EE123
Digital Signal Processing

Lecture 6
Properties of DFT

some of the material was based on slides by J.M. Kahn

M. Lustig, EECS UC Berkeley

Announcements

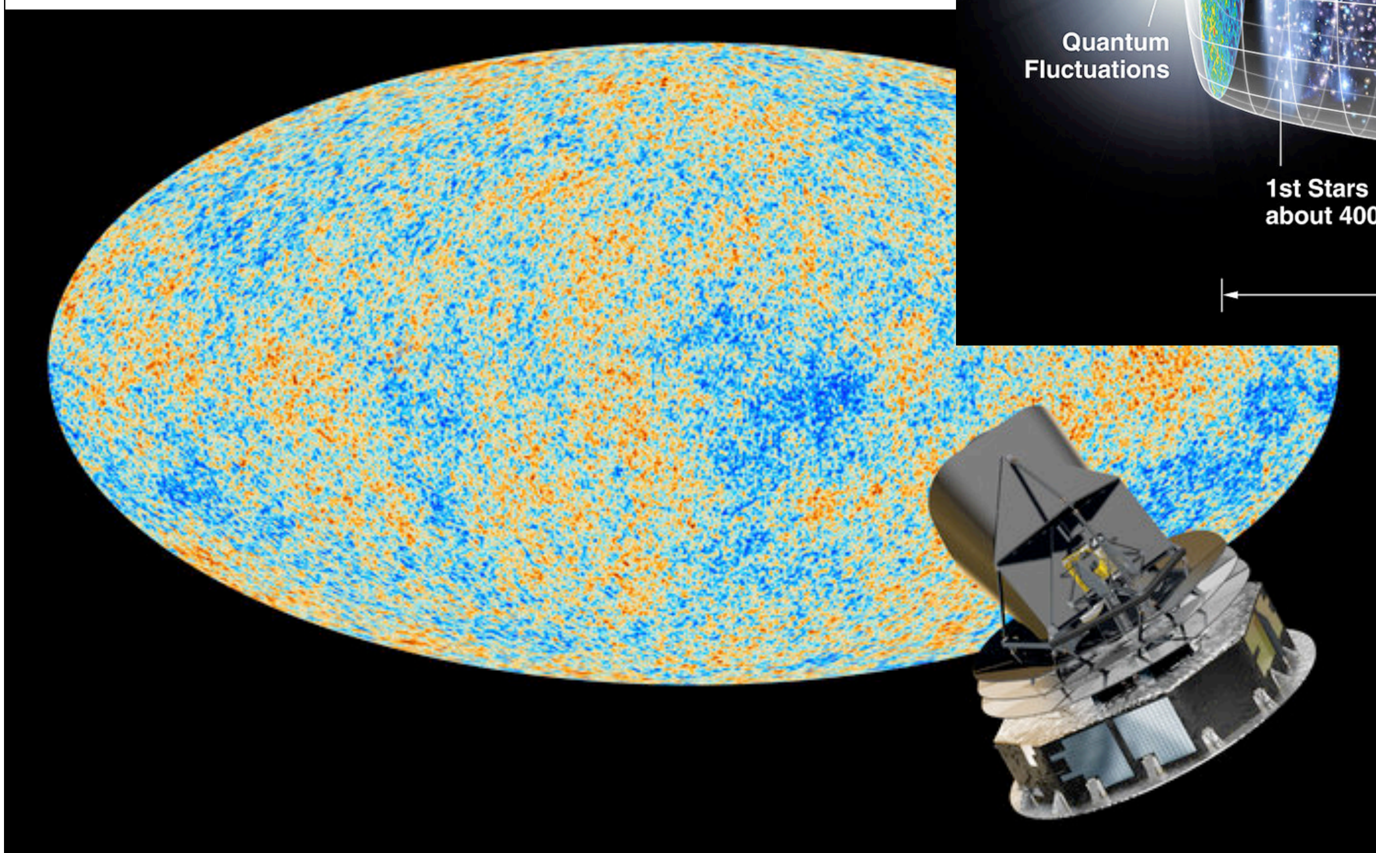
- HW1 solutions posted -- self grading due Tue
- HW2 + due Friday,
- Homework Slip policy (lowest grade homework will be dropped)

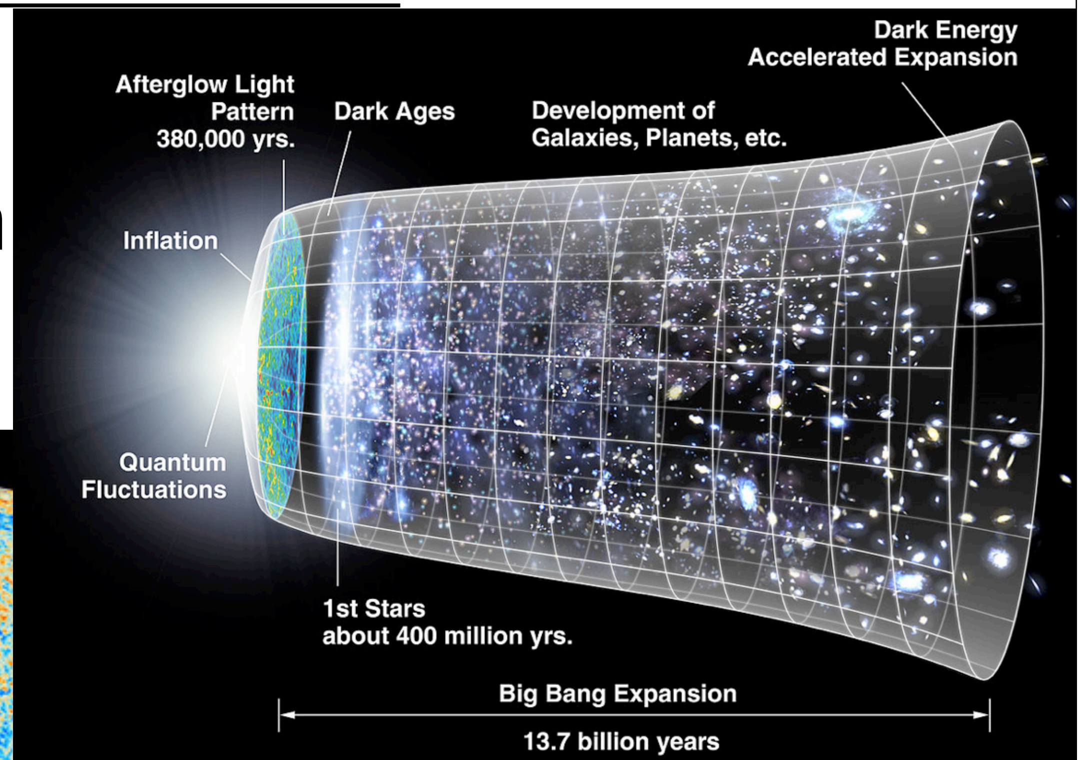
- SDR giveaway Thursday in lab
- Finish reading Ch. 8, start Ch. 9

- ham radio licensing lecture II W 6:30-8pm Cory 521

Cool things DSP

- Cosmic Microwave Background radiation





Last Time

- Discrete Fourier Transform
  - Similar to DFS
  - Sampling of the DTFT (subtlties....more later)
  - Properties of the DFT

- Today
  - Linear convolution with DFT
  - Overlap-Add / Save method for fast convolutions

Circular Convolution Sum

- Circular Convolution:

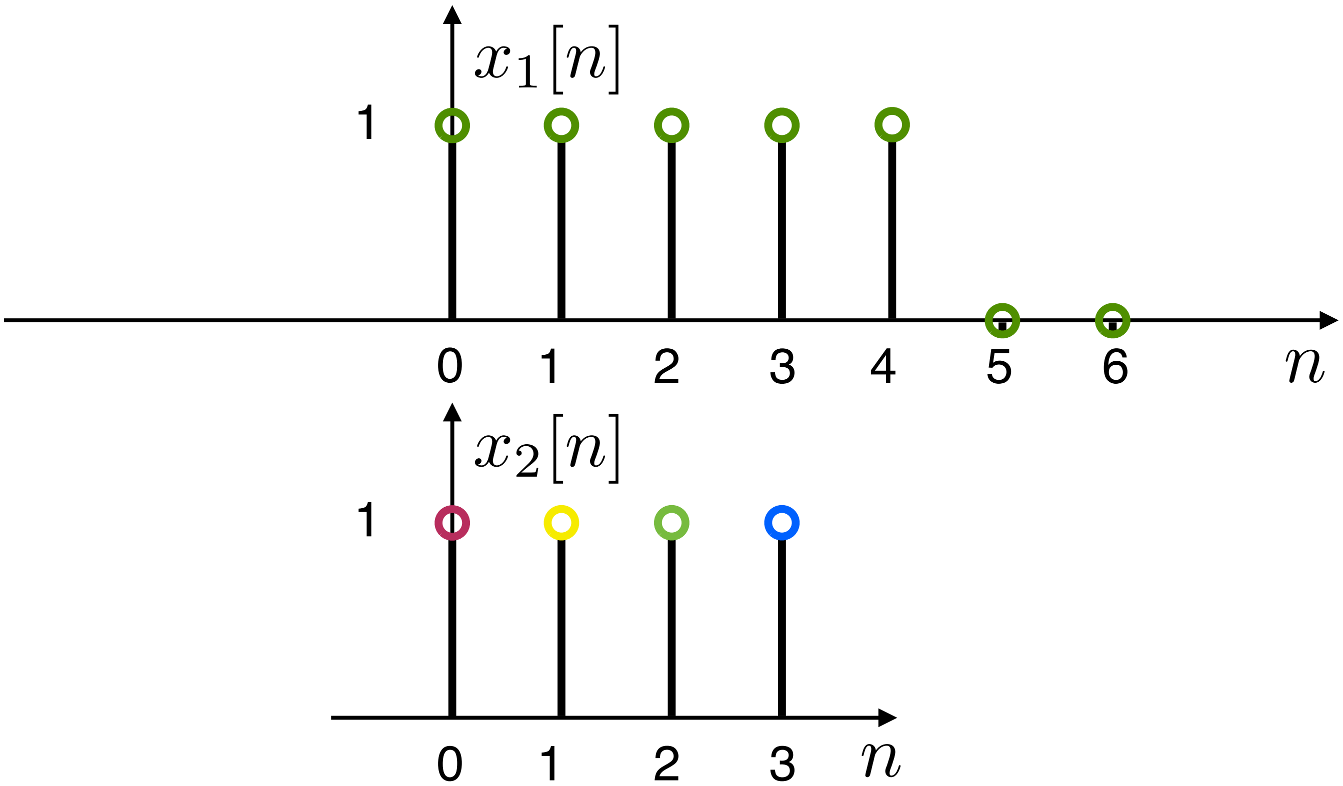
$$x_1[n] \circledast_N x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

for two signals of length N

- Note: Circular convolution is commutative

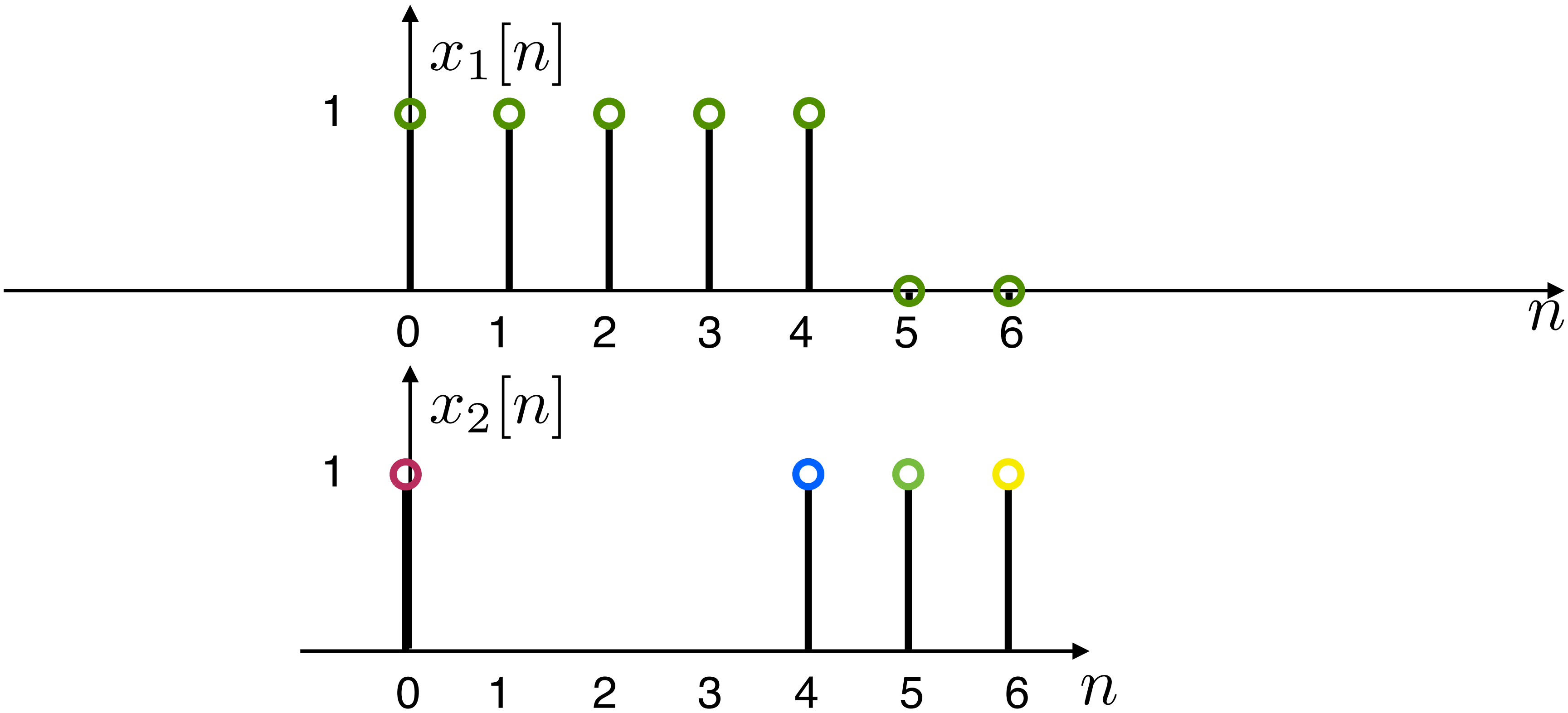
$$x_2[n] \circledast_N x_1[n] = x_1[n] \circledast_N x_2[n]$$

Compute Circular Convolution Sum



$$y[n] = x_1[n] \; ⑦ \; x_2[n] = ?$$

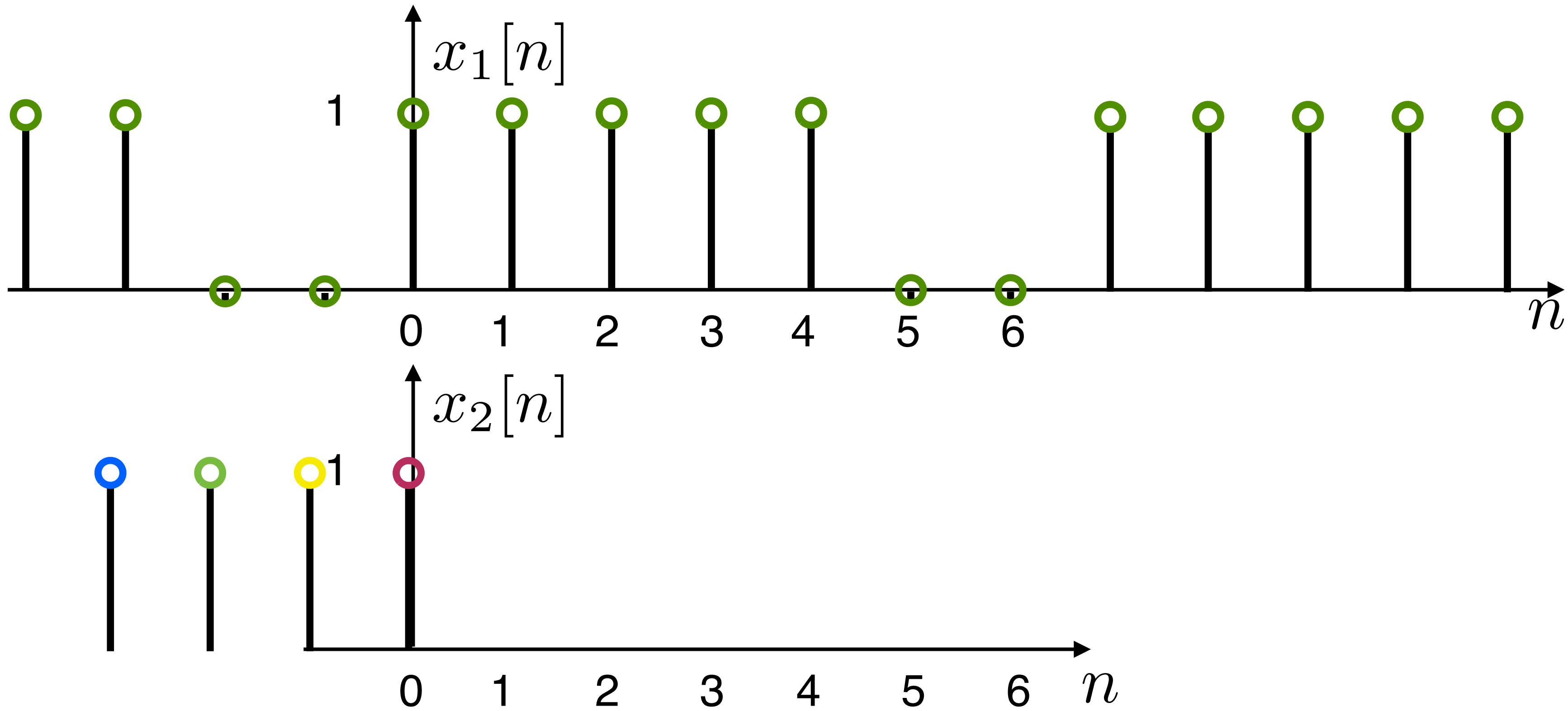
Compute Circular Convolution Sum



Circular 'flip'
multiply and add
Here: y[0]

$$y[n] = x_1[n] \; ⑦ \; x_2[n] = ?$$

Compute Circular Convolution Sum

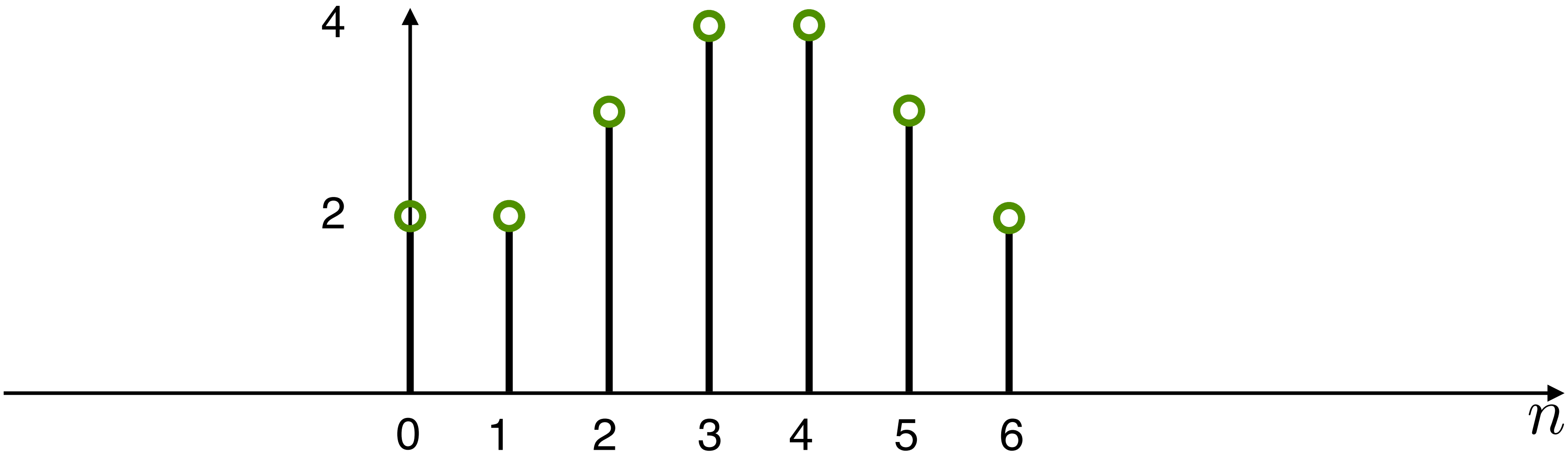


Equivalent periodic convolution over a period

$$y[n] = x_1[n] \; ⑦ \; x_2[n] = ?$$

Result

$$y[n] = x_1[n] \;\; ⑦ \;\; x_2[n] = ?$$



Properties of DFT

- **Circular Convolution:** Let x1[n], x2[n] be length N

$$x_1[n] \circledN x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

Very useful!!! (for linear convolutions with DFT)

- **Multiplication:** Let x1[n], x2[n] be length N

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \circledN X_2[k]$$

Linear Convolution

- Next....
  - Using DFT, circular convolution is easy
  - But, **linear** convolution is useful, not circular
  - So, show how to perform linear convolution with circular convolution
  - Used DFT to do linear convolution

Linear Convolution

- We start with two non-periodic sequences:

$$x[n] \quad 0 \leq n \leq L-1$$
$$h[n] \quad 0 \leq n \leq P-1$$

for example x[n] is a signal and h[n] an impulse response of a filter

- We want to compute the linear convolution:

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

y[n] is nonzero for $0 \leq n \leq L+P-2$ with length **M=L+P-1**

- Requires L·P multiplications

Linear Convolution via Circular Convolution

- Zero-pad x[n] by P-1 zeros

$$x_{\mathrm{zp}}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$$

- Zero-pad h[n] by L-1 zeros

$$h_{\mathrm{zp}}[n] = \begin{cases} h[n] & 0 \le n \le P-1 \\ 0 & P \le n \le L+P-2 \end{cases}$$

- Now, both sequences are of length M=L+P-1

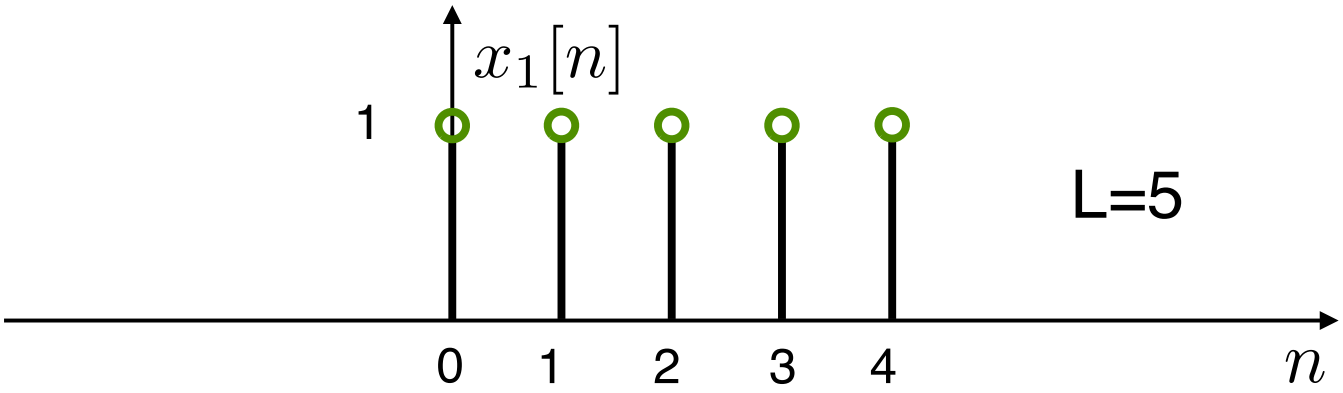
Linear Convolution via Circular Convolution

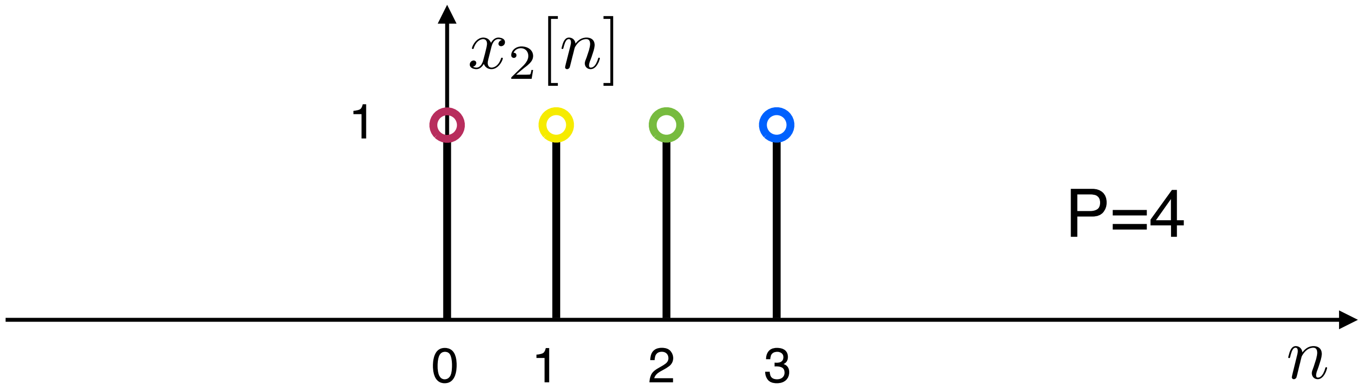
- Now, both sequences are of length M=L+P-1
- We can now compute the linear convolution using a circular one with length M = L+P-1

Linear convolution via circular

$$y[n] = x[n] * y[n] = \begin{cases} x_{\text{zp}}[n] \circledM h_{\text{zp}}[n] & 0 \leq n \leq M-1 \\ 0 & \text{otherwise} \end{cases}$$

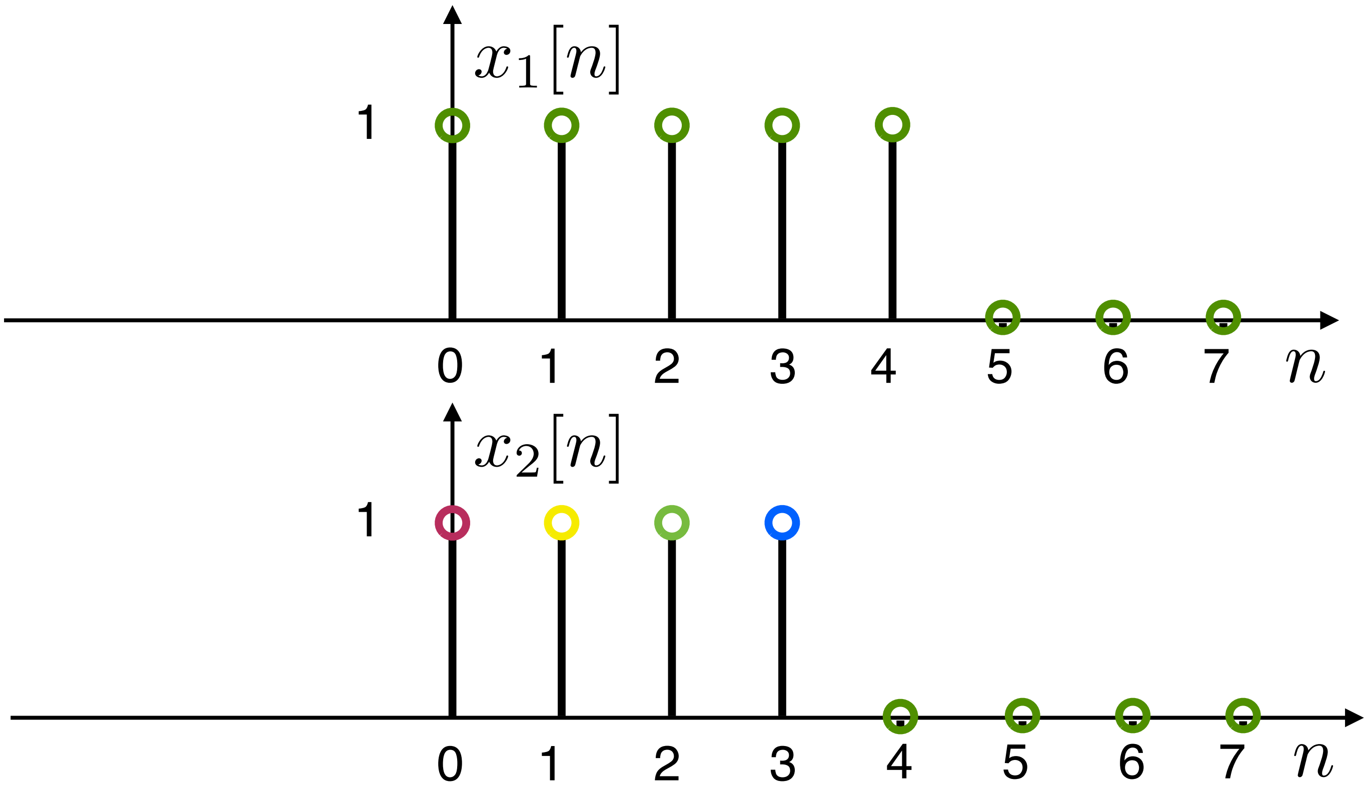
Example





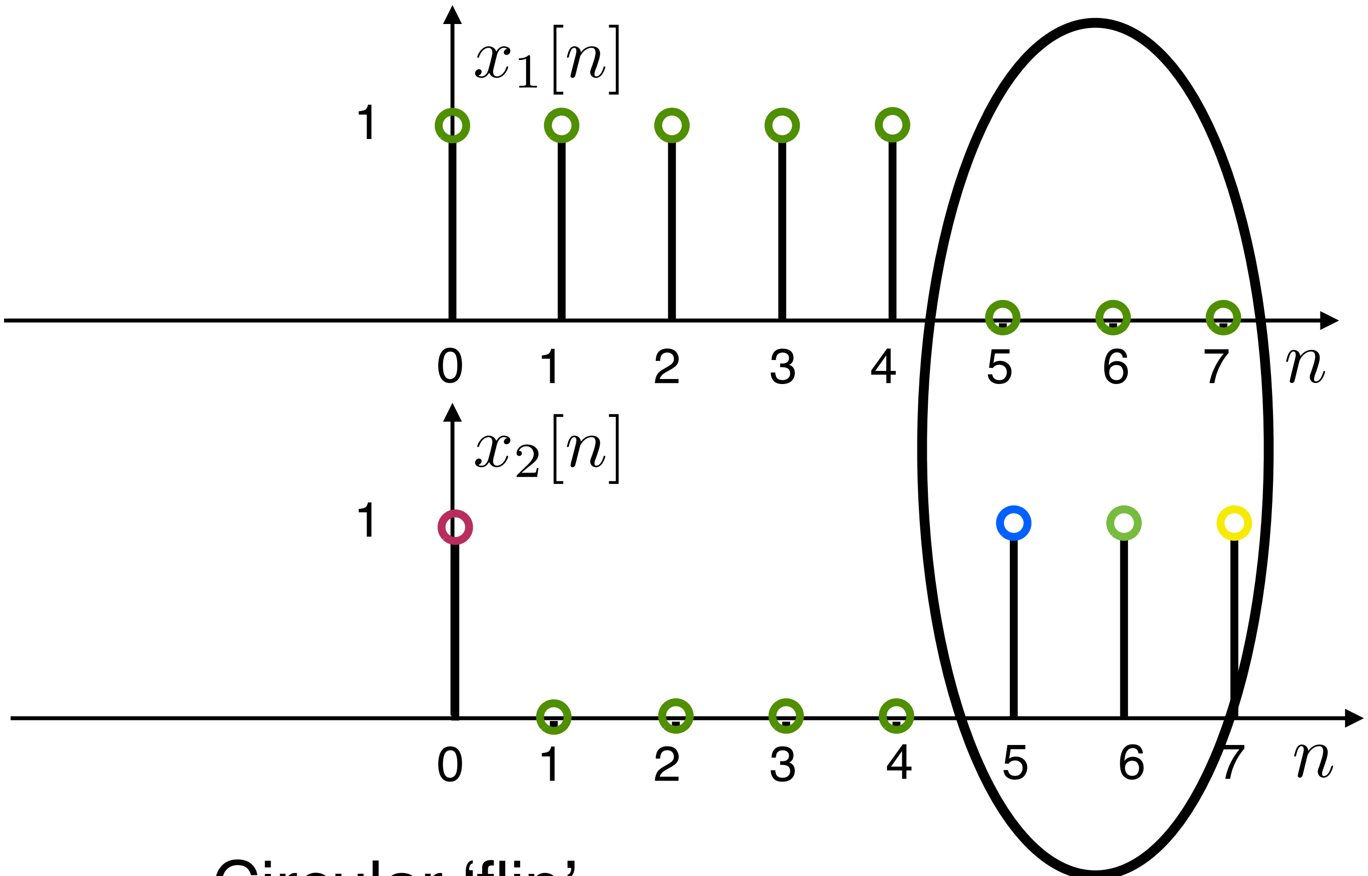
M = L + P - 1 = 8

Example



M = L + P - 1 = 8

Example



Circular 'flip'

M = L + P - 1 = 8

$$y[n] = x_1[n] \circledast_8 x_2[n] = x_1[n] * x_2[n]$$

Linear Convolution using DFT

- In practice we can implement a circulant convolution using the DFT property:

$$\begin{aligned} x[n] * h[n] &= x_{\text{zp}}[n] \circledM h_{\text{zp}}[n] \\ &= \mathcal{DFT}^{-1}\left\{\mathcal{DFT}\left\{x_{\text{zp}}[n]\right\} \cdot \mathcal{DFT}\left\{h_{\text{zp}}[n]\right\}\right\} \end{aligned}$$

for $0 \leq n \leq M-1$, $M=L+P-1$

- Advantage: DFT can be computed with $N\log_2 N$ complexity (FFT algorithm later!)
- Drawback: Must wait for all the samples -- huge delay -- incompatible with real-

Block Convolution

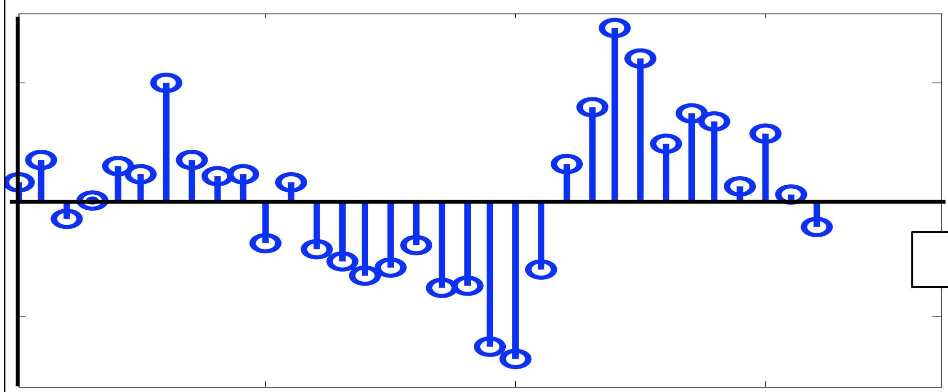
- Problem:
  - An input signal x[n], has very long length (could be considered infinite)
  - An impulse response h[n] has length P
  - We want to take advantage of DFT/FFT and compute convolutions in blocks that are shorter than the signal
- Approach:
  - Break the signal into small blocks
  - Compute convolutions
  - Combine the results

Block Convolution

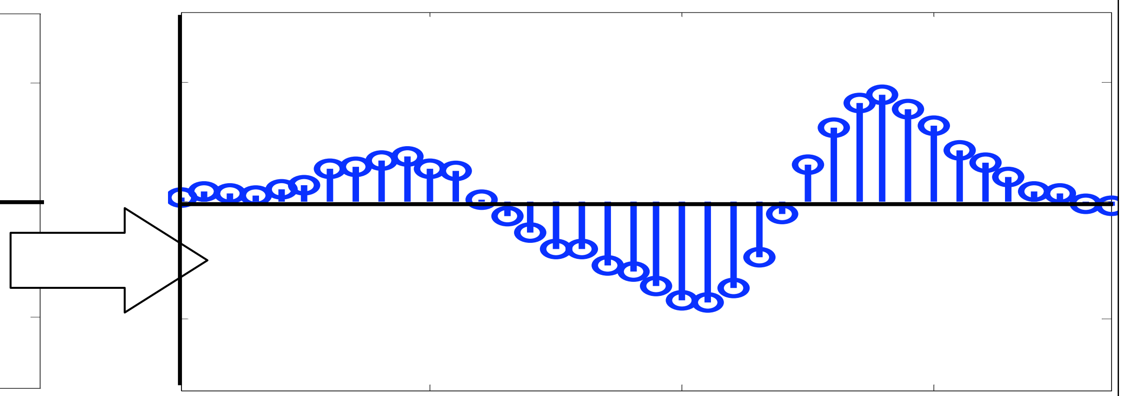
Example:

h[n] Impulse response, Length P=6

x[n] Input Signal, Length P=33



y[n] Output Signal, Length P=38



Overlap-Add Method

- Decompose into non-overlapping segments

$$x_r[n] = \begin{cases} x[n] & rL \le n < (r+1)L \\ 0 & \text{otherwise} \end{cases}$$

- The input signal is the sum of segments

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

Overlap-Add Method

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

- The output is:

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} x_r[n] * h[n]$$

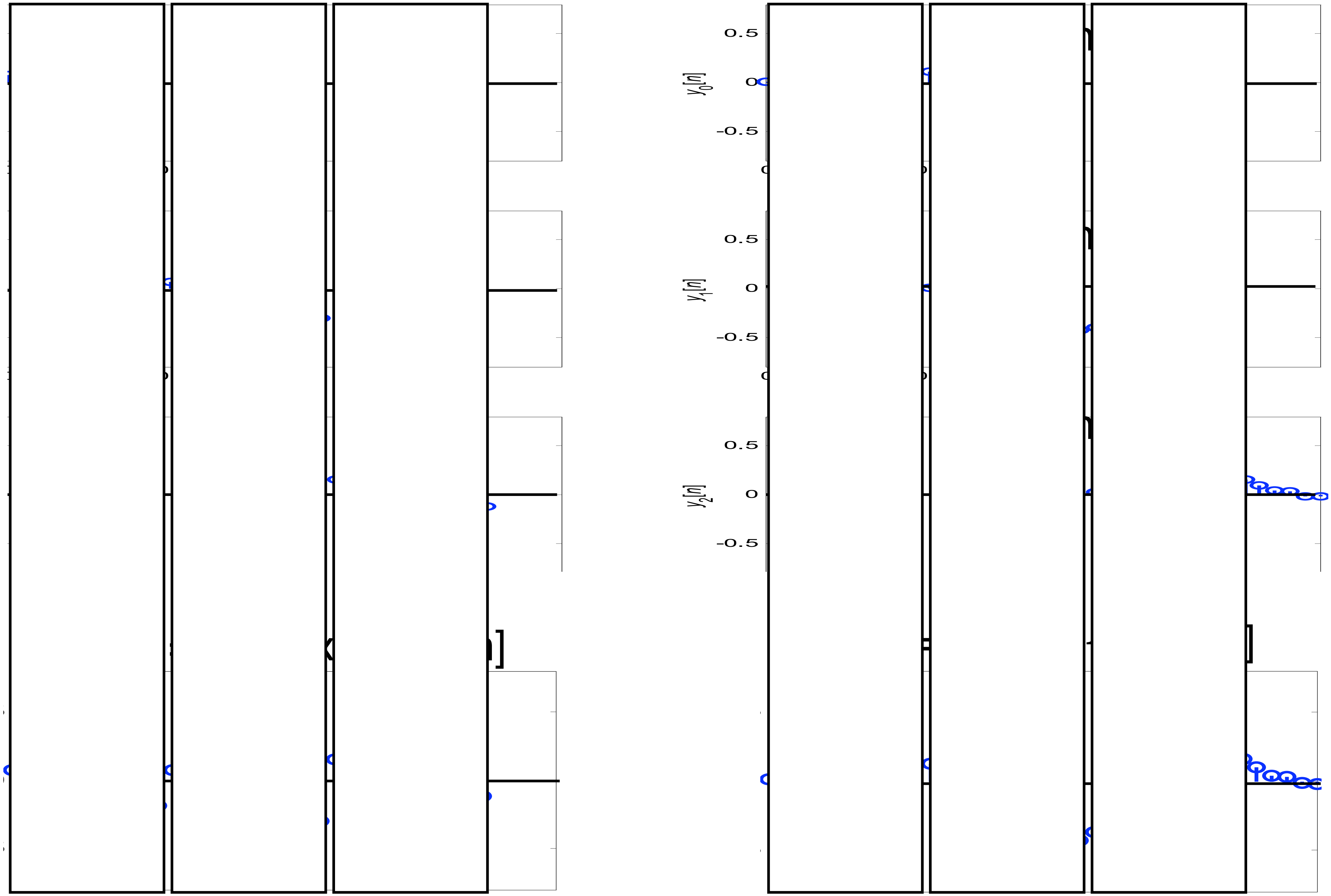
- Each output segment $x_r[n]*h[n]$ is length N=L+P-1

- We can compute $x_r[n] * h[n]$ using linear convolution
- Using the DFT:
  - Zero-pad $x_r[n]$ to length N
  - Zero-pad $h[n]$ to length N and compute $\mathrm{DFT}_N\{h_{zp}[n]\}$ (only once) WHY?
  - Compute

$$x_r[n] * h[n] = \mathrm{DFT}^{-1}\left\{\mathrm{DFT}\{x_{r,zp}[n]\} \cdot \mathrm{DFT}\{h_{zp}[n]\}\right\}$$

- Neighboring outputs overlap by P-1
  - Add overlaps to get final sequence

Example of overlap and add:



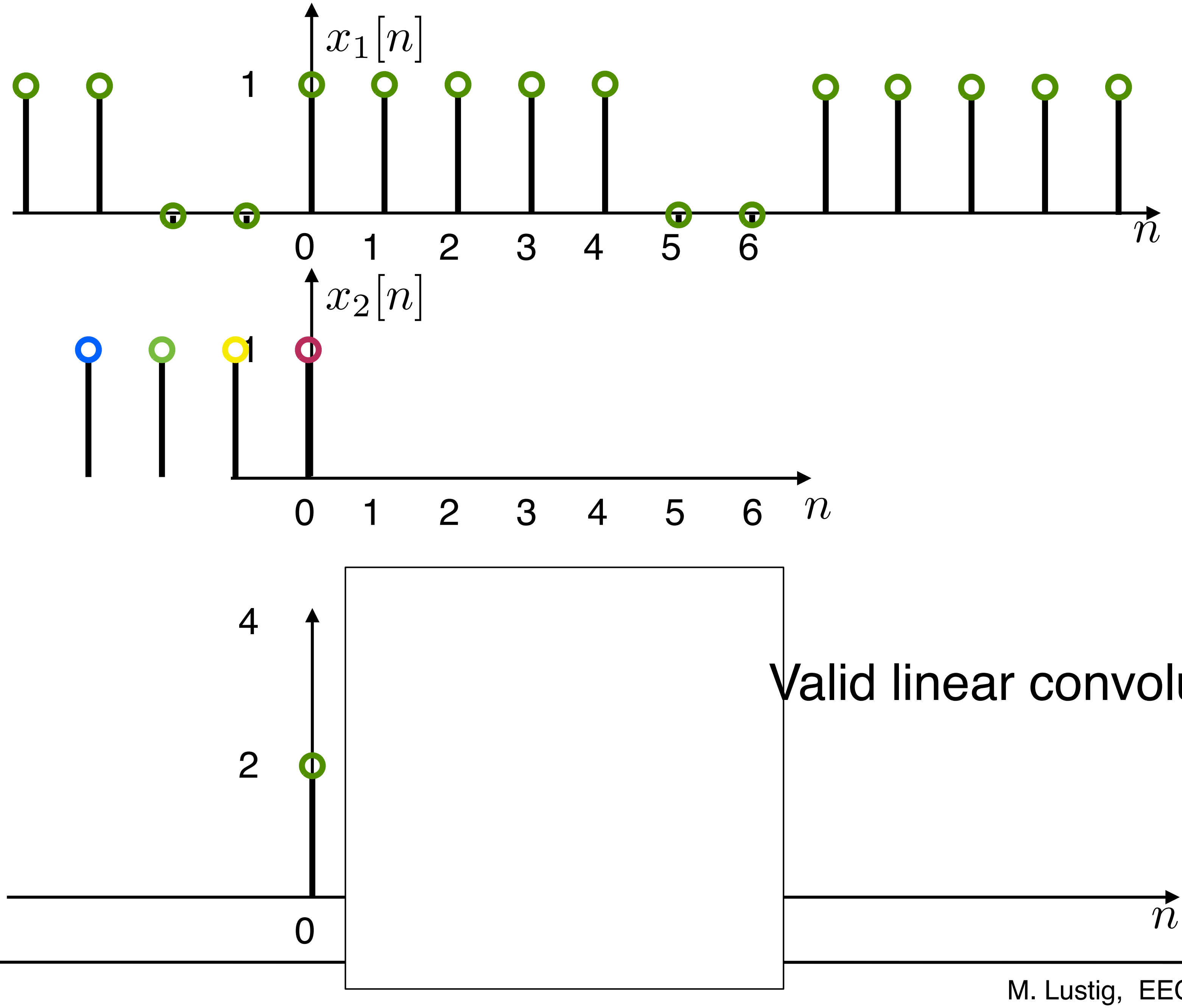
Overlap-Save Method

- Basic idea:
- Split input into (P-1) overlapping segments with length L+P-1

$$x_r[n] = \begin{cases} x[n] & rL \leq n < (r+1)L + P \\ 0 & \text{otherwise} \end{cases}$$

- Perform circular convolution in each segment, and keep the L sample portion which is a valid linear convolution

Recall:



Valid linear convolution!

Example of overlap and save: