Implementing LTI-System using DFT

$x[n]$: L-point

$h[n]$: P-point

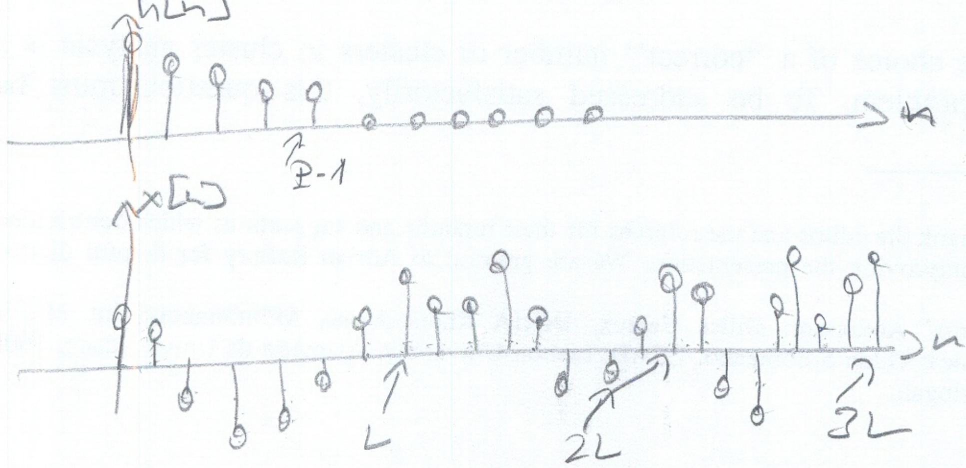
$$y[n] = h[n] * x[n] = \sum_i h[i]x[n-i]$$

$\geq$ L+P-1 -point => DFT's $X[k]$, $H[k]$ must have L+P-1 -points

x and h need to be filled with zero's until length = L+P-1 ("zero-padding")

- In many applications, such as in filtering ~~of~~ a speech waveform, input signal is of indefinite duration. Long sequences are not practical: ~~a~~ DFT computations or convolution too expensive!

<u>example:</u> finite-length impulse response and indefinite-length signal $x[n]$



-> When we take all input samples, no filtered samples can be computed until input ends.

Would like to avoid such a large delay in processing.

Solution : "block convolution"

- signal is segmented into sections of length L.
- each section can be convolved with the finite-length impulse response and the filtered sections fitted together.
- filtering of each block can be done by DFT

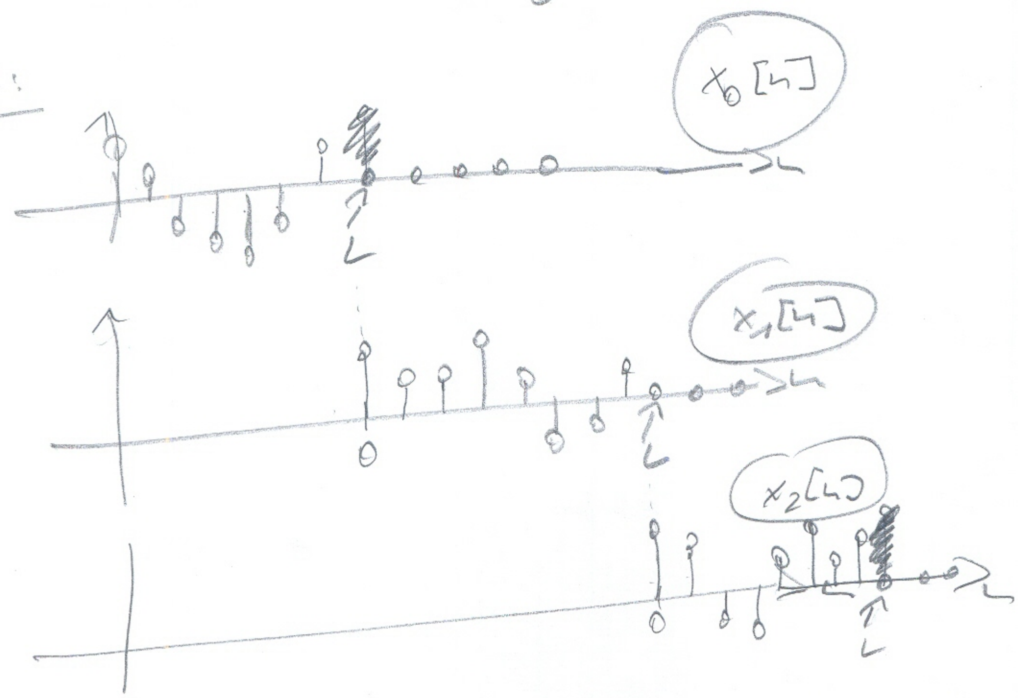
- consider $x[n] = 0$ for $n < 0$

$$x[n] = \sum_{r=0}^{\infty} x_r[n - rL]$$

"sum of shifted finite-length segments of length L"

where $x_r[n] = \begin{cases} x[n+rL] & , 0 \le n \le L-1 \\ 0 & , \text{o.w.} \end{cases}$

plot:



→ each block has convolution result of $L+P-1$ elements.

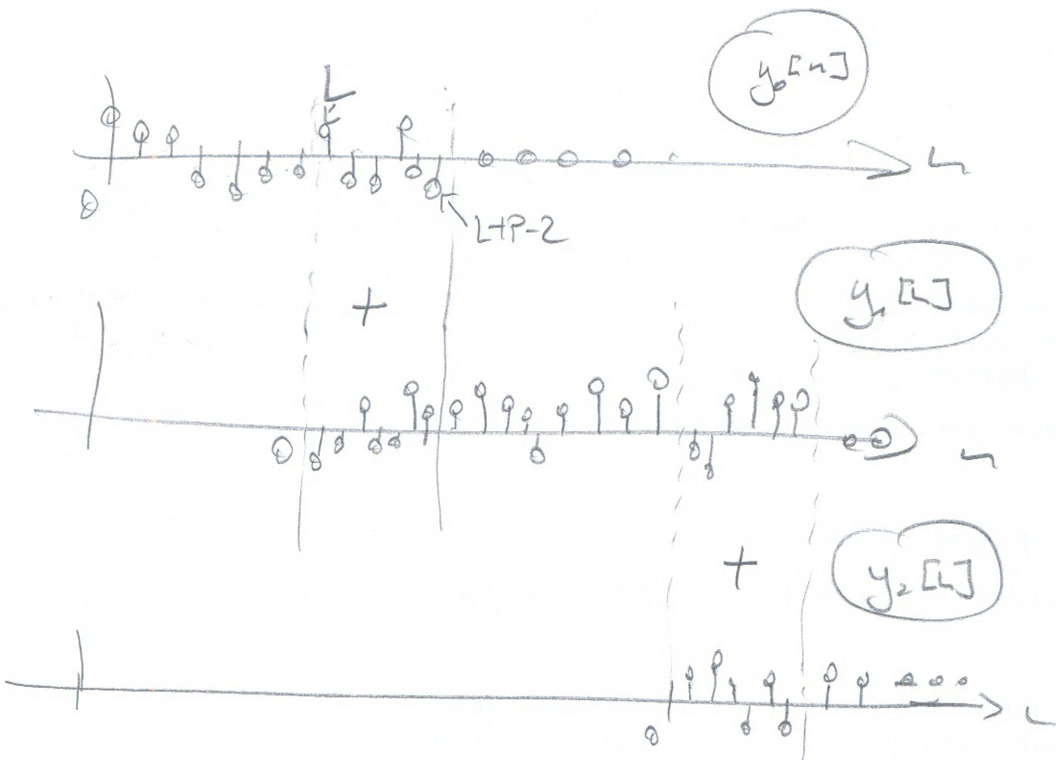
→ can use $N \ge L+P-1$ -point DFT

- each block has length L but resulting convolution of each block has length $L+P-1$

-> overlapping of $L+P-1-L = P-1$ elements

-> simply add the overlapping parts to compute the overall result $y[n]$:

-> $\boxed{\text{overlap-and-add method}}$



<u>other methods</u>

-> Overlap-and-save method:

- L-point circular convolution of a P-point impulse response $h[n]$ with an L-point segment $x_r[n]$
- identify the part of the circular convolution that corresponds to a linear convol.