TOPICS FOR TODAY'S LECTURE

DISCRETE FOURIER TRANSFORM (DFT)

1. Relations with: (Applications:)
   a. DTFS (to compute spectrum)
   b. DTFT (frequency response)
2. Cyclic Convolution (compute * fast)

DISCRETE FOURIER TRANSFORM (DFT)

Another !#$%&̂ transform?!
Now have: FT, DTFT, DTFS, DFT! Why?

Because DFT is the "missing link"
between DTFS & DTFT & $\mathcal{Z}$ & $\mathcal{F}$.

DISCRETE FOURIER TRANSFORM (DFT)

Forward: $X_k = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, k = 0, 1 \ldots N-1$

Inverse: $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{+j2\pi nk/N}, n = 0, 1 \ldots N-1$

Note: Same formulae except: (1) $e^{\pm j2\pi nk/N}$; (2) $\frac{1}{N}$.
Note: Same as DTFS except for $\frac{1}{N}$.

DFT vs. Fourier Transform

Forward: $X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$.

Inverse: $x(t) = \int_{-\infty}^{\infty} X(f) e^{+j2\pi ft} df$.

Note: Same formulae except $e^{\pm j\omega t}$.

Note: Set $\omega = 2\pi f$ in usual $\mathcal{F}$ formulae.

DFT vs. DTFS

Forward: $x_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}$

Inverse: $x[n] = \sum_{k=0}^{N-1} x_k e^{j2\pi nk/N}$

Note: DFT and DTFS **same** except $\frac{1}{N}$:

That is: DFT=X_k=$N x_k$=N(DTFS).

But: DFT for finite-length signals
and: DTFS for periodic signals

DFT vs. DTFT & $\mathcal{Z}$

DTFT: $X(e^{j\omega}) = X(z)|_{z=e^{j\omega}}$.

DFT: $X_k = X(e^{j\omega})|_{\omega=2\pi k/N} = X(z)|_{z=e^{j2\pi k/N}}$.

That is: DFT computes DTFT at $\omega = 2\pi k/N$.

and: DFT computes $\mathcal{Z}$ at points on unit circle.

Text: Uses $X(k)$ for DFT, not X_k.
But: I *hate* that–too easy to confuse with $X(z)$!

DFT: COMPUTATION

See DTFS examples and omit $\frac{1}{4}$.

Matlab: `F=fft(X,N)` and `X=ifft(F,N)`.
Note: If know that x[n] is real, use `X=real(ifft(F,N))` (roundoff error).

DFT: APPLICATIONS

Computing:

1. Frequency response $H(e^{j2\pi k/N})$ (DTFT).
2. Spectrum $X(e^{j2\pi k/N})$ (actually DTFS).
3. Computing convolution & deconvolution fast.

DFT for FREQ. RESPONSE [1/2]

Goal: Plot Gain=$|H(e^{j\omega})|$ for:

System: y[n]=3x[n]+x[n–1]+4x[n–2]+6x[n–3].

Soln: $H(e^{j\omega})=3+e^{-j\omega}+4e^{-j2\omega}+6e^{-j3\omega}$.

$|H(e^{j\omega})|=|3+e^{-j\omega}+4e^{-j2\omega}+6e^{-j3\omega}|$=mess.

DFT: `plot(abs(fft([3 1 4 6],256)))`

DFT for FREQ. RESPONSE [2/2]

Goal: Plot Gain=$|H(e^{j\omega})|$ for:

System: y[n]+2y[n–1]=3x[n]+4x[n–1].

Soln: $H(e^{j\omega})=\frac{3+4e^{-j\omega}}{1+2e^{-j\omega}} \rightarrow |H(e^{j\omega})|=\frac{|3+4e^{-j\omega}|}{|1+2e^{-j\omega}|}$=mess.

DFT: `plot(abs(fft([3 4],256))./abs(fft([1 2],256)))`

Easier: `freqz([3 4],[1 2],256)` (gain and phase plots)

DFT for SPECTRUM [1/2]

Given: $x(t)$ sampled at $1000\frac{\text{SAMPLE}}{\text{SECOND}}$.

Lengths: $x(t)$: 3 sec.→x[n]: 3000 samples.

Goal: Compute spectrum of $x(t)$ from x[n].

Soln: `F=fft(X,3000)` where `X`=x[n] vector.

Then: Index k of F represents Hertz:

f=$\frac{k-1}{3\text{ SEC}}$=(k–1)$\frac{1000\text{ HERTZ}}{3000\text{ SAMPLE}}$ Hertz.

Indexing: Matlab starts at 1; k starts at 0.

DFT for SPECTRUM [2/2]

Note: $\frac{\text{FREQUENCY}}{\text{RESOLUTION}} = \frac{1}{T}$
where: T=duration of $x(t)$ in sec.
Means: Resolve $\frac{k}{T}$ and $\frac{k+1}{T}$ Hertz.

Note: $\frac{\text{TEMPORAL}}{\text{RESOLUTION}} = \frac{1}{F}$
where: F=sampling rate in $\frac{\text{SAMPLE}}{\text{SECOND}}$.
Means: Resolve $\frac{n}{F}$ and $\frac{n+1}{F}$ sec.

Note: T="timewidth" and F="bandwidth"; use N=FT.

DFT for SPECTRUM: EX [1/2]

Get: Suppose the Matlab output is:

$$\{\underbrace{0,0\ldots0}_{150},3+j4,\underbrace{0,0\ldots0}_{2699},3-j4,\underbrace{0,0\ldots0}_{149}\}.$$

Then: Component at (151–1)/3=50 Hertz & conjugate location (2851–1)/3=950 Hertz.

Note: NO component at 950 Hertz! (aliased)

Answer: $x(t)=\frac{2|3+j4|}{3000}\cos(2\pi\frac{151-1}{3}t+\arg[3+j4])$

$x(t)=\frac{1}{300}\cos(2\pi 50t+53^o)$ is the signal.

DFT for SPECTRUM: EX [2/2]

Try it yourself!

```
T=[0:0.001:2.999];
% NOT linspace(0,3,3000)!
X=cos(2*pi*50*T+atan(4/3))/300;
F=fft(X); F([150:152 2850:2852])
```

CYCLIC CONVOLUTION [1/3]

DEF: y[n]=h[n]ⓒx[n]=$\sum_{i=0}^{N-1}$h[i]x[(n–i)mod(N)]

Order: "N-point"⇔"order N"⇔y[n],h[n],x[n] all length N.

Note: "Cyclic"⇔"circular"; $*$ ⇔"linear" convolution.

EX: $\{1,2,3\}$ⓒ$\{4,5,6\}$: 1 complete cycle of each:

y[0]=$^{1,2,3,1,2,3}_{4,6,5,4,6,5}$=$(1\cdot 4)+(2\cdot 6)+(3\cdot 5)$=31.

y[1]=$^{1,2,3,1,2,3}_{5,4,6,5,4,6}$=$(1\cdot 5)+(2\cdot 4)+(3\cdot 6)$=31.

y[2]=$^{1,2,3,1,2,3}_{6,5,4,6,5,4}$=$(1\cdot 6)+(2\cdot 5)+(3\cdot 4)$=28.

CYCLIC CONVOLUTION [2/3]

OR: $\{1,2,3\}*\{4,5,6\}$=$\{4,13,28,27,18\}$.

Alias: {4+27,13+18,28}=$\{31,31,28\}$.

Same as: Multiply z-transforms mod(z^N–1).

OR: Turning wheels (see text, recitation).

CYCLIC CONVOLUTION [3/3]

Why? Why care about cyclic conv.?

Since: y[n]=h[n]ⓒx[n]→ Y_k=H_kX_k

where: X_k=DFT{x[n]}, etc.

So: Compute cyclic convolution quickly.

by: Multiplying DFTs and taking IDFT.

So? 0-pad cyclic to linear convolution.

EXAMPLE #1 [1/2]

Goal: Compute $\{1,2,3\}*\{4,5\}$.

Plan: 0-pad to length of y[n] (4).

Soln: =$\{1,2,3,0\}$ⓒ$\{4,5,0,0\}$.

Plan: Compute 4-point DFTs of $\{1,2,3,0\}$ and $\{4,5,0,0\}$; multiply; 4-point IDFT.

EXAMPLE #1 [2/2]

`fft([1,2,3,0])=[6,-2-j2,2,-2+j2]`

`fft([4,5,0,0])=[9,4-j5,-1,4+j5]`

Multiply: `[54,-18+j2,-2,-18-j2]`

`ifft([54,-18+j2,-2,-18-j2])=[4,13,22,15]`.

EXAMPLE #2 [2/2]

`fft([6,19,32,21])=[78,-26+j2,-2,-26-j2]`

`fft([1,2,3,0])=[6,-2-j2,2,-2+j2]`

Divide point-by-point: `[13,6-j7,-1,6+j7]`

`ifft([13,6-j7,-1,6+j7])=[6,7,0,0]`.

EXAMPLE #2 [1/2]

Goal: Solve $\{\underline{1},2,3\} * x[n] = \{\underline{6},19,32,21\}$.

Plan: Compute DFTs H_k, Y_k; x[n]=IDFT$\{\frac{Y_k}{H_k}\}$

Provided: $H_k \neq 0$; else use $Y_k \frac{H_k^*}{|H_k|^2+\epsilon^2} \approx \frac{Y_k}{H_k}$.