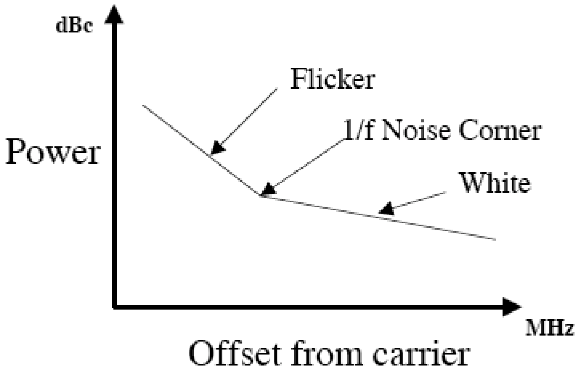
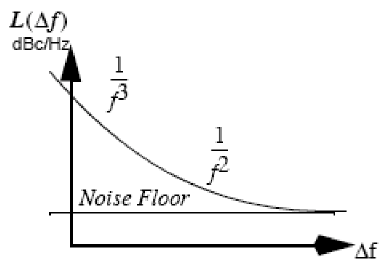
*=============**PREDICTING_OSCILLATOR_PHASE_NOISE**===========



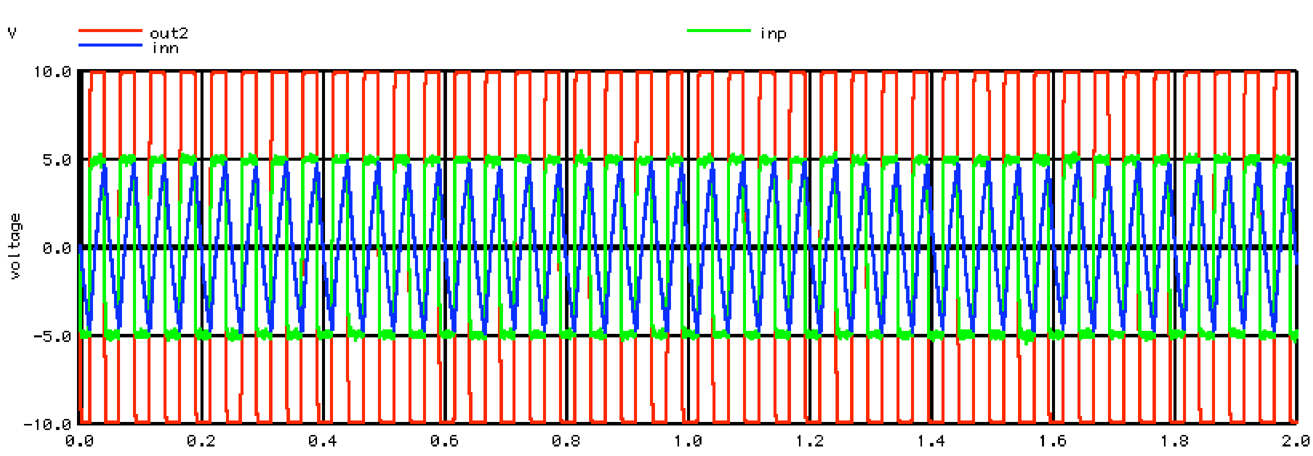


Oscillator phase noise is often described with the graphs above.

The noise floor of an oscillator is just the repackaging of signal to noise into the frequency and phase domain. A simple RC oscillator shown below can illustrate the point.

```
OSCILLATOR_Free_Running
*                R1  1K
*    _____/\  /\  /\__
*    |      \/  \/    _|_
* VFB|               ///
*    |___/\  /\  /\_____________
*          \/  \/          _|_        |
*    _|_         R2  1K    ___   C0   |
*   /   \  BPWL             |         |__|OUT|
*  //‾\  \                 _|_200uF   |  |___|
*  \   \//                 ///        |
*   \___/     INP  |\                 |
*     |____________|+\                |
*                  |  \_/\  /\  /\_   |
*      C1     INN  |  /    \/  \/     |
*  9.4uF  _________|-/      R0 1 Ohm  |
*      _|_    |    |/ BAMP            |
*                                     |
*      ___    |      R3  10K          |
*       |     |__/\  /\  /\___________|
*      _|_         \/  \/
*      ///
*
```

This oscillator is designed to produce a +/-10V 20Hz square wave. All noise is being modeled as a referred to input noise source to amplifier BAMP, and is being imported as a PWL noise file.

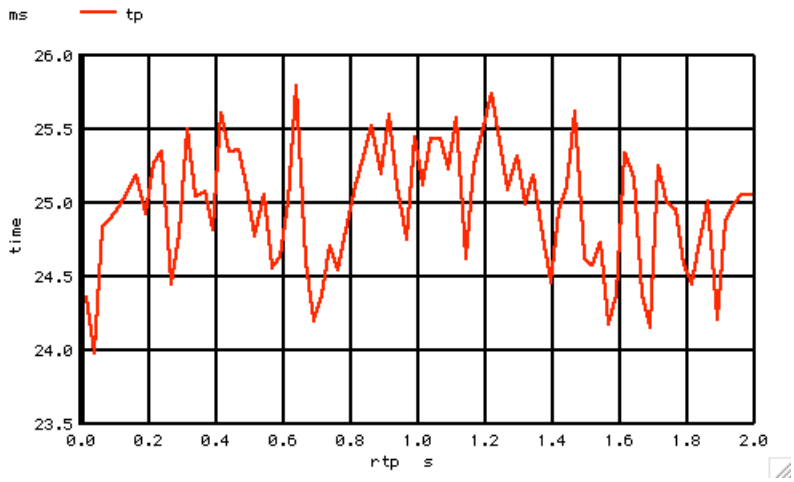


Capacitor C1 wants to swing +/- 5volts, and without the noise, the period to do so is set by R1 and C1. The noise source BPWL is going to jitter the voltage endpoints of C1. In this case the RMS value of BPWL is set to 100mV_rms. When C1 swings between +/- 5volts, it will see

two 100mV_rms endpoints. So the ratio of the randomness to the consistent timing for C1 will be.

randomness_tolerance = .141/10 = 1.414%

A jitter plot of the output wave form gives out about the same tolerance of the period between the edges.



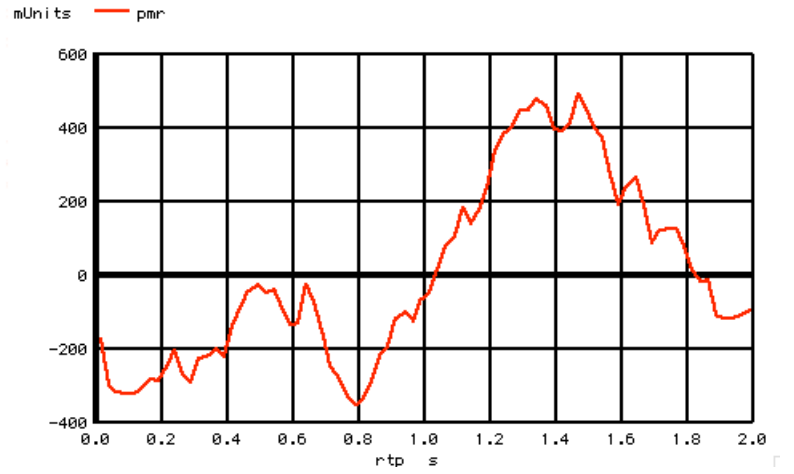
Edge2Edge_Period 0.024968 TPAC RMS SQUARE = 0.000404478

In this case rtp stands for time reference point. That is the time between when the transitions happen. The value tp stands for time period. This is the actual time between edges. Notice that the average time period is 25msec. A 20Hz square wave has two transitions within 50msec.

The ratio of the RMS value to the Average value which was simulated is....

TimePeriod_RMS/AVE = 0.0162 = 1.62%

But this modulation in time period is really just frequency modulation. To convert to phase modulation, one needs to integrate. The following shows the effective phase modulation in terms of radians.



Here pmr stands for phase modulation radians. When this waveform is used to phase modulate

a 20Hz square wave, it will produce almost the
same spectrum as the free running oscillator
which produced the jitter plot.

So how does one estimate the magnitude of the noise
sidebands at 19Hz or 21Hz? There are few rules which
enable this to be done.

RULE 1
Noise adds with the square root of the sum of the squares.

Capacitor C1 is seeing 100mV rms randomness at both
endpoints as it toggles +/- 5 volts. That is equivalent
to seeing 141mV of randomness compared to 10 volts.

RULE 2
Sampling Wide bandwidth Noise produces that same RMS magnitude regardless of the sample rate.

In this case a 100mV rms 1Khz noise signal gets sampled to
a 25msec rate. The result is a 100mV rms sample signal which
has a nyquist frequency of 20Hz. In other words, all the flat
band noise over a 1kHz bandwidth just got aliased into a
20Hz bandwidth.

RULE 3
The total sample time period sets the resolution by which one can know frequency.

In this case the total period is 2 seconds. So the simulation has
a frequency resolution of 0.5Hz.

RULE 4
Flat noise has the same amount of power in all frequency bands.

The 100mV rms noise got aliased into a 20Hz bandwidth. The fundamental
frequency bin resolution is 0.5 Hz. So there are 40 frequency bins.
And each bin will contain...

Noise_per_bin = 141mV/sqrt(40bins) = 22.3mV_(rms)

RULE 5
Randomness in voltage maps directly to randomness in timing.

C1_RMS_to_Ave = 22.3mV/10 = 0.223%_(rms) in time period per bin.

RULE 6
FM is converted to PM by integration.

A handy rule remember when translating FM to PM could be..

When Modulating any Carrier Frequency at 1Hz
1Hz FM produces 1radian PM

Since the noise is effecting the time period, it is doing FM
with white noise. If a 20Hz carrier gets modulated at 1Hz by
a 1Hz FM value, it will produce a 1radian PM. Note in this
case modulation is being done in magnitude format.

1Hz_(mag) FM on 20Hz = 1/20 = 5%_FM_(mag) {for 1rad PM}

All frequency bins produce 0.223%_(rms) tolerance in timing. A 5%_FM
signal at 1Hz will product 1 radian of PM. So a 0.223%_(rms) level

of FM at 1Hz will produce..

Expected_PM = 0.223%/5% = 0.0446 radians(rms) PM at 1Hz over 0.5Hz BW

RULE 7
A 1 radian PM signal comes close to looking like two 50% sidebands on a spectrum. (Actually, there is some rectangular to polar conversion error.)

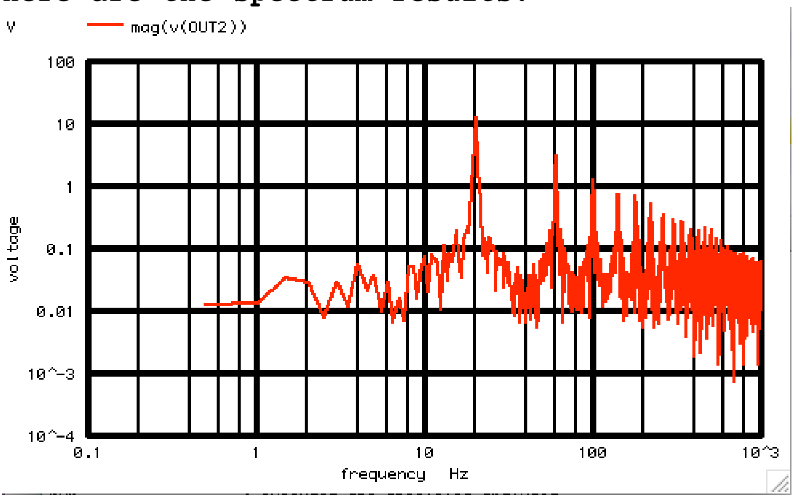
**The output square wave is +/- 10V. It will appears as a 14.14 level at the spectrum. One radian PM will be around the 7.07 volt level.
So 0.0446 radians(rms) of PM at 1Hz should have the following sideband magnitude.**

Expect_1Hz_SideBands_rms = 7.07*0.0446 = 315mV_rms

RULE 8
There is a 1.414 gain when converting from RMS to magnitude.

Sideband = 446mV_(mag)

Here are the spectrum results.



But its a little hard to display the spectrum to see the +/- 1Hz side bands to any detail.

A better way might be to FFT the PM jitter file. The PWL jitter file is modeling the noise floor of a free running oscillator as simply phase modulating a 20Hz square wave by a low frequency signal.

```
FFT_the_PWL_Jitter_File
*    OUT     Rload
*      ______/\  __        need to have a file called
*     _|_      \/  |       "rndsrc" in the "MacSpice"
*    /   \         |       folder inside "Doucments"
*  /VpwlT\         |
*  \     /        _|_      First typing in a MacSpice
*   \___/         ///      window "rndsrc .5m 1"
*     |           Gnd
*    _|_                   Then run this file
*    ///
*    Gnd
*
* VpwlT OUT 0         PWL( + 0.0005 0.988835 +.....
.include  PWL_FileJitter.inc
Rload     OUT    0      1k
*TRAN     TSTEP  TSTOP  TSTART TMAX      ?UIC?
.tran     1m     2      0      1m        UIC
.control
run
set       pensize =     2
plot      out
```

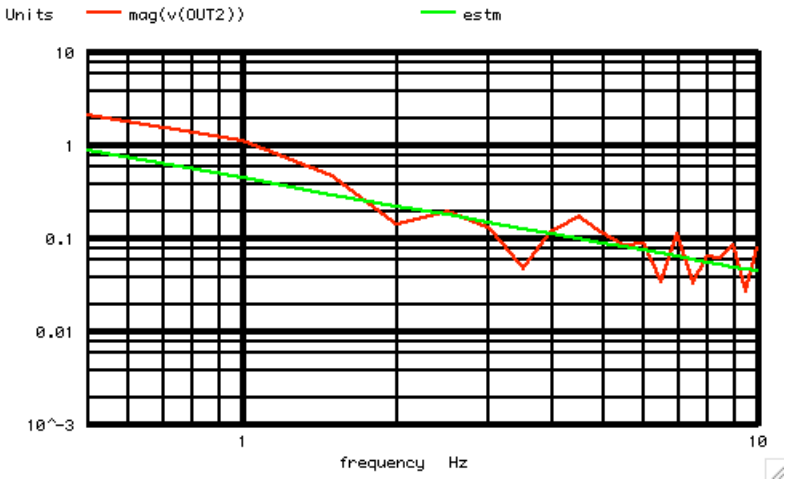
```
let       vmag = 14.14
let       radbbgain= .5
let       gain = vmag*radbbgain
let       out2 = gain*out
*=========================================================================
```

Now a little gain scaling is needed. One would expect a 1 radian signal to appear at a level of 7.07. Macspice is already doing the rms to magnitude conversion. The data in the PWL jitter file has already been formated into radians.

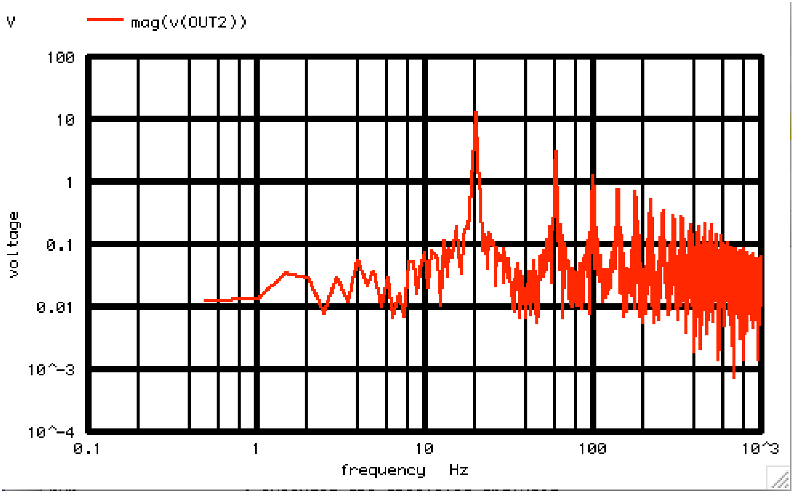
```
*=========Find_Spectrum=================================
linearize
set       specwindow =    "none"
*SPEC     FSTART  FSTOP    FSTEP   VECTOR
spec      .5       100     .5       v(OUT2)
*=======================================================
```

Why not create an estimated curve to show what the expected values would be over frequency? Use the expected 446mV value which was calculated.

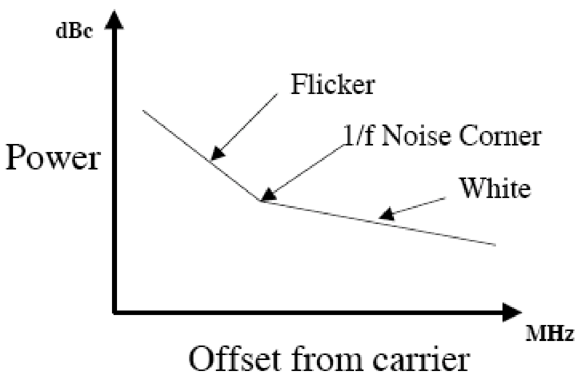
```
*=========Create_Estimate_Curve=========================
let       num =       length(out2)
compose   estm        start = 1 stop = $&num step =1
let       f = 1
let       i = 0
repeat    $&num
let       f = frequency[i]
let       estm[i]=.446/f
let i =   i +1
endrepeat
*=========Plot_Estimate_Curve===========================
plot      mag(v(OUT2)) estm   loglog xlimit .5 10
.endc
.end
*=======================================================
```

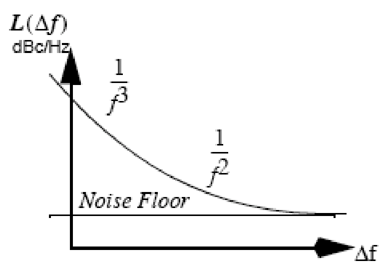


This is in the same scale as all the other spectrum plots. But it makes it easier to see what is happening at low frequencies



Now the effects of 1/f should be like modeling the low frequency noise as having some extra low frequency gain.





It is not obvious that all circuit tolerances can not be mapped to the frequency and phase domain.

```
===========Full_Netlist_For_Copy_Paste========================
FFT_the_PWL_Jitter_File
*    OUT      Rload
*       _____/\  __         need to have a file called
*     _|_      \/  |        "rndsrc" in the "MacSpice"
*    /   \         |        folder inside "Doucments"
*  /VpwlT\         |
*  \     /         |        First typing in a MacSpice
*   \___/         ///       window "rndsrc .5m 1"
*                 Gnd
*     _|_                   Then run this file
*     ///
*     Gnd
*
* VpwlT OUT 0        PWL( + 0.0005 0.988835 +.....
.include  PWL_FileJitter.inc
Rload     OUT    0     1k
*TRAN     TSTEP  TSTOP  TSTART TMAX      ?UIC?
.tran     1m     2      0      1m        UIC
.control
run
set       pensize =     2
plot      out
let       vmag = 14.14
let       radbbgain= .5
let       gain = vmag*radbbgain
let       out2 = gain*out
*=========Find_Spectrum===================================
linearize
set       specwindow =    "none"
*SPEC     FSTART  FSTOP    FSTEP    VECTOR
spec      .5       100     .5       v(OUT2)
```

```
*=========Create_Estimate_Curve=====================================
let       num =       length(out2)
compose   estm        start = 1 stop = $&num step =1
let       f = 1
let       i = 0
repeat    $&num
let       f = frequency[i]
let       estm[i]=.446/f
let i =   i +1
endrepeat
*=========Create_Estimate_Curve=====================================
plot      mag(v(OUT2)) estm   loglog xlimit .5 10
.endc
.end
```

2.18.10_12.15PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/