POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

DTFT:
Discrete Time
Fourier Transform

DTFT definition

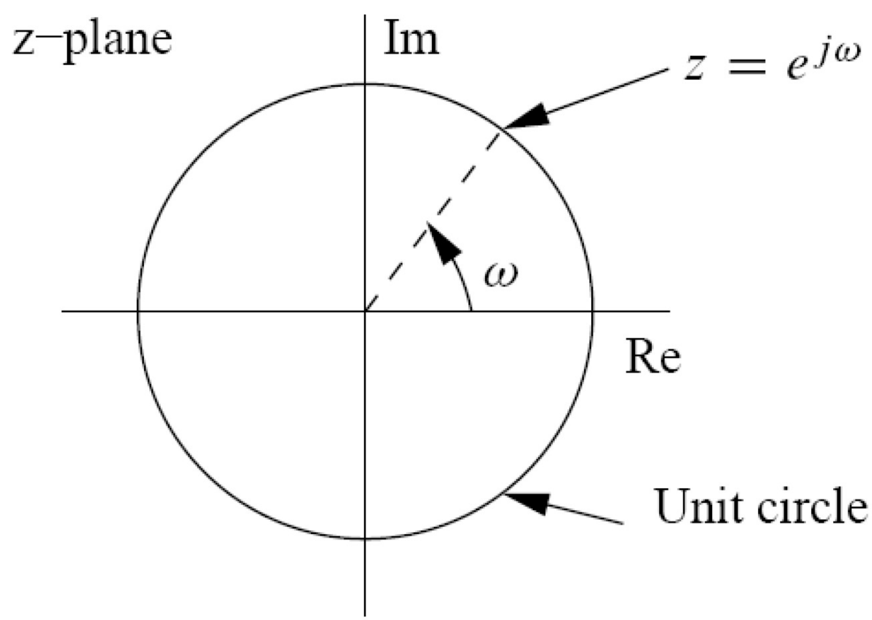
- The Discrete Time Fourier Transform (DTFT) of a sequence x(n) is defined as

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n}$$

- The DTFT is continuous
- f is the normalized frequency in $[-0.5, 0.5)$
- $\omega = 2\pi f$ is the normalized angular frequency in $[-\pi, \pi)$
- DTFT is periodic with period = 1/(sampling time), period = 2π in normalized omega, period = 1 in normalized frequency.
- Usually, we analyze the DTFT inside one period of frequencies.

Properties of DTFT

- $\text{DTFT}\{x(n) * y(n)\} = X(f) \cdot Y(f)$
- $\text{DTFT}\{x(n-k)\} = X(f) \cdot e^{-j2\pi f k}$
- If x(n) is real, $X(f) = X^*(-f)$
- Relationship with Z-transform: $X(f) = X(z)\Big|_{|z|=1}$



Filter frequency response

- Given a filter h(n), we can evaluate its frequency response H(f) (its DTFT) just by evaluating H(z) on the unitary circle.
- Given

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}}$$

- In MATLAB, we can compute an approximation of H(f) using [H_f, omega] = freqz(b, a, n, ‘whole’)

Filter frequency response

- In MATLAB, we can compute an approximation of H(f) using [H_f, omega] = freqz(b, a, n, 'whole'):
  - 'b' contains numerator coefficients (from b_0 to b_N)
  - 'a' contains denominator coefficients (from a_0 to a_D).
  - 'n' is the number of samples of H(f) we want.
  - 'H_f' is an approximation of H(f) on n samples.
  - 'omega' is defined between 0 and 2π.

Filter frequency response

- Given a filter h(n), we can evaluate its frequency response H(f) just by evaluating H(z) on the unitary circle.
- Given

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} \quad \Rightarrow \quad H(f) = H(z)\Big|_{|z|=1}$$

- The amplitude is

$$|H(f)| = |H(z)|\Big|_{|z|=1} = \left.\frac{|\sum_{k=0}^{N} b_k z^{-k}|}{|\sum_{k=0}^{D} a_k z^{-k}|}\right|_{|z|=1}$$

- The phase is

$$\angle(H(f)) = \angle(H(z))\Big|_{|z|=1} = \angle\left(\sum_{k=0}^{N} b_k z^{-k}\right) - \angle\left(\sum_{k=0}^{D} a_k z^{-k}\right)\Bigg|_{|z|=1}$$

Ex 12: filter frequency response

- Given $H(z) = 1 - 0.8z^{-1}$
- Compute its frequency response using N = 100 points and plot one period (omega in $[-\pi, \pi)$)
- Which are the zeros and poles?
- How does the phase behave?
- Try also H(z) = 1 - z^-1, H(z) = 1 + z^-1, H(z) = 1 – 3z^-1
- In which cases are zeros minimum-phase?
- When are zeros maximum-phase?
- Write a function to state if zeros are minimum or maximum phase

POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

DFT:
Discrete Fourier Transform

DFT definition

- Given a sequence x(n) defined on N samples,

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n}$$

- X(k) is defined as the DFT of the sequence. With respect to DTFT:

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n}$$

- DFT is discrete both in time and frequency and defined on N samples. It is like sampling one period of X(f) in f= k/N.

IDFT definition

- Given a sequence X(k) defined on N samples,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{k}{N} n}$$

- x(n) is defined as the IDFT of the sequence and it is defined on N samples.

DFT as a matrix product

- Given a sequence x(n) defined on N samples, the DFT is:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n}$$

$$\mathbf{W}_N = \begin{array}{cc} & \\ k \downarrow & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j2\pi/N} & e^{-j4\pi/N} & \dots & e^{-j2\pi(N-1)/N} \\ 1 & e^{-j4\pi/N} & e^{-j8\pi/N} & \dots & e^{-j4\pi(N-1)/N} \\ 1 & \dots & \dots & \dots & \dots \\ 1 & e^{-j2\pi(N-1)/N} & e^{-j4\pi(N-1)/N} & \dots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix} \\ & \xrightarrow{\hspace{4em}} \\ & n \end{array}$$

$$\mathbf{X} = \mathbf{W}_N \cdot \mathbf{x}$$

Ex 13.a: DFT

- Given

$$y(n) = -2y(n-1) - y(n-2) + x(n) + 2\rho cos(\theta)x(n-1) + \rho^2 x(n-2)$$

- $\rho = 0.9, \theta = \pi/8$
- The sequence is defined for n in [0, N-1], N = 1000.
- Which is the expression of h(n)? Compute it with 'filter'.
- Which is the amplitude of H(f)? Compute it with 'freqz' using N samples, for omega = [0, 2pi) (use command 'whole')
- Plot the amplitude of H(f) as a function of omega.
- Compute and plot the DFT of h(n) as $\mathbf{H} = \mathbf{W}_N \cdot \mathbf{h}$
- Are the two results equal? Comment on the results.

FFT and IFFT in MATLAB

- MATLAB implements DFT and IDFT in a fast way.
- To compute DFT, use ‘fft’
- To compute IDFT, use ‘ifft’

Ex 13.b: DFT

- Given

$$y(n) = -2y(n-1) - y(n-2) + x(n) + 2\rho cos(\theta)x(n-1) + \rho^2 x(n-2)$$

- $\rho = 0.9, \theta = \pi/8$
- The sequence is defined for n in [0, N-1], N = 1000.
- Which is the expression of h(n)? Compute it with 'filter'.
- Which is the amplitude of H(f)? Compute it with 'freqz' using N samples, for omega = [0, 2pi) (use command 'whole')
- Plot the amplitude of H(f) as a function of omega.
- Compute and plot the DFT of h(n) as $\mathbf{H} = \mathbf{W}_N \cdot \mathbf{h}$
- Are the two results equal? Comment on the results.
- Compute DFT of h(n) using 'fft' and check if results are equal.

DTFT-DFT relationship

- The DTFT is the Fourier transform of a discrete signal.
- DTFT is continuous over frequencies.
- The discrete signal is assumed being known from -∞ to ∞.
- Sampling in time = Periodicity in Fourier
- → DTFT is periodic with period 1/sampling time.
- Since DTFT is periodic, we usually analyze it over one period.
- But DTFT is continuous over frequencies → we need to sample it for numerical computing.

DTFT-DFT relationship

- DFT can be seen as sampling the DTFT over a single period. The sampling rate is 1/N.
- Sampling in Fourier = Periodicity in time
- → IDFT is periodic with period N.
- When computing the DFT of a discrete signal x(n) over N samples, we are assuming x(n) to be periodic with period N.
- We evaluate only one period of x(n).
- DFT is the DTFT of this periodic signal, evaluated over only one period.

DFT and IDFT periodicity

- Both DFT and IDFT are periodic
  with a period of N samples.

$$X(k + tN) = X(k) \quad \forall t \in [-\infty, \infty]$$

$$x(n) = \mathrm{IDFT}(X(k)) = x(n + tN) \quad \forall t \in [-\infty, \infty]$$

Ex 14: DFT-zero padding

- Given a FIR filter h(n) = 1, n in [0, 19]
- Evaluate the DFT of the filter
- Visualize the DFT versus normalized frequency [-0.5, 0.5)
- Pad the array h(n) with zeros until reaching 100 samples
- You can use MATLAB function ‘padarray’
- Evaluate the DFT of the padded h(n) and visualize it
- Are the two DFTs equal? Comment on the results

Ex 15: DFT-periodic signals

- Given a sinusoidal signal with frequency = 2Hz and duration = 1.3 seconds, sampled with Fs = 50 Hz,
- Plot the first N = 50 samples of the sinusoid vs time.
- Compute its DFT on N samples and visualize it vs frequencies.
- Compute the DFT of the complete signal and visualize it.
- Do the two Fourier spectra coincide?
- Which frequencies should correspond to peaks in Fourier? Why?
- How to find the actual peaks of the sinusoid in the Fourier domain?

Ex 16: DFT-periodic signals

- Given a sinusoidal signal with frequency = 2Hz defined over N = 50 samples, sampled with Fs = 50 Hz
- Add this signal to a second sinusoid with frequency 2.2Hz, equal duration and sampling rate.
- Plot the global signal vs time.
- Compute the DFT. Can you see the frequency peaks?
- How to see the correct Fourier spectrum?

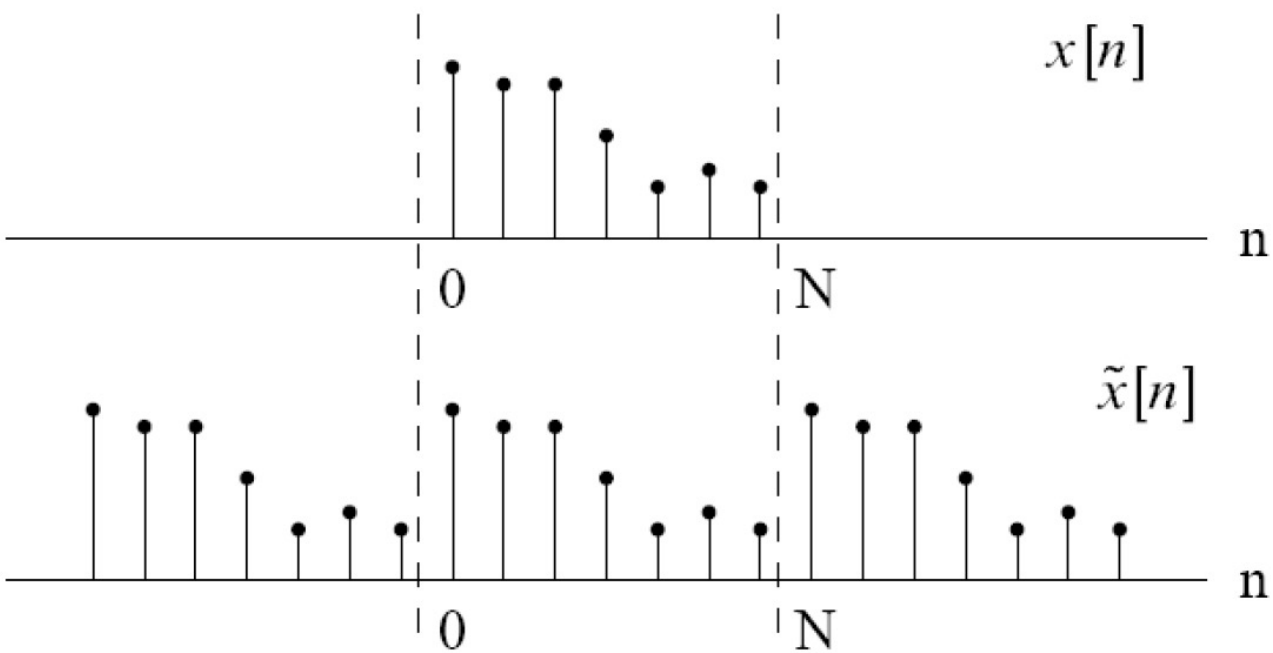
Zero padding and frequency resolution

- Zero padding helps interpolating the DFT samples (it increases the density of the spectrum) but does not increase the spectrum resolution! It does not introduce new information.
- To increase the spectrum resolution, we need more signal information = more measurements → we must evaluate the signal over a longer time window.

DFT and IDFT periodicity

- Given a non periodic sequence x(n) defined over N samples, its N-periodic extension is defined as:

$$\tilde{x}(n) = \sum_{t=-\infty}^{\infty} x(n - tN)$$



DFT and IDFT periodicity

- Given a non periodic sequence x(n) defined over N samples, its N-periodic extension is defined as:

$$\tilde{x}(n) = \sum_{t=-\infty}^{\infty} x(n - tN)$$

- The same can be done with the DFT, defining

$$\tilde{X}(k) = \sum_{t=-\infty}^{\infty} X(k - tN)$$

Properties of DFT

- Many properties remains the same as DTFT-transform, but:

$$\boxed{X_1(k) \cdot X_2(k) \neq x_1(n) * x_2(n)}$$

- Cyclic convolution property:

$$\boxed{X_1(k) \cdot X_2(k) = x_1(n) \circledast x_2(n)}$$

Cyclic/Circular convolution $\circledast$

- Cyclic (or circular) convolution between x(n) and y(n) can be seen as the linear convolution between x(n) and the periodic extension of y(n), taken at the first N samples (n in [0, N-1]).

$$x(n) \circledast y(n) = \sum_{m=0}^{N-1} x(m)\tilde{y}(n-m)$$

Ex 17: Cyclic convolution

- Given $x(n) = \delta(n-2)$, defined for n in [0, 4]
- $y(n) = 5\delta(n) + 4\delta(n-1) + 3\delta(n-2) + 2\delta(n-3) + \delta(n-4)$
- Compute z(n) as the linear convolution between x and y.
- Which is the support of z(n)?
- Compute the cyclic convolution between x and y.
- Try using matlab function 'cconv' and check the result.

Ex 18.a: Cyclic convolution

- Given $x(n)$ = rectangular pulse of width 6, n in [0, 5]
- Given $y(n) = x(n)$
- Compute $z(n)$ as the linear convolution between x and y.
- Compute the cyclic convolution between x and y.
- Compute the product $X(k)Y(k)$
- Check that the IDFT of $X(k)Y(k)$ and the cyclic convolution give the same result.

Linear vs Circular convolution

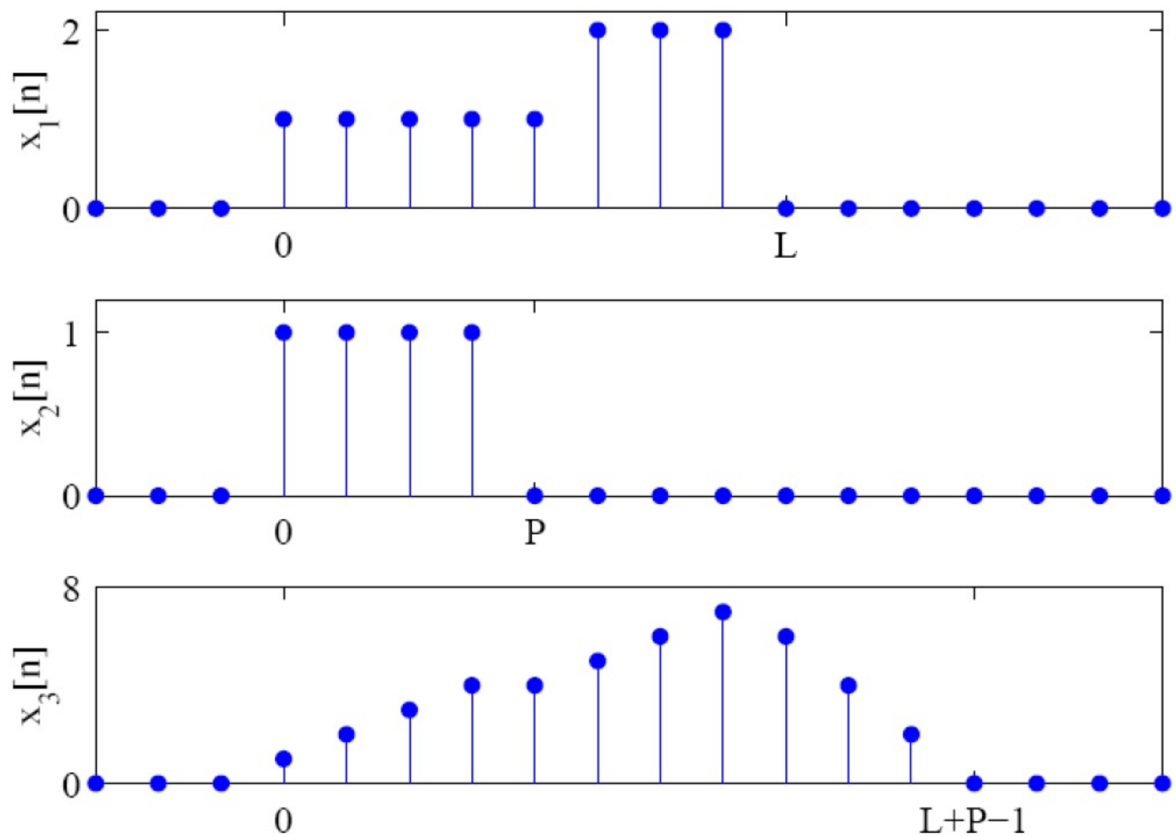
- Circular convolution property is fancy:

$$X_1(k) \cdot X_2(k) = x_1(n) \circledast x_2(n)$$

- To evaluate the result of circular convolution it's enough to compute IDFT($X_1(k) \cdot X_2(k)$)
- It is more efficient than computing $x_1(n) \circledast x_2(n)$
- BUT: LTI systems such as communication, radar, audio and image systems, etc... they all work with linear convolution! How to compute linear convolution in DFT domain?

Relationship between linear and cyclic conv

- Given a sequence with length L and a sequence with length P, the maximum length of the linear convolution is Nc = L + P -1



Relationship between linear and cyclic conv

- Given a sequence with length L and a sequence with length P, the maximum length of the linear convolution is Nc = L + P -1
- The cyclic convolution computed over a generic number N of samples is equal to the linear convolution if N is long enough to avoid periodic artifacts. Therefore,

$$N \geq L + P - 1$$

- How to do it? Pad with zeros the two sequences until reaching N samples.

Ex 18.b: Cyclic convolution

- Given $x(n)$ = rectangular pulse of width 6, n in $[0, 5]$
- Given $y(n) = x(n)$
- Compute $z(n)$ as the linear convolution between x and y.
- Compute the cyclic convolution between x and y.
- Compute the product $X(k)Y(k)$
- Check that the IDFT of $X(k)Y(k)$ and the cyclic convolution give the same result.
- Find the value of N such that cyclic convolution returns the same result as linear convolution.
- Compute the cyclic convolution over N samples.
- Find the same result using DFT.

Long convolutions

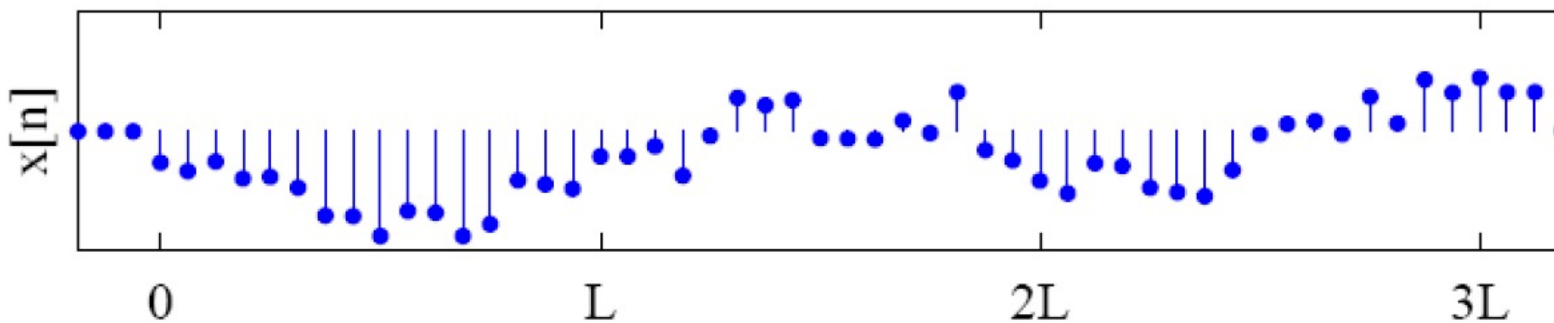
- The input sequence x(n) to a system can be very long
- x(n) can be unknown as well (real-time applications).

Zero-padding can be unfeasible

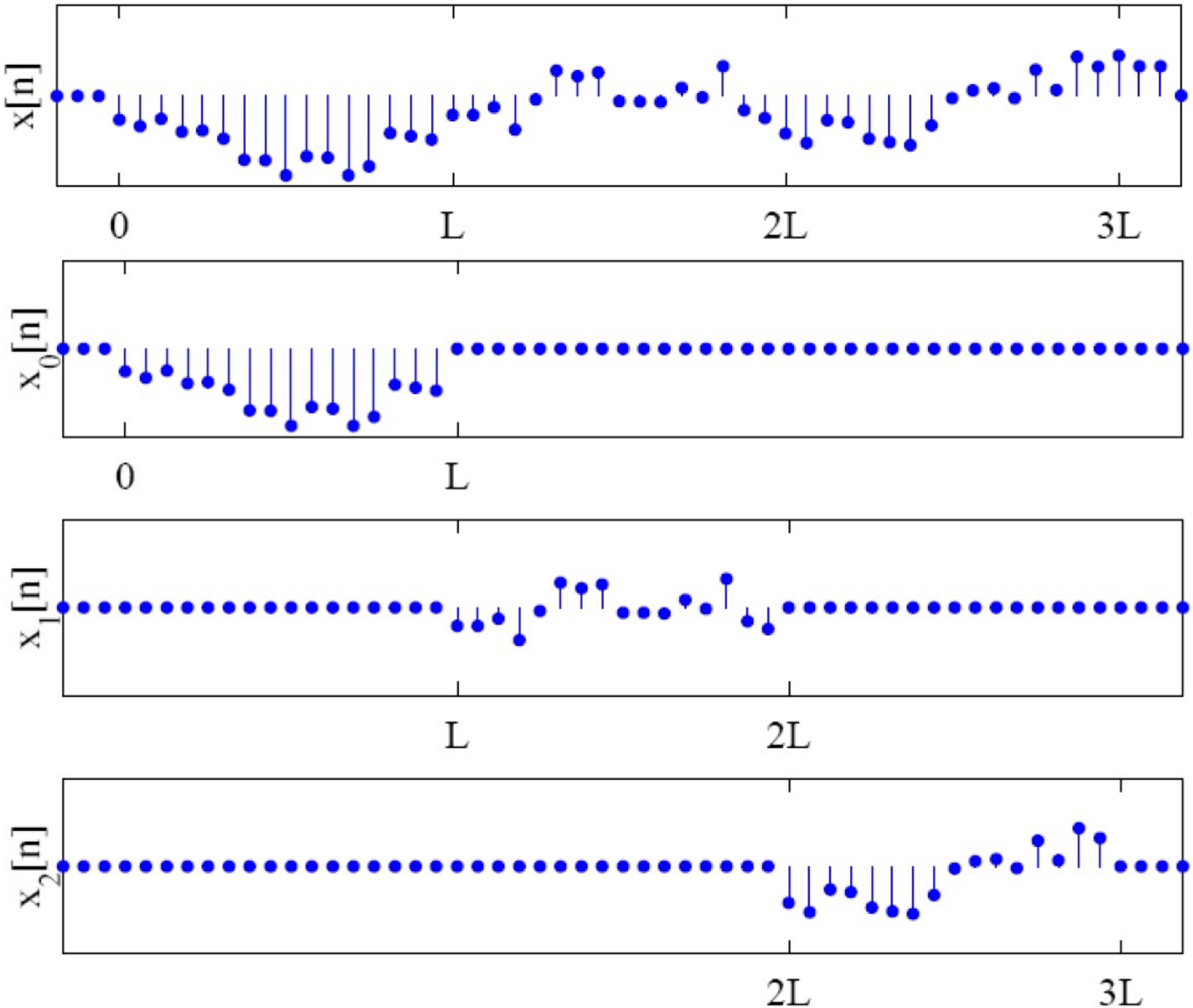
- We can segment the input signal into smaller blocks and process them separately.

Long convolutions: Overlap and Add

- Suppose the impulse response h(n) has length P.
  Decompose the input signal into **non-overlapped** blocks with length L



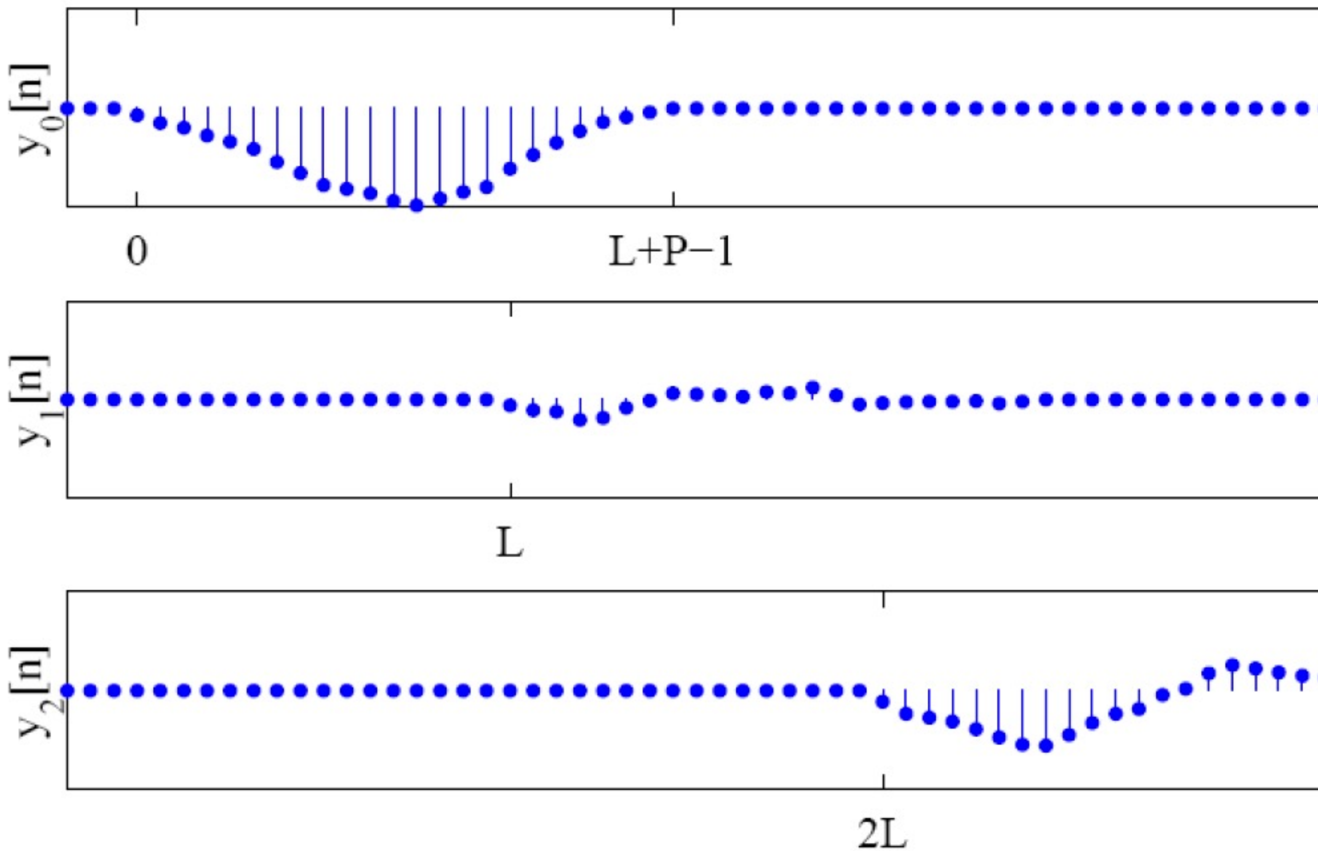
Long convolutions: Overlap and Add



Long convolutions: Overlap and Add

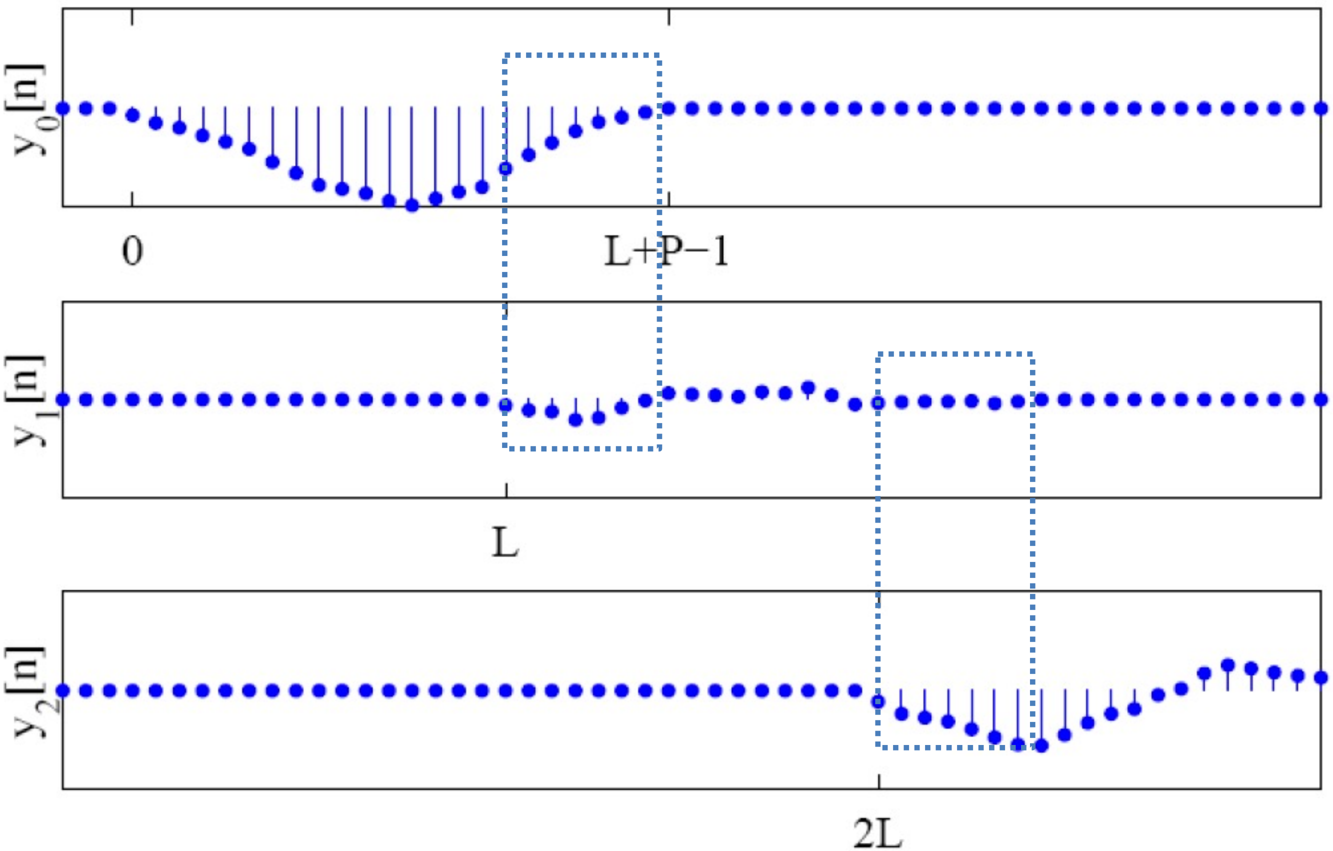
- For each block, compute the output as

$$\mathrm{IDFT}(X_n(k) \cdot H(k)), k \in [0, N = L + P - 1)$$



Long convolutions: Overlap and Add

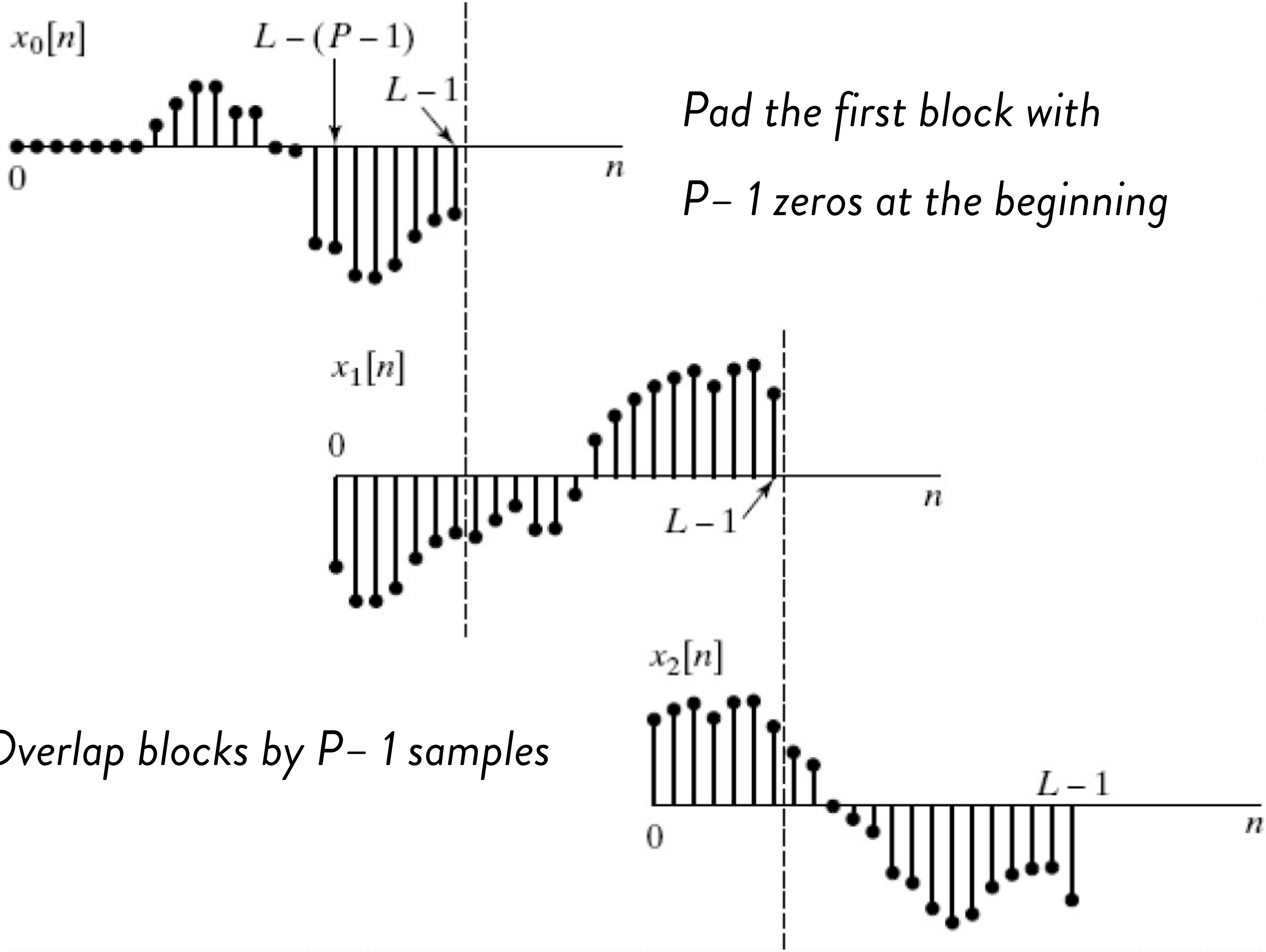
- Sum the overlapping portions between the results.



Long convolutions: Overlap and Save

- Suppose the filter has length P. Decompose the input signal into **overlapped** blocks with length L > P with overlap P – 1.
- The circular convolution evaluated over L samples is different from the linear convolution only in the first P – 1 samples (periodic artifacts).

Long convolutions: Overlap and Save

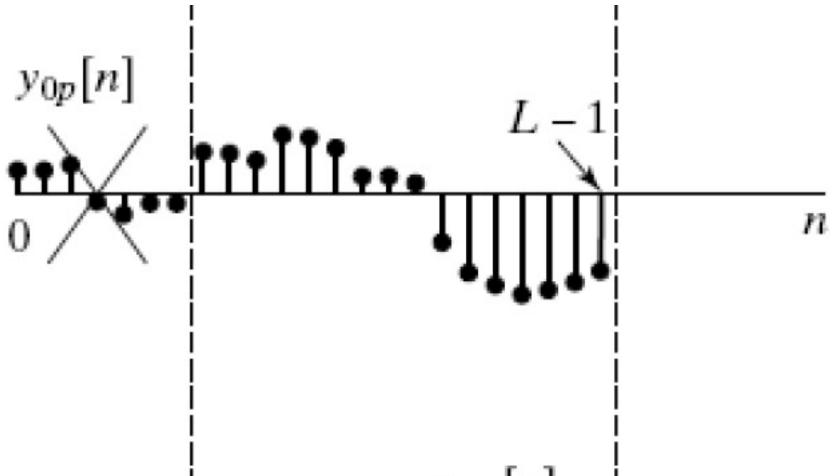


Pad the first block with

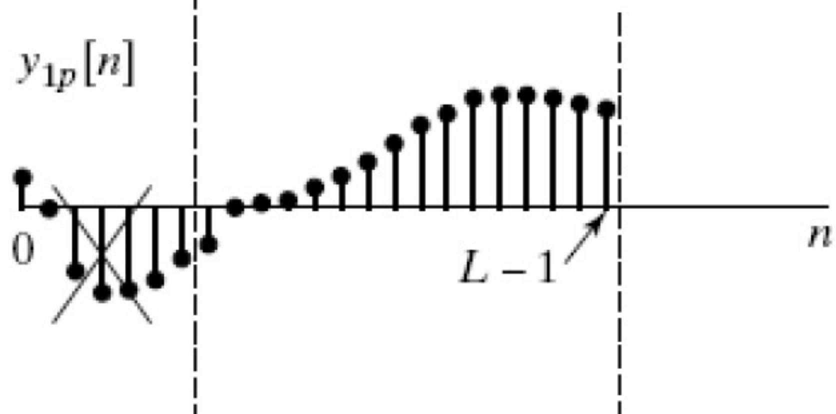
P– 1 zeros at the beginning

Overlap blocks by P– 1 samples

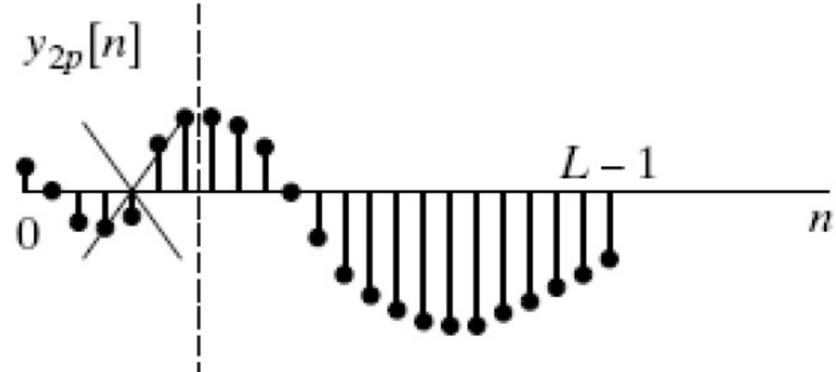
Long convolutions: Overlap and Save



For each block, compute the Circular convolution over L samples



Discard the first P – 1 samples



Ex 19: Overlap and Add / Save

- Given $x(n) = n + 1$, n in [0, 20]
- $h(n)$ is a FIR filter, $h(n) = [1, 0, -1]$, n in [0, 2]
- Compute the linear convolution between x and h
- Compute the same result with the overlap and add method, using $L = 6$.
- Compute the same result with the overlap and save method, using $L = 6$.