EE123
Digital Signal Processing

Lecture 5

based on slides by J.M. Kahn

M. Lustig, EECS UC Berkeley

Info

- Last time
  - Finished DTFT Ch. 2
  - z-Transforms Ch. 3
- Today: DFT Ch. 8

- Reminders:
  - HW Due tonight

The effects of sampling

What is going on here?

Rolling shutter effect

https://www.youtube.com/watch?v=bzjwnZDScxo

Motivation: Discrete Fourier Transform

- Sampled Representation in time and frequency
  - Numerical Fourier Analysis requires discrete representation
  - But, sampling in one domain corresponds to periodicity in the other...
  - What about DFS (DFT)?
    - Periodic in “time” ✓
    - Periodic in “Frequency” ✓
  - What about non-periodic signals?
    - Still use DFS(T), but need special considerations

Which transform? (Time-domain)

Continuity

Periodicity

	Discrete	Continuous
Periodic	DFT, DFS	CFS
Aperiodic	DTFT	CTFT

Sometimes we use this for aperiodic signals, though! (→ DFT, DFS)

Time		Frequency
Periodic	←→	Discrete
Discrete	←→	Periodic

Motivation: Discrete Fourier Transform

- Efficient Implementations exist
  - Direct evaluation of DFT: $O(N^2)$
  - Fast Fourier Transform (FFT): O(N log N)
    (ch. 9, next topic....)
  - Efficient libraries exist: FFTW
    - In Python:
      ```
      > X = np.fft.fft(x);
      > x = np.fft.ifft(X);
      ```
  - Convolution can be implemented efficiently using FFT
    - Direct convolution: $O(N^2)$
    - FFT-based convolution: O(N log N)

Discrete Fourier Series (DFS)

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

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

Discrete Fourier Series (DFS)

- Define:

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

- DFS:

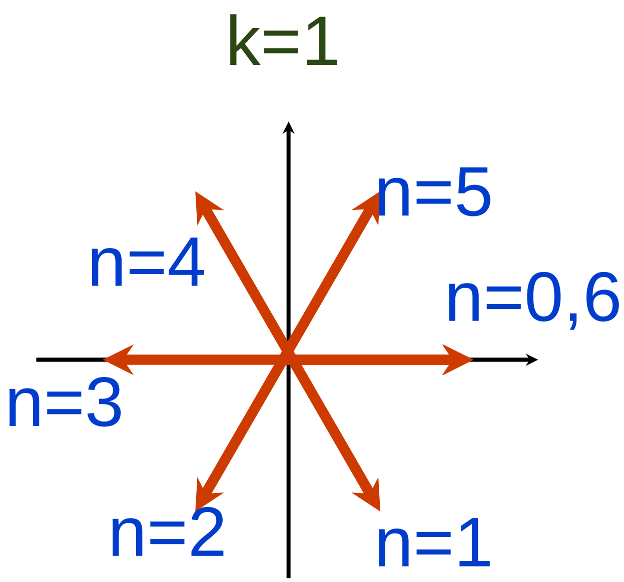
$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}[k]W_N^{-kn}$$

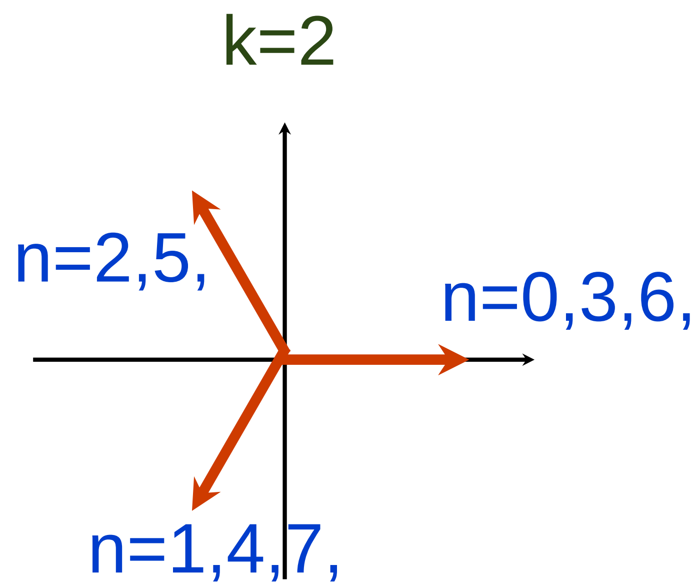
$$\tilde{X}[k] = \sum_{n=0}^{N-1}\tilde{x}[n]W_N^{kn}$$

Properties of W_N^{kn}?

Discrete Fourier Series (DFS)

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





- 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 same..... but different!

Discrete Fourier Transform

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

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

- It is understood that,

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

- Alternative formulation (not in book)
  Orthonormal DFT:

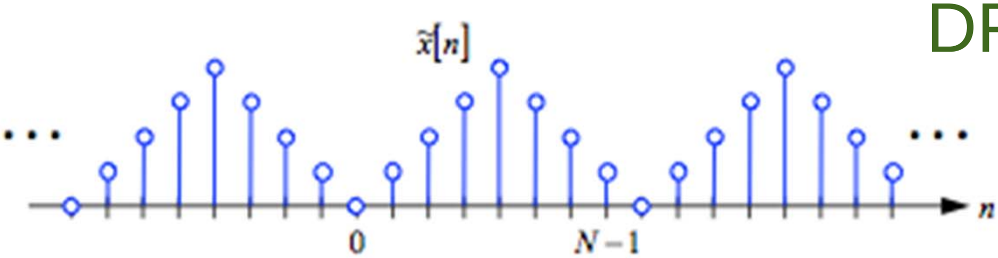
$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] W_n^{-kn}$$ Inverse DFT, synthesis

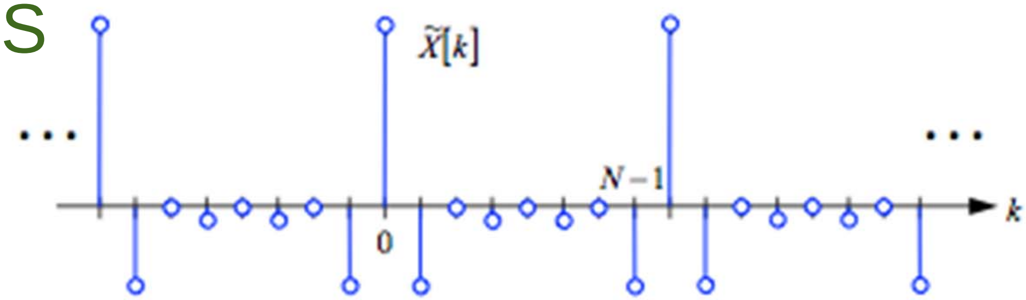
$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] W_n^{kn}$$ DFT, analysis

Why use this or the other?

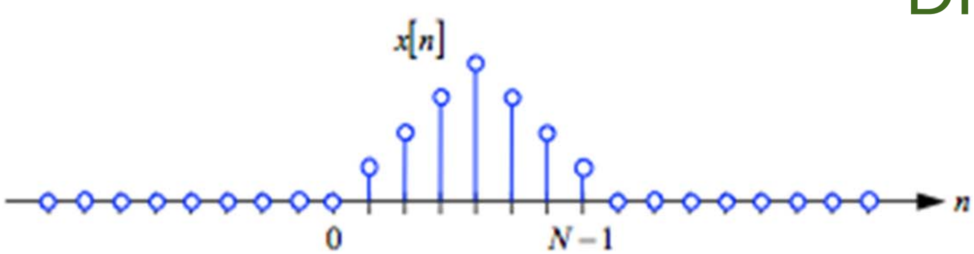
Comparison between DFS/DFT

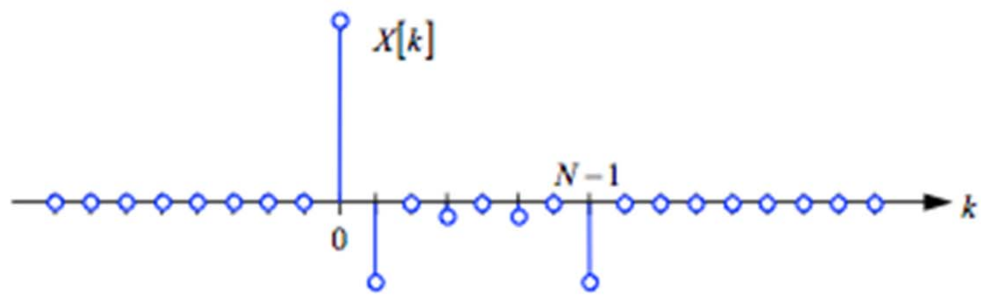
DFS



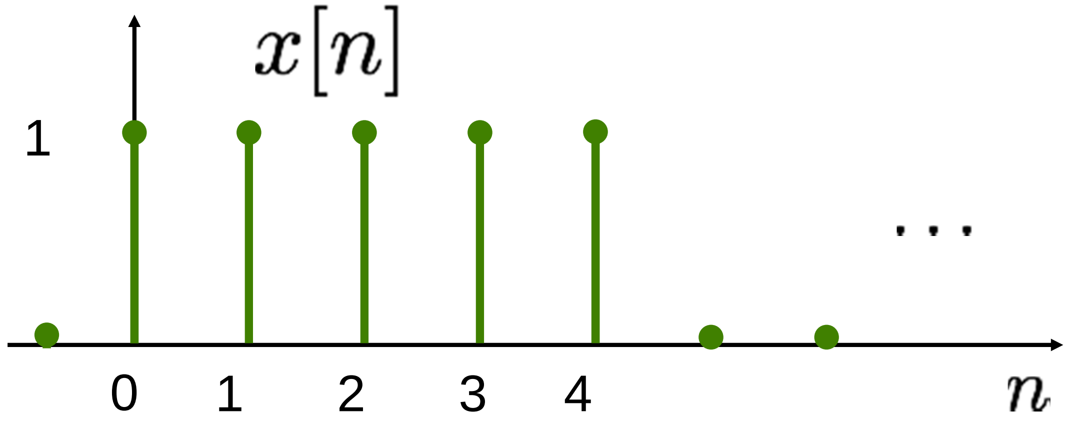


DFT





Example



- Take N=5

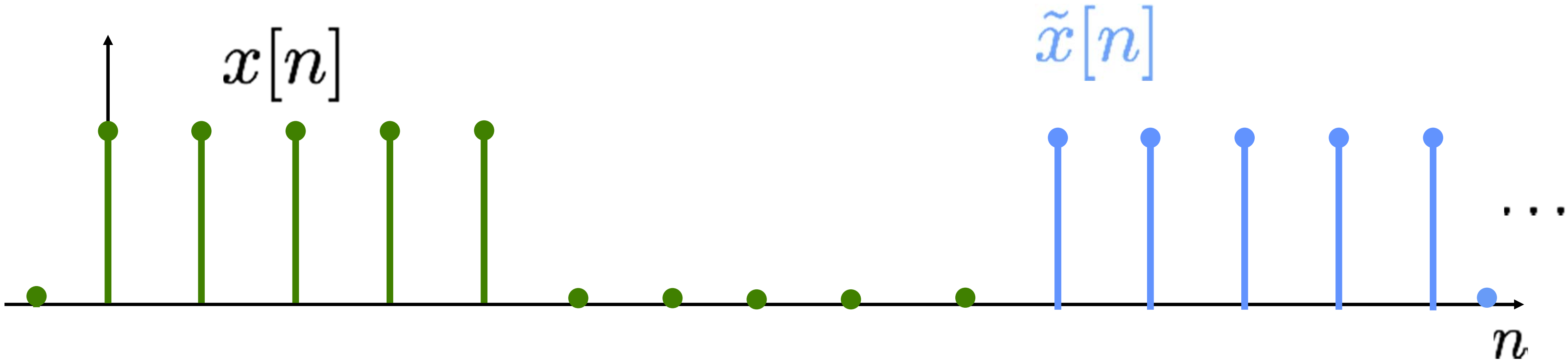
$$X[k] = \begin{cases} \sum_{n=0}^{4} W_5^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases}$$

$$= 5\delta[k]$$

"5-point DFT"

Example

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

- Show:

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

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

"10-point DFT"

DFT vs DTFT

- 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:

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

What is similar?

DFT vs DTFT

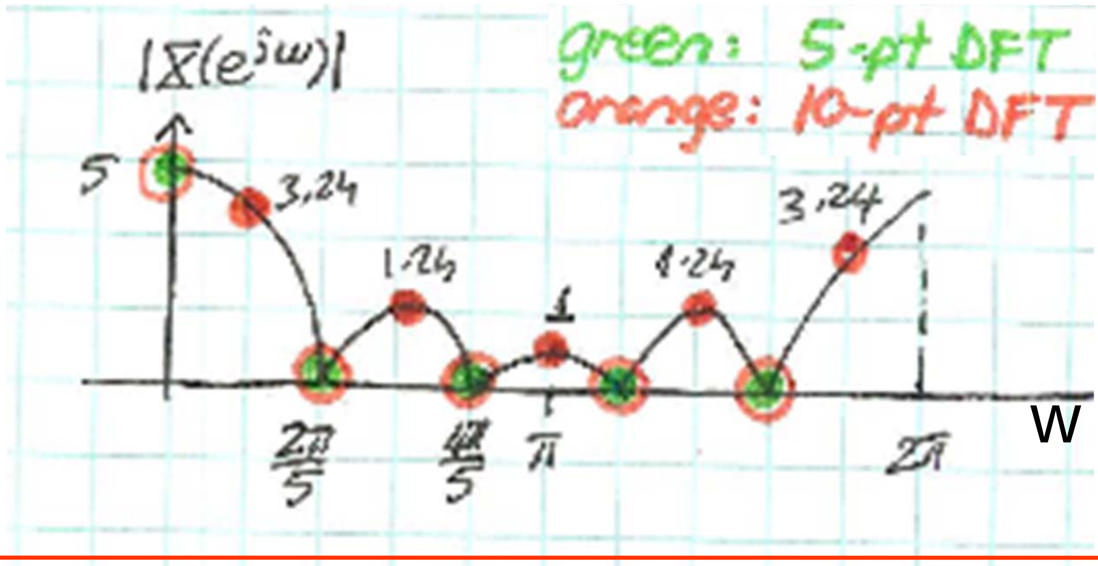
- The DFT are samples of the DTFT at N equally spaced frequencies

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

- Back to moving average example:

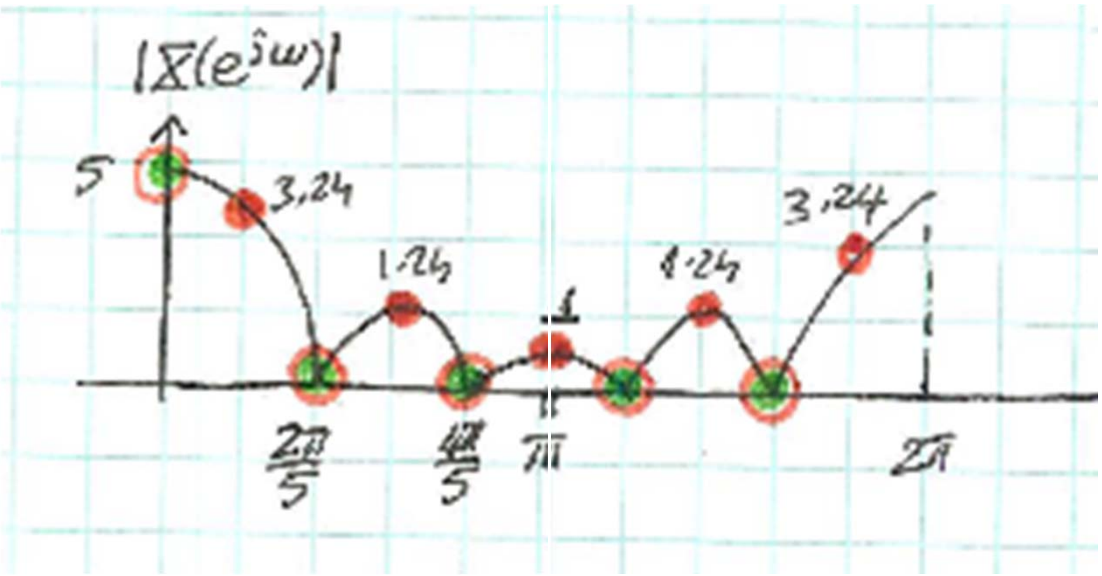
$$X(e^{j\omega}) = \sum_{n=0}^{4} e^{-j\omega n}$$

$$= e^{-j2\omega} \frac{\sin(\frac{5}{2}\omega)}{\sin(\frac{\omega}{2})}$$



FFTSHIFT

- Note that k=0 is w=0 frequency
- Use fftshift to shift the spectrum so w=0 in the middle.



DFT and Inverse DFT

- Both computed similarly.....let's play:

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

- Also....

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

DFT and Inverse DFT

- So,

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

or,

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

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

Why useful?

DFT as Matrix Operator

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:

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

straightforward implementation requires N^2 complex multiplies :-(

DFT as Matrix Operator

- Can write compactly as:

$$\mathbf{X} = \mathbf{W}_N \mathbf{x}$$
$$\mathbf{x} = \frac{1}{N} \mathbf{W}_N^* \mathbf{X}$$

- So,

$$\mathbf{x} = \frac{1}{N}\mathbf{W}_N^*\mathbf{X} = \frac{1}{N}\mathbf{W}_N^*\mathbf{W}_N\mathbf{x} = \frac{1}{N}(N\mathcal{I})\,\mathbf{x} = \mathbf{x}$$

WHY?

as expected.

Properties of DFT

- Inherited from DFS (EE120/20) so no need to be proved

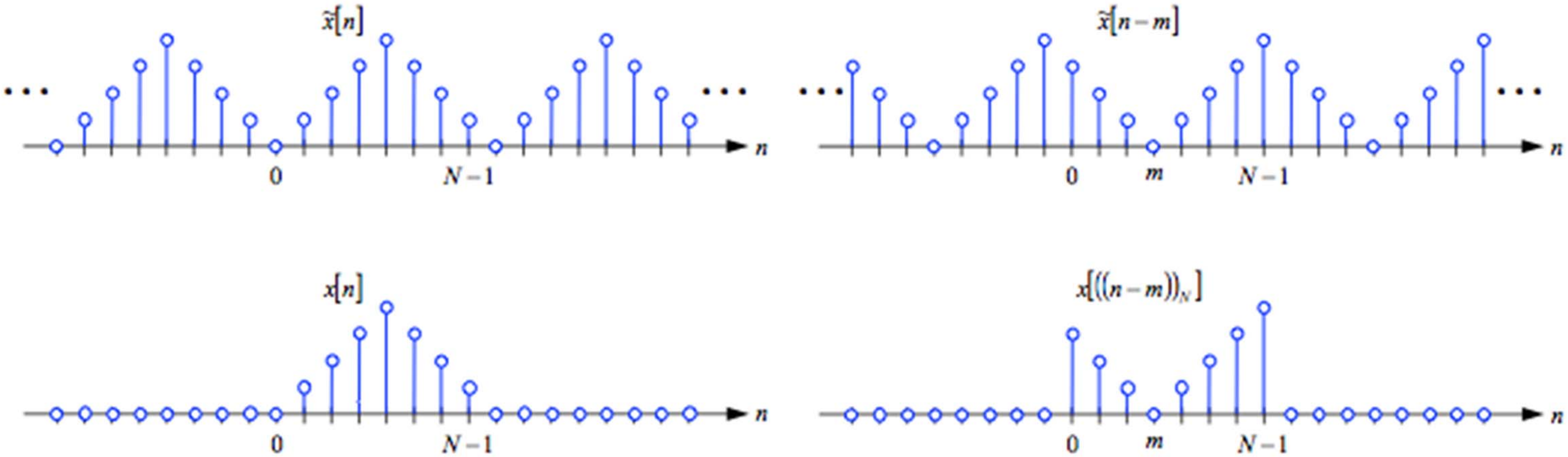
- Linearity

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

- 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

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

Show....

Examples

- 4-point DFT
  - Basis functions?
  - Symmetry
- 5-point DFT
  - Basis functions?
  - Symmetry

Cool DSP: Phase Vocoder, used in autotune

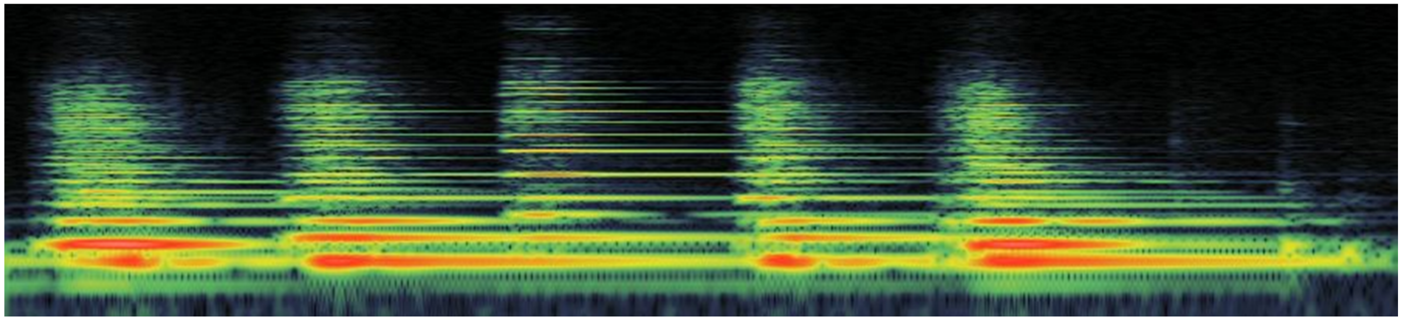
Changes the frequency and time domains of audio signals, while still accounting for phase information

Uses the DFT!
(we'll return to this at the end of lecture)

https://www.youtube.com/watch?v=6fTh0WRJoX4

How can this be used in a phase vocoder?

Frequency



Time

How would you slow this music down?

How about changing the pitch?

Phase correlations must be respected!

https://www.youtube.com/watch?v=O3_ihwhjHUw

- Parseval's Identity

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

- Proof (in matrix notation)

$$\mathbf{x}^*\mathbf{x} = \left(\frac{1}{N}\mathbf{W}_N^*\mathbf{X}\right)^* \left(\frac{1}{N}\mathbf{W}_N^*\mathbf{X}\right) = \frac{1}{N^2}\mathbf{X}^* \underbrace{\mathbf{W}_N\mathbf{W}_N^*}_{N\cdot\mathbf{I}}\mathbf{X} = \frac{1}{N}\mathbf{X}^*\mathbf{X}$$

Circular Convolution Sum

- 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

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

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