ESE 531: Digital Signal Processing

Lecture 18: March 24, 2022

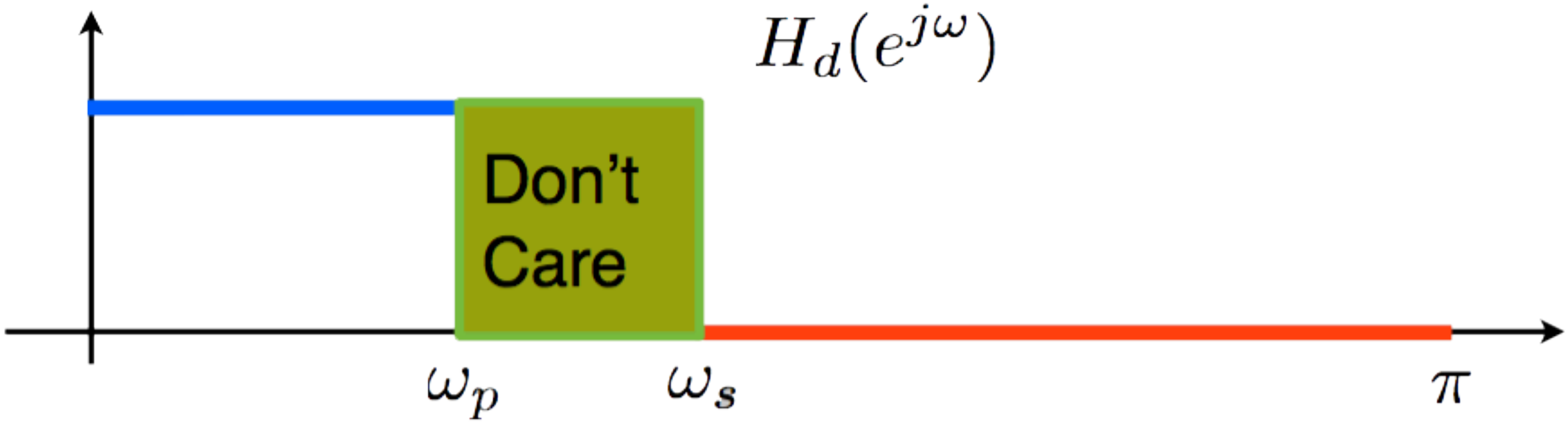
Discrete Fourier Transform

Penn ESE 531 Spring 2022 – Khanna
Adapted from M. Lustig, EECS Berkeley

Today

- Optimal Filter Design
- Discrete Fourier Series
- Discrete Fourier Transform (DFT)
- DFT Properties
- Circular Convolution

Optimality – Least Squares



- Least Squares:

$$\text{minimize} \quad \int_{\omega \in \text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

- Variation: Weighted Least Squares:

$$\text{minimize} \quad \int_{-\pi}^{\pi} W(\omega)|H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

Design Through Optimization

- Idea: Sample/discretize the frequency response

$$H(e^{j\omega}) \Rightarrow H(e^{j\omega_k})$$

- Sample points are fixed $\omega_k = k\frac{\pi}{P}$

$$-\pi \leq \omega_1 < \cdots < \omega_p \leq \pi$$

- M+1 is the filter order
- P >> M + 1 (rule of thumb P=15M)
- Yields a (good) approximation of the original problem

Example: Least Squares

- Target: Design M+1= 2N+1 filter
- First design non-causal $\tilde{H}(e^{j\omega})$ and hence $\tilde{h}[n]$
- Then, shift to make causal

$$h[n] = \tilde{h}[n - M/2]$$

$$H(e^{j\omega}) = e^{-j\frac{M}{2}}\tilde{H}(e^{j\omega})$$

Example: Least Squares

$$\tilde{h} = \left[\tilde{h}[-N], \tilde{h}[-N+1], \cdots, \tilde{h}[N]\right]^T$$

$$b = \left[H_d(e^{j\omega_1}), \cdots, H_d(e^{j\omega_P})\right]^T$$

$$A = \begin{bmatrix} e^{-j\omega_1(-N)} & \dots & e^{-j\omega_1(+N)} \\ \vdots & & \\ e^{-j\omega_P(-N)} & \dots & e^{-j\omega_P(+N)} \end{bmatrix}$$

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||_2^2$$

Least-Squares

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||_2^2$$

Solution:

$$\tilde{h} = (A^*A)^{-1}A^*b$$

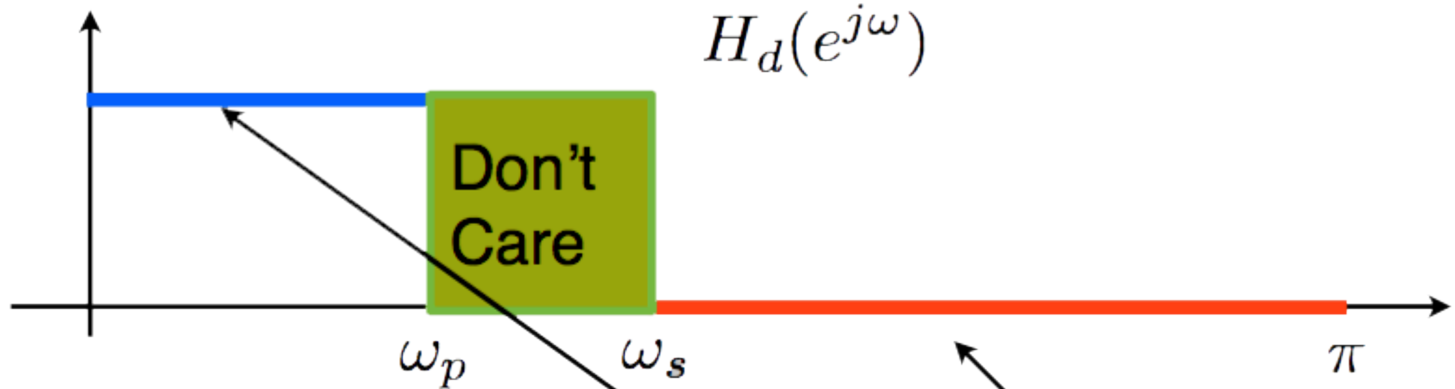
- Result will generally be non-symmetric and complex valued.
- However, if $\tilde{H}(e^{j\omega})$ is real, $\tilde{h}[n]$ should have symmetry!

Design of Linear-Phase L.P Filter

- Suppose:
  - $\tilde{H}(e^{j\omega})$ is real-symmetric
  - M is even (M+1 length)
- Then:
  - $\tilde{h}[n]$ is real-symmetric around midpoint
- So:

$$
\begin{aligned}
\tilde{H}(e^{j\omega}) &= \tilde{h}[0] + \tilde{h}[1]e^{-j\omega} + \tilde{h}[-1]e^{+j\omega} \\
&\quad + \tilde{h}[2]e^{-j2\omega} + \tilde{h}[-2]e^{+j2\omega} \ldots \\
&= \tilde{h}[0] + 2\cos(\omega)\tilde{h}[1] + 2\cos(2\omega)\tilde{h}[2] + \cdots
\end{aligned}
$$

Least-Squares Linear Phase Filter



Given M, ω_P, ω_S find the best LS filter:

$$A = \begin{bmatrix} 1 & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_p) \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix}$$

$$b = [1, 1, \cdots, 1, 0, 0, \cdots, 0]^T$$

Capital P

Least-Squares Linear Phase Filter

Given M, ω_P, ω_s find the best LS filter:

$$A = \begin{bmatrix} 1 & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_p) \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix} \qquad b = [1, 1, \cdots, 1, 0, 0, \cdots, 0]^T$$

$$\tilde{h}_+ = [\tilde{h}[0], \cdots, \tilde{h}[\frac{M}{2}]]^T = (A^*A)^{-1}A^*b$$

$$\tilde{h} = \begin{cases} \tilde{h}_+[n] & n \geq 0 \\ \tilde{h}_+[-n] & n < 0 \end{cases}$$

$$h[n] = \tilde{h}[n - M/2]$$

Extension:

- LS has no preference for pass band or stop band
- Use weighting of LS to change ratio

want to solve the discrete version of:

$$\text{minimize} \quad \int_{-\pi}^{\pi} W(\omega)|H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

where W(ω) is δp in the pass band and δs in stop band

Similarly: W(ω) is 1 in the pass band and δp/δs in stop band

Weighted Least-Squares

$$\operatorname{argmin}_{\tilde{h}_+} \quad (A\tilde{h}_+ - b)^* W^2 (A\tilde{h}_+ - b)$$

Solution:

$$\tilde{h}_+ = (A^* W^2 A)^{-1} W^2 A^* b$$

$$W = \begin{bmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & \cdots & & & \\ & & & \frac{\delta_p}{\delta_s} & & \\ & & & & \cdots & \\ 0 & & & & & \frac{\delta_p}{\delta_s} \end{bmatrix}$$

Optimality – min-max

- Chebychev Design (min-max)

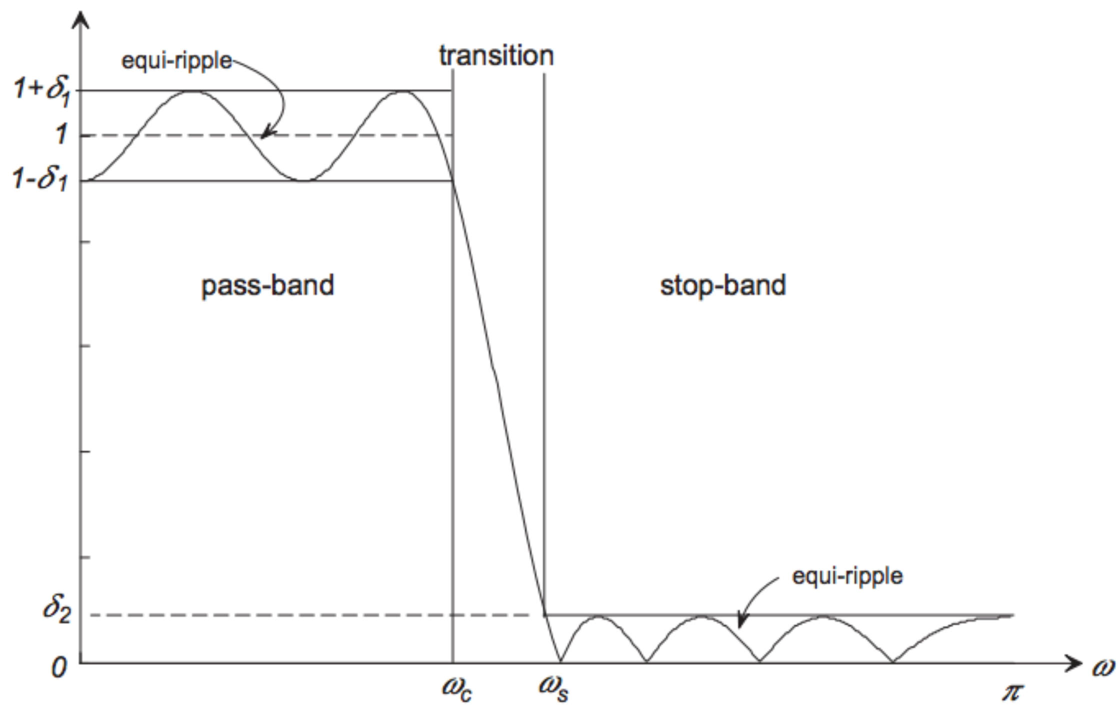
$$\text{minimize}_{\omega \in \text{care}} \quad \max |H(e^{j\omega}) - H_d(e^{j\omega})|$$

  - Parks-McClellan algorithm - equiripple
  - Also known as Remez exchange algorithms (signal.remez)
  - Can also use convex optimization

Parks-McClellan

- Allows for multiple pass- and stop-bands.
- Is an equi-ripple design in the pass- and stop-bands, but allows independent weighting of the ripple in each band.
- Allows specification of the band edges.

Parks-McClellan: LP Filter



- For the low-pass filter shown above the specifications are

$$\begin{array}{rcccl} 1-\delta_1 & < & H(e^{j\omega}) & < & 1+\delta_1 \quad \text{in the pass-band } 0 < \omega \leq \omega_c \\ -\delta_2 & < & H(e^{j\omega}) & < & \delta_2 \quad \text{in the stop-band } \omega_s < \omega \leq \pi. \end{array}$$

Min-Max Filter Design

- Constraints:
  - min-max pass-band ripple

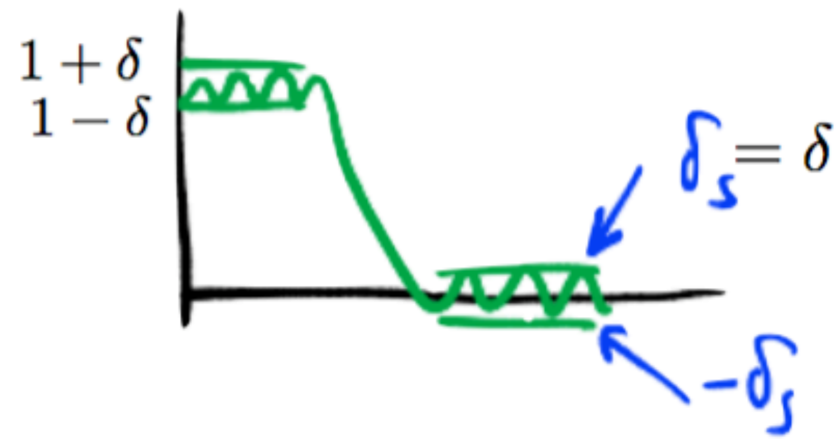
$$1-\delta_p \le |H(e^{j\omega})| \le 1+\delta_p, \qquad 0 \le w \le \omega_p$$

  - min-max stop-band ripple

$$|H(e^{j\omega})| \le \delta_s, \qquad \omega_s \le w \le \pi$$

Min-Max Ripple Design

- Given ω_p, ω_s, M, find δ, $\tilde{h}_+$



$$
\begin{aligned}
&\text{minimize} \quad && \delta \\
&\text{Subject to :} && \\
& && 1-\delta \le \tilde{H}(e^{j\omega_k}) \le 1+\delta \qquad 0 \le \omega_k \le \omega_p \\
& && -\delta \le \tilde{H}(e^{j\omega_k}) \le \delta \qquad \omega_s \le \omega_k \le \pi \\
& && \delta > 0
\end{aligned}
$$

- Formulation is a linear program with solution δ, $\tilde{h}_+$
- A well studied class of problems with good solvers

Min-Max Ripple via LP

$$\begin{aligned} \text{minimize} \quad & \delta \\ \text{subject to :} \quad & \\ & 1-\delta \preceq A_p\tilde{h}_+ \preceq 1+\delta \\ & \text{-}\delta \preceq A_s\tilde{h}_+ \preceq \delta \\ & \delta > 0 \end{aligned}$$

$$A_p = \begin{bmatrix} 1 & 2\cos(\omega_1) & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ & \vdots & & \\ 1 & 2\cos(\omega_p) & \cdots & 2\cos(\frac{M}{2}\omega_p) \end{bmatrix}$$

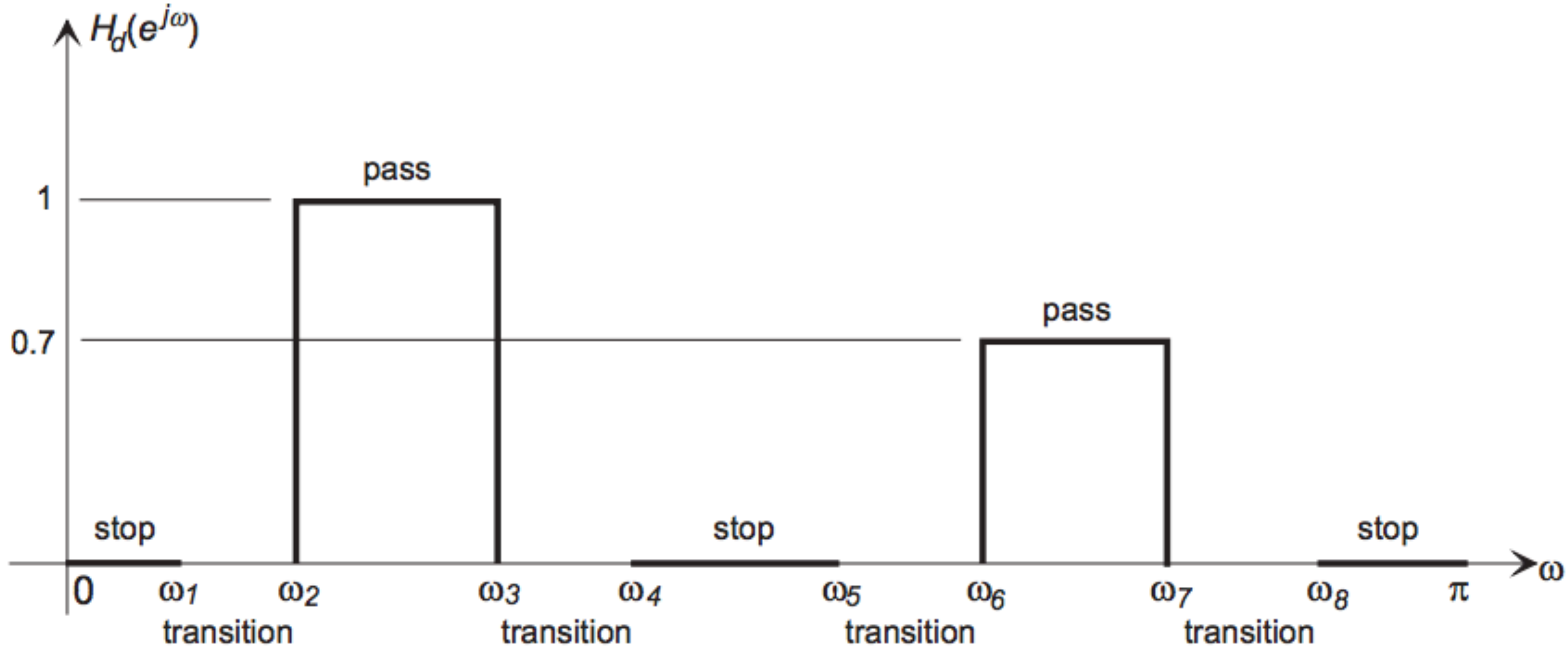
$$A_s = \begin{bmatrix} 1 & 2\cos(\omega_s) & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ & \vdots & & \\ 1 & 2\cos(\omega_P) & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix}$$

capital P

MATLAB Parks-McClellan Function

- `b = firpm(M,F,A,W)`
  - `b` is the array of filter coefficients (impulse response)
  - `M` is the filter order (`M+1` is the length of the filter),
  - `F` is a vector of band edge frequencies in ascending order
  - `A` is a set of filter gains at the band edges
  - `W` is an optional set of relative weights to be applied to each of the bands

MATLAB Parks-McClellan Function



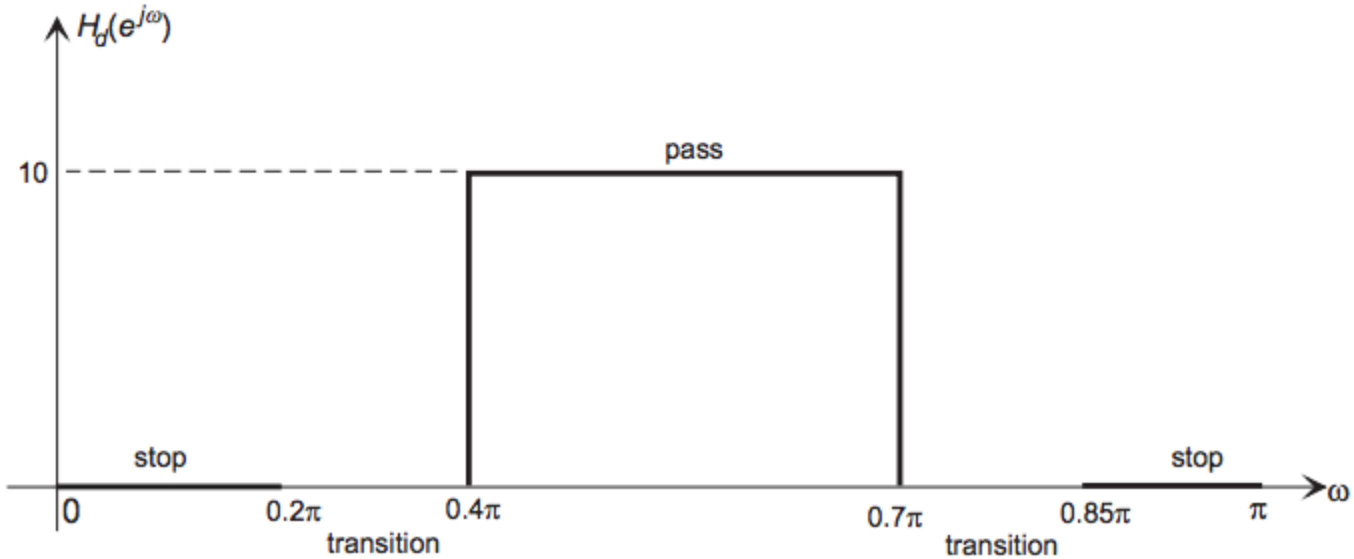
$$F = [0 \;\; \omega_1 \;\; \omega_2 \;\; \omega_3 \;\; \omega_4 \;\; \omega_5 \;\; \omega_6 \;\; \omega_7 \;\; \omega_8 \;\; 1]$$

$$A = [0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0.7 \;\; 0.7 \;\; 0 \;\; 0]$$

$$W = [10 \;\; 1 \;\; 10 \;\; 1 \;\; 10]$$

MATLAB Example

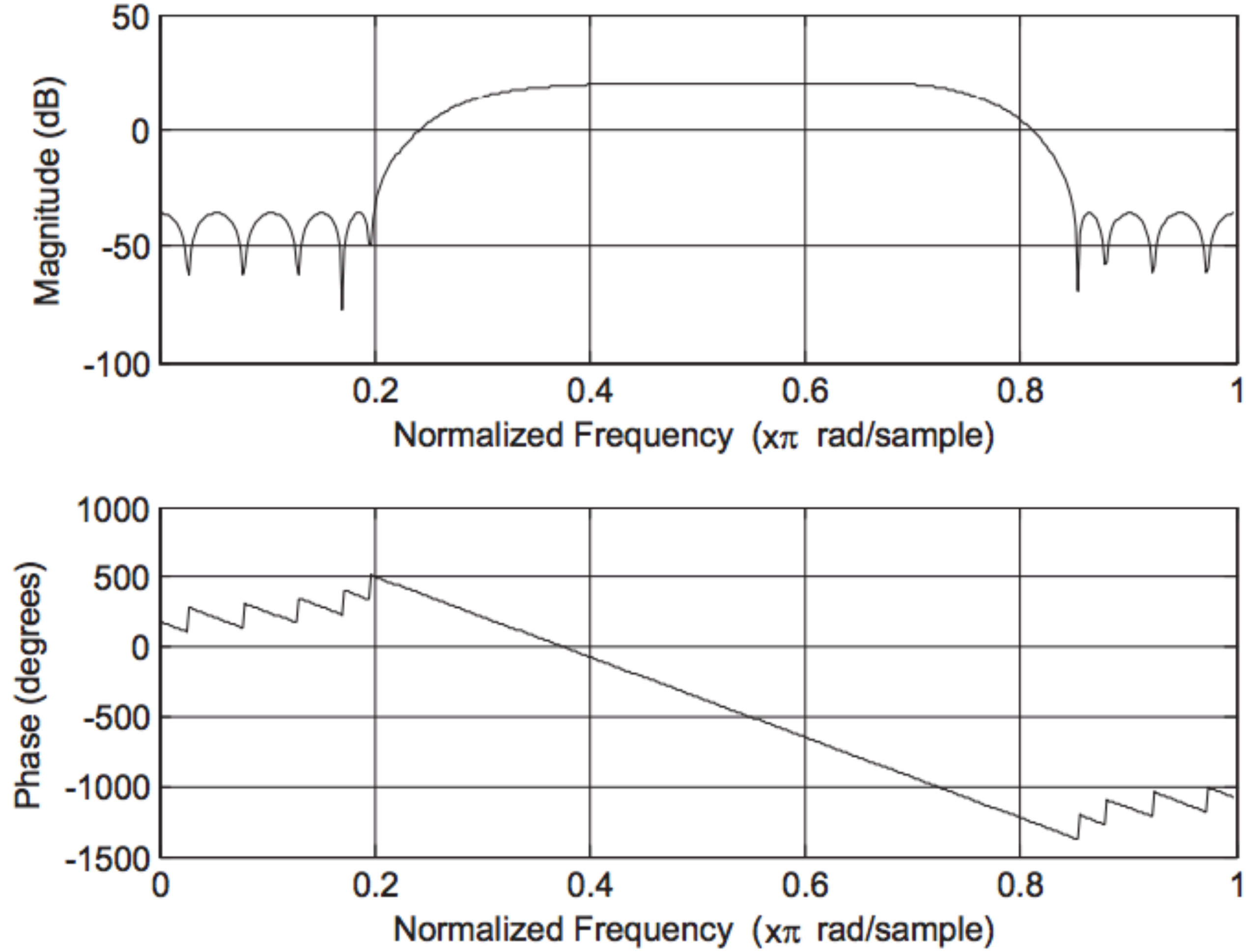
- Design a 33 length PM band-pass filter and weight the stop-band ripple 10x more than the pass-band ripple




```
h=firpm(32,[0 0.2 0.4 0.7 0.85 1],[0 0 10 10 0 0],[10 1 10])
freqz(h,1)
```

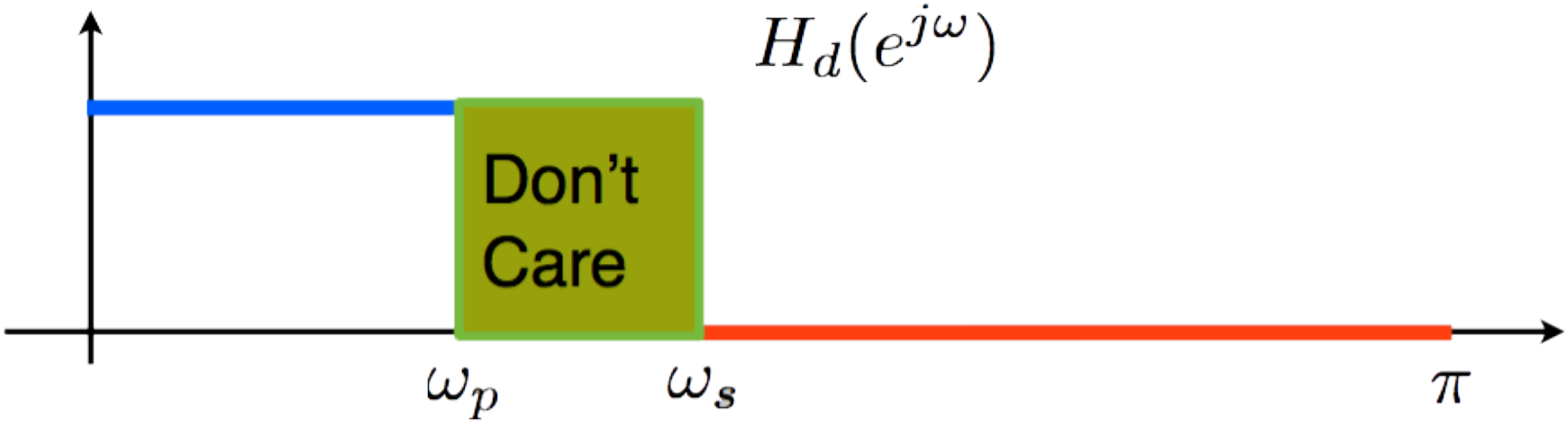
MATLAB Example

- Des op-
  ban




```
h=firp          1 10])
freqz(
```

Optimality – Least Squares



- Least Squares:

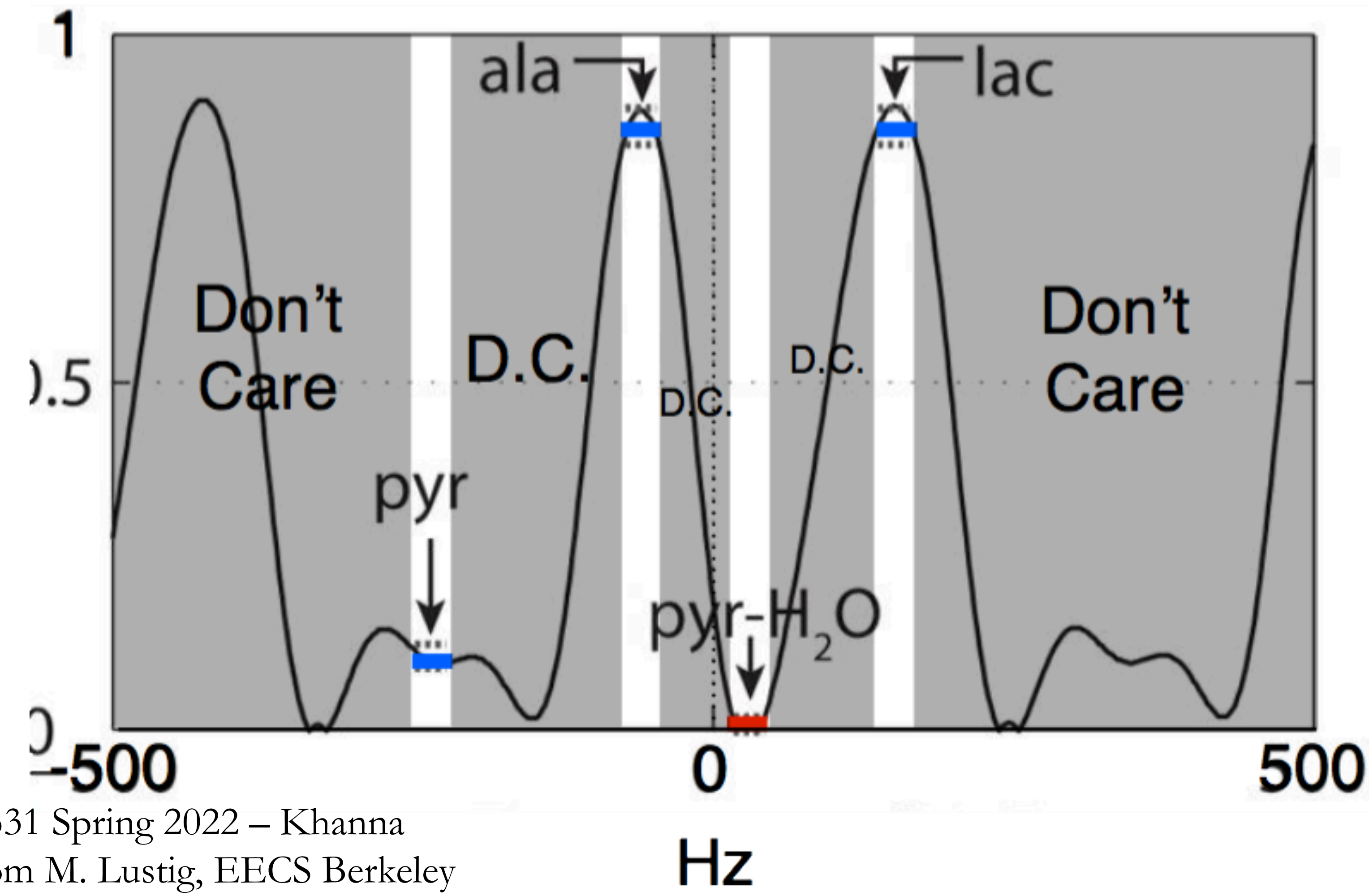
$$\text{minimize} \int_{\omega \in \text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

- Parks-McClellan

$$\min_{\{h_e[n]:0 \le n \le L\}} \left(\max_{\omega \in F} |E(\omega)| \right),$$

Example of Complex Filter

- Larson et. al, "Multiband Excitation Pulses for Hyperpolarized 13C Dynamic Chemical Shift Imaging" JMR 2008;194(1):121-127
- Need to design length 11 filter with following frequency response:

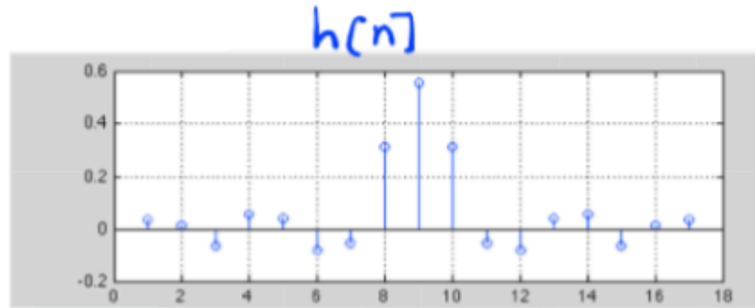


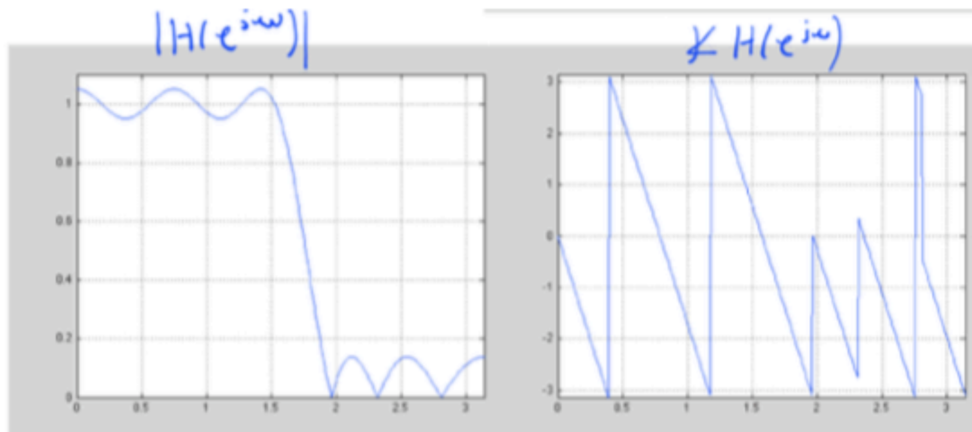
Convex Optimization

- Many tools and Solvers
- Tools:
  - CVX (Matlab) http://cvxr.com/cvx/
  - CVXOPT, CVXMOD (Python)
- Engines:
  - Sedumi (Free)
  - MOSEK (commercial)

Using CVX (in Matlab)

```
M = 16;
wp = 0.5*pi;
ws = 0.6*pi;
MM = 15*M;
w = linspace(0,pi,MM);

idxp = find(w <=wp);
idxs = find(w >=ws);

Ap = [ones(length(idxp),1) 2*cos(kron(w(idxp)',
[1:M/2]))];
As = [ones(length(idxs),1) 2*cos(kron(w(idxs)',
[1:M/2]))];

% optimization
cvx_begin
   variable hh(M/2+1,1);
   variable d(1,1);

   minimize(d)
   subject to
      Ap*hh <=1+d;
      Ap*hh >=1-d;
      As*hh <  d;
      As*hh > -d;
      ds>0;
cvx_end
h = [hh(end:-1:1) ; hh(2:end)];
```





Discrete Fourier Series

Reminder: Eigenvalue (DTFT)

- $x[n]=e^{j\omega n}$

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

$$= \sum_{k=-\infty}^{\infty} e^{j\omega(n-k)}h[k]$$

$$= e^{j\omega n} \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

$$= H(e^{j\omega})e^{j\omega n}$$

$$\boxed{H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}}$$

- Describes the change in amplitude and phase of signal at frequency ω
- Frequency response
- Complex value
  - Re and Im
  - Mag and Phase

Discrete Fourier Series

- Definition:
  - Consider N-periodic signal:

$$\tilde{x}[n+N] = \tilde{x}[n] \quad \forall n$$

  - Frequency-domain also periodic in N:

$$\tilde{X}[k+N] = \tilde{X}[k] \quad \forall k$$

  - "~" indicates periodic signal/spectrum

Discrete Fourier Series

- Define:

$$W_N \triangleq e^{-j2\pi/N}$$

- DFS:

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}[k]W_N^{-kn}$$

$$\tilde{X}[k] = \sum_{n=0}^{N-1}\tilde{x}[n]W_N^{kn}$$

Discrete Fourier Series

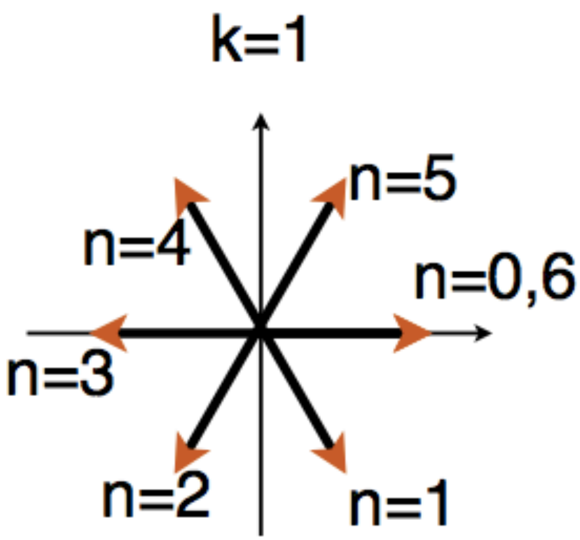
$$W_N \triangleq e^{-j2\pi/N}$$

- Properties of W_N:
  - $W_N^0 = W_N^N = W_N^{2N} = \ldots = 1$
  - $W_N^{k+r} = W_N^k W_N^r$ and, $W_N^{k+N} = W_N^k$

Discrete Fourier Series

$$W_N \triangleq e^{-j2\pi/N}$$

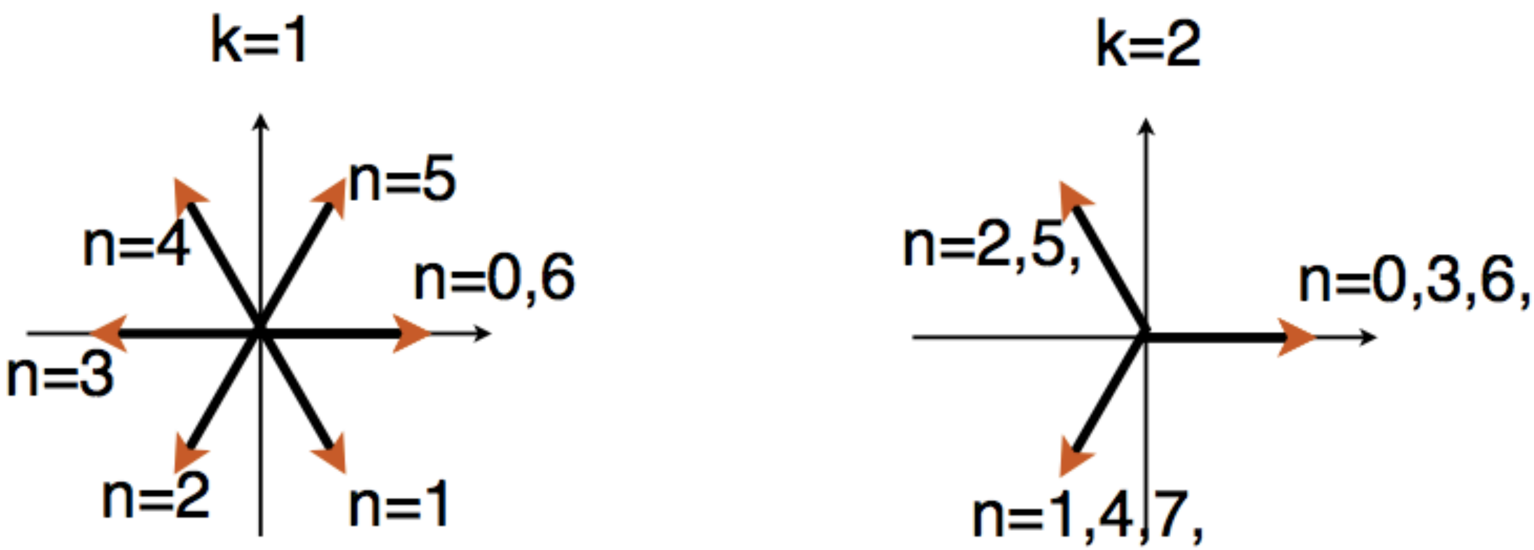
- Properties of W_N:
  - $W_N^0 = W_N^N = W_N^{2N} = \ldots = 1$
  - $W_N^{k+r} = W_N^k W_N^r$ and, $W_N^{k+N} = W_N^k$
- Example: W_N^{kn} (N=6)



Discrete Fourier Series

$$W_N \triangleq e^{-j2\pi/N}$$

- Properties of W_N:
  - $W_N^0 = W_N^N = W_N^{2N} = \ldots = 1$
  - $W_N^{k+r} = W_N^k W_N^r$ and, $W_N^{k+N} = W_N^k$
- Example: W_N^{kn} (N=6)



Discrete Fourier Transform

- By convention, work with one period:

$$x[n] \triangleq \begin{cases} \tilde{x}[n] & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

$$X[k] \triangleq \begin{cases} \tilde{X}[k] & 0 \leq k \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

Same, but different!

Discrete Fourier Transform

- The DFT

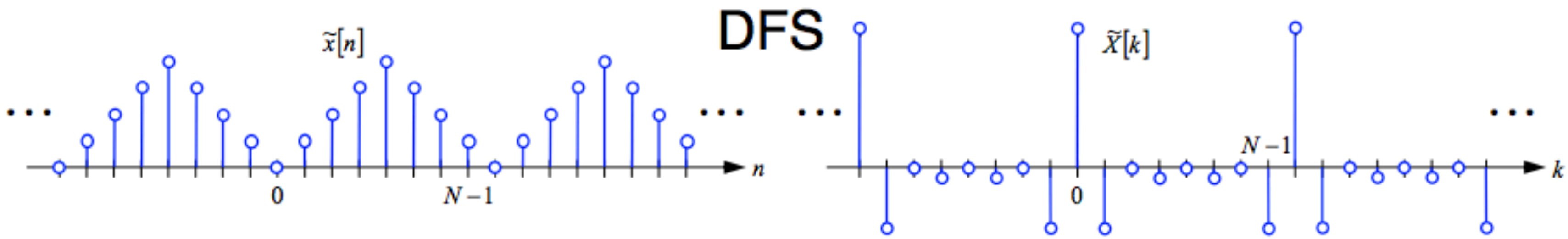
$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn} \quad \text{Inverse DFT, synthesis}$$

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} \quad \text{DFT, analysis}$$

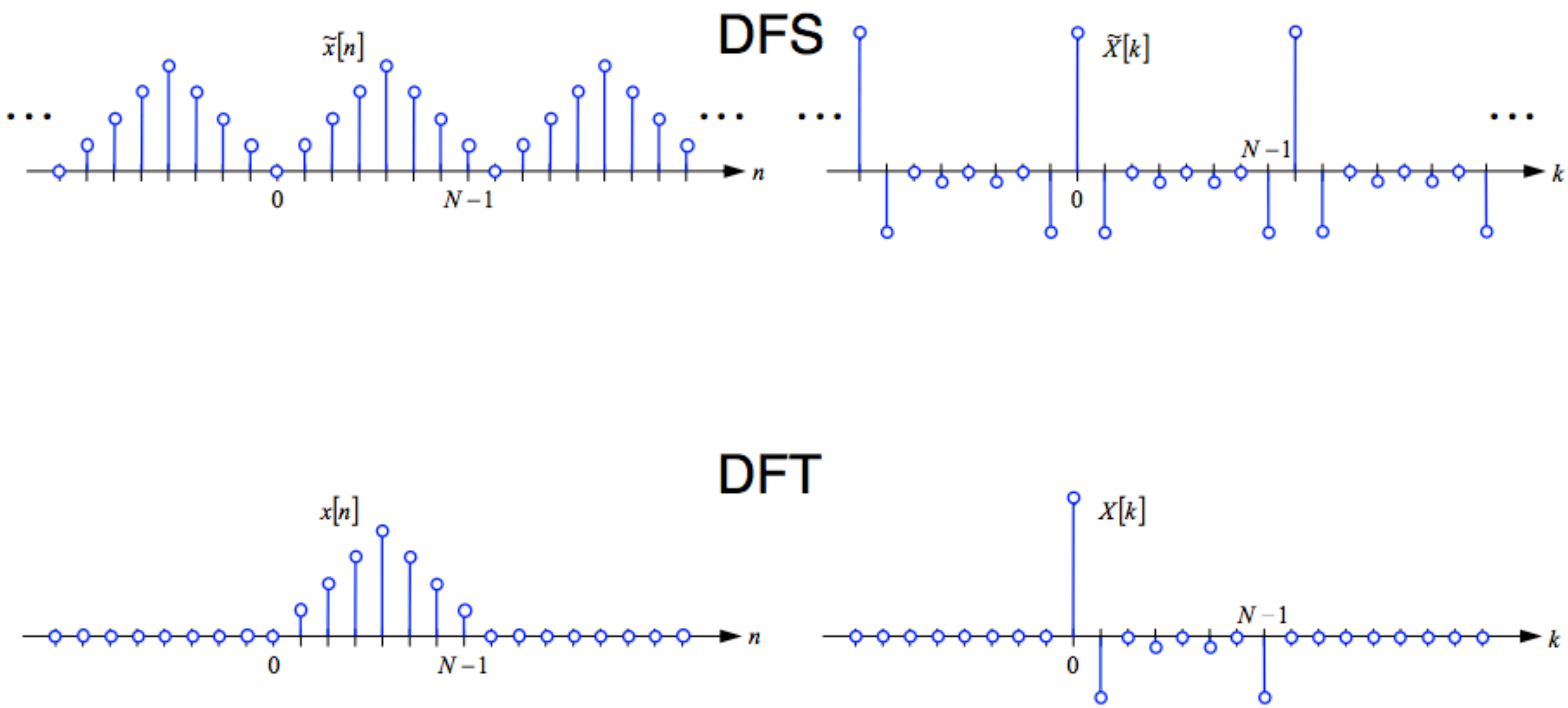
- It is understood that,

$$\begin{aligned} x[n] &= 0 \quad \text{outside } 0 \le n \le N-1 \\ X[k] &= 0 \quad \text{outside } 0 \le k \le N-1 \end{aligned}$$

DFS vs. DFT

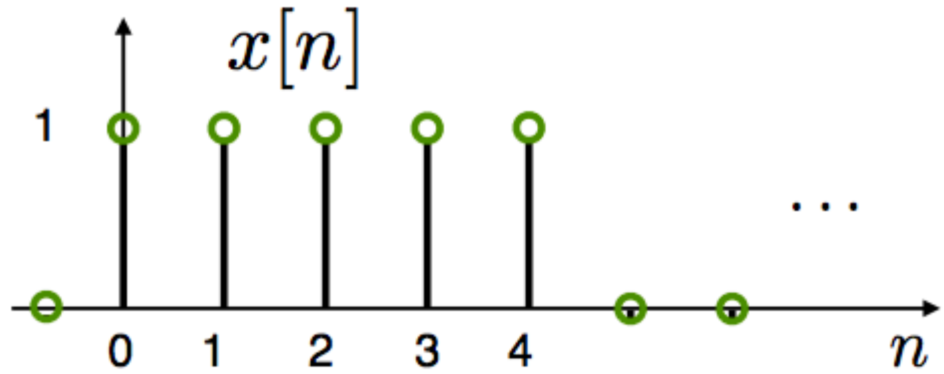


DFS vs. DFT



Example

$$W_N \triangleq e^{-j2\pi/N}$$

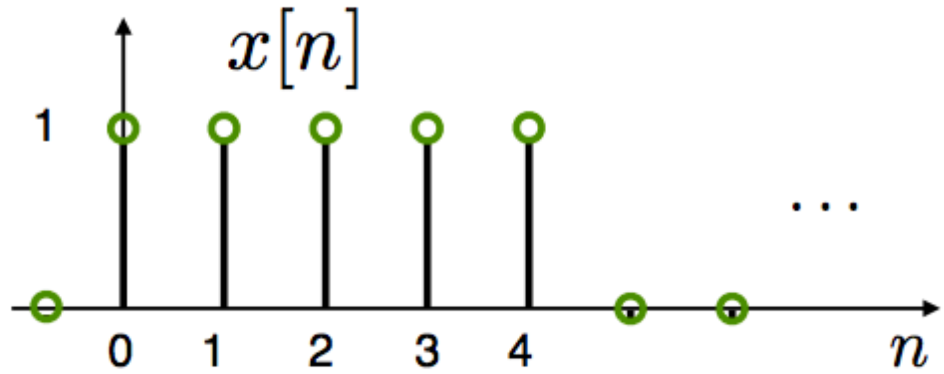


Take N=5

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

Example

$$W_N \triangleq e^{-j2\pi/N}$$



- Take N=5

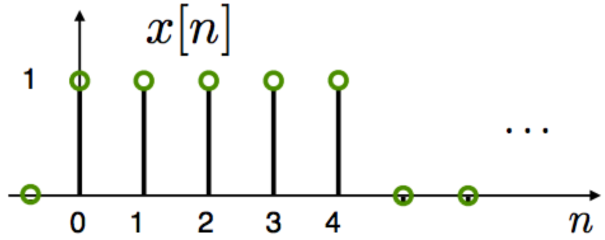
$$X[k] = \begin{cases} \sum_{n=0}^{4} W_5^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases}$$

$$= 5\delta[k]$$

"5-point DFT"

Example

$$W_N \triangleq e^{-j2\pi/N}$$



- Take N=5

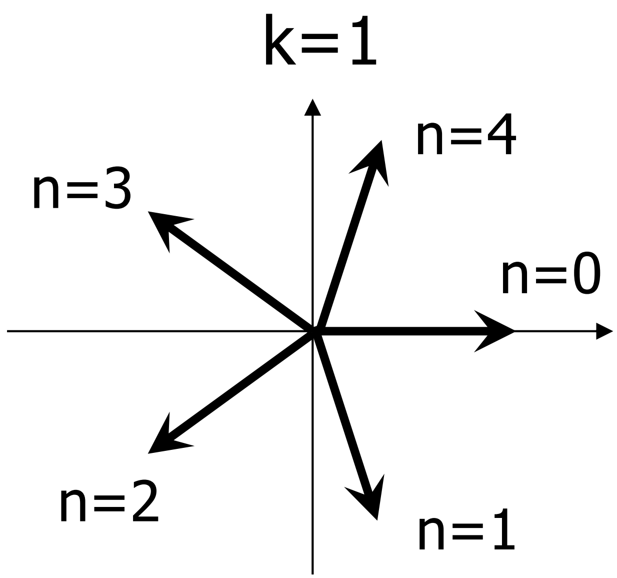
$$\begin{aligned} X[k] &= \begin{cases} \sum_{n=0}^{4} W_5^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases} \\ &= 5\delta[k] \end{aligned}$$

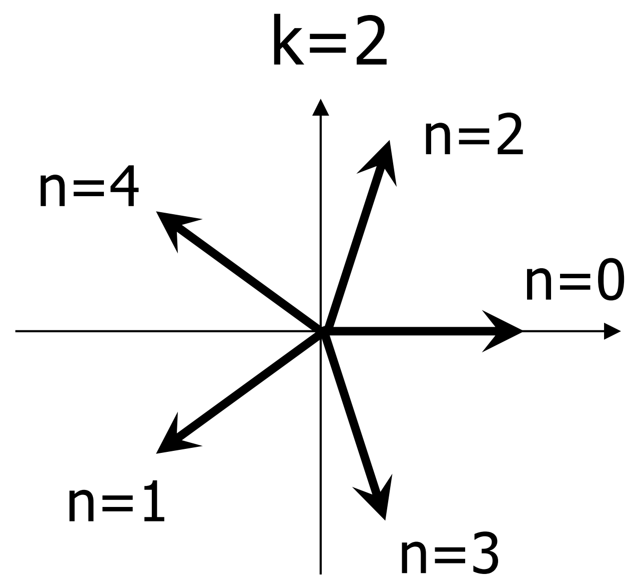
"5-point DFT"

Discrete Fourier Series

$$W_N \triangleq e^{-j2\pi/N}$$

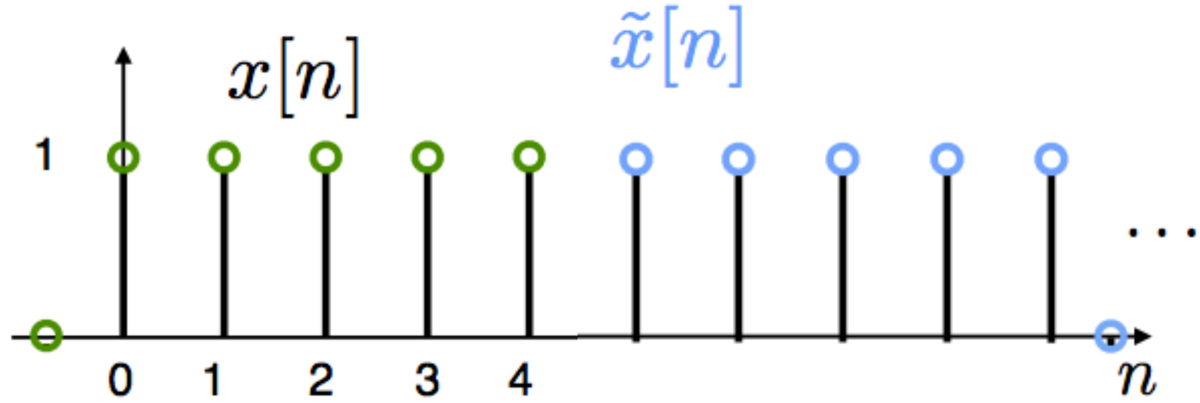
- Properties of WN:
  - $W_N^0 = W_N^N = W_N^{2N} = \ldots = 1$
  - $W_N^{k+r} = W_N^k W_N^r$ and, $W_N^{k+N} = W_N^k$
- Example: W_N^{kn} (N=5)





Example

$$W_N \triangleq e^{-j2\pi/N}$$



Take N=5

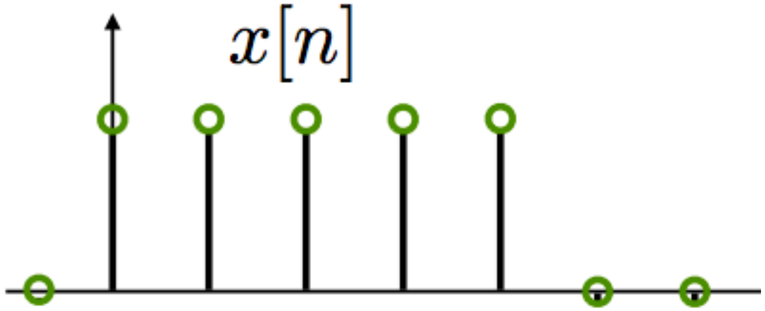
$$\begin{aligned} X[k] &= \begin{cases} \sum_{n=0}^{4} W_5^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases} \\ &= 5\delta[k] \end{aligned}$$

"5-point DFT"

Example

$$W_N \triangleq e^{-j2\pi/N}$$

- Q: What if we take N=10?
- A: $X[k] = \tilde{X}[k]$ where $\tilde{x}[n]$ is a period-10 seq.



Example

$$W_N \triangleq e^{-j2\pi/N}$$

- Q: What if we take N=10?
- A: $X[k] = \tilde{X}[k]$ where $\tilde{x}[n]$ is a period-10 seq.

$$X[k] = \begin{cases} \sum_{n=0}^{4} W_{10}^{nk} & k = 0, 1, 2, \cdots, 9 \\ 0 & \text{otherwise} \end{cases}$$

"10-point DFT"

Example

- Now, sum from n=0 to 9

$$X[k] = \sum_{n=0}^{9} x[n] W_{10}^{nk}$$

Example

- Now, sum from n=0 to 9

$$X[k] = \sum_{n=0}^{9} x[n] W_{10}^{nk}$$

$$= \sum_{n=0}^{4} W_{10}^{nk}$$

$$= e^{-j\frac{4\pi}{10}k} \frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$

"10-point DFT"

DFT vs. DTFT

- For finite sequences of length N:
  - The N-point DFT of x[n] is:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)nk} \qquad 0 \le k \le N-1$$

  - The DTFT of x[n] is:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \qquad -\infty < \omega < \infty$$

DFT vs. DTFT

- The DFT are samples of the DTFT at N equally spaced frequencies

$$X[k] = X(e^{j\omega})\big|_{\omega = k\frac{2\pi}{N}} \qquad 0 \le k \le N-1$$

DFT vs DTFT

- Back to example

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{4} W_{10}^{nk} \\
&= e^{-j\frac{4\pi}{10}k} \frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}
\end{aligned}
$$

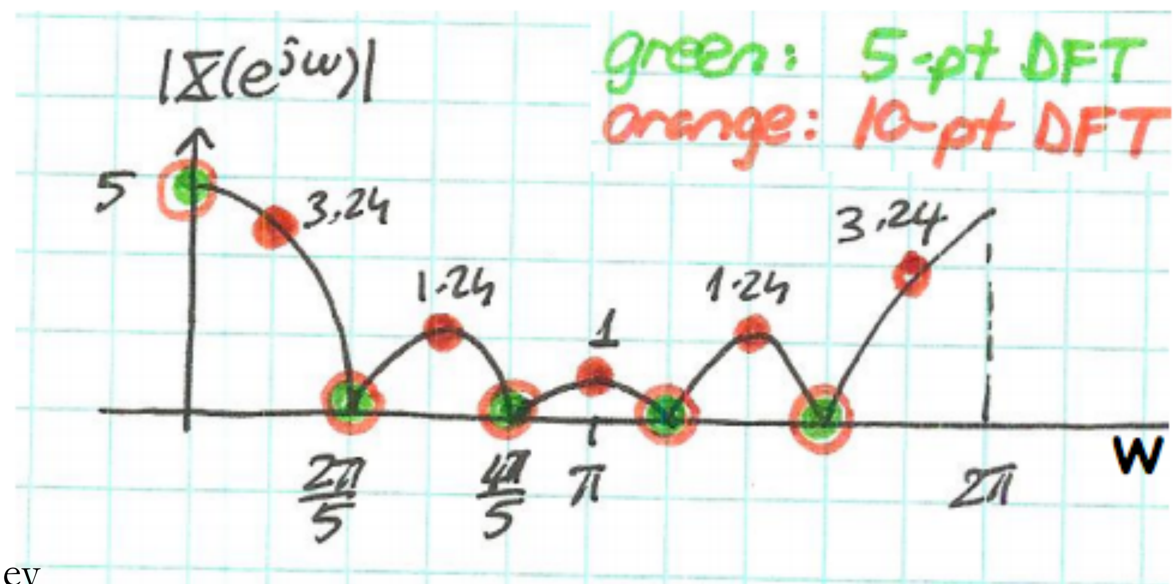
"10-point DFT"

DFT vs DTFT

- Back to example

$$X[k] = \sum_{n=0}^{4} W_{10}^{nk}$$

$$= e^{-j\frac{4\pi}{10}k}\frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$

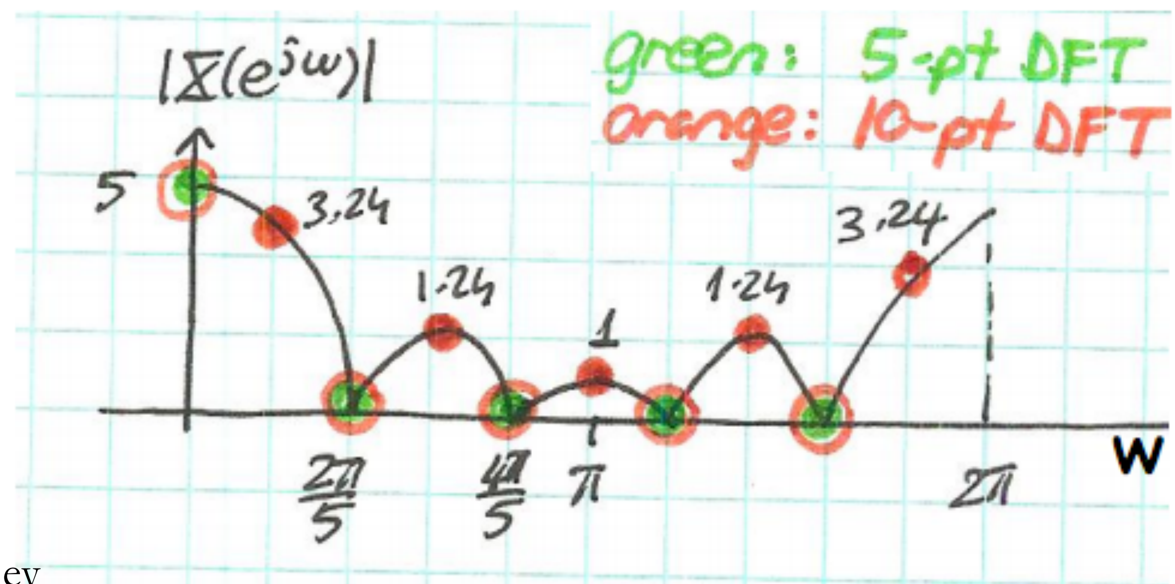


DFT vs DTFT

- Back to example

$$X[k] = \sum_{n=0}^{4} W_{10}^{nk}$$
$$= e^{-j\frac{4\pi}{10}k}\frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$

Use `fftshift` to center around dc



DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

$$N \cdot x^*[n] \quad = \quad N\left(\mathcal{DFT}^{-1}\left\{X[k]\right\}\right)^*$$

DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

$$\begin{aligned} N \cdot x^*[n] &= N\left(\mathcal{DFT}^{-1}\{X[k]\}\right)^* \\ &= N\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-kn}\right)^* \end{aligned}$$

DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

$$\begin{aligned} N \cdot x^*[n] &= N \left(\mathcal{DFT}^{-1} \{X[k]\} \right)^* \\ &= N \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \right)^* \\ &= \sum_{k=0}^{N-1} X^*[k] W_N^{kn} \end{aligned}$$

DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

$$
\begin{aligned}
N \cdot x^*[n] &= N\left(\mathcal{DFT}^{-1}\{X[k]\}\right)^* \\
&= N\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}\right)^* \\
&= \sum_{k=0}^{N-1} X^*[k]W_N^{kn} \\
&= \mathcal{DFT}\{X^*[k]\}.
\end{aligned}
$$

DFT and Inverse DFT

- Use the DFT to compute the inverse DFT. How?

$$
\begin{aligned}
N \cdot x^*[n] &= N\left(\mathcal{DFT}^{-1}\{X[k]\}\right)^* \\
&= N\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}\right)^* \\
&= \sum_{k=0}^{N-1} X^*[k]W_N^{kn} \\
&= \mathcal{DFT}\{X^*[k]\}.
\end{aligned}
$$

DFT and Inverse DFT

- So

$$\mathcal{DFT}\left\{X^*[k]\right\} = N\left(\mathcal{DFT}^{-1}\left\{X[k]\right\}\right)^*$$

DFT and Inverse DFT

- So

$$\mathcal{DFT}\left\{X^*[k]\right\} = N\left(\mathcal{DFT}^{-1}\left\{X[k]\right\}\right)^*$$

$$\mathcal{DFT}^{-1}\left\{X[k]\right\} = \frac{1}{N}\left(\mathcal{DFT}\left\{X^*[k]\right\}\right)^*$$

DFT and Inverse DFT

- So

$$\mathcal{DFT}\{X^*[k]\} = N\left(\mathcal{DFT}^{-1}\{X[k]\}\right)^*$$

$$\mathcal{DFT}^{-1}\{X[k]\} = \frac{1}{N}\left(\mathcal{DFT}\{X^*[k]\}\right)^*$$

- Implement IDFT by:
  - Take complex conjugate
  - Take DFT
  - Multiply by 1/N
  - Take complex conjugate

DFT as Matrix Operator

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

DFT:

$$\begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix} = \begin{pmatrix} W_N^{00} & \cdots & W_N^{0n} & \cdots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{k0} & \cdots & W_N^{kn} & \cdots & W_N^{k(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \cdots & W_N^{(N-1)n} & \cdots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix}$$

DFT as Matrix Operator

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

DFT:

$$\begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix} = \begin{pmatrix} W_N^{00} & \cdots & W_N^{0n} & \cdots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{k0} & \cdots & W_N^{kn} & \cdots & W_N^{k(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \cdots & W_N^{(N-1)n} & \cdots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix}$$

IDFT:

$$\begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} W_N^{-00} & \cdots & W_N^{-0k} & \cdots & W_N^{-0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-n0} & \cdots & W_N^{-nk} & \cdots & W_N^{-n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-(N-1)0} & \cdots & W_N^{-(N-1)k} & \cdots & W_N^{-(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix}$$

DFT as Matrix Operator

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

DFT:

$$\begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix} = \begin{pmatrix} W_N^{00} & \cdots & W_N^{0n} & \cdots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{k0} & \cdots & W_N^{kn} & \cdots & W_N^{k(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \cdots & W_N^{(N-1)n} & \cdots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix}$$

IDFT:

$$\begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} W_N^{-00} & \cdots & W_N^{-0k} & \cdots & W_N^{-0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-n0} & \cdots & W_N^{-nk} & \cdots & W_N^{-n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-(N-1)0} & \cdots & W_N^{-(N-1)k} & \cdots & W_N^{-(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix}$$

N^2 complex multiplies

DFT as Matrix Operator

- Can write compactly as

$$
\begin{aligned}
\mathbf{X} &= \mathbf{W}_N \, \mathbf{x} \\
\mathbf{x} &= \frac{1}{N} \, \mathbf{W}_N^* \, \mathbf{X}
\end{aligned}
$$

Properties of the DFT

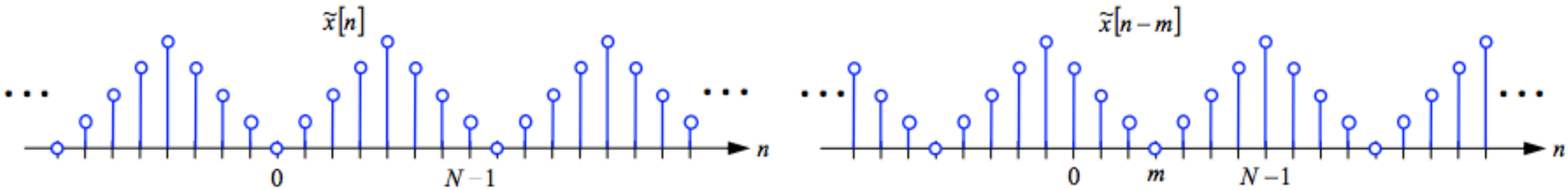
- Properties of DFT inherited from DFS
- Linearity

$$\alpha_1 x_1[n] + \alpha_2 x_2[n] \leftrightarrow \alpha_1 X_1[k] + \alpha_2 X_2[k]$$

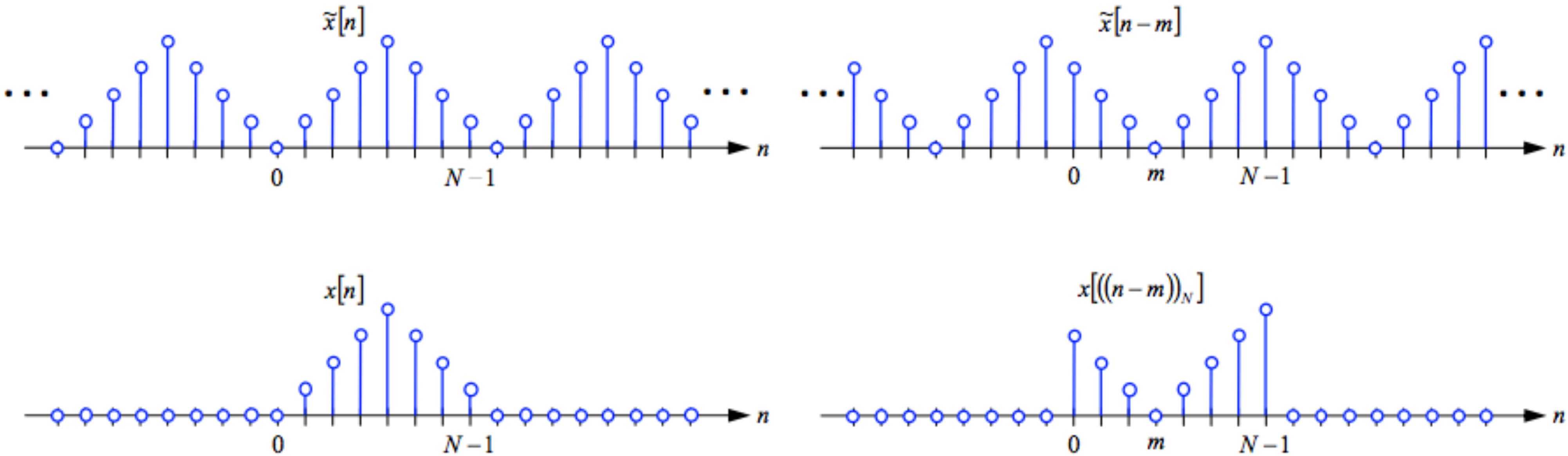
- Circular Time Shift

$$x[((n-m))_N] \leftrightarrow X[k]e^{-j(2\pi/N)km} = X[k]W_N^{km}$$

Circular Shift



Circular Shift



Properties of DFT

- Circular frequency shift

$$x[n]e^{j(2\pi/N)nl} = x[n]W_N^{-nl} \leftrightarrow X[((k-l))_N]$$

- Complex Conjugation

$$x^*[n] \leftrightarrow X^*[((-k))_N]$$

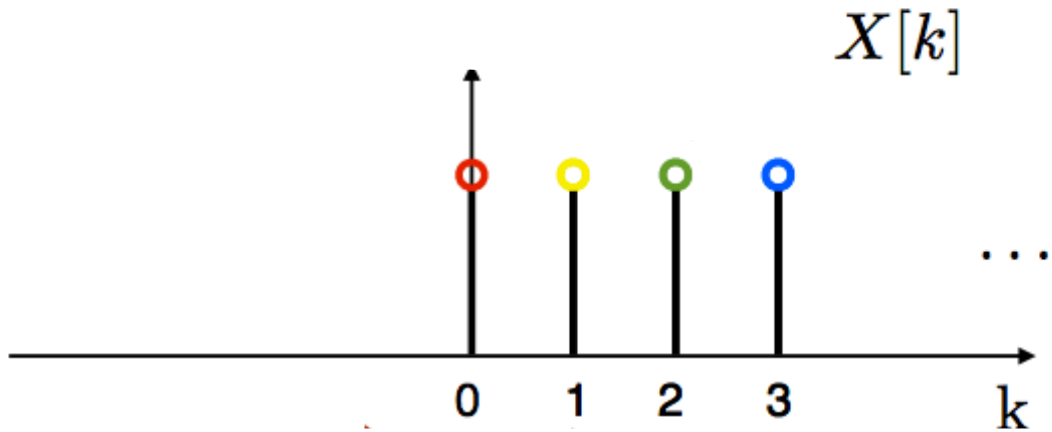
- Conjugate Symmetry for Real Signals

$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

- Symmetry

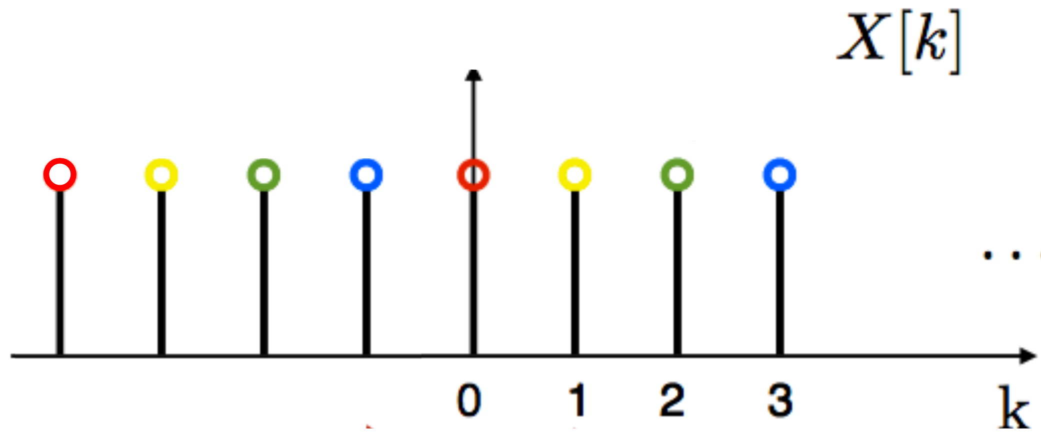


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

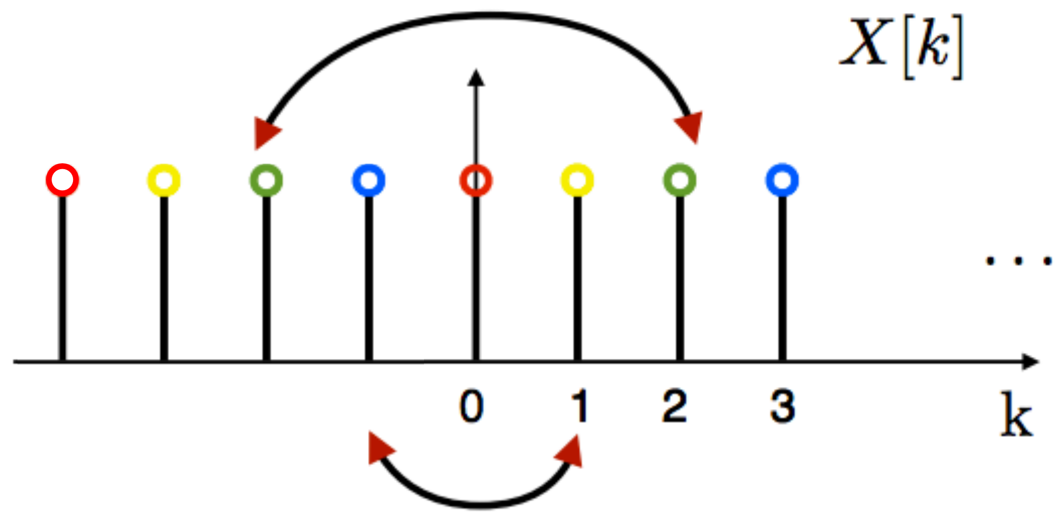
–Symmetry



$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT
–Symmetry

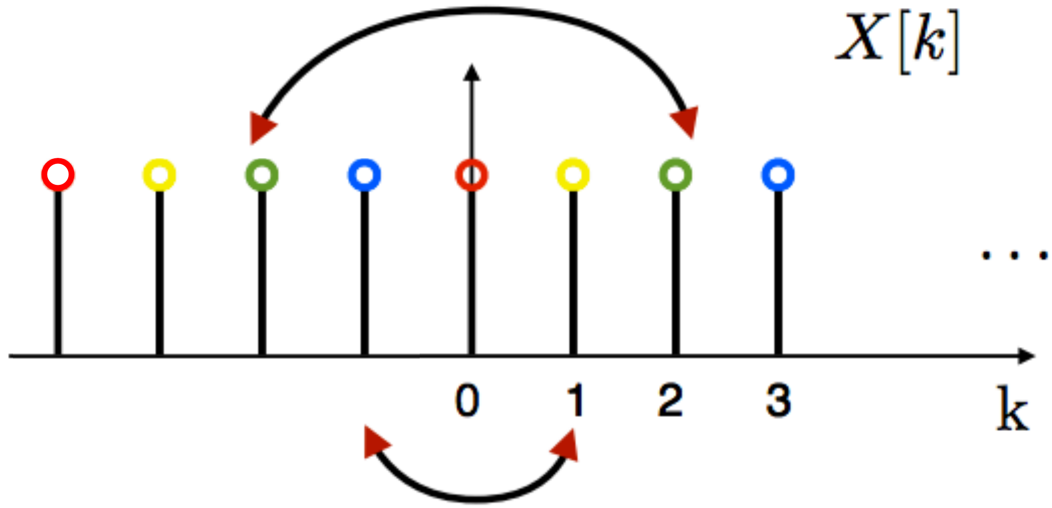


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

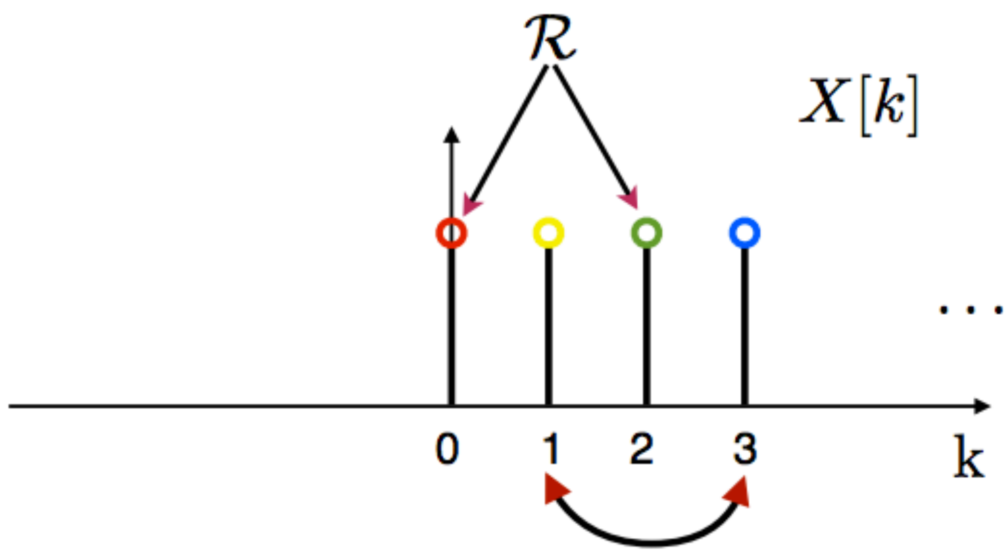
4-point DFT

–Symmetry



4-point DFT

–Symmetry

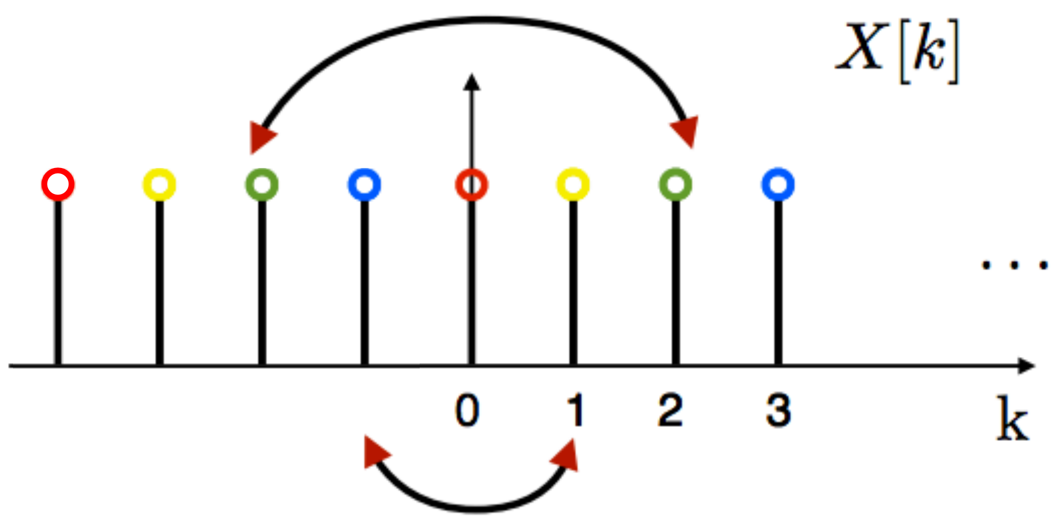


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

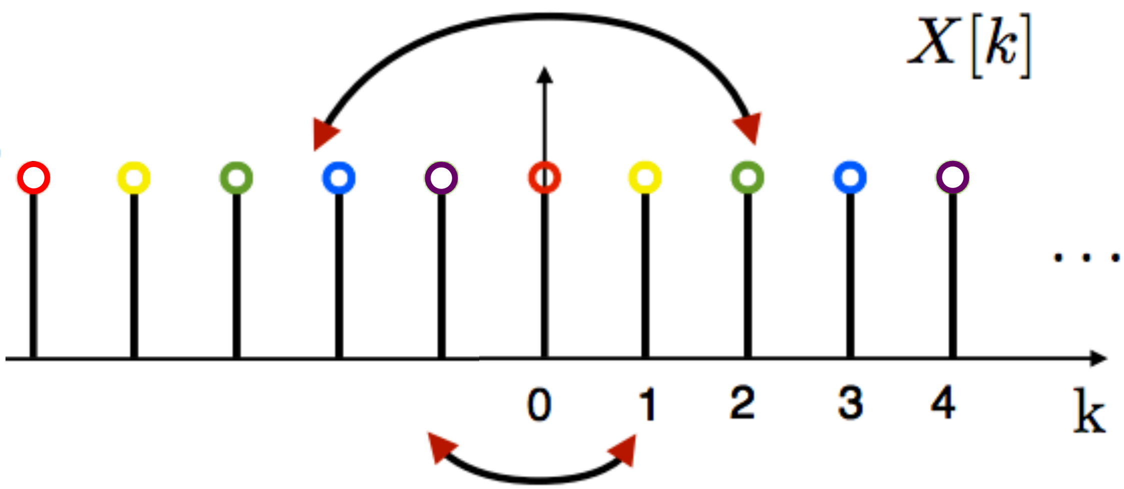
4-point DFT

–Symmetry



5-point DFT

–Symmetry

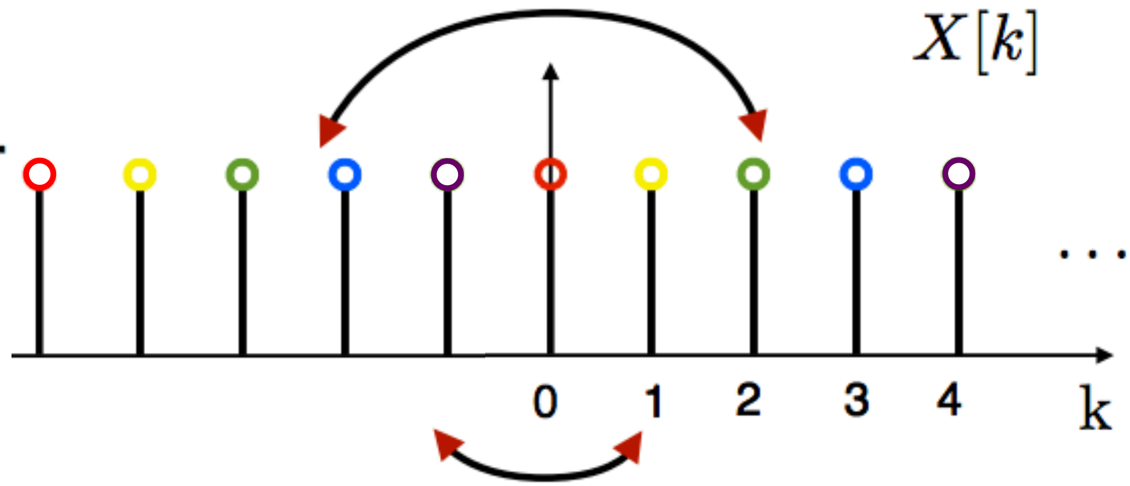


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example

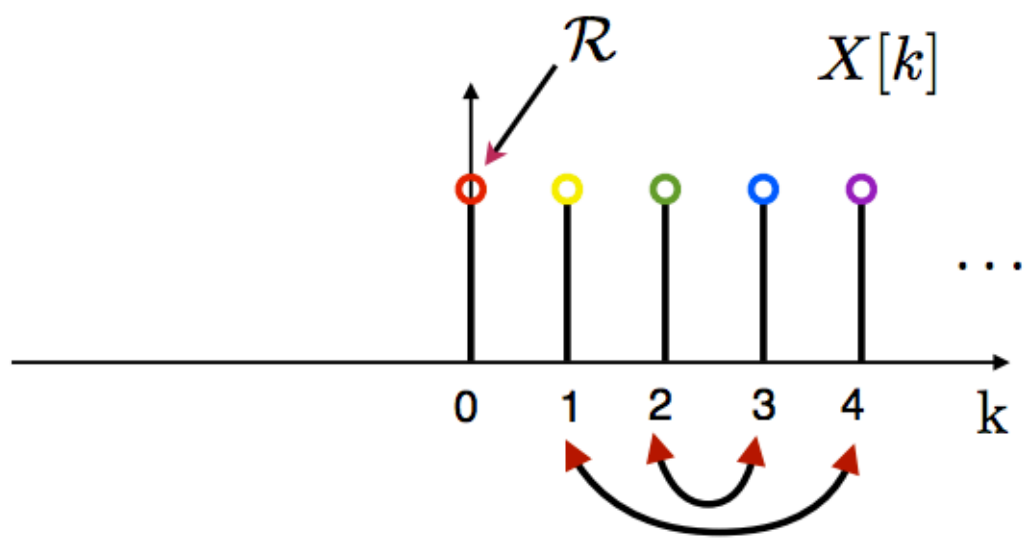
5-point DFT

–Symmetry



5-point DFT

–Symmetry



Properties of the DFS/DFT

Discrete Fourier Series			Discrete Fourier Transform		
Property	***N*-periodic sequence**	***N*-periodic DFS**	**Property**	***N*-point sequence**	***N*-point DFT**
	$\tilde{x}[n]$ $\tilde{x}_1[n],\ \tilde{x}_2[n]$	$\tilde{X}[k]$ $\tilde{X}_1[k],\ \tilde{X}_2[k]$		$x[n]$ $x_1[n],\ x_2[n]$	$X[k]$ $X_1[k],\ X_2[k]$
Linearity	$a\tilde{x}_1[n]+b\tilde{x}_2[n]$	$a\tilde{X}_1[k]+b\tilde{X}_2[k]$	Linearity	$ax_1[n]+bx_2[n]$	$aX_1[k]+bX_2[k]$
Duality	$\tilde{X}[n]$	$N\,\tilde{x}[-k]$	Duality	$X[n]$	$N\,x[((-k))_N]$
Time Shift	$\tilde{x}[n-m]$	$W_N^{km}\tilde{X}[k]$	Circular Time Shift	$x[((n-m))_N]$	$W_N^{km}X[k]$
Frequency Shift	$W_N^{-ln}\tilde{x}[n]$	$\tilde{X}[k-l]$	Circular Frequency Shift	$W_N^{-ln}x[n]$	$X[((k-l))_N]$
Periodic Convolution	$\sum_{m=0}^{N-1}\tilde{x}_1[m]\tilde{x}_2[n-m]$	$\tilde{X}_1[k]\tilde{X}_2[k]$	Circular Convolution	$\sum_{m=0}^{N-1}x_1[m]x_2[((n-m))_N]$	$X_1[k]X_2[k]$
Multiplication	$\tilde{x}_1[n]\tilde{x}_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1}\tilde{X}_1[l]\tilde{X}_2[k-l]$	Multiplication	$x_1[n]x_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1}X_1[l]X_2[((k-l))_N]$
Complex Conjugation	$\tilde{x}^*[n]$	$\tilde{X}^*[-k]$	Complex Conjugation	$x^*[n]$	$X^*[((-k))_N]$

Properties (Continued)

Time-Reversal and Complex Conjugation	$\tilde{x}^*[-n]$	$\tilde{X}^*[k]$	Time-Reversal and Complex Conjugation	$x^*[((-n))_N]$	$X^*[k]$
Real Part	$\mathrm{Re}\{\tilde{x}[n]\}$	$\tilde{X}_{ep}[k]=\frac{1}{2}\left(\tilde{X}[k]+\tilde{X}^*[-k]\right)$	Real Part	$\mathrm{Re}\{x[n]\}$	$X_{ep}[k]=\frac{1}{2}\left(X[k]+X^*[((-k))_N]\right)$
Imaginary Part	$j\,\mathrm{Im}\{\tilde{x}[n]\}$	$\tilde{X}_{op}[k]=\frac{1}{2}\left(\tilde{X}[k]-\tilde{X}^*[-k]\right)$	Imaginary Part	$j\,\mathrm{Im}\{x[n]\}$	$X_{op}[k]=\frac{1}{2}\left(X[k]-X^*[((-k))_N]\right)$
Even Part	$\tilde{x}_{ep}[n]=\frac{1}{2}\left(\tilde{x}[n]+\tilde{x}^*[-n]\right)$	$\mathrm{Re}\{\tilde{X}[k]\}$	Even Part	$x_{ep}[n]=\frac{1}{2}\left(x[n]+x^*[((-n))_N]\right)$	$\mathrm{Re}\{X[k]\}$
Odd Part	$\tilde{x}_{op}[n]=\frac{1}{2}\left(\tilde{x}[n]-\tilde{x}^*[-n]\right)$	$j\,\mathrm{Im}\{\tilde{X}[k]\}$	Odd Part	$x_{op}[n]=\frac{1}{2}\left(x[n]-x^*[((-n))_N]\right)$	$j\,\mathrm{Im}\{X[k]\}$
Symmetry for Real Sequence	$\tilde{x}[n]=\tilde{x}^*[n]$	$\tilde{X}[k]=\tilde{X}^*[-k]$ $\begin{cases}\mathrm{Re}\{\tilde{X}[k]\}=\mathrm{Re}\{\tilde{X}[-k]\}\\ \mathrm{Im}\{\tilde{X}[k]\}=-\mathrm{Im}\{\tilde{X}[-k]\}\end{cases}$ $\begin{cases}\lvert\tilde{X}[k]\rvert=\lvert\tilde{X}[-k]\rvert\\ \angle\tilde{X}[k]=-\angle\tilde{X}[-k]\end{cases}$	Symmetry for Real Sequence	$x[n]=x^*[n]$	$X[k]=X^*[((-k))_N]$ $\begin{cases}\mathrm{Re}\{X[k]\}=\mathrm{Re}\{X[((-k))_N]\}\\ \mathrm{Im}\{X[k]\}=-\mathrm{Im}\{X[((-k))_N]\}\end{cases}$ $\begin{cases}\lvert X[k]\rvert=\lvert X[((-k))_N]\rvert\\ \angle X[k]=-\angle X[((-k))_N]\end{cases}$
Parseval's Identity	$\sum_{n=0}^{N-1}\tilde{x}_1[n]\tilde{x}_2^*[n]=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_1[k]\tilde{X}_2^*[k]$ $\sum_{n=0}^{N-1}\lvert\tilde{x}[n]\rvert^2=\frac{1}{N}\sum_{k=0}^{N-1}\lvert\tilde{X}[k]\rvert^2$		Parseval's Identity	$\sum_{n=0}^{N-1}x_1[n]x_2^*[n]=\frac{1}{N}\sum_{k=0}^{N-1}X_1[k]X_2^*[k]$ $\sum_{n=0}^{N-1}\lvert x[n]\rvert^2=\frac{1}{N}\sum_{k=0}^{N-1}\lvert X[k]\rvert^2$	

Circular Convolution

- Circular Convolution:

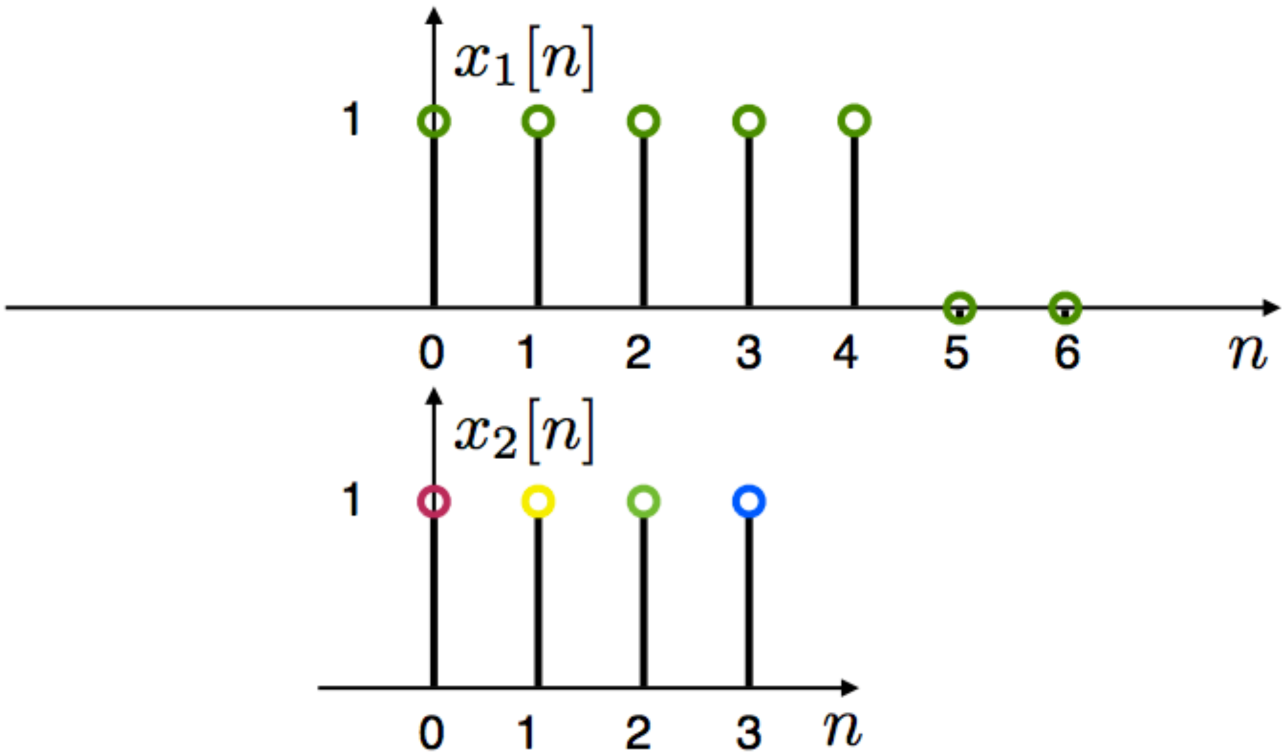
$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

For two signals of length N

Note: Circular convolution is commutative

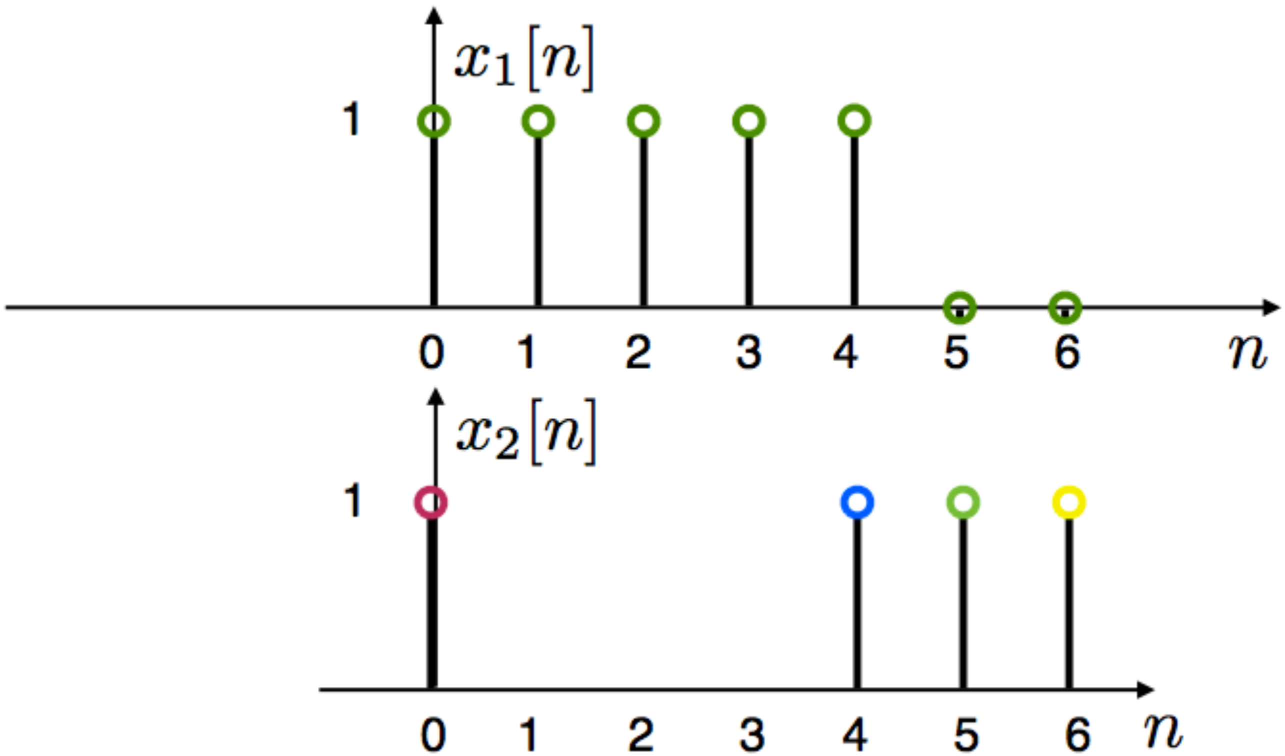
$$x_2[n] \circledN x_1[n] = x_1[n] \circledN x_2[n]$$

Compute Circular Convolution Sum



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

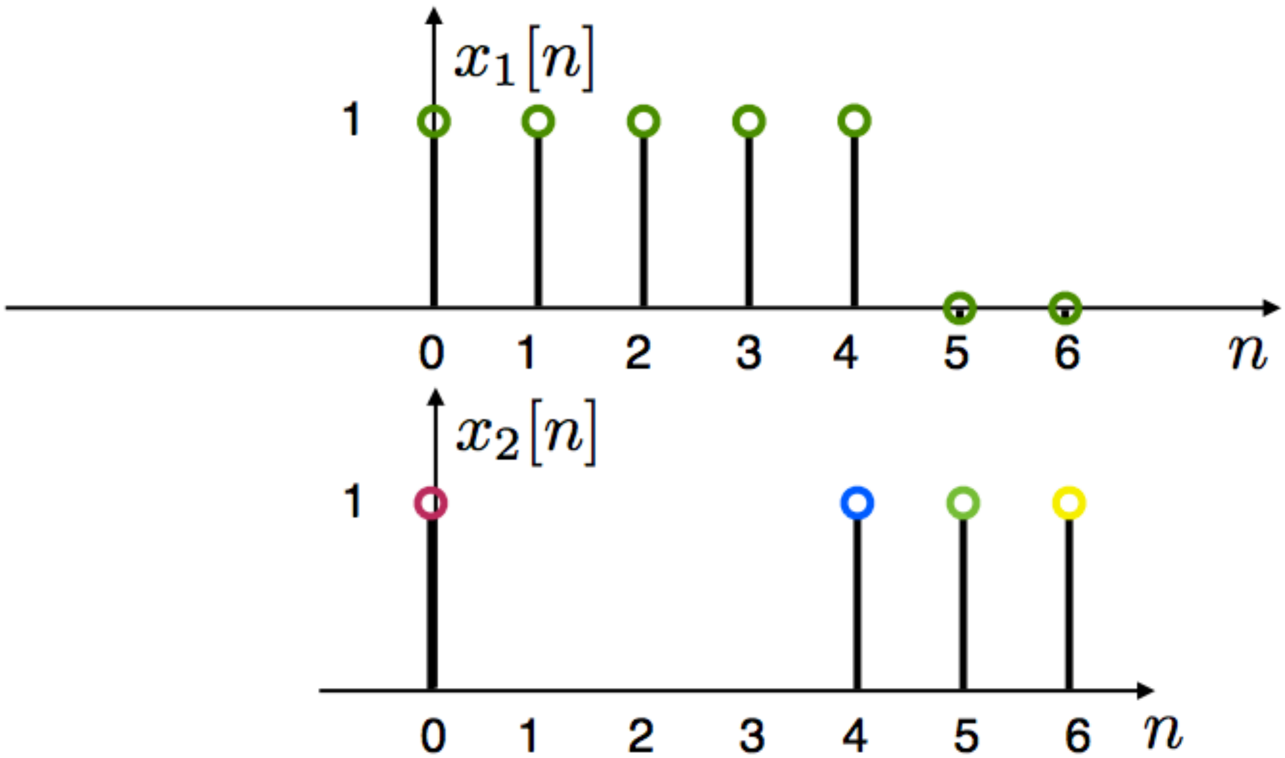
Compute Circular Convolution Sum



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

Compute Circular Convolution Sum

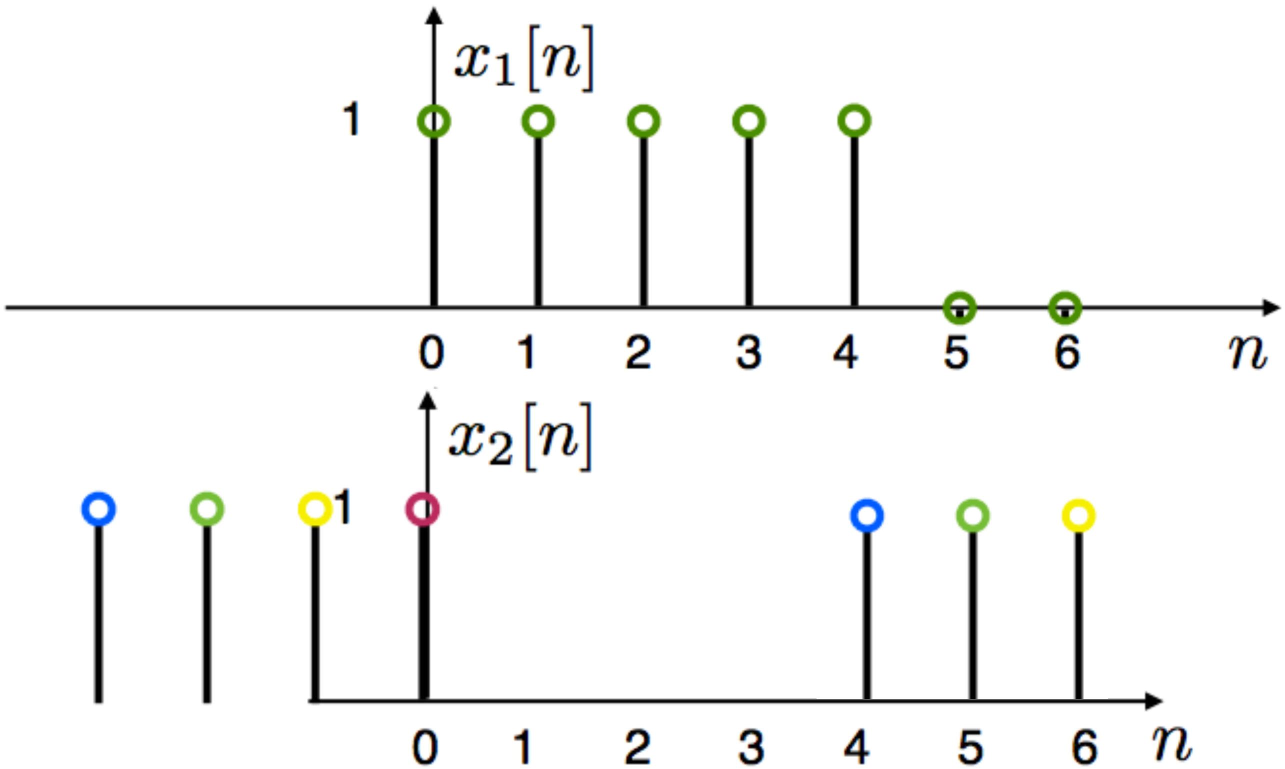
y[0]=2



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

Compute Circular Convolution Sum

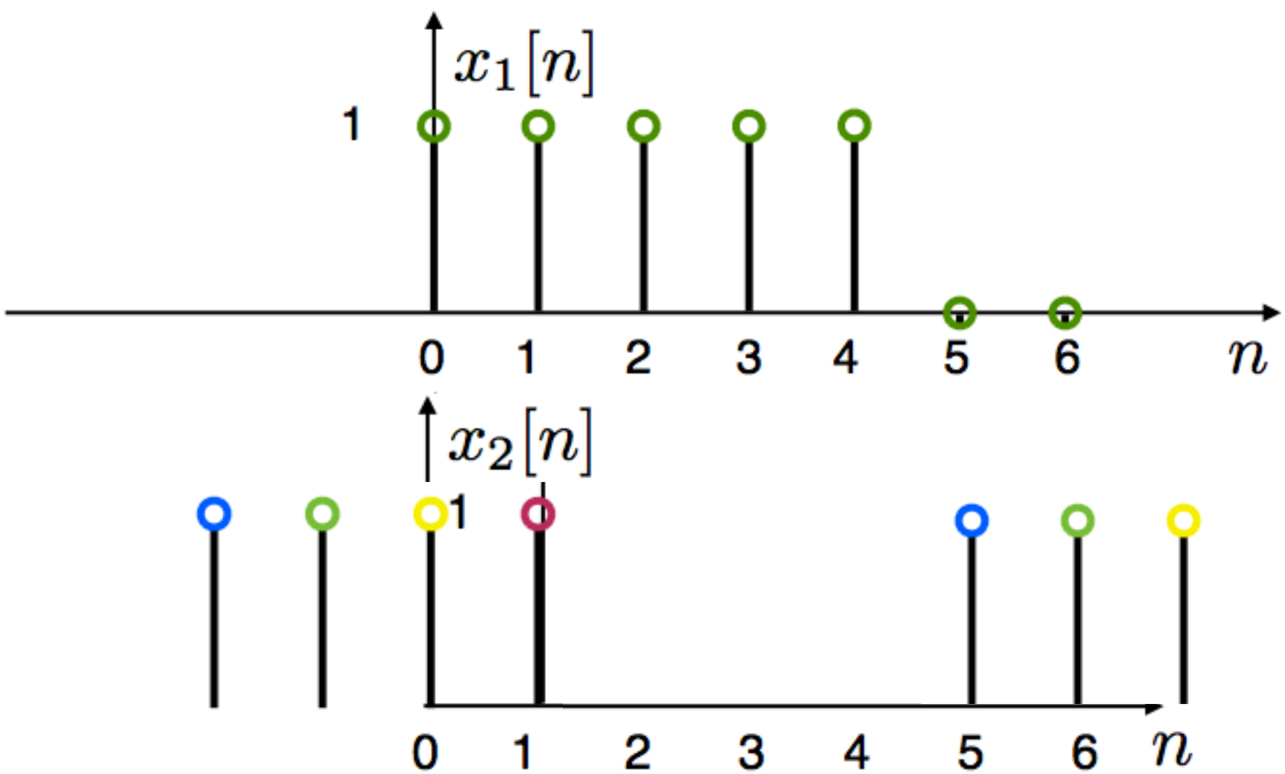
y[0]=2



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

Compute Circular Convolution Sum

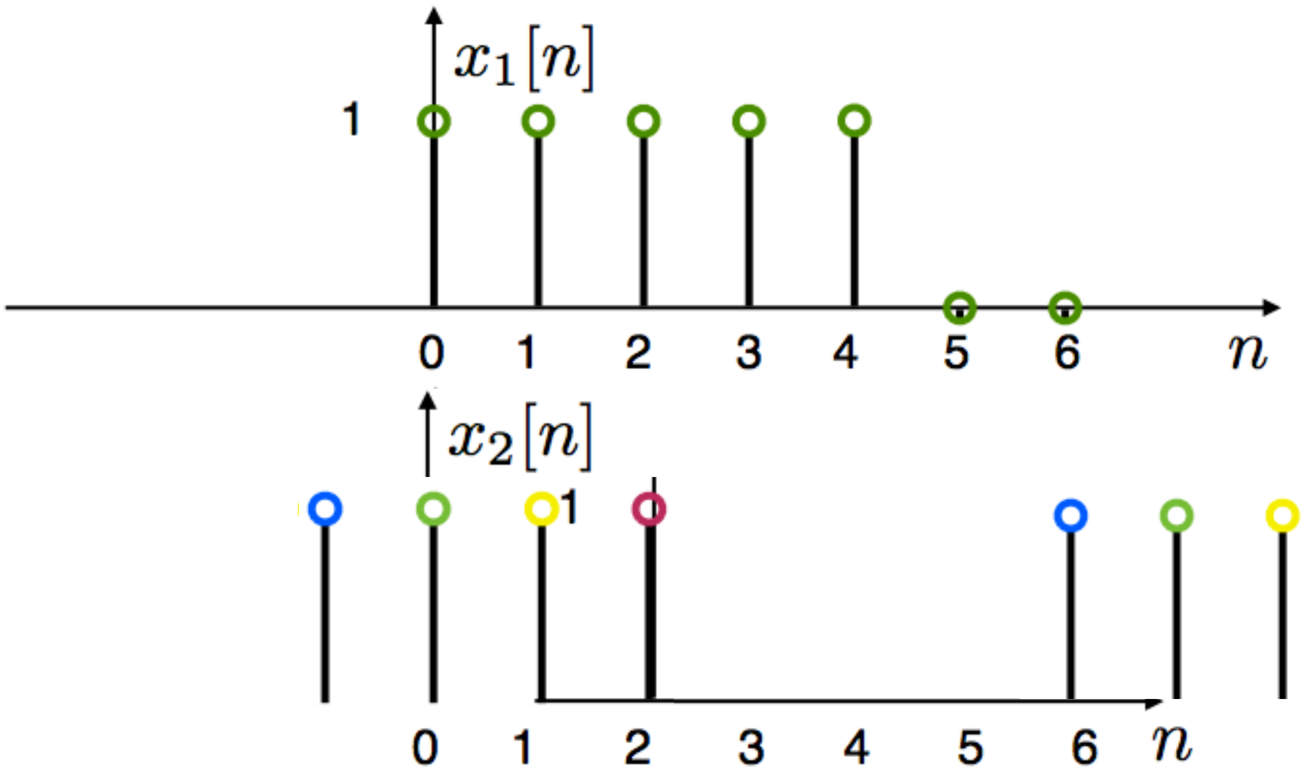
y[0]=2
y[1]=2



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

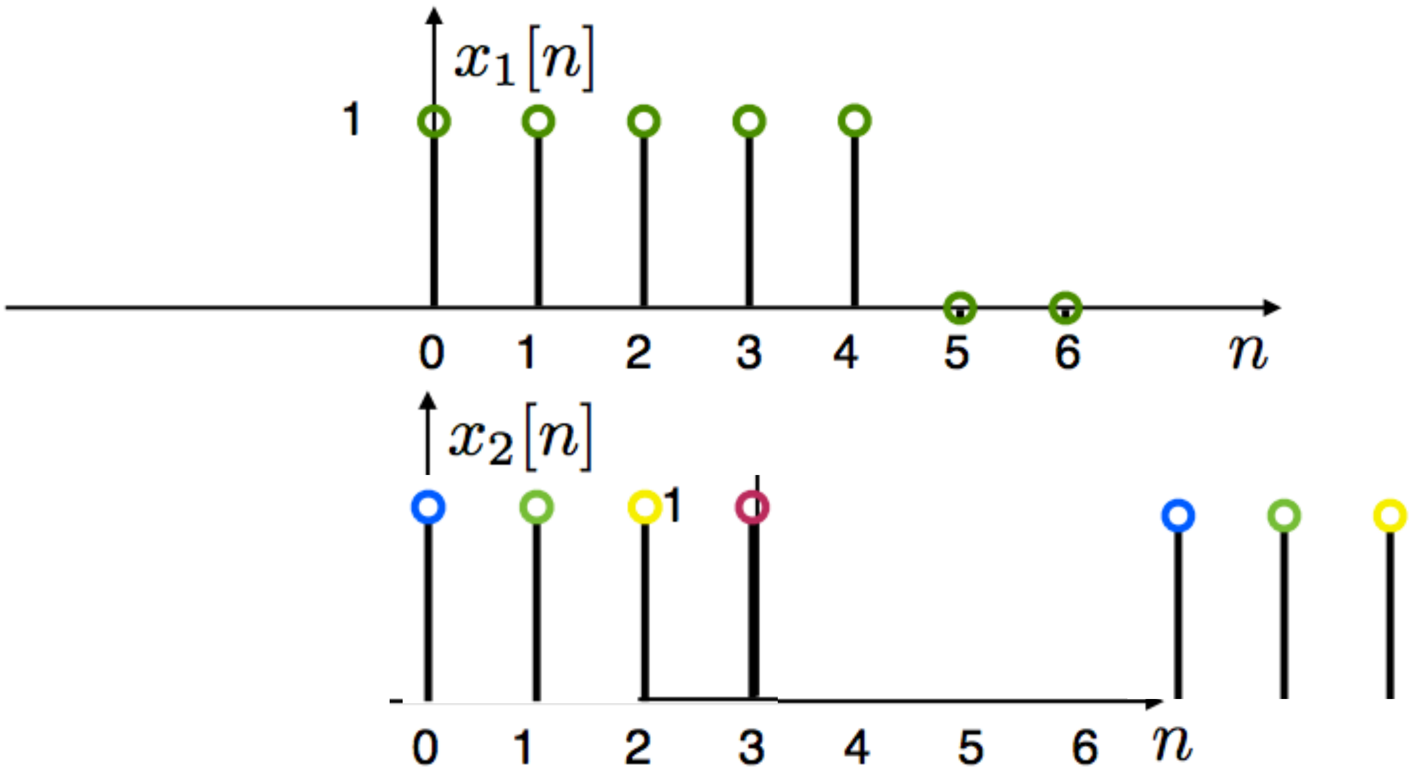
Compute Circular Convolution Sum

y[0]=2
y[1]=2
y[2]=3



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

Compute Circular Convolution Sum

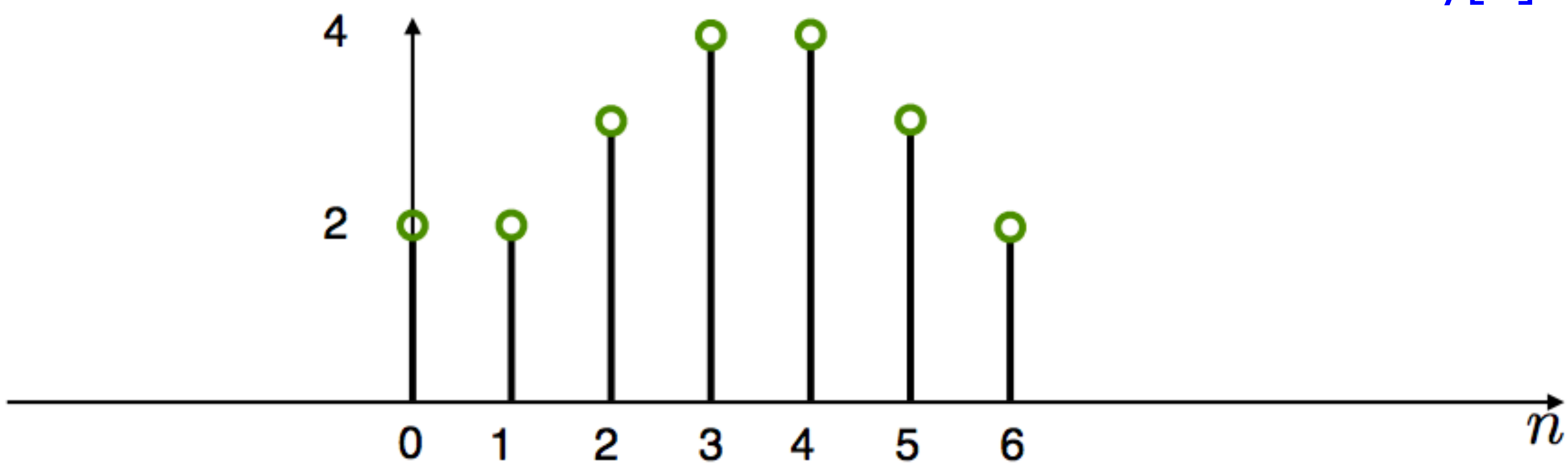


y[0]=2
y[1]=2
y[2]=3
y[3]=4

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

Result

y[0]=2
y[1]=2
y[2]=3
y[3]=4



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

Circular Convolution

- For $x_1[n]$ and $x_2[n]$ with length N

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

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

Multiplication

- For $x_1[n]$ and $x_2[n]$ with length N

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

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
  - Use DFT to do linear convolution

Big Ideas

- Discrete Fourier Transform (DFT)
  - For finite signals assumed to be zero outside of defined length
  - N-point DFT is sampled DTFT at N points
  - Useful properties allow easier linear convolution
- DFT Properties
  - Inherited from DFS, but circular operations!

Admin

- Project 1
  - Due Tuesday 3/29
- HW 7 out on Tuesday 3/29
- No lecture on Tuesday 3/29
- Tania OH tomorrow shifted to Monday
  - Same time, same link

Herman P. Schwan Distinguished Lecture: "Nucleoside-modified mRNA LNP Therapeutics"

Drew Weissman, M.D., Ph.D.
Roberts Family Professor in Vaccine Research
Department of Medicine
Perelman School of Medicine
University of Pennsylvania

Tuesday, March 29, 2022
3:30-5:00 PM
Reception to follow