Quantifying Signal Quality

Practical Tools for High-Fidelity Waveform Analysis

GNU Radio Conference 2025 – C. Daniel Boschen

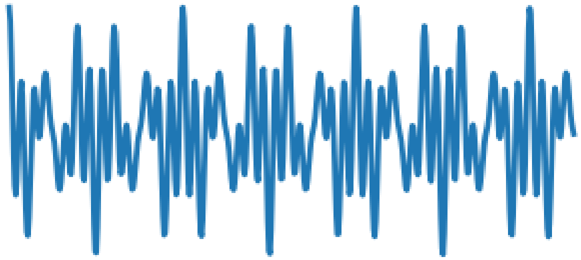
dsp-coach.com

Quantifying Signal Quality – Practical Tools for High-Fidelity Waveform Analysis
Content is derived from Signal Processing and Python courses taught by Dan Boschen

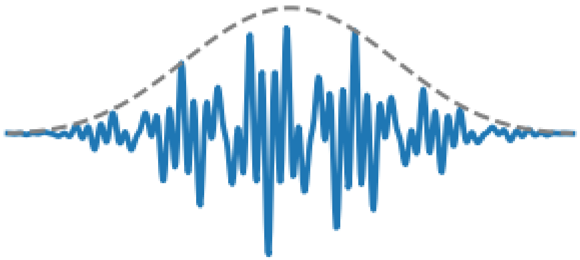
For additional resources and courses, visit dsp-coach.com

Intro / Goals

Accurate Signal and Noise Measurement

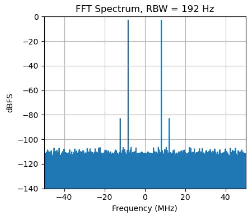
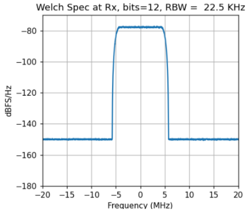
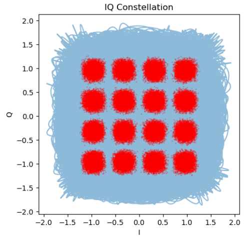
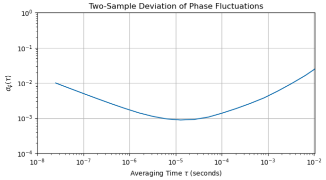


Time and Frequency Techniques



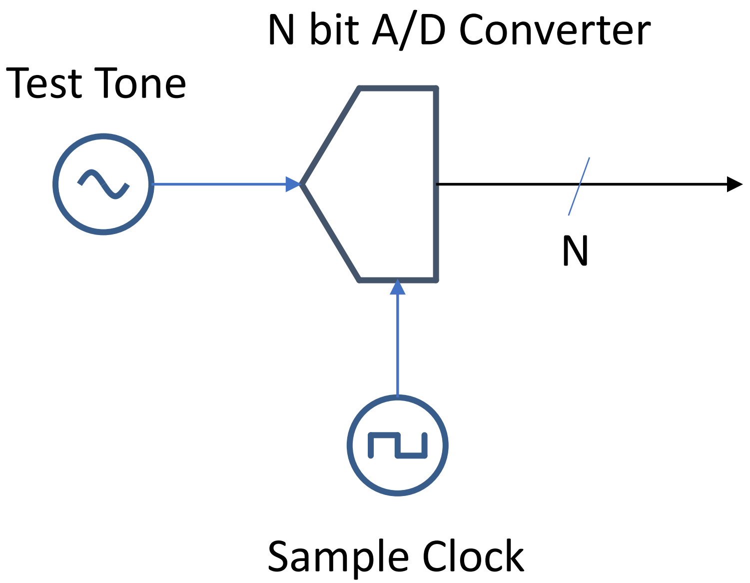
Post Processing Applications: Simplicity over Efficiency

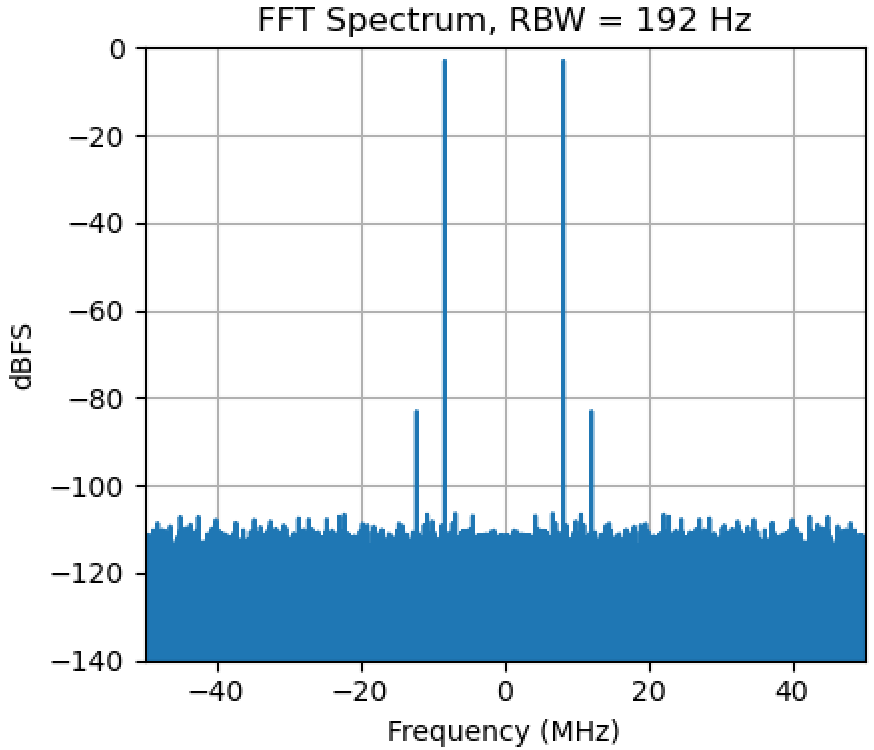
Summary

	Tools		Application
	FFT Spectrum	→	Narrowband: power of individual tones
	Welch PSD	→	Wideband: noise density
	EVM, Correlation	→	SNR
	ADEV	→	Stationarity, max observation time

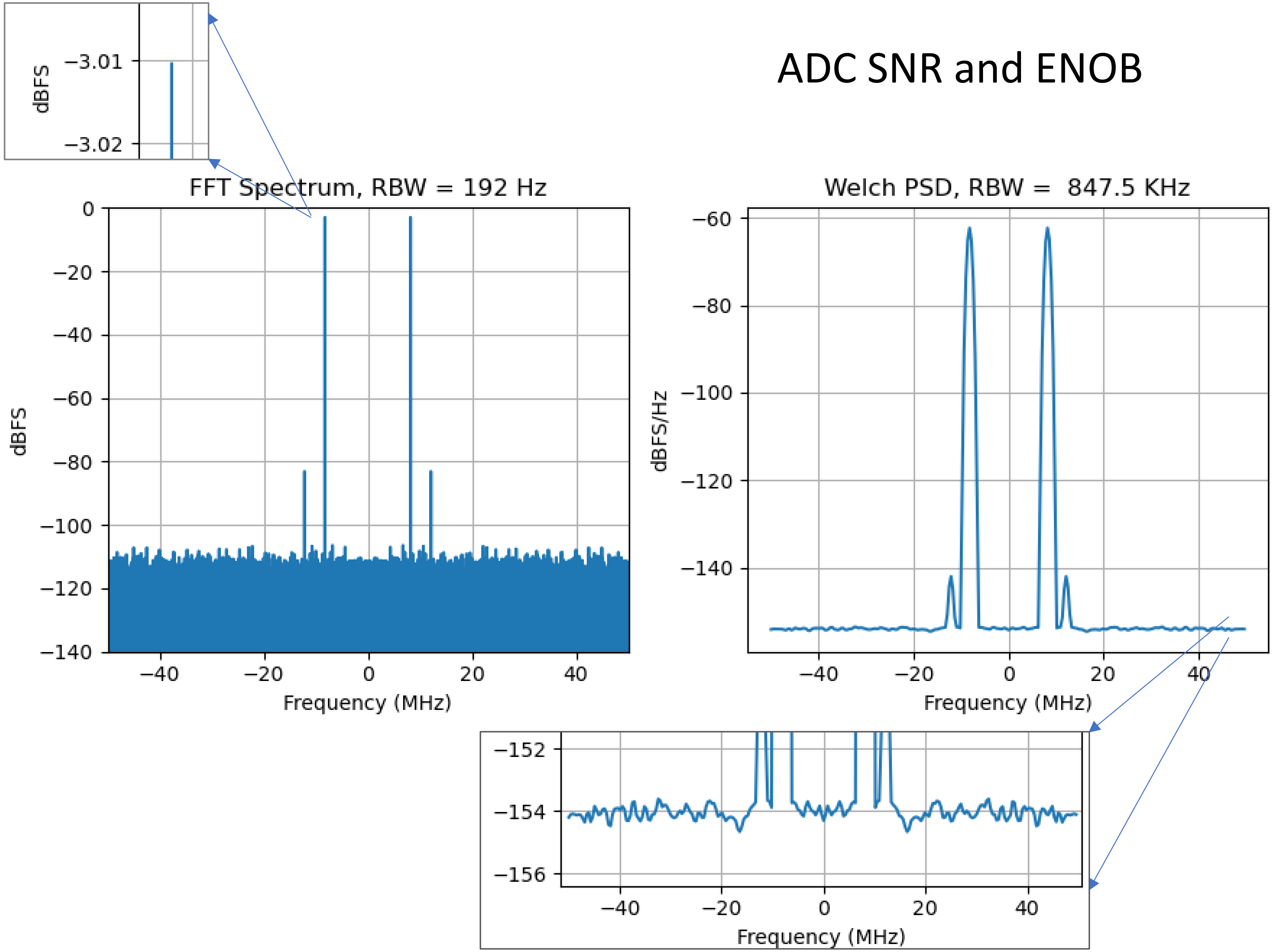
Compelling Examples

ADC SNR, Spurs and ENOB



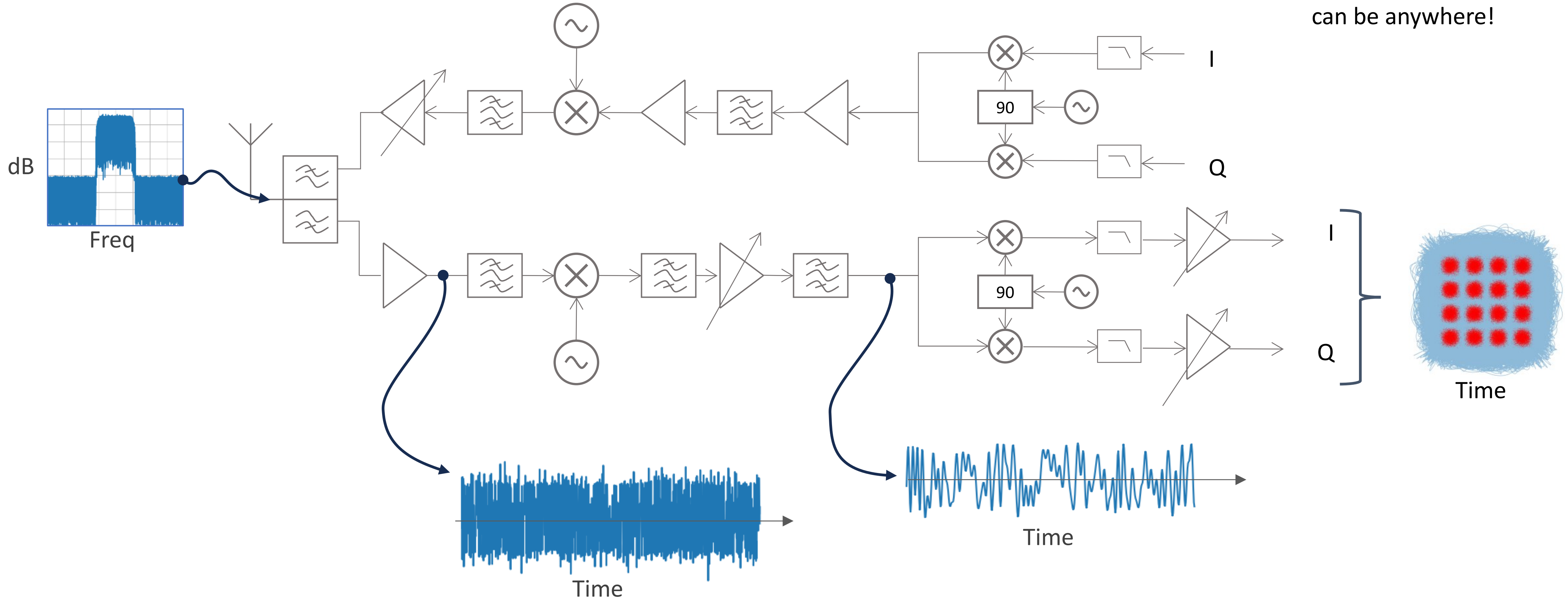


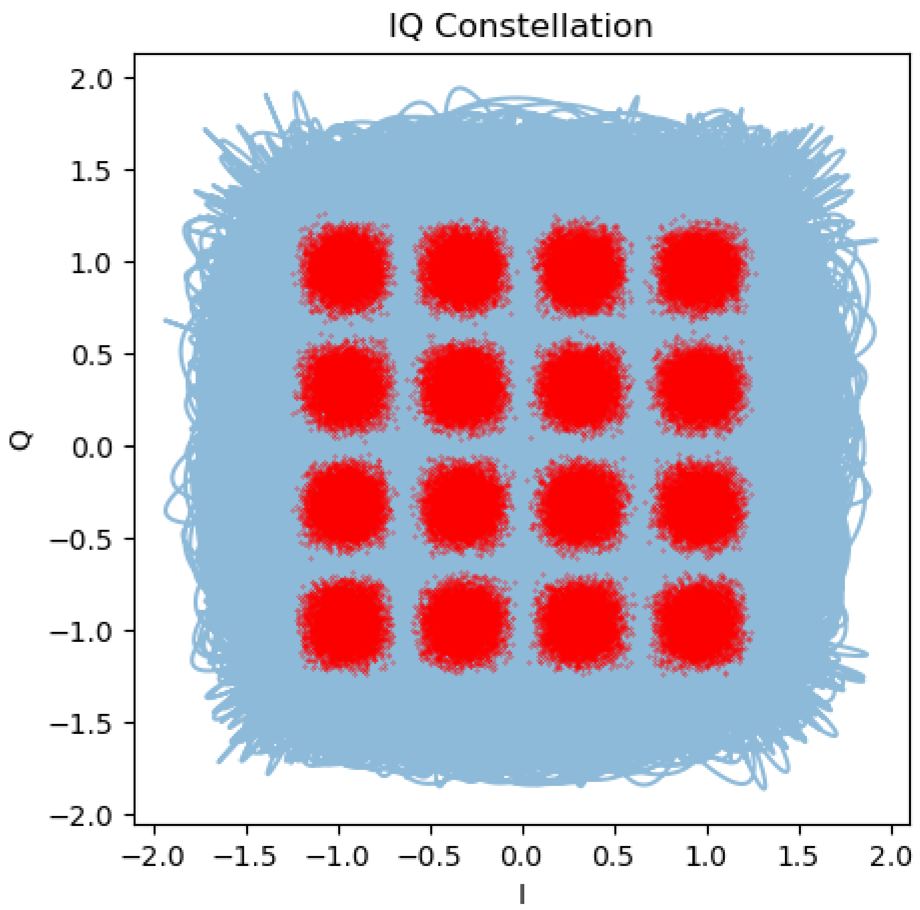
ADC SNR and ENOB

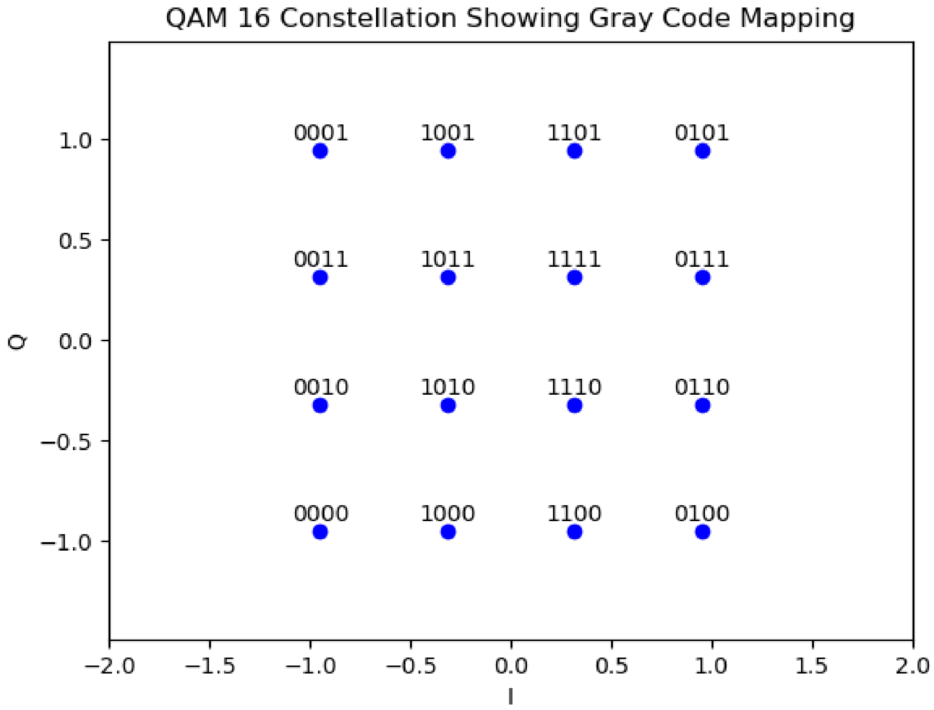


Waveform Quality in a Processing Chain

A/D and D/A boundary can be anywhere!

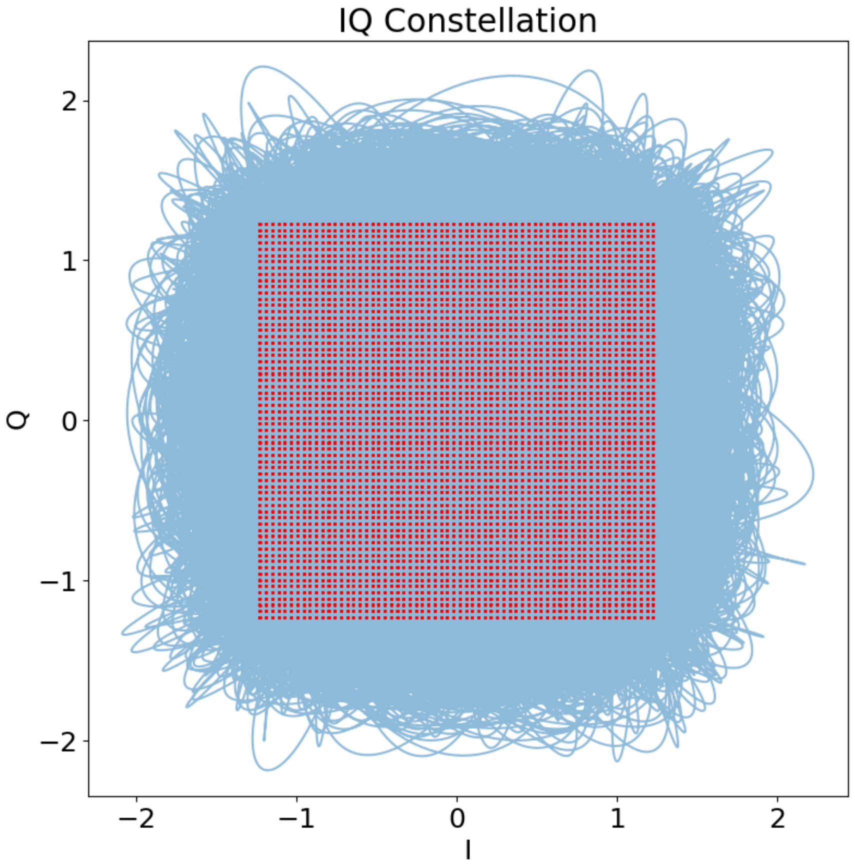


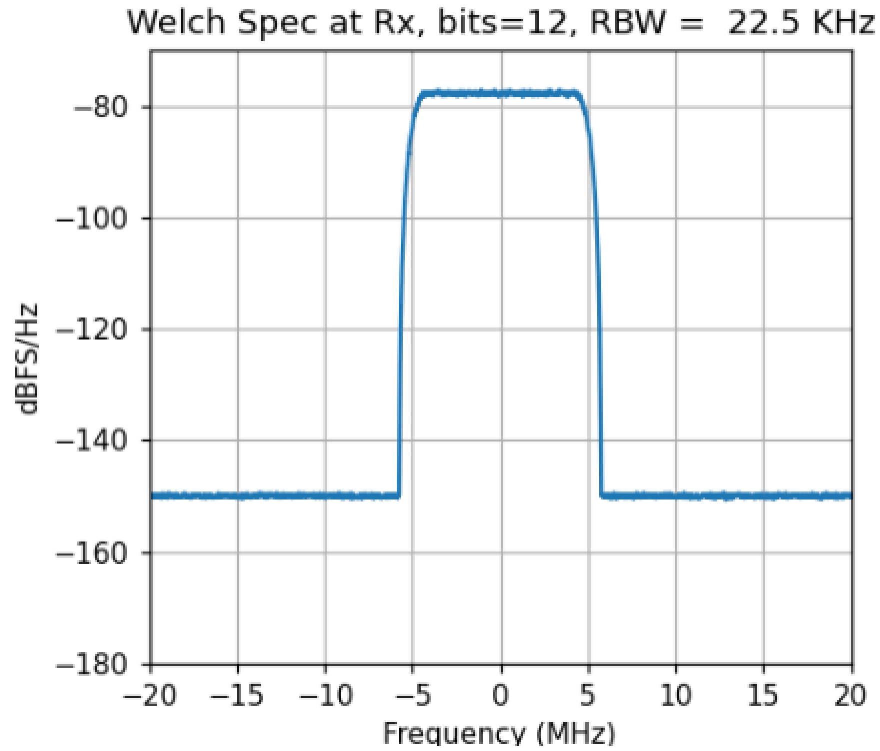




16-QAM

EVM, SNR, Spectrums

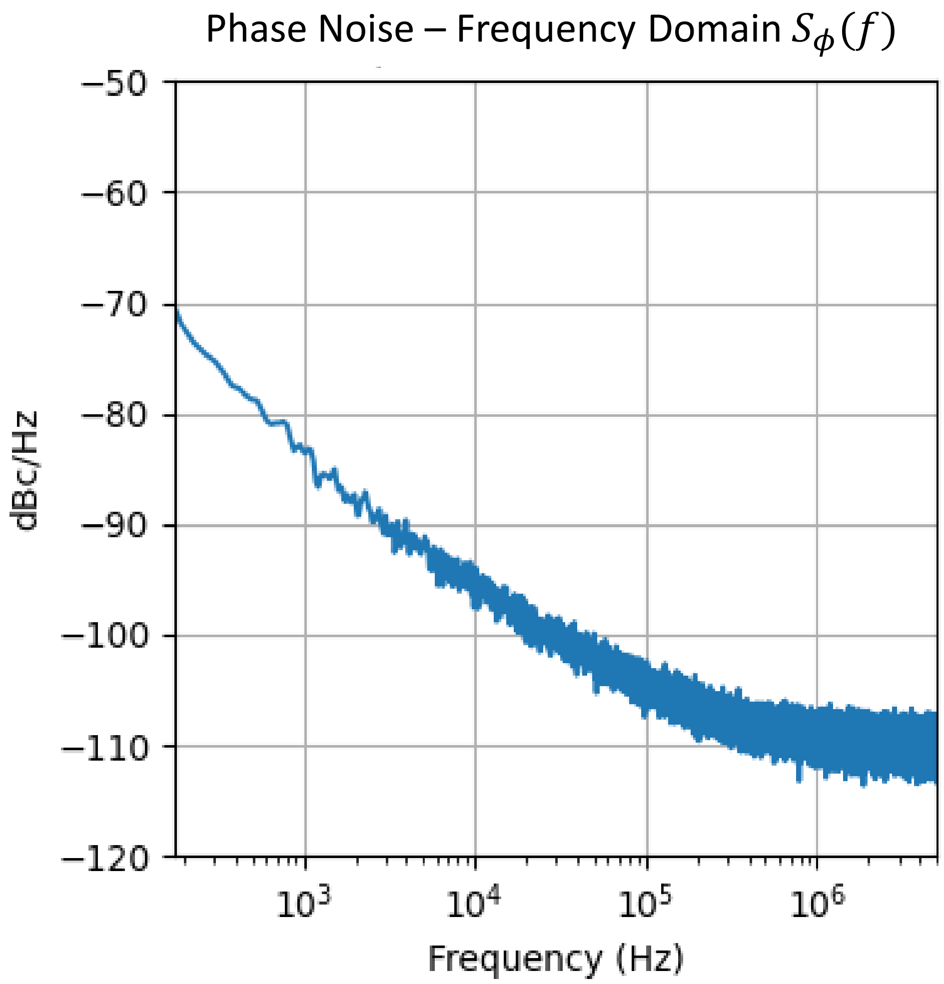


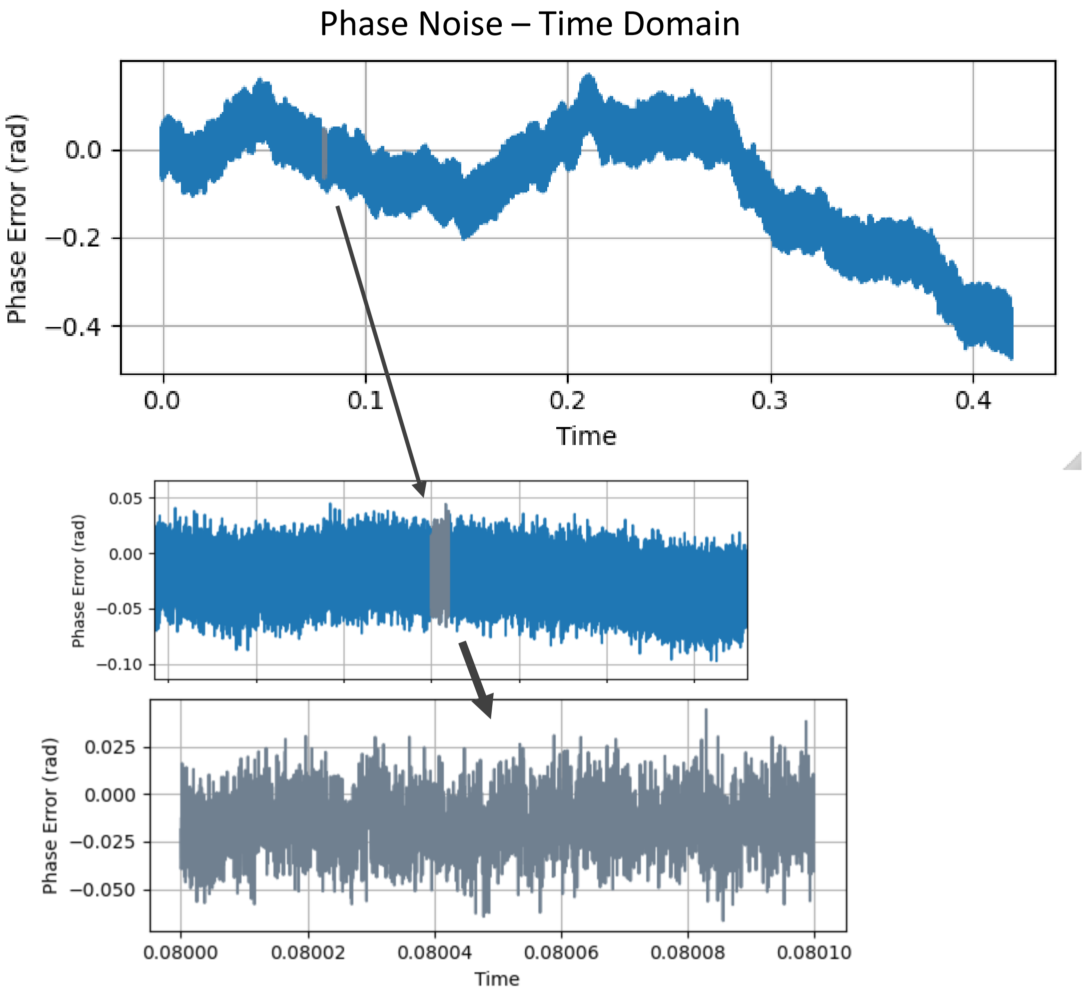


4096-QAM

EVM, SNR, Spectrums

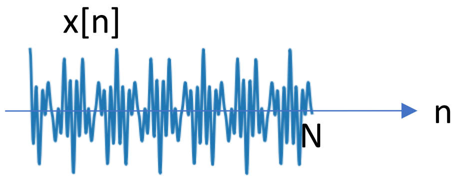
Consideration for Non-Stationary Signals



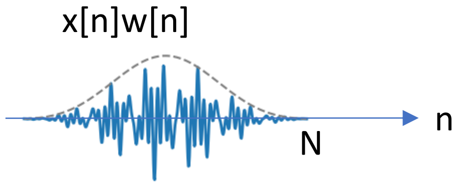


Power Measurement of Tones

FFT Spectrum Processing (for Tones)



Duration N sets resolution bandwidth (RBW)



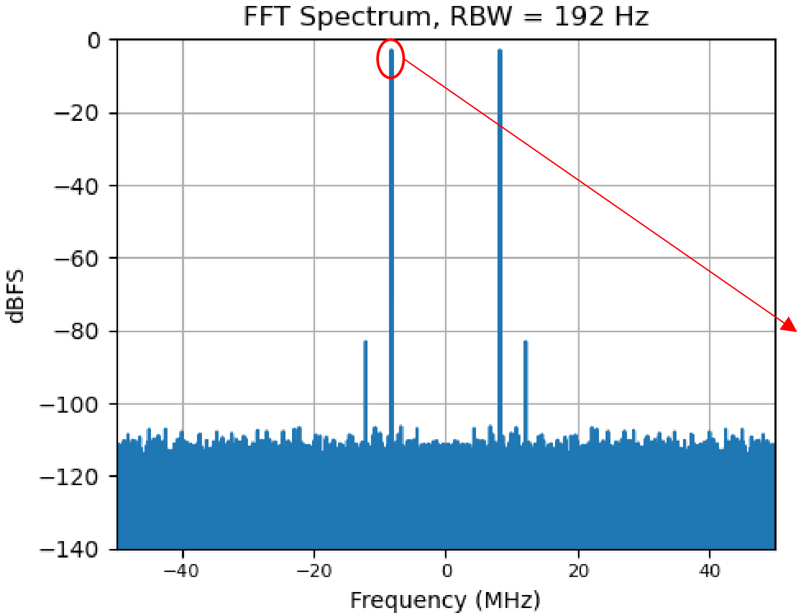
Window (Kaiser, vary β for dynamic range needed)

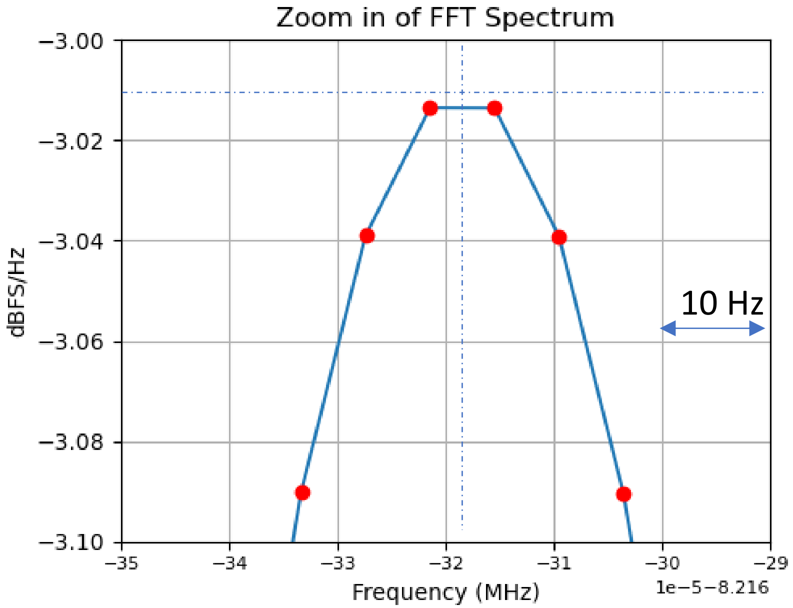
Zero pad (P=4 to 10x, next pow 2)

$$X[k] = \frac{1}{\sum w[n]} \sum_{n=0}^{PN-1} y[n] W_{PN}^{nk}$$

Scale FFT by sum of window

Results of FFT Spectrum





Test waveform:

$$x(t) = A_1 \cos(2\pi f_1 t) + A_2 \cos(2\pi f_2 t)$$

Sampling Rate: f_s = 100 MHz
Total number of samples: N=1,048,576
Primary tone: f_1=8.2163185 MHz, A_1=1
Spur: f_2=12.13 MHz, A_2 = 1e-4

Sampled and quantized to 12 bits to be x[n]

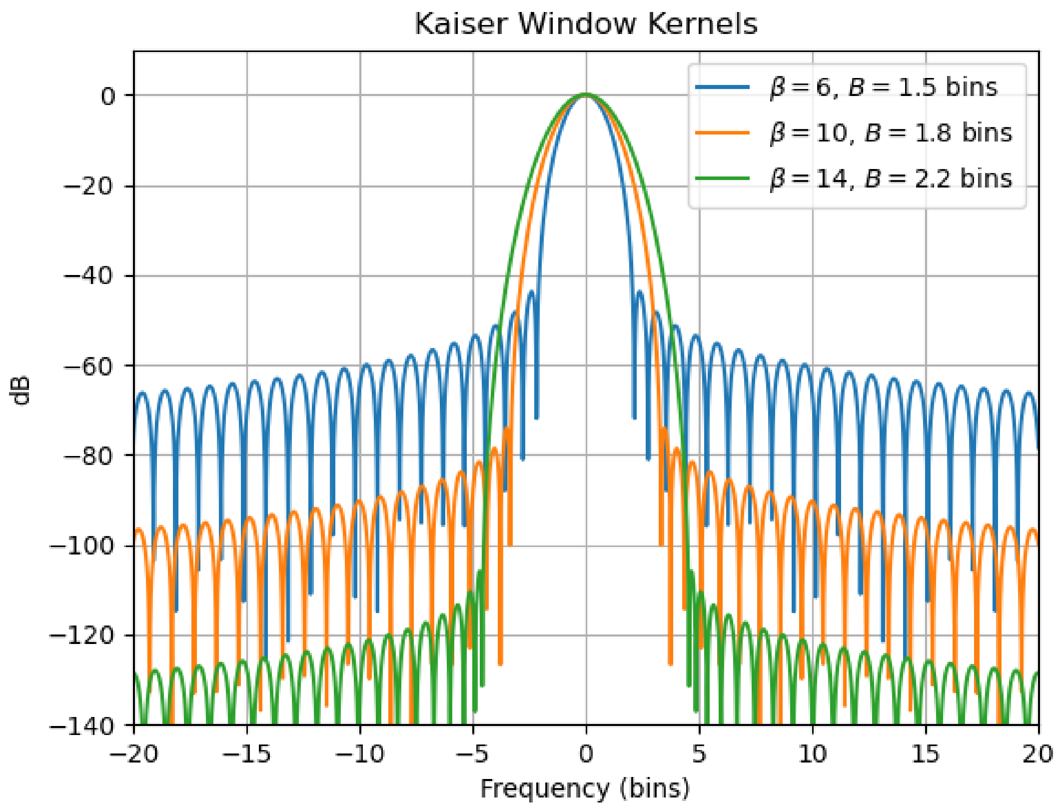
Resolution BW and Dynamic Range

Resolution Bandwidth and Dynamic Range: trade space with window used.

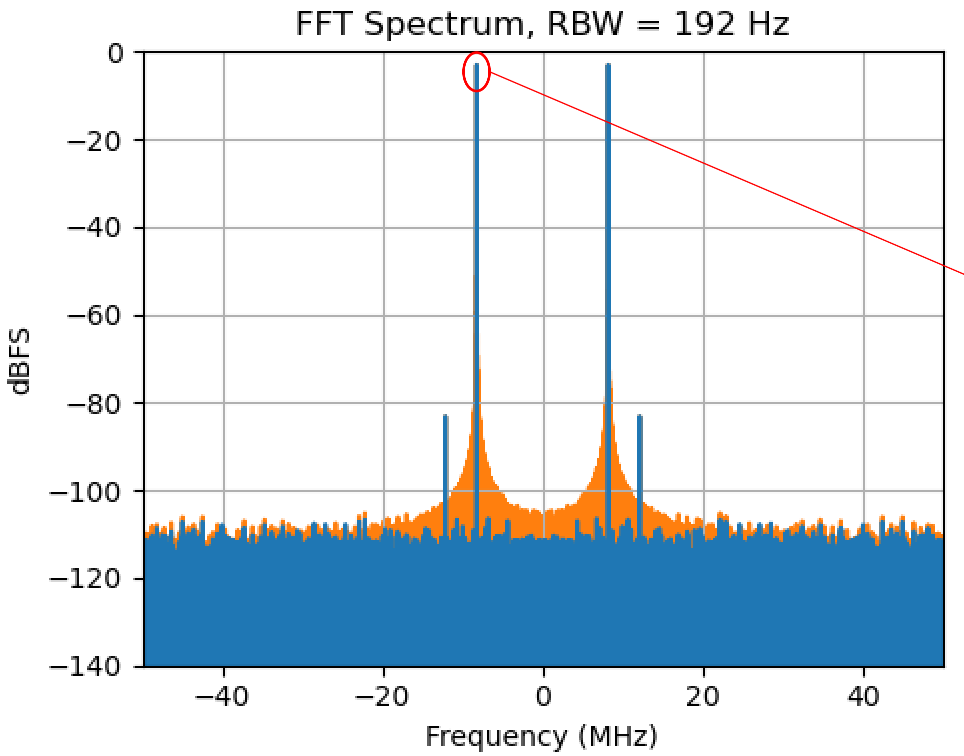
$$B = N\frac{\sum(w[n]^2)}{(\sum w[n])^2}$$

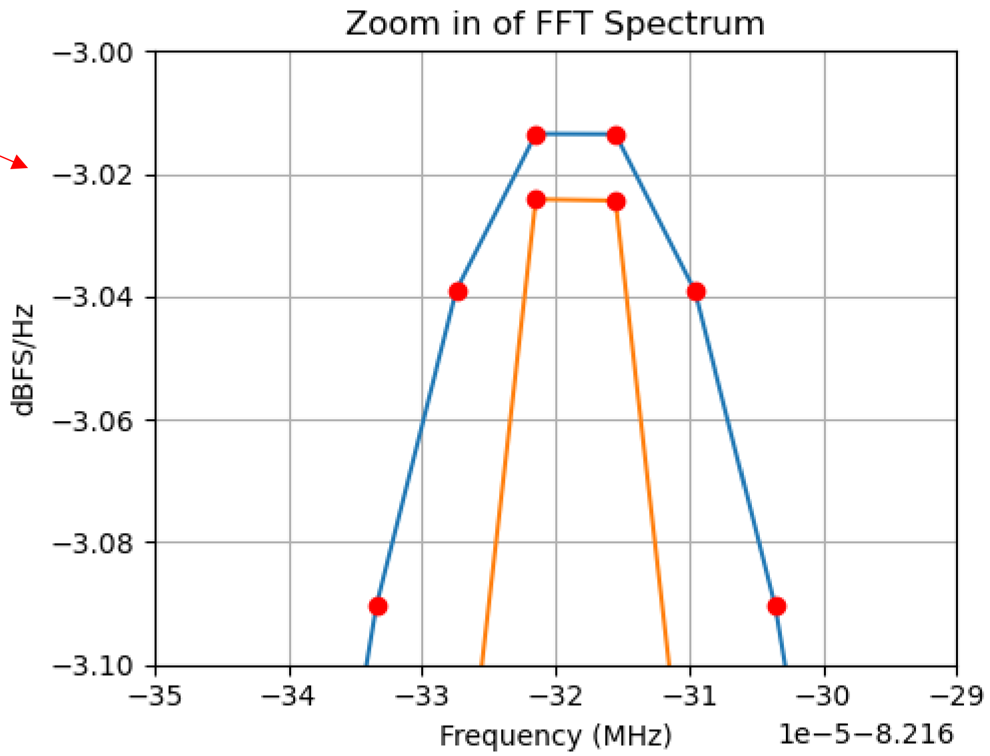
B result is in number bins of the original sequence, NOT the zero-padded length

Resolution BW in Hz = $B\frac{f_s}{N}$

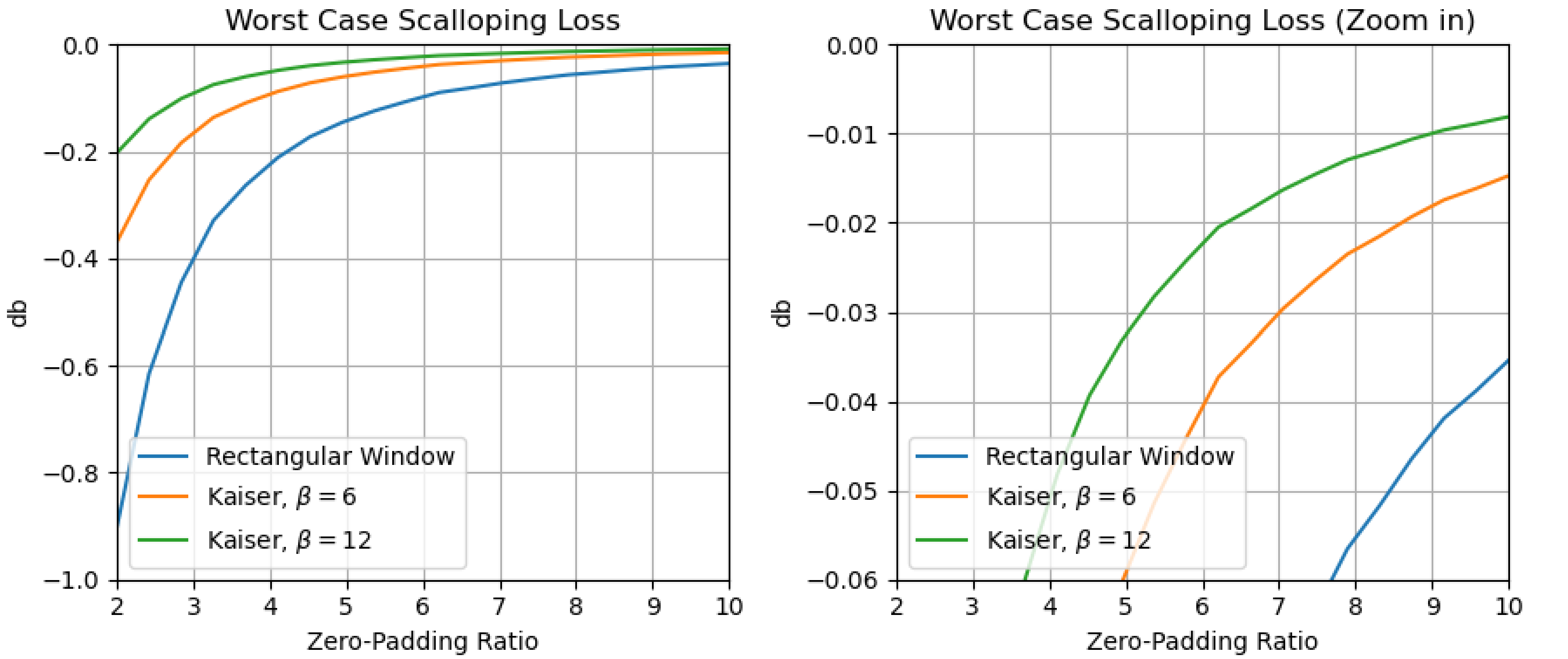


FFT Spectrum if not Windowed



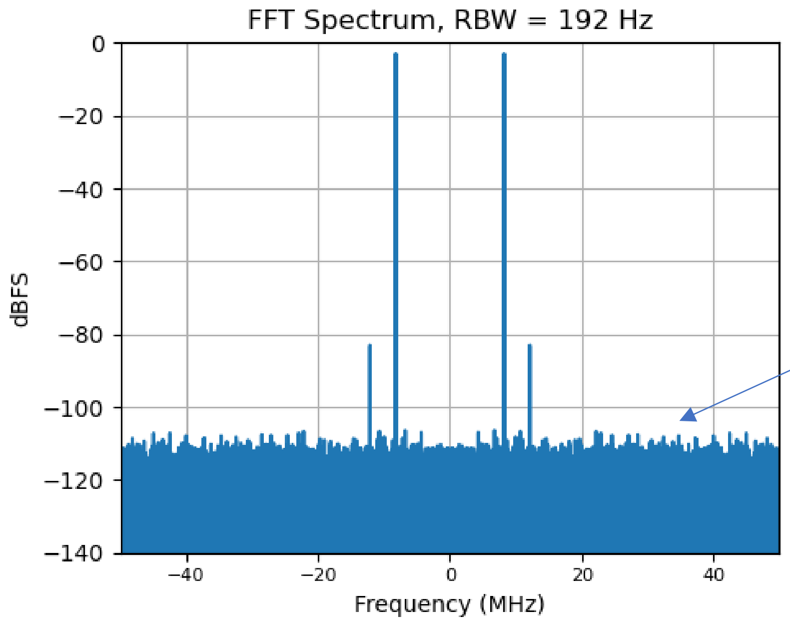


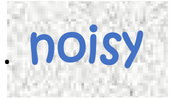
Scalloping Loss



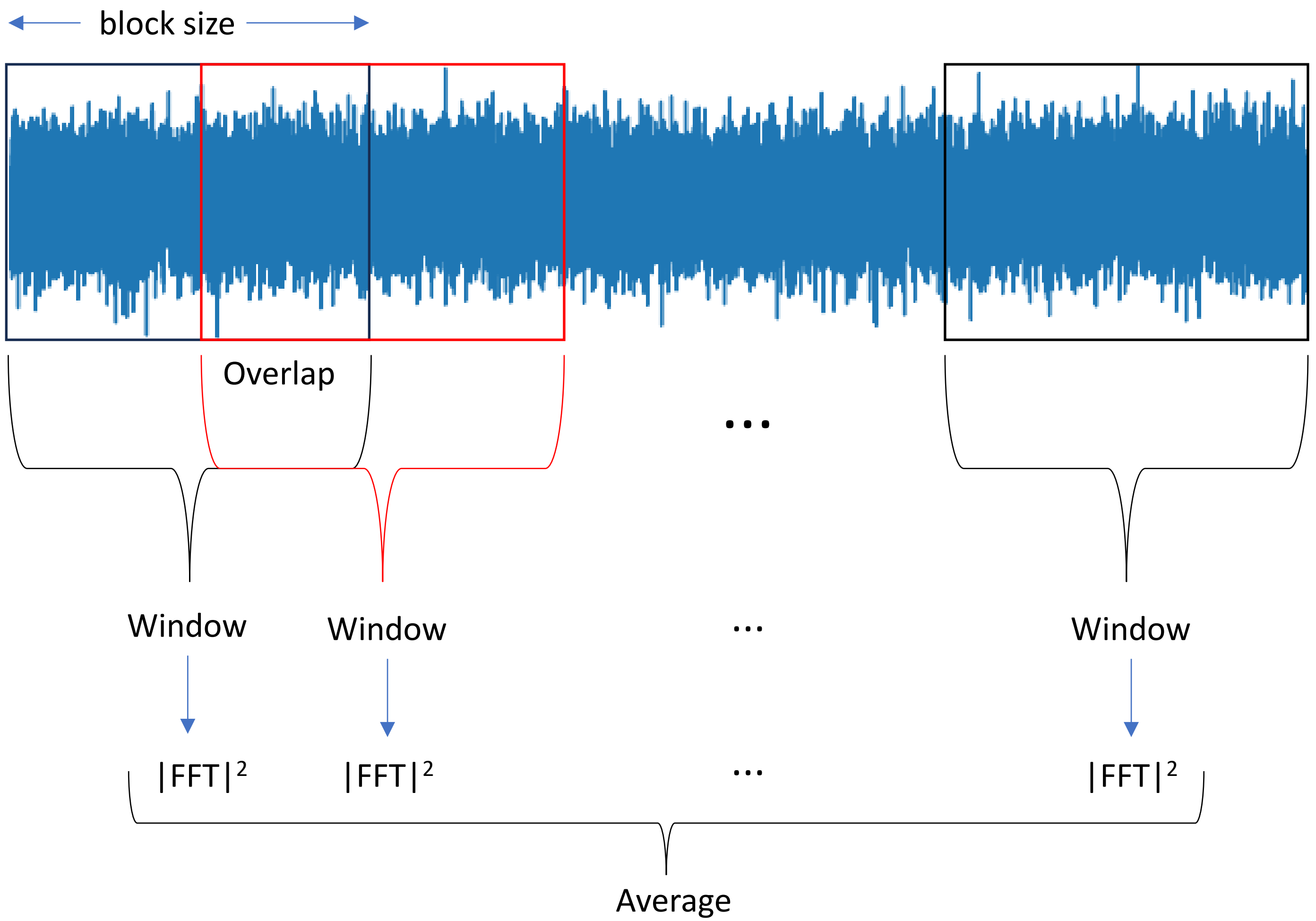
Power Measurement of Noise

Motivation for using Welch

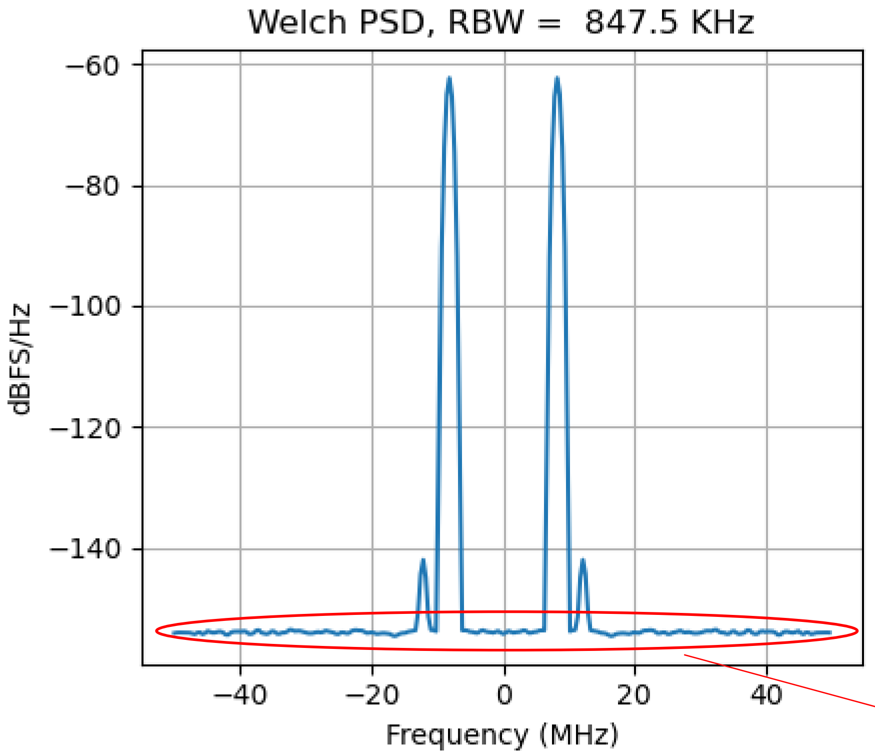


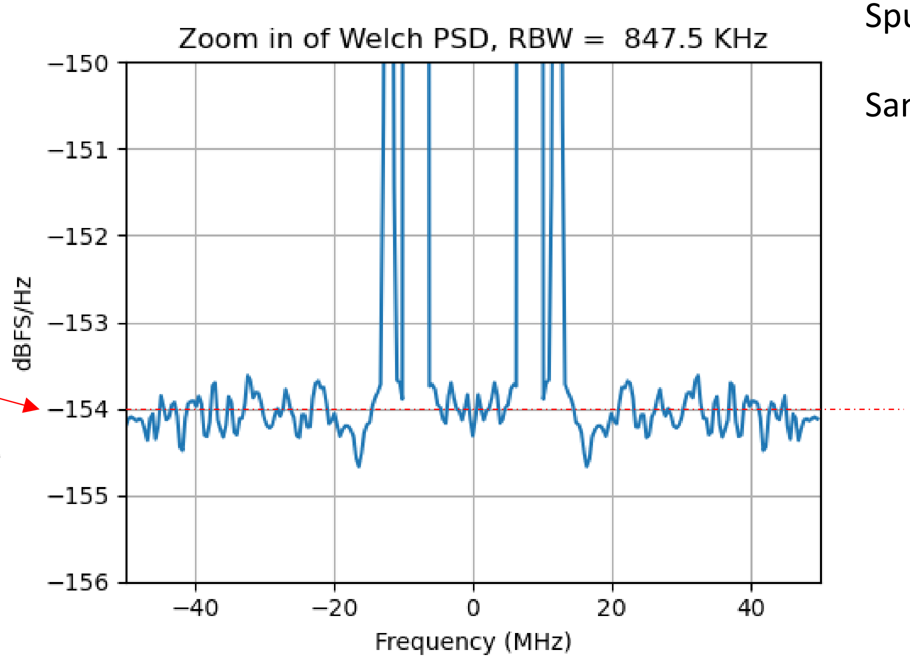
Noise in the FFT Spectrum is ... 

Welch Power Spectral Density for Noise



Results of Welch Power Spectral Density





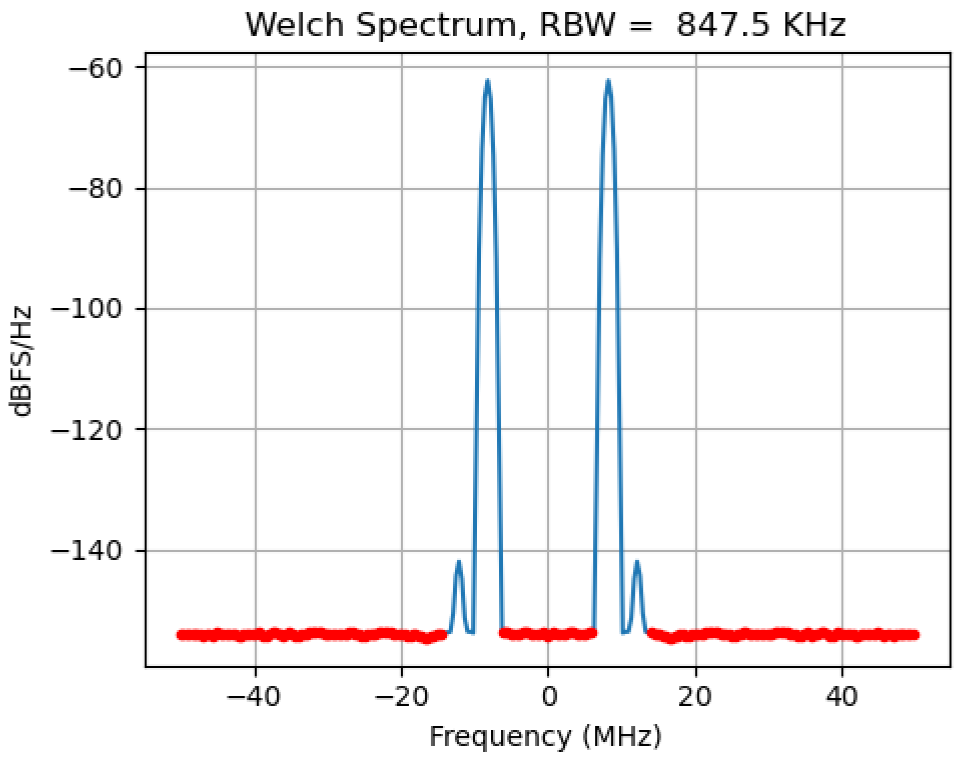
Test waveform:

$$x(t) = A_1\cos(2\pi f_1 t) + A_2\cos(2\pi f_2 t)$$

Sampling Rate: f_s = 100 MHz
Total number of samples: N=1,048,576
Primary tone: f_1=8.2163185 MHz, A_1=1
Spur: f_2=12.13 MHz, A_2 = 1e-4

Sampled and quantized to 12 bits to be x[n]

Results of Power Spectral Density

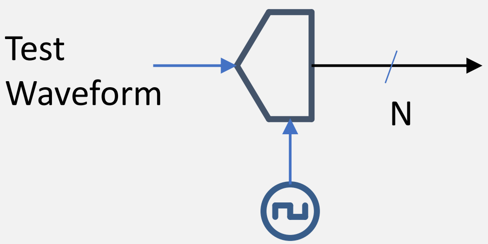


Extract the peaks and average the noise:

Result: -154.06 dBFS/Hz

ADC ENOB Example:

(Perfect) 12 bit A/D Converter



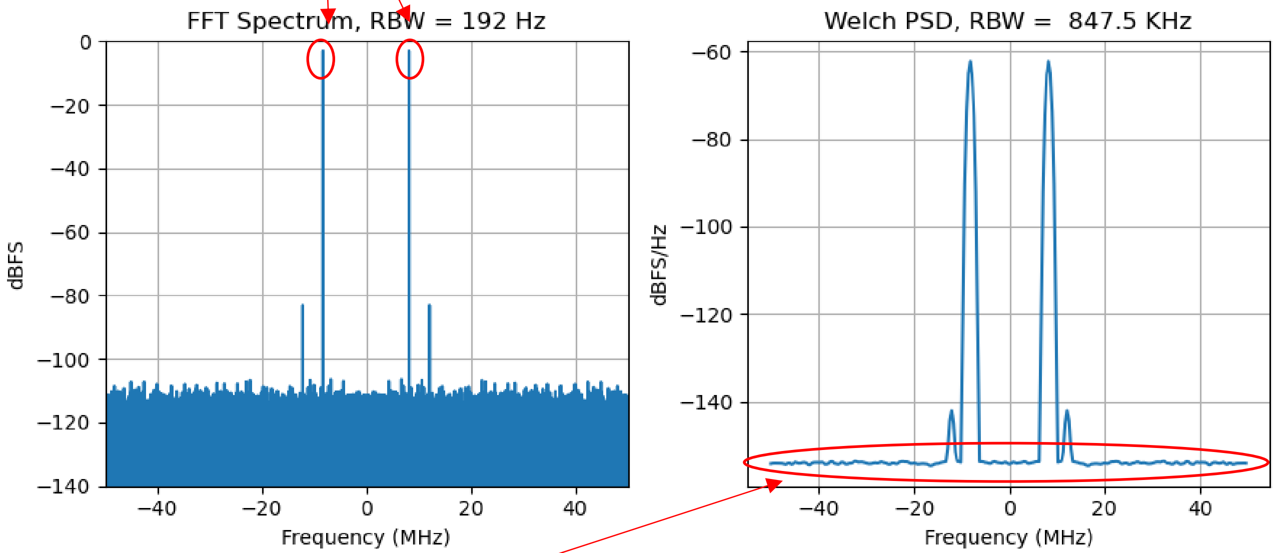
SNR = 6.02 dB/bit + 1.76 dB

SNR = 0 dBFS-(-74.06 dBFS) = 74.06 dB

ENOB = (74.06 -1.76)/6.02 = 12.01 bits

From FFT Spectrum: Total Signal Power = 0 dBFS (-3.01 dBFS as two complex tones)

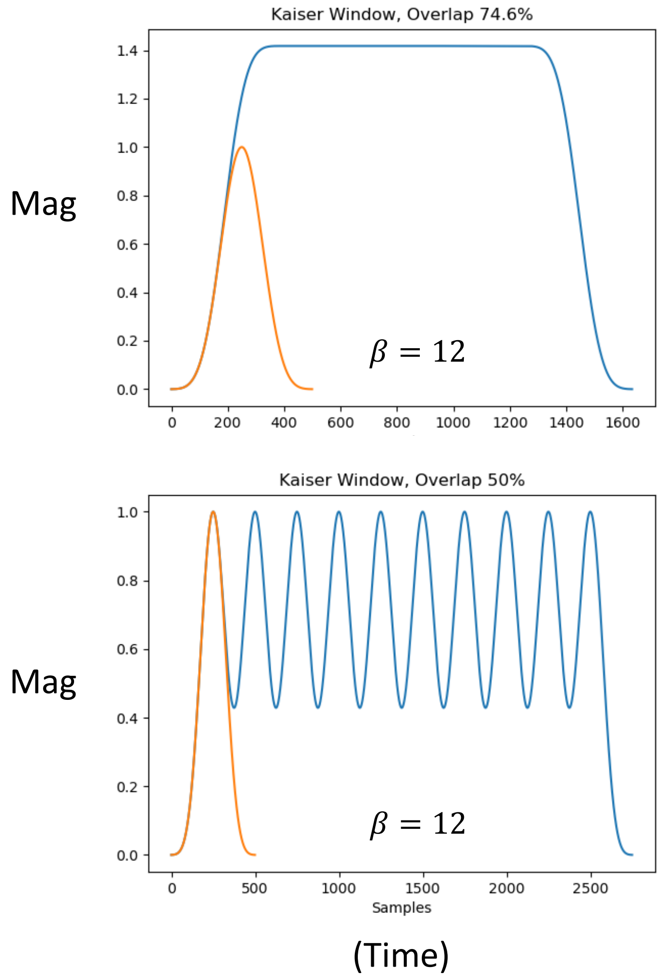
***One** Sine Wave = **Two** Complex Tones!*



From Welch PSD:
Total noise power -154.06 dBFS/Hz spread over 100 MHz
= -154.06 dBFS/Hz + $10\log_{10}(100e6)$
= -74.06 dBFS

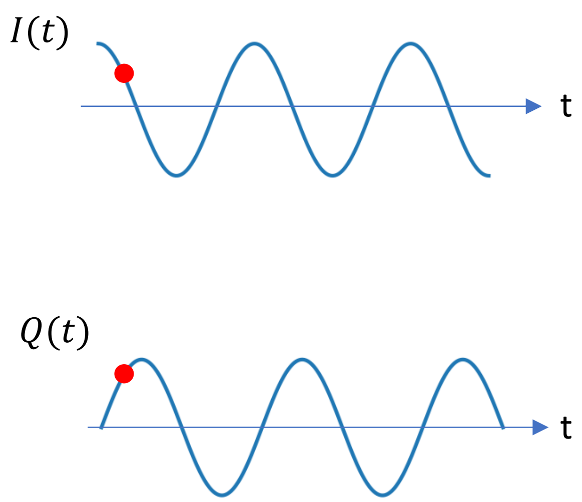
Recommended Overlap for Windows

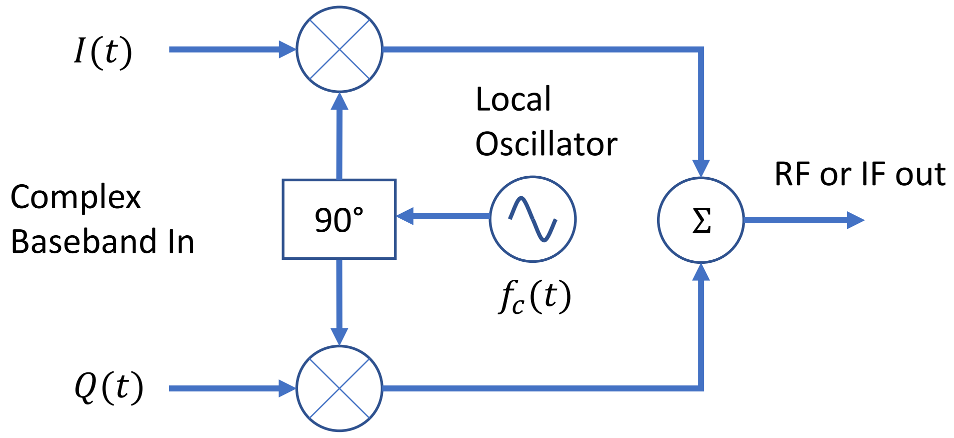
Window	Optimum Overlap
Hann	0.5
Kaiser, $\beta = 6$	0.637
Kaiser, $\beta = 8$	0.692
Kaiser, $\beta = 10$	0.735
Kaiser, $\beta = 12$	0.746
Kaiser, $\beta = 14$	0.768
Kaiser, $\beta = 16$	0.772

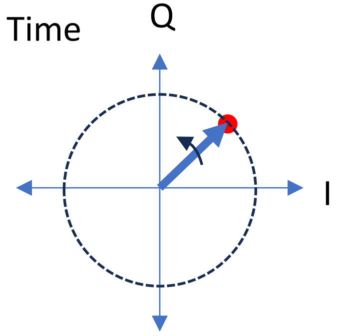


SNR using EVM and Correlation

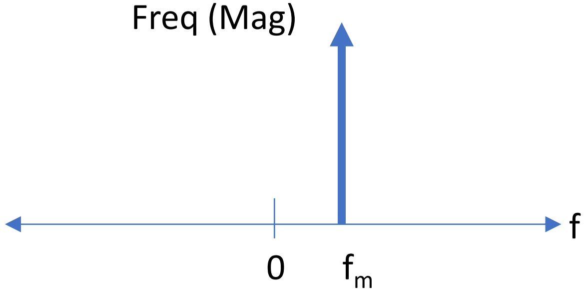
Brief "IQ" Intro



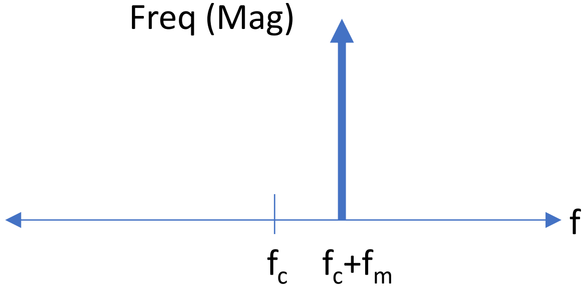




Baseband Spectrum



Passband Spectrum



Correlation Coefficient

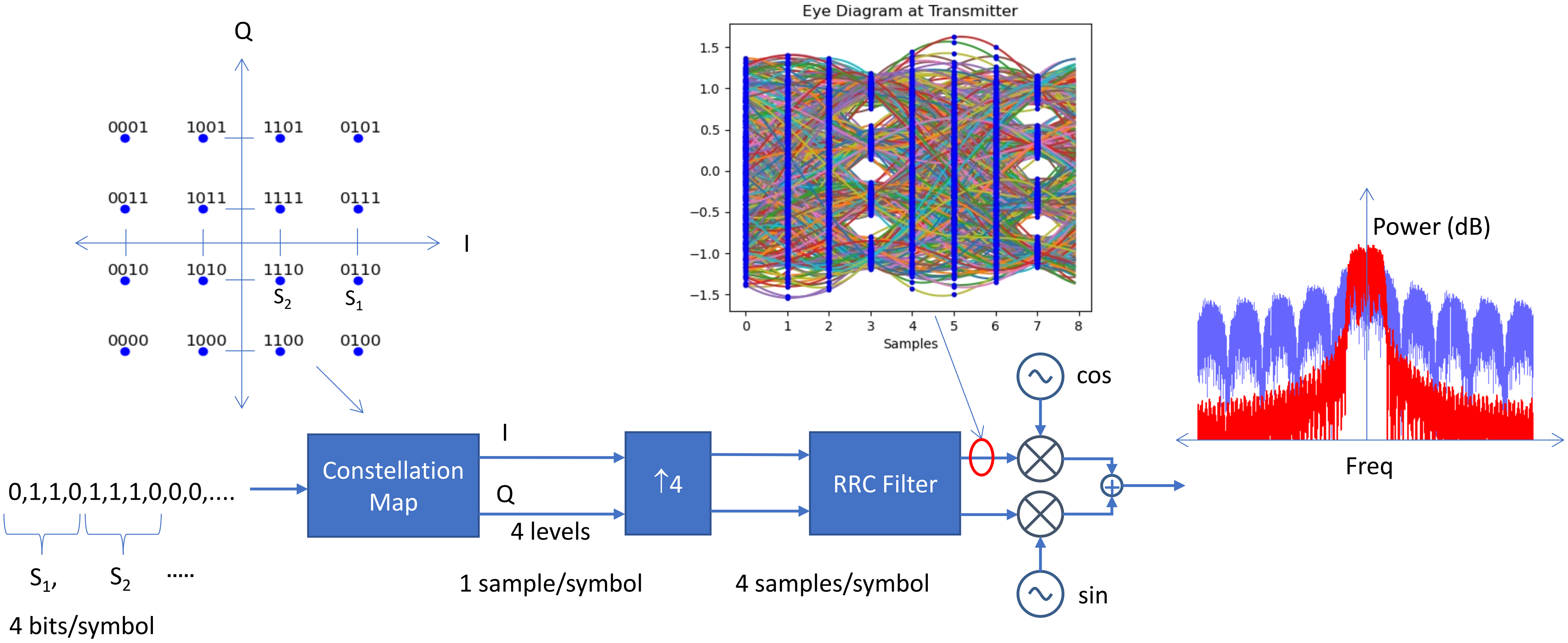
$$\rho = \frac{\mathrm{Cov}(V, \hat{V})}{\sigma_V \sigma_{\hat{V}}}$$

$$\rho = \frac{\frac{1}{N}\sum\left((V - \bar{V})\left(\hat{V} - \bar{\hat{V}}\right)^*\right)}{\sqrt{\frac{1}{N}\sum|V - \bar{V}|^2}\sqrt{\frac{1}{N}\sum\left|\hat{V} - \bar{\hat{V}}\right|^2}} = \frac{\sum\left((V - \bar{V})\left(\hat{V} - \bar{\hat{V}}\right)^*\right)}{\sqrt{\sum\sum|V - \bar{V}|^2}\sqrt{\sum\left|\hat{V} - \bar{\hat{V}}\right|^2}}$$

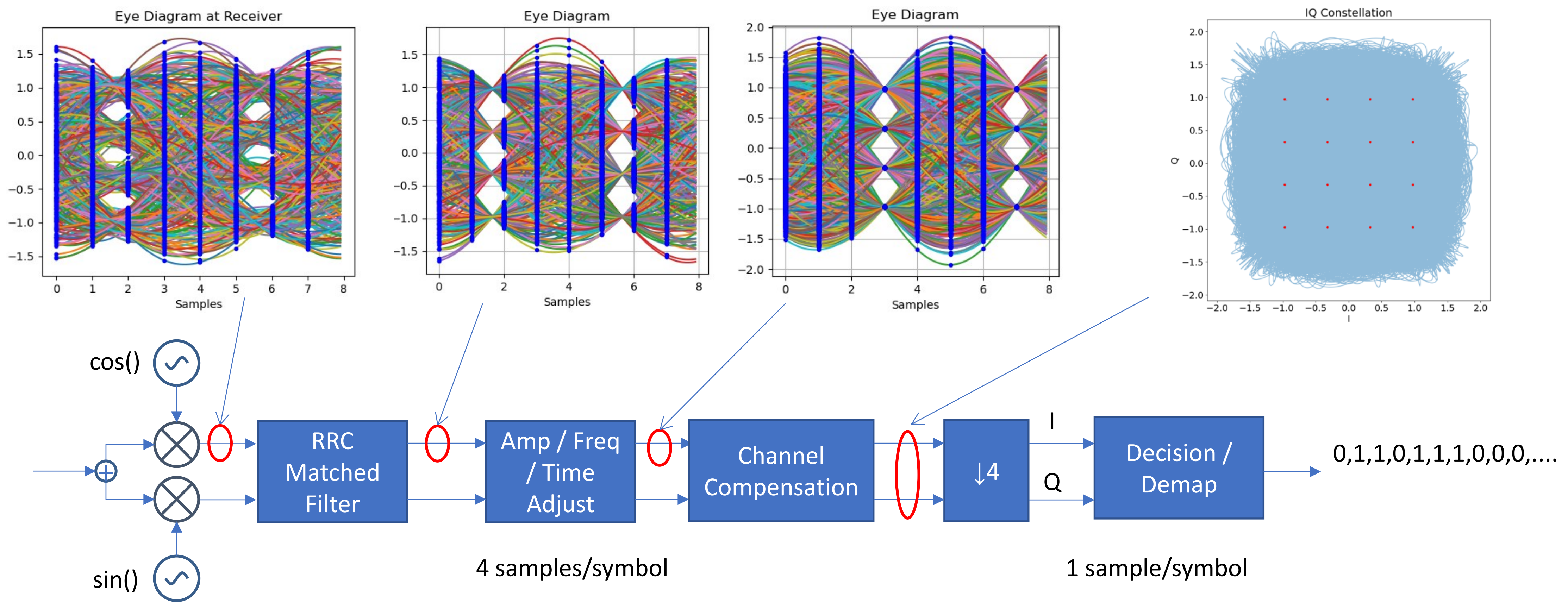
If zero mean: $$\rho = \frac{\sum\left(V\hat{V}^*\right)}{\sqrt{\sum|V|^2}\sqrt{\sum\left|\hat{V}\right|^2}}$$

$$\mathbf{SNR_{dB} = 10\log_{10}\frac{|\rho|^2}{1 - |\rho|^2}}$$

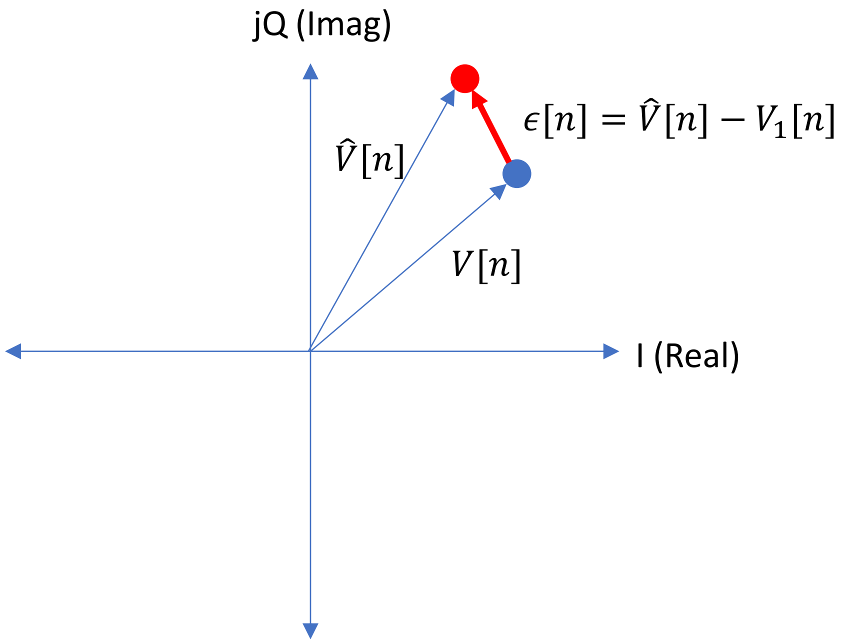
Example Waveform: 16-QAM Transmitter



Example Waveform: 16-QAM Receiver



EVM: Error Vector Magnitude



one error vector at sample index n in time

$$EVM = \frac{\sqrt{\frac{\sum \left| V[n] - \hat{V}[n] \right|^2}{N}}}{S} \times 100\%$$

S is "scale"
when $S = V_{rms}$ then SNR = 1/(EVM)^2

in dB: **SNR_{dB} = -20Log(EVM)**

16-QAM Waveform Parameters

Total number of symbols:	262,144
Symbol Rate (R):	10 MHz
Headroom:	8 dB below Full Scale
Number of bits:	8 bits I and Q
Sampling Rate (f_s):	40 MHz
RRC impulse response duration:	30 symbols
RRC roll-off factor:	0.2

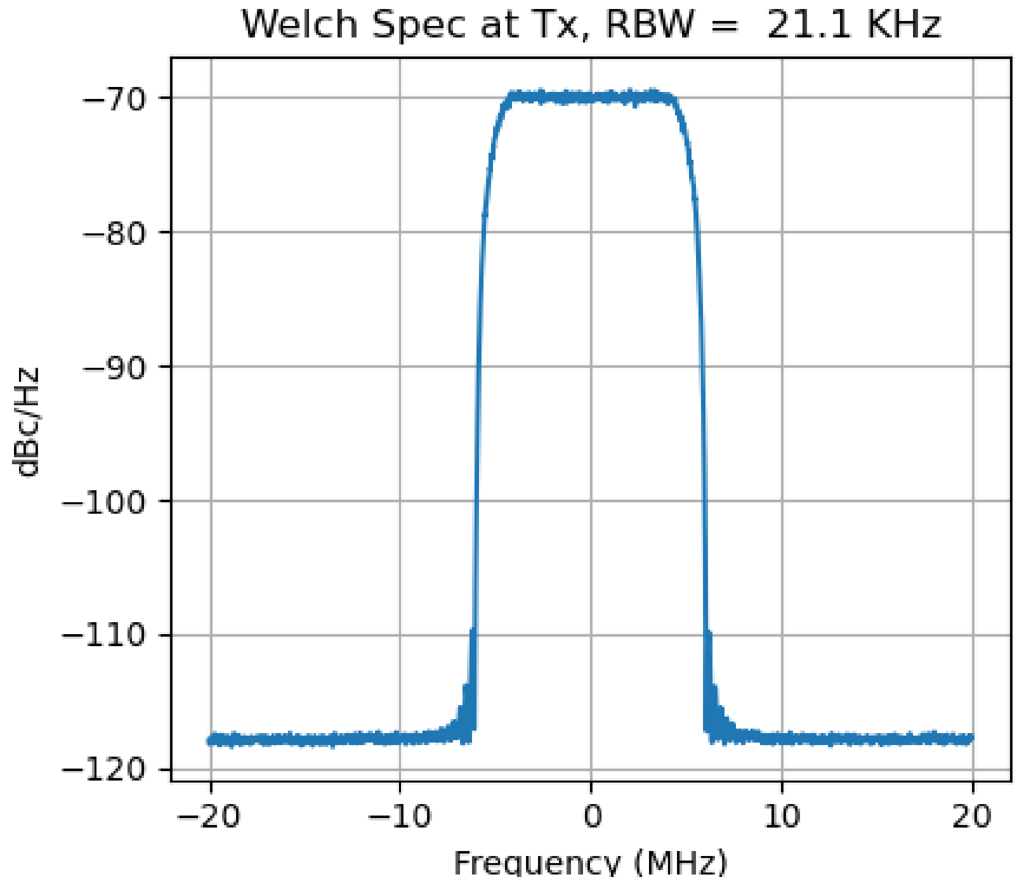
Predicted SNR as limited by quantization noise:

SNR (from Full Scale) = 6.02 dB/bit + 1.76 dB = (6.02)(8) + 1.76 = 49.92 dB

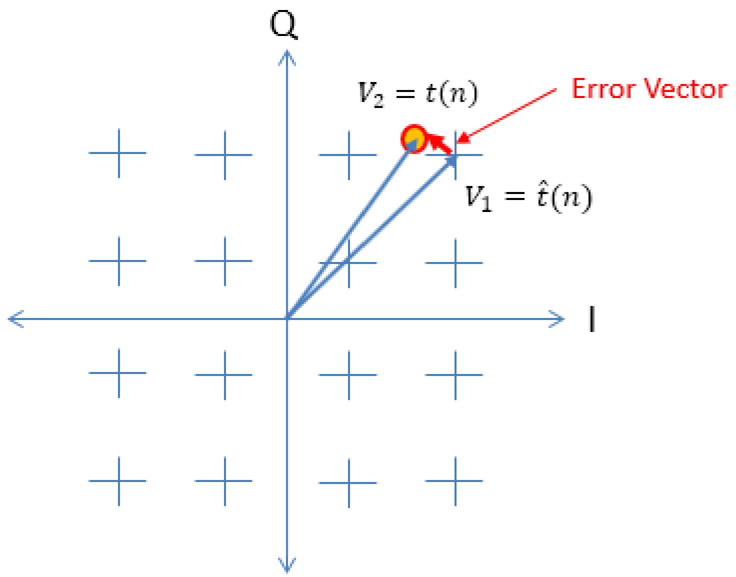
SNR after Headroom = 49.92 dB – 8 dB = 41.92 dB

Potential SNR increase due to oversampling = 10log(f_s/R) = 6.02 dB

SNR = 41.92 dB + 6.02 dB = 47.94 dB



SNR using EVM



$$EVM = \frac{\sqrt{\frac{\sum|t_n - \hat{t}_n|^2}{N}}}{S} \times 100\%$$

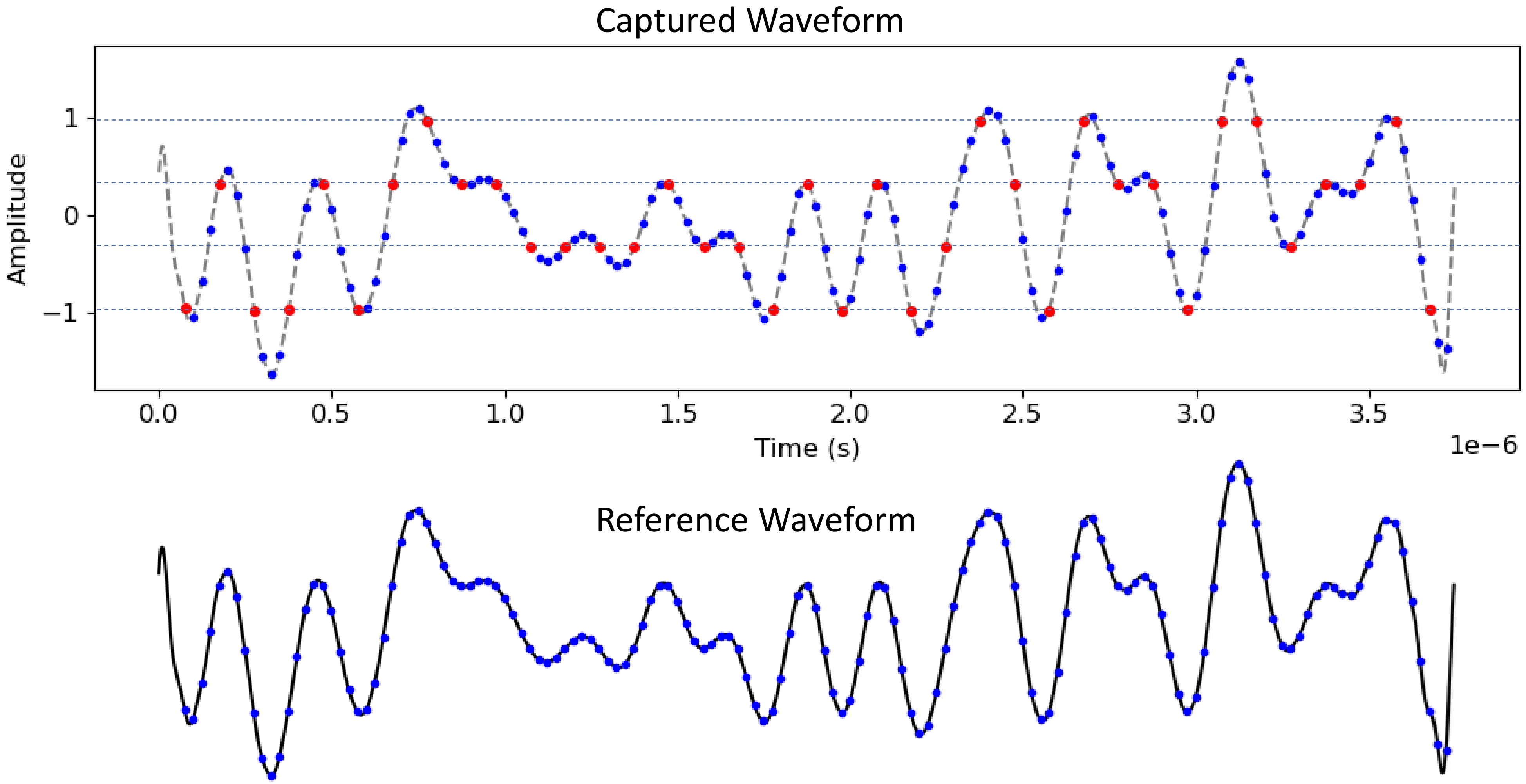
When S = RMS of all possible symbols in the constellation

then:

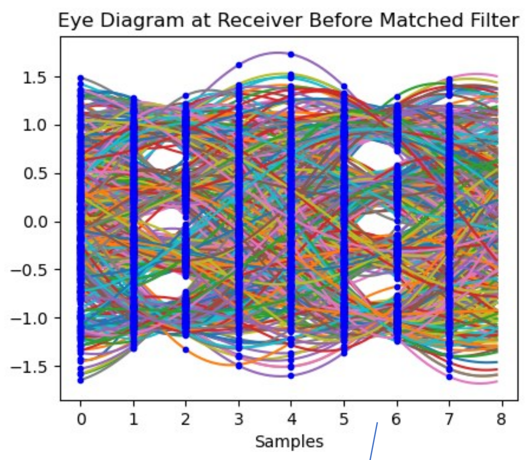
$1/EVM^2$ = SNR

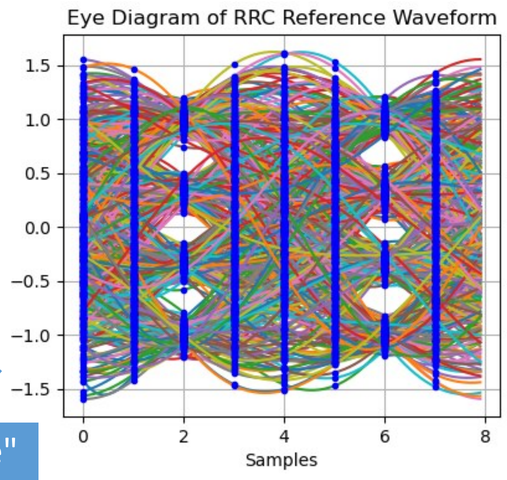
in dB: SNR (dB) = -20Log(EVM)

"All Sample" EVM

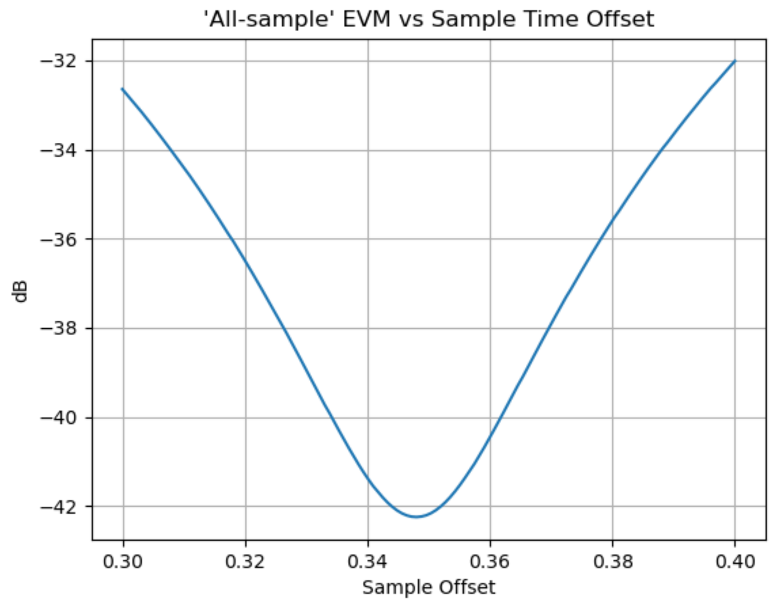


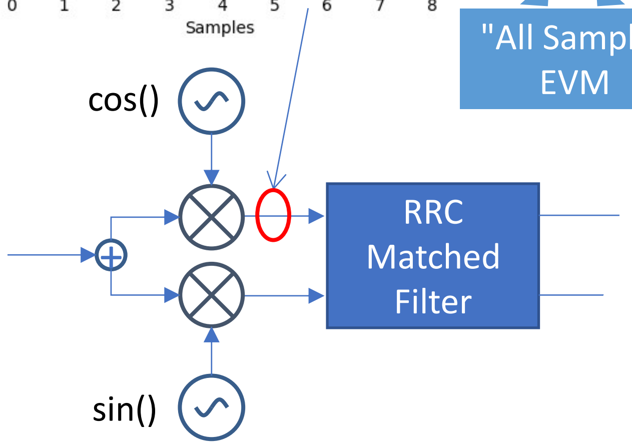
EVM for 16-QAM prior to Matched Filter





"All Sample" EVM





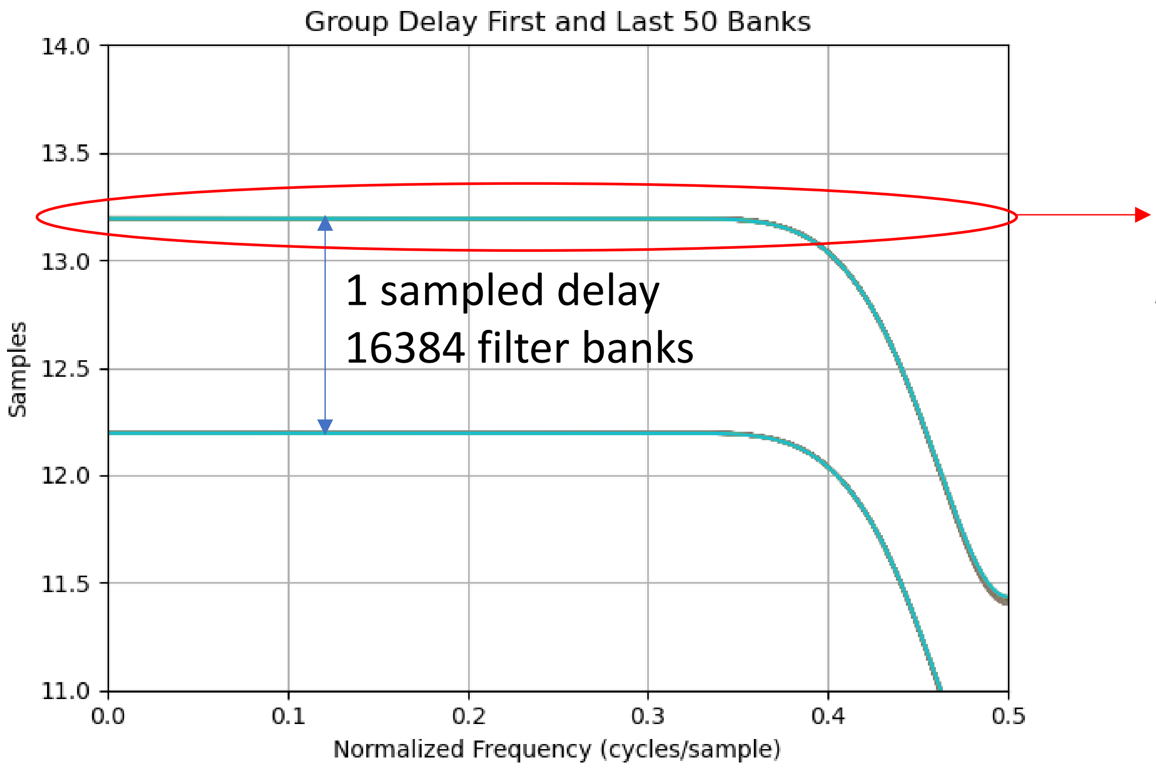
Alignment of Mag, Freq/Phase, and Time

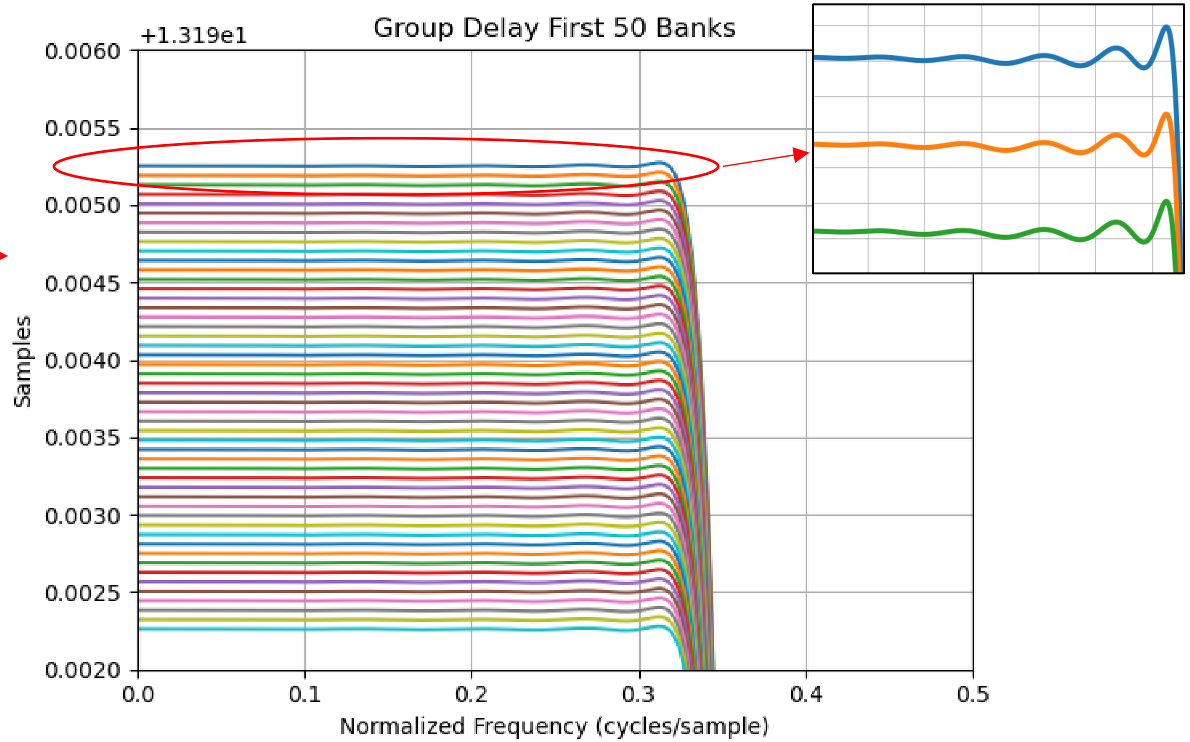
Magnitude: Normalize by the standard deviation

Freq / Phase: Use correlation to get phase, frequency is slope of phase

Time: Use correlation for course alignment, polyphase for fine adjust

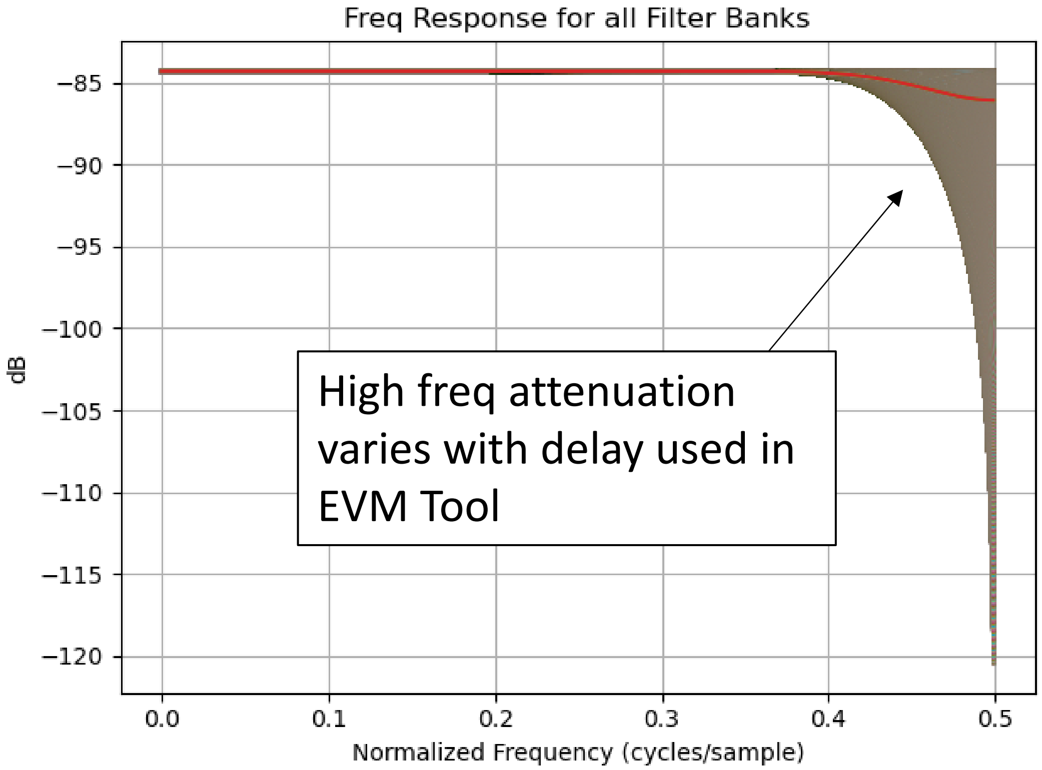
Time Alignment with Polyphase Filter Bank

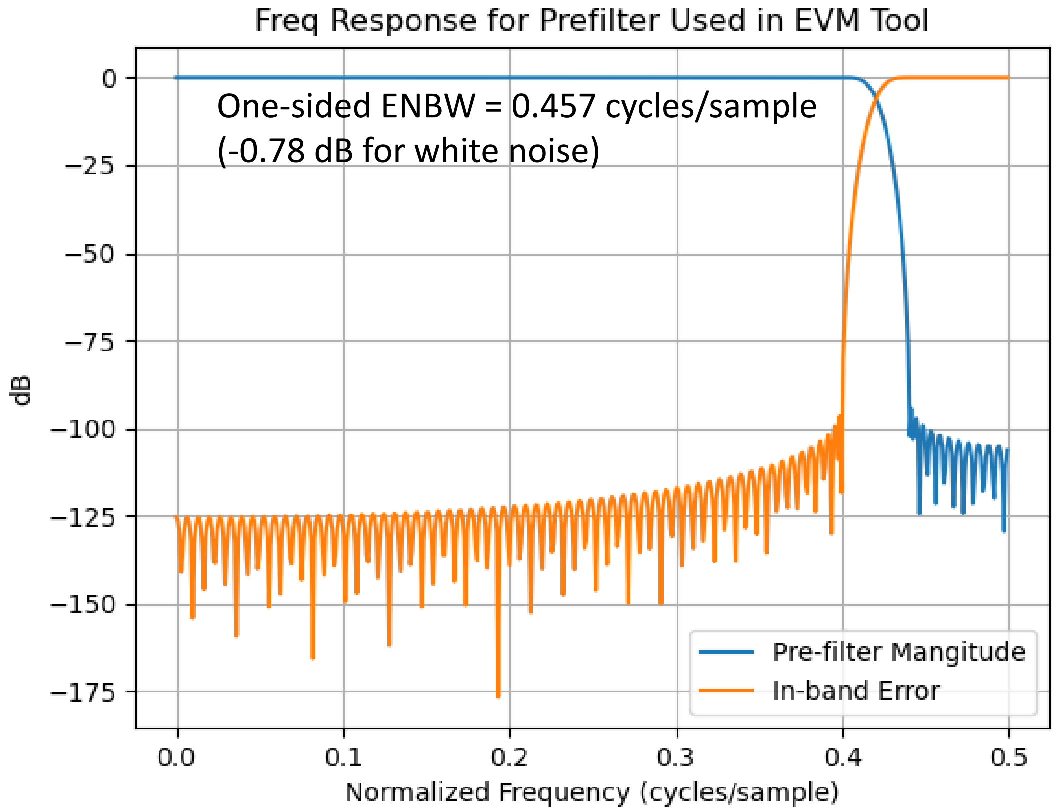




Each Bank is a 27 tap FIR filter

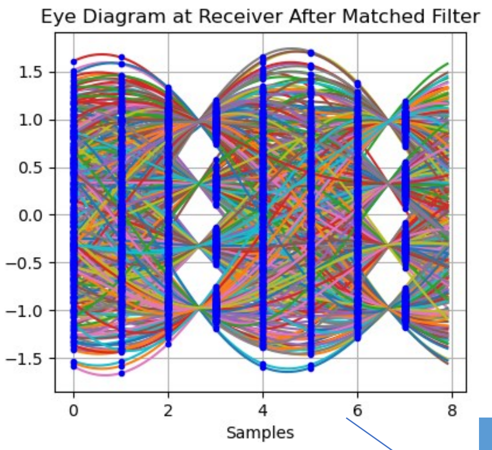
Magnitude Response of Polyphase Filter Bank

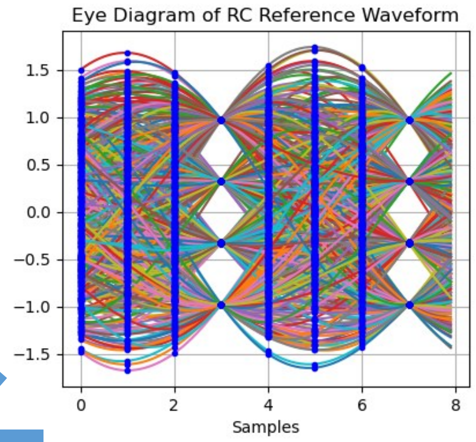


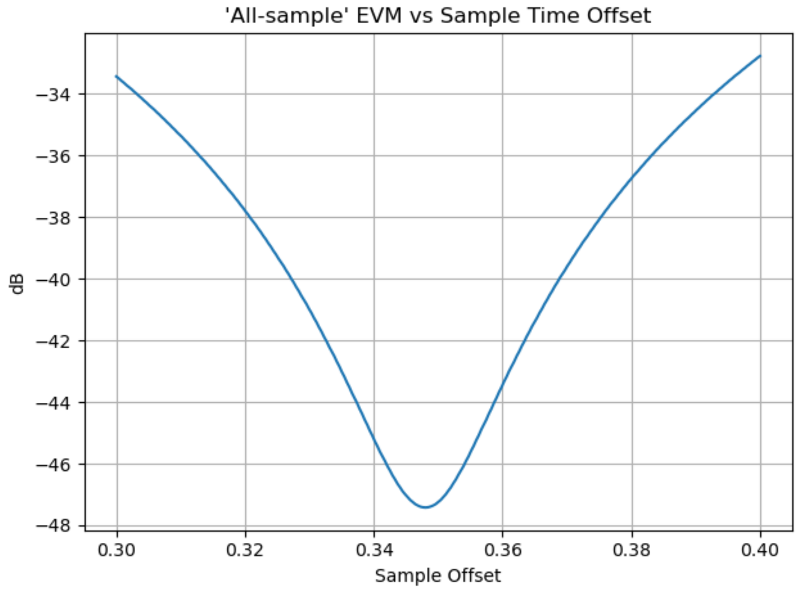


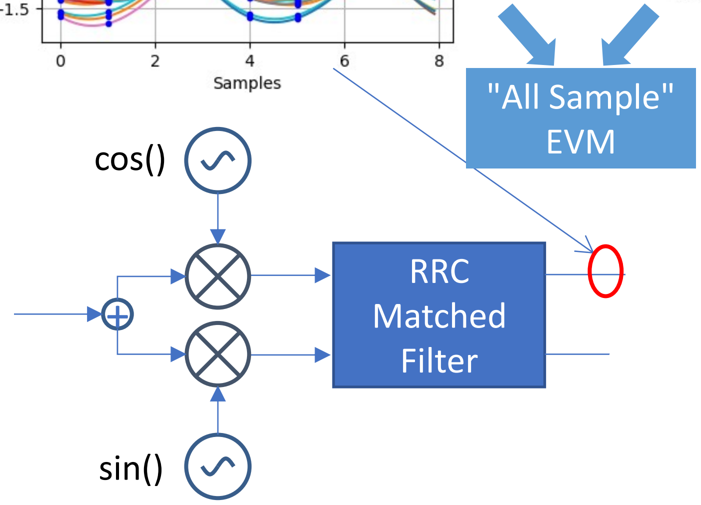
Prefilter for waveform and reference used in EVM Tool to eliminate delay variability in EVM result

EVM for 16-QAM after Matched Filter

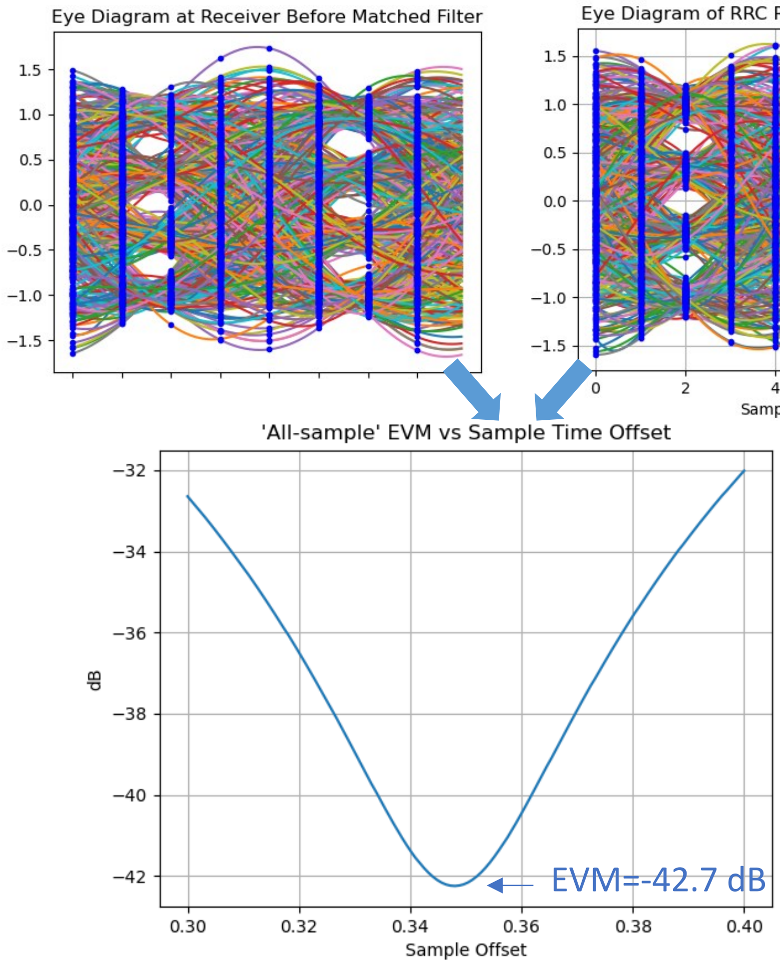


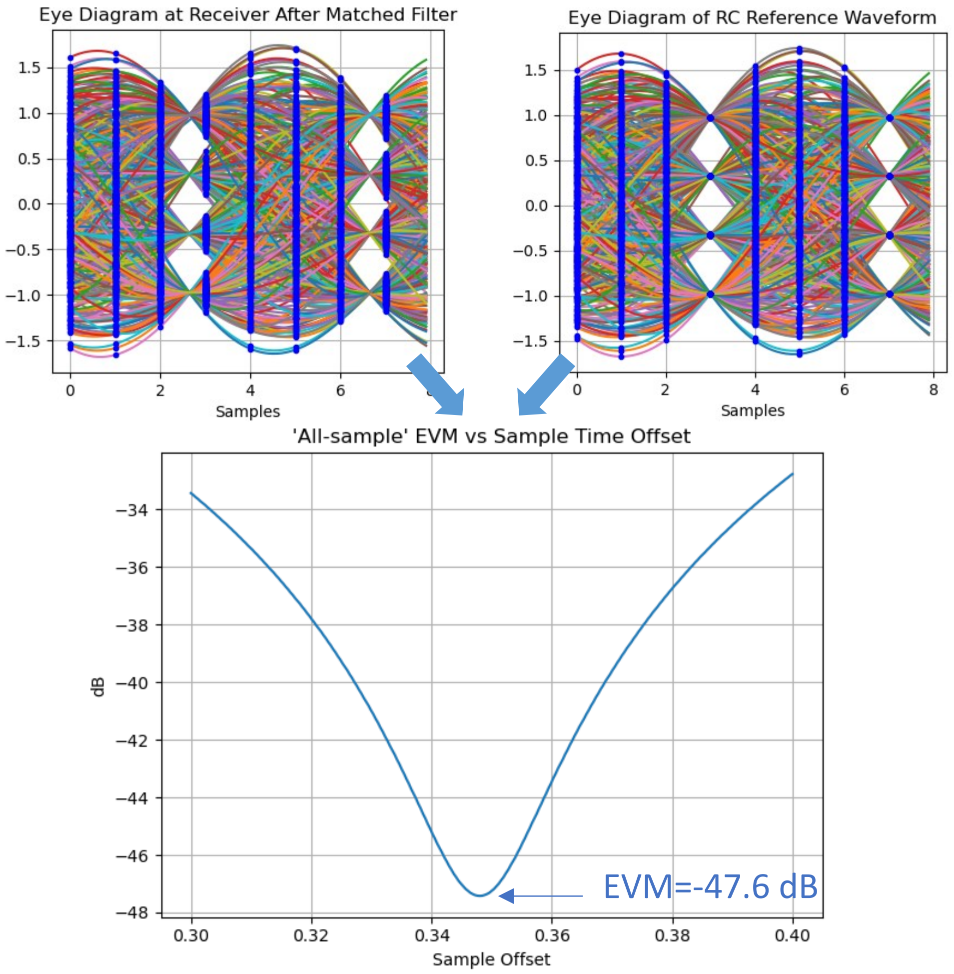






Summary of EVM Receiver Measurements





EVM Tool – 16 QAM

Rx Waveform Measurement Before Matched Filter:

```
In: tools.evm(rx_before_mf, reference_rrc)
Out: (0.007323, 0.347961, 4.519621e-07)
In: tools.db(0.007323)
Out: -42.7062
```

EVM, time offset (samples), and phase offset (radians)

-42.7 dB

Rx Waveform Measurement After Matched Filter:

```
In: tools.evm(rx_after_mf, reference_rc)
Out: (0.004149, 0.348022, 8.341614e-08)
In: tools.db(0.004149)
Out: -47.64107
```

-47.6 dB

Measured EVM at Decision Samples Only: -47.9 dB

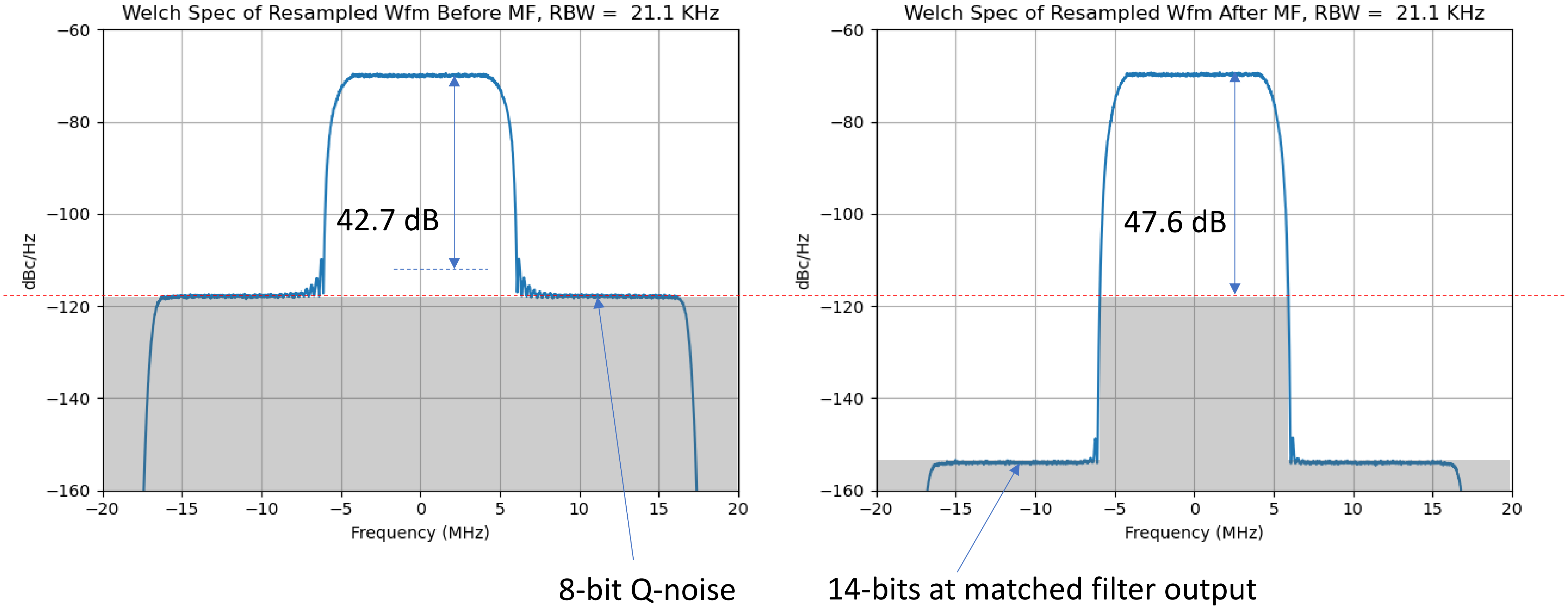
Predicted SNRs – 16 QAM

16 QAM SNR Prior to Matched filter
Quantization: 8 bits I and Q
AGC: -8.0 dBFS
Raised Cosine Roll-off Factor (α): 0.2
One-sided ENBW of Prefilter in EVM Tool: 0.457 cycles/sample
Sampling Rate (R): 4 samples / symbol

SNR using 6.02 dB/bit + 1.76 dB: 49.92 dBFS
SNR after AGC Back-off: 49.92 dBFS - 8.0 dB = 41.92 dB
Prefilter noise reduction: 20Log(2 x ENBW) = -0.78 dB
Predicted "All-Sample" EVM from EVM Tool: -41.92 dB - 0.78 dB = -42.70 dB

16 QAM SNR After Matched filter
Potential SNR increase due to oversampling: $20\log_{10}(R)$ = 6.02 dB
Predicted EVM after 2nd RRC fitler (removes out of band noise): -41.92-6.02 =-47.94 dB

Measured SNRs and Welch Spectrums



SNR using the Correlation Coefficient

Step 1: Remove time, frequency, phase offsets.
Step 2: Remove the mean.
Step 3: Normalize by the standard deviation to get waveform **x[n]** and reference **r[n]**.

Compute the normalized correlation coefficient using same length N for x[n] and r[n]:

$$\rho = \frac{1}{N}\sum_{N} x[n]r^*[n]$$

$$\mathbf{SNR_{dB} = 10\log_{10}\frac{|\rho|^2}{1-|\rho|^2}}$$

SNR from EVM and Correlation – 16 QAM

Using EVM:

$$EVM = \sqrt{\frac{\sum_N |x[n] - r[n]|^2}{N}}$$

$$SNR_{dB} = -20\log_{10}(EVM)$$

SNR after Matched Filter for 16-QAM Example:
All samples: **-47.6 dB**
Decision samples only: **-47.9 dB**

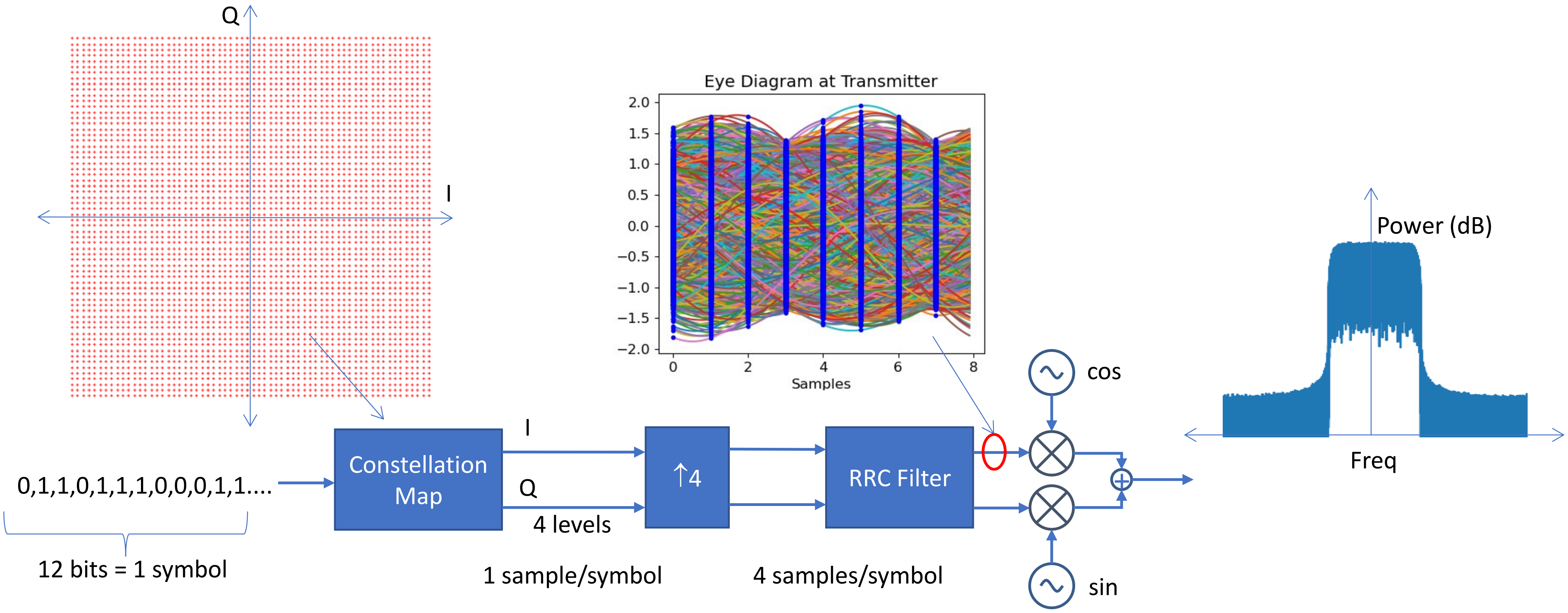
Using Correlation Coefficient:

$$\rho = \frac{\sum_N x[n]r^*[n]}{N}$$

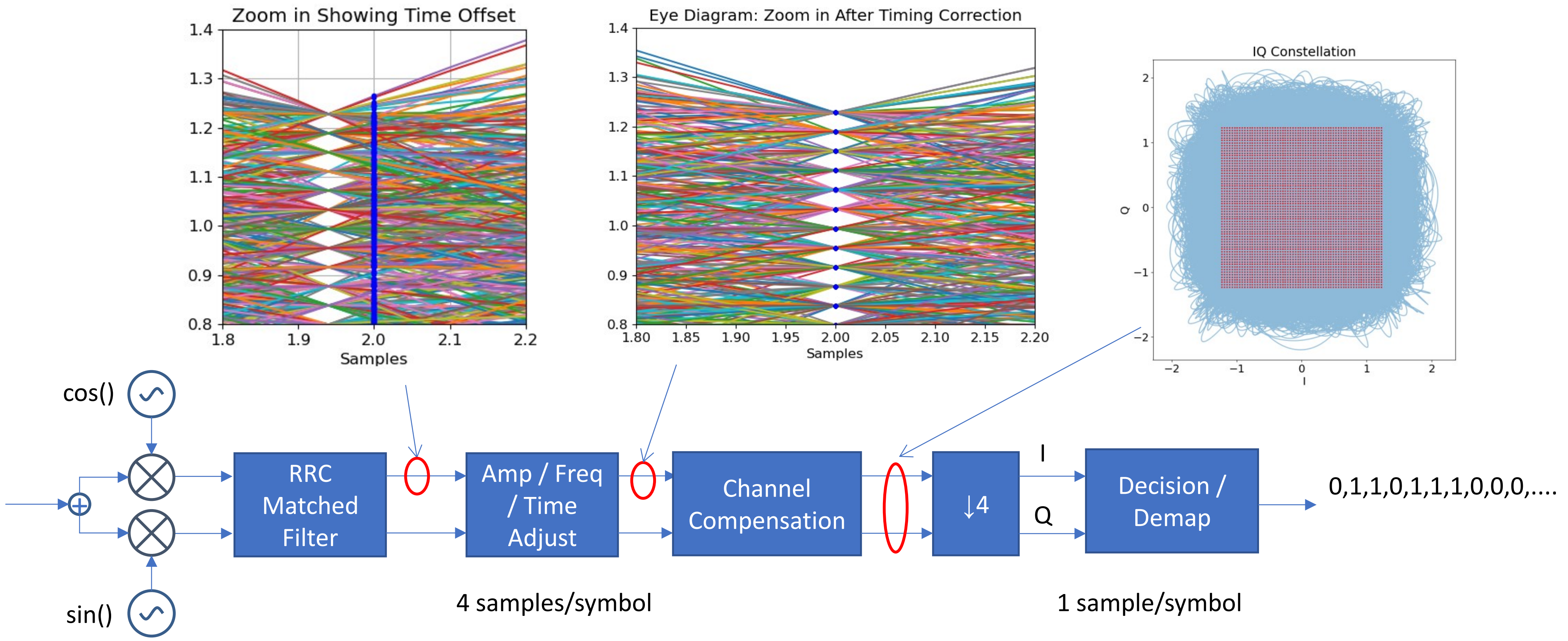
$$SNR_{dB} = 10\log_{10}\frac{|\rho|^2}{1-|\rho|^2}$$

SNR after Matched Filter for 16-QAM Example:
All samples: **-47.6 dB**
Decision samples only: **-47.9 dB**

Example Waveform: 4096-QAM Transmitter



Example Waveform: 4096-QAM Receiver



4096-QAM Waveform Parameters

Total number of symbols:	262,144
Symbol Rate (R):	1 MHz
Headroom:	8 dB below Full Scale
Number of bits:	15 bits I and Q (SOTA = 15 bits)
Sampling Rate (f_s):	8 MHz
RRC impulse response duration:	400 symbols
RRC roll-off factor:	0.2

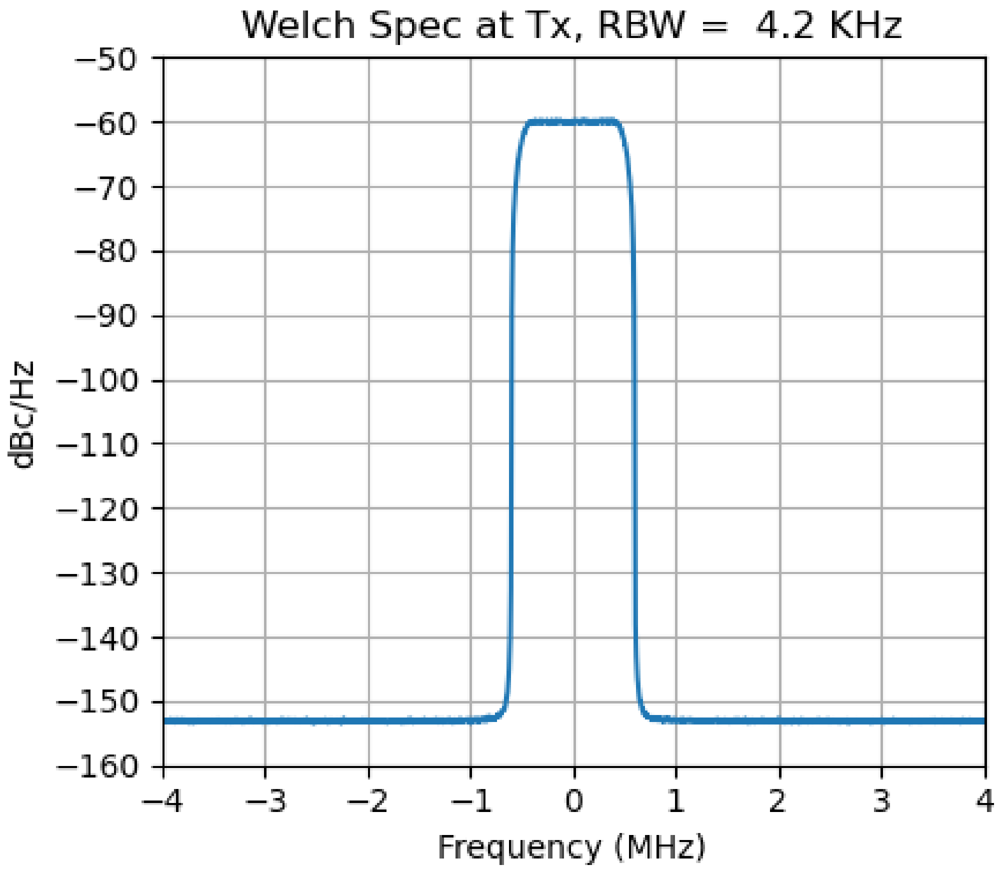
Predicted SNR as limited by quantization noise:

SNR (from Full Scale) = 6.02 dB/bit + 1.76 dB = (6.02)(15) + 1.76 = 92.06 dB

SNR after Headroom = 104.10 dB – 8 dB = 84.06 dB

Potential SNR increase due to oversampling = $10\log(f_s/R)$ = 9.03 dB

SNR = 84.06 dB + 9.03 dB = 93.09 dB



EVM Tool – 4096 QAM

Rx Waveform Measurement Before Matched Filter:

```
In: tools.evm(rx_before_mf, reference_rrc)
Out: (5.689957e-05, 0.059998, 9.174326e-10)
In: tools.db(5.689957e-05)
Out: -84.8978
```

EVM, time offset (samples), and phase offset (radians)

-84.9 dB

Rx Waveform Measurement After Matched Filter:

```
In: tools.evm(rx_after_mf, reference_rc)
Out: (2.276750e-05, 0.059998, -4.41151e-08)
In: tools.db(2.276750e-05)
Out: -92.8537
```

-92.9 dB

Measured EVM at Decision Samples Only: -93.0 dB

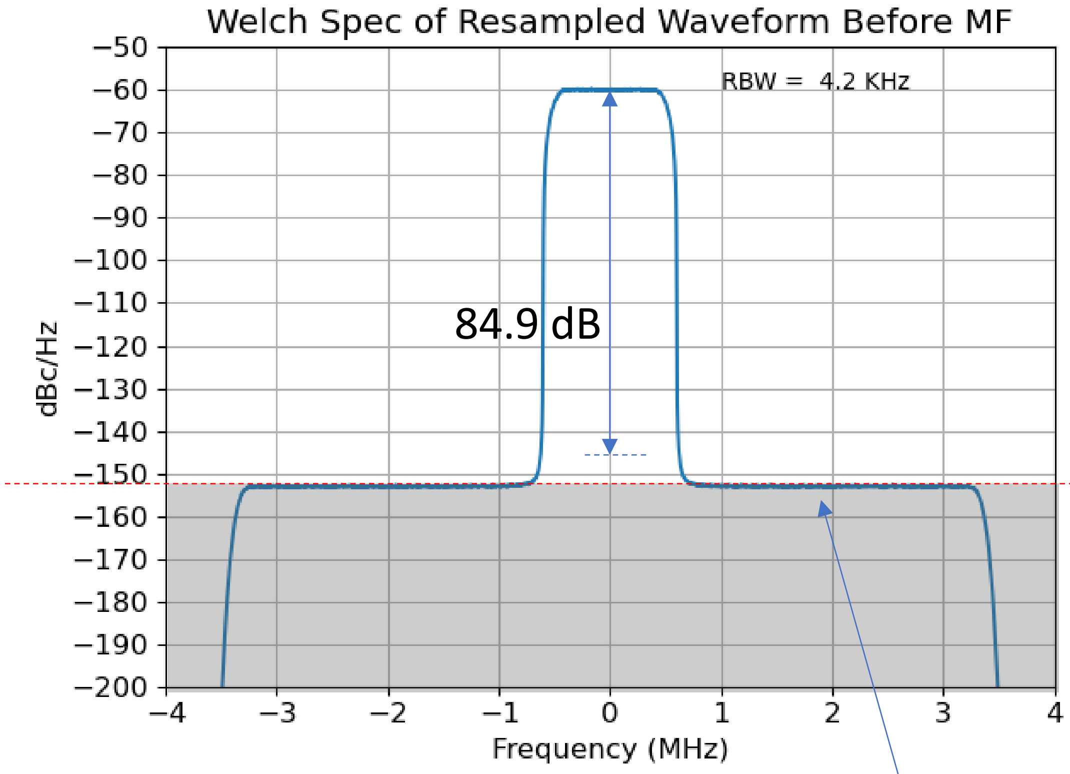
Predicted SNRs – 4096 QAM

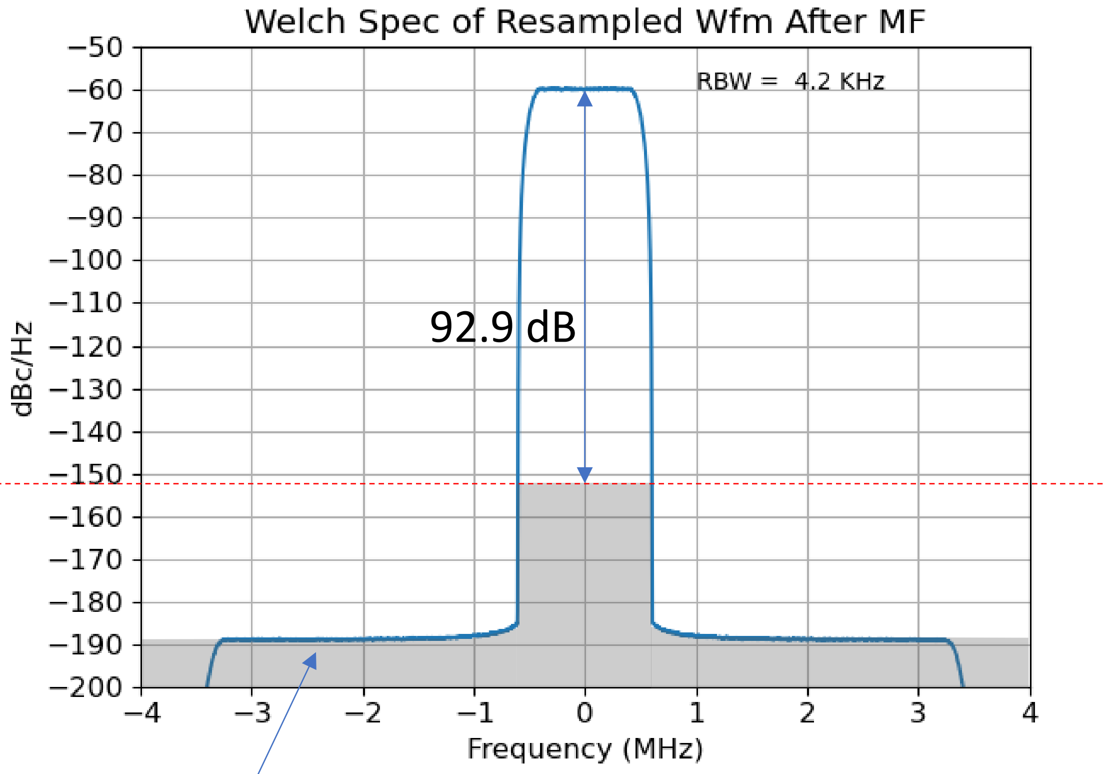
4096 QAM SNR Prior to Matched filter
Quantization: 15 bits I and Q
AGC: -8.0 dBFS
Raised Cosine Roll-off Factor (α): 0.2
One-sided ENBW of Prefilter in EVM Tool: 0.457 cycles/sample
Sampling Rate (R): 8 samples / symbol

SNR using 6.02 dB/bit + 1.76 dB: 92.06 dBFS
SNR after AGC Back-off: 92.06 dBFS - 8.0 dB = 84.06 dB
Prefilter noise reduction: 20Log(2 x ENBW) = -0.78 dB
Predicted "All-Sample" EVM from EVM Tool: -84.06 dB - 0.78 dB = 84.84 dB

4096 QAM SNR After Matched filter
Potential SNR increase due to oversampling: $20\log_{10}(R)$ = 9.03 dB
Predicted EVM after 2nd RRC fitler (removes out of band noise): 84.06+9.03 =93.09 dB

Measured SNRs and Welch Spectrums





15-bit Q-noise

21-bits at matched filter output

SNR from EVM and Correlation – 4096 QAM

Using EVM:

$$EVM = \sqrt{\frac{\sum_N |x[n] - r[n]|^2}{N}}$$

$$SNR_{dB} = -20\log_{10}(EVM)$$

SNR after Matched Filter for 4096-QAM Example:
All samples: **-92.9 dB**
Decision samples only: **-93.0 dB**

Using Correlation Coefficient:

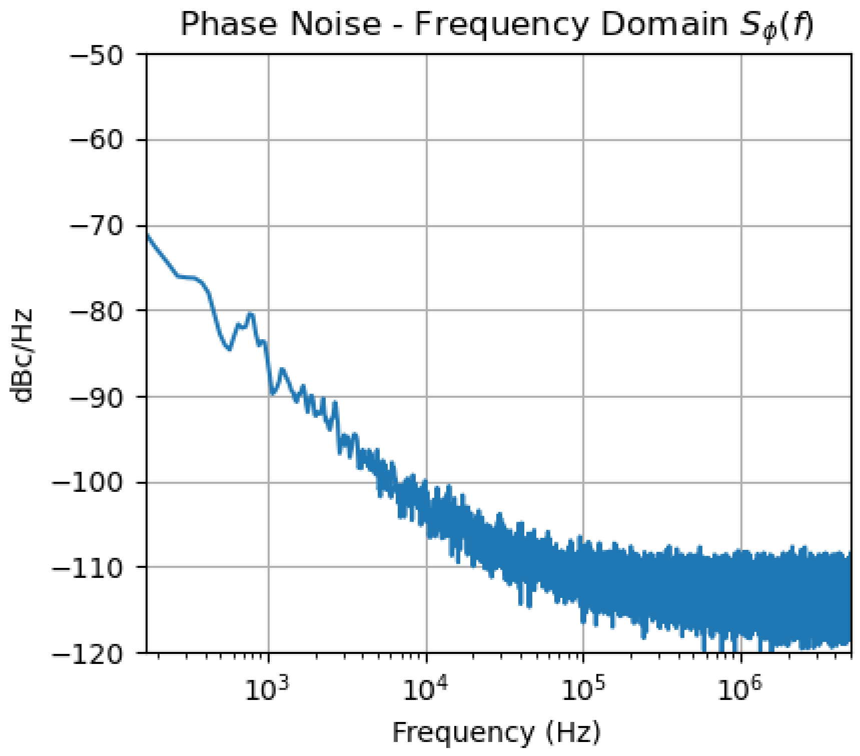
$$\rho = \frac{\sum_N x[n] r^*[n]}{N}$$

$$SNR_{dB} = 10\log_{10}\frac{|\rho|^2}{1 - |\rho|^2}$$

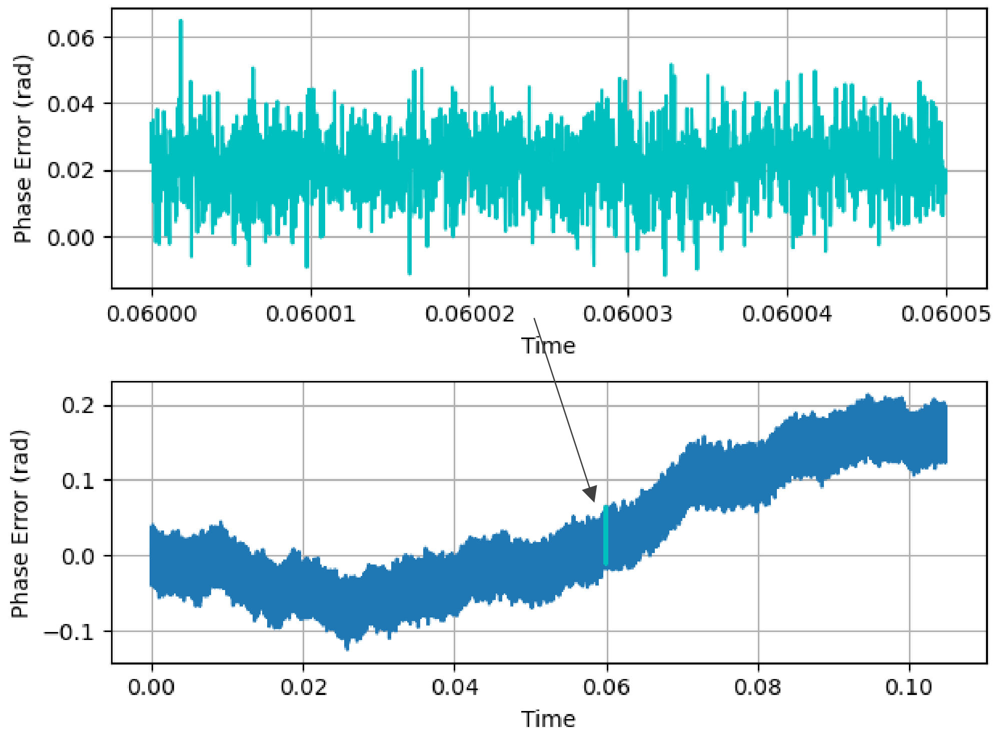
SNR after Matched Filter for 4096-QAM Example:
All samples: **-92.9 dB**
Decision samples only: **-93.0 dB**

Non-Stationary Signals

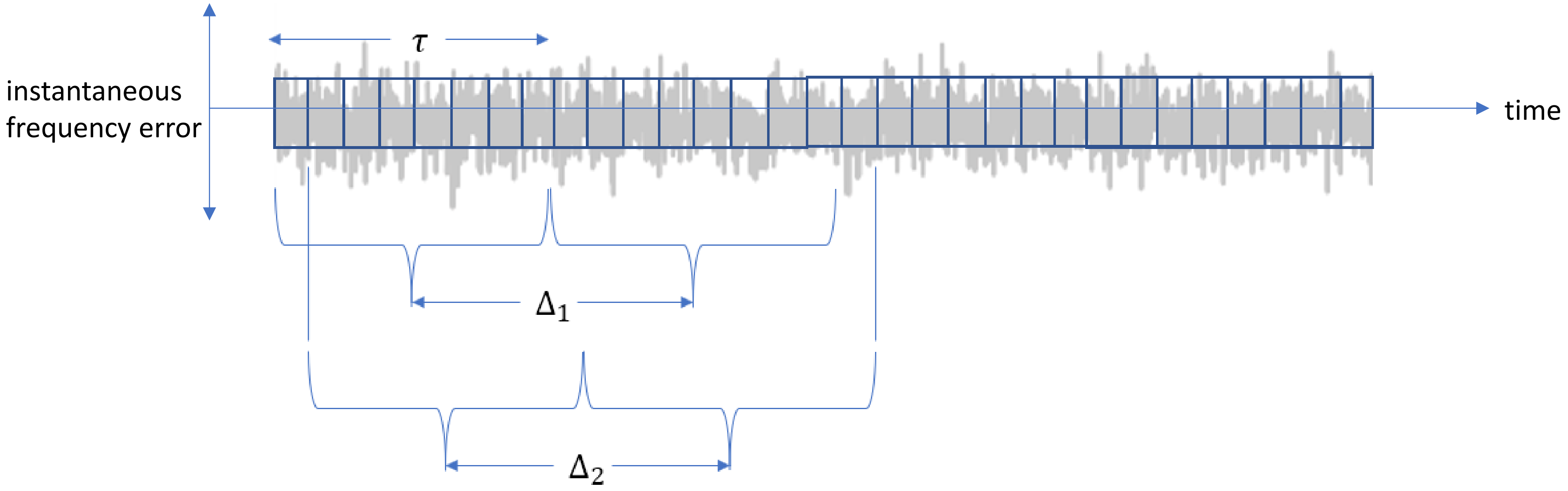
Consideration for Non-Stationary Signals



Phase Noise – Time Domain



Allan Deviation (ADEV)



Overlap ADEV:

Average the frequency error over duration τ

Shift one sample, compute a new average for every shift

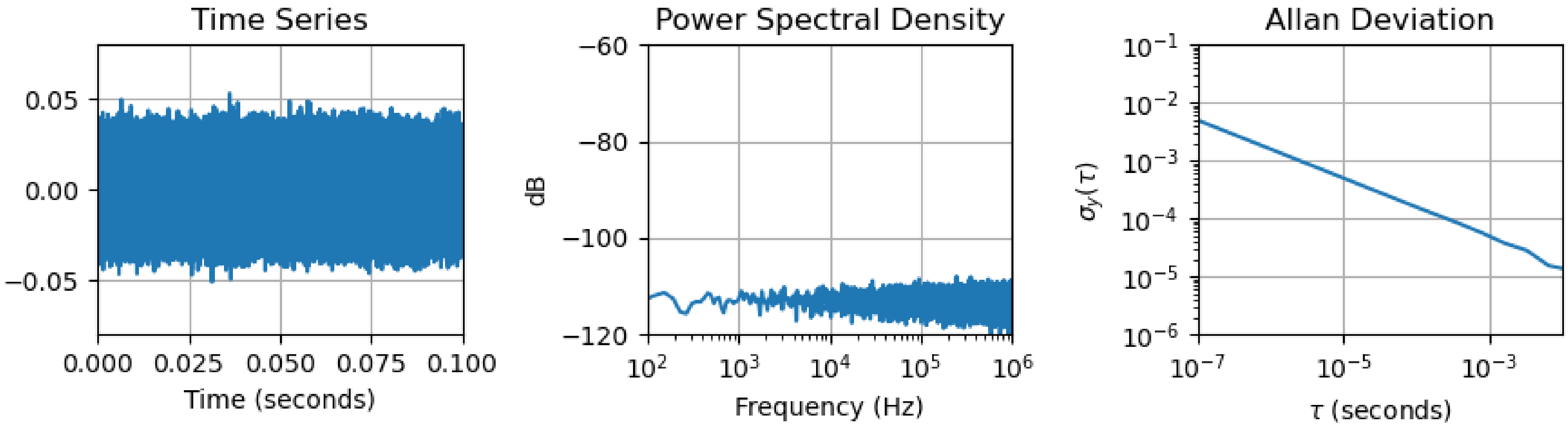
Subtract averages that are τ apart in time to get a difference error

RMS all the differences

Normalize answer by dividing by $\sqrt{2}$

ADEV – White Noise

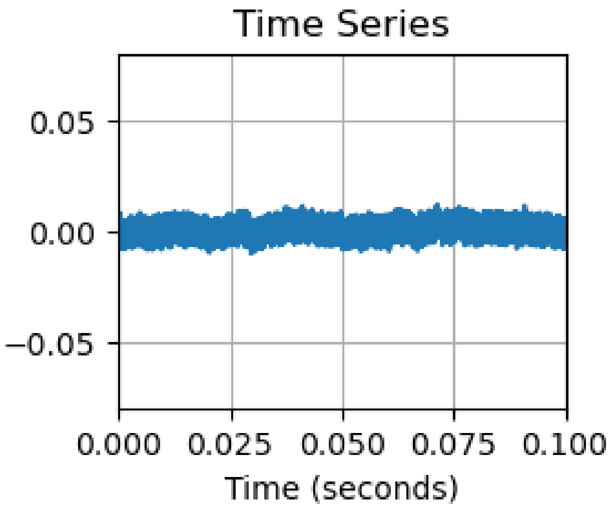
constant $\mathbf{1/\sqrt{\tau}}$

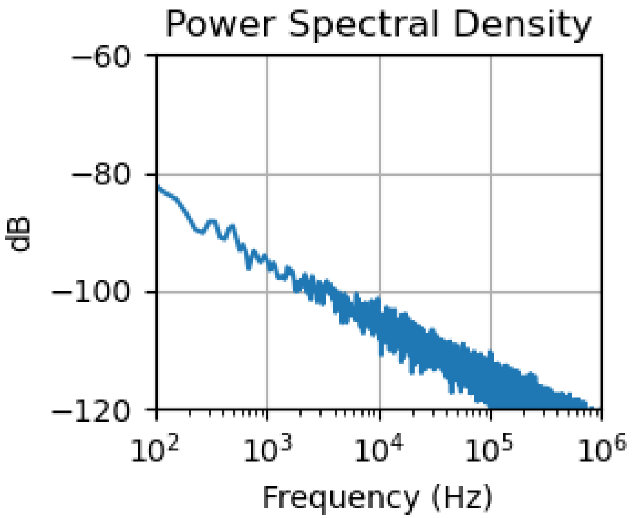


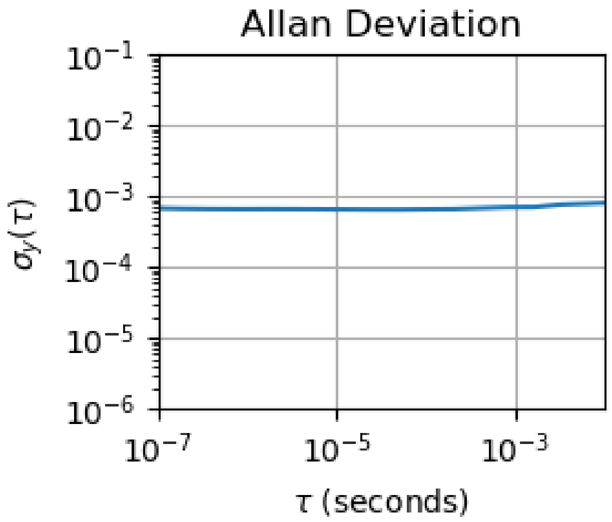
ADEV – Pink Noise

$1/f$

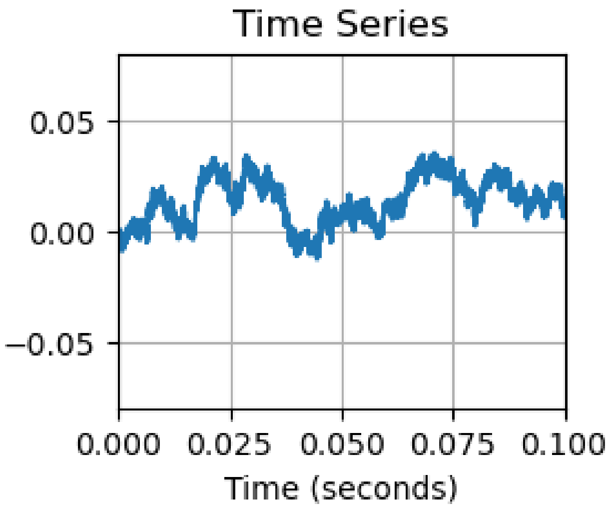
constant



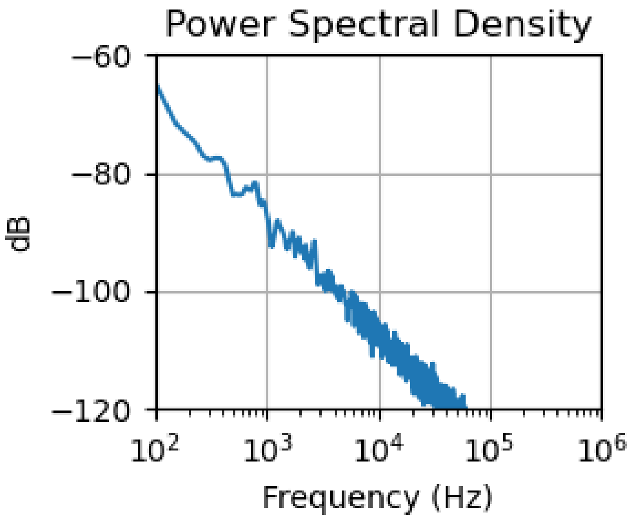




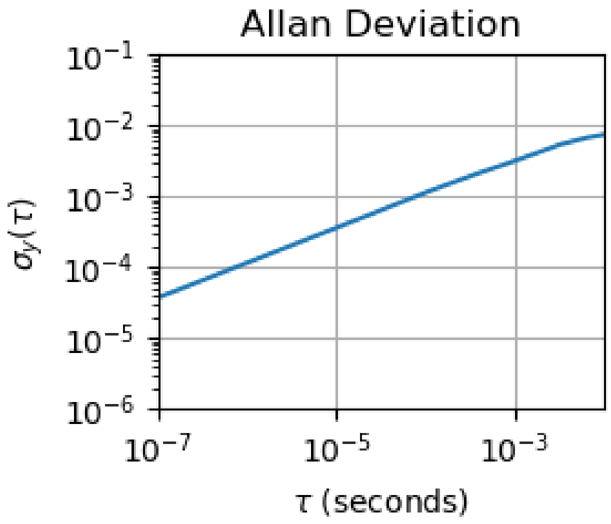
ADEV – Red (Brownian) Noise



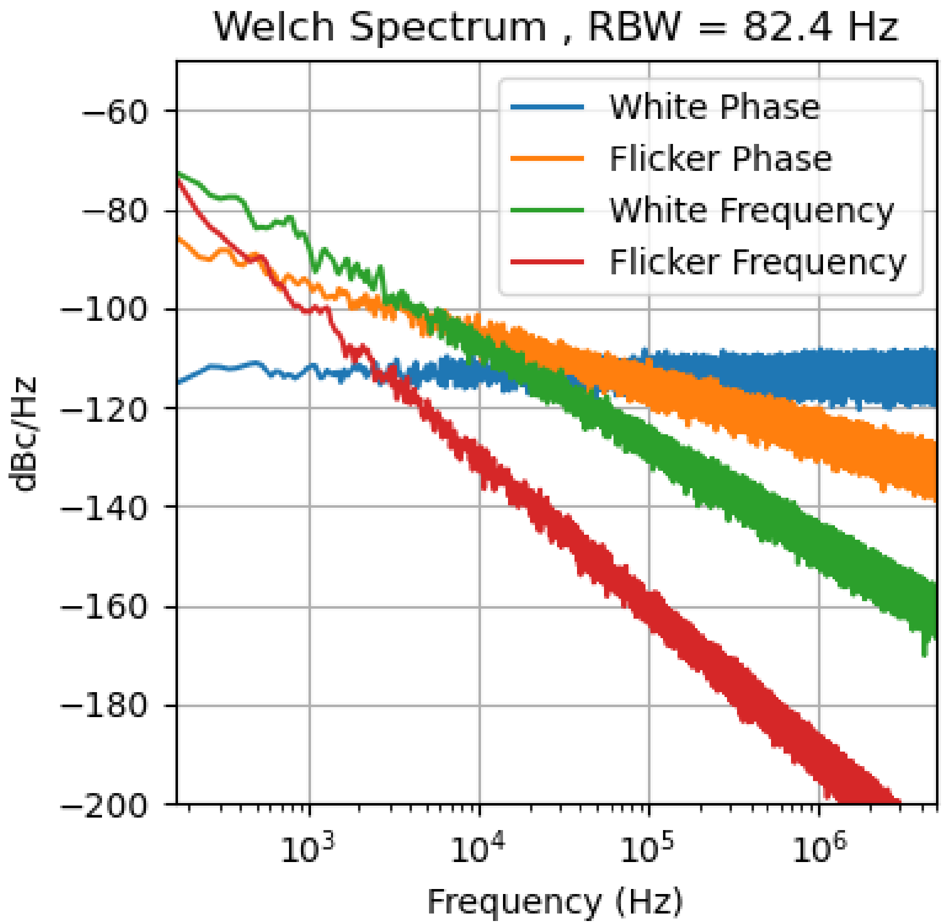
$\mathbf{1/f^2}$

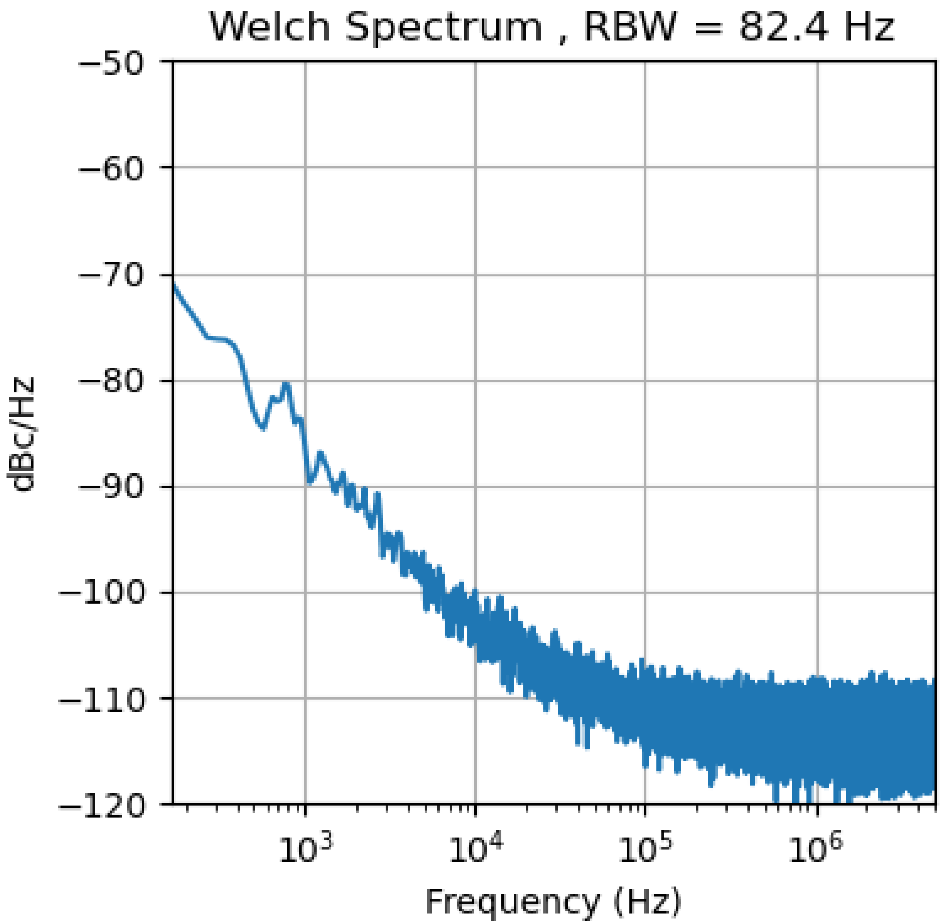


$\sqrt{\tau}$



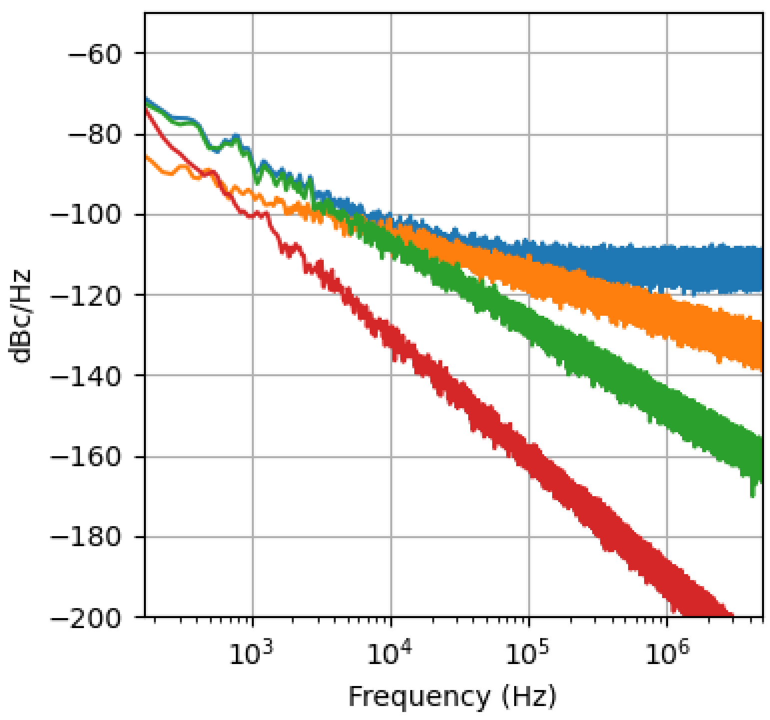
Phase Power Spectral Density - Components

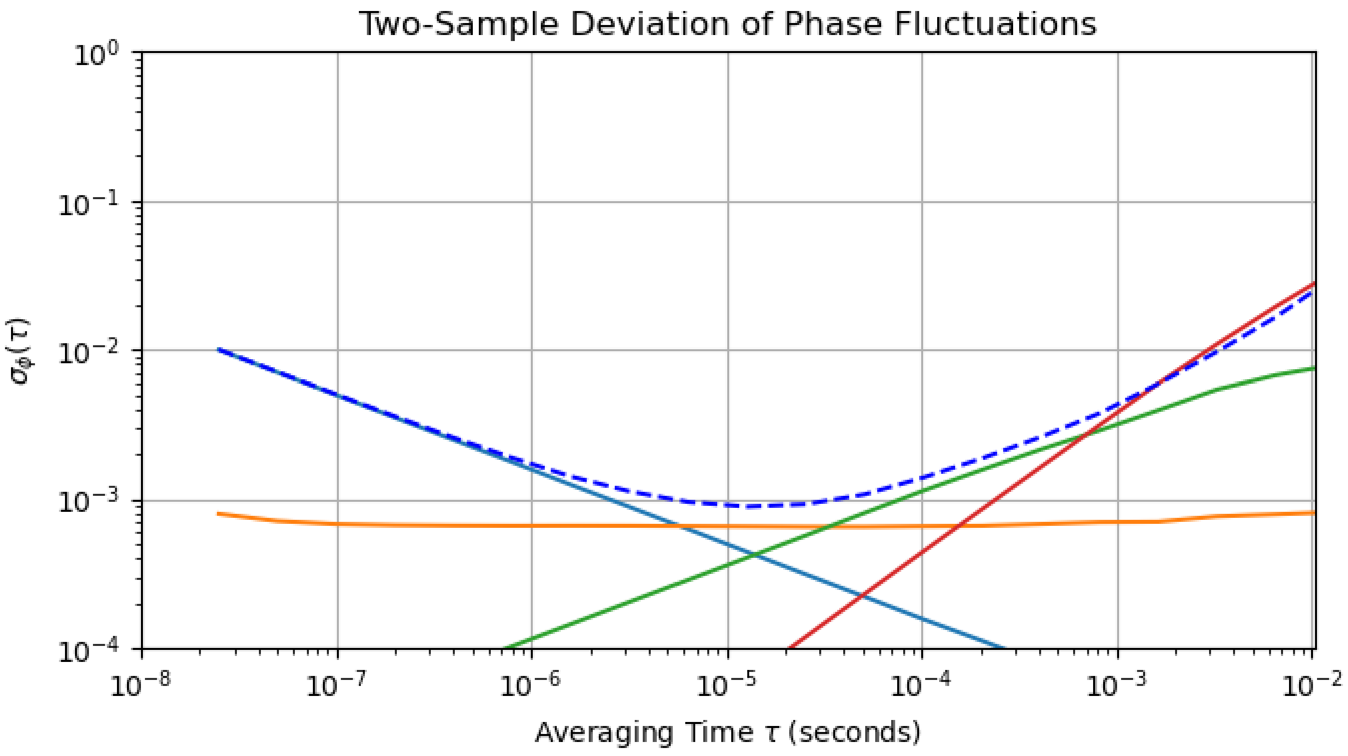




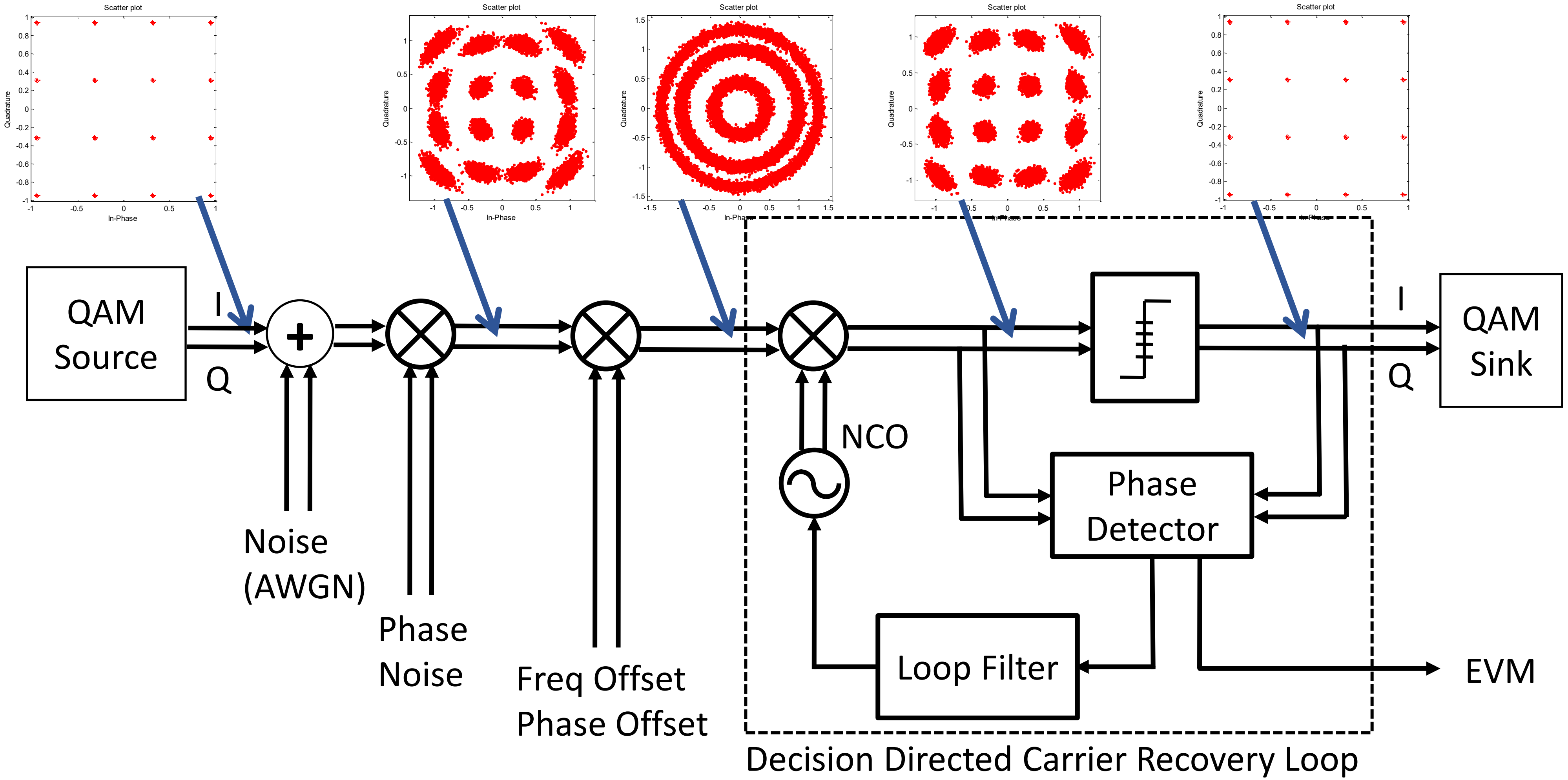
"ADEV" for Phase Fluctuations

$S_\phi(f)$ Two-Sided Power Spectral Density due to Phase Fluctuations



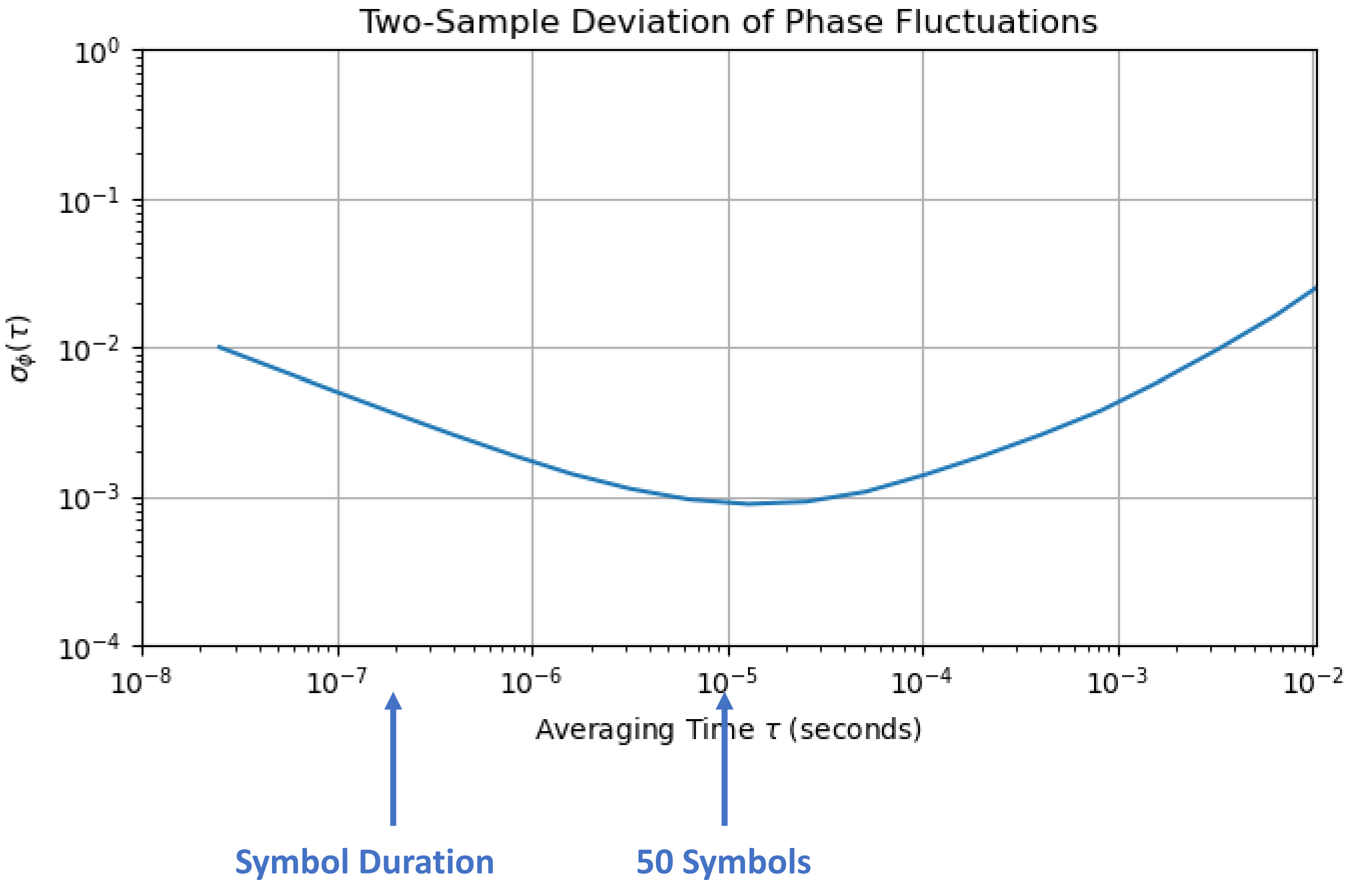


QAM Carrier Recovery and Phase Noise

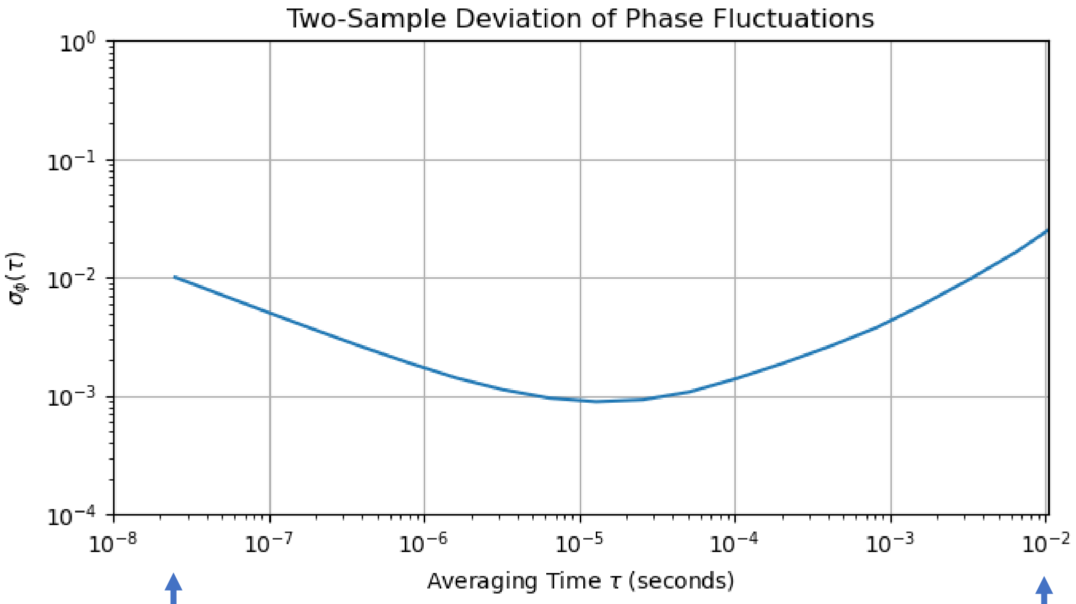


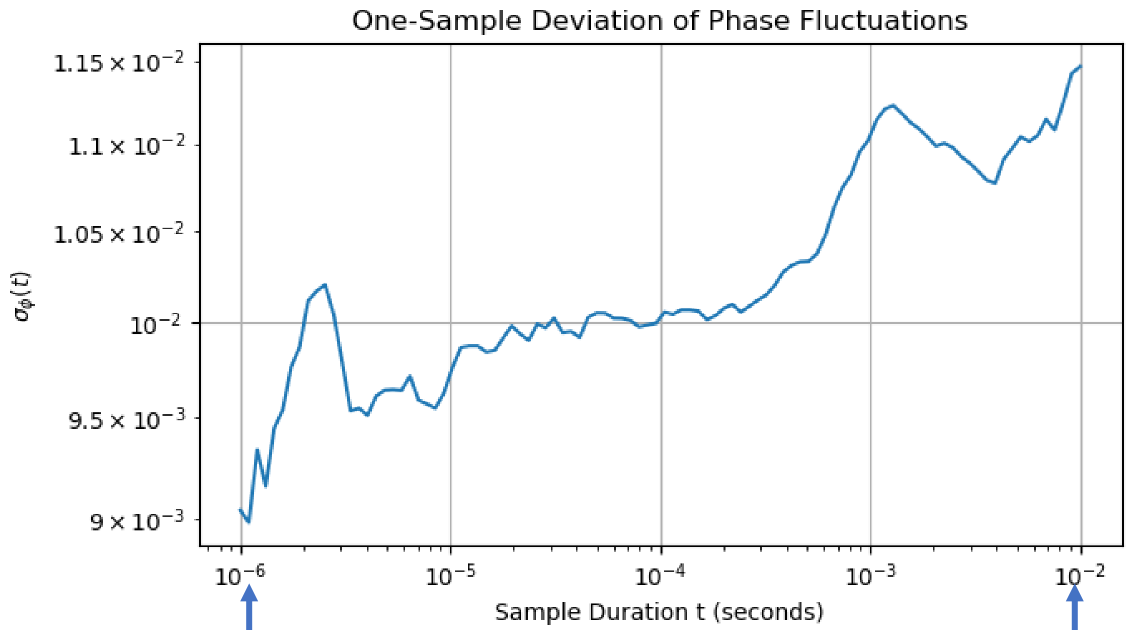
QAM Example Showing ADEV Utility

4096 QAM
Symbol Rate 5 MHz (duration 200 ns)
Sample Rate 40 MHz
RC alpha = 0.25



"Two-Sample" vs "One-Sample" Deviation



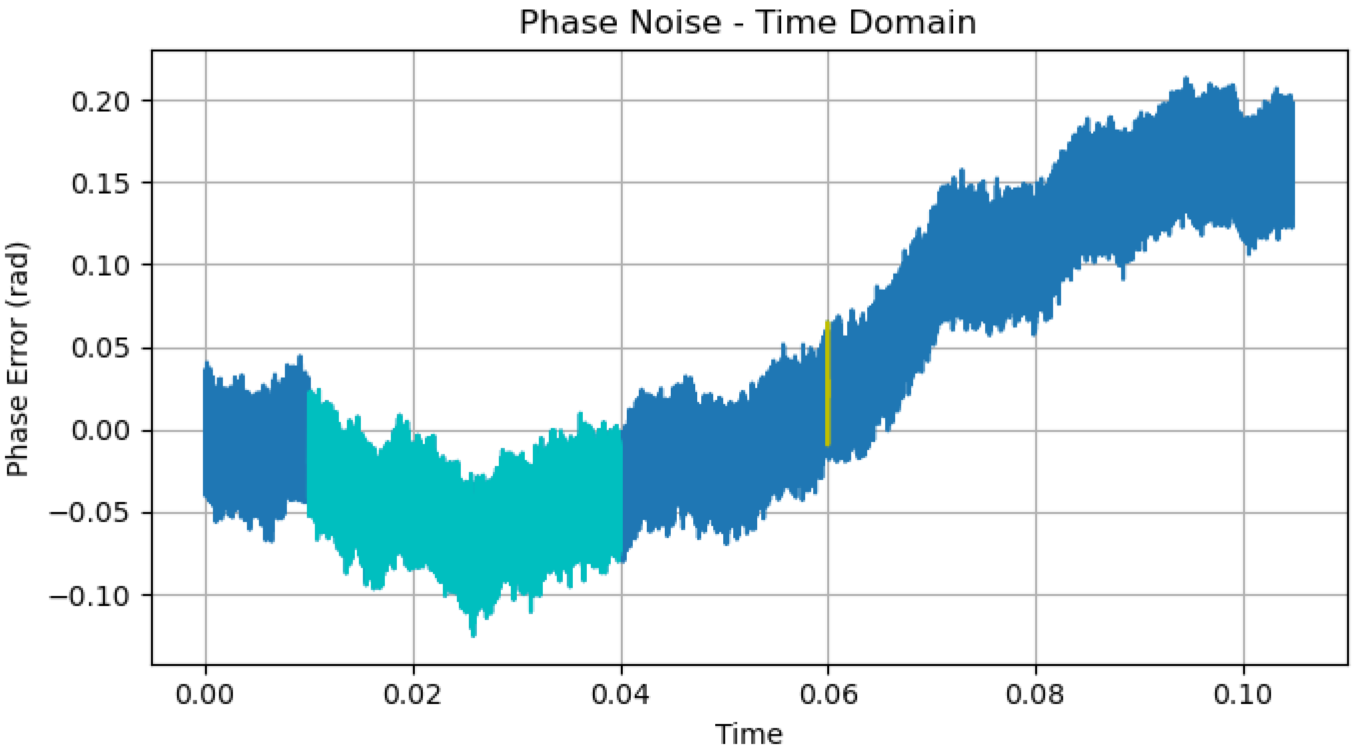


Only 1 sample **400,000 samples** **Only 40 samples** **400,000 samples**

1) Avg each block
2) std of the differences between block avgs

1) std of each block
2) avg the stds of each block

4096 QAM EVM Results



30ms, 150,000 symbols

-37.2 dB

9μs, 360 symbols

-49.4 dB

Recap For Future Reference

FFT Spectrum to measure Tones

Zero pad FFT 4x to 10x to minimize "scalloping loss" (next pow2)

Window with Kaiser Window, Adjust beta to adjust RBW

Scale FFT by sum of the window for accurate tone measurement

```
N = len(x)
win = sig.windows.kaiser(N, beta)
fout = fft.fft(x * win, N * 10) / np.sum(win)
```

Understand **Resolution Bandwidth**

```
rbw = len(win) * np.sum(win**2) / np.sum(win)**2     # rbw in bins
```

Welch Spectrum to measure Spectral Density

Primary knobs

Block size (nperseg): trades resolution BW and resolution

Window (win): eliminates spectral leakage

Block overlap (noverlap): minimize double counting: 50% for Hann, 75% for Kaiser

Additional details

One-sided or two-sided (return_onesided)

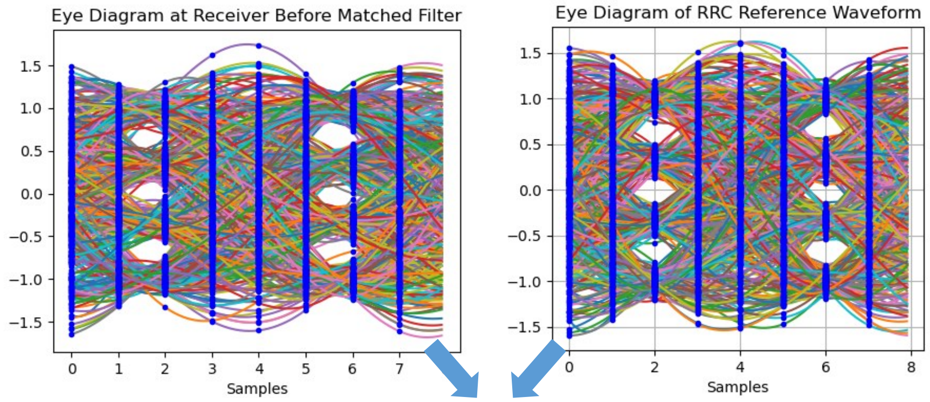
Average with 'mean' vs 'median'

Detrend (I prefer False)

Returns a **Power Quantity** (use 10log10() to convert to dB)

```
fout, psd = sig.welch(x, nperseg=512, win=(sig.windows.kaiser(512,12)),
                             return_onsided=False, noverlap=0.75, fs=fs, detrend=False)
```

EVM, Correlation and SNR



x: waveform
r: reference
(both mean removed and normalized by their std)

Using **EVM**:

$$EVM = \sqrt{\frac{\sum_N |x[n] - r[n]|^2}{N}}$$

$$SNR_{dB} = -20\log_{10}(EVM)$$

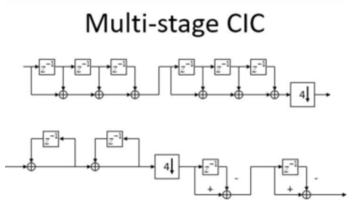
Using **Correlation Coefficient**:

$$\rho = \frac{\sum_N x[n]r^*[n]}{N}$$

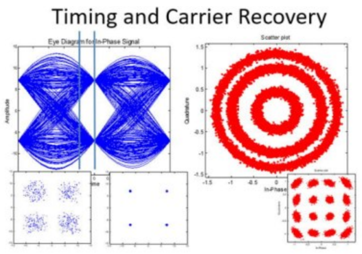
$$SNR_{dB} = 10\log_{10}\frac{|\rho|^2}{1-|\rho|^2}$$

Want More Signal Processing??

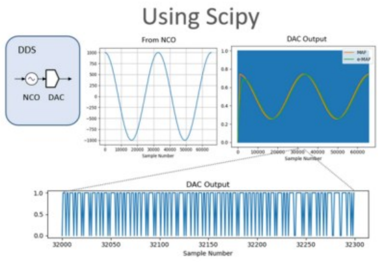
https://dsp-coach.com



Signal Processing for Wireless Communications

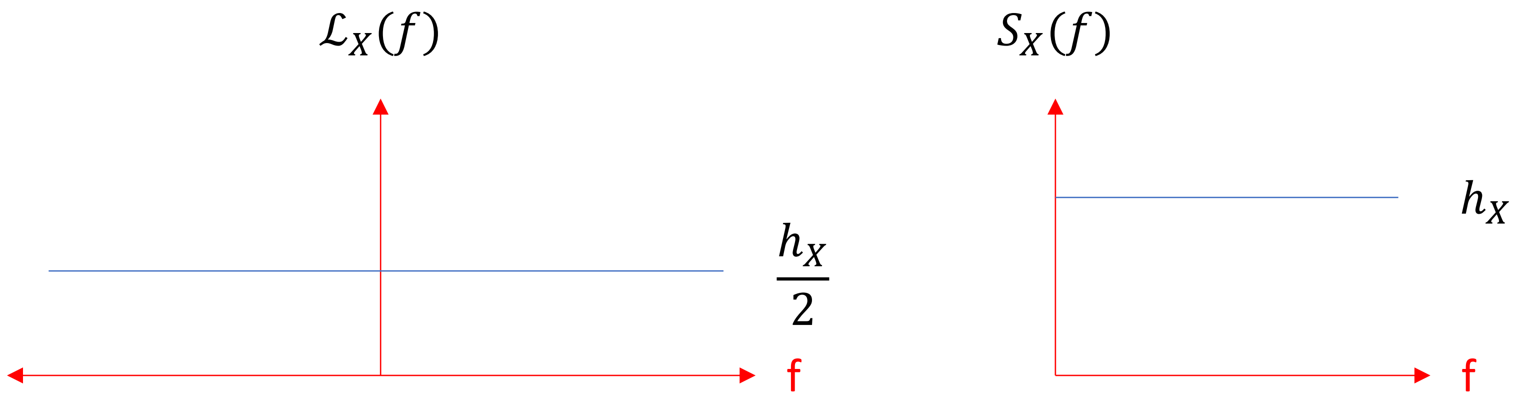


Signal Processing for Software Radio



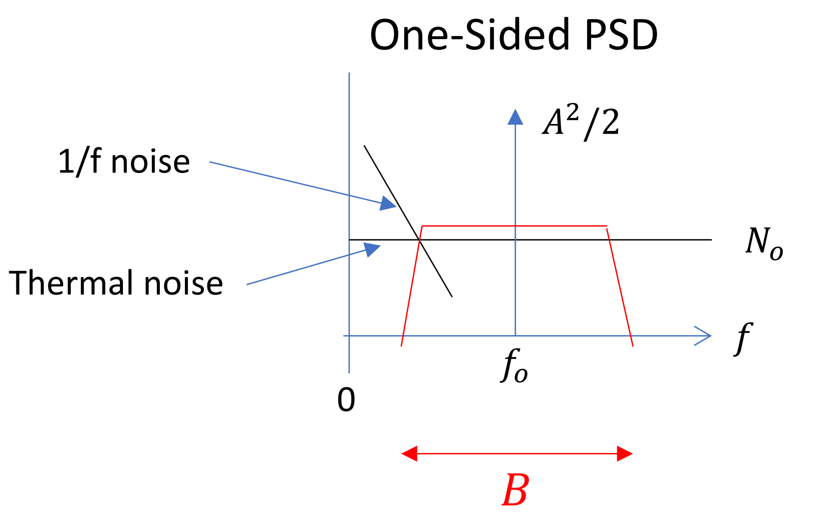
Python Applications for Digital Design and Signal Processing

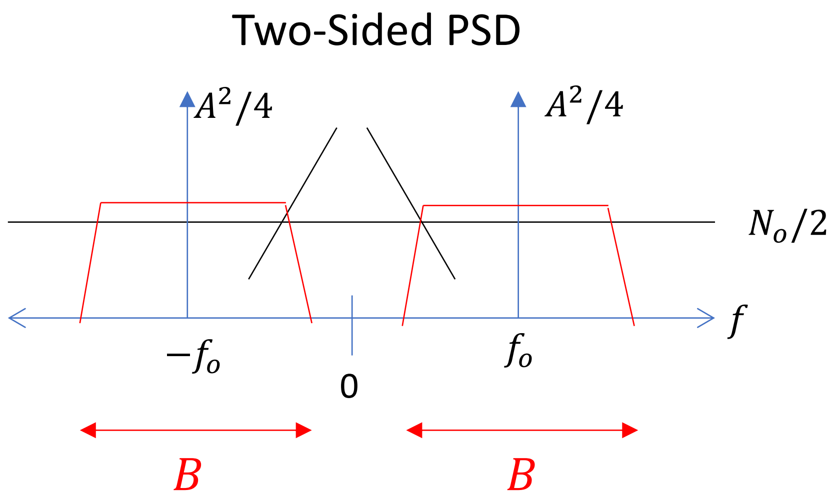
Backup Slides



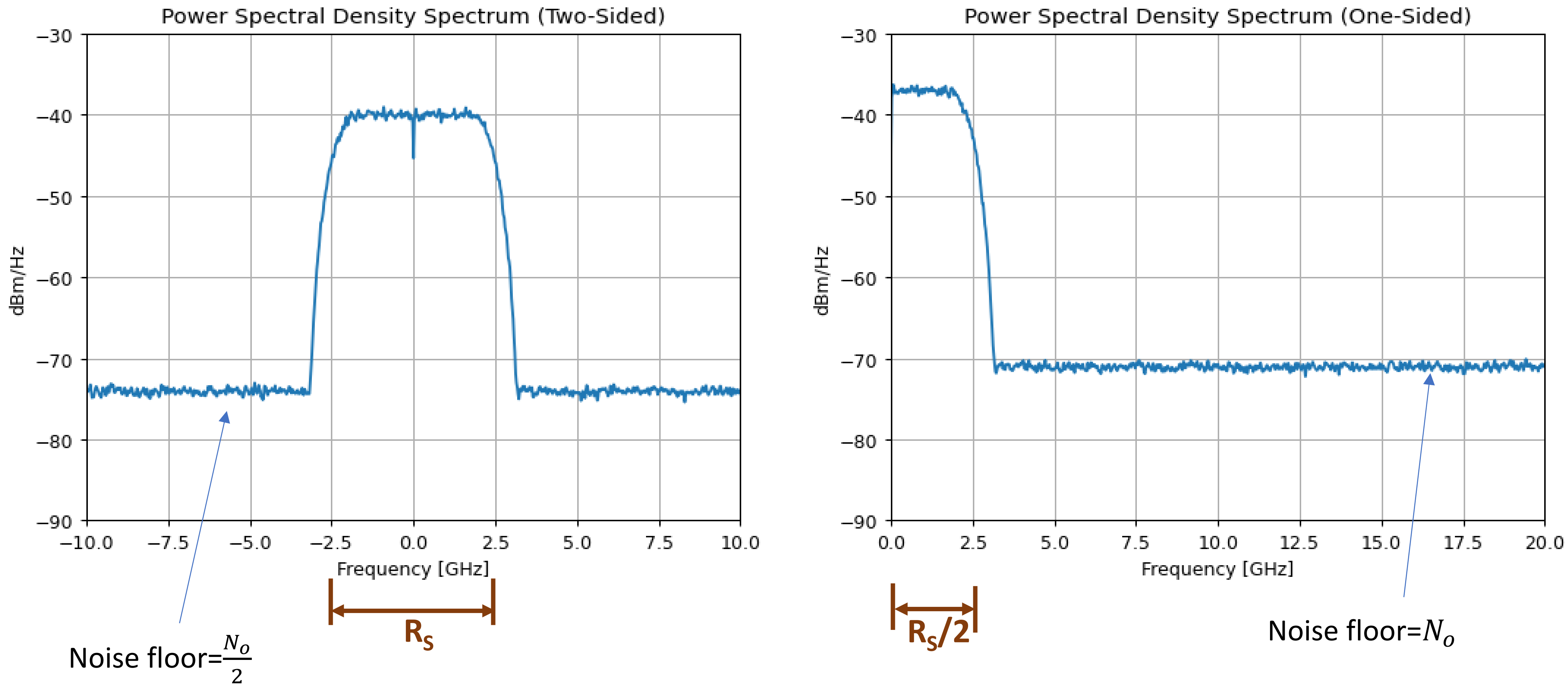
Two-Sided Spectrum

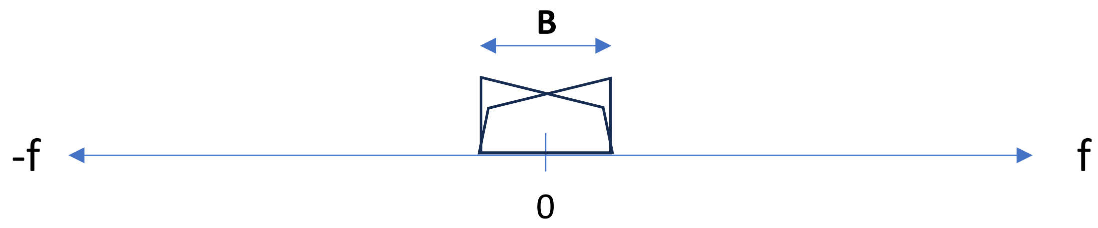
One-Sided Spectrum


One-Sided PSD
1/f noise
Thermal noise
$A^2/2$
N_o
f
f_o
0
B

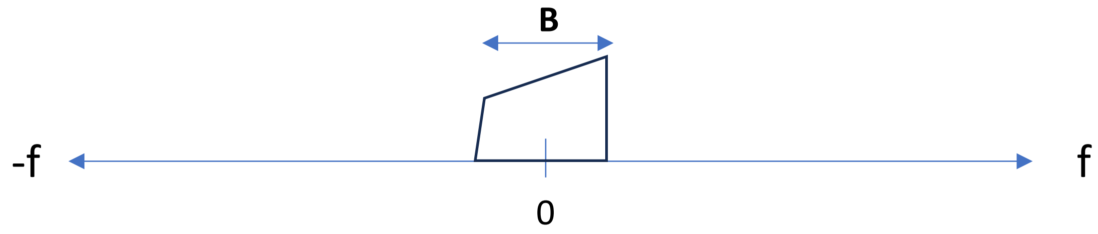

Two-Sided PSD
$A^2/4$
$A^2/4$
$N_o/2$
f
$-f_o$
f_o
0
B
B

QPSK symbol rate: R_s=5 GSymbols/sec (Raised Cosine $\alpha = 0.3$)

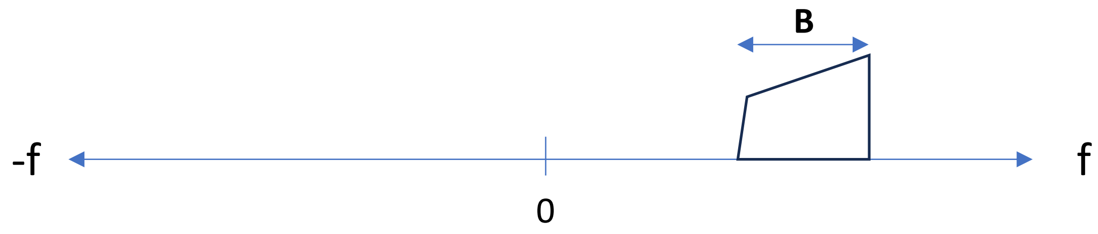




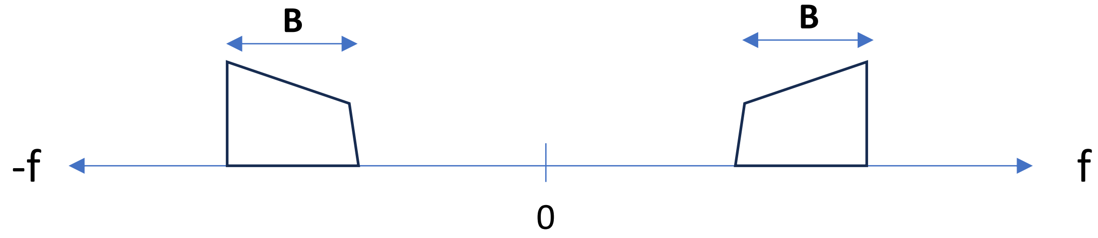
Real Baseband Spectrum



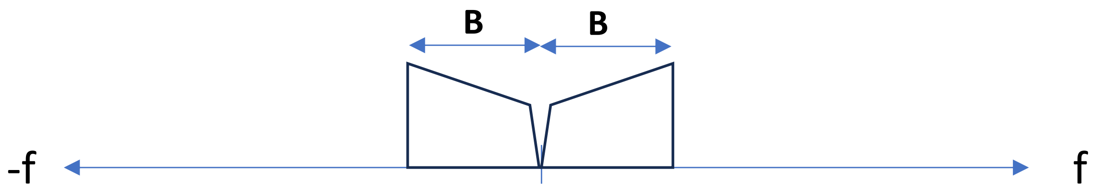
Complex Baseband Spectrum



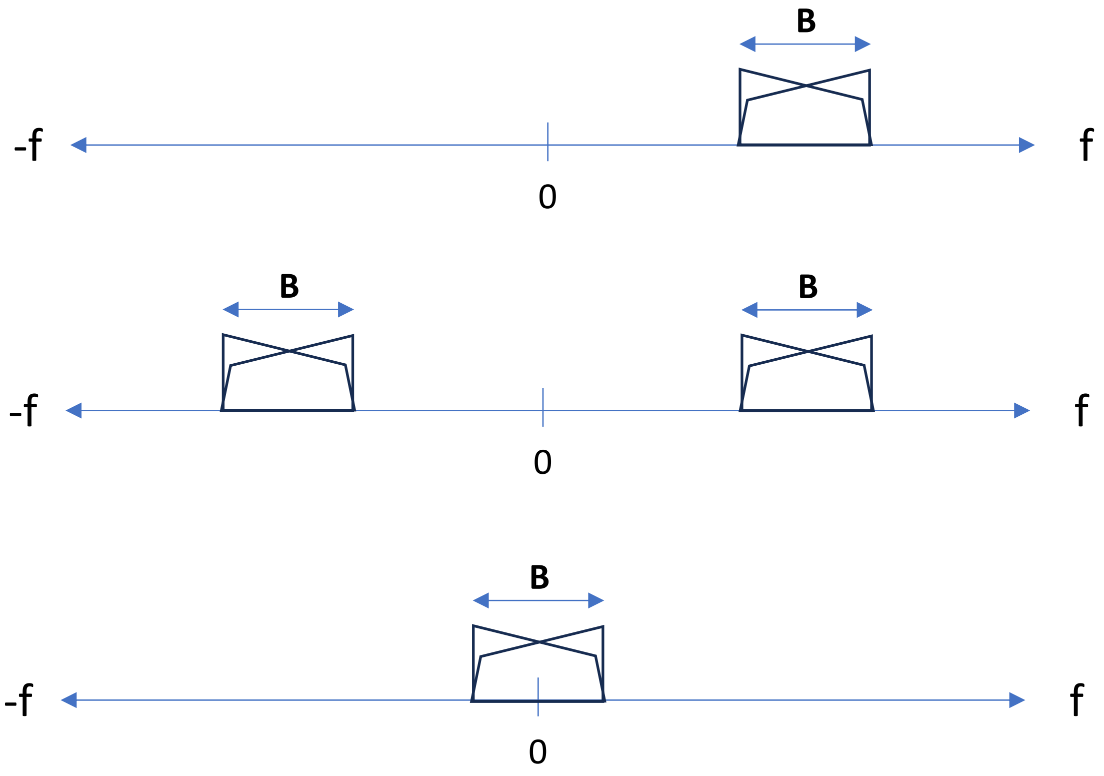
Complex Passband Spectrum



Real Passband Spectrum



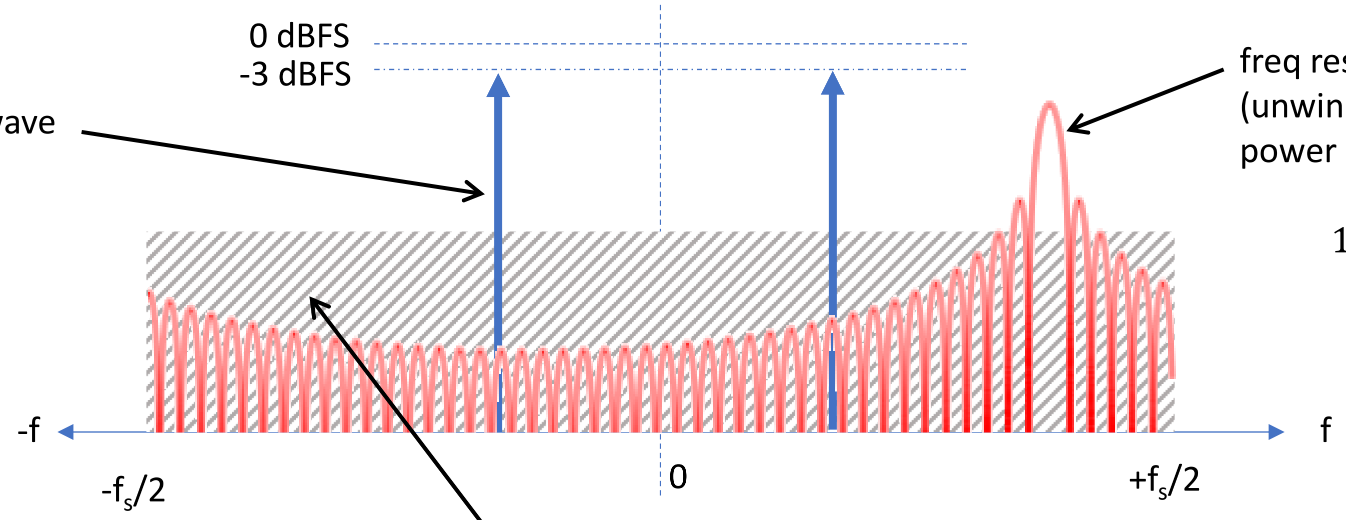
Real Passband Spectrum



Complex Passband Spectrum

Real Passband Spectrum

Real Baseband Spectrum

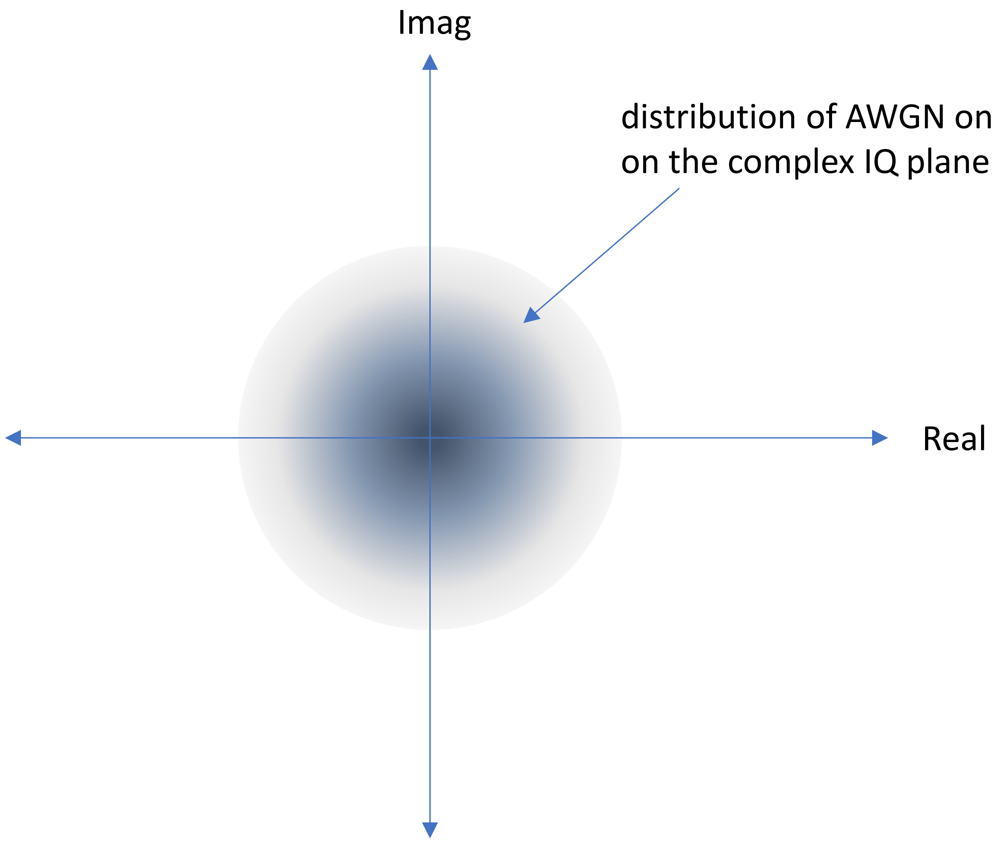


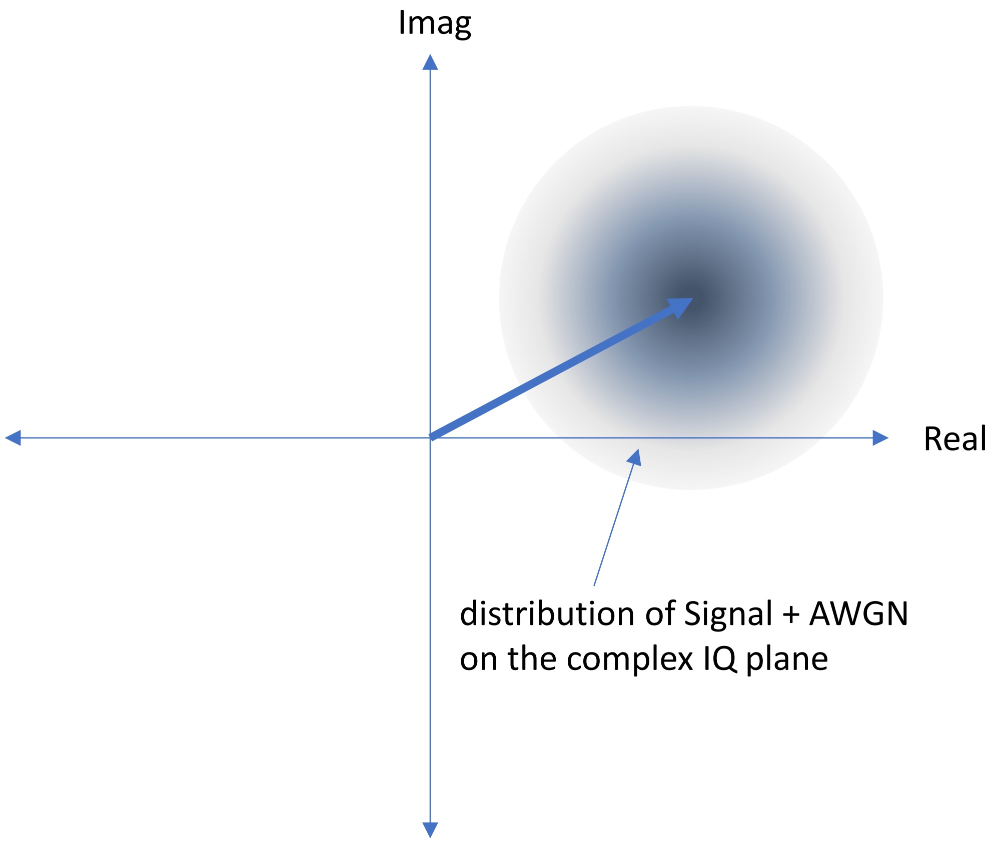
full-scale sine wave

freq response of one DFT bin (unwindowed) for M point DFT. Total power in bin = P_Q/M or in dB:

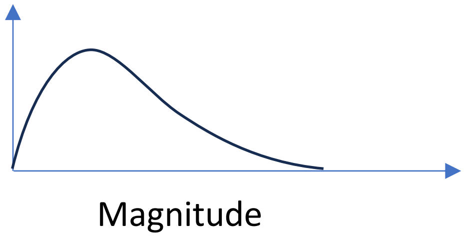
$$10\log_{10} P_Q - 10\log_{10}(M)$$

Total area where hashed is P_Q the total power due to the quantization noise relative to total power of a full scale sine wave, spread evenly over first Nyquist zone +/- $f_s/2$

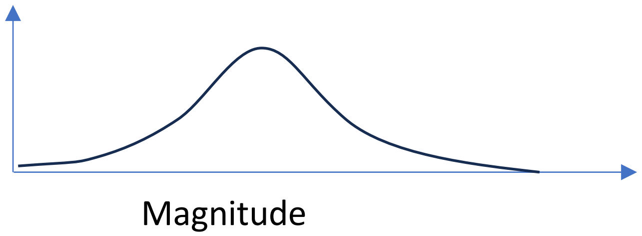


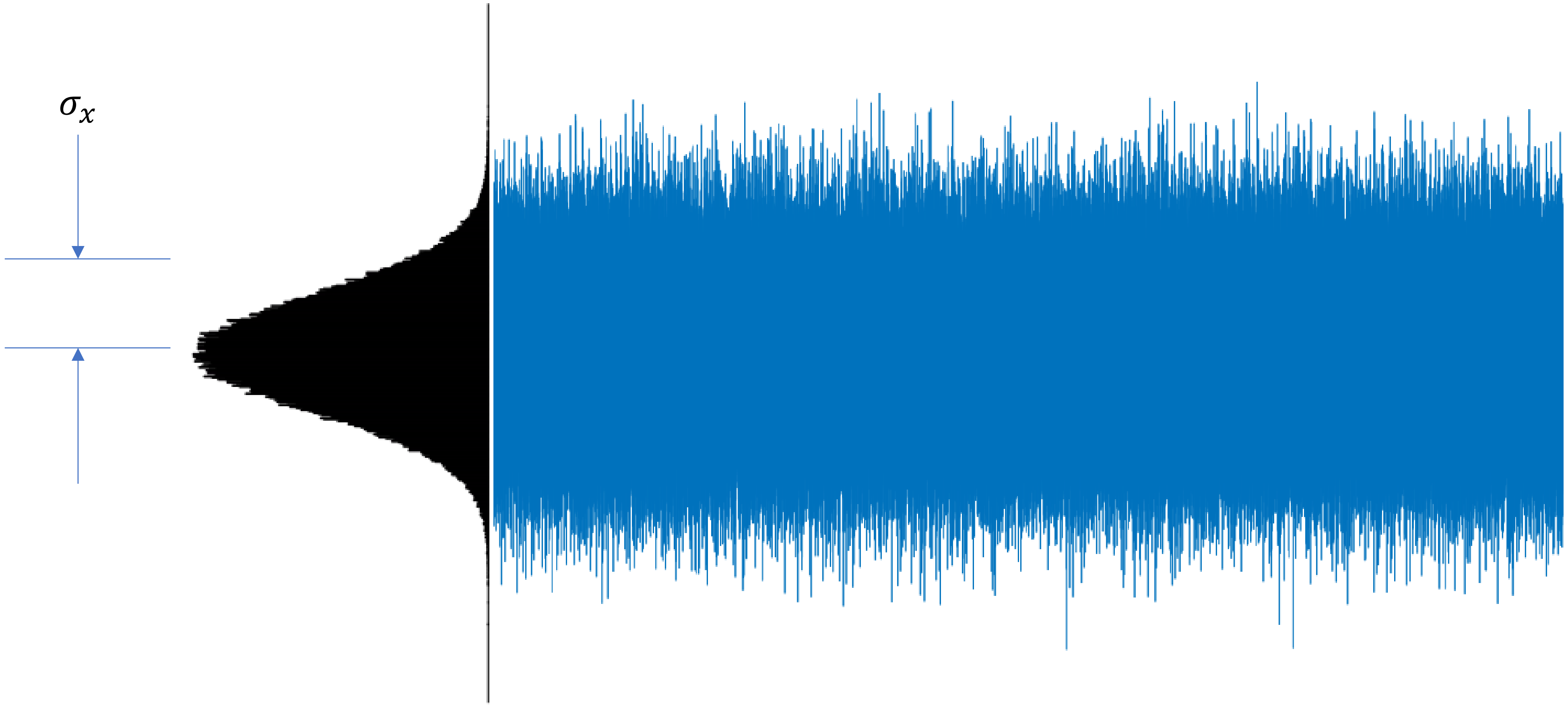


Case 1: Noise Only, The Magnitude is **Rayleigh** Distributed

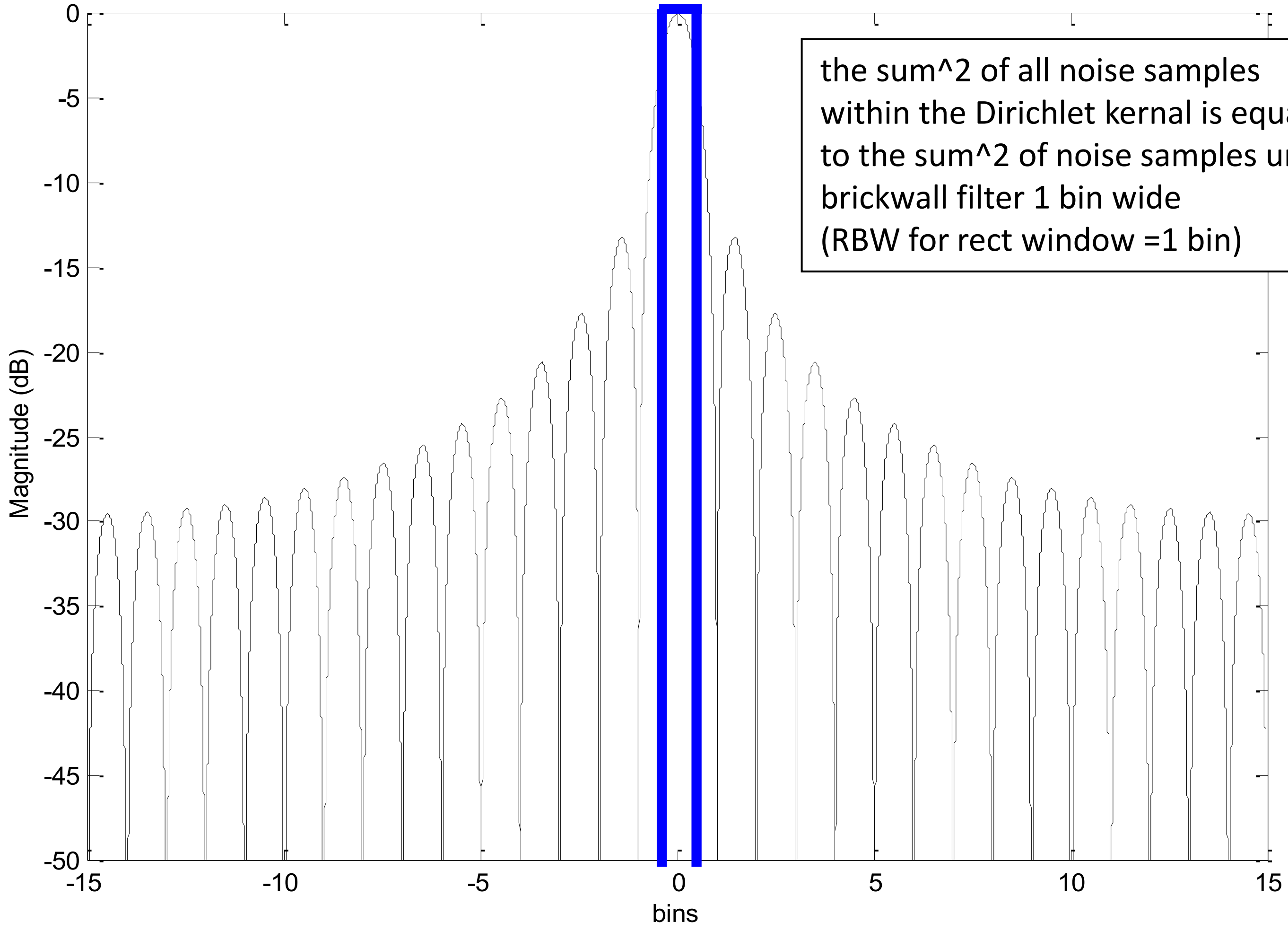


Case 2: Signal + Noise, The Magnitude is **Rician** Distributed



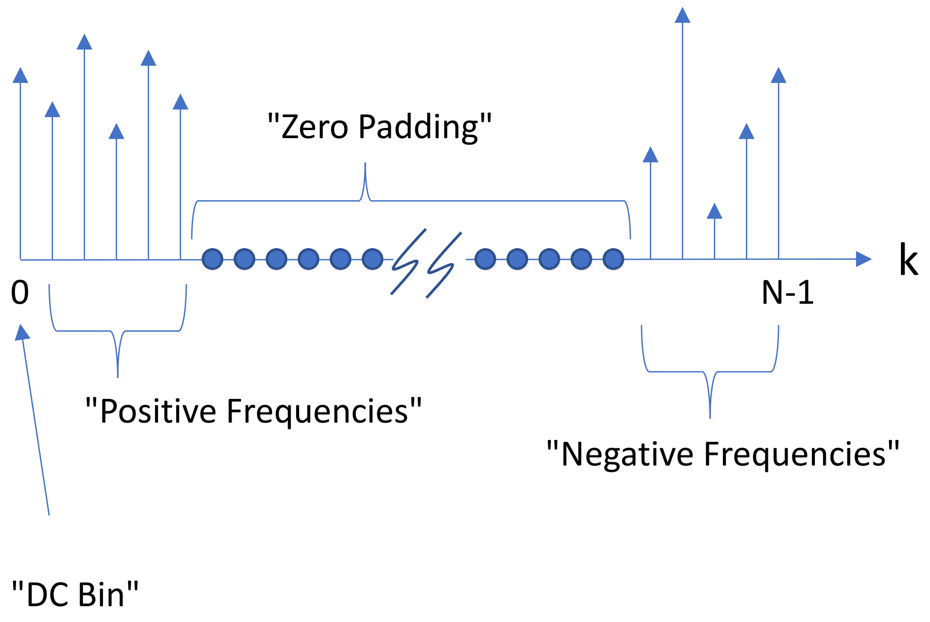

σ_x

Resolution Bandwidth

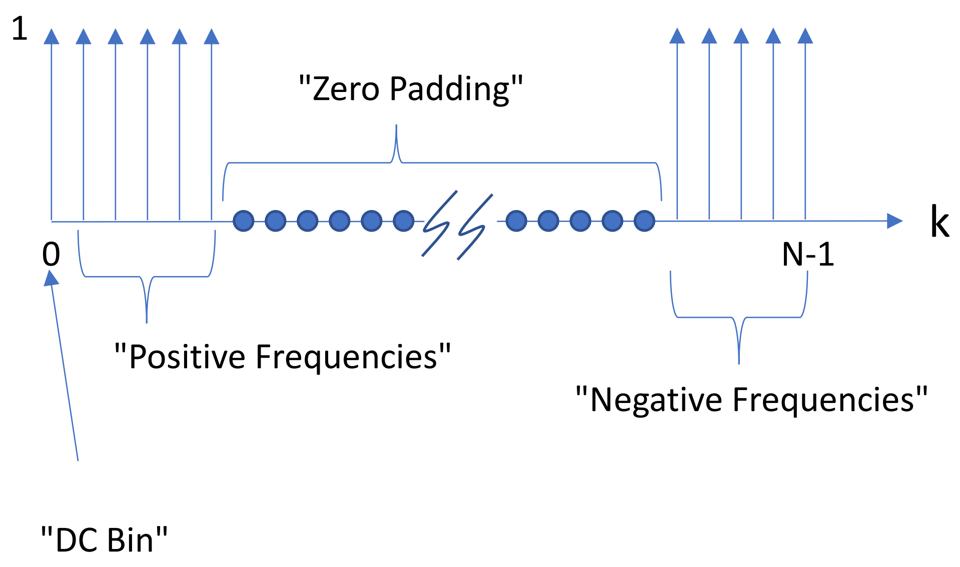


the sum^2 of all noise samples
within the Dirichlet kernal is equal
to the sum^2 of noise samples under a
brickwall filter 1 bin wide
(RBW for rect window =1 bin)

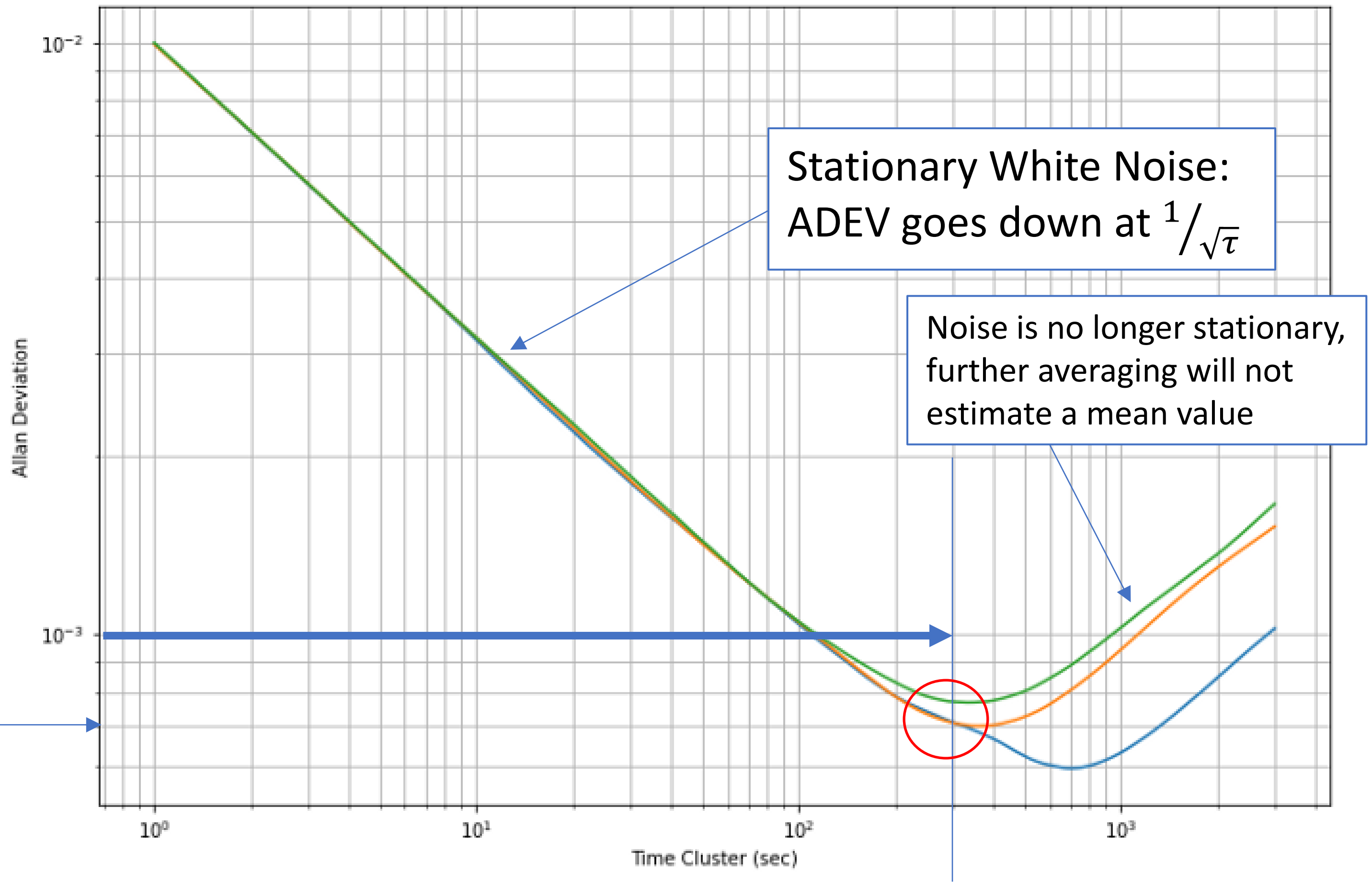
DFT Magnitude



DFT Window

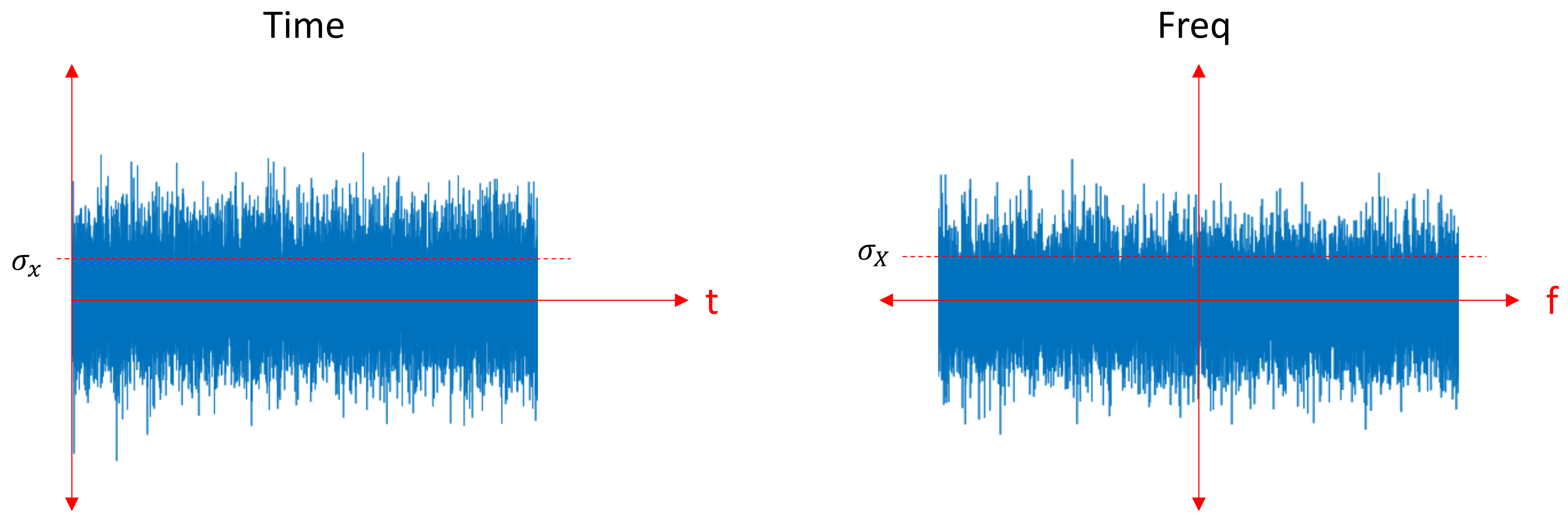


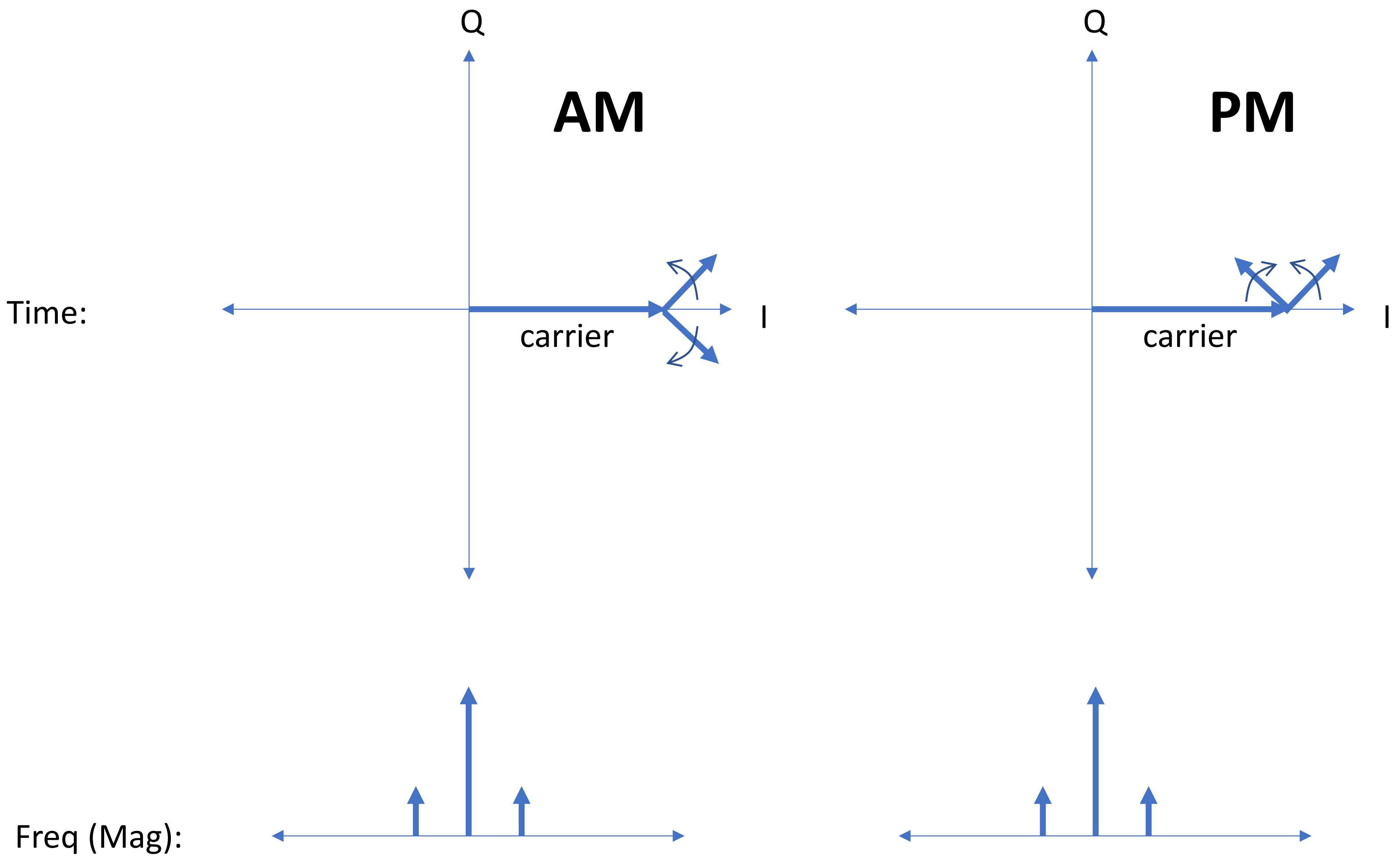
https://towardsdatascience.com/allan-variance-in-python-d1175e5aeee2



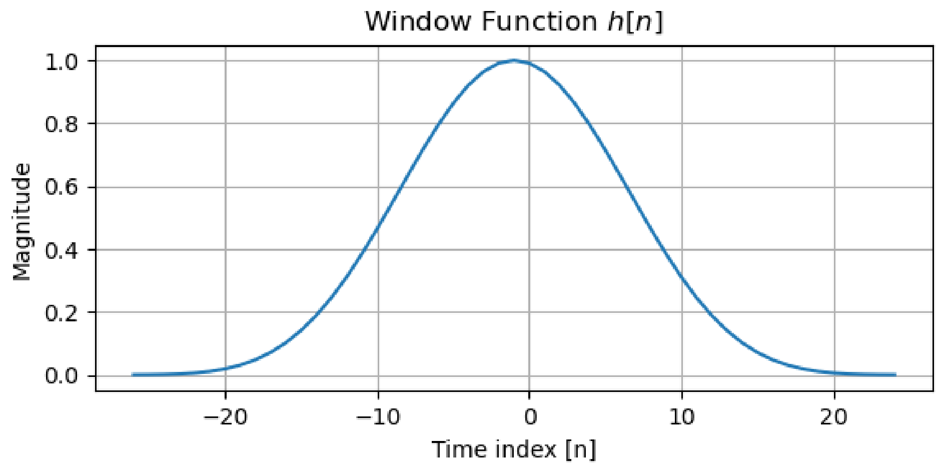
For a "white noise" process this will be the standard deviation of the estimated mean when averaged for τ seconds (in this case of the orange curve, 300 seconds)

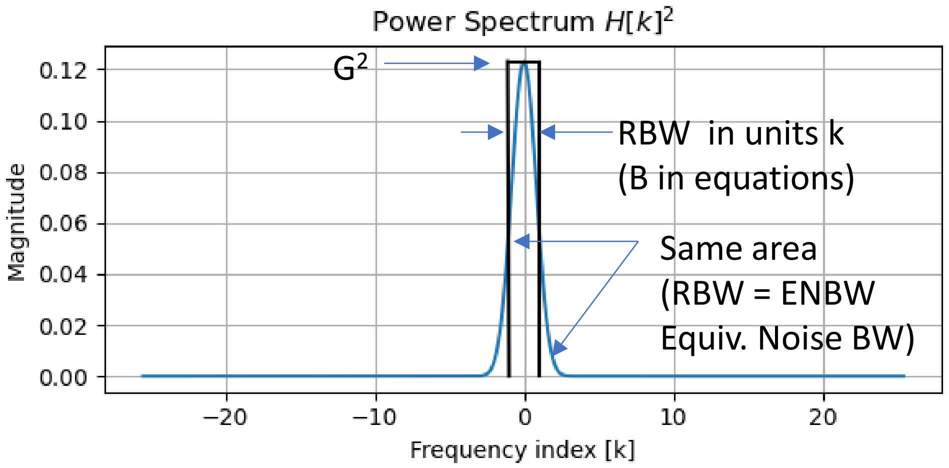
τ: averaging time


Time
σ_x
t
Freq
σ_X
f


AM
PM
Q
Q
I
I
carrier
carrier
Time:
Freq (Mag):

Resolution BW (RBW)





Given scaled DFT as $H[k] = \dfrac{1}{N}\sum_{n=0}^{N-1} h[n]W_N^{nk}$

$$\frac{1}{N}\sum h[n]^2 = \sum |H[k]|^2 \qquad (1)$$

(1) Parseval's Theorem

$$G = \frac{1}{N}\sum h[n] \qquad (2)$$

(2) DC Gain of window (DC bin in DFT)

$$\sum |H[k]|^2 = BG^2 \qquad (3)$$

(3) Area under Power Spectrum of window = Area under equivalent brickwall filter

Combining (1), (2), and (3):

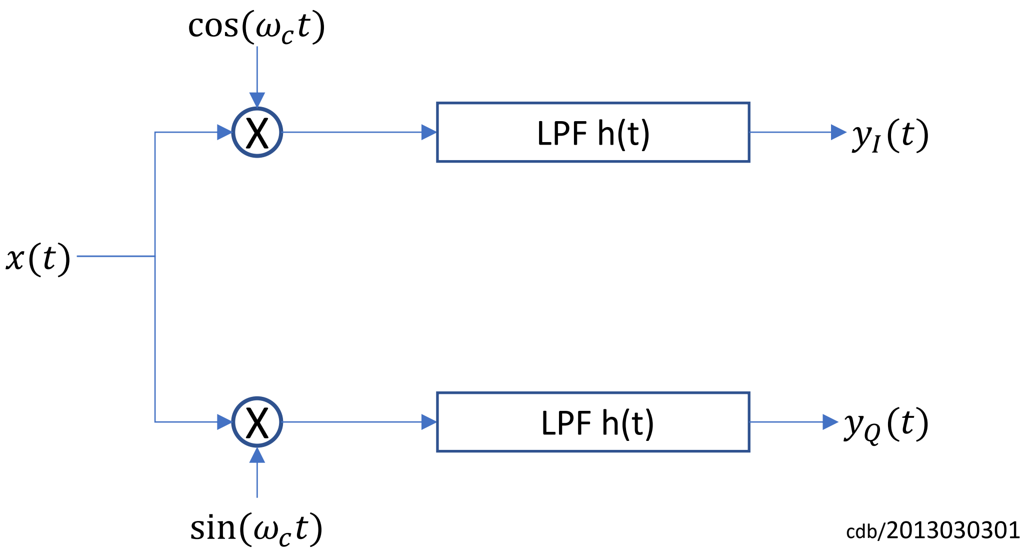
$$B\left(\frac{1}{N}\sum h[n]\right)^2 = \frac{1}{N}\sum h[n]^2$$

Solve for B:

$$\boxed{B = N\frac{\sum h[n]^2}{(\sum h[n])^2}}$$

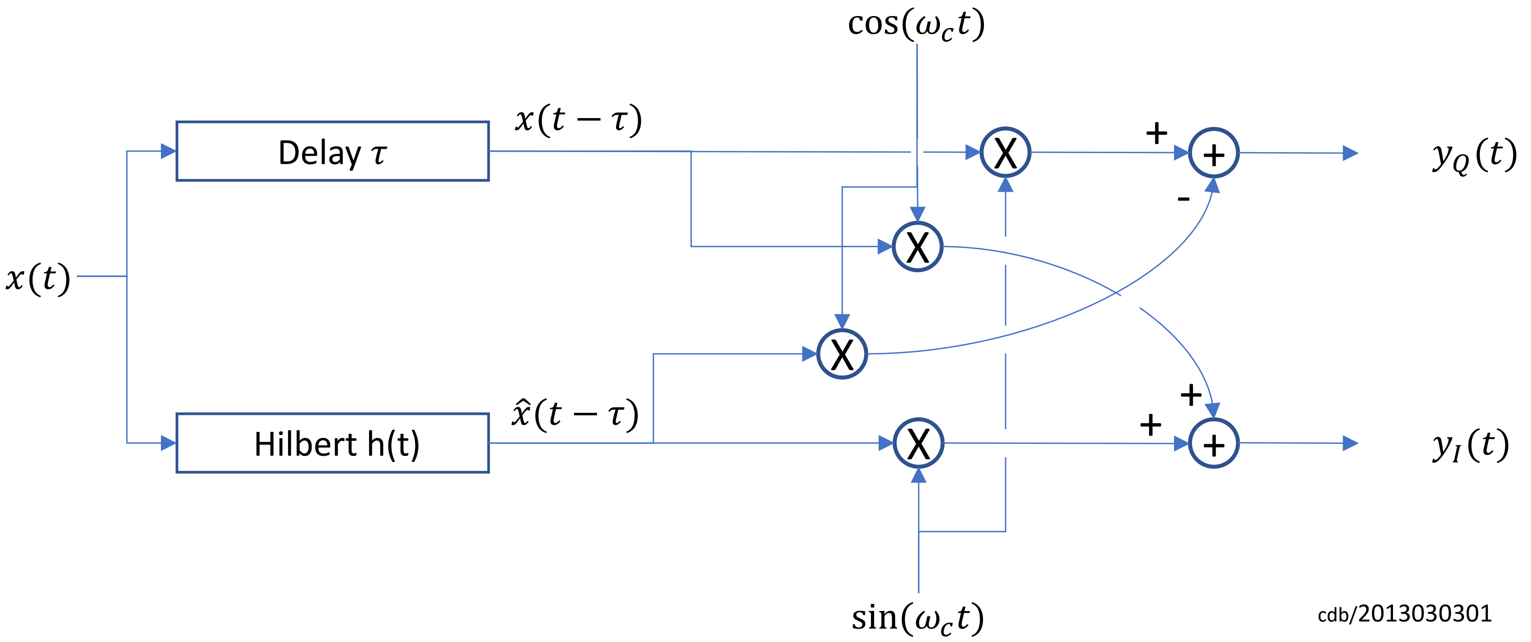
https://dsp.stackexchange.com/questions/86932/converting-real-samples-to-iq/86936#86936

Method 1



Methods for translating Real IF to Complex Baseband I and Q

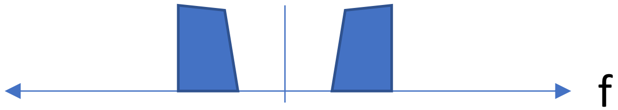
Method 2



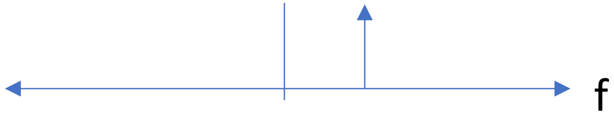
Methods for translating Real IF to Complex Baseband I and Q

Spectrums for "Method 1"

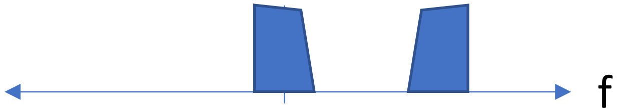
Real Input: $x(t)$



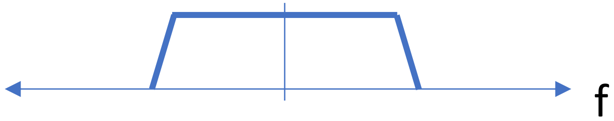
Complex LO: $e^{j\omega_c t}$



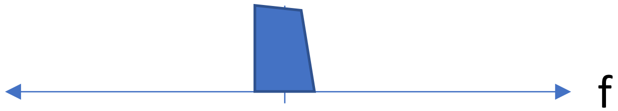
Product (complex) :$x(t)e^{j\omega_c t}$



Real LPF: $h(t)$

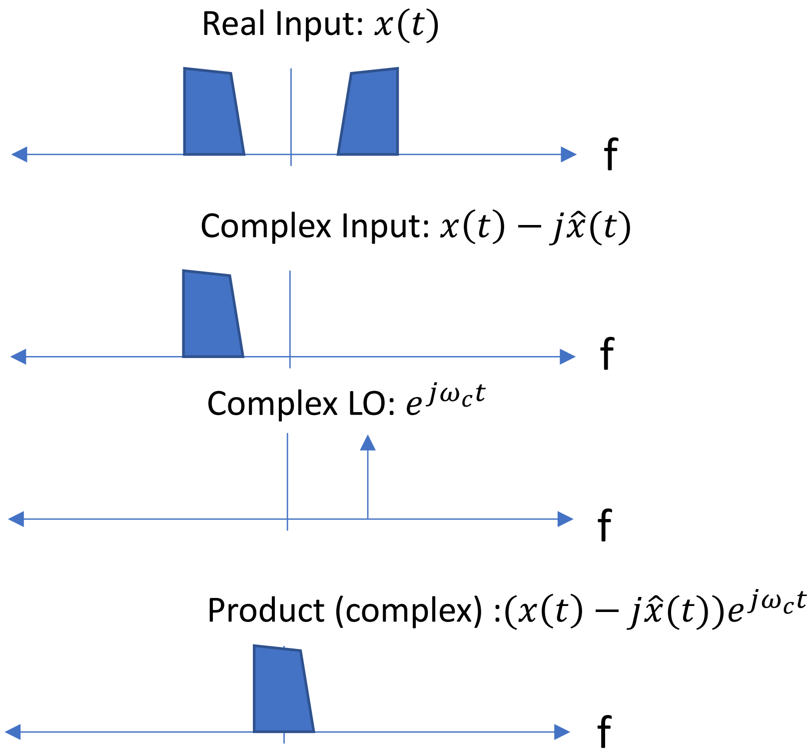


Filtered Product (complex) :$\mathrm{h(t)} * x(t)e^{j\omega_c t}$



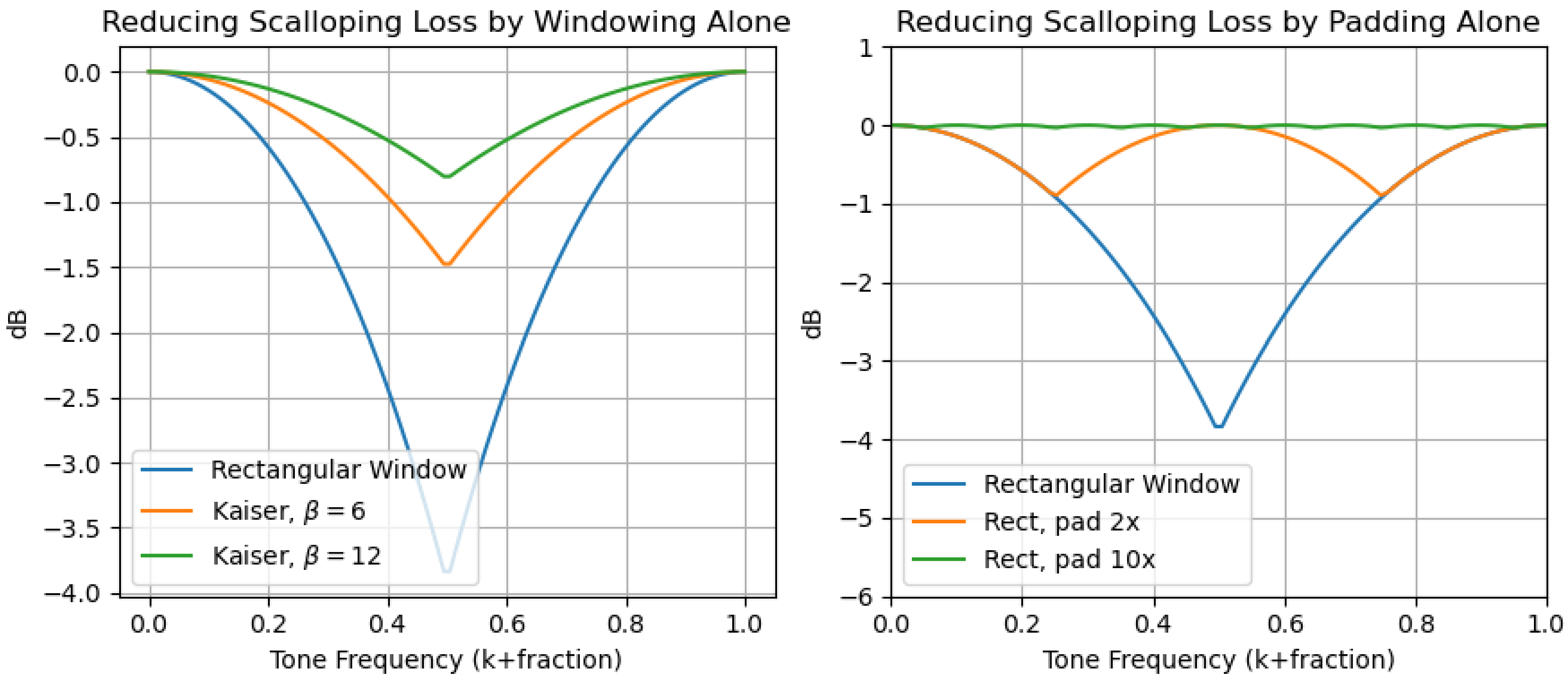
cdb/2013030301

Spectrums for "Method 2"



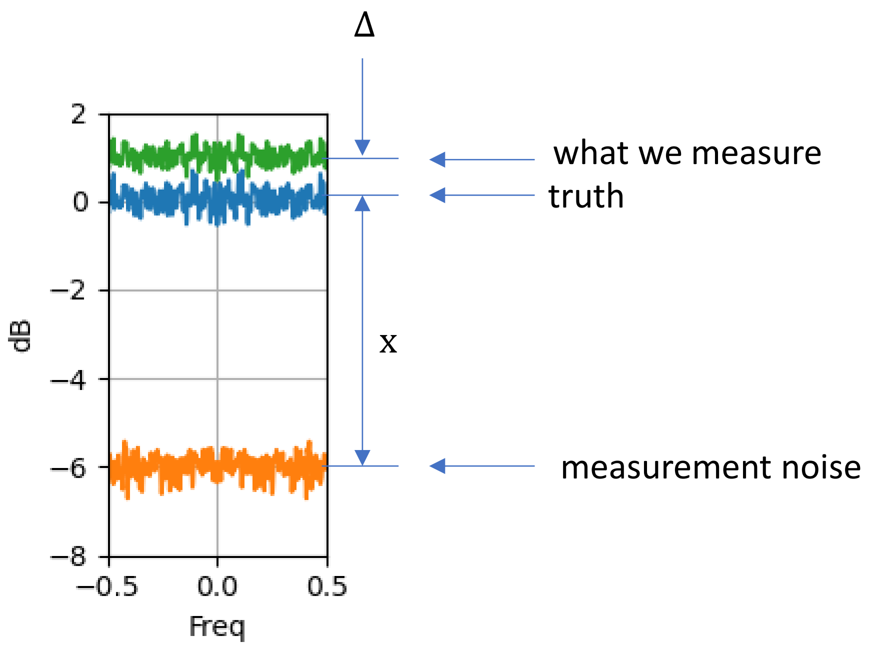
cdb/2013030301

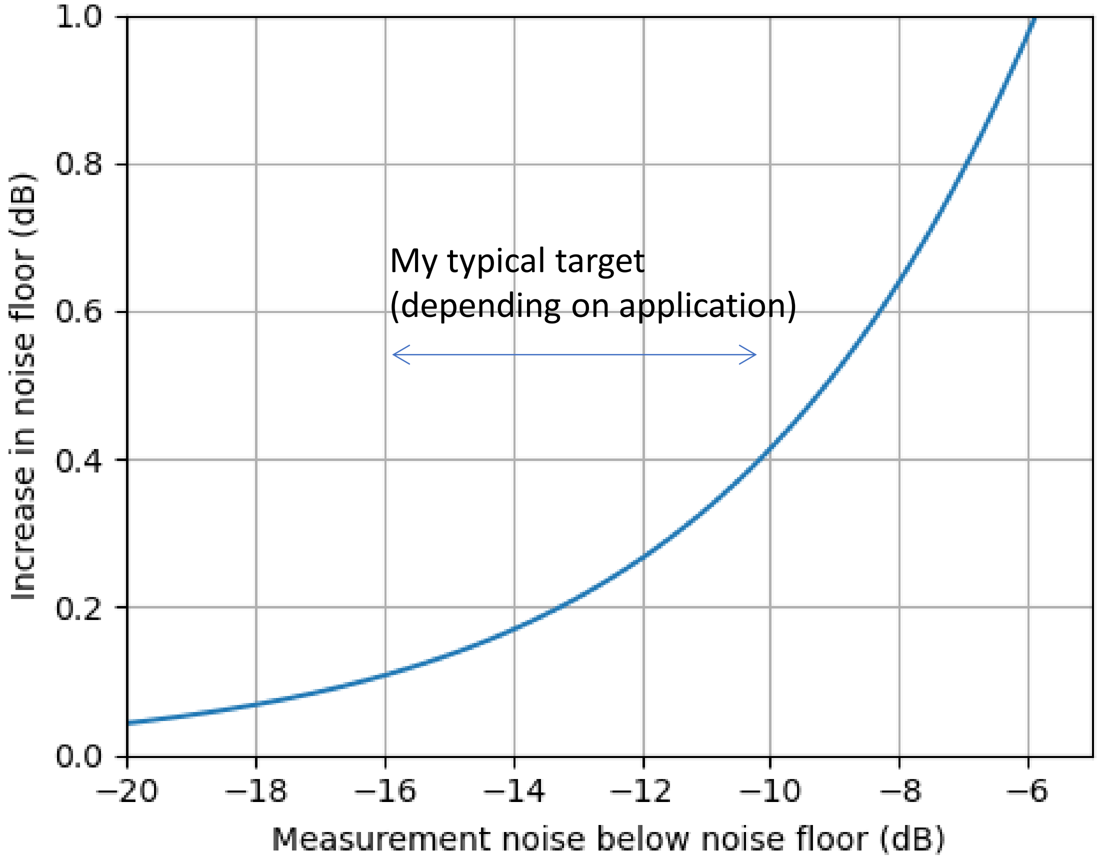
Scalloping Loss



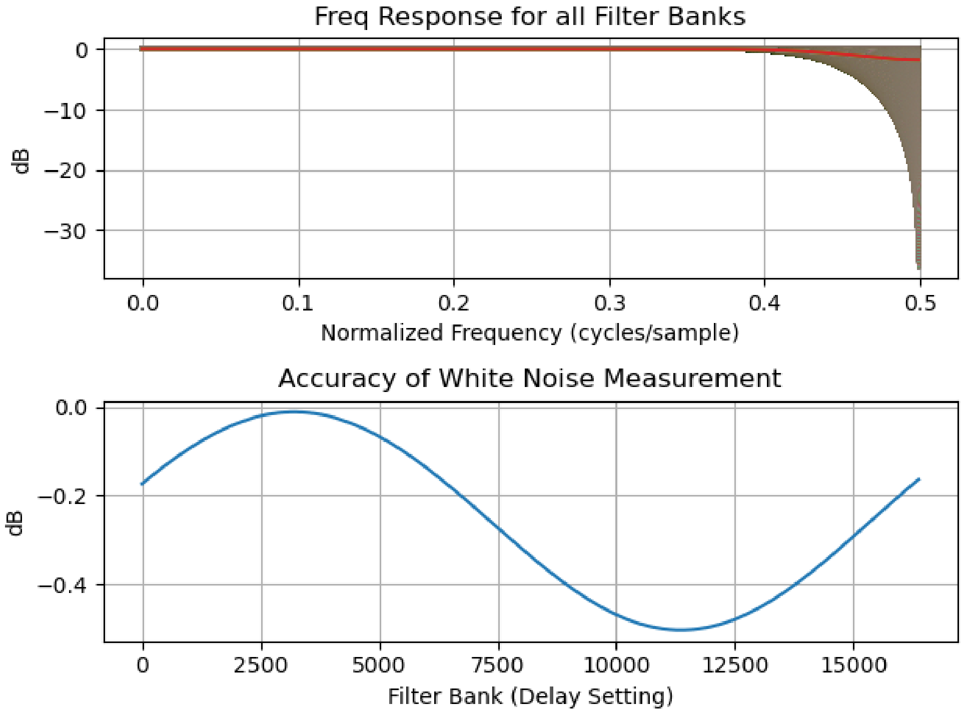
Measurement Noise

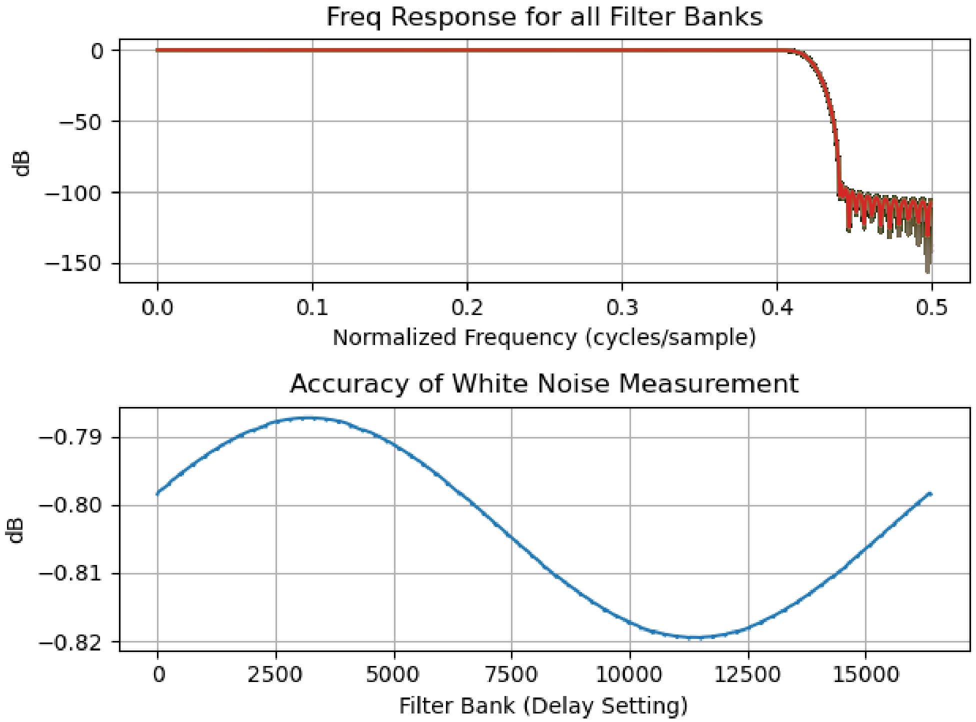
$$\Delta = 10\log_{10}(1 + 10^{x/10})$$





Error in Noise Measurement for White Noise vs Delay Adjustment with and without Prefilter in EVM Tool





Without Prefilter

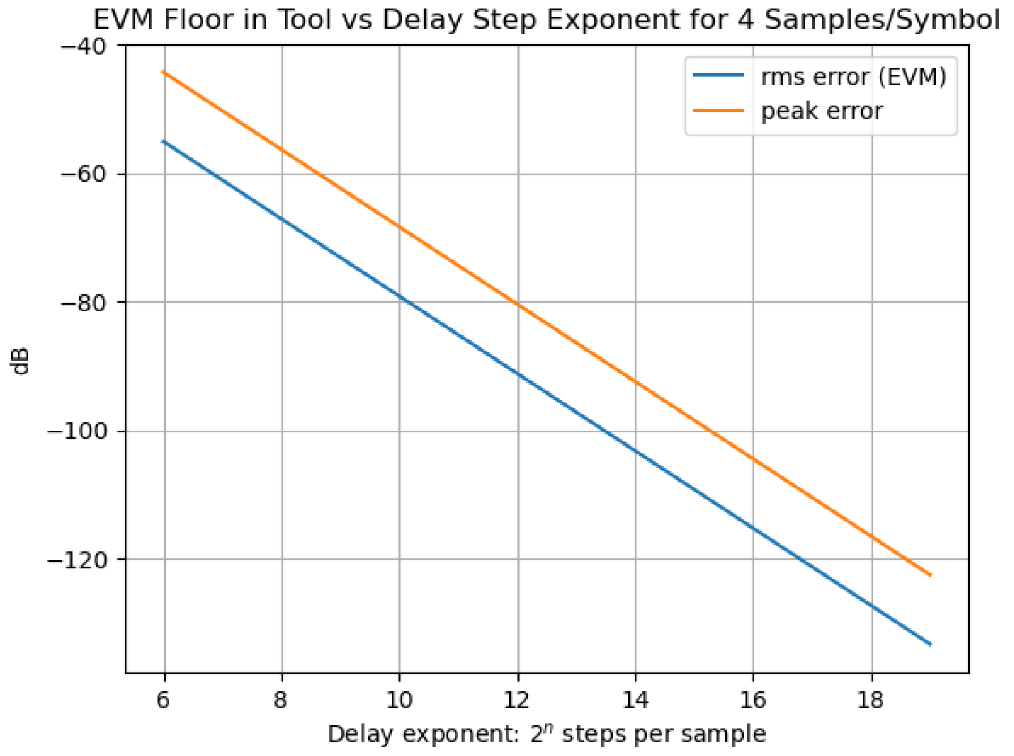
0.5 dB variation, minimum offset

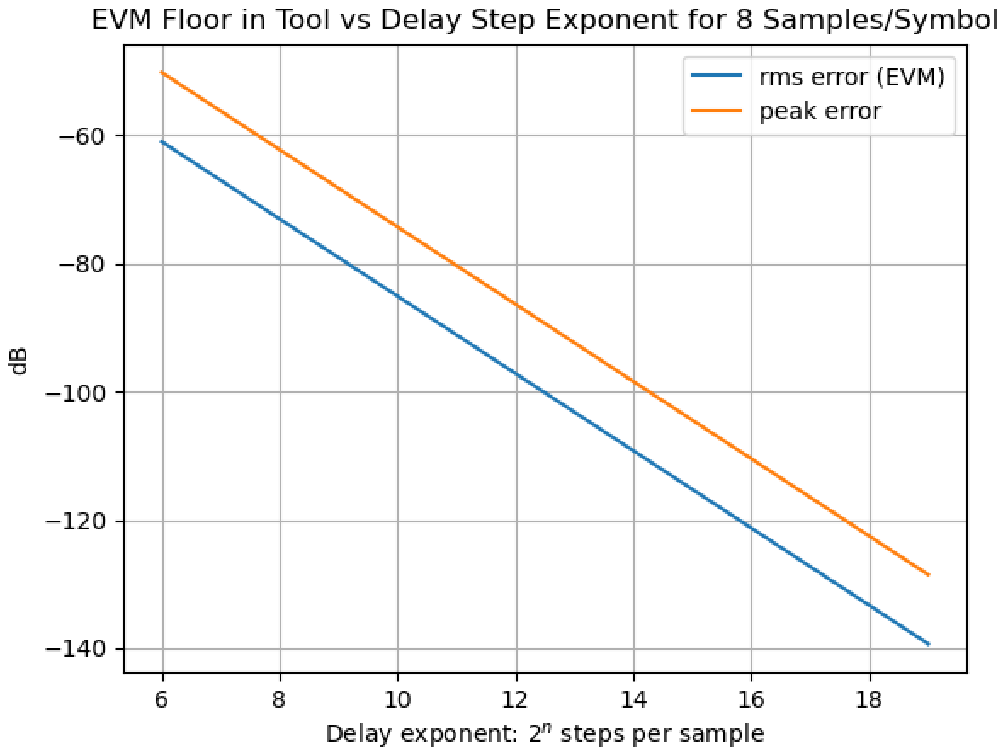
With Prefilter

0.035 dB variation, fixed offset for white noise

(Inconsequential for waveforms that occupy <80 % of Nyquist)

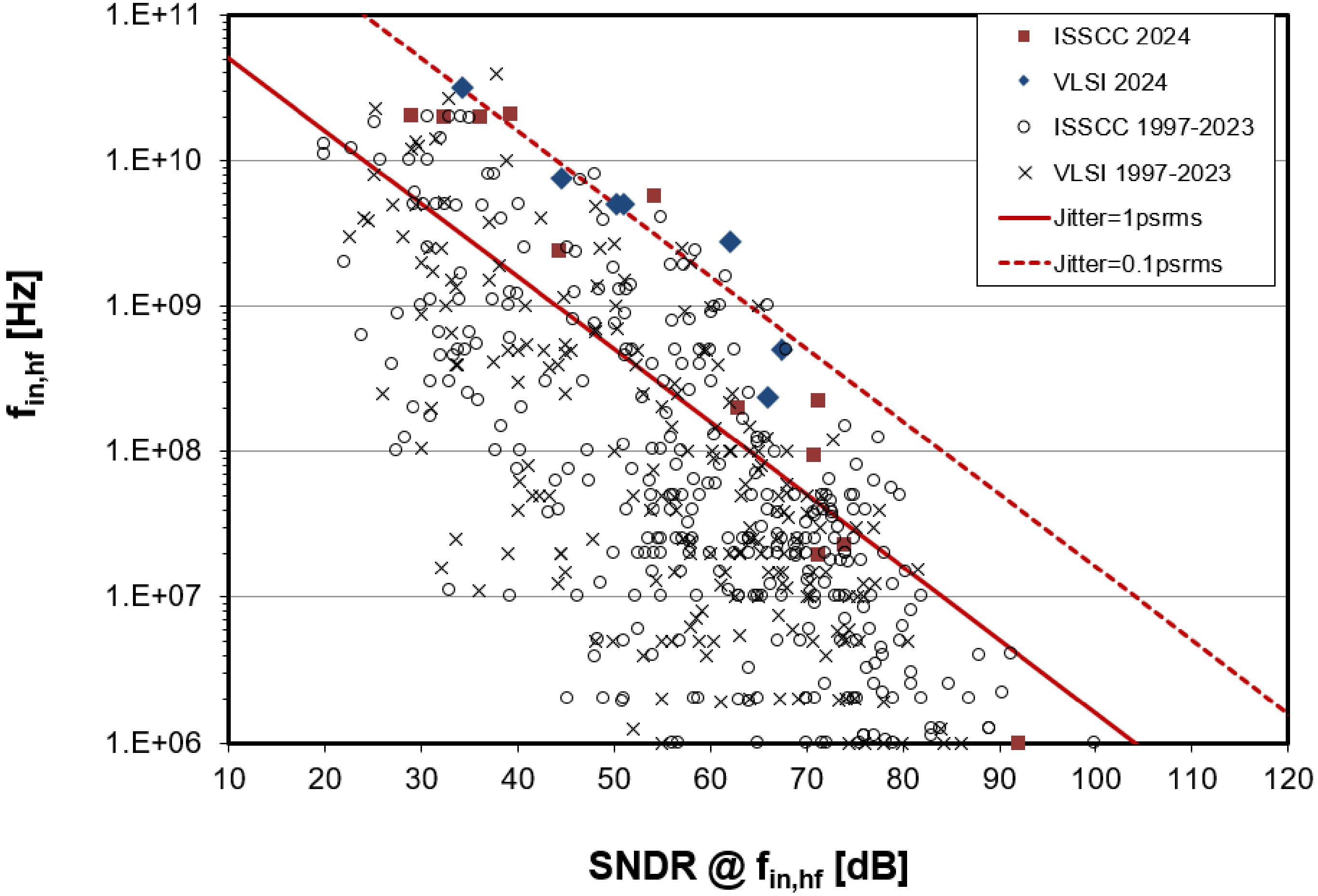
What time step precision is needed?





ADC SOTA

"B. Murmann, "ADC Performance Survey 1997-2024," [Online]. Available: https://github.com/bmurmann/ADC-survey."



dB Scale

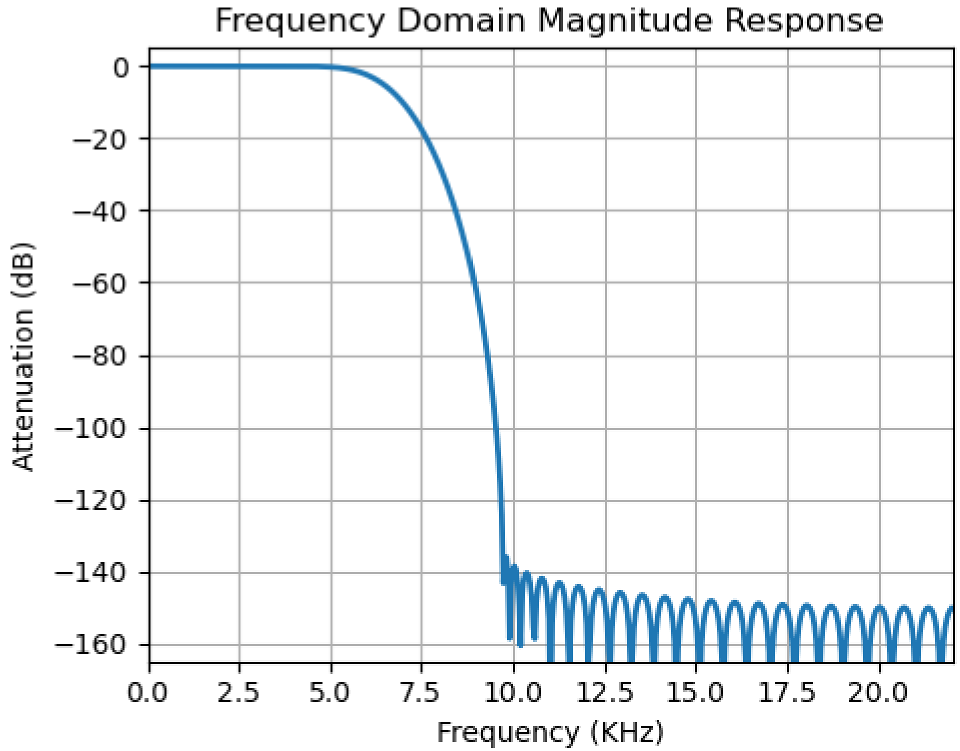


image source: https://www.cochlea.org/en/hear/human-auditory-range

References and Further Reading

harris, fred, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proceedings of the IEEE, vol 66, No 1, Jan 1978

Enrico's Chart of Phase Noise and Two-Sample Variances: https://rubiola.org/pdf-static/Enrico's-chart-EFTS.pdf

Allan, David. *Statistics of Atomic Frequency Standards*, pages 221–230. Proceedings of the IEEE, Vol. 54, No 2, February 1966.

Anders Wallin, The allantools python library: https://allantools.readthedocs.io/en/latest/

P. Welch, “The use of the fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, IEEE Trans. Audio Electroacoust. vol. 15, pp. 70-73, 1967.

Scipy.signal welch function: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.welch.html

My StackExchange posts including detailing non-traditional use of ADEV for non-clock or gyroscope waveforms:

https://dsp.stackexchange.com/questions/82004/how-to-interpret-adev-plot/82005#82005

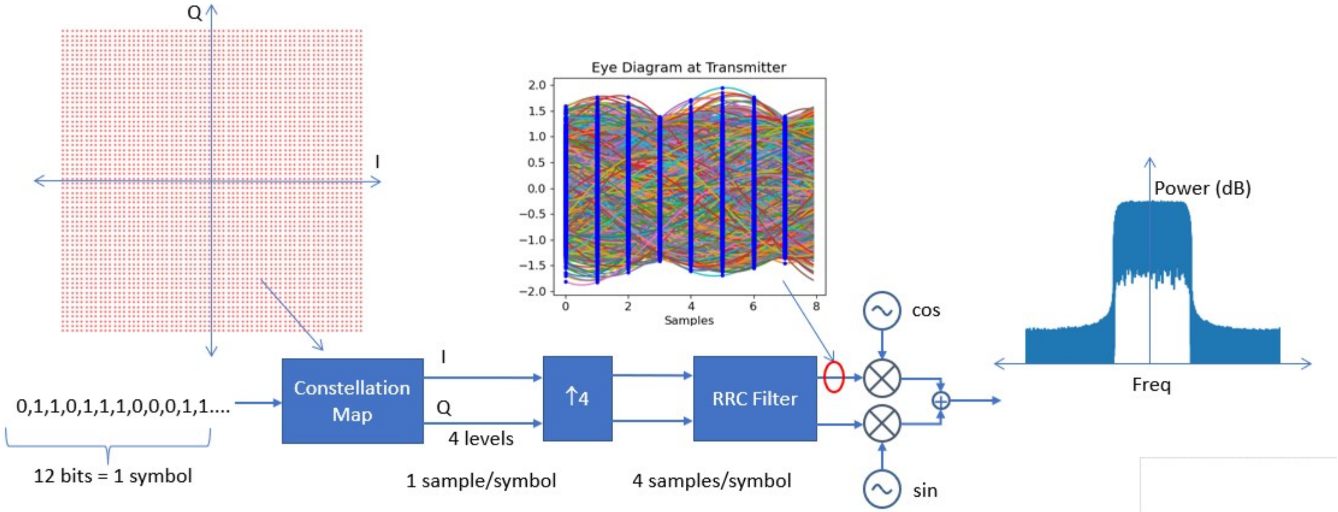
https://dsp.stackexchange.com/questions/88879/1-f-noise-why-does-the-allan-deviation-remain-constant-while-standard-error-of/88892#88892

https://dsp.stackexchange.com/questions/87466/usefulness-of-allan-deviation-with-dc-signals/87468#87468

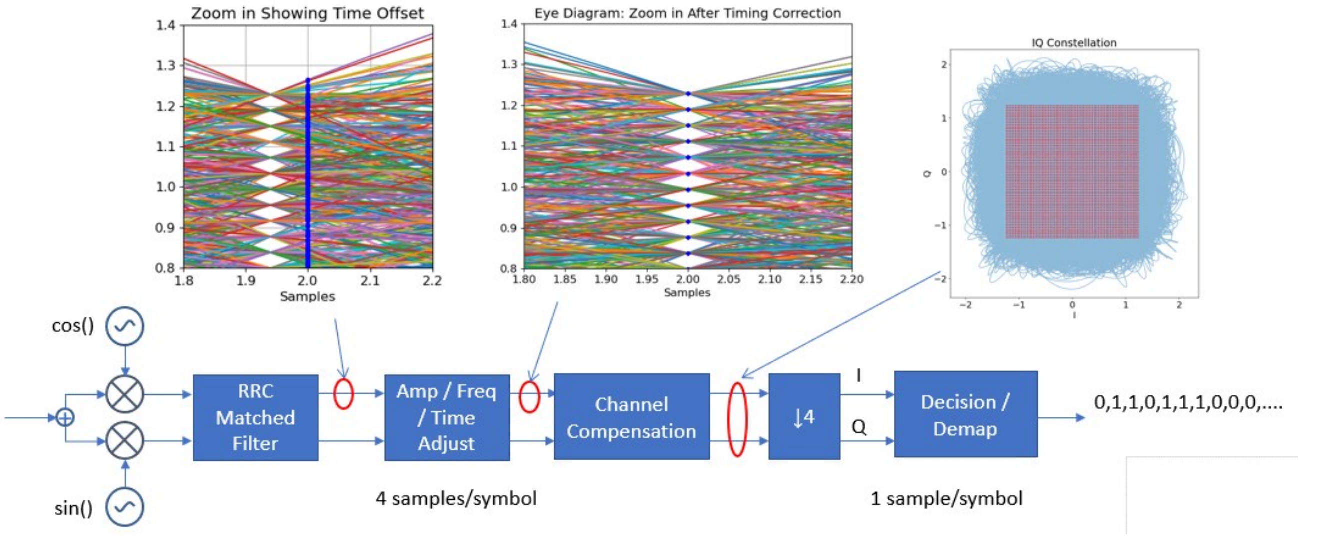
https://dsp.stackexchange.com/questions/83350/fft-to-work-out-optimum-number-of-samples-to-average/83354#83354

https://dsp.stackexchange.com/questions/87402/allan-deviation-to-determine-averaging-time/87403#87403

Example Waveform: 4096-QAM Transmitter



Example Waveform: 4096-QAM Receiver



Banner Graphic for Notebook