ECE310: Lecture 32: Fast Convolution

Let $x[n]$: length 'L' sequence

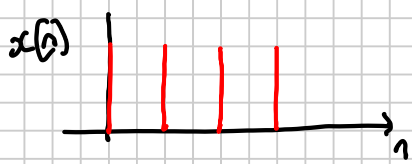
$h[n]$: length 'M' sequence

RECALL: Linear Convolution:

$$y[n] = \sum_{m=0}^{N'-1} x[m]\,h[n-m]$$

$\boxed{y[n] \equiv \text{Length } N' = L+M-1}$, $\boxed{L = M = N \Rightarrow N' = 2N-1}$

Example: $x[n] = [1 \ 1 \ 1 \ 1]$ $h[n] = [1 \ 1 \ 1 \ 1]$

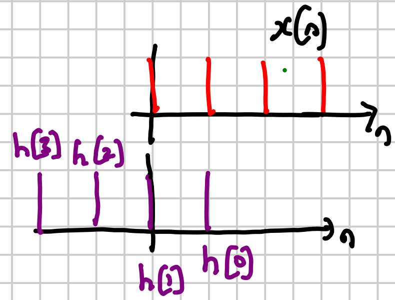


$$y[n] = \sum_{m=0}^{6} x[m]\,h[n-m]$$

(1) $y[0] = x[0]\,h[0-0] + x[1]\,\cancel{h[0-1]}^{\,0} + x[2]\,\cancel{h[0-2]}^{\,0} + x[3]\,\cancel{h[0-3]}^{\,0} + \dots$

$y[0] = x[0]\,h[0] = 1$

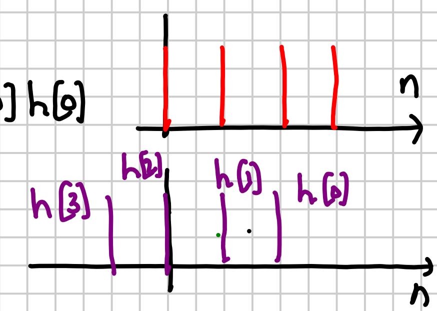
(2) $$y[1] = \sum_{m=0}^{6} x[m]\,h[1-m]$$



$y[1] = x[0]\,h[1] + x[1]\,h[0]$

$\Rightarrow \boxed{y[1] = 2}$

(3) $y[2] = x[0]\,h[2] + x[1]\,h[1] + x[2]\,h[0]$



$\Rightarrow \boxed{y[2] = 3}$

Proceeding as above:

$y[2] = 3$, $y[3] = 4$, $y[4] = 3$, $y[5] = 2$, $y[6] = 1$

In the frequency domain:

$$Y_d(\omega) = X_d(\omega)\, H_d(\omega)$$

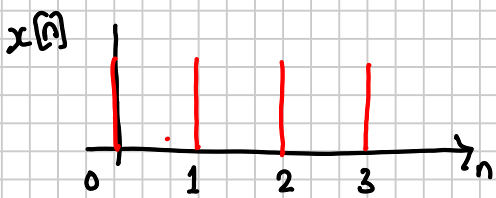
$$X_d(\omega) = \text{DTFT}\{x[n]\}, \quad H_d(\omega) = \text{DTFT}\{h[n]\}$$

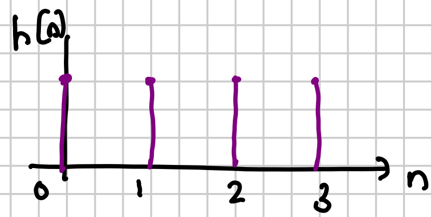
$$\Rightarrow \boxed{y[n] = \text{IDTFT}\{Y_d(\omega)\}}$$

(2) Recall: Cyclic Convolution

$$y[n] = \sum_{m=0}^{N'-1} x[m]\, h[(n-m)_{N'}], \quad N' = \max(L, M)$$

$$x[n] = [1 \;\; 1 \;\; 1 \;\; 1] \qquad h[n] = [1 \;\; 1 \;\; 1 \;\; 1]$$





$$y[n] = \sum_{m=0}^{3} x[m]\, h[(n-m)_4]$$

(1)

$$y[0] = x[0]h[0] + x[1]h[(0-1)_4] + x[2]h[(0-2)_4] + x[3]h[(0-3)_4]$$

$$\Rightarrow y(0) = x[0]h[0] + \underbrace{x[1]h[3] + x[2]h[2] + x[3]h[1]}_{\text{not zero}}$$

$$\Rightarrow \boxed{y[0] = 4}$$

(2)

$$y(1) = \sum_{m=0}^{3} x[m]\, h[(1-m)_4]$$

$$= x[0]h[(1-0)_4] + x[1]h[(1-1)_4] + x[2]h[(1-2)_4] + x[3]h[(1-3)_4]$$

$$y(1) = x[0]h[1] + x[1]h[0] + \underbrace{x[2]h[3] + x[3]h[2]}_{\text{not zero}}$$

$$\boxed{y[1] = 4}$$

proceeding as above, $\boxed{y[2] = y[3] = 4}$

Note : $y[3]$ is same both in linear & cyclic convolution

In the transform domain :

$$\boxed{Y(k) = X(k)\,H[k]} \qquad k = 0, \ldots, N-1$$

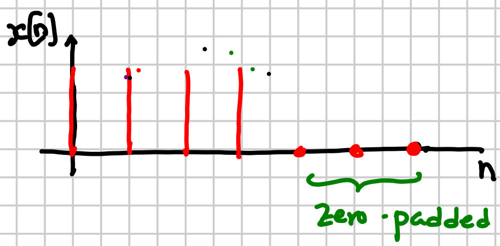
$$H[k] = \text{DFT}\{h[n]\}, \qquad X(k) = \text{DFT}\{x[n]\}$$

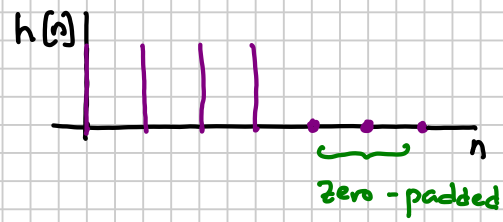
$$\Rightarrow \boxed{y[n] = \text{IDFT}\{Y(k)\}} \qquad n = 0, 1, \ldots, N-1$$

Cyclic Convolution can be implemented using DFT

Linear Convolution from Cyclic Convolution :

Zero-pad the sequences to length $\boxed{N = L+M-1}$





$$y[n] = \sum_{m=0}^{N-1} x[m]\,h[((n-m))_N] \qquad \boxed{N = 7}$$

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

$$y[0] = \sum_{m=0}^{N-1} x(m)\,h[((0-m))_7]$$

$$= x[0]\,h(0) + x[1]\,h[6] + x[2]\,h[5] + \ldots$$

$$\Rightarrow \boxed{y[0] = 1}$$

proceeding as above, $\boxed{y[n] = [1 \;\; 2 \;\; 3 \;\; 4 \;\; 3 \;\; 2 \;\; 1]}$

= Linear Conv. output

Cyclic convolution can be implemented in the transform domain using DFT. Recall that DFT can be computed using FFT at reduced complexity. For filtering long data sequences, the transform domain implementation can be more efficient.

Computational Complexity :

Computational Complexity of DSP algorithms is measured in terms of the number of multiply - accumulates (MA's)

Example: ① $x[n] = [1 \quad 1 \quad 1 \quad 1]$ $\quad h[n] = [1 \quad 1 \quad 1 \quad 1]$

①
$$\begin{array}{l} \quad\;\;[1 \quad 1 \quad 1 \quad 1] \\ [1 \quad 1 \quad 1 \quad 1] \end{array} \rightarrow \begin{array}{l} y[0] = 1 \times 1 \\ \# \text{ of MA} = 1 \end{array}$$

②
$$\begin{array}{l} \quad\;\;[1 \quad 1 \quad 1 \quad 1] \\ [1 \quad 1 \quad 1 \quad 1] \end{array} \longrightarrow y[1] = 1 \times 1 + 1 \times 1$$

$\# \text{ of MA's} = 2$

③
$$\begin{array}{l} \quad\;\;[1 \quad 1 \quad 1 \quad 1] \\ [1 \quad 1 \quad 1 \quad 1] \end{array} \longrightarrow y[2] = 1 \times 1 + 1 \times 1 + 1 \times 1$$

$\Rightarrow \text{MA's} = 3$

$\therefore$ to Compute $y[0] \rightarrow y[3]$ we need

$$1 + 2 + 3 + 4 = \frac{4 \times 5}{2} = 10 \text{ MA's}$$

to Compute $y[4] \rightarrow y[6]$ we need

$$3 + 2 + 1 = \frac{3 \times 4}{2} = 6 \text{ MA's}$$

$$\Rightarrow \boxed{\text{Total MA's} = 16}$$

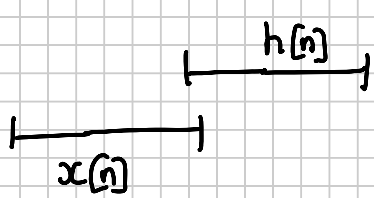
② Let $\{x[n]\}_{n=0}^{7000} \Rightarrow 7001$ Length sequence

$\{h[n]\}_{n=0}^{1100} \Rightarrow 1101$ length Sequence

$\Rightarrow y[n] = $ length 8101 : $\{y[n]\}_{n=0}^{8100}$

Convolution can be Computed in 3 Stages :

①



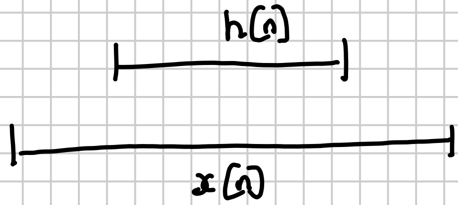
$\#$ of MA's

$= 1 + 2 + 3 + \dots + 1100$

$= \frac{1100 \times 1101}{2}$

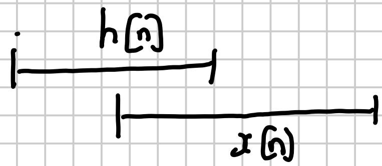
$=$

(2)



\# of MA's $= (1101)(7001-1100)$

(3)

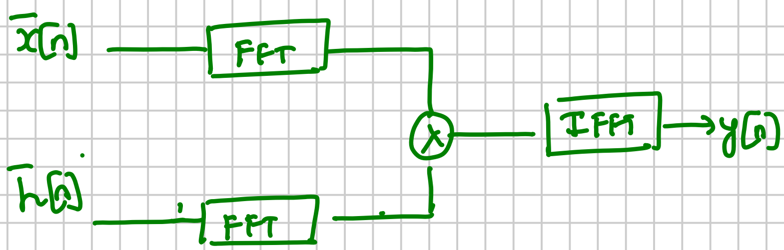


\# of MA's $= 1100 + 1099 + \dots + 1$

$= \dfrac{(1100)(1101)}{2}$

$\boxed{\text{Total Complexity} = 7{,}708{,}101}$

Now assume the above convolution is computed using DFT



$\bar{x}[n]$ and $\bar{h}[n]$ obtained by zero-padding $x[n]$ and $h[n]$

$\therefore$ $\bar{x}[n]$ and $\bar{h}[n]$ must be zero-padded to atleast length $N' = 8101 - 1 = 8100$

As discussed FFT is most efficient for $N = 2^m$

$\therefore$ we zero pad to length $\boxed{N = 8192}$

$\therefore$ Computational Complexity

$$= \underbrace{3(N \log_2 N)}_{\text{(2-FFT's, 1 IFFT)}} + \underbrace{N}_{\text{mult. in Transform domain}} = 3 \times (8192 \times 13) + 8192$$

$$= \boxed{327\,680}$$

Filtering of data Sequences :

Let $x[n] \rightarrow$ Length N' and $\boxed{N' \gg M}$

$h[n] \rightarrow$ Length M

If $N' \gg M \Rightarrow$ Long FFT's will be required.

$\Rightarrow$ long delay in Computing Convolution.

To overcome this problem the input data is segmented into smaller blocks. Each segment of data is convolved with filter $h[n]$ and the results are Combined.

Method I : Input data is segmented into blocks of length L. The size of DFT and IDFT is

$$\boxed{N = L + M - 1}$$

$\Rightarrow$ Data block and filter are zero-padded to length N. i.e. $M-1$ zeros are appended to data block and $L-1$ zeros are appended to the filter. This can be represented as follows :

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

$$x_2[n] = [x[L], x[L+1], \ldots x[2L-1], \underbrace{0, 0, \ldots 0}_{M-1 \text{ zeros}}]$$

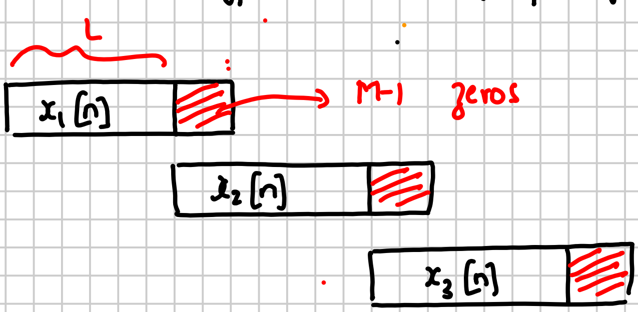
$\vdots$

$$x_3[n] = [x[2L], x[2L+1], \ldots x[3L-1], \underbrace{0, 0, \ldots 0}_{M-1 \text{ zeros}}]$$

The two N-point DFT's are multiplied together to form

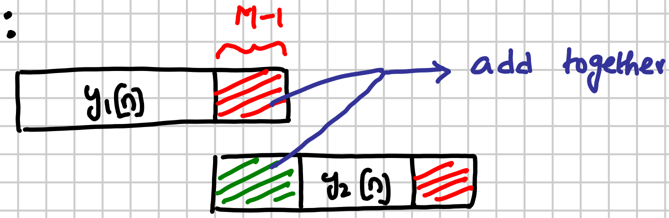
$$\boxed{Y_p[k] = H[k] X_p[k]} \qquad k = 0, 1, \ldots N-1$$

then $y_p[n] = \text{IDFT}\{Y_p[k]\}$



Since each data block is terminated with M-1 zeros, the last M-1 points from each output block must be overlapped and added to the first M-1 points of the succeeding block

Output :



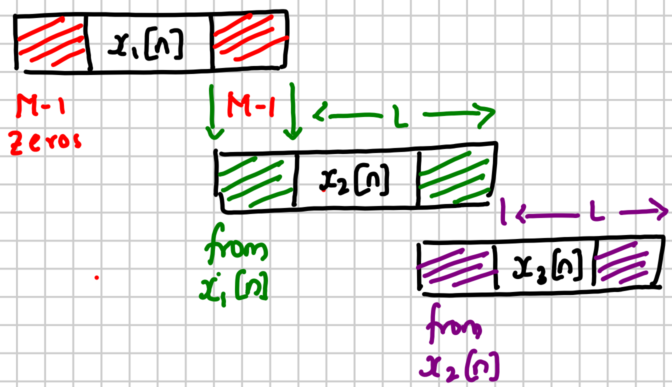
Method II : Overlap-Save method

Input data block is of length , $N = L + M - 1$

Size of DFT = N

Each data block consists of the last M-1 samples of the previous data block followed by L new samples

$x_1[n]$



N-length DFT is computed for each data block

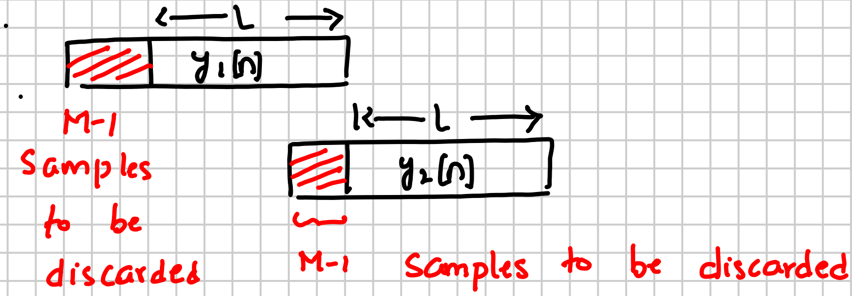
L-1 zeros are added to h[n] and N-length DFT is computed and stored. Then for the p-th data block we have,

$$\boxed{Y_p[k] = X_p(k)\, H(k)} \qquad 0 \leq k \leq N-1$$

The N-point IDFT yields,

$$y_p[n] = [\, y_p(0) \;\; y_p(1) \ldots y_p(M-1),\; y_p[M] \ldots y[N-1] \,]$$

Since the input is not zero padded the first M-1 samples are corrupted by aliasing and must be discarded. The last L samples of $y_p[n]$ are same as the results of linear convolution


L
$y_1(n)$
M-1 Samples to be discarded
L
$y_2(n)$
M-1 Samples to be discarded