Lecture 9
(Part 2)

The Fast Fourier Transform (FFT) Applications

Fast FIR Filtering

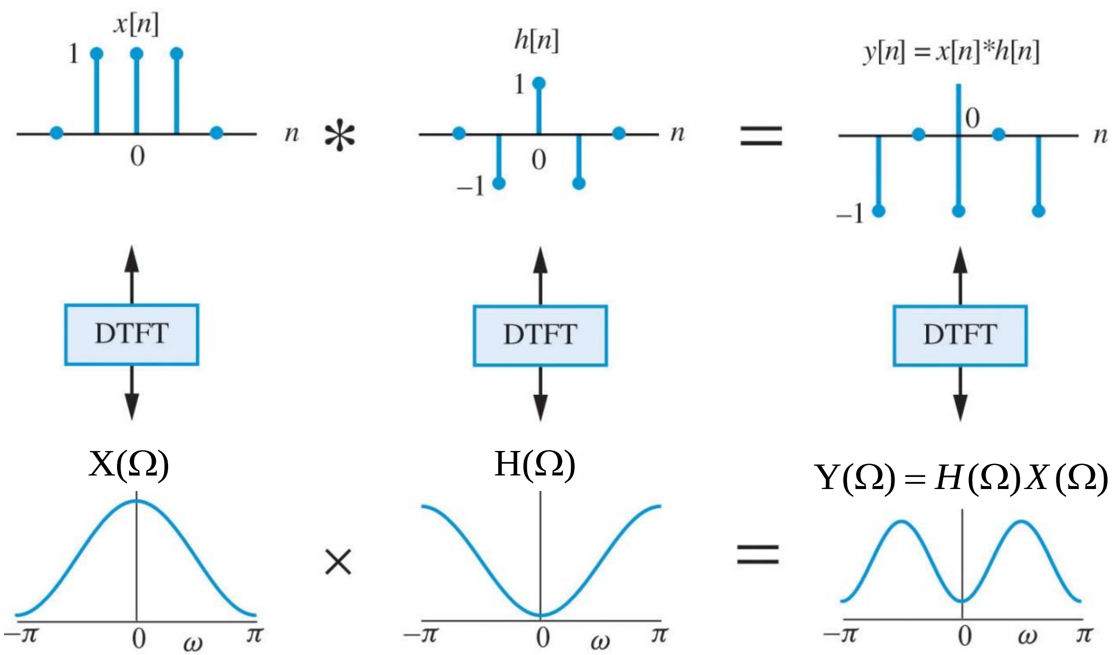
Dr. Dimitris Agrafiotis, Dept. Of Electrical & Electronic Engineering

FFT Applications

Fast Filtering

Linear convolution

DTFT: Convolution in the time domain is equivalent to multiplication in the frequency domain



FIR filtering is linear convolution in time - Implement filtering as multiplication in frequency

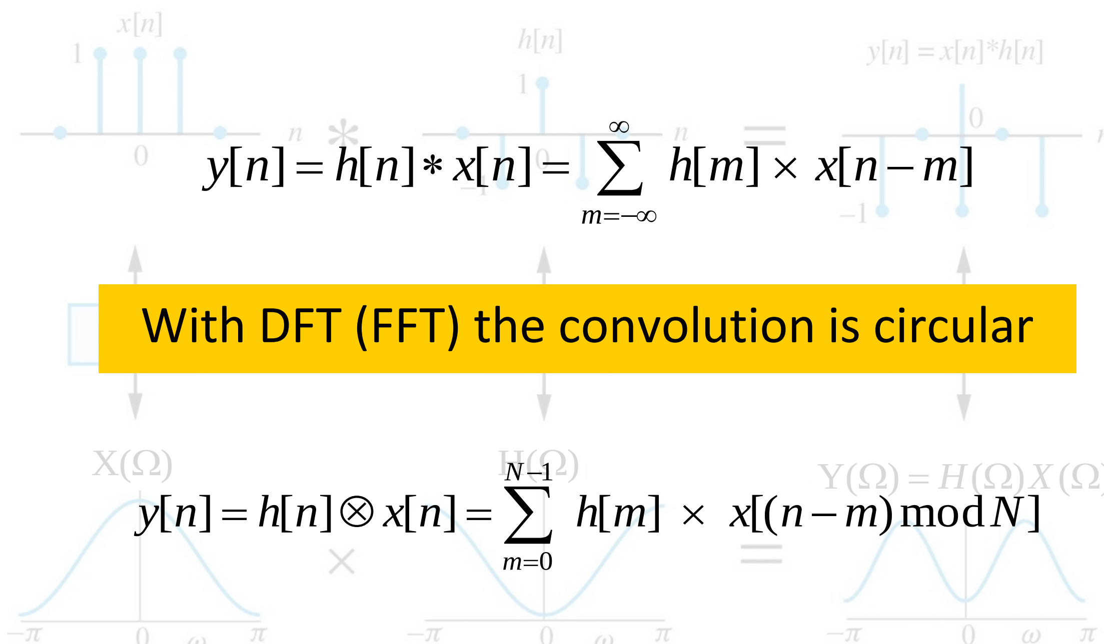
Use FFT to make FIR filtering with long filters / long sequences efficient

FFT Applications

Fast Filtering

Linear convolution

DTFT: Convolution in the time domain is equivalent to multiplication in the frequency domain



$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} h[m] \times x[n-m]$$

With DFT (FFT) the convolution is circular

$$y[n] = h[n] \otimes x[n] = \sum_{m=0}^{N-1} h[m] \times x[(n-m) \bmod N]$$

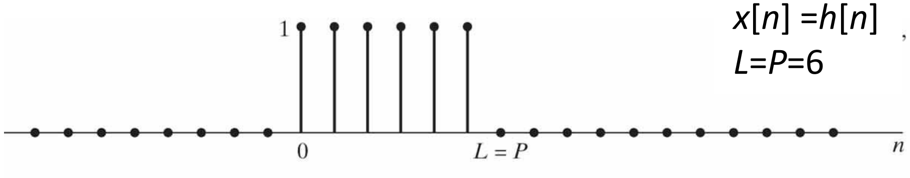
FIR filtering is linear convolution in time - Implement filtering as multiplication in frequency

Use FFT to make FIR filtering with long filters / long sequences efficient

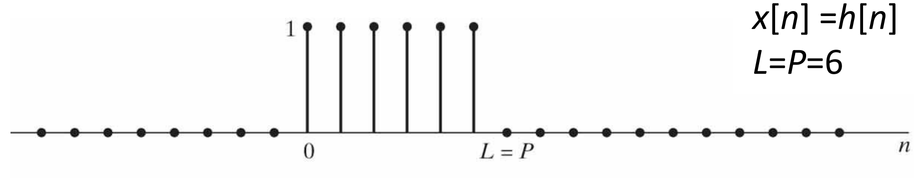
Fast Filtering

Linear vs. Circular convolution

Signals to be convolved



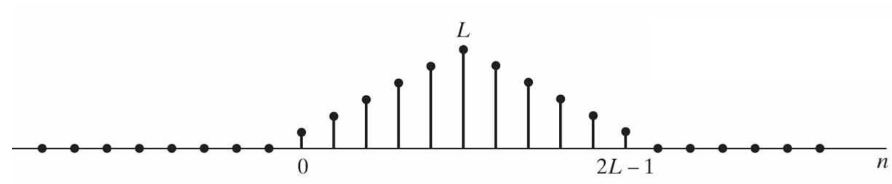
$x[n] = h[n]$
$L=P=6$



$x[n] = h[n]$
$L=P=6$

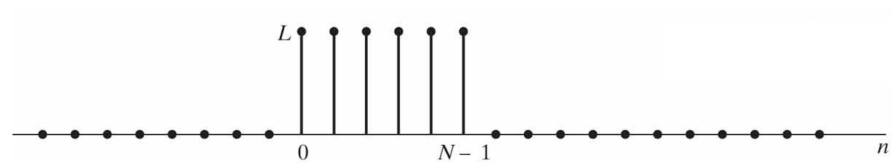
Result with linear convolution

$$y[n] = h[n] * x[n] = \sum_{m=0}^{L-1} h[m] \times x[n-m]$$



Result with circular convolution

$$y[n] = h[n] \otimes x[n] = \sum_{m=0}^{N-1} h[m] \times x[(n-m) \bmod N]$$



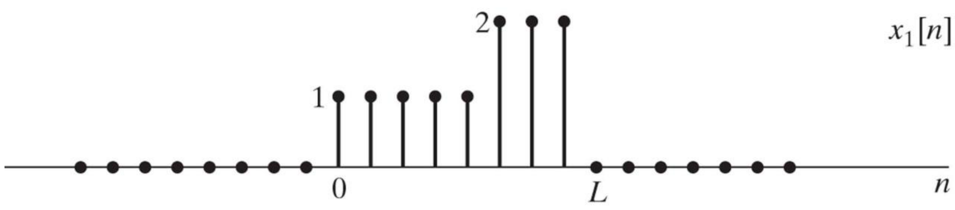
FFT Applications

Fast Filtering

Performing linear convolution using the FFT

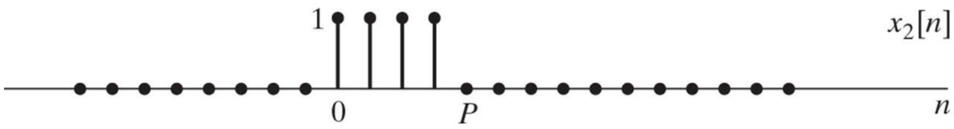
$x_1[n]$: Length L

$$x_1[n] = 0, \quad n < 0 \text{ and } n \geq L$$



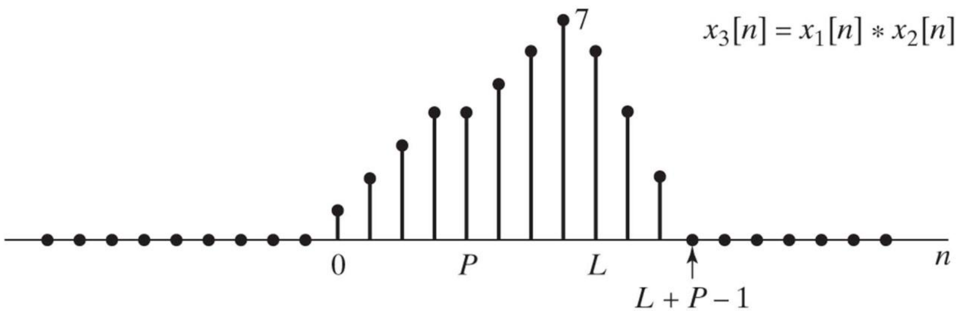
$x_2[n]$: Length P

$$x_2[n] = 0, \quad n < 0 \text{ and } n \geq P$$



y[n] : Length L+P-1

$$y[n] = x_1[n] * x_2[n] = \sum_{k=0}^{P-1} x_2[k] \times x_1[n-k]$$



FFT Applications

Fast Filtering

Performing linear convolution using the FFT

$x_1[n]$: Length L

$x_2[n]$: Length P

$$y[n] = x_1[n] * x_2[n] = \sum_{k=0}^{P-1} x_2[k] \times x_1[n-k]$$

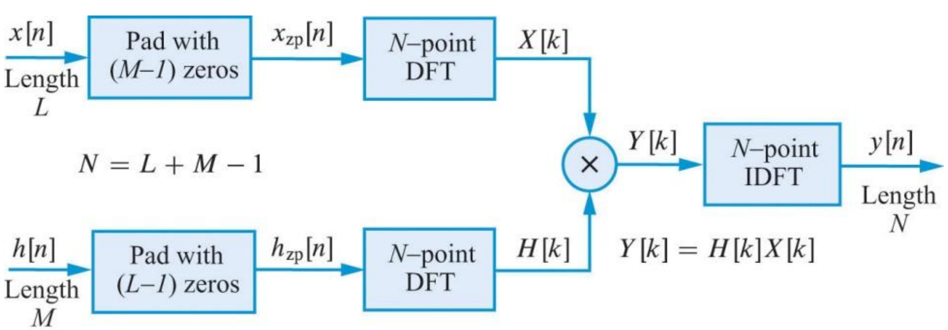
$y[n]$: Length L+P-1

$$\text{Y}(\Omega) = X_1(\Omega) X_2(\Omega) \overset{N \geq L+P-1}{\Leftrightarrow} \text{Y}[k] = Y(\Omega)_{\Omega = 2\pi k/N}, \ k = 0,1,\ldots, N-1$$

$$\overset{N \geq L+P-1}{\Leftrightarrow} \text{Y}[k] = X_1(\Omega) X_2(\Omega)_{\Omega = 2\pi k/N}, k = 0,1,\ldots, N-1 \overset{N \geq L+P-1}{\Leftrightarrow} \text{Y}[k] = X_1[k] \ X_2[k], k = 0,1,\ldots, N-1$$

N-point DFTs

1. Pad $x_1[n]$ with **P-1** & $x_2[n]$ with **L-1** zeros
2. Take the **N≥L+P-1** point DFT of $x_1[n]$ & $x_2[n]$
3. Multiply the DFTs
4. Perform the IDFT to get y[n]
5. $y[n] = x_1[n] \otimes x_2[n] = x_1[n] * x_2[n]$

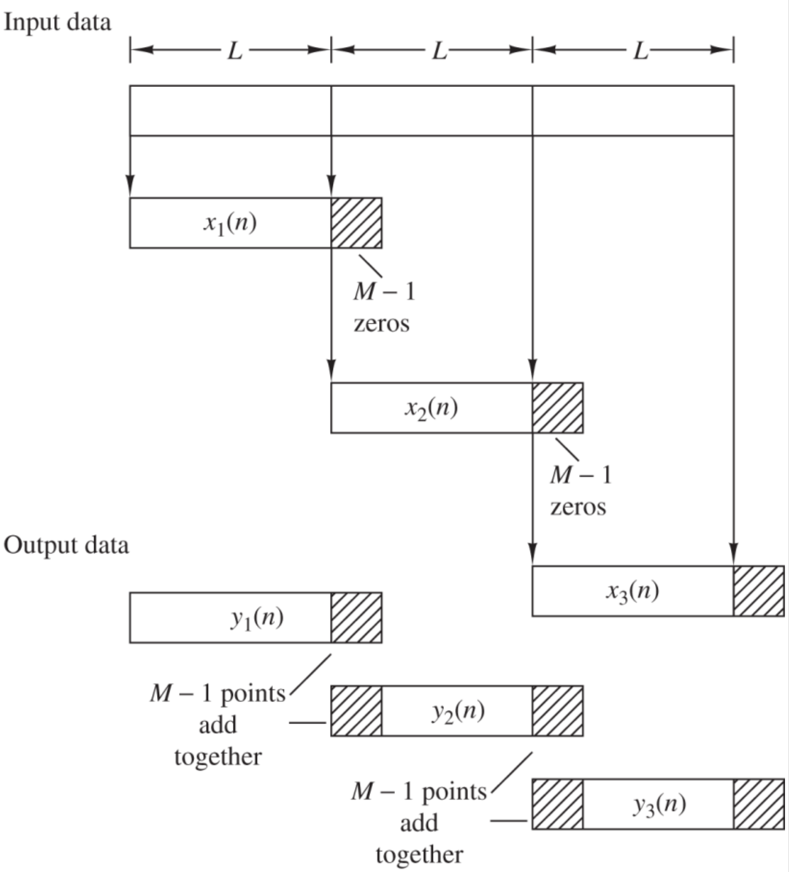


FFT Applications

Fast Filtering of Long Sequences

Linear filtering of long sequences on a block-by-block basis using the DFT

Overlap-Add



Linear FIR filtering by the overlap-add method.

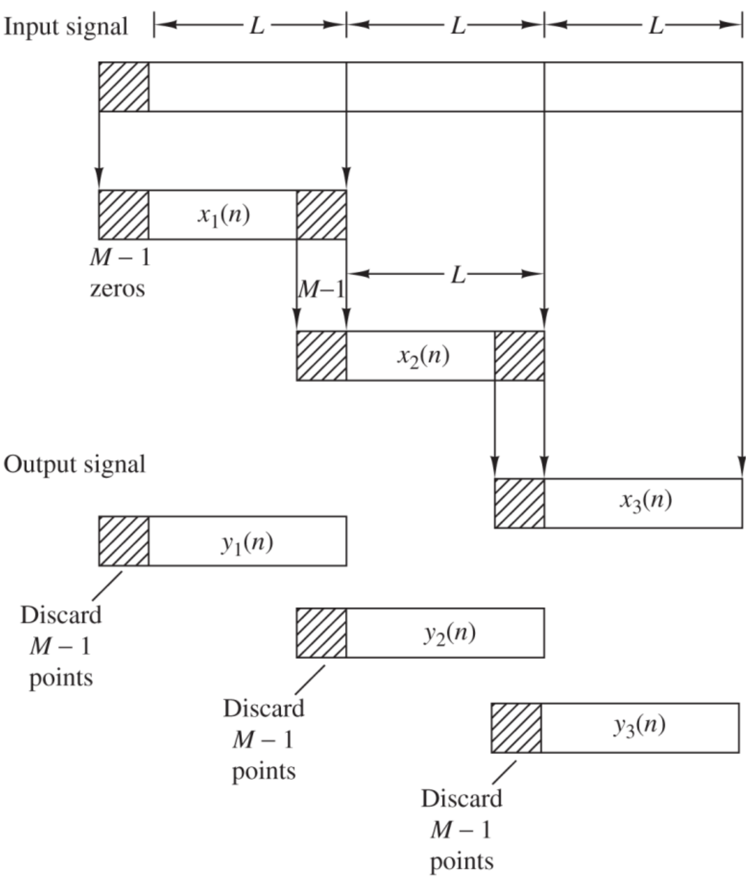
- Signal segmented into blocks of length L
- Filter length is M
- Size of data block is N=L+M-1
- DFT & IDFT are of length N
- Each block consists of the L data points of the block followed by M-1 zeros (padding)
- The filter is padded with L-1 zeros
- DFT/Multiplication/IDFT (length N)
- Overlap and add to next block

$$
\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \\ y[7] \\ y[8] \\ y[9] \end{bmatrix}
=
\begin{bmatrix}
h[0] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
h[1] & h[0] & 0 & 0 & 0 & 0 & 0 & 0 \\
h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 & 0 \\
0 & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 \\
0 & 0 & h[2] & h[1] & h[0] & 0 & 0 & 0 \\
0 & 0 & 0 & h[2] & h[1] & h[0] & 0 & 0 \\
0 & 0 & 0 & 0 & h[2] & h[1] & h[0] & 0 \\
0 & 0 & 0 & 0 & 0 & h[2] & h[1] & h[0] \\
0 & 0 & 0 & 0 & 0 & 0 & h[2] & h[1] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h[2]
\end{bmatrix}
\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}.
$$

Fast Filtering of Long Sequences

Linear filtering of long sequences on a block-by-block basis using the DFT

Overlap-Save



Linear FIR filtering by the overlap-save method.

- Signal segmented into blocks of length L
- Filter length is M
- Size of data block is N=L+M-1
- DFT & IDFT are of length N
- Each block consists of the last M-1 data points of the previous block (overlap) followed by the L new data
- The filter is padded with L-1 zeros
- DFT/Multiplication/IDFT (length N)
- First M-1 points aliased - Discard and keep last L points
- Last M-1 points of each block saved for next block

$$
\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \\ y[7] \\ y[8] \\ y[9] \end{bmatrix}
=
\begin{bmatrix}
h[0] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
h[1] & h[0] & 0 & 0 & 0 & 0 & 0 & 0 \\
h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 & 0 \\
0 & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 \\
0 & 0 & h[2] & h[1] & h[0] & 0 & 0 & 0 \\
0 & 0 & 0 & h[2] & h[1] & h[0] & 0 & 0 \\
0 & 0 & 0 & 0 & h[2] & h[1] & h[0] & 0 \\
0 & 0 & 0 & 0 & 0 & h[2] & h[1] & h[0] \\
0 & 0 & 0 & 0 & 0 & 0 & h[2] & h[1] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & h[2]
\end{bmatrix}
\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}.
$$