Spectrum analysis using DFT (Harris)

Introduces standard metrics / "figures of merit"

0) "bin"

when we take FFT (DFT), get answers in "bins"

N-point FFT means
- divide discrete frequency ω into N bins $[0, 2\pi)$ range

  $\Delta\omega = \frac{2\pi}{N}$
- divide continuous frequency range $[0, F_s]$ into N bins

  $\Delta F = \frac{F_s}{N} = \frac{1}{TN}$

We can see these agree:

$\frac{\Delta\omega}{2\pi} = \frac{\Delta F}{F_s}$

but ~~Fs~~ TN = (sampling interval)(# samples)

$= T_{window}$

length of window in sec

so $\Delta F \sim 1/T_{window}$

Reminder:

by zero-padding, we can change apparent bin width.

However, this does not change resolution – resolution is set by actual length of data used

* Harris doesn't consider zero-padding

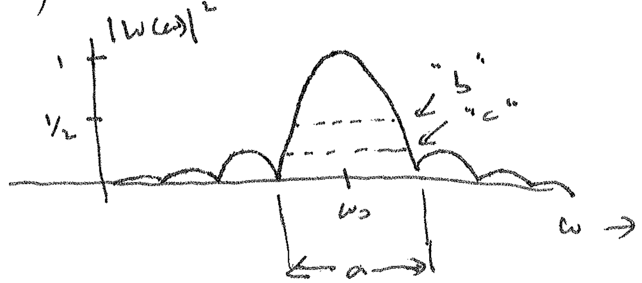
1) Resolution

Related to window main lobe width: multiple definitions

widest a) Proakis & Manolakis : distance between first nulls

b) −3 dB (half-power) point: most common

narrowest c) −6 dB (¼-power) point: clearly resolve two tones



~~Harris~~

Resolution cont

Harris gives resolution in terms of bins, no zeropadding

so for rectangular, -3 dB BW is 0.89 bin

$$0.89\left(\frac{2\pi}{N}\right) \longrightarrow \text{vs. } \frac{4\pi}{N} \text{ for P\&M definition}$$

2) Sidelobe levels

Specify in terms of highest (worst) SL and fall-off per octave (frequency doubling)

3) Coherent gain

When we apply a window, we ~~are deweighting~~ can change the signal energy input to the DFT.

→ This affects amplitude: easy to calculate & correct for.

assume we have input signal exactly at the bin center ω_k

$$x(n) = A\, e^{j\omega_k n}$$

then,

$$X(k) = \sum_{n=0}^{N-1} w(n)\, x(n)\, e^{-j\omega_k n}$$

$$= A \sum_{n=0}^{N-1} w(n)\, e^{j\omega_k n} e^{-j\omega_k n}$$

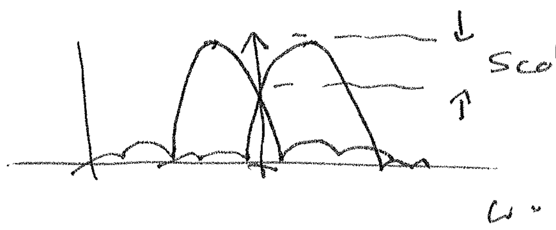
$$= A \sum_{n=0}^{N-1} w(n)$$

$= N$ for rectangular, less for others.

Q: what do you expect for triangle?

4) <u>Scalloping loss</u> → signal ~~a~~ probably is not exactly at FFT bin center.

Worst-case: signal is exactly between two bins



this is a situation where <u>zero padding</u> <u>can help</u>

topic 5 Noise bandwidth

consider a sinusoid in noise

$$x(n) = A \sin \omega_0 n + \text{noise}$$

If noise is uncorrelated from one sample to another, its autocorrelation will look like

$$r_{xx}(l) = \sigma^2 \delta(l)$$

which Fourier transforms to a flat spectrum — "white noise"

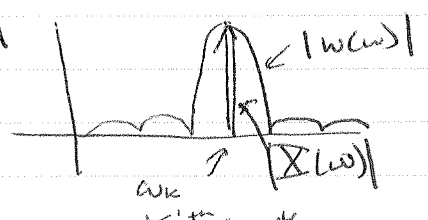
Typically we talk about AWGN:

- **A**dditive (instead of multiplying signal)
- **W**hite (uncorrelated sample-to-sample)
- **G**aussian (each sample comes from Gaussian distribution)
- **N**oise

~~We can th~~

When we do spectral analysis, we get a combination of noise & signal in each frequency bin.

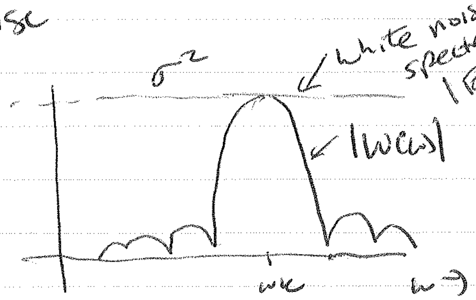
① signal



← picture for rectangular window

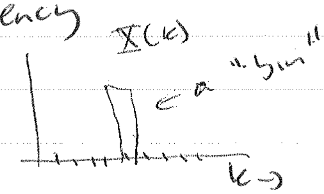
for a sinusoid perfectly centered in our bin, the signal is passed through

② noise



(OVER)

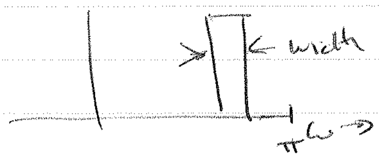
note: "bin" means the region of frequency around ~~and~~ each DFT/FFT center frequency



"noise" continued ..

from the last picture, we can see that the noise output for the k^{th} bin will contain noise from many frequencies
- sidelobe leakage
- mainlobe

If our windows gave rectangular windows in the frequency domain, it would be easy to calculate SNR in the bin



→ our signal would pass thru
→ noise would be attenuated by factor of $\frac{\Delta\omega}{\pi}$

so SNR would be

$$SNR = \frac{S}{\frac{\Delta\omega}{\pi}\sigma^2}$$

a longer window (bigger M) makes mainlobe smaller, so $M\uparrow$ $\Delta\omega \downarrow$ and SNR $\uparrow$

Real windows don't do this, but people have computed <u>noise-equivalent</u> <u>bandwidths</u> for various windows.

Can be used for comparison.

However, longer windows always help the SNR – so integration time is good for weak signals in noise

matlab example: integration_time_example.m //