===================Accumulated_Randomness=====================================

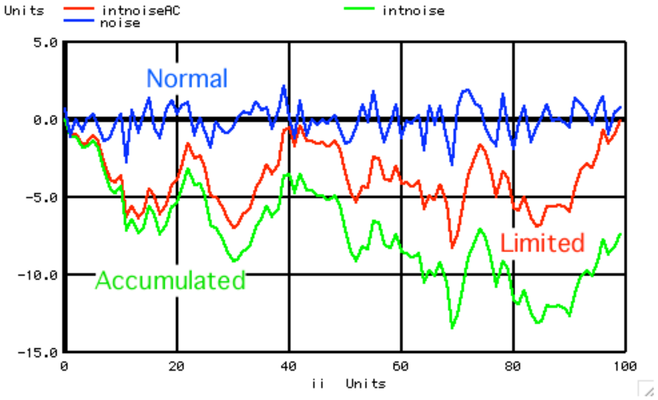
1) Normal Randomness is when each sample is independent of the previous sample.
2) Standard deviation is the RMS of data with the average removed.
3) For Normal Randomness, more points makes the average and sd more precise
4) Normal Randomness with N samples, ave_calc = ave_real +/sd/sqrt(N)
5) Normal Randomness with N samples, sd_calc = sd_real +/sd/sqrt(N*2)

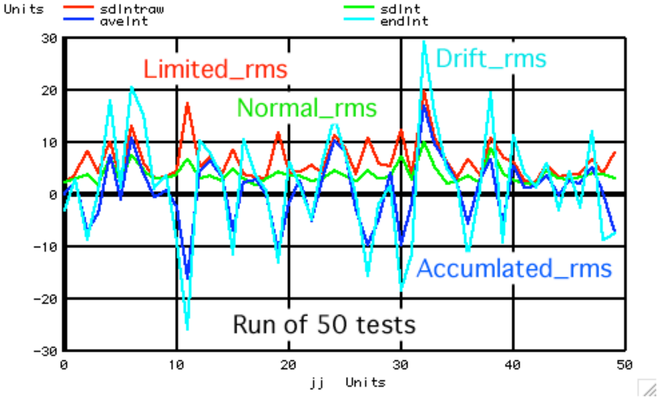
6) Accumulated Randomness is when each sample is added to the sum of the previous samples
7) Accumulated Randomness has an endpoint that drifts over time.
8) For N samples, Standard deviation of the endpoint drift is sd*sqrt(N).
   This is because randomness adds with power.

This spice simulation tests Accumulated Randomness to yield the following...

9) The RMS value of Accumulated Randomness is the power sum of AC randomness and drift.
10) The RMS value of Accumulated Randomness approaches sd*sqrt(N)/sqrt(2)

11) Limited Accumulated Randomness is when accummulation becomes limited over N Samples.
    In other words, when endpoint drift starts to approachs zero over a long enough period.
12) The RMS value of Limited Accumulated Randomness approaches sd*sqrt(N)/sqrt(2*4)





Numb	AVE	SD	AveErr_SD	SDErr_SD	Int_EndErr_SD	Int_AveErr_SD	Int_RMS_SD	IntAC_RMS_SD
N	**0**	**0**	**1/sqrt(N)**	**.707/sqrt(N)**	**sqrt(N)**	**sqrt(N)/2**	**.707*sqrt(N)**	**.707*sqrt(N)/2**
10000	0	1	0.0102146	0.00816485	120.704	58.1437	79.211	42.3499
10000	0	1	0.0104502	0.00646921	123.299	54.7969	77.8572	34.8961
10000	0	1	0.00924566	0.00817316	116.63	50.4919	71.2871	38.5661
10000	**0**	**1**	**0.01**	**0.00707**	**100.00**	**50.000**	**70.7**	**35.35**
1000	0	1	0.0311	0.0231	31.2436	18.0411	22.2068	12.2452
1000	0	1	0.03241	0.019070	37.3343	17.9726	25.9551	12.7892
1000	0	1	0.03214	0.0235831	32.4218	17.0015	21.6558	14.1455
1000	**0**	**1**	**0.0316227**	**0.02236**	**31.622**	**15.81138**	**22.36**	**11.1803**
100	0	1	0.0860596	0.0747909	8.53102	4.85218	6.21992	4.34292
100	0	1	0.0993542	0.0669429	10.3007	5.96419	7.1801	4.40603
100	0	1	0.0996981	0.0702651	10.7141	5.76873	7.5566	3.7209
100	0	1	0.0963	0.0737681	9.79913	5.59188	6.98267	3.7243
100	**0**	**1**	**0.1**	**0.0707**	**10.00**	**5.000**	**7.07**	**3.535**

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

```
Accumulated_Randomness
*================Need_A_voltage_Source_to_alter====================="
V1                V1       0       0    dc
.control
set               pensize = 2

*echo             "===================k_tests====================="
unlet             aveave2
unlet             sdave2
unlet             avesd2
unlet             sdsd2
unlet             kk
let aveave2 =     vector(50)
let sdave2 =      vector(50)
let avesd2 =      vector(50)
let sdsd2 =       vector(50)
let kk =          vector(50)
let intnoissd =   vector(50)
let intnoissdAC   = vector(50)
*echo             "===================j_tests_Arrays====================="
unlet             sd
unlet             sdraw
unlet             ave
unlet             sdInt
unlet             sdIntraw
unlet             aveInt
unlet             sdIntAC
unlet             sdIntACraw
unlet            aveIntAC
unlet            jj
let sd =         vector(50)
let sdraw =      vector(50)
let ave =        vector(50)
let endInt =     vector(50)
let sdInt =      vector(50)
let sdIntraw =   vector(50)
let aveInt =     vector(50)
let sdIntAC =    vector(50)
let sdIntACraw = vector(50)
let aveIntAC =   vector(50)
let jj =         vector(50)
*echo            "===================create_number_points_Arrays====================="
let n =          100
unlet            noise
unlet            intnoise
unlet            intnoiseAC
unlet            noisAC
unlet            ii
let noise =      vector($&n)
let Intnoise =   vector($&n)
let IntnoiseAC = vector($&n)
let ii =         vector($&n)
let noisAC =     vector($&n)
*echo            "===================loop_j====================="
let j =          0
repeat           50
*echo            "===================create_noise_array====================="
let index =      0
repeat           $&n
let              ii[index] = index
let              noise[index] = 1.0*(rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)+rnd(127)-507.5)/102.879 +.04
let index =      index + 1
end
*plot             noise  vs ii
*echo            "===================create_Integrated_noise_array====================="
let              intnoise[0] =0
let index =      1
let n2 =         n -1
repeat           $&n2
let              intnoise[index] = noise[index]+intnoise[index-1]
let index =      index + 1
end
*plot             intnoise noise  vs ii
*echo            "===================create_AC_Integrated_noise_array====================="
let index =      0
repeat           $&n
let              intnoiseAC[index] = intnoise[index] - intnoise[n-1]*index/n
let index =      index + 1
end
*plot              intnoiseAC intnoise noise  vs ii
*echo            "===================Find_Ave_Rms_Noise====================================="
let averVal =    mean(noise)
let noisAC =     noise - averVal
let RmsVal =     sqrt(mean(noisAC* noisAC))
let RmsRawVal =  sqrt(mean(noise* noise))
*echo             "noise"
*echo             "number Points      $&n"
*echo             "Average level      $&averVal"
*echo             "RMS level          $&RmsVal"
let              sd[j]  = RmsVal
let              ave[j] = averVal
let              sdraw[j]  = RmsRawVal
*echo            "===================Find_Ave_Rms_IntegrateNoise====================================="
let averVal =    mean(intnoise)
let noisAC =     intnoise - averVal
let RmsVal =     sqrt(mean(noisAC* noisAC))
let RmsRawVal =  sqrt(mean(intnoise*intnoise))
let endpt     =  intnoise[n2]
*echo             "Integreated_noise"
*echo             "number Points      $&n"
*echo             "Average level      $&averVal"
*echo             "RMS level          $&RmsVal"
let              sdInt[j]  = RmsVal
let              aveInt[j] = averVal
let              sdIntraw[j]  = RmsRawVal
let              endInt[j] = endpt
*echo            "===================Find_Ave_Rms_AC_IntegrateNoise====================================="
let averVal =    mean(intnoiseAC)
let noisAC =     intnoiseAC - averVal
let RmsVal =     sqrt(mean(noisAC* noisAC))
let RmsRawVal =  sqrt(mean(intnoiseAC* intnoiseAC))
*echo             "Integreated_noise"
*echo             "number Points      $&n"
*echo             "Average level      $&averVal"
*echo             "RMS level          $&RmsVal"
let              sdIntAC[j]  = RmsVal
let              aveIntAC[j] = averVal
let              sdIntACraw[j]  = RmsRawVal
let              jj[j]  = j
let j =          j + 1
endrepeat
plot            sdraw sd ave vs jj
plot            sdIntraw sdInt aveInt endInt vs jj
plot            sdIntACraw sdIntAC aveIntAC vs jj
let sdaveraw    = sqrt(mean(sdraw* sdraw))
let aveave      = mean(ave)
unlet           noisave
let noisave     = ave - mean(ave)
let sdave       = sqrt(mean(noisave* noisave))
let avesd       = mean(sd)
```

```
unlet           noissd
let noissd      = sd - mean(sd)
let sdsd  =     sqrt(mean(noissd* noissd))
echo            "NumPoint  $&n  "
echo            "Average   $&aveave  +/- $&sdave  "
echo            "StanDev   $&avesd   +/- $&sdsd  "
echo            "StanDevRaw $&sdaveraw  "
let endraw  =   sqrt(mean(endInt* endInt))
let intsdaveraw = sqrt(mean(sdIntraw* sdIntraw))
let intaveave   = mean(aveInt)
unlet           intnoisave
let intnoisave  = aveInt - mean(aveInt)
let intsdave    = sqrt(mean(intnoisave* intnoisave))
let intavesd    = mean(sdInt)
unlet           intnoissd
let inrnoissd   = sdInt - mean(sdInt)
let intsdsd     = sqrt(mean(inrnoissd* inrnoissd))
echo            "NumPoint  $&n  "
echo            "IntAverage   $&intaveave  +/- $&intsdave  "
echo            "IntStanDev   $&intavesd   +/- $&intsdsd  "
echo            "intStanDevRaw $&intsdaveraw  "
echo            "EndPtRaw $&endraw  "
let raw2end  =   endraw/intsdaveraw
echo            "endptsd/raw_SD $&raw2end  "
let sdAve2end =  intsdave/endraw
echo            "AveSD/endsd $&sdAve2end  "
let intsdaverawAC = sqrt(mean(sdIntACraw* sdIntACraw))
let intaveaveAC   = mean(aveIntAC)
unlet             intnoisaveAC
let intnoisaveAC  = aveIntAC - mean(aveIntAC)
let intsdaveAC    = sqrt(mean(intnoisaveAC* intnoisaveAC))
let intavesdAC    = mean(sdIntAC)
unlet             intnoissdAC
let inrnoissdAC   = sdIntAC - mean(sdIntAC)
let intsdsdAC     = sqrt(mean(inrnoissdAC* inrnoissdAC))
echo            "NumPoint  $&n  "
echo            "IntAverageAC   $&intaveaveAC  +/- $&intsdaveAC  "
echo            "IntStanDevAC   $&intavesdAC   +/- $&intsdsdAC  "
echo            "intStanDevRawAC $&intsdaverawAC "
let ac2end =    intsdaverawAC/endraw
echo            "EndptAdjustSD/endsd $&ac2end  "

.endc
.end
```

4.4.11_11.24AM
dsauersanjose@aol.com
Don Sauer