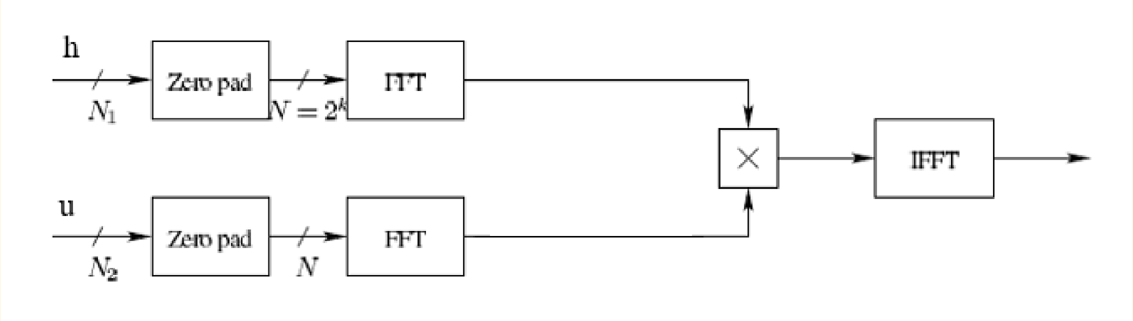
Signal (section): $x_2(n)$

- two structures: Complex conjugation can be used as a substitute for time reversal.

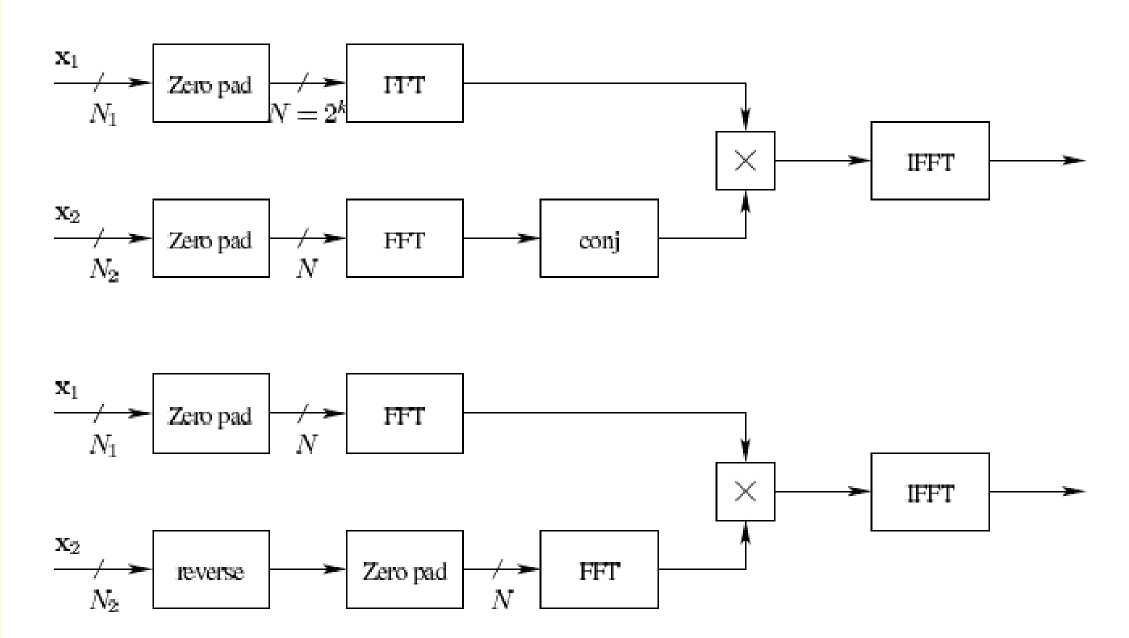
In addition: By definition, if complex signals, one must take conjugate of other and use it as an input

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

Convolution



Cross correlation



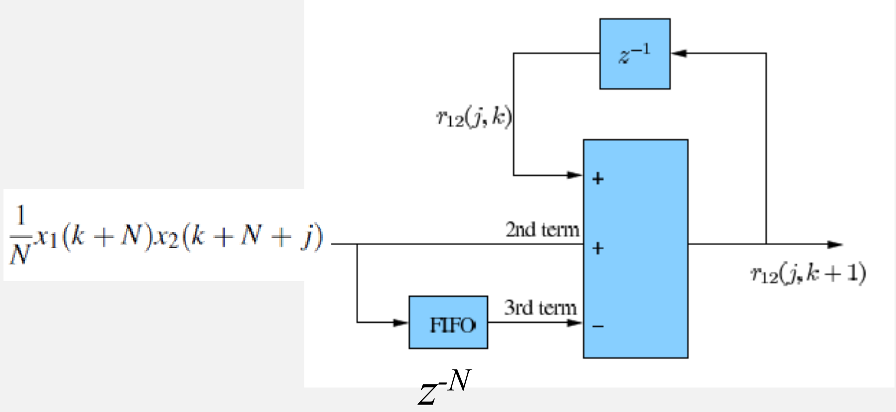
3. Recursive implementation of cross-correlation

A technique for measuring the lag ***j*** between two correlating signals.

The other signal is a source for the template x_1, which is <u>changed at each time instant</u> ***k***. The template is not a fixed one.

Recursive computation is implemented for each considered lag *j* separately. Basis for that:

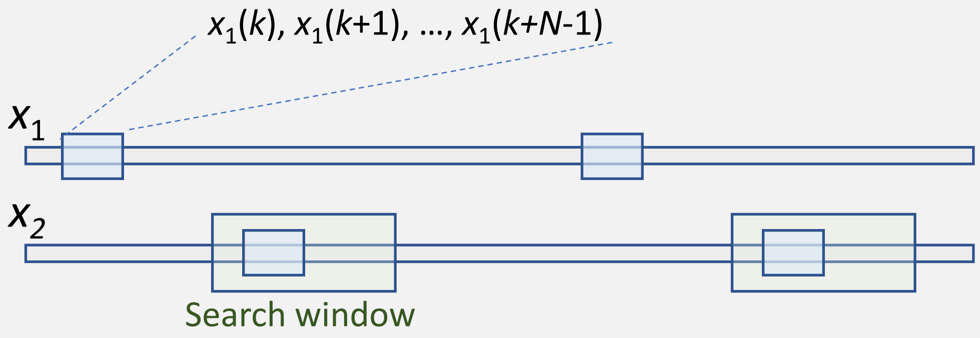
$$\begin{aligned} r_{12}(j,k+1) &= \frac{1}{N}\sum_{n=k+1}^{k+N} x_1(n)x_2(n+j) \\ &= r_{12}(j,k) + \frac{1}{N}x_1(k+N)x_2(k+N+j) - \frac{1}{N}x_1(k)x_2(k+j) \end{aligned}$$



Used definition:

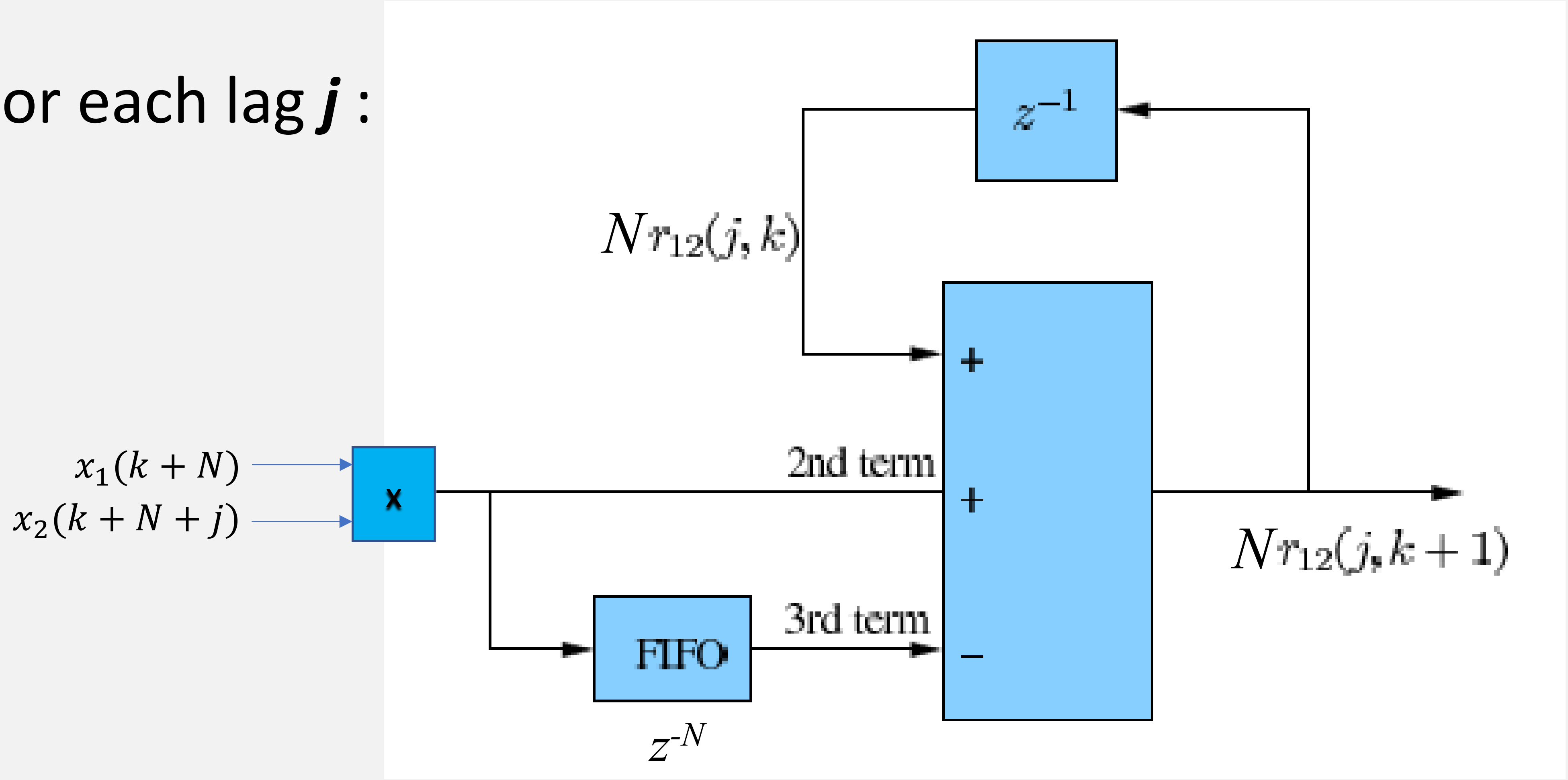
$$r_{12}(j,k) = \frac{1}{N}\sum_{n=k}^{k+N-1} x_1(n)x_2(n+j)$$

Used template at time instant *k*:



In practice, divisions by N are not necessary, we may use cross-correlation scaled by N.

For each lag $\boldsymbol{j}$:



If there are M lags to evaluate then each time instant k requires M multiplications, M additions and M subtractions.

Summary

- Concepts: cross-correlation, auto-correlation, convolution
- Different kinds of correlation measures are used in applications
- Basis for implementing **linear** convolutions/correlations in frequency domain was provided. An important approach, when convolution filter/correlation template is long.***
- Recursive implementation of correlation can be used in continuous estimation of changing lag between two signals.

****** In Lecture #11, efficient implementation of DFT, FFT, will be discussed. It will be demonstrated when the frequency domain provides advantage in implementations.*