Discrete Fourier Transform

Lecture 4

The Fourier Transform kingdom

h(u) / H(ω)	Non-Periodic	Periodic
Non-Periodic	(Continuous Time) Fourier Transform	Fourier Series
Periodic	Discrete Time Fourier Transform (DTFT)	Discrete Fourier Transform (DFT/FFT)

Discrete Fourier Transform

The discrete-time Fourier transform (DTFT) of a sequence is a continuous function of ω, and repeats with period 2π. In practice we usually want to obtain the Fourier components using digital computation, and can only evaluate them for a discrete set of frequencies. The discrete Fourier transform (DFT) provides a means for achieving this.

The DFT is itself a sequence, and it corresponds roughly to samples, equally spaced in frequency, of the Fourier transform of the signal. The discrete Fourier transform of a length N signal $x[n], n = 0, 1, \ldots, N-1$ is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn}.$$

This is the analysis equation.

iDFT (synthesis equation)

The corresponding synthesis equation is

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}.$$

When dealing with the DFT, it is common to define the complex quantity

$$W_N = e^{-j(2\pi/N)}.$$

Analysis and Synthesis equations

With this notation the DFT analysis-synthesis pair becomes

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}.$$

An important property of the DFT is that it is cyclic, with period N, both in the discrete-time and discrete-frequency domains. For example, for any integer r,

$$X[k + rN] = \sum_{n=0}^{N-1} x[n] W_N^{(k+rN)n} = \sum_{n=0}^{N-1} x[n] W_N^{kn} (W_N^N)^{rn}$$

$$= \sum_{n=0}^{N-1} x[n] W_N^{kn} = X[k],$$

Numerical examples

- The DFT for N samples can be obtained as the multiplication of the N samples by the W matrix

$$W = W_N^{kn}\Big|_{k,n=0,\ldots,N-1}$$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-1)} \\ 1 & W_N^3 & W_N^6 & W_N^9 & \cdots & W_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & W_N^{3(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

DFT examples

- N=2

$$W = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- N=4

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

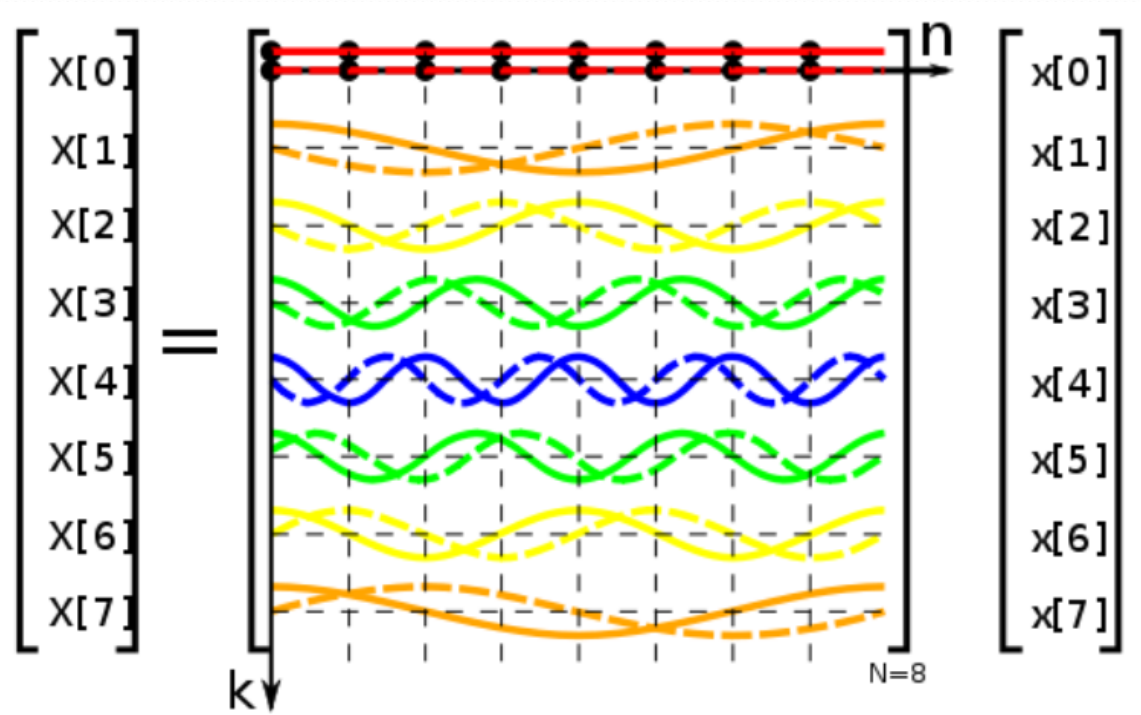
- N=8

$$W = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & \cdots & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & \cdots & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & \cdots & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & \cdots & W_8^{21} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & \cdots & W_8^{49} \end{bmatrix}$$

$$W_8 = e^{-j\frac{2\pi}{8}} = \\ = \frac{1}{\sqrt{2}} - \frac{j}{\sqrt{2}}$$

DFT for N=8

- The DFT for N samples seen as the projection on N complex exponential sequences



Example

- Consider a periodic signal whose period is [0 1 2 3]

Filtered by a FIR filter $y(n) = x(n) - x(n-1)$

If we want to get the DFT of the input signal we can write:

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

Example (cont.)

- While the DFT of the filter will be:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1+j \\ 2 \\ 1-j \end{bmatrix}$$

- And the product of each element of H by the corresponding element of X will be:

$$Y = \begin{bmatrix} 0 \\ -4 \\ -4 \\ -4 \end{bmatrix}$$

Inverse DFT

- The inverse of the W matrix will be equal to its conjugate transpose divided by N, for example for N=4

$$W^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

- The iDFT for the previous example will then be:

$$y = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 0 \\ -4 \\ -4 \\ -4 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example (cont.)

- The previous result is exactly what we would obtain using the convolution of the periodic signal x(n) with h(n):

  the circular convolution will give:

  0,1,2,3,0,1,2,3,0,1,2,3, 0,1,2,3,0,1,2,3 with [1, -1] =

  = [-3, 1, 1, 1].

Periodicity assumption

since $W_N^N = e^{-j(2\pi/N)N} = e^{-j2\pi} = 1$. Similarly, it is easy to show that $x[n + rN] = x[n]$, implying periodicity of the synthesis equation. This is important — even though the DFT only depends on samples in the interval 0 to $N - 1$, it is implicitly assumed that the signals repeat with period N in both the time and frequency domains.

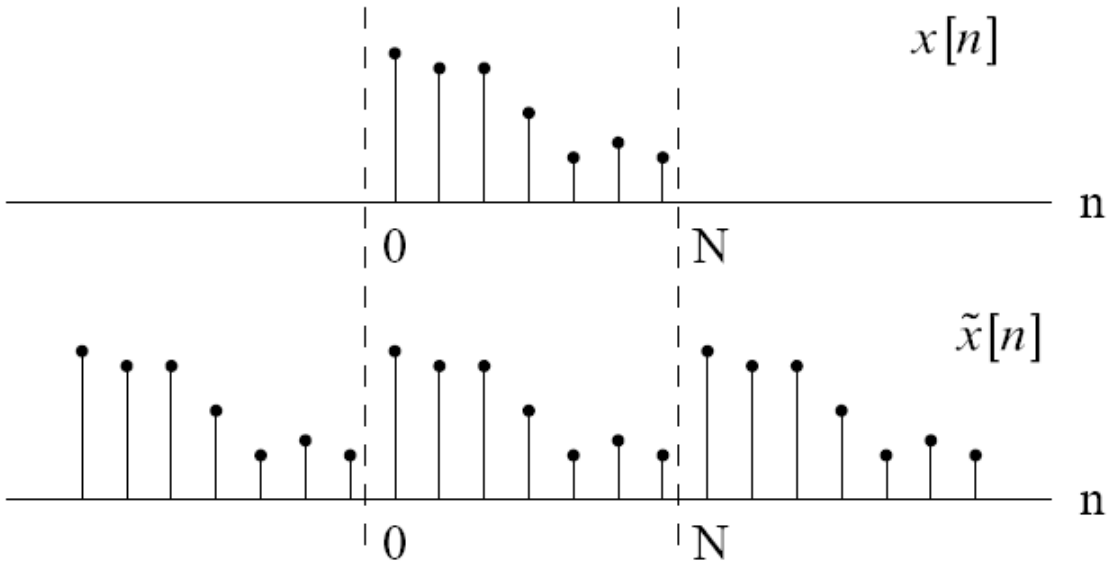
Periodic extension

To this end, it is sometimes useful to define the periodic extension of the signal $x[n]$ to be

$$\tilde{x}[n] = x[n \bmod N] = x[((n))_N].$$

Here $n \bmod N$ and $((n))_N$ are taken to mean n modulo N, which has the value of the remainder after n is divided by N. Alternatively, if n is written in the form $n = kN + l$ for $0 \leq l < N$, then

$$n \bmod N = ((n))_N = l.$$



Periodicity in the frequency domain

Similarly, the periodic extension of $X[k]$ is defined to be

$$\tilde{X}[k] = X[k \bmod N] = X[((k))_N].$$

It is sometimes better to reason in terms of these periodic extensions when dealing with the DFT. Specifically, if $X[k]$ is the DFT of $x[n]$, then the inverse DFT of $X[k]$ is $\tilde{x}[n]$. The signals $x[n]$ and $\tilde{x}[n]$ are identical over the interval 0 to $N-1$, but may differ outside of this range. Similar statements can be made regarding the transform $X[k]$.

Properties of the DFT

Many of the properties of the DFT are analogous to those of the discrete-time Fourier transform, with the notable exception that all shifts involved must be considered to be circular, or modulo N.

Defining the DFT pairs $x[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, $x_1[n] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k]$, and $x_2[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, the following are properties of the DFT:

- **Symmetry:**

$$X[k] = X^*[((-k))_N]$$
$$\mathrm{Re}\{X[k]\} = \mathrm{Re}\{X[((-k))_N]\}$$
$$\mathrm{Im}\{X[k]\} = -\mathrm{Im}\{X[((-k))_N]\}$$
$$|X[k]| = |X[((-k))_N]|$$
$$\sphericalangle X[k] = -\sphericalangle X[((-k))_N]$$

Properties of the DFT

- **Linearity:** $ax_1[n] + bx_2[n] \overset{\mathcal{D}}{\longleftrightarrow} aX_1[k] + bX_2[k]$.
- **Circular time shift:** $x[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} W_N^{km} X[k]$.
- **Circular convolution:**

$$\sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k]X_2[k].$$

Circular convolution between two N-point signals is sometimes denoted by $x_1[n] \circledast x[n]$.

- **Modulation:**

$$x_1[n]x_2[n] \overset{\mathcal{D}}{\longleftrightarrow} \frac{1}{N} \sum_{l=0}^{N-1} X_1[l]X_2[((k-l))_N].$$

Time shift property

Some of these properties, such as linearity, are easy to prove. The properties involving time shifts can be quite confusing notationally, but are otherwise quite simple. For example, consider the 4-point DFT

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{kn}$$

of the length 4 signal $x[n]$. This can be written as

$$X[k] = x[0]W_4^{0k} + x[1]W_4^{1k} + x[2]W_4^{2k} + x[3]W_4^{3k}$$

The product $W_4^{1k} X[k]$ can therefore be written as

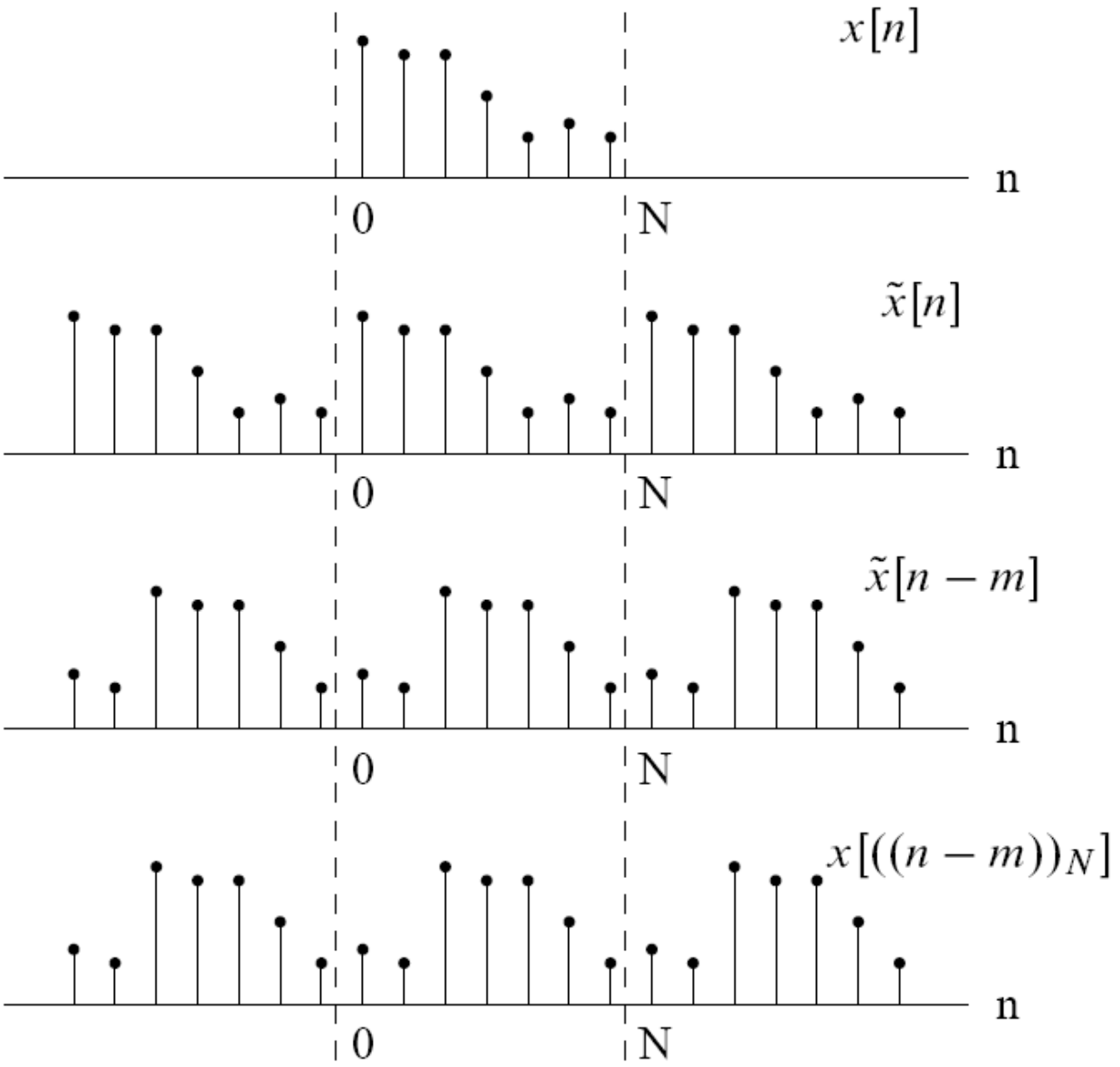
$$\begin{aligned} W_4^{1k} X[k] &= x[0]W_4^{1k} + x[1]W_4^{2k} + x[2]W_4^{3k} + x[3]W_4^{4k} \\ &= x[3]W_4^{0k} + x[0]W_4^{1k} + x[1]W_4^{2k} + x[2]W_4^{3k} \end{aligned}$$

On the circular time shift property

since $W_4^{4k} = W_4^{0k}$. This can be seen to be the DFT of the sequence $x[3], x[0], x[1], x[2]$, which is precisely the sequence $x[n]$ *circularly shifted* to the right by one sample. This proves the time-shift property for a shift of length 1. In general, multiplying the DFT of a sequence by W_N^{km} results in an N-point *circular* shift of the sequence by m samples. The convolution properties can be similarly demonstrated.

It is useful to note that the circularly shifted signal $x[((n-m))_N]$ is the same as the linearly shifted signal $\tilde{x}[n-m]$, where $\tilde{x}[n]$ is the N-point periodic extension of $x[n]$.

Periodicity



Circular (periodic) convolution

On the interval 0 to $N-1$, the circular convolution

$$x_3[n] = x_1[n] \circledast x_2[n] = \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

can therefore be calculated using the linear convolution product

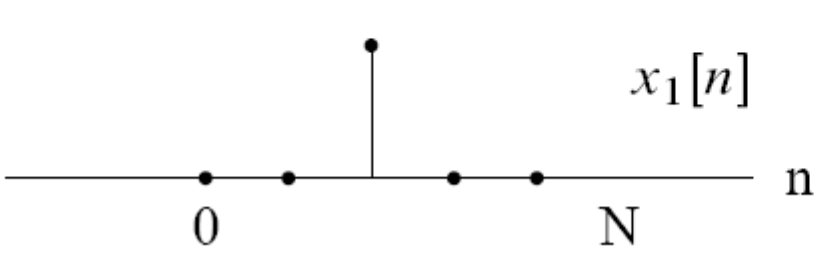
$$x_3[n] = \sum_{m=0}^{N-1} x_1[m] \tilde{x}_2[n-m].$$

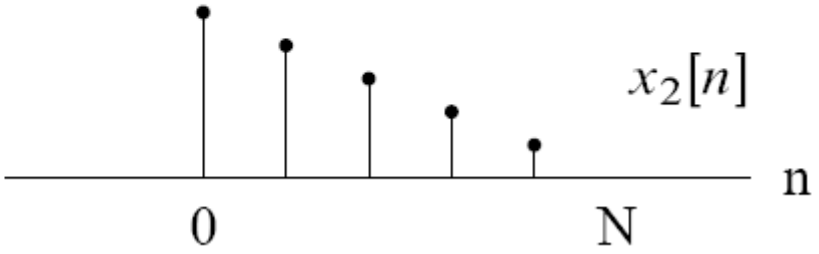
Circular convolution is really just periodic convolution.

Circular convolution with a delayed impulse sequence

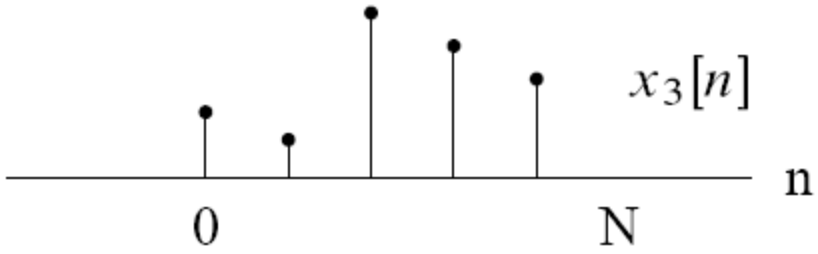
Example: Circular convolution with a delayed impulse sequence

Given the sequences





the circular convolution $x_3[n] = x_1[n] \circledast x_2[n]$ is the signal $\tilde{x}[n]$ delayed by two samples, evaluated over the range 0 to $N-1$:



Circular convolution of two rectangular pulses

Let

$$x_1[n] = x_2[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise.} \end{cases}$$

If $N = L$, then the N-point DFTs are

$$X_1[k] = X_2[k] = \sum_{n=0}^{N-1} W_N^{kn} = \begin{cases} N & k = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Since the product is

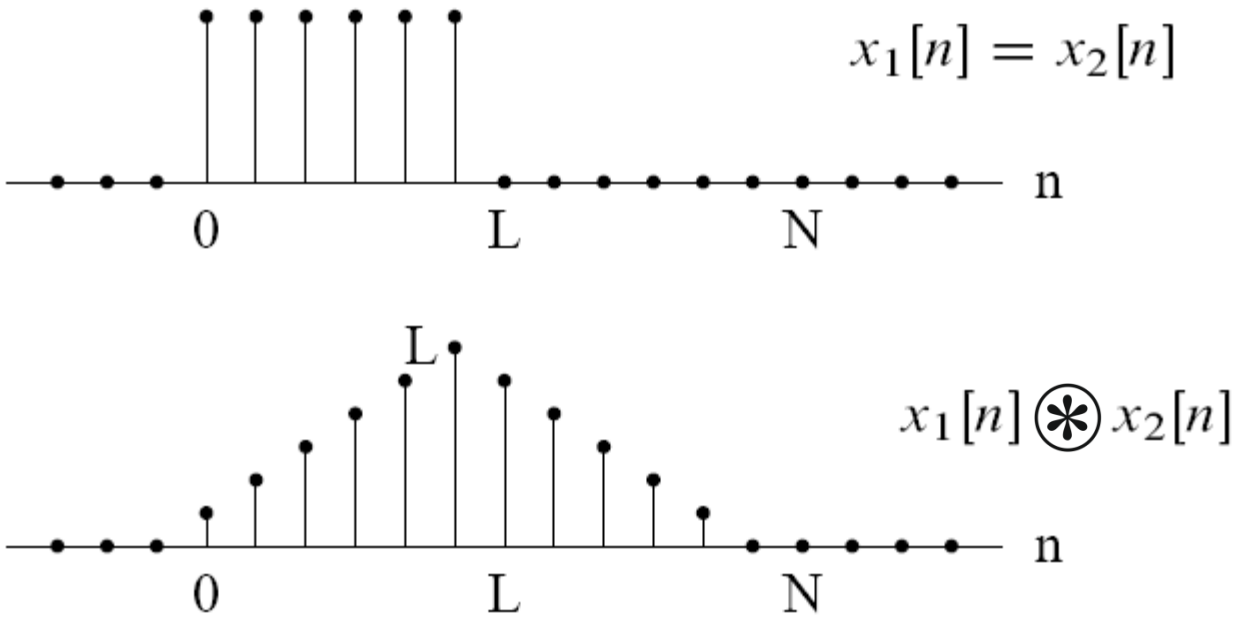
$$X_3[k] = X_1[k]X_2[k] = \begin{cases} N^2 & k = 0 \\ 0 & \text{otherwise,} \end{cases}$$

it follows that the N-point circular convolution of $x_1[n]$ and $x_2[n]$ is

$$x_3[n] = x_1[n] \circledast x_2[n] = N, \qquad 0 \le n \le N-1.$$

Circular convolution of two rectangular pulses

Suppose now that $x_1[n]$ and $x_2[n]$ are considered to be length $2L$ sequences by augmenting with zeros. The $N = 2L$-point circular convolution is then seen to be the same as the linear convolution of the finite-duration sequences $x_1[n]$ and $x_2[n]$:



Linear convolution using the DFT

Using the DFT we can compute the circular convolution as follows

- Compute the N-point DFTs $X_1[k]$ and $X_2[k]$ of the two sequences $x_1[n]$ and $x_2[n]$.
- Compute the product $X_3[k] = X_1[k]X_2[k]$ for $0 \leq k \leq N-1$.
- Compute the sequence $x_3[n] = x_1[n] \circledast x_2[n]$ as the inverse DFT of $X_3[k]$.

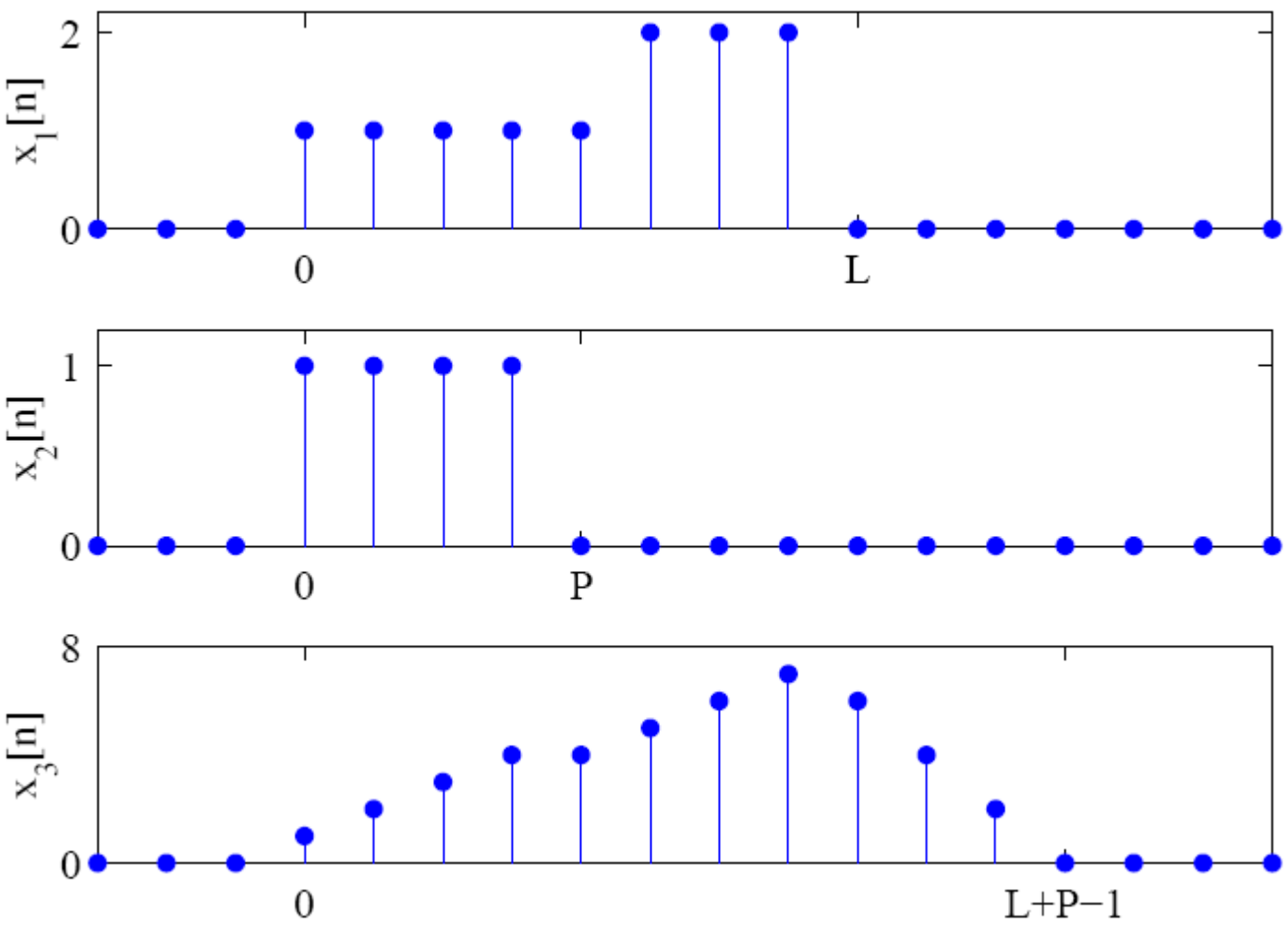
This is computationally useful due to efficient algorithms for calculating the DFT. The question that now arises is this: how do we get the *linear* convolution (required in speech, radar, sonar, image processing) from this procedure?

Linear convolution of two finite-length sequences

Consider a sequence $x_1[n]$ with length L points, and $x_2[n]$ with length P points. The linear convolution of the sequences,

$$x_3[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m],$$

is nonzero over a maximum length of $L + P - 1$ points:


2
$x_1[n]$
0
0
L
1
$x_2[n]$
0
0
P
8
$x_3[n]$
0
0
L+P−1

Linear convolution of two finite-length sequences

Therefore $L + P - 1$ is the maximum length of $x_3[n]$ resulting from the linear convolution.

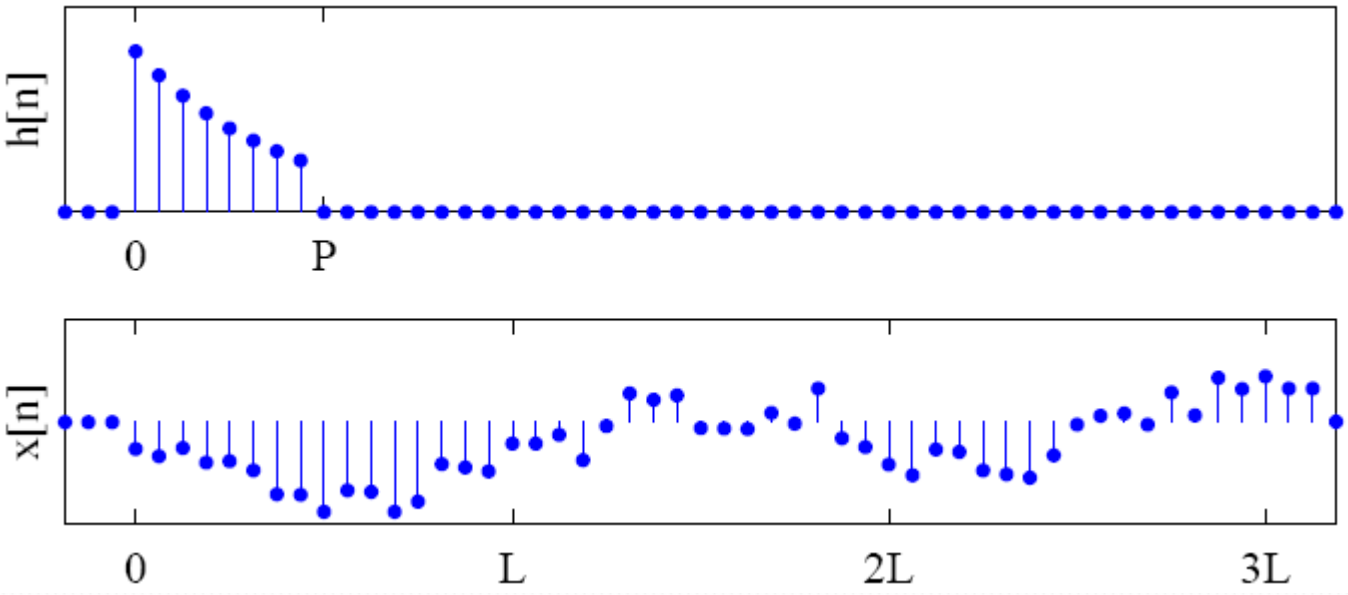
The N-point circular convolution of $x_1[n]$ and $x_2[n]$ is

$$x_1[n] \circledast x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] = \sum_{m=0}^{N-1} x_1[m]\tilde{x}_2[n-m] :$$

It is easy to see that the circular convolution product will be equal to the linear convolution product on the interval 0 to $N - 1$ as long as we choose $N \geq L + P - 1$. The process of augmenting a sequence with zeros to make it of a required length is called **zero padding**.

Convolution by sectioning (OLA, Overlap and Add)

Suppose that for computational efficiency we want to implement a FIR system using DFTs. It cannot in general be assumed that the input signal has a finite duration, so the methods described up to now cannot be applied directly:



Overlap and Add (Cont.)

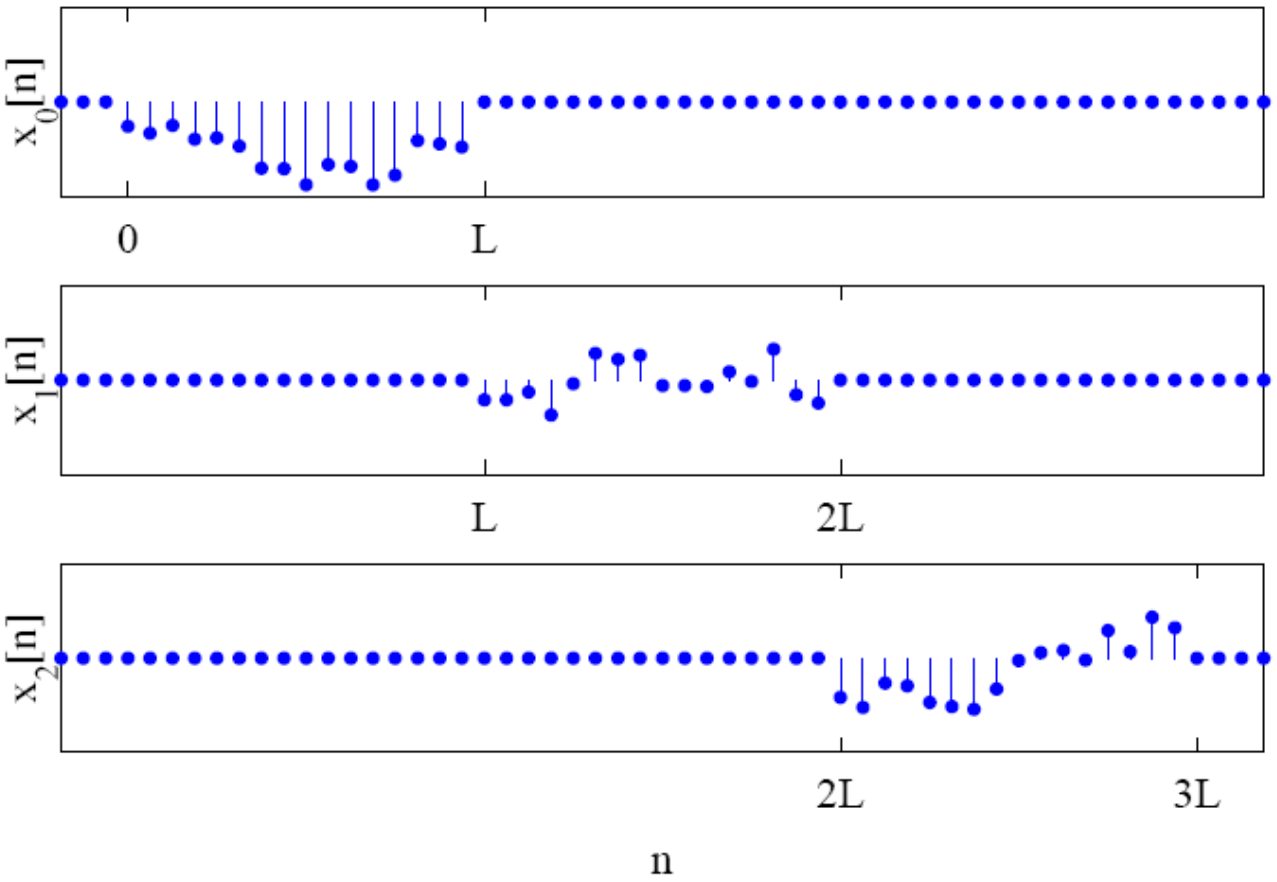
The solution is to use **block convolution**, where the signal to be filtered is segmented into sections of length L. The input signal $x[n]$, here assumed to be causal, can be decomposed into blocks of length L as follows:

$$x[n] = \sum_{r=0}^{\infty} x_r[n - rL],$$

where

$$x_r[n] = \begin{cases} x[n + rL] & 0 \le n \le L - 1 \\ 0 & \text{otherwise.} \end{cases}$$

Overlap and Add (Cont.)

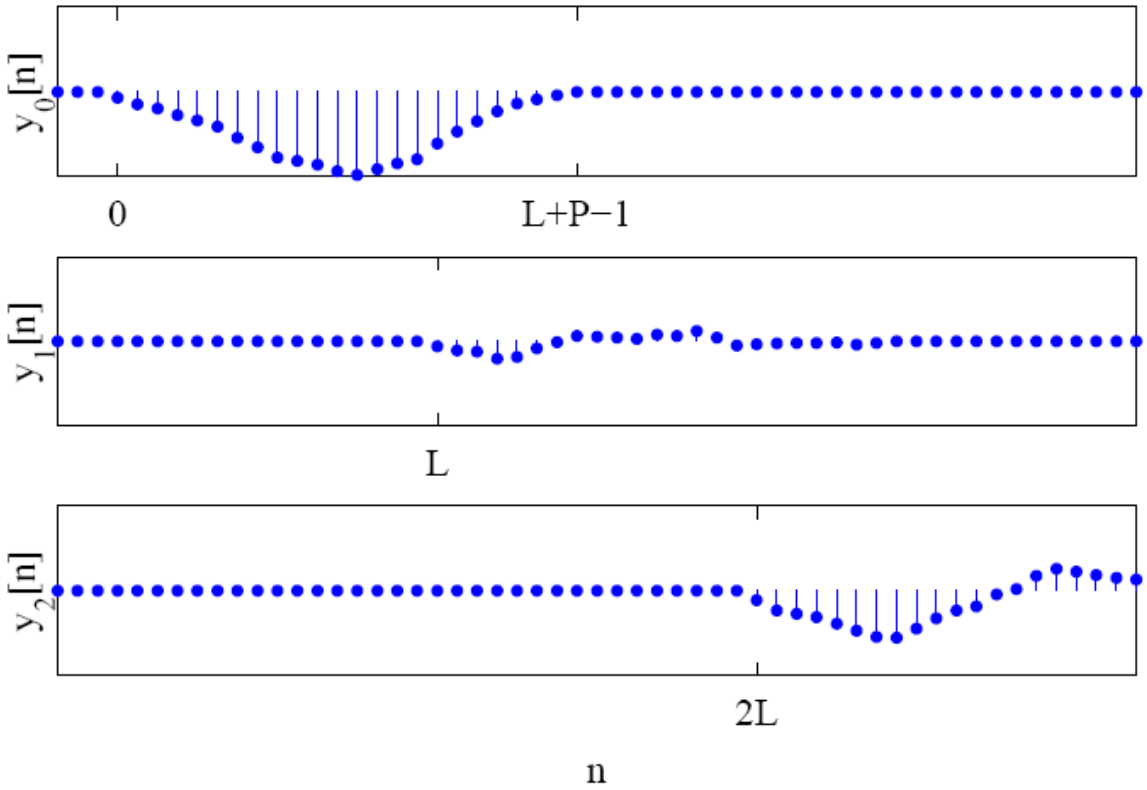


The convolution product can therefore be written as

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} y_r[n - rL],$$

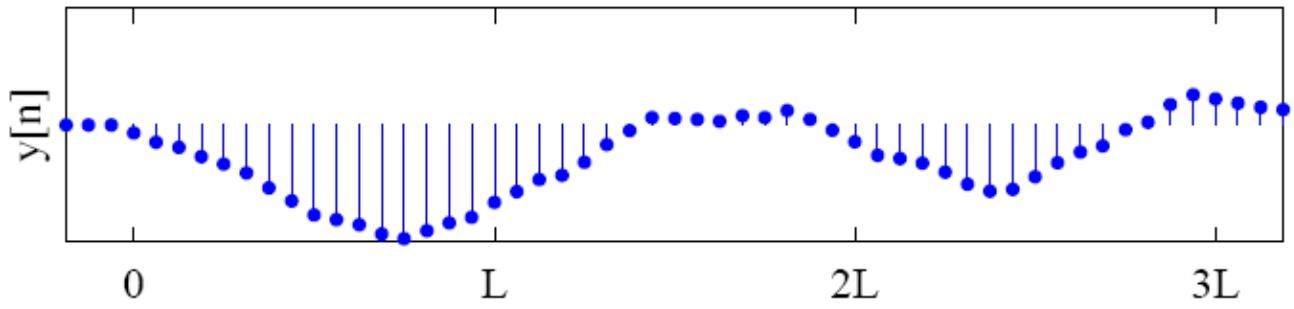
where $y_r[n]$ is the response

$$y_r[n] = x_r[n] * h[n].$$



Overlap and Add (Cont.)

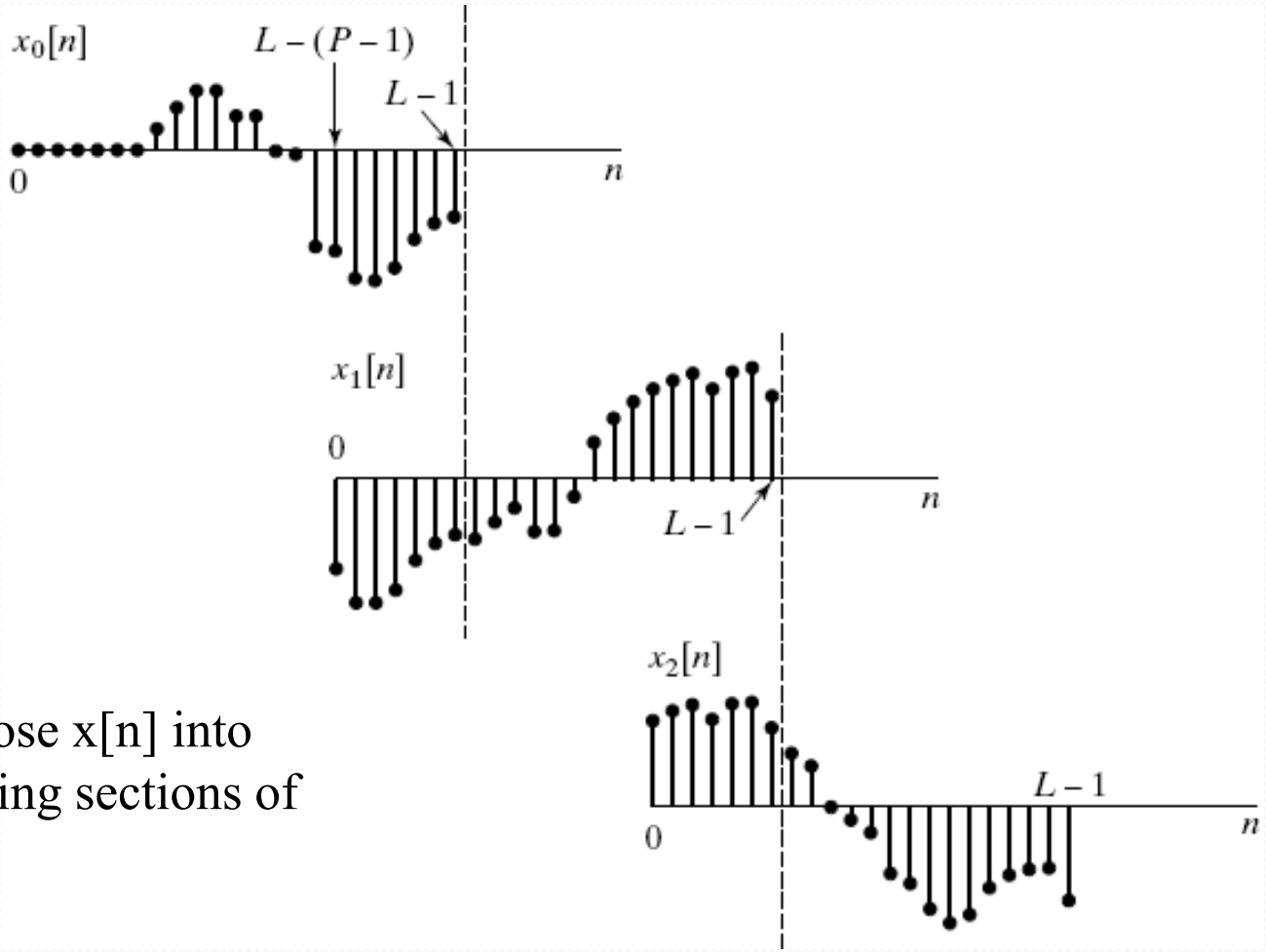
Since the sequences $x_r[n]$ have only L nonzero points and $h[n]$ is of length P, each response term $y_r[n]$ has length $L + P - 1$. Thus linear convolution can be obtained using N-point DFTs with $N \geq L + P - 1$. Since the final result is obtained by summing the overlapping output regions, this is called the **overlap-add** method.



Overlap and Save

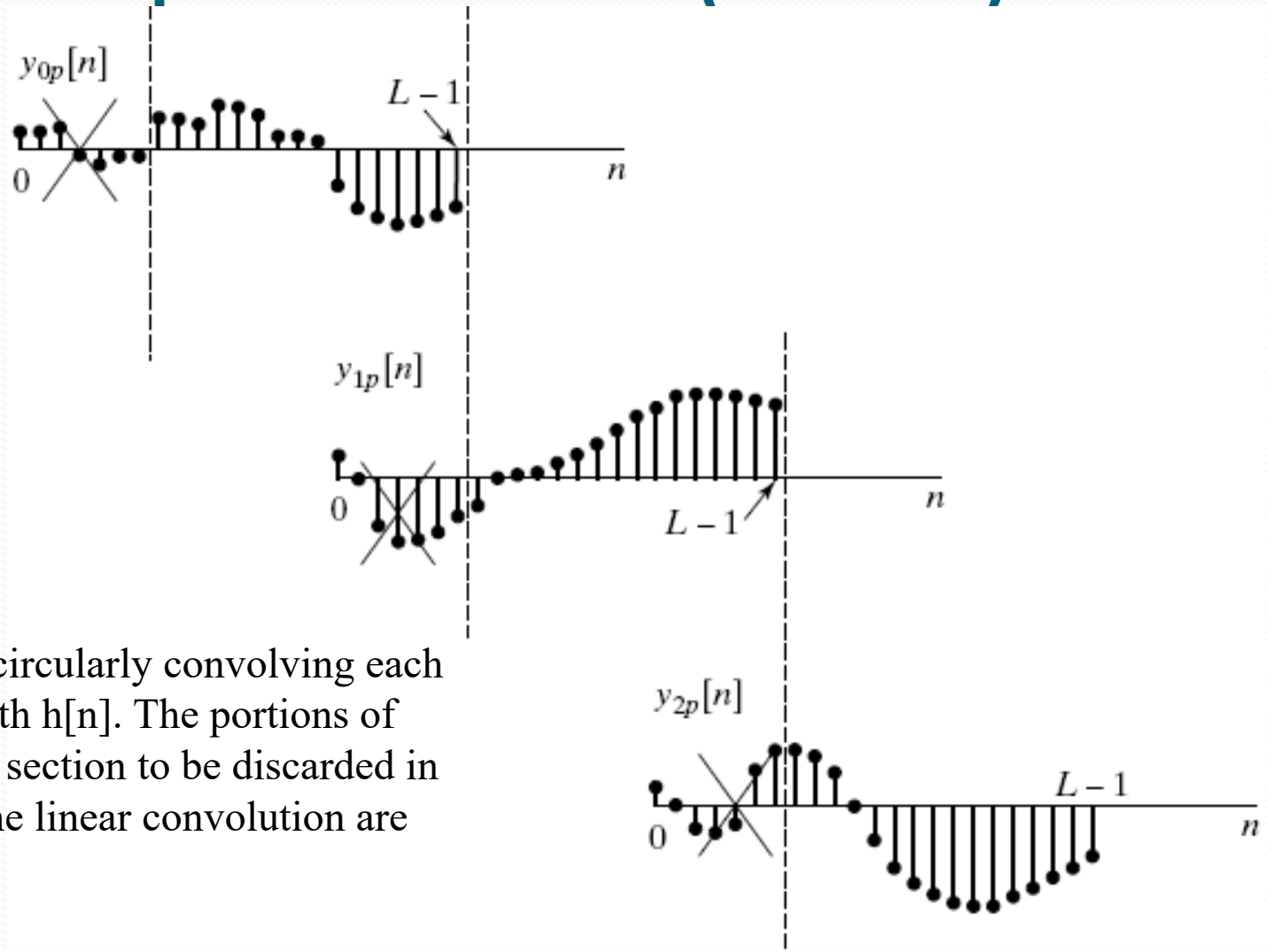
An alternative block convolution procedure, called the **overlap-save** method, corresponds to implementing an L-point circular convolution of a P-point impulse response $h[n]$ with an L-point segment $x_r[n]$. The portion of the output that corresponds to linear convolution is then identified (consisting of $L-(P-1)$ points), and the resulting segments patched together to form the output.

Overlap and Save (cont.)



Decompose x[n] into overlapping sections of length L

Overlap and Save (Cont.)



Result of circularly convolving each section with h[n]. The portions of each filter section to be discarded in forming the linear convolution are indicated