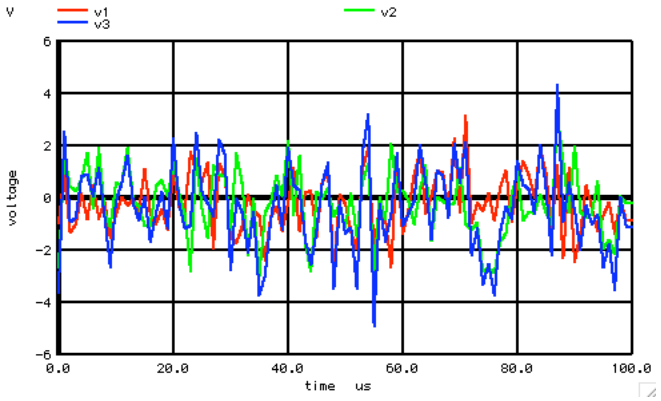
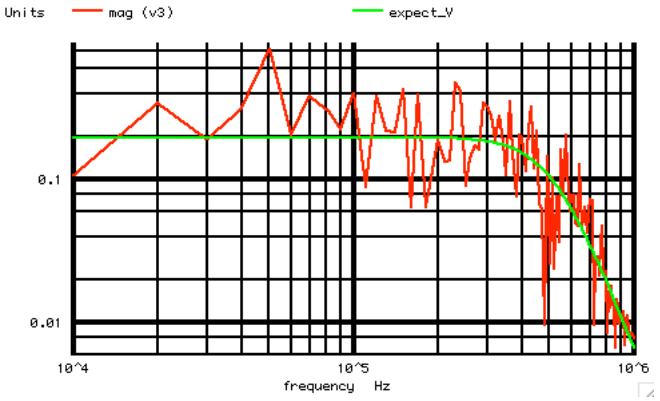
==========Noise_Adds_With_Power==========

1) The laws of thermal dynamics apply to random signal.
2) Adding two equal level of power produces twice the power.
3) The equivalent voltage or current is found from the RMS.





This is a working demo were two independent 1V_rms waveforms (v1 and v2) have been added together to produce v3 which has a 1.414_rms result.

```
==========Want_100_1us_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_Rms1==========
Average_level_Expect    0       Average_level1 -0.216172
RMS_level_Expect        1       RMS_level1      0.948643
==========Find_Ave_Rms2==========
Average_level_Expect    0       Average_level2 -0.231652
RMS_level_Expect        1       RMS_level2      1.10187
==========Find_Ave_Rms3==========
Average_level_Expect    0.000   Average_level3 -0.447824
RMS_level_Expect        1.414   RMS_level3      1.43048
==========FFT_and_Plot==========
FFT_BandWidth_Hz=       1E+06
FFT_resolution_Hz=      10000
==========done==========
```

==========MacSpiceCode==========

Noise_Adds_With_Power

```
*==========Need_A_voltage_Source_to_alter==========
V1              V1      0       0     dc
V2              V2      0       0     dc
BV3             V3      0       V=    V(V1)+V(V2)
.control
set             pensize = 2
echo            "==========Want_100_1us_steps==========
let n =         100
let tstep =     1us
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 =            $&nyquist"
echo            "==========Create_PWL_array_and_Index_and_Plot=========="
let pwl_1 =     vector(2*n)*tstep*0.5
let pwl_2 =     vector(2*n)*tstep*0.5
let ii =        vector(2*$&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             pwl_2[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
alter           @v2[pwl] =              pwl_2
echo            "==========Run_and_Plot=========="
let period_s =  tstep/2
let trans_per = tstep/20
tran            $&trans_per $&period_t  0  $&trans_per
plot            v1 v2 v3
echo            "==========Find_Ave_Rms1=========="
let averVal =   mean(v1)
let noisAC =    v1 - averVal
let RmsVal =    sqrt(mean(noisAC* noisAC))
echo            "Average_level_Expect    0       Average_level1 $&averVal "
```

```
echo            "RMS_level_Expect       1      RMS_level1     $&RmsVal  "
unlet averVal
unlet RmsVal
echo            "======================Find_Ave_Rms2======================================="
let averVal =   mean(v2)
let noisAC =    v2 - averVal
let RmsVal =    sqrt(mean(noisAC* noisAC))
echo            "Average_level_Expect   0      Average_level2 $&averVal "
echo            "RMS_level_Expect       1      RMS_level2     $&RmsVal  "
unlet averVal
unlet RmsVal
echo            "======================Find_Ave_Rms3======================================="
let averVal =   mean(v3)
let noisAC =    v3 - averVal
let RmsVal =    sqrt(mean(noisAC* noisAC))
echo            "Average_level_Expect   0.000  Average_level3 $&averVal "
echo            "RMS_level_Expect       1.414  RMS_level3     $&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(v3)
let expect_V =  (sqrt(2)/sqrt(500k/10k))/(1+(frequency/550k)*(frequency/500k)*(frequency/500k)*(frequency/500k)*(frequency/500k))
plot            mag (v3) expect_V loglog
echo            "======================done================================================"
.endc
.end
```

4.4.11_10.59AM
dsauersanjose@aol.com
Don Sauer