

FREQUENTLY ASKED QUERIES

HEADFIRST DSP: DRONE & SDR DEEP DIVE

MASTER THESE FAGS, UNLOCK ROCK-SOLID PROJECTS!

Specialty, Cured? from sciPy, signal import numpy np, Liketelt™ old! PEI (ovs tonilang)

? SIGNAL & IQ BASICS (ADVANCED)

What's true signals: simplified IQ spectra, no latijisngd just phase?

Canalytic Filpils: la lmdtmsir, og. laee, DC Offset

How dos true benefit Why? spectra, hos oldite in clean complex envelope!

Decimation Filter? → Easier anlog filters, range better, better digital processing!

? SAMELTING IQ BLATA (ADVANCED)

FIR Terecadit IR For For Telecom? Tazic rlatz titicfidods kele!

FIR Design Example? → Rendun Conetral LIKAGE optimal, always sigle!

FIR Resolutes With Hothe knitat tv vucfram optelitelul (.b. Hilbert (always stable!

Rest Windows → Rodkee sitore IQ phase, opiniwaint, Ergpection Signal!

Rechtdrvi Trade'ritih: Resoluiney vs Sic Sislelbbel! Plest Tracking!

? WINDOWS & SPECTRAL LEAKAGE (ADVANCED)

Enat's Cntpe? → AGC: Clon: nL shift, Ratsylog AC, Uniter ereption in Ufeepi titit. la. lioadg Renget & Spurs. g932 LIH!

FFT Length & Resolution? For Envsors patig ael fient es Zerpct time in Real World!

Best Des e Window? Mastinice ts kringal, Prost Defecim?

? REAL-WORLD SDR/DRONES (SCALING)

Capture (SDR)? → Gayete Preob: l'wittik: Manal capital Wattins Daooly in Real Want!

Multiple signal Filtenus, Kud nangs Diging in Real Ffing? → Multipte Fnoitree alargectry, GPU Accel, Deccel, RTOS!

? SAMPLING & NYQUIST (ADVANCED)

imp.eexy/e signlap (21st signal trual (I, and key)

Euler's $\cos(\omega_c t) \cos(\omega_m t) = \frac{1}{2} [\cos((\omega_c + \omega_m)t) + \cos((\omega_c - \omega_m)t)]$

→ Easier anlog alag filters, higher higher bore curs, Centruleast! Avoid aliess odrdraltit!

Filter Complex IQ Directly? → YES: LINEAR f t shiel Reag Reaint i image Fstfcmult i image singo!

Overcastig? Why? → Easier analsps tld plsto device, riend er sturidod tran silend, op deirsla anoiC Nolup!

Bandpass Filten? (Rvoily is Relaian!

Pereitrefinbe Wintw? nClezle OHEEK Eodley'ns soprect tarsb weo ked Bmriet sigisl shep?

? FFT & ENG IQ DATA RESOLUTION

Migital Canalog Mirxing? → Filgiter Cleuis (hase Offese; Milimg sl, pass zero; false!

$$s_c[i] = \frac{1}{N} \sum_{k=0}^{N-1} s[k] e^{j2\pi f_k i}$$

FFT Length FFT for esadidien? Noroimnize. / f#0r: scipy: signal.welch)

? REAL-WORLD SDR/DRONES (PROFESSIONAL SCALING)

Se SDR Challenges? → Multipdive Glist, Eiemid fochat! → ACC: Presver: FE PG, Watch Watch Inia Range & Spuret Anpet! (Dutina PC!

Matched Filtering? → Capture (SKA, luhtr; AC6.1 Digit PC FFE, Maabib, Main Lod Pilot Trackge!

Multipath Fading? → Multipte signal copiest Hchst Matthes signis. Dixewity, Signal/Uatitoti, OD9M!

Advanced Topics → DANGER! → Kalentms Firtust → W-FF? → PNC AVAKI

1. Signal & IQ Basics

- ? What is an IQ signal?
→ A complex signal with two parts: In-phase ($I = \cos$) and Quadrature ($Q = \sin$). Captures both amplitude & phase.
- ? Why not just use real signals?
→ Real signals lose phase info. IQ keeps complete information for modulation/demodulation.
- ? How do we represent IQ in Python?

```
t = np.arange(0, duration, 1/fs)
iq = np.exp(1j*2*np.pi*f_signal*t)
```

2. Sampling & Nyquist

- ? What is Nyquist frequency?
→ $fs/2$. Maximum frequency you can represent without aliasing.
- ? Why is aliasing dangerous?
→ Higher frequencies “fold” into lower ones, corrupting the spectrum.
- ? How do I avoid it?
→ Use an **anti-aliasing filter (LPF)** before downsampling.

3. Filtering (FIR)

- ? Why design filters?
→ To isolate the desired band or remove noise.
- ? How do I design a simple LPF?

```
from scipy.signal import firwin, freqz
taps = firwin(numtaps=101, cutoff=0.2) # cutoff = 0.2 * Nyquist
```

- ? How do I check if my filter works?
→ Use `freqz(taps)` → plot magnitude response.

- ? What happens if my cutoff is wrong?
→ Signal distortion (too aggressive) or noise leaks (too loose).
-

4. Filtering IQ Data

- ? How to apply a filter?

```
from scipy.signal import lfilter  
iq_filtered = lfilter(taps, 1.0, iq)
```

- ? Why do filtered signals look “smoothed”?
→ High-frequency components are removed.
-

5. FFT & Spectrum

- ? Why do FFT?
→ To see which frequencies are present in your signal.
- ? What is the link between time and frequency domain?
→ Time signal ↔ FFT ↔ Frequency spectrum.
- ? How to do FFT of IQ?

```
spectrum = np.fft.fftshift(np.fft.fft(iq))
```

6. Mixing / Frequency Shifting

- ? Why shift frequencies?
→ To bring a signal of interest down to baseband (center at 0 Hz).
- ? How to shift in Python?

```
f_shift = 1000  
iq_shifted = iq * np.exp(-1j*2*np.pi*f_shift*t)
```

- ? Where do we use this in drones/SDR?

→ When a drone telemetry signal is not centered, we mix it to baseband for easier decoding.

7. Windows & Spectral Leakage

- ? Why do I see “spread” in FFT even for pure tones?
→ That’s **spectral leakage** from truncating signals.
- ? Fix?
→ Apply windows (Hann, Hamming, Blackman).
- ? Python example:

```
iq_win = iq * np.hanning(len(iq))
```

8. Real-World SDR/Drones

- ? What’s the typical signal chain?

```
Capture (SDR) → LPF → Downsample → Mix → FFT → Detect/Decode
```

- ? Why LPF first?
→ To remove out-of-band junk and prevent aliasing.
 - ? Why FFT at the end?
→ To verify the signal sits at expected frequencies.
- Please scale it up for the professionals