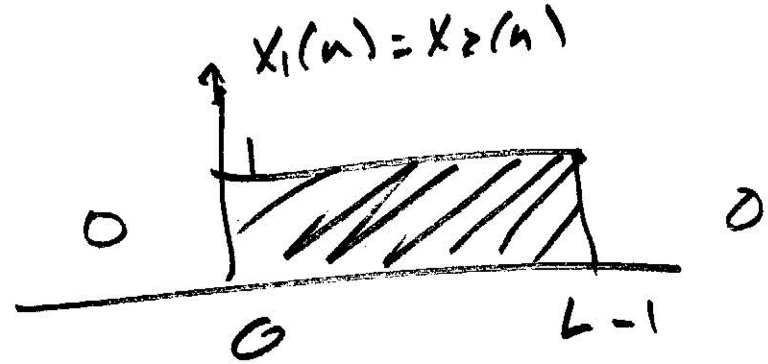
Ex Circular Convolution of 2 seq.

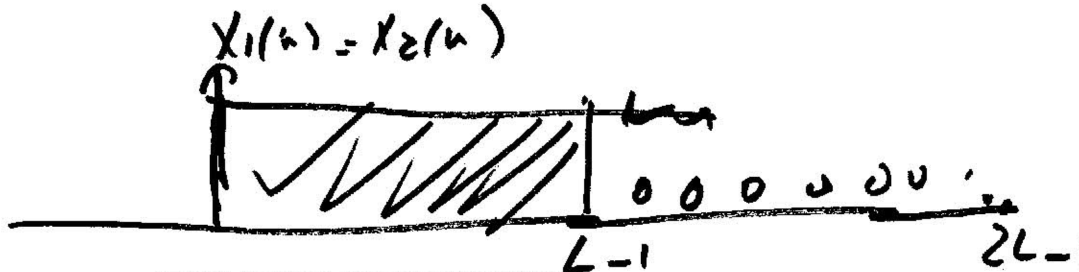
$$x_1(n) = x_2(n) = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise.} \end{cases}$$



N pt circular Convol x_1 & x_2

(1) $N = L$

(2) $N = 2L$

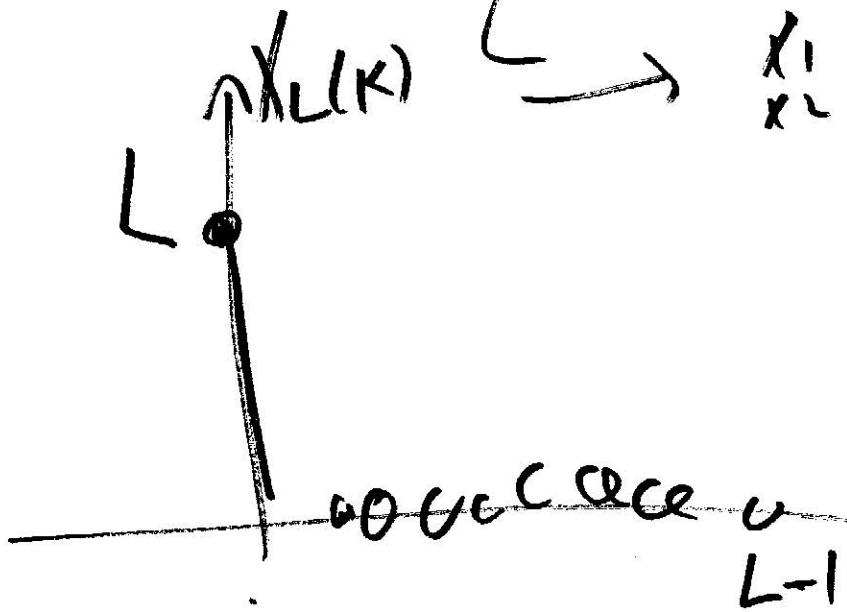


Case (1): use DFT to do circular convolut.

take N point = L point DFT of x_1

$$X_L(K) = \sum_{n=0}^{L-1} x_1(n)\, e^{-j\frac{2\pi n K}{L}}$$

$$= \begin{cases} L & k = 0 \\ 0 & \text{otherwise} \end{cases} \rightarrow x_1, x_2$$

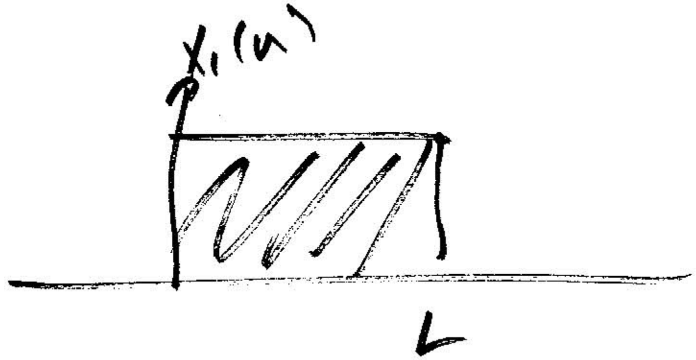


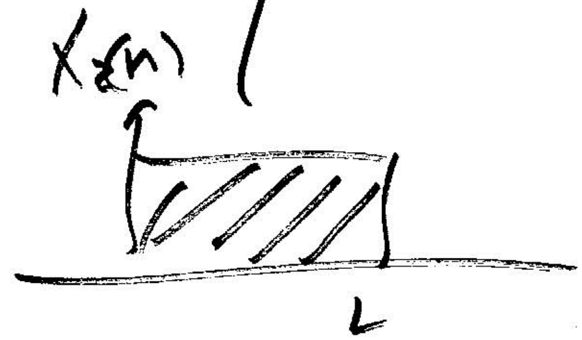
$$X_3(K) = X_2(K)\, X_L(K)$$

$$= \begin{cases} L^2 & k = 0 \\ 0 & \text{otherwise} \end{cases}$$

2

$$L \text{ pt } \mathrm{IDFT}\{X_3(k)\} = \begin{cases} L & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

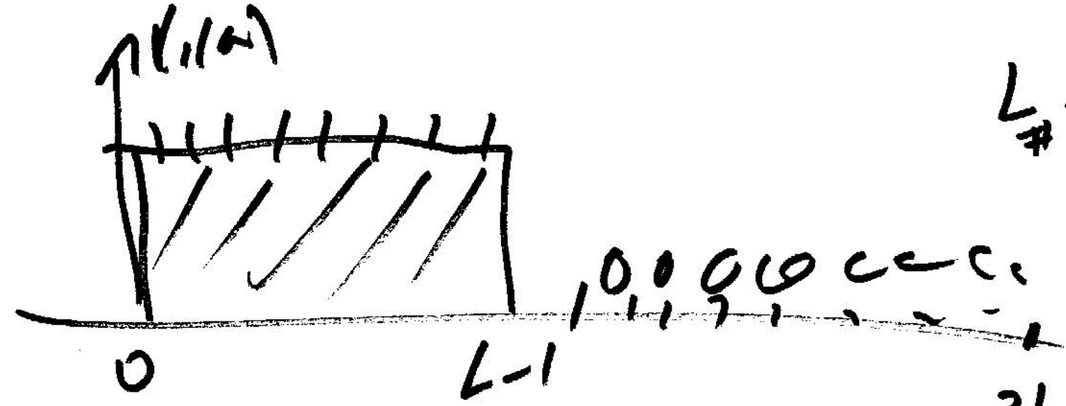


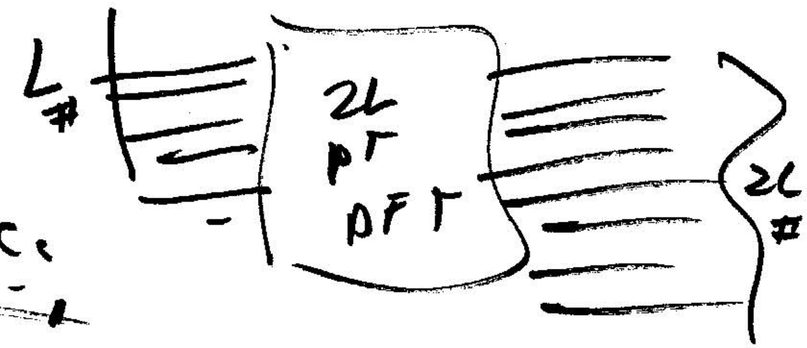


$x_3(n) = x_1 \circledast x_2(n) \longrightarrow L$ point

Case (2) $N = 2L$

$2L$ point circular convol x_1 & x_2.





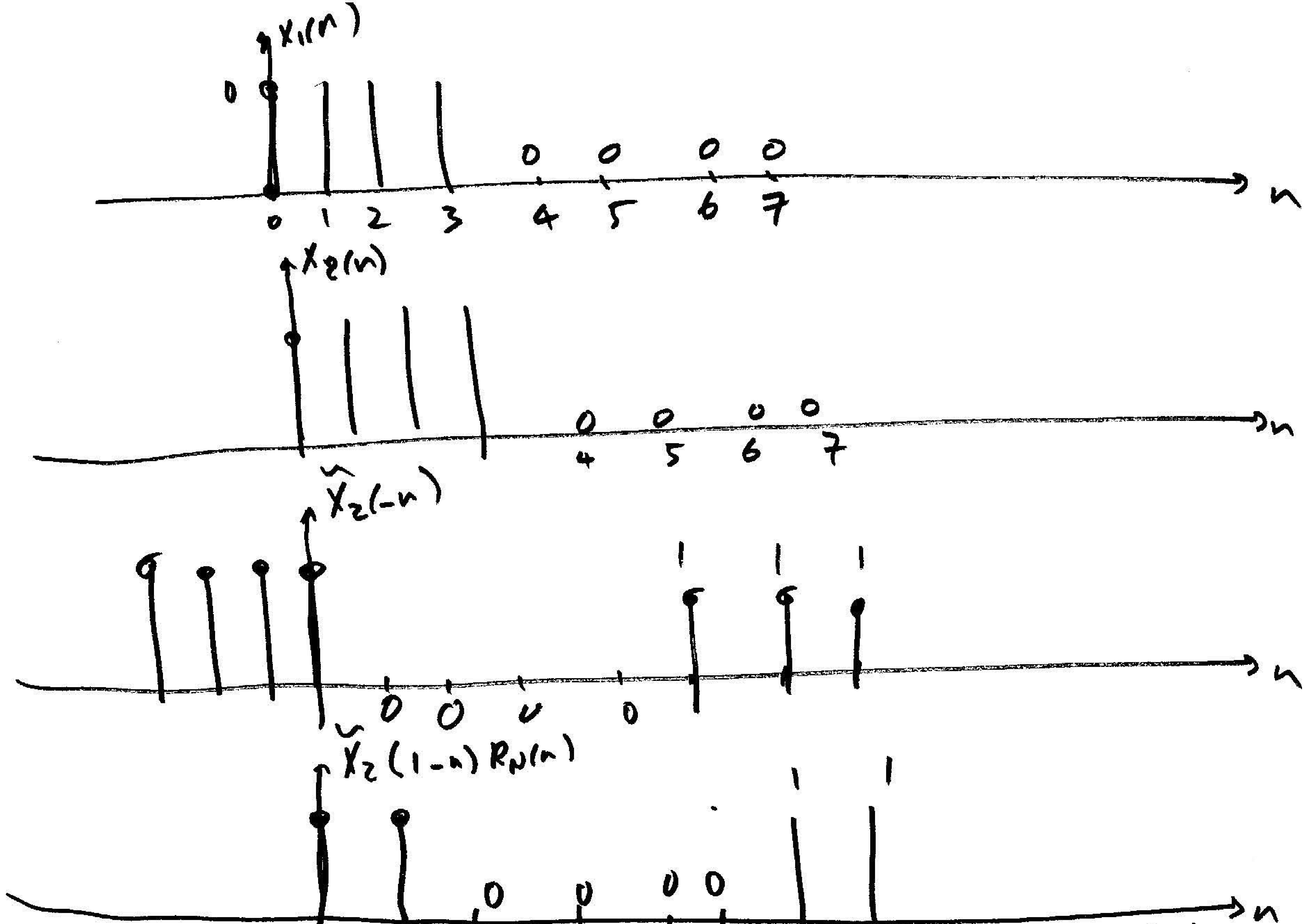
Compute $2L$ pt DFT of x_1 and x_2

$$X_{2L}(k) = \sum_{n=0}^{2L-1} x_1(n)\, e^{-j\frac{2\pi nk}{2L}}$$

$$= \sum_{n=0}^{L-1} x(n)\, e^{-j\frac{2\pi nk}{2L}}$$

$$\mathrm{IDFT}_{2L} \left\{ X_{2L}(k)\, X_{2L}(k) \right\} = x_1 \underset{2L \text{ pt.}}{\circledast} x_2$$

Pictorial circular convolut. 8.16 O&S



location 0 of output: 1×1
location 1 of output: 1×1 + 1(×1)

location 2 : 3
" 3 : 4
" 4 : 3

loc 5 : 2
loc 6 : 1
loc 7 : 0

How To use DFT To do Linear Convolution

Consider $x_1(n) \longrightarrow L$ pt

$x_2(n) \longrightarrow P$ point

$N > L \qquad N > P$

Goal $x_3(n) = x_1 * x_2 =$ linear convolution

$$x_3(n) = \sum_m x_1(m)\, x_2(n-m) \leftarrow L+P-1 \text{ pt.}$$

$$\text{DTFT}\{x_3(n)\} = X_3(\omega) = \sum_n x_3(n) e^{-j\omega n}$$

Properties of DTFT: $X_3(\omega) = X_1(\omega)\, X_2(\omega)$

DTFT of $x_1(n)$ DTFT of $x_2(n)$

Suppose sample $X_3(\omega)$ at N equally spaced points to get $Y(K)$

$$Y(K) = \left[X_3(\omega) \right]_{\omega = \frac{2\pi K}{N}}$$

$$\underset{N\,pt}{\mathrm{IDFT}} \left\{ Y(K) \right\} = y(n) = \begin{cases} \displaystyle\sum_{r=-\infty}^{+\infty} X_3(n + rN) & 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$$

thought exp from 2 lectures ago.

$$Y(k) = \left[X_1(\omega) \right]_{\omega = \frac{2\pi k}{N}} \left[X_2(\omega) \right]_{\omega = \frac{2\pi k}{N}}$$

$x_1 \rightarrow L$ pt $x_2 \rightarrow P$ point $N > L$, $N > P$

$$Y(k) = X_1(k) \quad X_2(k)$$

N pt DFT of $x_1(n)$ N pt DFT of $x_2(n)$

Apply Circular Convolution.

$$y(n) = x_1 \circledast x_2$$

N pt circular pt.

Thought Exp

100pt

N00pt DFT

100 pts

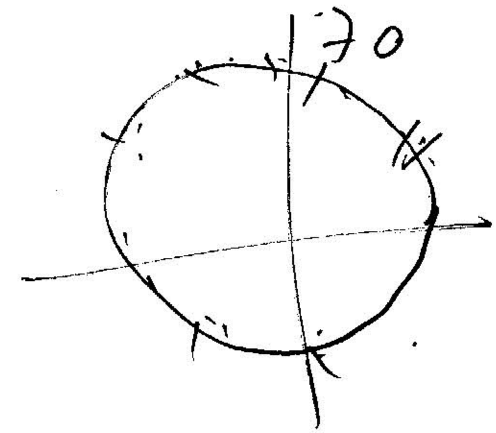
x(n) 100pt → DTFT → $\left[X(\omega)\right]_{\omega = \frac{2\pi k}{70}}$

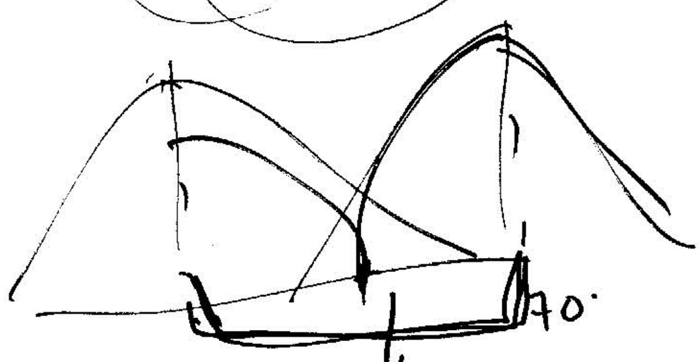
↓

70pt

IDFT

70 pts





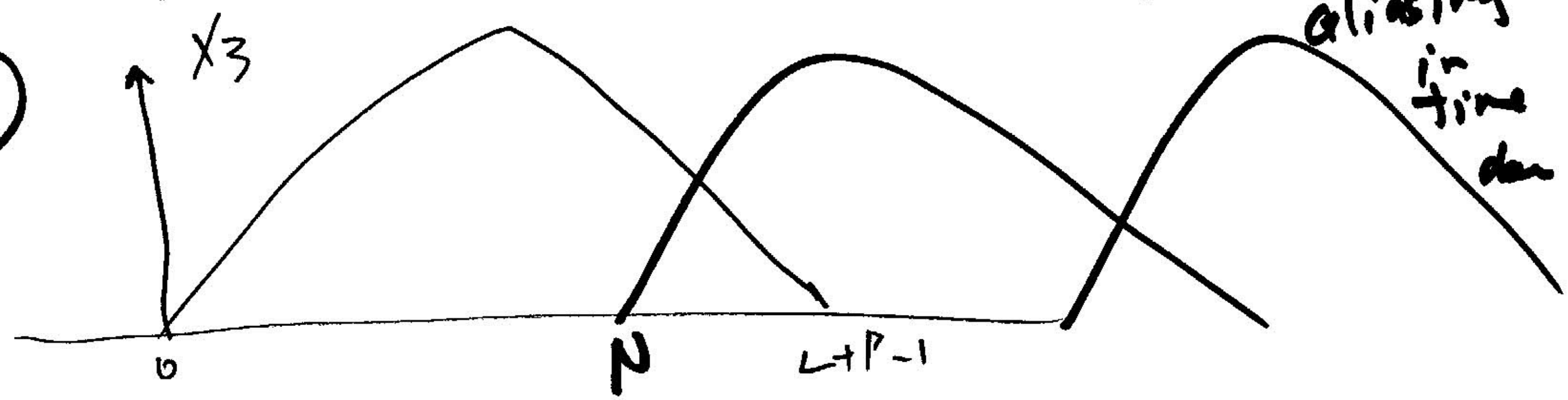
$$y(n) = \boxed{x_1 \circledast x_2 = \begin{cases} \sum_{r=-\infty}^{+\infty} x_3(n+rN) & 0 \le n < N \\ 0 & \text{otherwise.} \end{cases}}$$

<u>Conclusion</u> N pt circular convolution of x_1 and x_2 is the same as their linear convolution of x_1 and x_2, periodicized with period N, and take one period.

$x_3 \longrightarrow L+P-1$ pt sequence.

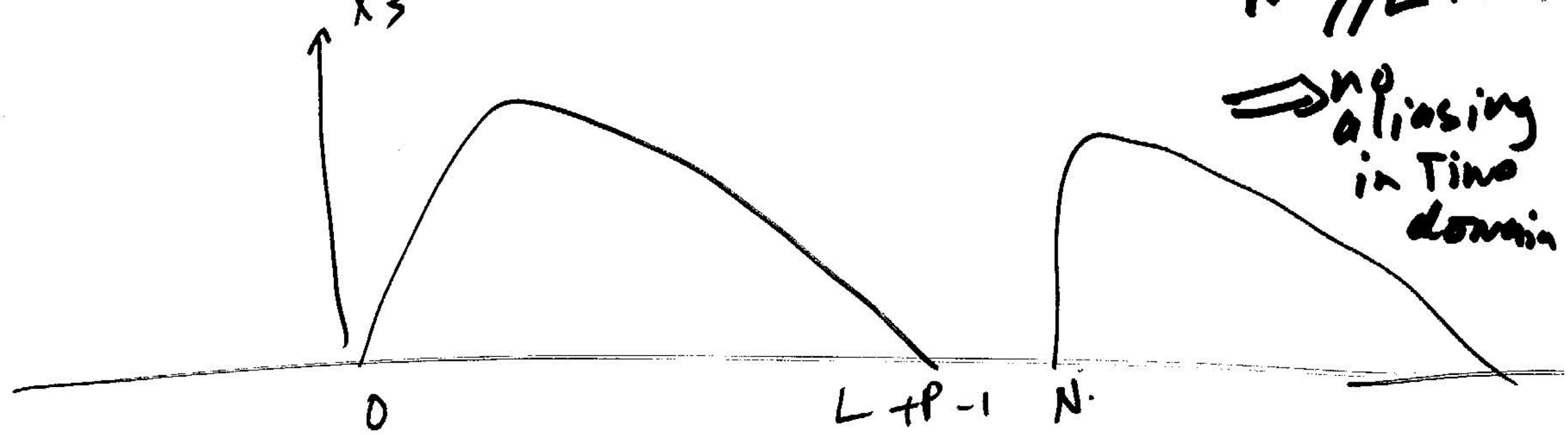
$N < L+P-1$ aliasing in time dom

Case ①



$N \geq L+P-1$

$\Rightarrow$ no aliasing in Time domain



Conclusion: If $N \geq L+P-1$ $\Rightarrow x_3 = x_1 \circledast x_2$ (N pt.)

$\Rightarrow$ no aliasing $\Rightarrow$

N pt circular conv of x_1 and x_2 results in linear convolution of x_1 and x_2

$\Rightarrow$ Can use Product of DFTs To do Linear Convolution $\Rightarrow$ DFT can be used for LTI processing

What is The process for Computing Linear Conv. using DFT

$x_1 \rightarrow L$ pt
$x_2 \rightarrow P$ pt.

$\Big\} \longrightarrow x_3 = x_1 * x_2 \quad L+P-1$

$N \geq L+P-1$

① Pad x_1 with $P-1$ or more zeros to get N pt seq.

② Pad x_2 with $L-1$ or more zero to get N pt seq.

③ take N pt DFT of padded x_1 and padded x_2

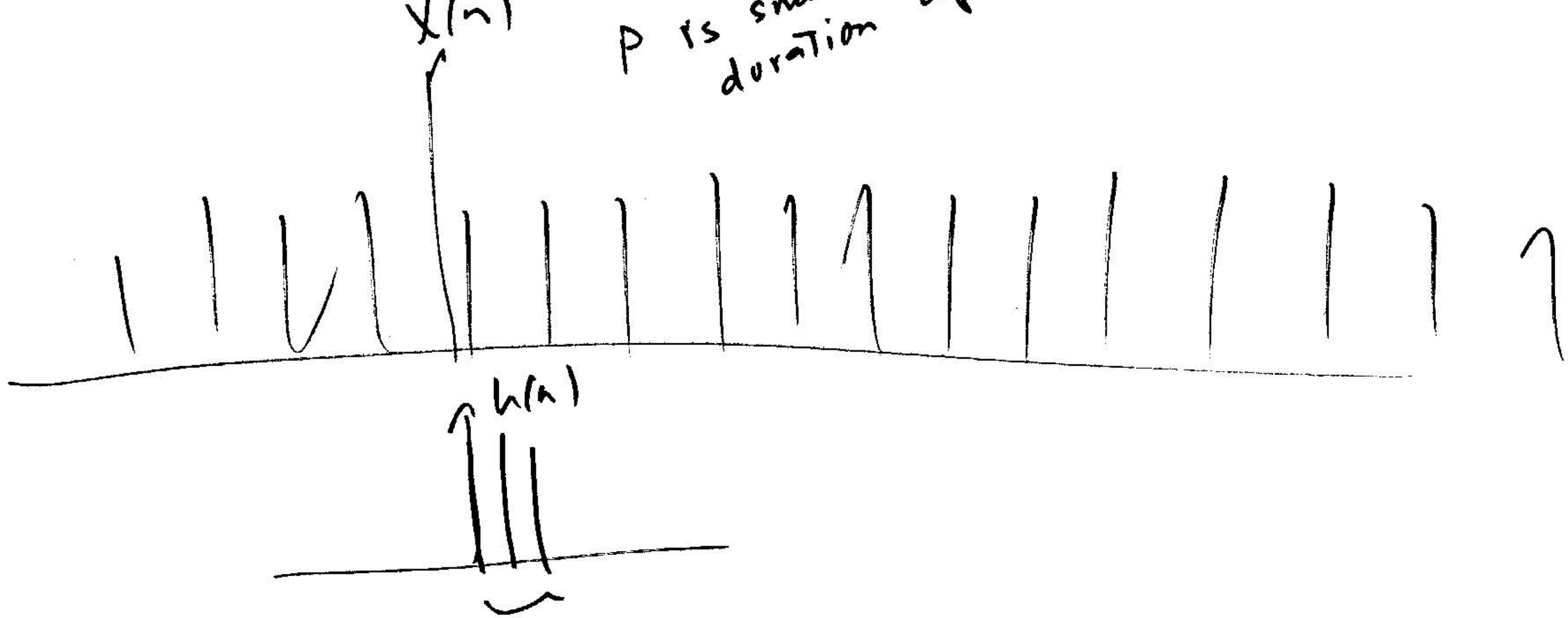
④ Take ~~Inverse N pt DFT~~ Multiply The 2 N pt DFTs in step 3

⑤ take IDFT ⟸ N pt of The DFT is step 4

$\Longrightarrow x_3$ is linear convolution of x_1 and x_2

Using DFT for LTI filtering of only long sequence

FIR filter.
P taps, P coeff.
P is small compared to duration of input x



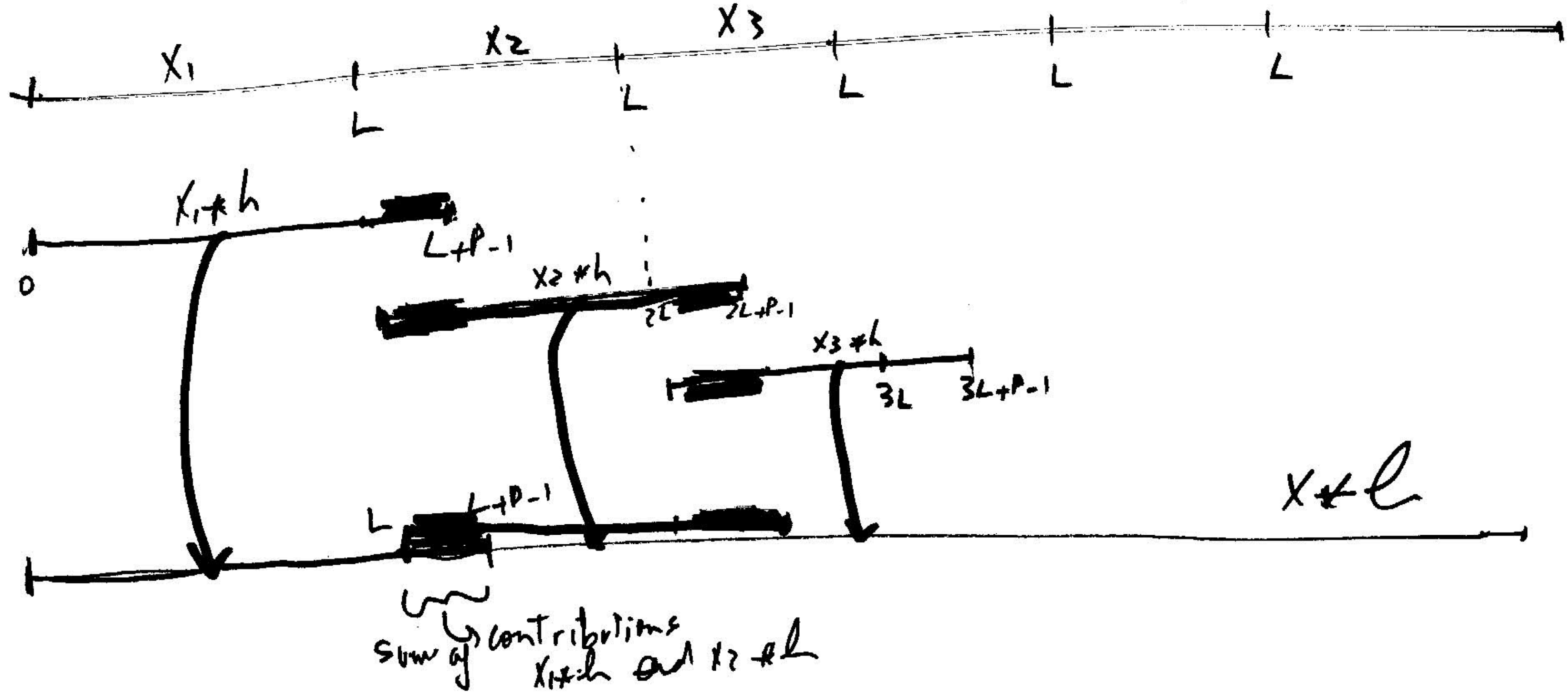
2 Methods : 1 Overlap add
2 Overlap Save.

Overlap Add :

Take advantage of linearity prop of convolut.

$[x_1(n) + x_2(n)] * h = (x_1 * h) + (x_2 * h)$

$h(n) \to P$ pt.



Steps to do overlap add:

(1) segment only long sequence int L nonoverlapping chunks.

(2) Convolve each L pt chunk with h to get (L + P - 1) new points.

(3) add up (while lining up properly) the resulting convolution of $x_i * h$ to get $x * h$

See Fig 8.23 in O&S.