ESE 531: Digital Signal Processing

Lec 19: April 6, 2017

Discrete Fourier Transform

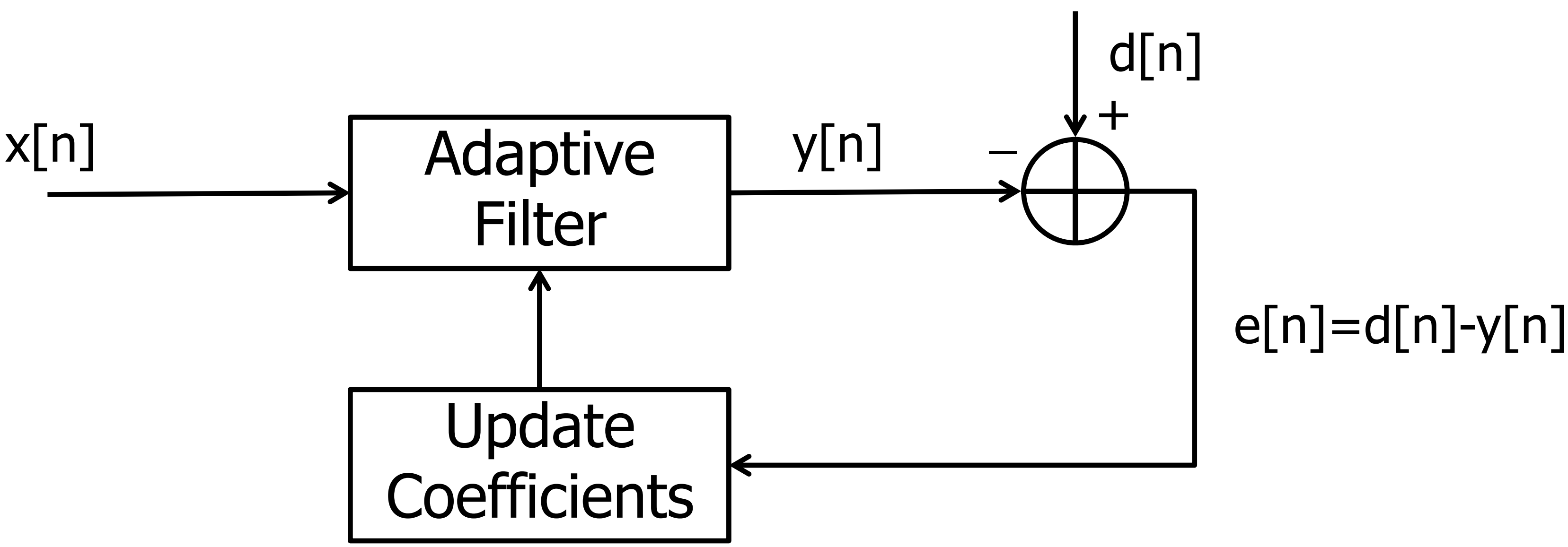
Penn ESE 531 Spring 2017 – Khanna
Adapted from M. Lustig, EECS Berkeley

Today

- Adaptive filtering
  - Blind equalization
- Discrete Fourier Series
- Discrete Fourier Transform (DFT)
- DFT Properties

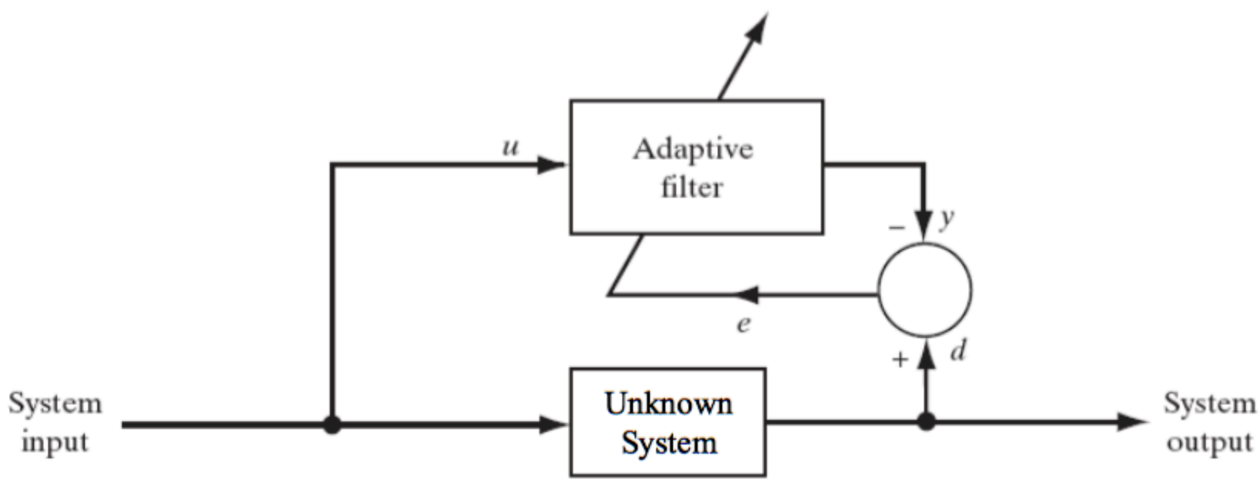
Adaptive Filters

- An adaptive filter is an adjustable filter that processes in time
  - It adapts…



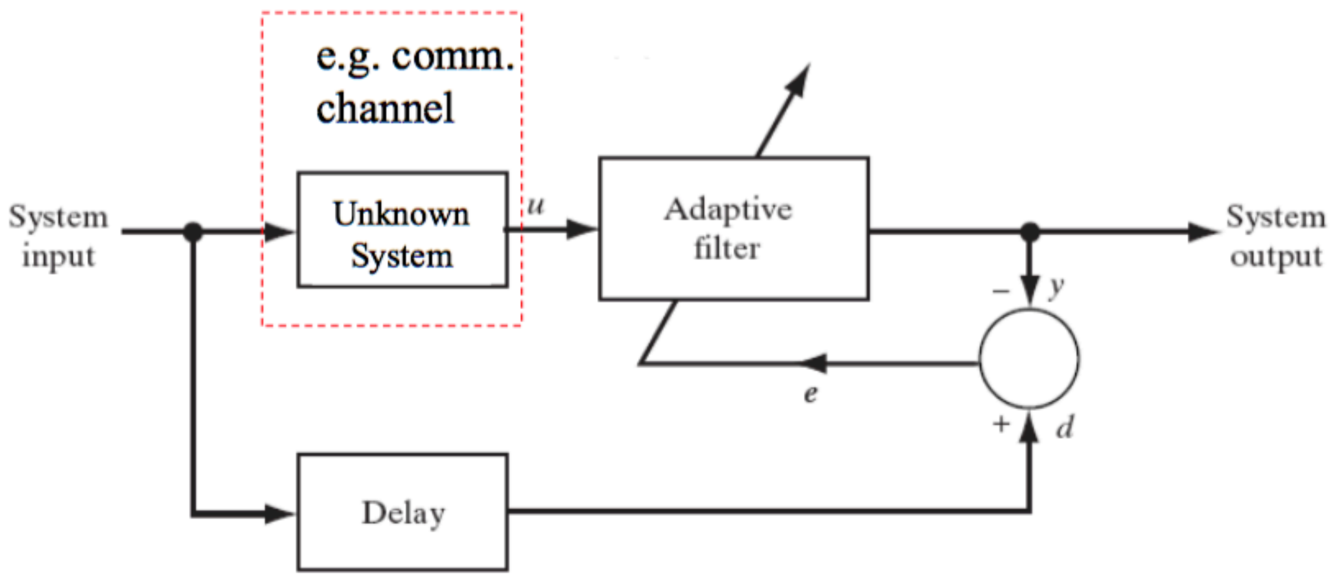
Adaptive Filter Applications

- System Identification



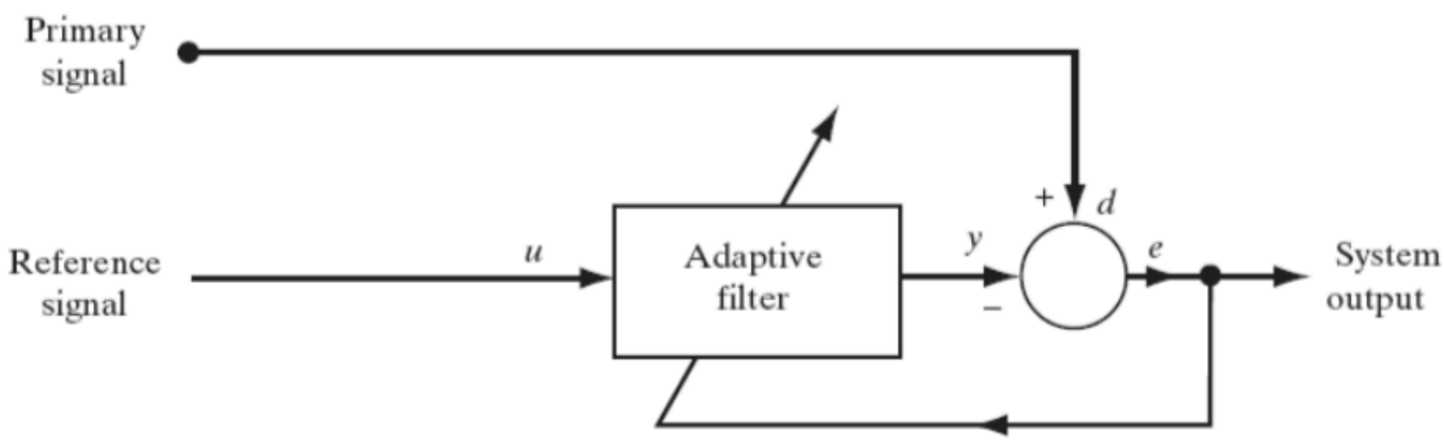
Adaptive Filter Applications

- Identification of inverse system



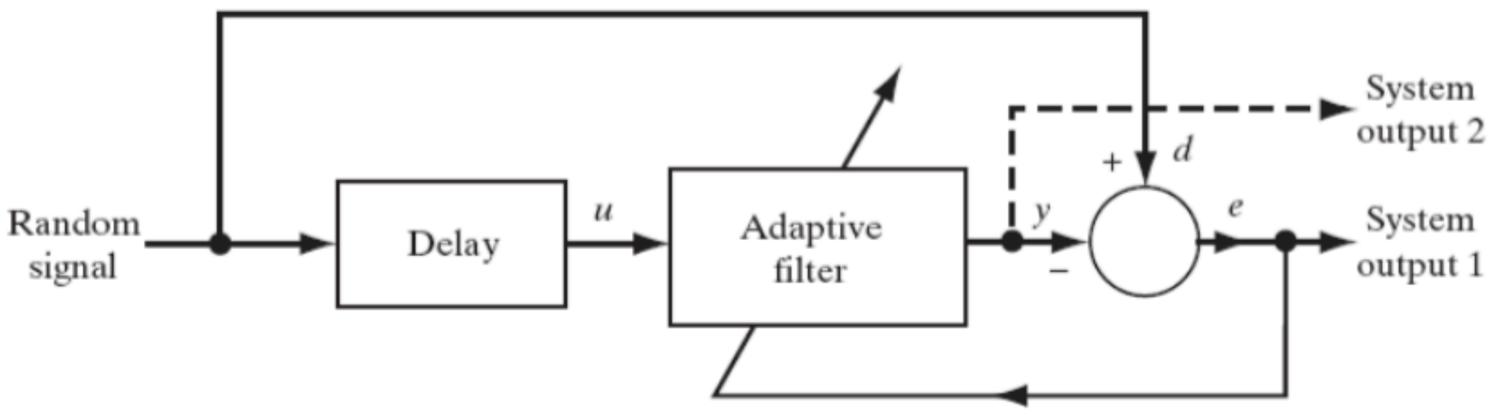
Adaptive Filter Applications

- Adaptive Interference Cancellation



Adaptive Filter Applications

- Adaptive Prediction

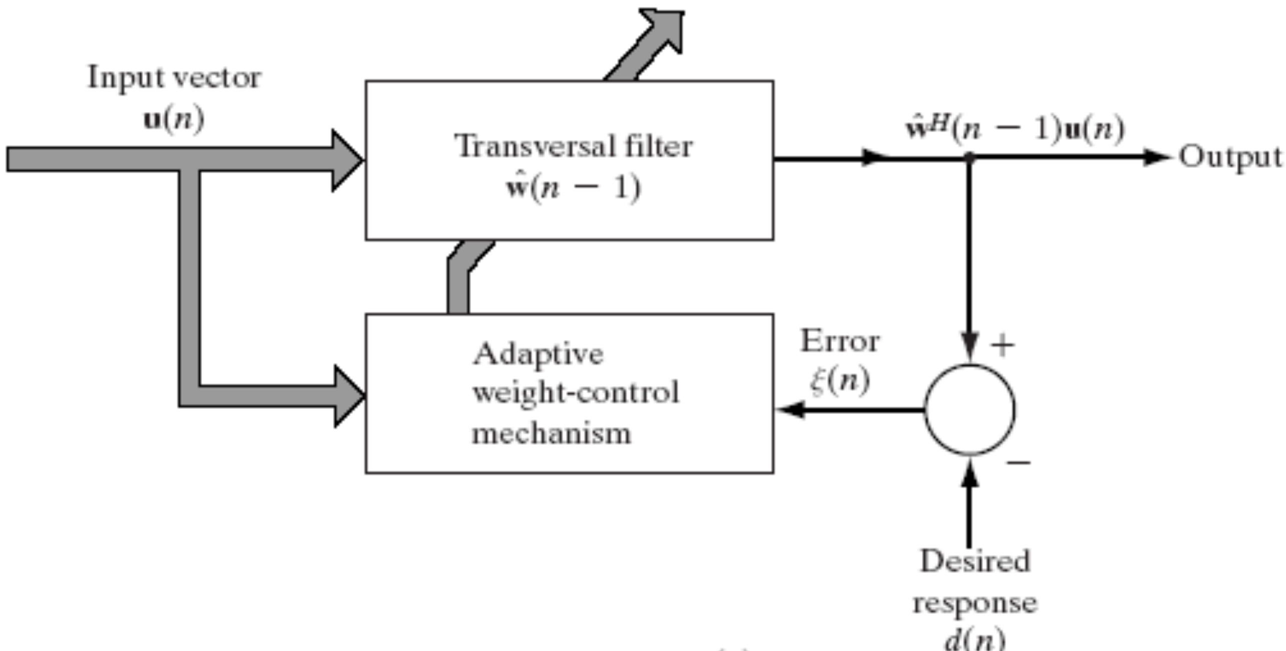


Stochastic Gradient Approach

- Most commonly used type of Adaptive Filters
- Define cost function as mean-squared error
    - Difference between filter output and desired response
- Based on the method of steepest descent
    - Move towards the minimum on the error surface to get to minimum
    - Requires the gradient of the error surface to be known

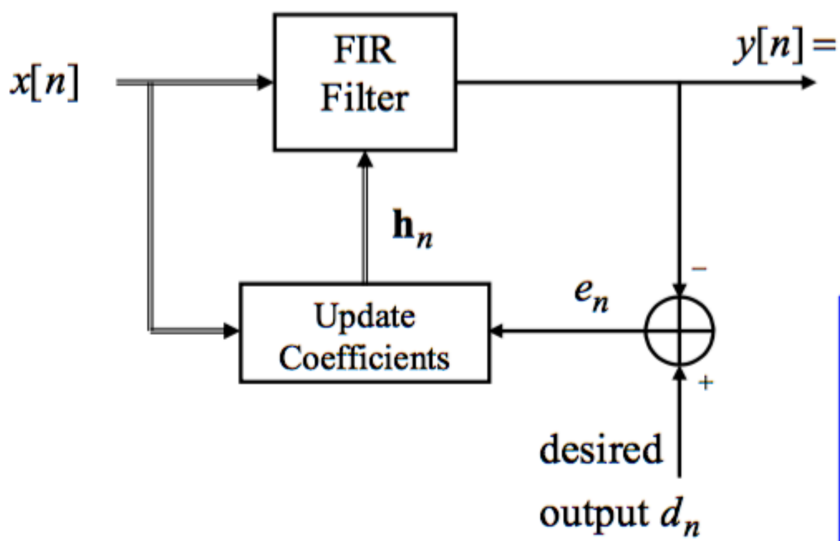
Least-Mean-Square (LMS) Algorithm

- The LMS Algorithm consists of two basic processes
  - Filtering process
    - Calculate the output of FIR filter by convolving input and taps
    - Calculate estimation error by comparing the output to desired signal
  -



(a)

Adaptive FIR Filter: LMS



$$y[n] = \mathbf{h}_n^T \mathbf{x}_n$$

$$\mathbf{x}_n = (x[n], x[n-1] \ldots, x[n-M])^T$$

$$\mathbf{h}_n = (h_n[0], h_n[1], \ldots, h_n[M])^T$$

$$e_n^{\,2} = (d[n] - y[n])^2 = (d[n] - \mathbf{h}_n^T \mathbf{x}_n)^2$$

$$\frac{d\, e_n^{\,2}}{d\, \mathbf{h}_n} = -2(d[n] - \mathbf{h}_n^T \mathbf{x}_n)\mathbf{x}_n = -2 e_n \mathbf{x}_n$$

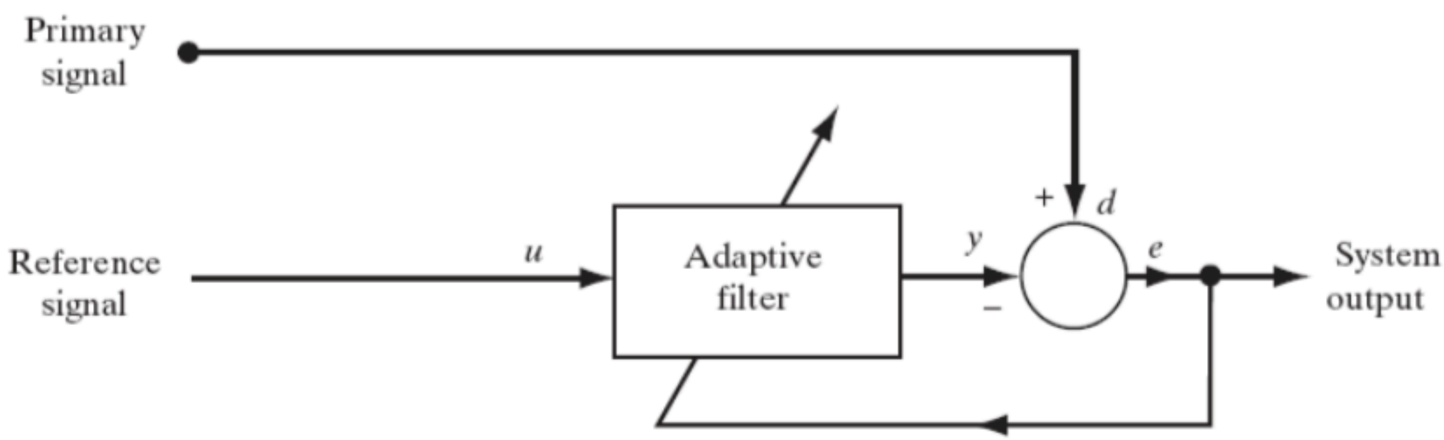
Coefficient Update: Move in direction *opposite* to sign of gradient, proportional to magnitude of gradient

$$\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu e_n \mathbf{x}_n$$

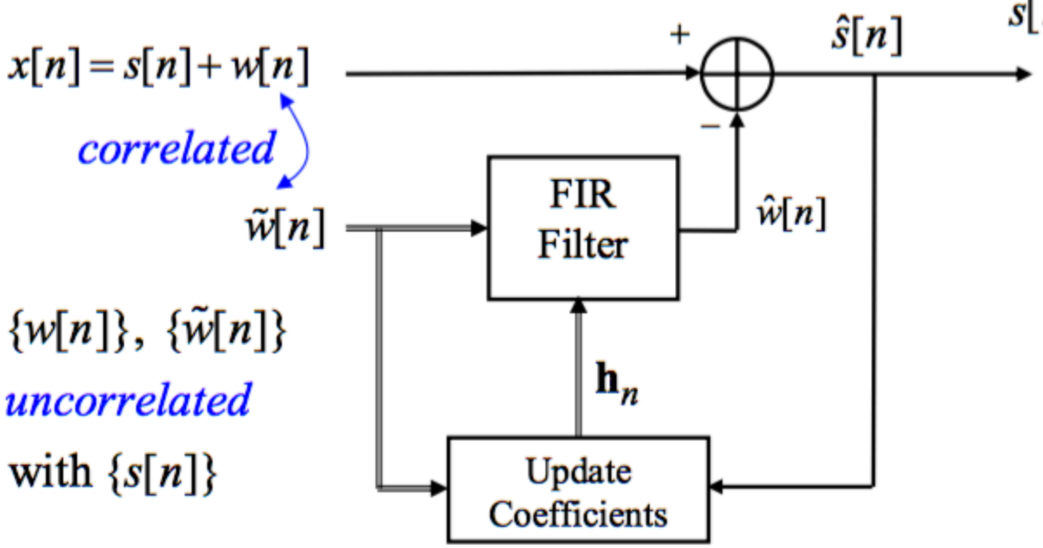
Stochastic Gradient Algorithm

Adaptive Filter Applications

- Adaptive Interference Cancellation



Adaptive Interference Cancellation



$$\hat{s}[n] = x[n] - \mathbf{h}_n^T \tilde{\mathbf{w}}_n$$

$$= s[n] + \{w[n] - \mathbf{h}_n^T \tilde{\mathbf{w}}_n\}$$

uncorrelated

Minimize *instantaneous power* $\hat{s}^2[n]$ for best interference cancellation

$$\frac{d\left(\hat{s}[n]\right)^2}{d\,\mathbf{h}_n} = -2\hat{s}[n]\,\tilde{\mathbf{w}}_n$$

$$\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu\,\hat{s}[n]\,\tilde{\mathbf{w}}_n}$$

Stability of LMS

- The LMS algorithm is convergent in the mean square if and only if the step-size parameter satisfy

$$0 < \mu < \frac{2}{\lambda_{max}}$$

- Here λ_{max} is the largest eigenvalue of the correlation matrix of the input data
- More practical test for stability is

$$0 < \mu < \frac{2}{\text{input signal power}}$$

- Larger values for step size
  - Increases adaptation rate (faster adaptation)
  - Increases residual mean-squared error

Discrete Fourier Series

Reminder: Eigenvalue (DTFT)

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

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

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

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

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

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

- 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:

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

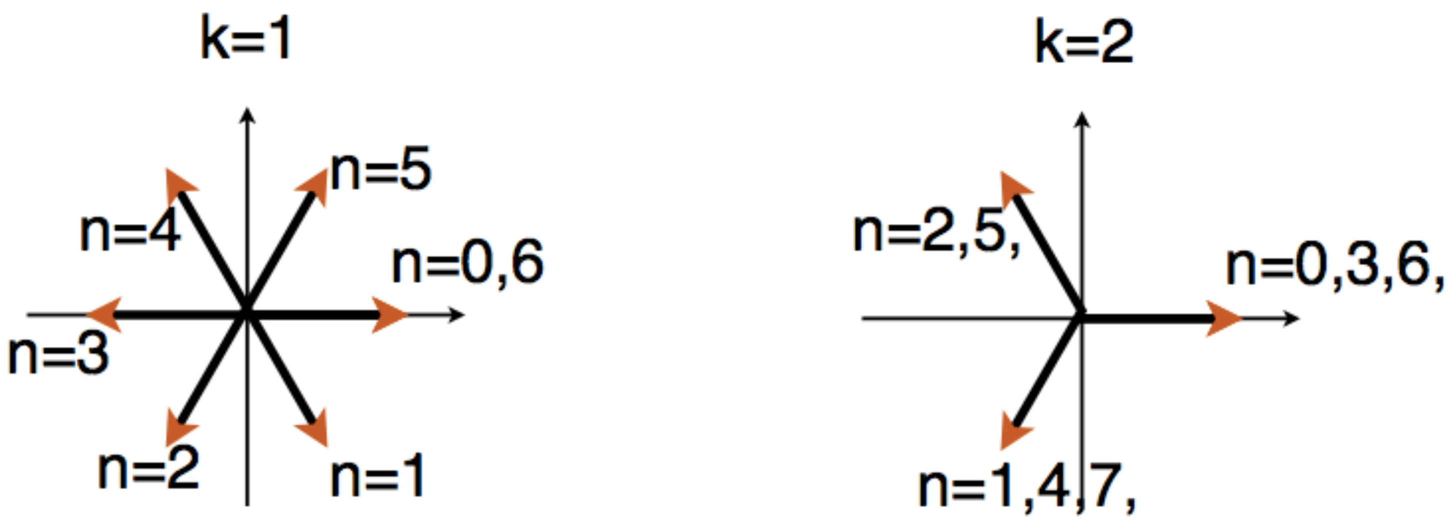
$$\tilde{X}[k] = \sum_{n=0}^{N-1}\tilde{x}[n]W_N^{kn}$$

Discrete Fourier Series

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=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

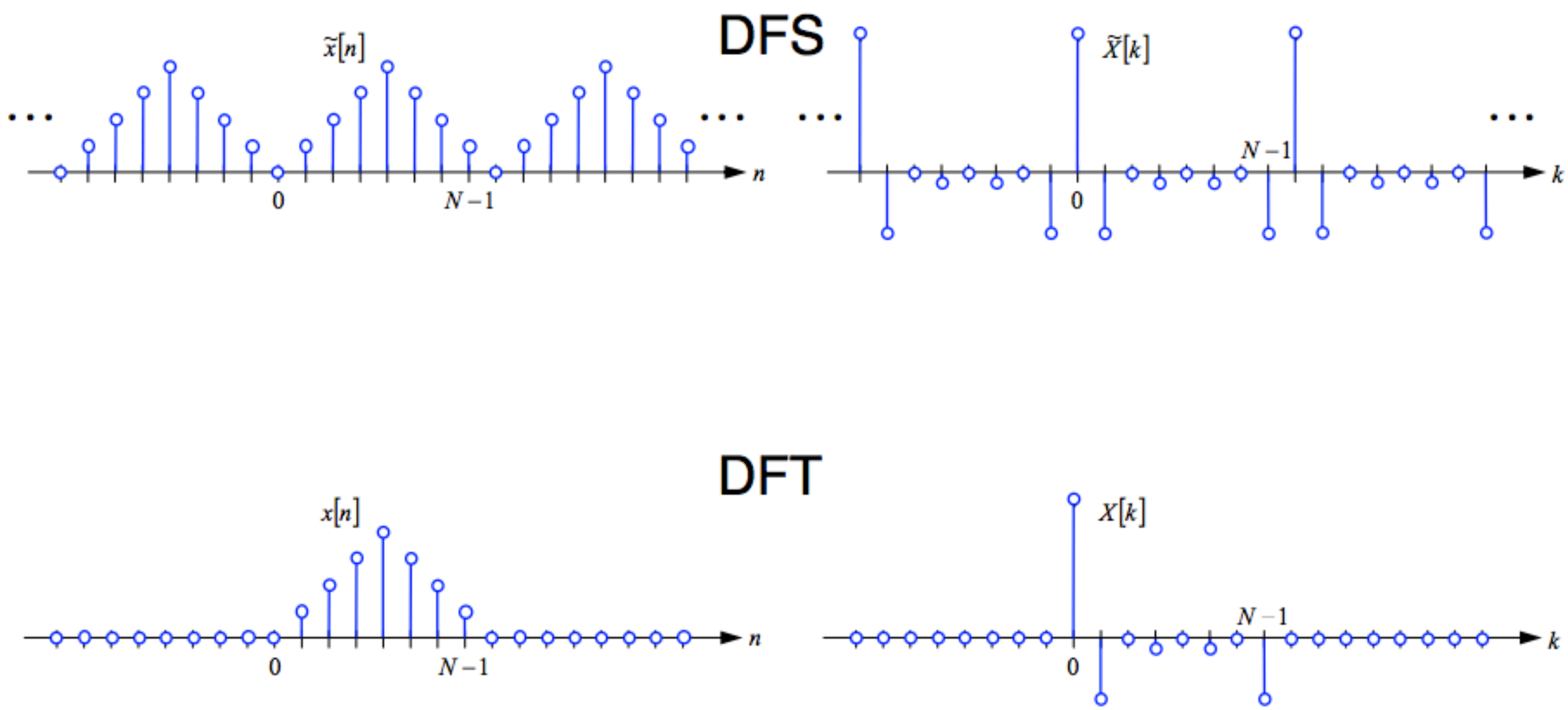
$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-kn}$$ **Inverse DFT, synthesis**

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

- It is understood that,

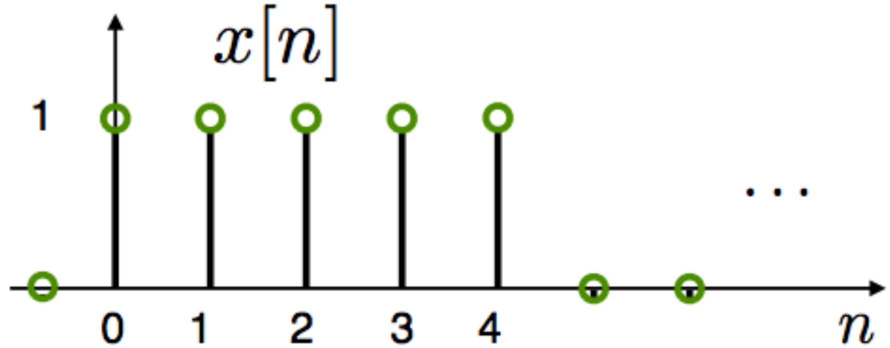
$$x[n] = 0 \quad \text{outside } 0 \le n \le N-1$$
$$X[k] = 0 \quad \text{outside } 0 \le k \le N-1$$

DFS vs. DFT



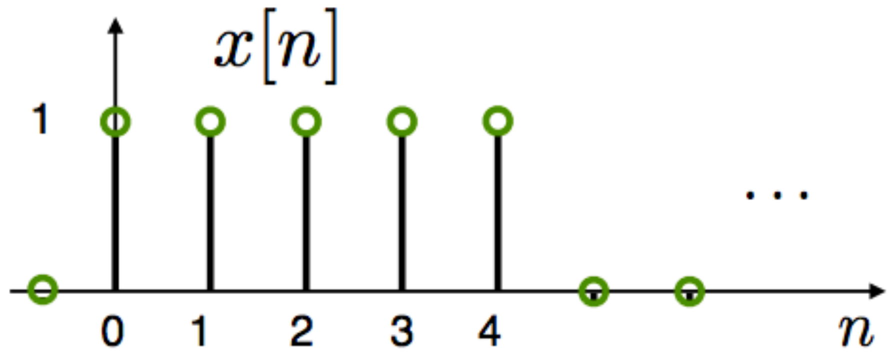
Example

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



Example

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



- Take N=5

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

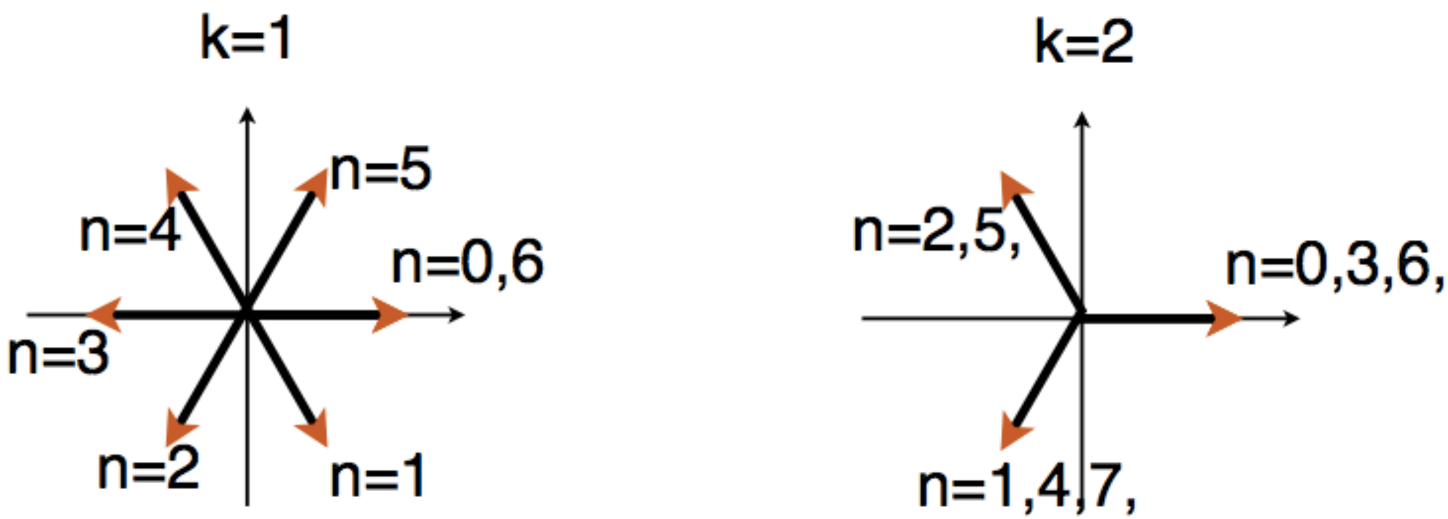
$$= 5\delta[k]$$

"5-point DFT"

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=6)



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

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

Example

- Now, sum from n=0 to 9

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{9} W_{10}^{nk} \\
&= \sum_{n=0}^{4} W_{10}^{nk} \\
&= e^{-j\frac{4\pi}{10}k}\frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}
\end{aligned}
$$

"10-point DFT"

DFT vs. DTFT

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

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

  - 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

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

DFT vs DTFT

- Back to example

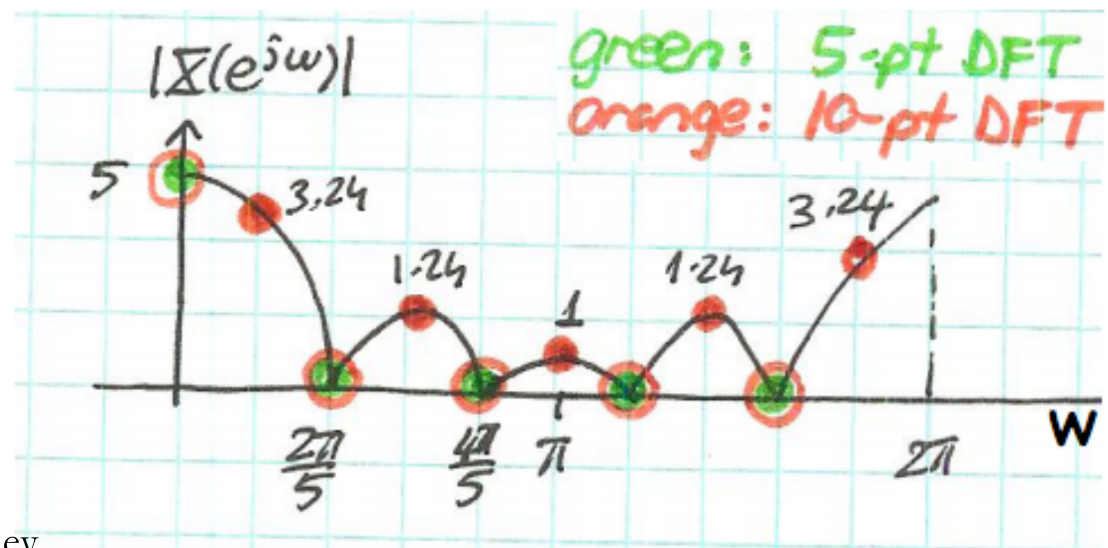
$$X[k] = \sum_{n=0}^{4} W_{10}^{nk}$$
$$= e^{-j\frac{4\pi}{10}k} \frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$

"10-point DFT"

DFT vs DTFT

- Back to example

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



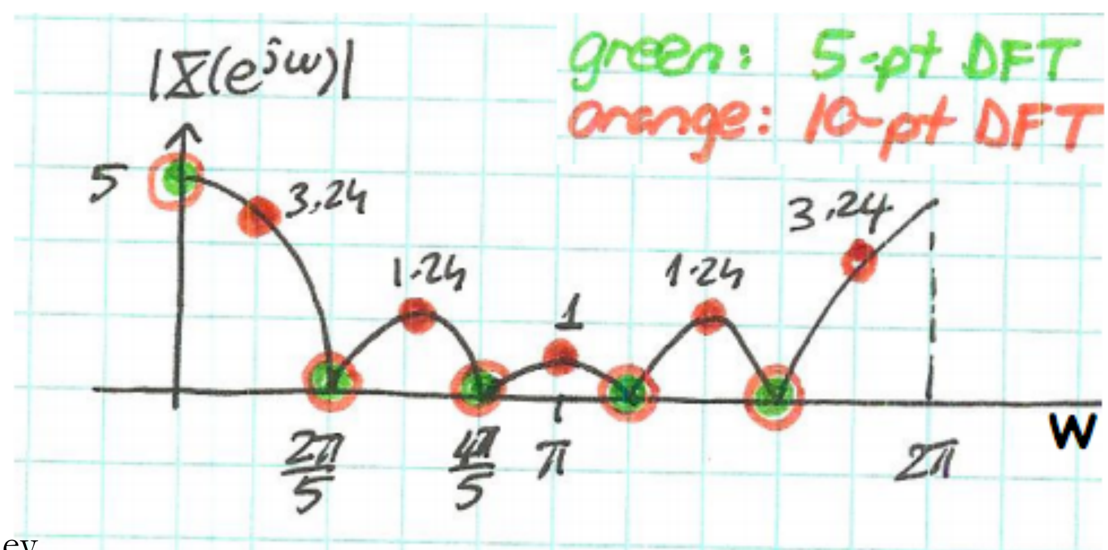
DFT vs DTFT

- 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?

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

DFT and Inverse DFT

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

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

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

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

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

- 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 multiples

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

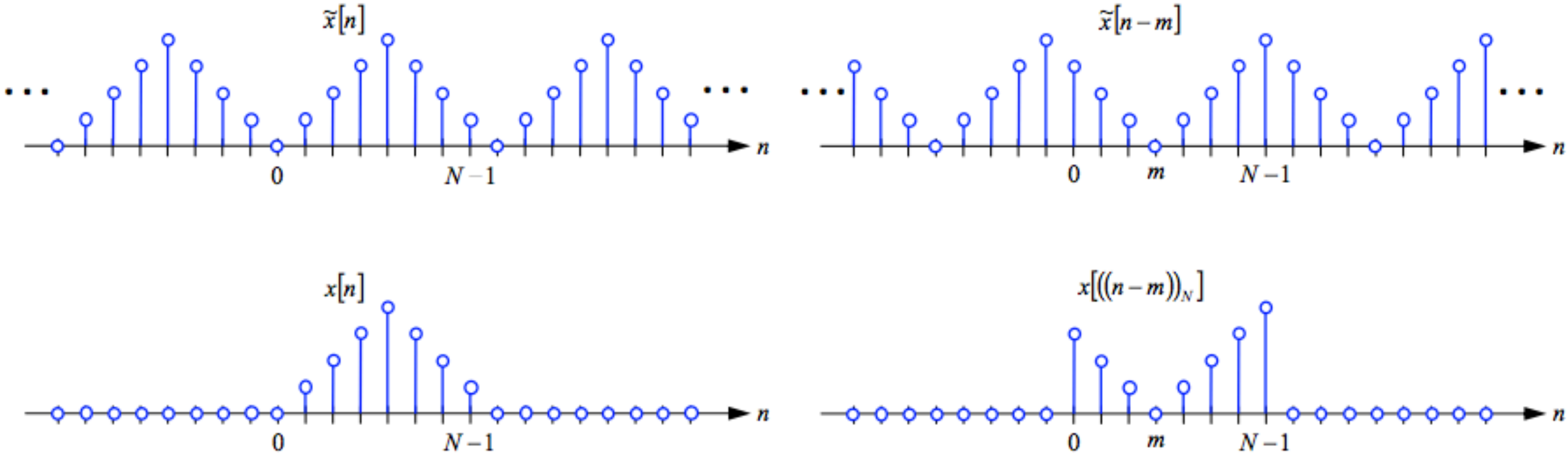
- Properties of DFT inherited from DFS
- Linearity

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

- Circular Time Shift

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

Circular Shift



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

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

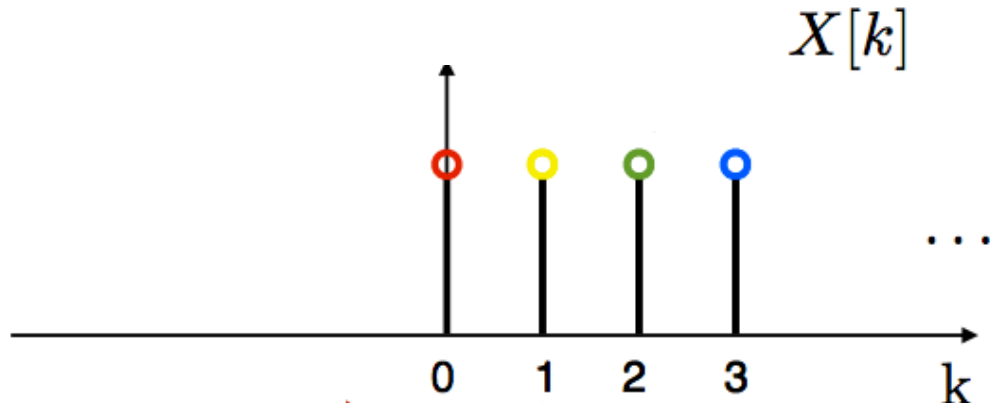
- Conjugate Symmetry for Real Signals

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

Example: Conjugate Symmetry

4-point DFT

– Symmetry

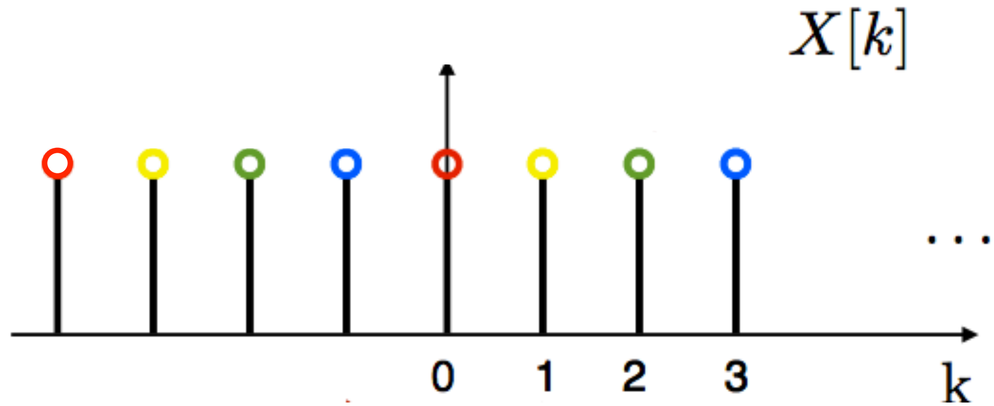


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

–Symmetry

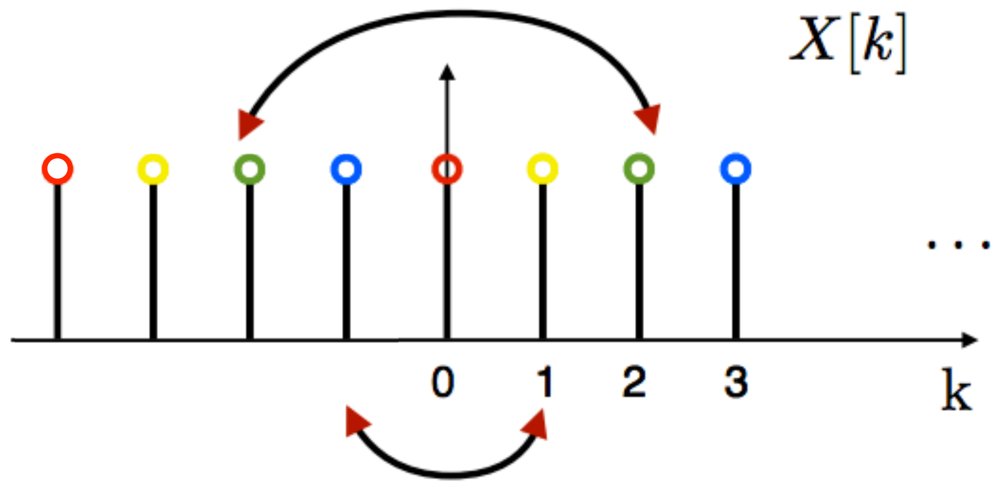


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

–Symmetry

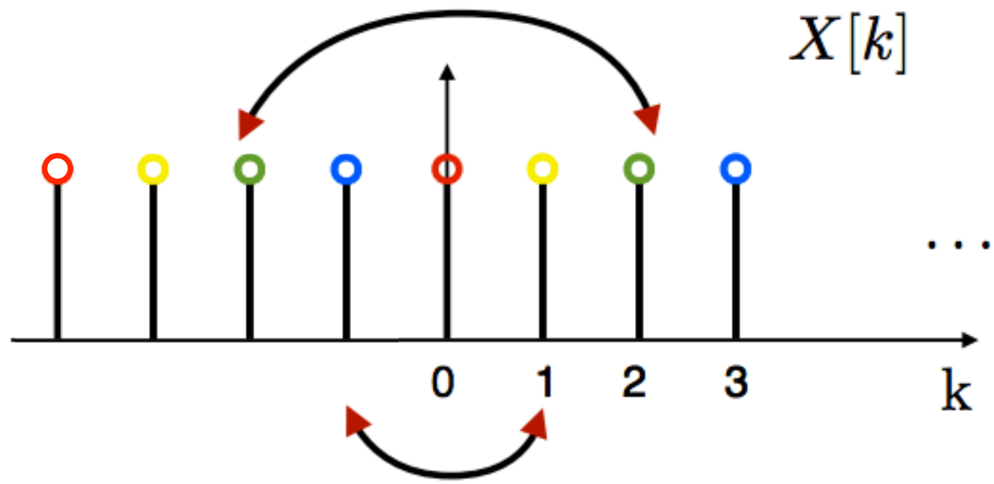


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

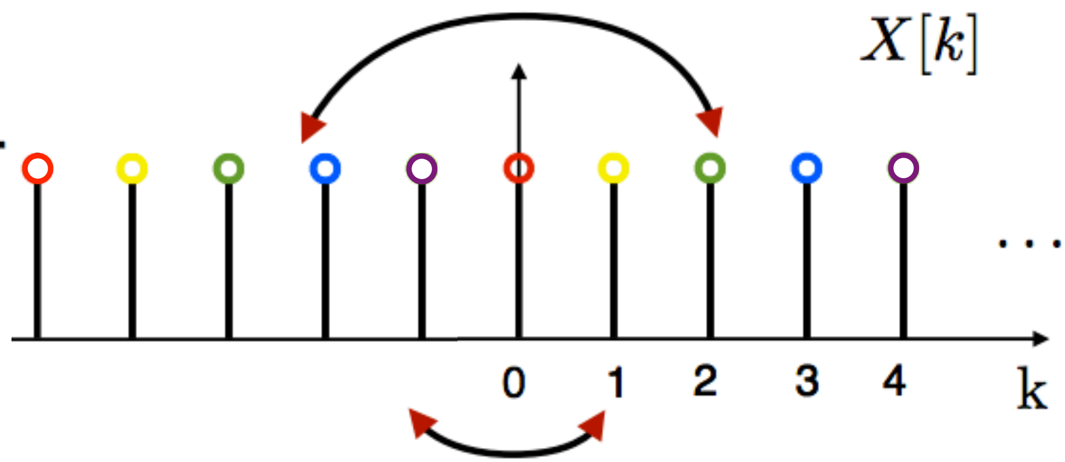
4-point DFT

–Symmetry



5-point DFT

–Symmetry

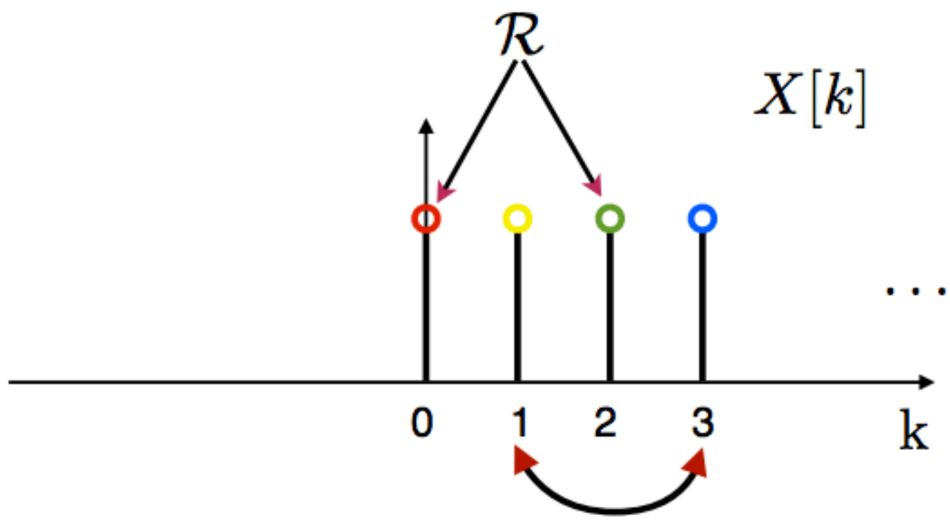


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example

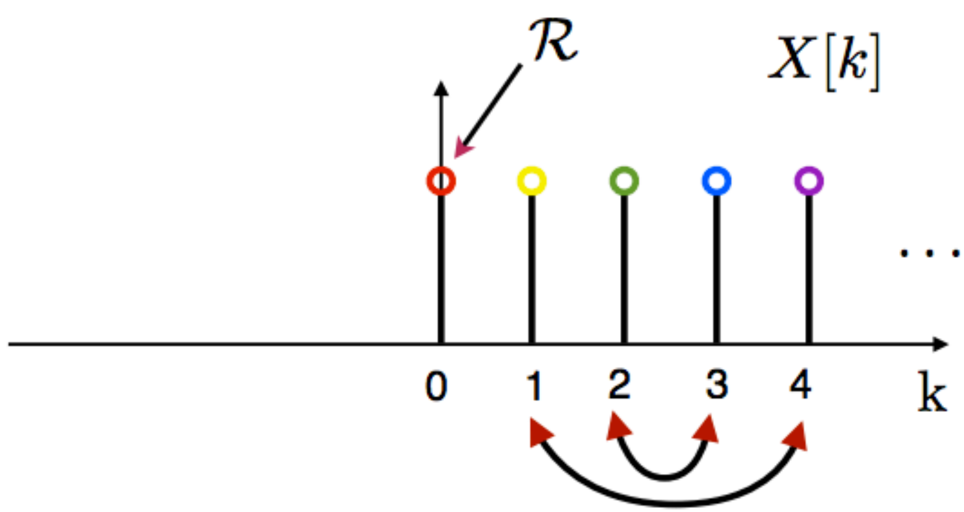
4-point DFT

–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

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

Circular Convolution

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

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

Multiplication

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

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

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
  - Use DFT to do linear convolution

Big Ideas

- Adaptive filtering
  - Use LMS algorithm to update filter coefficients for applications like system ID, channel equalization, and signal prediction
- 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

Admin

- HW 7 out now
  - Due tonight

- Project posted after class tonight
  - Work in groups of up to 2
    - Can work alone if you want
    - Use Piazza to find partners
    - Report your groups to me by 4/11 by email
      - taniak@seas.upenn.edu