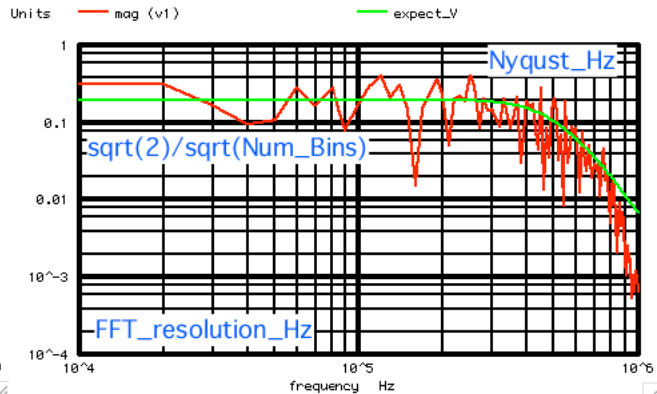
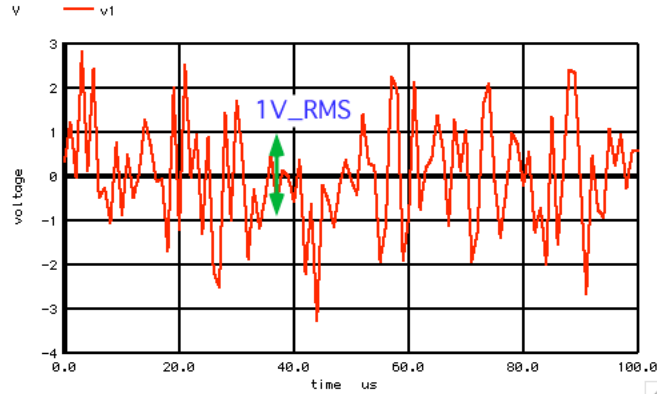


=====**Randomn\_Transient\_Spice\_Simulation**=====

- |                                       |                                      |
|---------------------------------------|--------------------------------------|
| 1) Noise_pk is inconvenient           | = Noise_rms is better format         |
| 2) Signal_rms                         | = sqrt(2)*Signal_pk                  |
| 3) Create a 1V_rms Noise array        | = install it as a PWL signal and run |
| 4) Noise spreads evenly over FFT bins | = sqrt(2)/sqrt(Num_Bins)             |
| 5) Num_Bins for FFT                   | = Nyquist_Hz/FFT_resolution_Hz       |
| 6) FFT_resolution_Hz                  | = 1/total_time_sec                   |
| 7) Nyquist_Hz                         | = 0.5/Sample_period_sec              |



- 0) For this version of spice, a 1V\_pk\_100%\_AM signal will produce a carrier signal with a magnitude of one with two 0.5 magnitude sidebands.
- 1) But using a peak value for noise is inconvenient since it depends on probability and time. A RMS value to onis is more convenient.
  - 2) When using RMS, remember a rms value is sqrt(2) larger than a peak values. A 1V\_pk is measured as a .707V\_rms value

```

=====Want_100_us_steps=====
Total_Period_s = 0.0001
Bin_Resolutio Hz = 10000
Sample_Period_s = 1E-06
Nyquist_Hz = 500000
=====Create_PWL_array_and_Index_and_Plot=====
=====Add_1Vrms_Noise_to_PWL_array=====
=====Install_the_PWL_array=====
=====Run_and_Plot=====
=====Find_Ave_Rms=====
Average_level_Expect 0 Average_level -0.00406862
RMS_level_Expect 1 RMS_level 1.05424
=====FFT_and_Plot=====
FFT_BandWidth_Hz= 1E+06
FFT_resolution_Hz= 10000
=====done=====

```

- 3) A random 1V\_rms signal can be added to an array and installed as a PWL voltage source.
- 4) The random 1V\_rms signal will get spread evenly over all the FFT frequency bins.
- 5) The number of bins is set by Nyquist\_Hz/FFT\_resolution\_Hz.
- 6) Total sample time defines the minimum frequency of the spectrum or FFT\_resolution\_Hz.
- 7) The maximum frequency (Nyquist\_Hz) will be visible on the spectrum.

=====**MacSpiceCode**=====

```

Randomn_Transient_Spice_Simulation
*=====Need_A_voltage_Source_to_alter=====
v1 v1 0 0 dc
.control
set pensize = 2
echo "=====Want_100_us_steps=====
let n = 100
let tstep = lus
let period_t = n*tstep
let Bin_Hz = 1/period_t
let nyquist = .5/tstep
echo "Total_Period_s = $&period_t"
echo "Bin_Resolutio_Hz = $&Bin_Hz"
echo "Sample_Period_s = $&tstep"
echo "Nyquist_Hz = $&nyquist"
echo "=====Create_PWL_array_and_Index_and_Plot=====
let pwl_1 = vector(2*n)*tstep*0.5
let ii = vector(2*$&n)
echo "=====Add_1Vrms_Noise_to_PWL_array=====
let index = 0
repeat $&n
let pwl_1[1+2*index] = 1.2*(rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)-507.5)/102.879
let index = index + 1
end
echo "=====Install_the_PWL_array=====
alter @v1[pwl] = pwl_1
echo "=====Run_and_Plot=====
let period_s = tstep/2
let trans_per = tstep/20
tran $&trans_per $&period_t 0 $&trans_per

```

```

plot          v1
echo          "=====Find_Ave_Rms===== "
let averVal = mean(v1)
let noisAC = v1 - averVal
let RmsVal = sqrt(mean(noisAC* noisAC))
echo          "Average_level_Expect 0   Average_level $&averVal "
echo          "RMS_level_Expect 1   RMS_level $&RmsVal "
unlet averVal
unlet RmsVal
echo          "=====FFT_and_Plot===== "
linearize
let          FFT_BandWidth_Hz = 1Meg
let          FFT_resolution_Hz = 10k
echo          "FFT_BandWidth_Hz= $&FFT_BandWidth_Hz"
echo          "FFT_resolution_Hz= $&FFT_resolution_Hz"
set          specwindow= "rectangular"
spec          $&FFT_resolution_Hz $&FFT_BandWidth_Hz $&FFT_resolution_Hz v(v1)
let expect_V = (sqrt(2)/sqrt(500k/10k))/(1+(frequency/500k)*(frequency/500k)*(frequency/500k)*(frequency/500k))
plot          mag (v1) expect_V loglog
echo          "=====done===== "
.endc
.end

```

4.4.11\_10.55AM  
dsauersanjose@aol.com  
Don Sauer