FFTs and noise and $\frac{\text{stuff}}{\sqrt{\text{Hz}}}$

WCC

September 3, 2015

I always have to re-learn how to get an FFT to give me real units, so this is my attempt to put this stuff down in one place. I have the same problem with anytime I have to think about something per root Hz, which makes my head hurt, so I hope to write that down here as well.

1 Fourier transforms

We will typically be taking the numerical Fourier transform of a time-domain signal, which I will assume has units of Volts. If we had an analytic expression for this over all time, we could take the Fourier transform directly:

$$v(t) = \int_{-\infty}^{\infty} \mathrm{d}f\ \tilde{V}(f)\, e^{-i2\pi ft} \tag{1}$$

$$\tilde{V}(f) = \int_{-\infty}^{\infty} \mathrm{d}t\ v(t)\, e^{i2\pi ft}. \tag{2}$$

From Eq. 2 it is clear that $\tilde{V}(f)$ must have units of $\mathrm{V{\cdot}s} = \mathrm{V/Hz}$. When studying noise, we typically assume that the phase information is essentially completely scrambled and what we really want to know is how the power or energy is distributed in frequency space. Looking at the units of $\tilde{V}(f)$, it's clear that since the power delivered to some resistance R would be proportional to $|v(t)|^2$ in the time domain, we will be squaring this thing to get power and energy information. Without proof, I will state the definition of the energy spectral density:

$$G_{VV}(f) \equiv |\tilde{V}(f)|^2. \tag{3}$$

The units of $G_{VV}(f)$ in our definition are $\mathrm{V^2/Hz^2} = \mathrm{V^2{\cdot}s/Hz}$, so if we divide ESD by the resistance of the load to which it's being applied, we'd get J/Hz, hence the term "energy spectral density."

The integral of the ESD over the entire frequency domain is the total energy in the signal over the entire time domain, which is known as Parseval's Theorem, and is a great way to check your numerical results. I will assume a resistive load of resistance R to make this look like real energy units:

$$E = \frac{1}{R}\int_{0}^{T} \mathrm{d}t\ |v(t)|^2 = \frac{1}{R}\int_{-1/2T}^{1/2T} \mathrm{d}f\ |\tilde{V}(f)|^2. \tag{4}$$

Those integration limits should be checked, but it's something like this.

Since the energy in a signal will tend to increase with the amount of time we look at it, it's often more convenient to work in power units than energy units. We can then define the Power Spectral Density as follows:

$$S_{VV}(f) \equiv \frac{G_{VV}(f)}{T} = \frac{|\tilde{V}(f)|^2}{T} \tag{5}$$

where T is the total time duration of the signal we're analyzing. The PSD is especially useful for quantifying noise. White noise will have a constant PSD, independent of the duration of the signal and independent of frequency.

The units of $S_{VV}(f)$ in our definition are V^2/Hz, so dividing this by resistance would give you W/Hz, hence the name "power spectral density." Just to be clear, I'll separately define the versions of these quantitites that have a load resistance included:

$$G^R_{VV}(f) \equiv \frac{G_{VV}(f)}{R} \tag{6}$$

$$S^R_{VV}(f) \equiv \frac{S_{VV}(f)}{R} \tag{7}$$

1.1 the spectrum analyzer

So, what the heck does the spectrum analyzer display as the y-axis? I think it displays the power into 50 Ω within the Resolution Bandwidth (RBW) of the instrument, so the y-axis really is in power units, not PSD units. In other words, the y-axis is

$$y_{\text{linear}} = S^R_{VV}(f) \times RBW \tag{8}$$

$$y_{\log} = 10 \log_{10} \left[\frac{S^R_{VV}(f) \times RBW}{1 \text{ mW}} \right] \tag{9}$$

where the units are W or dBm for the linear and log scales, respectively. Since the RBW is usually equal to $1/T$, we could also write this as $y_{\text{linear}} = |\tilde{V}(f)|^2/(R\,T^2)$, but I think that makes the origin of this term less clear.

You have to be very careful when quantifying noise on the spectrum analyzer due to these y-axis units. Imagine, for a moment, that the thing really gave you PSD on the y-axis. Narrow-band signals (all I mean is more narrow than the RBW) would actually get higher and higher on the screen as you increased the frequency resolution by lowering the RBW, while white-noise would stay fixed. This is essentially why they don't do this–you typically want to keep your signal heights fixed as you do things like zoom way in on the x-axis, etc. So they choose instead to display real power units, though you need to remember that it's actually power per unit RBW.

So for narrow-band signals (again, narrow compared to the RBW), the y-axis really is power, and if you have a synthesizer giving you 22 dBm, that peak on the screen will always be at 22 dBm. For noise and other signals that are wider than the RBW, however, it is very important that you *specify the RBW when showing the plot.* Anytime there is a spectrum taken with the spectrum analyzer where you are trying to say anything whatsoever about broad-band signals, you have to specify the RBW, which should probably be written right on the plot. That's the only way to convert those signals into PSD, which is the proper way to specify them.

Remember: the PSD of a narrow-band source may be 10^{784} dBm/Hz, so the total power is usually what you care about there, not the PSD. In contrast, the total power of a noise source depends upon the frequency width over which you integrate it, so the PSD is typically what you care about there. If you integrate a white noise source over a huge RBW, you will again get some absurdly-huge number, say, 10^{784} (dBm in this case).

So now that we have a working definition of PSD ($S_{VV}(f)$), which has units of V^2/Hz, we can ask about how to quantify noise sources on the voltage itself. For instance, let's say I am looking at the noise on the voltage output from some device and I want to know how many volts of actual signal I would need to see it above this noise? Well, this should clearly depend upon the bandwidth I am spanning when I look for my signal, because if I know it's at, say, 100 kHz $\pm$ 1 Hz, I don't care about any of the noise in the 0-99 kHz

range or from 101 kHz on up. The proper quantity to define here is the Voltage Spectral Density,

$$A_V(f) \equiv \sqrt{S_{VV}(f)} = \sqrt{\frac{|\tilde{V}(f)|^2}{T}}. \tag{10}$$

This thing has units of $\mathrm{V}/\sqrt{\mathrm{Hz}}$. What's the deal with that? Well, let's say we're looking for a voltage signal around DC. If I tell you that the voltage noise floor around DC is $A_V^{\mathrm{NF}} = 2\ \mathrm{V}/\sqrt{\mathrm{Hz}}$ and your signal is 1 V, you can immediately see that you won't be able to see it if you integrate for 1 s (because your measurement bandwidth is at least $BW = 1/(1\ \mathrm{s}) = 1$ Hz wide). So you want to average for an even longer time, since your signal is 1 V and your voltage noise has an rms amplitude of $A_V^{\mathrm{NF}} \times \sqrt{BW} = 2$ V. One thing you know about random noise is that the amplitude of the noise decreases like the square root of the averaging time, so that's consistent with this picture. If my noise amplitude is 2 times my signal amplitude, I clearly need to average 4 times as long! That's why these wacky units of stuff$/\sqrt{\mathrm{Hz}}$ make sense. They automatically tell you useful information about your sensitivity by quantifying the noise in the appropriate units. If you want to know how long you need to measure to see a (transform-limited) signal emerge from the noise, you need only compare your signal level to the noise spectral density divided the sqaure root of the measurement time (a.k.a $A_V^{\mathrm{NF}} \times \sqrt{BW}$).

Last, as pointed out to me by Amar Vutha, anytime you specify a signal-to-noise ratio (SNR), you should either explicitly specify the measurement bandwidth, or, better yet, quote it with units of $1/\sqrt{\mathrm{Hz}}$. Pretty much anything dealing with noise will have similar considerations involved.

Example 1.1 Most photodetectors output a voltage that is proportional to the incident optical power. The detector itself will have noise on this voltage, so one way to specify this noise would be to quote a VSD and be done with it. However, it's slightly more useful to include the optical-power-to-voltage conversion in this number, which is called the Noise Equivalent Power (NEP). The units of NEP are $\mathrm{W}/\sqrt{\mathrm{Hz}}$, so to figure out what minimum optical power is necessary to see signal above the noise, you need only to determine the measurement bandwidth and multiply its square root by the NEP.

2 FFTs and their units

The numerical Fourier transforms that are available as parts of software packages tend to rely on an algorithm called the Fast Fourier Transform (FFT). The function `Fourier[]` in Mathematica, the function `np.fft.fft()` in numpy, and `fft()` in Matlab all use this algorithm.

For reasons I don't understand, the algorithm works best if you give it a numerical signal that has been sampled a number of times N_{samples} such that N_{samples} is a power of 2. So some of these algorithms will actually pad your time-domain data with zeros to make the total number of samples equal to 2^m for some integer m.

The next thing to realize is that these algorithms don't know anything about the time-domain units. You don't send them ordered pairs of voltage and time, but rather just an array of voltages. So the output units of the thing are something like "Volts per point" or some nonsense, which we will have to divide by Hz/point to get into legit units.

Last, there are kooky conventions out there about whether there are factors of $1/\sqrt{N_{\mathrm{samples}}}$ in front of things, so the algorithms I present here are based entirely on my using Parseval's theorem (Eq. 4) to figure out what to put in front of what.

2.1 Noise in the time-domain

It is useful for both data analysis and simulation of noise to be able to convert between RMS noise in the time-domain and noise power / spectral density in the frequency domain. This requires that one specify the sampling bin width (in time) of the time-domain signal, because that will determine how much RMS noise you see. To see this, imagine that you take a white noise signal and look at it in time while making the width

of the time bins finer and finer. As you do this, the amplitude of the noise you see will increase. Wider time bins permit you to "average" longer during the bin, resulting in smaller deviations from the mean value.

Here are a whole bunch of useful formulas. For real power units, use the actual load resistance for R. To keep things in voltage units, ignore R.

quantity	symbol	formula	units
measurement duration	T		s
number of samples	N_{samples}		
noise power		$\frac{V_{\text{rms}}^2}{R}$	W
total bandwidth	$f_{\max}$	$\frac{N_{\text{samples}}}{T}$	Hz
total energy	E	$\frac{V_{\text{rms}}^2}{R} \times T$	J
energy spectral density (ESD)	$\mathrm{G}_{VV}^R(f)$	$\frac{E}{f_{\max}} = \frac{V_{\text{rms}}^2}{R} \times \frac{T^2}{N_{\text{samples}}}$	J/Hz
power spectral density (PSD)	$S_{VV}^R(f)$	$\frac{G_{VV}^R(f)}{T} = \frac{V_{\text{rms}}^2}{R} \times \frac{T}{N_{\text{samples}}}$	W/Hz
resolution bandwidth	RBW	$\frac{1}{T}$	Hz
power in RBW (linear)	y_{linear}	$S_{VV}^R(f) \times RBW = \frac{V_{\text{rms}}^2}{R} \times \frac{1}{N_{\text{samples}}}$	W
power in RBW (log)	$y_{\log}$	$10 \log_{10} \left[\left(\frac{V_{\text{rms}}^2}{R} \times \frac{1}{N_{\text{samples}}} \right) / (1 \text{ mW}) \right]$	dBm

Example 2.1 A 50 Ω spectrum analyzer shows that a signal has a flat noise floor at −25 dBm when the resolution bandwidth is set to 1 MHz. (a) What is the power spectral density of this noise? (b) Where will the noise floor move to on the spectrum analyzer y-axis if you change the RBW to 10 kHz? (c) What RMS voltage noise does this correspond to if you sample this signal at 10^7 samples per second? (Answers: (a) 3.2 pW/Hz, (b) −45 dBm, (c) 40 mV)

2.2 Mathemtaica

If we are given a time-domain signal as a pair of arrays `Vees` and `Tees` with `Vees` in volts and `Tees` in seconds, the following snippet of code will generate the relevant FFT-related quantities.

```
If[OddQ[Length[Tees]], Tees = Drop[Tees, -1]; Vees = Drop[Vees, -1]];
NumSamples = Length[Tees];
SignalDuration = Tees[[-1]] - Tees[[1]]; (* seconds *)
LoadResistance = 50; (* Ohms.  Set this to 1 to keep your units in Volts.  *)
ResolutionBandwidth = 1/SignalDuration; (* Hz *)
(* DS = double-sided, SS = single-sided *)
DSFrequencyArray = Table[(j - NumSamples/2)*ResolutionBandwidth, {j, 0, NumSamples-1}];(*Hz*)
DSFFT = Chop[RotateRight[Fourier[Vees, FourierParameters -> {0,1}], NumSamples/2]]
                          /(ResolutionBandwidth*Sqrt[NumSamples]); (* V/Hz *)
SSFrequencyArray = Drop[DSFrequencyArray, NumSamples/2]; (* Hz *)
SSFFT = Sqrt[2] * Drop[DSFFT, NumSamples/2]; (* V/Hz *)
ESD = Conjugate[SSFFT]*SSFFT/LoadResistance; (* J/Hz *)
PSD = ESD/SignalDuration; (* W/Hz *)
VoltageSpectralDensity = Sqrt[PSD*LoadResistance]; (* V/Sqrt[Hz] *)
PowerInRBW = PSD*ResolutionBandwidth; (* W *)
dBmInRBW = 10*Log[10, PowerInRBW*10^3]; (* dBm *)
```

I have tried to make it clear where I am taking the double-sided Fourier domain functions and turning them into single-sided quantities.

There is a very sneaky factor of $1/\sqrt{N_{\text{samples}}}$ in there, so be sure to check that you have this. You can always run a sanity check by integrating the instantaneous time-domain power over the total time duration and seeing if it matches the integral of the ESD over the total frequency domain.