ATME College of Engineering

SIGNALS AND DIGITAL SIGNAL PROCESSING BEE502

Module – 2

Discrete Fourier Transforms (DFT): Introduction to DFT, definition of DFT and its inverse, matrix relation to find DFT and IDFT ,Properties of DFT, linearity, circular time shift, circular frequency shift, circular folding, symmetry of : real valued sequences, real even and odd sequences, DFT of complex conjugate sequence, multiplication of two DFTs- the circular convolution, Parseval's theorem, circular correlation, Digital linear filtering using DFT. Signal segmentation , overlap-save and overlap-add method

Bloom's Taxonomy Level	L1 – Remembering, L2 – Understanding, L3 – Applying, L – 4 Analysing,

Discrete-Time Systems

- Discrete-Time Sequence is a mathematical operation that maps a given input sequence x[n] into an output sequence y[n]

$$y[n] = T\{x[n]\}$$

$$x[n] \longrightarrow \boxed{T\{.\}} \longrightarrow y[n]$$

- Example Discrete-Time Systems
  - Moving (Running) Average

$$y[n] = x[n] + x[n-1] + x[n-2] + x[n-3]$$

  - Maximum

$$y[n] = \max\{x[n], x[n-1], x[n-2]\}$$

  - Ideal Delay System

$$y[n] = x[n-n_o]$$

Linearity

A **linear** system is one that obeys the principle of superposition,

$$T\{a_1 x[n] + a_2 x[n]\} = a_1 y_1[n] + a_2 y_2[n]$$

where the output of a linear combination of inputs is the same linear combination applied to the individual outputs. This result means that a complicated system can be decomposed into a linear combination of elementary functions whose transformation is known, and then taking the same linear combination of the results. Linearity also implies that the behavior of the system is independent of the magnitude of the input.

Linear Systems

- Linear System: A system is linear if and only if

$$T\{x_1[n] + x_2[n]\} = T\{x_1[n]\} + T\{x_2[n]\} \quad \text{(additivity)}$$

and

$$T\{ax[n]\} = aT\{x[n]\} \quad \text{(scaling)}$$

- Examples
  - Ideal Delay System

$$y[n] = x[n - n_o]$$

$$T\{x_1[n] + x_2[n]\} = x_1[n - n_o] + x_2[n - n_o]$$
$$T\{x_2[n]\} + T\{x_1[n]\} = x_1[n - n_o] + x_2[n - n_o]$$
$$T\{ax[n]\} = ax_1[n - n_o]$$
$$aT\{x[n]\} = ax_1[n - n_o]$$

Time (Shift) Invariance

A system is said to be **shift invariant** if the only effect caused by a shift in the position of the input is an equal shift in the position of the output, that is

$$T\{x[n-n_0]\} = y[n-n_0]$$

The magnitude and shape of the output are unchanged, only the location of the output is changed.

Time-Invariant Systems

- Time-Invariant (shift-invariant) Systems
  - A time shift at the input causes corresponding time-shift at output

$$y[n] = T\{x[n]\} \Rightarrow y[n - n_o] = T\{x[n - n_o]\}$$

- Example
  - Square

$$y[n] = (x[n])^2$$

Delay the input the output is $y_1[n] = (x[n - n_o])^2$

Delay the output gives $y[n - n_o] = (x[n - n_o])^2$

- Counter Example
  - Compressor System

$$y[n] = x[Mn]$$

Delay the input the output is $y_1[n] = x[Mn - n_o]$

Delay the output gives $y[n - n_o] = x[M(n - n_o)]$

Impulse Response

When the input to a system is a single impulse, the output is called the **impulse response**. Let $h[n]$ be the impulse response, given by

$$T\{\delta[n]\} = h[n]$$

A general sequence $f[x]$ can be represented as a linear combination of impulses, since

$$f(x) = f(x) * \delta(x) = \int_{-\infty}^{\infty} f(u)\delta(x-u)\,du$$

$$f[n] = f[n] * \delta[n] = \sum_{k=-\infty}^{\infty} f[k]\delta[n-k]$$

Linear Shift-Invariant Systems

Suppose that T{} is a linear, shift-invariant system with $h[n]$ as its impulse response.

Then, using the principle of superposition,

$$T\{s[n]\} = T\left\{\sum_{k=-\infty}^{\infty} s[k]\delta[n-k]\right\} = \sum_{k=-\infty}^{\infty} s[k]T\{\delta[n-k]\}$$

and finally after invoking shift-invariance

$$T\{s[n]\} = \sum_{k=-\infty}^{\infty} s[k]T\{\delta[n-k]\} = \sum_{k=-\infty}^{\infty} s[k]h[n-k]$$

$$T\{s[n]\} = s[n] * h[n]$$

This very important result says that the output of any linear, shift-invariant system is given by the convolution of the input with the impulse response of the system.

Causality

A system is *causal* if, for every choice of n_0, the output sequence at the index $n = n_0$ depends only on the input sequence values for $n \leq 0$.

All physical time-based systems are causal because they are unable to look into the future and anticipate a signal value that will occur later.

Causal System

- Causality

A system is causal it's output is a function of only the current and previous samples

- Examples
  - Backward Difference

$$y[n] = x[n] - x[n-1]$$

- Counter Example
  - Forward Difference

$$y[n] = x[n+1] + x[n]$$

Stability

A system is stable in the bounded-input, bounded-output (BIBO) sense if and only it every bounded input produces a bounded output sequence.

The input $x[n]$ is bounded if there exists a fixed positive finite value B_x such that

$$|x[n]| \leq B_x < \infty \qquad \text{for all } n$$

Stability requires that for any possible input sequence there exist a fixed positive value B_y such that

$$|y[n]| \leq B_y < \infty$$

Stable System

- Stability (in the sense of bounded-input bounded-output BIBO)
  - A system is stable if and only if every bounded input produces a bounded output

$$|x[n]| \le B_x < \infty \Rightarrow |y[n]| \le B_y < \infty$$

- Example
  - Square

$$y[n] = (x[n])^2$$

$$\text{if input is bounded by } |x[n]| \le B_x < \infty$$

$$\text{output is bounded by } |y[n]| \le B_x^2 < \infty$$

- Counter Example
  - Log

$$y[n] = \log_{10}(|x[n]|)$$

$$\text{even if input is bounded by } |x[n]| \le B_x < \infty$$

$$\text{output not bounded for } x[n] = 0 \Rightarrow y[0] = \log_{1}(|x[n]|) = -\infty$$

Memory (State)

A system is referred to as *memoryless* if the output $y[n]$ at every value of n depends only on the input $x[n]$ at the same value of n.

If the system has no memory, it is called a *static* system. Otherwise it is a *dynamic* system.

Impulse Response of LTI Systems

Find the impulse response by computing the response to $\delta[n]$.

Systems whose impulse responses have only a finite number of nonzero samples are called *finite-duration impulse response* (FIR) systems.

Systems whose impulse responses are of infinite duration are called *infinite-duration impulse response* (IIR) systems.

If $h[n] = 0$ for $n < 0$, the system is *causal*.

Impulse Response for Examples

Find the impulse response by computing the response to $\delta[n]$

Ideal Delay System $y[n] = \delta[n - n_d]$ FIR

Moving Average System $y[n] = \begin{cases} \dfrac{1}{M_2 + M_1 + 1}, & -M_1 \le n \le M_2 \\ 0, & \text{otherwise} \end{cases}$ FIR

Accumulator System $y[n] = \sum_{k=-\infty}^{n} \delta[k] = \begin{cases} 1, & n \ge 0 \\ 0, & n < 0 \end{cases}$ IIR

$y[n] = u[n]$

Backward Difference System $y[n] = \delta[n] - \delta[n-1]$ FIR

Stability Condition for LTI Systems

An LTI system is BIBO stable if and only if its impulse response is absolutely summable, that is

$$S = \sum_{k=-\infty}^{\infty} |h[k]| < \infty$$

Stable and Causal LTI Systems

- An LTI system is (BIBO) stable if and only if
  - Impulse response is absolute summable

$$\sum_{k=-\infty}^{\infty} |h[k]| < \infty$$

  - Let's write the output of the system as

$$|y[n]| = \left|\sum_{k=-\infty}^{\infty} h[k]x[n-k]\right| \le \sum_{k=-\infty}^{\infty} |h[k]||x[n-k]|$$

  - If the input is bounded

$$|x[n]| \le B_x$$

  - Then the output is bounded by

$$|y[n]| \le B_x \sum_{k=-\infty}^{\infty} |h[k]|$$

  - The output is bounded if the absolute sum is finite
- An LTI system is causal if and only if

$$h[k] = 0 \quad \text{for} \quad k < 0$$

Difference Equations

An important subclass of LTI systems are defined by an Nth-order linear constant-coefficient difference equation:

$$\sum_{0}^{N} a_k y[n-k] = \sum_{0}^{M} b_m x[n-m]$$

Often the leading coefficient $a_0 = 1$. Then the output $y[n]$ can be computed recursively from

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{m=0}^{M} b_m x[n-m]$$

A causal LTI system of this form can be simulated in MATLAB using the function `filter`

```
y = filter(a,b,x);
```

Homogeneous Solution

Given the homogeneous equation: $\sum_{k=0}^{N} a_k y_h[n-k] = 0$

Assume that the homogeneous solution is of the form

$$y_h[n] = \lambda^n$$

then

$$y_h[n] = \sum_{k=0}^{N} a_k \lambda^{n-k} = \lambda^{n-N}(a_0\lambda^N + a_1\lambda^{N-1} + \cdot \quad + a_N) = 0$$

defines an Nth order characteristic polynomial with roots $\lambda_1, \lambda_2 \ldots \lambda_N$

The general solution is then a sequence $y_h[n]$

$$y_h[n] = \sum_{1}^{N} A_m \lambda_m^n$$

(if the roots are all distinct)

The coefficients A_m may be found from the initial conditions.

Particular Solution

The particular solution is required to satisfy the difference equation for a specific input signal $x[n]$, $n \geq 0$.

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

To find the particular solution we assume for the solution $y_p[n]$ a form that depends on the form of the specific input signal $x[n]$.

$$y[n] = y_h[n] + y_p[n]$$

Fourier representation of signals

- A *discrete-time sinusoidal signal* $x[n]$ is obtained by sampling a continuous-time sinusoid $x(t) = \cos(2\pi F_0 t + \theta)$ at equally spaced points $t = nT$, which results in

$$x[n] = A\cos(2\pi F_0 nT + \theta)\Big|_{T=1/F_s} = A\cos\left(2\pi\frac{F_0}{F_s}n + \theta\right)$$

where F_0 (Hz) is the fundamental frequency of $x(t)$ and F_s is the *sampling frequency*.

- The *normalized frequency* variable is defined as

$$f \triangleq \frac{F}{F_s} = FT$$

where T is the *sampling period*.

- Similarly, the *normalized angular frequency* variable is defined as

$$\omega \triangleq 2\pi f = 2\pi\frac{F}{F_s} = \Omega T$$

- In this case, the discrete sinusoidal signal can be expressed as

$$x[n] = A\cos(2\pi f_0 n + \theta) = A\cos(\omega_0 n + \theta)$$

Fourier representation of signals

- **Periodicity in time:** By definition $x[n]$ is *periodic* if $x[n+N] = x[n], \forall n.$

  $x[n+N] = A\cos(2\pi f_0 n + 2\pi f_0 N + \theta) = A\cos(2\pi f_0 n + \theta) = x[n]$

  which is possible if and only if $2\pi f_0 N = 2\pi k$, with $k \in \mathbb{Z}$.

Result

The sequence $x[n] = A\cos(2\pi f_0 n + \theta)$ is periodic iff $f_0 = k/N$, that is, f_0 is a rational number. If k and N are a pair of *mutually prime* integers, then N is a fundamental period of $x[n]$.

- **Periodicity in frequency:** We can see that

$$A\cos[(\omega_0 + k\,2\pi)n + \theta] = A\cos(\omega_0 n + \underbrace{kn}_{\in\mathbb{Z}}\,2\pi + \theta) = A\cos(\omega_0 n + \theta)$$

Result

The sequence $x[n] = A\cos(\omega_0 n + \theta)$ is periodic in ω_0 with fundamental period 2π and periodic in f_0 with fundamental period one.

Fourier representation of signals

- All distinct sinusoidal sequences have frequencies within an interval of 2π radians. We shall use the *fundamental frequency ranges*

$$-\pi \leq \omega < \pi \quad \text{or} \quad 0 \leq \omega < 2\pi$$

Therefore, if $0 \leq \omega_0 < 2\pi$, the frequencies ω_0 and $\omega_0 + m\,2\pi$ are indistinguishable in terms of their values.

- Since $A\cos(\omega_0[n+n_0]+\theta] = A\cos(\omega_0 n + (\omega_0 n_0 + \theta))$, a time shift is equivalent to a phase change.
- The rate of oscillation of a discrete-time sinusoid increases as ω_0 goes from $\omega_0 = 0$ to $\omega_0 = \pi$. Yet, as ω_0 increases from $\omega_0 = \pi$ to $\omega_0 = 2\pi$, the oscillations become slower. Therefore:

$$\begin{aligned} \text{Vicinity of } \omega_0 = k\,2\pi &\implies \text{Low frequencies} \\ \text{Vicinity of } \omega_0 = \pi + k\,2\pi &\implies \text{High frequencies} \end{aligned}$$

Discrete complex exponentials

- Similar properties hold for the *discrete-time complex exponentials*

$$s_k = A_k e^{j\omega_k n}$$

- For $s_k[n]$ to be periodic with fundamental period N, the frequency ω_k should be a rational multiple of 2π, that is $\omega_k = 2\pi k/N$.

All distinct complex exponentials with period N and frequency in the fundamental range, have frequencies equal to $\{\omega_k = 2\pi k/N\}_{k=0}^{N-1}$.

- The discrete complex exponentials are N-periodic in both the n- and k-variables.

$$\begin{aligned} s_k[n+N] &= s_k[n] \quad \text{(periodic in time)} \\ s_{k+N}[n] &= s_k[n] \quad \text{(periodic in frequency)} \end{aligned}$$

- The complex exponentials are also *orthogonal*, viz.

$$\langle s_k, s_m \rangle \triangleq \sum_{n=0}^{N-1} s_k[n]\, s_m^*[n] = \begin{cases} N, & k = m \\ 0, & k \neq m \end{cases}$$

Discrete Fourier Series

- Given a periodic sequence $\tilde{x}[n]$ with period N so that

$$\tilde{x}[n] = \tilde{x}[n + rN]$$

- The Fourier series representation can be written as

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

- The Fourier series representation of continuous-time periodic signals require infinite many complex exponentials
- Not that for discrete-time periodic signals we have

$$e^{j(2\pi/N)(k+mN)n} = e^{j(2\pi/N)kn} e^{j(2\pi mn)} = e^{j(2\pi/N)kn}$$

- Due to the periodicity of the complex exponential we only need N exponentials for discrete time Fourier series

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

Discrete Fourier Series Pair

- A periodic sequence in terms of Fourier series coefficients

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

- The Fourier series coefficients can be obtained via

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

- For convenience we sometimes use

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

- Analysis equation

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

- Synthesis equation

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

Fourier series for discrete-time periodic signals

- Consider a linear combination of N complex exponentials

$$x[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn} = \sum_{k=0}^{N-1} c_k s_k[n]$$

which is periodic with fundamental period N.

- To determine the series expansion coefficients c_k, we exploit the orthogonality of $s_k[n]$ as follows

$$\langle x, s_m \rangle = \sum_{n=0}^{N-1} x[n]\, s_m^*[n] = \sum_{k=0}^{N-1} c_k \langle s_k, s_m \rangle = N\, c_m, \quad m = 0, \ldots, N-1$$

- Therefore, we have

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

which is periodic in k with the fundamental period equal to N.

Discrete-time Fourier series (DTFS)

DTFS

The *Discrete Time Fourier Series (DTFS)* pair is defined as

$$x[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn} \Longleftrightarrow c_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

- **Parseval's relation:** The average power in one period of $x[n]$ can be expressed in terms of the Fourier series coefficients as

$$P_{\text{av}} = \frac{1}{N}\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{k=0}^{N-1} |c_k|^2$$

- The value of $|c_k|^2$ provides the portion of the average power P_{av} of $x[n]$ that is contributed by its k-th harmonic component. Since $c_{k+N} = c_k$, there are only N distinct harmonic components.
- The graph of $|c_k|^2$ (as a function of either $f = k/N$, $\omega = 2\pi k/N$, or simply k) is known as the *power spectrum* of $x[n]$.

Fourier representation of aperiodic signals

- Consider a finite duration sequence $x[n]$, such that $x[n] = 0$ outside the range $-L_1 \leq n \leq L_2$. Define a *periodized version* $x_p[n]$ of $x[n]$ as

$$x_p[n] = \sum_{l \in \mathbb{Z}} x[n - l\,N], \quad \text{with } N > L_1 + L_2 + 1$$

- The DTFS of $x_p[n]$ is given by

$$x_p[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn}$$

where

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p[n]e^{-j\frac{2\pi}{N}kn} = \frac{1}{N}\sum_{n=-\infty}^{\infty} x[n]e^{-j\frac{2\pi}{N}kn}$$

- Define the "envelope" function $X(e^{j\omega})$ as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

Discrete-time Fourier transform (DTFT)

- Noticing that $1/N = \Delta\omega/2\pi$, we have

$$x_p[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn} = \frac{1}{2\pi}\sum_{k=0}^{N-1} X(e^{jk\Delta\omega})e^{j(k\Delta\omega)n}\Delta\omega$$

- As $N \to \infty$, $x_p[n] = x[n]$, $\forall n$. Also, as $N \to \infty$, $\Delta\omega \to 0$, and the summation above passes to an integral over the frequency range from 0 to 2π. As a result, we have

DTFT

The *Discrete Time Fourier Transform (DTFT)* pair is defined as

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \Longleftrightarrow X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

- The quantities $X(e^{j\omega})$, $|X(e^{j\omega})|$, and $\angle X(e^{j\omega})$ are known as the *spectrum*, *magnitude spectrum*, and *phase spectrum* of $x[n]$.
- **Parseval's relation:**

$$\sum |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega$$

Discrete Fourier Transform

- Periodic sequence and DFS coefficients

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

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

- Since summations are calculated btw 0 and (N-1)

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n] W_N^{kn}, & 0 \le k \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$x[n] = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Generally

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

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

Discrete Fourier Transform

- Given N samples $x[n]$, $0 \leq n \leq N-1$ of an N-length sequence, its *Discrete Fourier Transform* (DFT) $X[k]$ is defined by

$$X[k] \triangleq \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad 0 \leq k \leq N-1$$

- Given N DFT coefficients $X[k]$, $0 \leq k \leq N-1$, their related N "time-domain" samples $x[n]$, $0 \leq n \leq N-1$ can be recovered by the *inverse DFT* (IDFT) given by

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

- Note that $X[k]$ is a function of the discrete frequency index k, which corresponds to $\omega_k = 2\pi/N$, $k = 0, 1, \ldots, N-1$.

In summary: The DFT pair

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad \overset{\text{DFT}}{\longleftrightarrow} \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad W_N \triangleq e^{-j\frac{2\pi}{N}}$$

Discrete Fourier Transform

- The correctness of the DFT formulas can be validated through:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{m=0}^{N-1} X[m] W_N^{-mn} \right] W_N^{nk} =$$
$$= \sum_{m=0}^{N-1} X[m] \left[\frac{1}{N} \sum_{n=0}^{N-1} W_N^{(k-m)n} \right]$$

- The orthogonality of discrete complex exponentials suggests

$$\frac{1}{N} \sum_{n=0}^{N-1} W_N^{(k-m)n} = \frac{1}{N} \langle W_N^k, W_N^m \rangle = \begin{cases} 1, & k - m = rN \\ 0, & \text{otherwise} \end{cases}$$

which concludes the proof.

- Note that the N complex numbers $\{W_N^{-k}\}_{k=0}^{N-1}$ satisfy

$$\left(W_N^{-k}\right)^N = e^{j2\pi k} = 1$$

and therefore they form the *roots of unity* (i.e., the N solutions of $z^N - 1 = 0$). Note that these roots are equally spaced around the unit circle with the angular spacing of $2\pi/N$ radians.

Discrete Fourier Transform

- The N equations for the DFT coefficients can be expressed in matrix form as

$$\underbrace{\begin{bmatrix} X[0] \\ X[1] \\ \cdots \\ X[N-1] \end{bmatrix}}_{X_N} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \cdots & \cdots & \ddots & \cdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}}_{\mathbf{W}_N} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \cdots \\ x[N-1] \end{bmatrix}}_{x_N}$$

- Thus, we have

$$X_N = \mathbf{W}_N x_N$$

- Note that $\mathbf{W}_N$ is symmetric ($\mathbf{W}_N = \mathbf{W}_N^T$) and *orthogonal*, *viz.*

$$\mathbf{W}_N^H \mathbf{W}_N = N \mathbf{I}_N \implies \mathbf{W}_N^{-1} = \frac{1}{N}\mathbf{W}_N^H = \frac{1}{N}\mathbf{W}_N^*$$

- Therefore, x_N can be recovered (synthesized) from X_N according to

$$x_N = \mathbf{W}_N^{-1} X_N = \frac{1}{N}\mathbf{W}_N^* X_N$$

which is nothing else but a matrix representation of IDFT.

Discrete Fourier Transform

- The *twiddle factor* $W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$ is periodic in both k and n with fundamental period N, namely

$$W_N^{(k+N)n} = W_N^{kn} \quad \text{and} \quad W_N^{k(N+n)} = W_N^{kn}$$

- Letting $k \in \mathbb{Z}$ results in the *Discrete Fourier Series* (DFS):

$$\tilde{X}[k] = \tilde{X}[k+N] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad \forall k \in \mathbb{Z}$$

- If n is allowed to take upon any integer value, the values of $x[n]$ will repeat with fundamental period N, resulting in the *Inverse Discrete Fourier Series* (IDFS).

$$\tilde{x}[n] = \tilde{x}[n+N], \quad \forall n \in \mathbb{Z}$$

- These periodicities are an inherent property of DFT, which stem from the discrete nature of time and frequency variables.

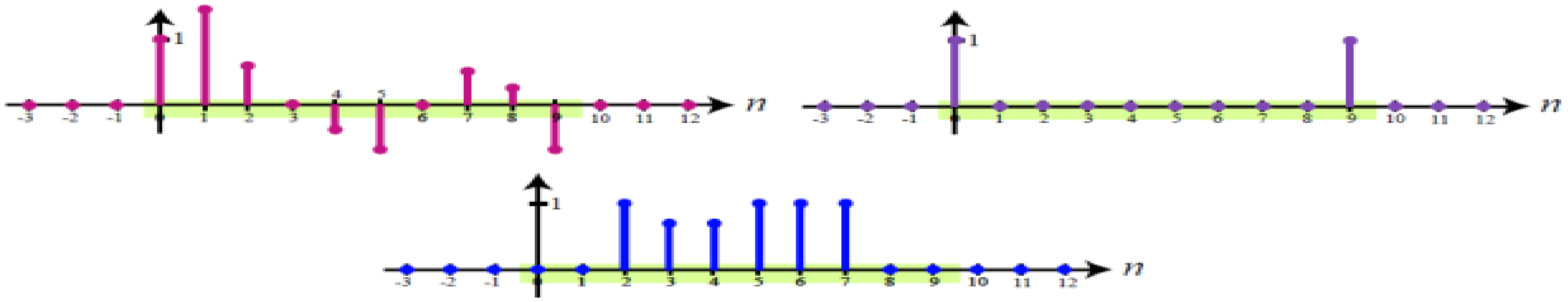
Summary of properties

	Property	N-point sequence	N-point DFT
		$x[n], h[n], v[n]$	$X[k], H[k], V[k]$
		$x_1[n], x_2[n]$	$X_1[k], X_2[k]$
1.	Linearity	$a_1x_1[n] + a_2x_2[n]$	$a_1X_1[k] + a_2X_2[k]$
2.	Time shifting	$x[(n-m)_N]$	$W_N^{km}X[k]$
3.	Frequency shifting	$W_N^{-mn}x[n]$	$X[(k-m)_N]$
4.	Modulation	$x[n]\cos(2\pi/N)k_0n$	$\frac{1}{2}X[(k+k_0)_N] + \frac{1}{2}X[(k-k_0)_N]$
5.	Folding	$x[(-n)_N]$	$X^*[k]$
6.	Conjugation	$x^*[n]$	$X^*[(-k)_N]$
7.	Duality	$X[n]$	$Nx[(-k)]_N]$
8.	Convolution	$h[n] \, (N) \, x[n]$	$H[k]X[k]$
9.	Correlation	$x[n] \, (N) \, y[(-n)_N]$	$X[k]Y^*[k]$
10.	Windowing	$v[n]x[n]$	$\frac{1}{N}V[k] \, (N) \, X[k]$
11.	Parseval's theorem	$\sum_{n=0}^{N-1} x[n]y^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]Y^*[n]$	
12.	Parseval's relation	$\sum_{n=0}^{N-1} \lvert x[n]\rvert^2 = \frac{1}{N}\sum_{k=0}^{N-1} \lvert X[k]\rvert^2$	

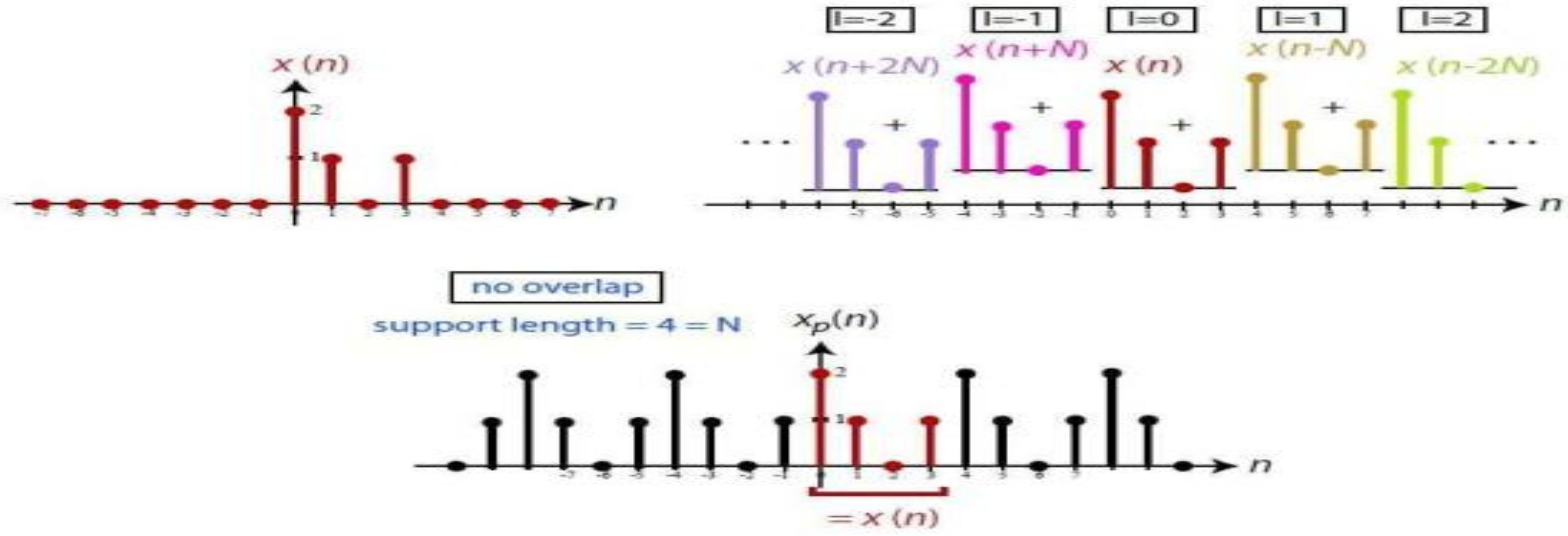
Circular convolution

Assume: $x_1(n)$ and $x_2(n)$ have support $n = 0, 1, \dots, N-1$.

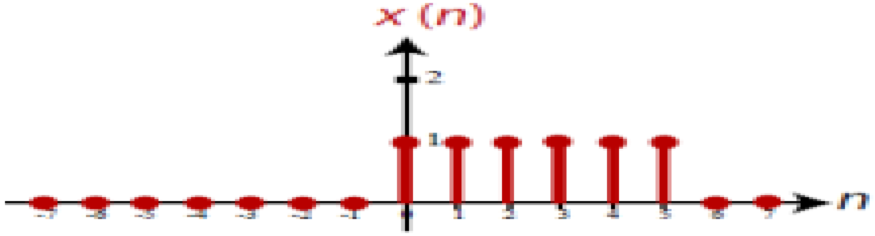
Examples: $N = 10$ and support: $n = 0, 1, \dots, 9$

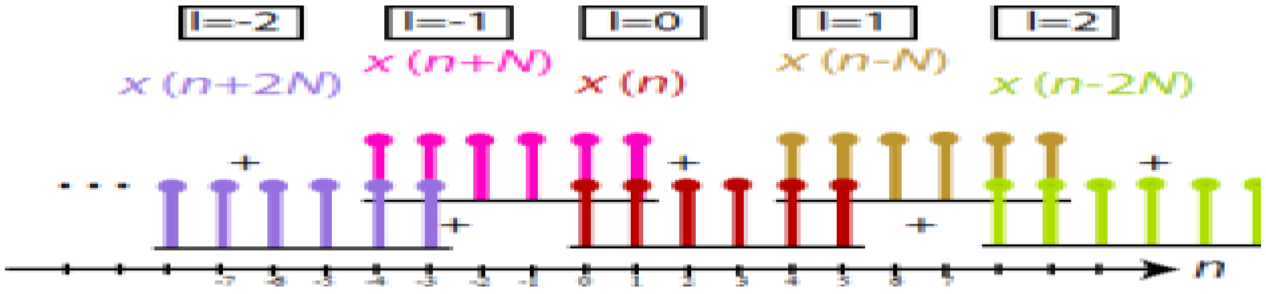


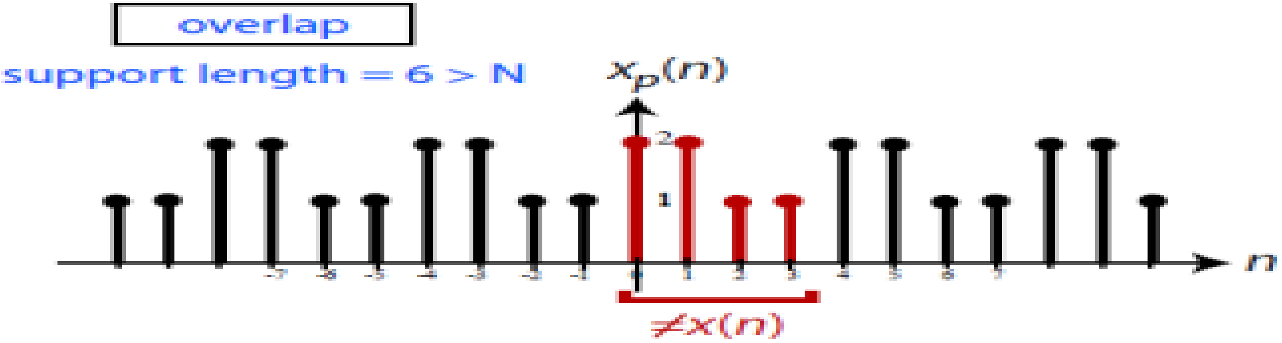
Periodic repetition: N=4



Periodic repetition: N=4







Circular convolution

Assume: $x_1(n)$ and $x_2(n)$ have support $n = 0, 1, \ldots, N-1$.

To compute $\sum_{k=0}^{N-1} x_1(k)x_2((n-k))_N$ (or $\sum_{k=0}^{N-1} x_2(k)x_1((n-k))_N$):

1. Take the periodic repetition of $x_2(n)$ with period N:

$$x_{2p}(n) = \sum_{l=-\infty}^{\infty} x_2(n - lN)$$

2. Conduct a standard linear convolution of $x_1(n)$ and $x_{2p}(n)$ for $n = 0, 1, \ldots, N-1$:

$$x_1(n) \otimes x_2(n) = x_1(n) * x_{2p}(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_{2p}(n-k) = \sum_{k=0}^{N-1} x_1(k)x_{2p}(n-k)$$

Note: $x_1(n) \otimes x_2(n) = 0$ for $n < 0$ and $n \geq N$.

Circular convolution

$$\sum_{k=0}^{N-1} x_1(k)\boxed{x_2((n-k))_N} = \sum_{k=0}^{N-1} x_1(k)\boxed{x_{2p}(n-k)}$$

... which makes sense, since $x((n))_N = x_p(n)$.

Circular convolution-another interpretation

Assume: $x_1(n)$ and $x_2(n)$ have support $n = 0, 1, \ldots, N-1$.

To compute $\sum_{k=0}^{N-1} x_1(k)x_2((n-k))_N$ (or $\sum_{k=0}^{N-1} x_2(k)x_1((n-k))_N$):

1. Conduct a linear convolution of $x_1(n)$ and $x_2(n)$ for all n:

$$x_L(n) = x_1(n) * x_2(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k) = \sum_{k=0}^{N-1} x_1(k)x_2(n-k)$$

2. Compute the periodic repetition of $x_L(n)$ and window the result for $n = 0, 1, \ldots, N-1$:

$$x_1(n) \otimes x_2(n) = \sum_{l=-\infty}^{\infty} x_L(n - lN), \qquad n = 0, 1, \ldots, N-1$$

Using DFT for Linear Convolution

Therefore, circular convolution and linear convolution are related as follows:

$$x_C(n) = x_1(n) \otimes x_2(n) = \sum_{l=-\infty}^{\infty} x_L(n - lN)$$

for $n = 0, 1, \ldots, N-1$

Q: When can one recover $x_L(n)$ from $x_C(n)$?
When can one use the DFT to compute linear convolution?

A: When there is no overlap in the periodic repetition of $x_L(n)$.
When support length of $x_L(n) \le N$.

Using DFT for Linear Convolution

- The *linear convolution* of two finite-length sequences $\{x[n]\}_{n=0}^{L-1}$ and $\{h[n]\}_{n=0}^{M-1}$ is a sequence $y[n]$ of length $L+M-1$, given by

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k], \quad n = 0, 1, \ldots, L+M-2$$

- The convolution sequence $y[n]$ has DTFT given by

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

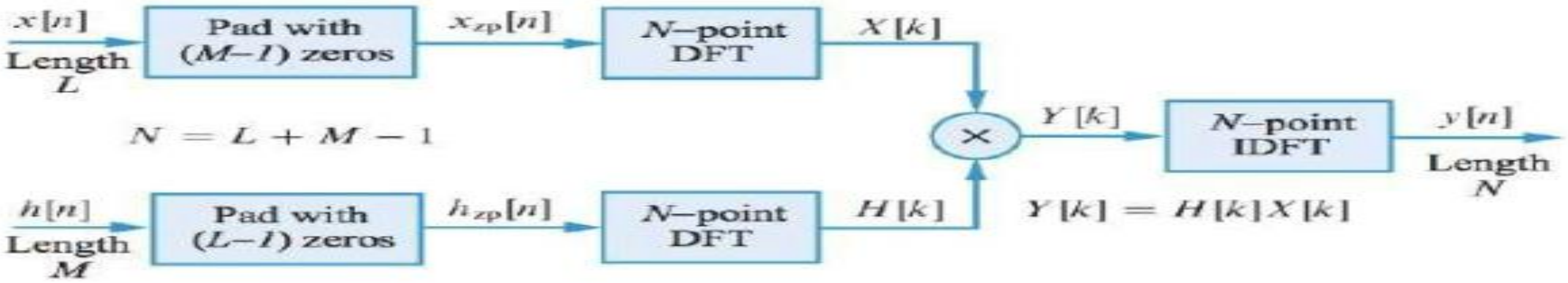
- If we sample $Y(e^{j\omega})$ at $\omega_k = 2\pi k/N$, where $N \geq L+M-1$, we can uniquely recover $y[n]$ from $Y[k] = Y(e^{j2\pi k/N})$.
- On the other hand, the IDFTs of $H(e^{j\frac{2\pi k}{N}})$ and $X(j\frac{2\pi k}{N})$ yield the sequences $h[n]$ and $x[n]$ padded with $(N-M)$ and $(N-L)$ zeros, respectively. As a result,

$$y_{zp}[n] = x_{zp}[n] \; Ⓝ \; h_{zp}[n] \quad \Longleftrightarrow \quad Y[k] = X[k]H[k]$$

- Note that if $N \geq L+M-1$, $y[n] = y_{zp}[n]$, $0 \leq n \leq L+M-2$, that is, circular convolution is identical to linear convolution.

Using DFT for Linear Convolution

- Thus, linear convolution can be implemented by means of the DFT as shown below.



- The length M of the impulse response at which the DFT based approach is more efficient than direct computation of convolution depends on the hardware and software available to implement the computations.

Using DFT for cicular Convolution

$N = L + M - 1.$

Let $x_m(n)$ have support $n = 0, 1, \ldots, N - 1.$

Let $h(n)$ have support $n = 0, 1, \ldots, M - 1.$

We zero pad $h(n)$ to have support $n = 0, 1, \ldots, N - 1.$

1. Take N-DFT of $x_m(n)$ to give $X_m(k)$, $k = 0, 1, \ldots, N - 1.$
2. Take N-DFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N - 1.$
3. Multiply: $Y_m(k) = X_m(k) \cdot H(k)$, $k = 0, 1, \ldots, N - 1.$
4. Take N-IDFT of $Y_m(k)$ to give $y_{C,m}(n)$, $n = 0, 1, \ldots, N - 1.$

Using DFT for cicular Convolution

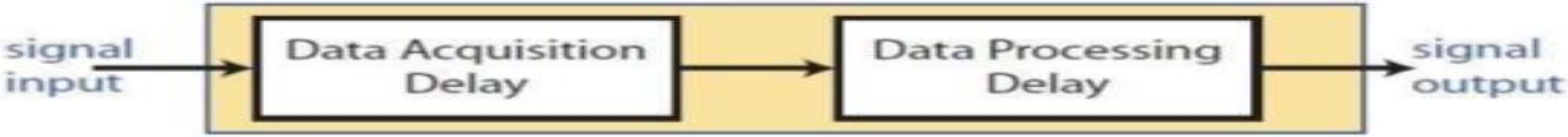
- Let's compute $y[n]$ for the case of $N = 4$. We have

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix} = \underbrace{\begin{bmatrix} x[0] & x[3] & x[2] & x[1] \\ x[1] & x[0] & x[3] & x[2] \\ x[2] & x[1] & x[0] & x[3] \\ x[3] & x[2] & x[1] & x[0] \end{bmatrix}}_{\mathbf{X}_N} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \end{bmatrix}$$

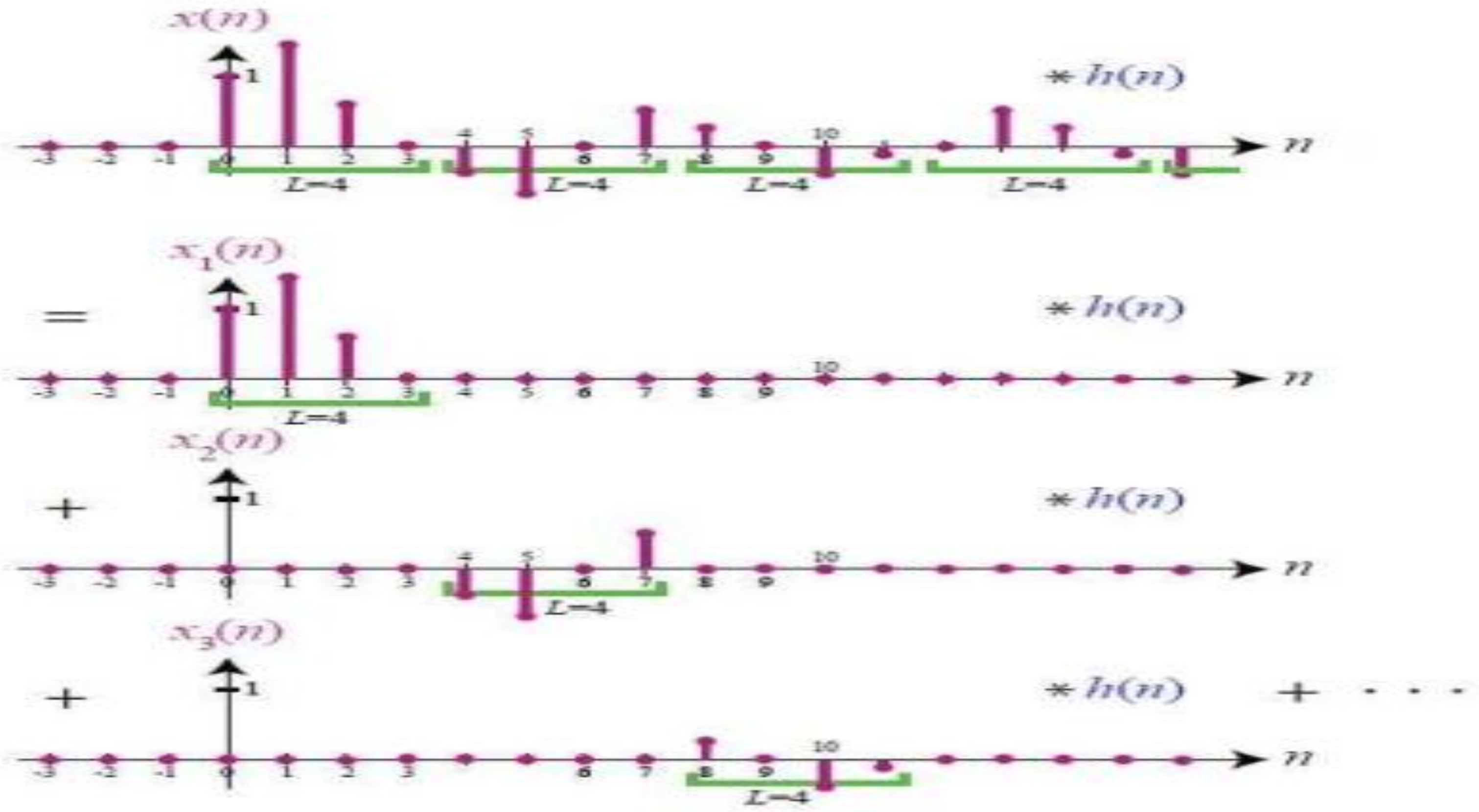
- We note that the column of $\mathbf{X}_N$ are generated by circularly shifting $x[n]$. A matrix generated by this process is called a *circulant matrix*.

Filtering of Long Data Sequences

- All N-input samples are required simultaneously by the DFT operator.
- If N is too large as for long data sequences, then there is a significant delay in processing that precludes real-time processing.



Over-lap Add



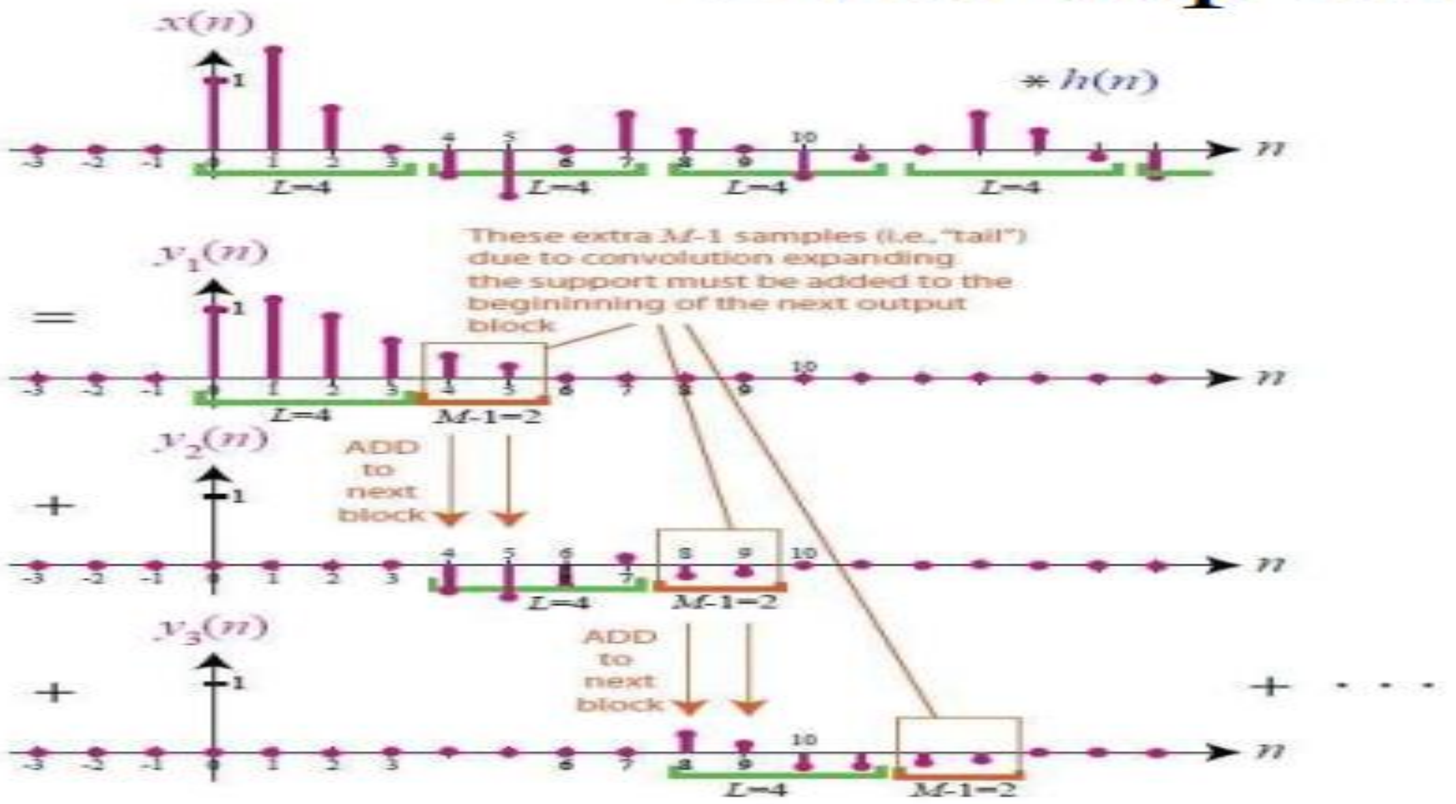
Input $x(n)$ is divided into non-overlapping blocks $x_m(n)$ each of length L.

Each input block $x_m(n)$ is individually filtered as it is received to produce the output block $y_m(n)$.

Over-lap Add

- makes use of the N-DFT and N-IDFT where: $N = L + M - 1$
  - Thus, zero-padding of $x(n)$ and $h(n)$ that are of length $L, M < N$ is required.
  - The actual implementation of the DFT/IDFT will use the fast Fourier Transform (FFT) for computational simplicity.

Over-lap Add



Output blocks $y_m(n)$ must be fitted together appropriately to generate:

$$y(n) = x(n) * h(n)$$

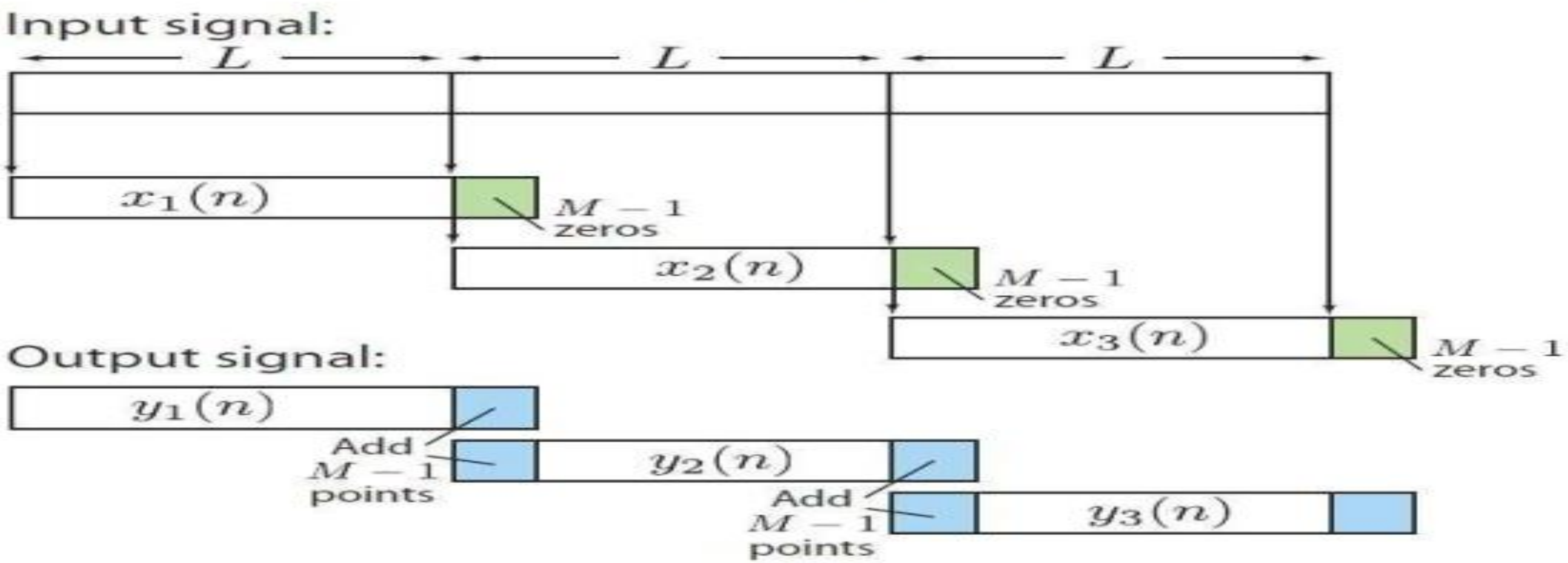
The support overlap amongst the $y_m(n)$ blocks must be accounted for.

Over-lap Add

From the Addititvity property, since:

$$
\begin{aligned}
x(n) &= x_1(n) + x_2(n) + x_3(n) + \cdots = \sum_{m=1}^{\infty} x_m(n) \\
x(n) * h(n) &= (x_1(n) + x_2(n) + x_3(n) + \cdots) * h(n) \\
&= x_1(n) * h(n) + x_2(n) * h(n) + x_3(n) * h(n) + \cdots \\
&= \sum_{m=1}^{\infty} x_m(n) * h(n) = \sum_{m=1}^{\infty} y_m(n)
\end{aligned}
$$

Over-lap Add


Input signal:
L
L
L
$x_1(n)$
$M-1$ zeros
$x_2(n)$
$M-1$ zeros
$x_3(n)$
$M-1$ zeros
Output signal:
$y_1(n)$
Add $M-1$ points
$y_2(n)$
Add $M-1$ points
$y_3(n)$

Over-lap Add

1. Break the input signal $x(n)$ into non-overlapping blocks $x_m(n)$ of length L.
2. Zero pad $h(n)$ to be of length $N = L + M - 1$.
3. Take N-DFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N - 1$.
4. For each block m:
    4.1 Zero pad $x_m(n)$ to be of length $N = L + M - 1$.
    4.2 Take N-DFT of $x_m(n)$ to give $X_m(k)$, $k = 0, 1, \ldots, N - 1$.
    4.3 Multiply: $Y_m(k) = X_m(k) \cdot H(k)$, $k = 0, 1, \ldots, N - 1$.
    4.4 Take N-IDFT of $Y_m(k)$ to give $y_m(n)$, $n = 0, 1, \ldots, N - 1$.
5. Form $y(n)$ by overlapping the last $M - 1$ samples of $y_m(n)$ with the first $M - 1$ samples of $y_{m+1}(n)$ and adding the result.

Over-lap save

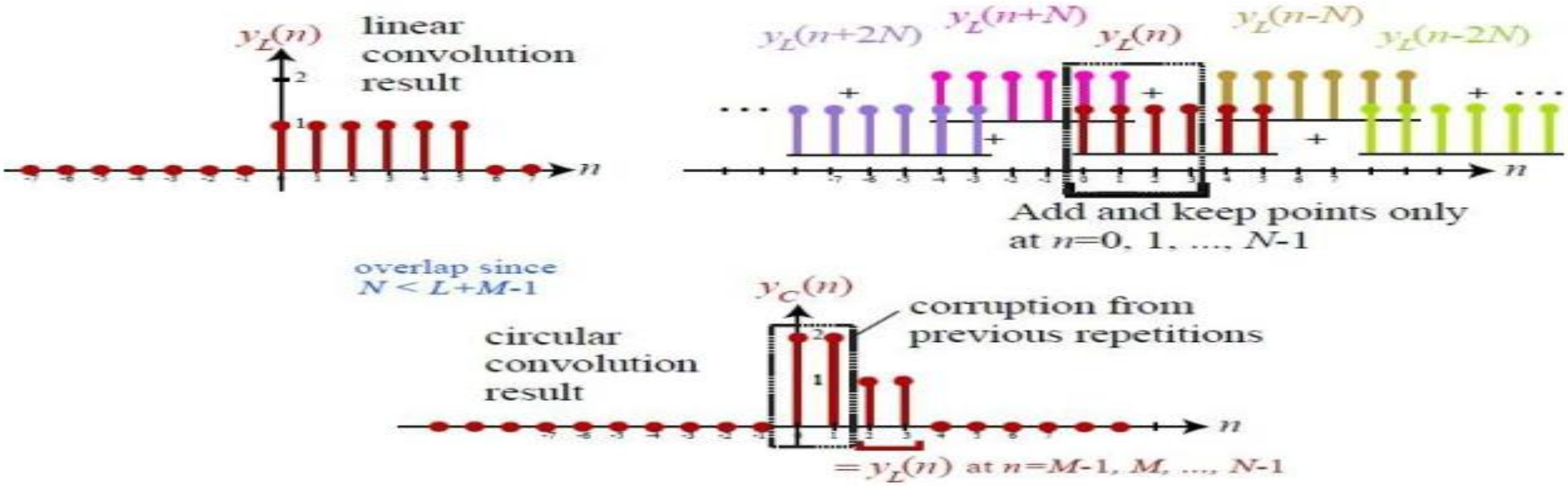
Deals with the following signal processing principles:

- The $N = (L + M - 1)$-circular convolution of a discrete-time signal of length N and a discrete-time signal of length M using an N-DFT and N-IDFT.

- Time-Domain Aliasing:

$$x_C(n) = \sum_{l=-\infty}^{\infty} \underbrace{x_L(n - lN)}_{\text{support}=M+N-1}, \qquad n = 0, 1, \ldots, N-1$$

Over-lap save



Over-lap save

- Convolution of $x_m(n)$ with support $n = 0, 1, \ldots, N-1$ and $h(n)$ with support $n = 0, 1, \ldots, M-1$ via the N-DFT will produce a result $y_{C,m}(n)$ such that:

$$y_{C,m}(n) = \begin{cases} \text{aliasing corruption} & n = 0, 1, \ldots, M-2 \\ y_{L,m}(n) & n = M-1, M, \ldots, N-1 \end{cases}$$

where $y_{L,m}(n) = x_m(n) * h(n)$ is the desired output.

- The first $M-1$ points of a the current filtered output block $y_m(n)$ must be discarded.
- The previous filtered block $y_{m-1}(n)$ must compensate by providing these output samples.

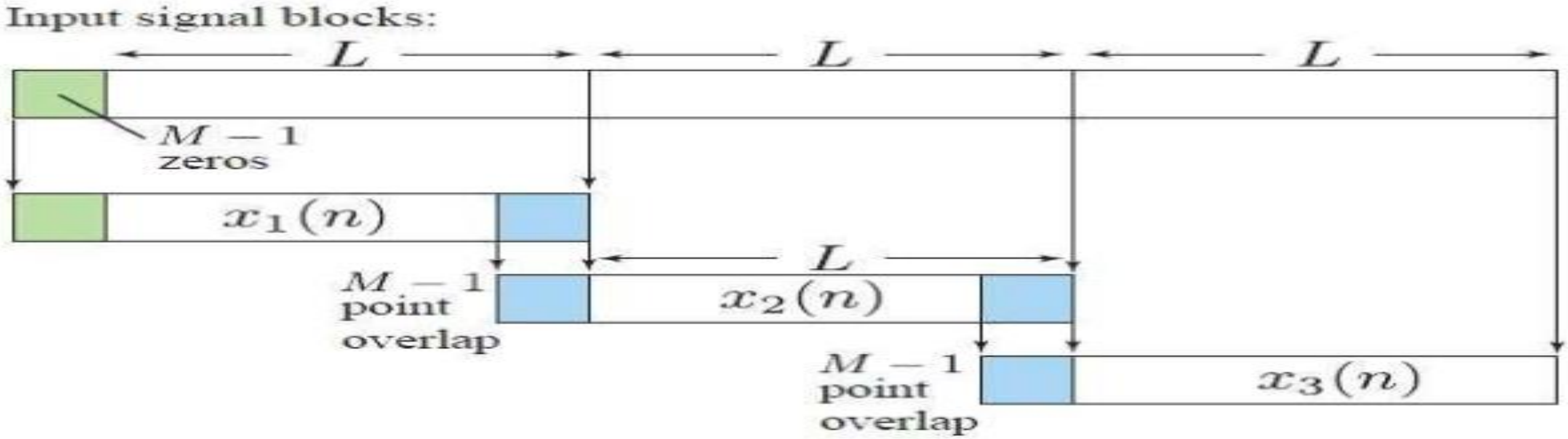
Over-lap save input segment stage

1. All input blocks $x_m(n)$ are of length $N = (L + M - 1)$ and contain sequential samples from $x(n)$.

2. Input block $x_m(n)$ for $m > 1$ overlaps containing the first $M - 1$ points of the previous block $x_{m-1}(n)$ to deal with aliasing corruption.

3. For $m = 1$, there is no previous block, so the first $M - 1$ points are zeros.

Over-lap save input segment stage



Over-lap save input segment stage

$$x_1(n) = \{\underbrace{0, \quad 0, \quad \ldots \quad 0,}_{M-1 \text{ zeros}} \; x(0), x(1), \ldots, x(L-1)\}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots x(L-1)}_{\text{last } M-1 \text{ points from } x_1(n)}, x(L), \ldots, x(2L-1)\}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots x(2L-1)}_{\text{last } M-1 \text{ points from } x_2(n)}, x(2L), \ldots, x(3L-1)\}$$

$$\vdots$$

The last $M-1$ points from the previous input block must be saved for use in the current input block.

Over-lap save filtering stage

- makes use of the N-DFT and N-IDFT where: $N = L + M - 1$
  - Only a one-time zero-padding of $h(n)$ of length $M \ll L < N$ is required to give it support $n = 0, 1, \ldots, N - 1$.
  - The input blocks $x_m(n)$ are of length N to start, so no zero-padding is necessary.
  - The actual implementation of the DFT/IDFT will use the fast Fourier Transform (FFT) for computational simplicity.

Over-lap save output blocks

$$y_{C,m}(n) = \begin{cases} \text{aliasing} & n = 0, 1, \ldots, M-2 \\ y_{L,m}(n) & n = M-1, M, \ldots, N-1 \end{cases}$$

where $y_{L,m}(n) = x_m(n) * h(n)$ is the desired output.

Over-lap save output blocks

$$y_1(n) = \{\underbrace{y_1(0), \quad y_1(1), \quad \ldots \quad y_1(M-2)}_{M-1 \text{ points corrupted from aliasing}}, \; y(0), \ldots, y(L-1)\}$$

$$y_2(n) = \{\underbrace{y_2(0), \quad y_2(1), \quad \ldots \quad y_2(M-2)}_{M-1 \text{ points corrupted from aliasing}}, y(L), \ldots, y(2L-1)\}$$

$$y_3(n) = \{\underbrace{y_3(0), \quad y_3(1), \quad \ldots \quad y_3(M-2)}_{M-1 \text{ points corrupted from aliasing}}, y(2L), \ldots, y(3L-1)\}$$

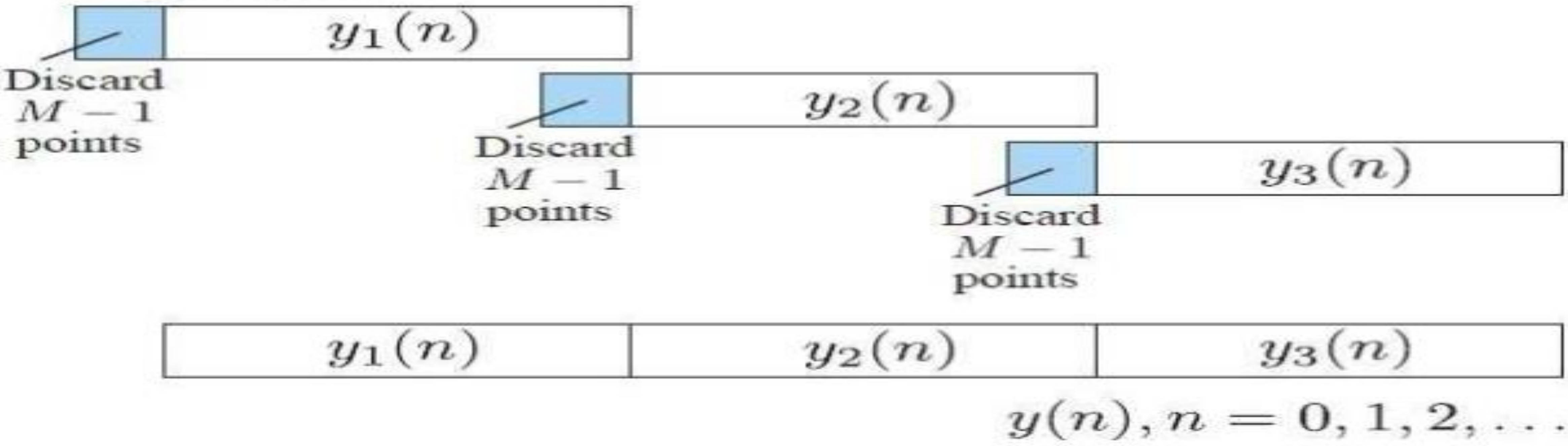
where $y(n) = x(n) * h(n)$ is the desired output.

The first $M-1$ points of each output block are discarded.

The remaining L points of each output block are appended to form $y(n)$.

Over-lap save output blocks

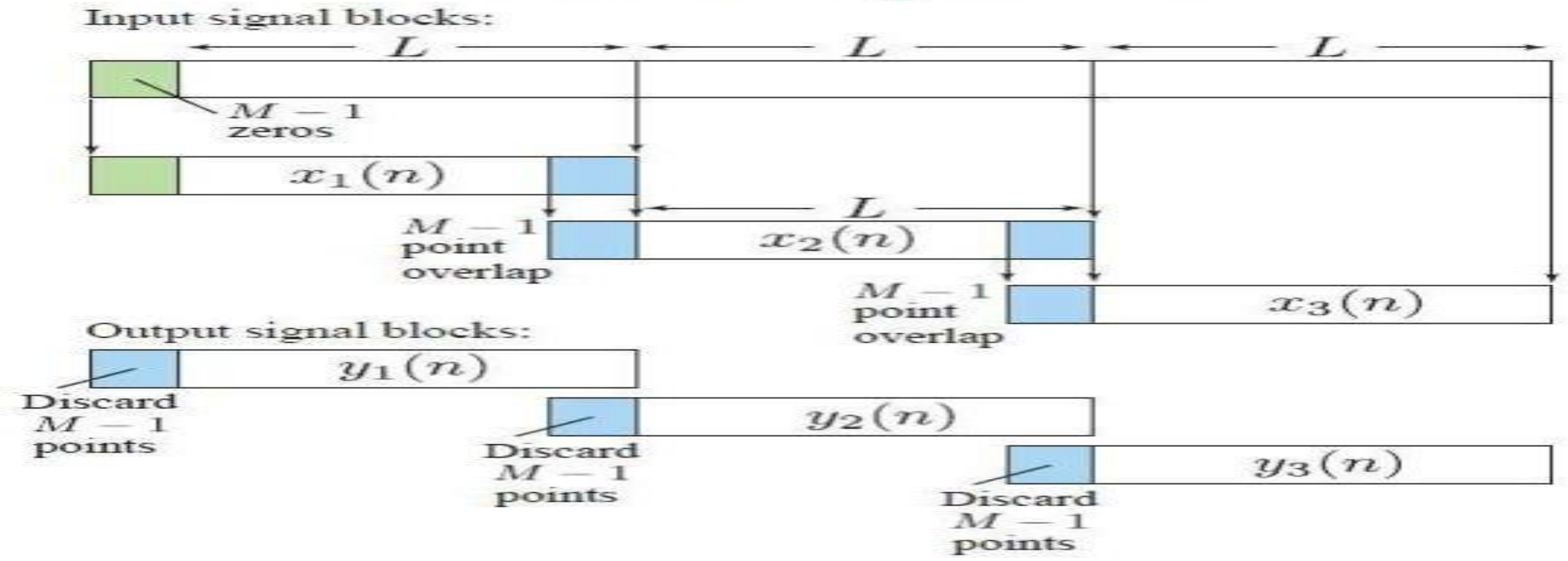
Output signal blocks:



Over-lap save

1. Insert $M-1$ zeros at the beginning of the input sequence $x(n)$.
2. Break the padded input signal into overlapping blocks $x_m(n)$ of length $N = L + M - 1$ where the overlap length is $M - 1$.
3. Zero pad $h(n)$ to be of length $N = L + M - 1$.
4. Take N-DFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N-1$.
5. For each block m:
   - 5.1 Take N-DFT of $x_m(n)$ to give $X_m(k)$, $k = 0, 1, \ldots, N-1$.
   - 5.2 Multiply: $Y_m(k) = X_m(k) \cdot H(k)$, $k = 0, 1, \ldots, N-1$.
   - 5.3 Take N-IDFT of $Y_m(k)$ to give $y_m(n)$, $n = 0, 1, \ldots, N-1$.
   - 5.4 Discard the first $M-1$ points of each output block $y_m(n)$.
6. Form $y(n)$ by appending the remaining (i.e., last) L samples of each block $y_m(n)$.

Over-lap save



Relationships between CTFT, DTFT, & DFT

- The N-point DFT provides a unique representation of the N-samples of a finite duration sequence.
- The DFT provides samples of the DTFT of the sequence at a set of equally spaced frequencies.
- Suppose that we are given a continuous-time signal $x_c(t)$ with Fourier transform $X_c(j\Omega)$.
- Its relate discrete signal $x[n] = x_c(nT)$ has the DTFT given by

$$X(e^{j\Omega T}) = \sum_n x_c(nT)e^{-j\Omega T n} = \frac{1}{T}\sum_m X_c\left(j\left(\Omega - \frac{2\pi}{T}m\right)\right)$$

- Since $\omega = \Omega T$, the N-point DFT $X[k]$ is obtained by sampling $X(e^{j\omega})$ at $\omega_k = \frac{2\pi}{N}k$ (or, alternatively, by sampling $X(e^{j\Omega T})$ at $\Omega_k = \frac{2\pi}{TN}k$). Formally,

$$X[k] = \frac{1}{T}\sum_k X_c\left(j\left(\frac{2\pi}{NT}k - \frac{2\pi}{T}m\right)\right), \quad k = 0, 1, \ldots, N-1$$

Relationships between CTFT, DTFT, & DFT

- Sampling the DTFT of $x[n]$ is equivalent to the periodic repetition of $x[n]$ with period N or equivalently of $x_c(nT)$ with period NT. The result is

$$\tilde{x}[n] = \sum_k x_c(nT - NTk)$$

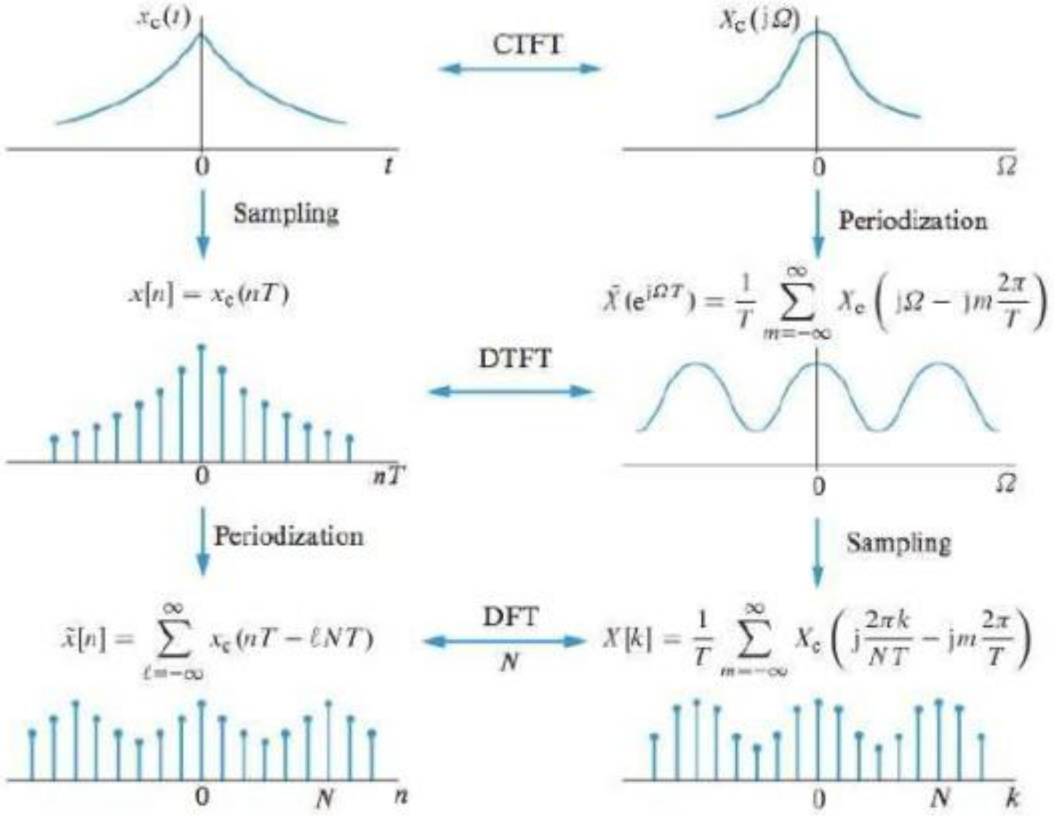
- Therefore, we have the following N-point DFT pair

$$\sum_k x_c(nT - NTk) \Longleftrightarrow \frac{1}{T}\sum_k X_c\left(\jmath\left(\frac{2\pi}{NT}k - \frac{2\pi}{T}m\right)\right)$$

where $0 \leq n \leq N-1$ and $0 \leq k \leq N-1$.

- The above relation reveals a frequency-domain aliasing caused by time-domain sampling and a time-domain aliasing caused by frequency-domain sampling (which, in turn, explains the inherent periodicity of the DFT).

Relationships between CTFT, DTFT, & DFT



226