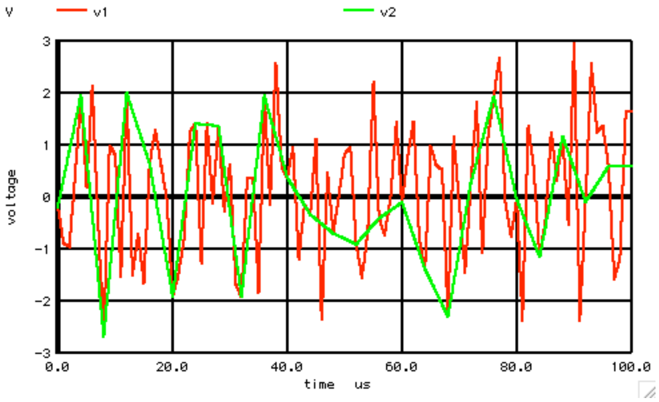
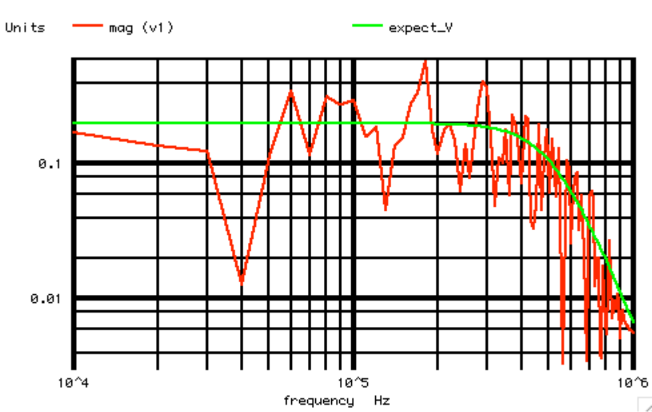
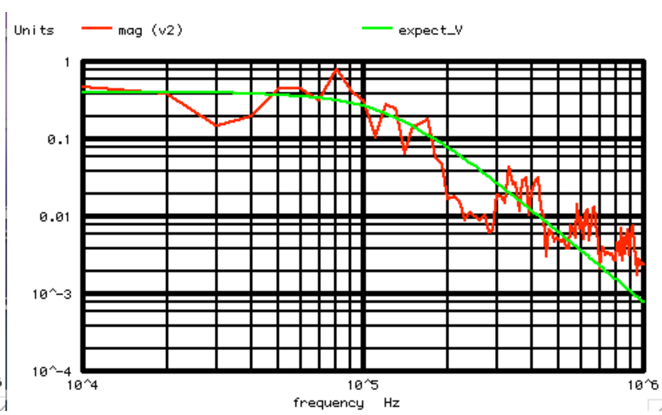
=====A RANDOM RMS VALUE IS SAMPLE RATE INDEPENDANT=====



Regardless of what rate a 1Vrms noise signal gets sampled at,
its data still has the same 1V standard deviation.
This is provided the noise signal has a high bandwidth.

```
=======================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=====================
=======================UnderSample_By_one_Fourth=====================
=======================Install_the_PWL_array============================
=======================Run_and_Plot=====================================
=======================Find_Ave_Rms1=====================================
Average_level_Expect    0      Average_level1 0.106446
RMS_level_Expect        1      RMS_level1      1.07804
=======================Find_Ave_Rms2=====================================
Average_level_Expect    0      Average_level2 -0.0106962
RMS_level_Expect        1      RMS_level2      1.02624
=======================FFT_and_Plot_V1===================================
FFT_BandWidth_Hz=       1E+06
FFT_resolution_Hz=      10000
Noise_Per_10KHz=        0.2
=======================FFT_and_Plot_V2===================================
FFT_BandWidth_Hz=       1E+06
FFT_resolution_Hz=      10000
Noise_Per_10KHz=        0.399795
===================done============================================
```





Lowering the sample rate only changes the Nqyuist.
It only changes the number of frequency bins that store the RMS input noise.

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

Under_Sample_Noise

```
*================Need_A_voltage_Source_to_alter====================="
V1              V1      0       0     dc
V2              V2      0       0     dc
.control
set             pensize = 2
echo            "=======================Want_100_1us_steps==============================="
let n =         100
let n2 =        25
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=============="
unlet pwl_1
```

```
unlet pwl_2
let pwl_1 =      vector(2*n)*tstep*0.5
let pwl_2 =      vector(.5*n)*tstep*2
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             "=======================UnderSample_By_one_Fourth====================="
let index =      0
repeat           $&n2
let              pwl_2[1+2*index] = pwl_1[1+8*index]
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
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             "=======================FFT_and_Plot_V1===================================="
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/550k)*(frequency/500k)*(frequency/500k)*(frequency/500k)*(frequency/500k))
plot             mag (v1) expect_V loglog
let              Nois_per10K  = expect_V[0]
echo             "Noise_Per_10KHz=       $&Nois_per10K"
echo             "=======================FFT_and_Plot_V2===================================="
destroy
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(v2)
let expect_V =   (2*sqrt(2)/sqrt(500k/10k))/(1+(frequency/125k)*(frequency/125k)*(frequency/125k))
plot             mag (v2) expect_V loglog
let              Nois_per10K  = expect_V[0]
echo             "Noise_Per_10KHz=       $&Nois_per10K"
echo             "===================done==========================================="
.endc
.end
```

4.18.11_1.08PM
dsauersanjose@aol.com
Don Sauer