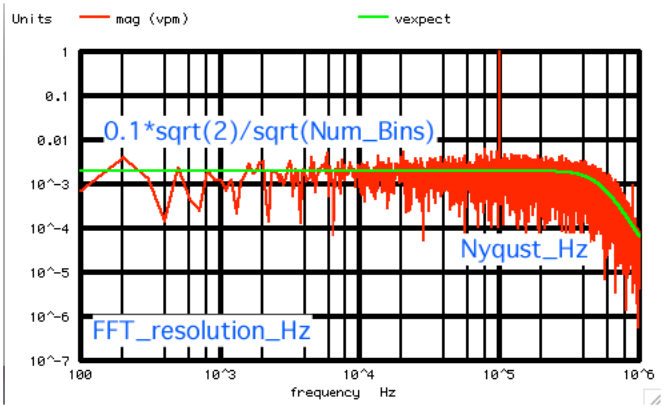
==========SIMPLE_RANDOM_PM_WAVEFORM_GENERATION==========

1) 0.1_radian_pk_PM = 1_carrier_pk and 0.05_sidebands_pk
2) Noise_pk is inconvenient = Noise_rms is better format
2) Signal_rms = sqrt(2)*Signal_pk
3) PM_noise spread evenly over FFT bins = 1/sqrt(Num_Bins)
4) FFT_resolution_Hz = 1/total_time_sec
5) Nyqust_Hz = 0.5/Sample_time_sec
6) Num_Bins for FFT = Nyqust_Hz/FFT_resolution_Hz



This is a working demo on how a +/-0.1 radian PM signal will spread the energy of the noise spectrum evenly over all the frequency Bins of a FFT.

```
==========Want_10000_1us_steps==========
Total_Period_s =        0.01
Bin_Resolutio_Hz =      100
Sample_Period_s =       1E-06
Nyquist_Hz =            500000
Total_Bins =            5000
==========Create_PWL_array_and_Index_and_Plot==========
==========Add_.1Vrms_Noise_to_PWL_array==========
==========Find_Ave_Rms_pwl_1==========
RMS_level_Expect        .1    RMS_level_RM   0.10019
==========Install_the_PWL_array==========
==========FFT_and_Plot_V2==========
FFT_BandWidth_Hz=       1E+06
FFT_resolution_Hz=      100
vexpect dc should be    .1*sqrt(2)/sqrt(5000)  which is 1.414m
```

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

SIMPLE_RANDOM_PM_WAVEFORM_GENERATION

```
*=========Create_Signal==========
VTime       VTime   0       DC      0     PWL(     0       0     1       1)
Vfreq1      Vfreq1  0       DC      2
V1          V1      0       DC      0
BMOD        VMOD    0       V       = cos(6.2831853*2000*V(VTime))
BPM         VPM     0       V       = 1*cos(6.2831853*100k*V(VTime)+.1*V(V1))
BCOS        VCOS    0       V       = 1*cos(6.2831853*100k*V(VTime))
```

.control

```
echo                "==========Want_10000_1us_steps=========="
let n =             10000
let tstep =         1us
let period_t =      n*tstep
let Bin_Hz =        1/period_t
let nyquist =       .5/tstep
let binsTotal=      nyquist/Bin_Hz
echo                "Total_Period_s =        $&period_t"
echo                "Bin_Resolutio_Hz =      $&Bin_Hz"
echo                "Sample_Period_s =       $&tstep"
echo                "Nyquist_Hz =            $&nyquist"
echo                "Total_Bins =            $&binsTotal"
echo                "==========Create_PWL_array_and_Index_and_Plot=========="
let pwl_1 =         vector(2*n)*tstep*0.5
let ii =            vector(2*$&n)
echo                "==========Add_.1Vrms_Noise_to_PWL_array=========="
let index =         0
repeat              $&n
let                 pwl_1[1+2*index] = 1.414*(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                "==========Find_Ave_Rms_pwl_1=========="
let averVal =       mean(pwl_1)
let noisAC =        pwl_1 - averVal
let RmsVal =        .1*sqrt(mean(noisAC* noisAC))
echo                "RMS_level_Expect        .1    RMS_level_RM   $&RmsVal   "
unlet averVal
unlet RmsVal
echo                "==========Install/run_PWL_array=========="
alter               @v1[pwl] = pwl_1
*TRAN               TSTEP   TSTOP   TSTART TMAX    ?UIC?
tran                .1u     10m     0       .1u
set                 pensize = 2
echo                "==========FFT_and_Plot_V2=========="
```

```
linearize
let             FFT_BandWidth_Hz =      1meg
let             FFT_resolution_Hz =     100
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(vpm)
let vexpect =   (.1*sqrt(2)/sqrt(5000))/(1+(frequency/550k)*(frequency/500k)*(frequency/500k)*(frequency/500k)*(frequency/500k))
plot            mag (vpm) vexpect loglog
echo            "vexpect dc should be   .1*sqrt(2)/sqrt(5000)  which is 1.414m "
.endc
.end
```

4.4.11_12.06PM

dsauersanjose@aol.com
Don Sauer