Circular Convolution

Sequences of length N and the N–point DFT

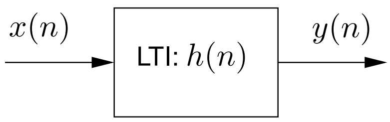
$$\mathrm{FFT}[x_1(n)] = X_1(k)$$
$$\mathrm{FFT}[x_2(n)] = X_2(k)$$
$$\mathrm{FFT}[x_3(n)] = X_3(k) = X_1(k)X_2(k)$$

$$\Longleftrightarrow$$

$$x_3(n) = \sum_{k=0}^{N-1} x_1(k)x_2((n-k)_{\mathrm{mod}\ N}) = x_1(n) \circledN x_2(n)$$

Notation: $x_1(n) \circledN x_2(n)$, circular convolution of length N.

Linear filtering using FFT



$$y(n) = x(n) * h(n)$$
$$Y(f) = X(f)H(f)$$

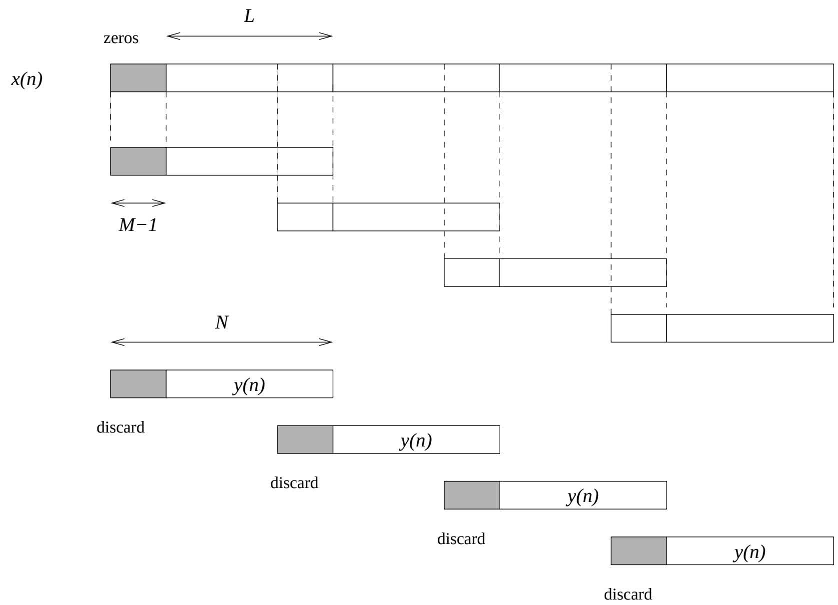
The FFT algorithm can be used to calculate $y(n)$ very **efficiently**.

Problem: $X(k)H(k)$ corresponds to **circular convolution**.

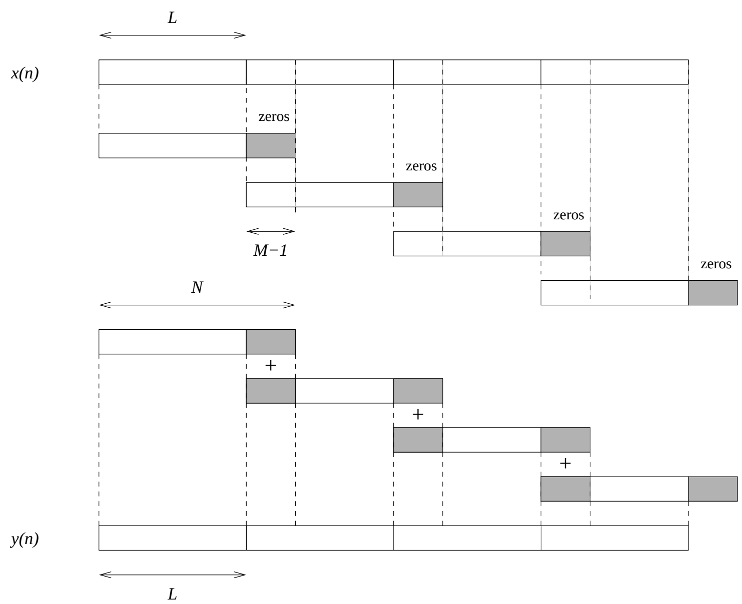
Set $N \geq$ length$\{h(n)\}$+length$\{x(n)\} - 1$ to avoid aliasing from the circular convolution.

Long input sequences are divided into segments, using **overlap-save** or **overlap-add**.

Overlap-Save



Overlap-Add



COMPUTATIONS

Computational complexity of overlap-save and overlap-add:

(filter length: M)

$$\frac{N \log_2(2N)}{N - M + 1} \text{ multiplications per sample}$$

Direct implementation:

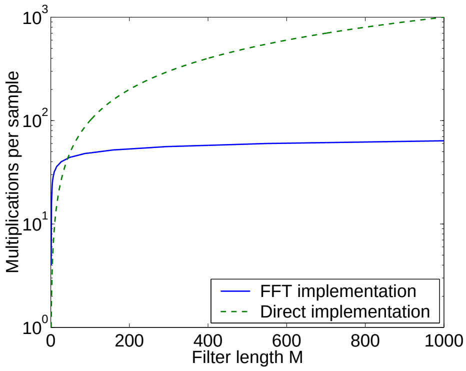
$$M \text{ multiplications per sample}$$

COMPUTATIONS (CONT.)

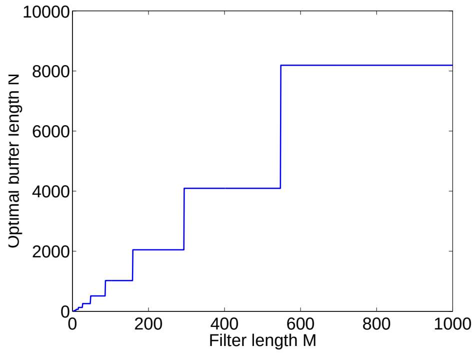
There is an optimal value of $N = 2^p$ for each filter length M:

M	N	mult./sample
5	16	6.67
10	64	8.15
15	64	8.96
20	128	9.39
50	512	11.06
100	1024	12.18
1000	8192	15.94

Computations (cont.)



Number of multiplications.



Optimal FFT length.