Here is a popular model for phase noise of an Oscillator. It does not claim that this model is predictive.

**Leeson’s model**

- For the LC oscillator
  \[ L(\Delta \omega) = 10 \log \left( \frac{2kT}{P_{\text{sig}}} \left( \frac{\omega_0}{2Q\Delta \omega} \right)^2 \right) \]

- Leeson’s model includes flicker noise
  \[ L(\Delta \omega) = 10 \log \left[ \frac{2FkT}{P_{\text{sig}}} \left( 1 + \frac{\omega_0}{2Q\Delta \omega} \right)^2 \left( 1 + \frac{\Delta \omega C}{(\Delta \omega)^3} \right) \right] \]

- \( F \) is an “effective noise figure” for the oscillator
- It is not a predictive model

Crystals are modeled as linear tuned circuits. It is not obvious that phase noise can not be derived from the spice models. There may be a simple way to predict phase noise using a graph and a calculator.

The following are some crystal tables and equations off the web. The values in light blue are extrapolated.

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>RS (Ohms)</th>
<th>LS (mH)</th>
<th>CS (pF)</th>
<th>CP (pF)</th>
<th>CR+CP/(CR+CP)</th>
<th>CP/CS</th>
<th>CR*CP/(CR+CP)</th>
<th>CP/CS</th>
<th>RP (Ohms)</th>
<th>Q</th>
<th>Z0 (Ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
<td>200</td>
<td>520</td>
<td>0.012</td>
<td>4</td>
<td>0.0119641</td>
<td>333</td>
<td>22.33Meg</td>
<td>33k</td>
<td>6582k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>50</td>
<td>115</td>
<td>0.010</td>
<td>3</td>
<td>0.0099668</td>
<td>300</td>
<td>4.53Meg</td>
<td>67k</td>
<td>3396k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>34.6</td>
<td>16.1</td>
<td>0.0093</td>
<td>4.5</td>
<td>0.0092808</td>
<td>483.87</td>
<td>8.13Meg</td>
<td>38k</td>
<td>1315k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td>30</td>
<td>12.5</td>
<td>0.009</td>
<td>5</td>
<td>0.0089838</td>
<td>555.55</td>
<td>9.29Meg</td>
<td>39k</td>
<td>1178k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.00</td>
<td>20</td>
<td>4.7</td>
<td>0.006</td>
<td>3</td>
<td>0.005988</td>
<td>500</td>
<td>5.02Meg</td>
<td>44k</td>
<td>885k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.00</td>
<td>11</td>
<td>1.2</td>
<td>0.002</td>
<td>1.5</td>
<td>0.001998</td>
<td>750</td>
<td>6.18Meg</td>
<td>70k</td>
<td>775k</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- \( F_{\text{series}} = \frac{1}{2\pi \sqrt{\text{CS} \times \text{LS}}} \)
- \( C_\text{loop} = \frac{\text{CR} \times \text{CP}}{\text{CR} + \text{CP}} \)
- \( F_{\text{parallel}} = \frac{1}{2\pi \sqrt{\text{C}_\text{loop} \times \text{LS}}} \)
- \( R_P = \frac{\text{RS} \times (1 + \text{CP} / \text{CR})^2}{2} \)
- \( Q = \frac{2\pi \times F_{\text{series}} \times \text{LS} / \text{RS}}{\text{CP} / \text{CS}} \)

Look at the Phase Noise Plot of a real 13MHz Crystal Oscillator.
Here is the spice model for just the crystal.

```
 1 --\ |  \ --\ |  \\
\ |  \ |  \ |  \ |  \ |  \ |  \ 1
\ |  \ |  \ |  \ |  \ |  \ |  \ 2
\ |  \ |  \ |  \ |  \ |  \ |  \  |
RS 34.6  CS  0.0093pF  LS  16.1m
\ |  \ |  \ |  \ |  \ |  \ |  \  |
\ |  \ |  \ |  \ |  \ |  \ |  \  |
CP  4.5pF
```

Freq = 13meg
Z0 = 8.13meg
Q = 38K
CLoop = 0.009281pF

Add a horizontal Phase Noise line which describes the flat band phase noise. Now add a vertical line at 13MegHz to show the oscillation frequency.

Now added a 20db/dec slope curve such that it is tangent to the Actual Phase Noise curve. That slope will be called the Integrated or Accumulated Phase Noise level. In an oscillator, the phase noise of each cycle gets accumulated over time. It crosses 13MHz at -229 dB which will be called the crossing level.

When looking at the Actual Phase Noise curve at 10Hz, one cannot detect something happening at 10Hz with out taking 100ms worth of data.

In other words, 1.3 million clock cycles will be needed to measure something at 10Hz. The pre-accumulated phase noise of each cycle will accumulate over that 100msec period. It turns out rms power increases by the square root of the number of clock cycles. So think of Phase noise in terms of the accumulation that is taking place.

Filter the thermal noise of RS by the bandwidth of the Crystal.
Assume the crystal is running at 1Vrms which is 0dB. Only half of this filtered RMS voltage is going into Phase Modulation. The resulting rms voltage is then being sampled at 13MHz Nyquist. The resulting rms voltage will be spread uniformly over Nyquist at 13MHz. This produces a pre-accumulated PM level at -231 dB.
Notice it is almost the same as the crossing level.

When a pre-accumulated PM noise level get accumulated, it will equal the accumulation PM level when looking over a period of one clock cycle. No accumulations will be taking place.
This is the jitter one sees when viewing and synching one cycle on a scope.

So the non-accumulated phase noise due to the RS thermal noise should match the accumulated phase Noise when crossing 13MHz.

This example is assuming the crystal is seeing 1Vrms across it. This oscillator runs at 3.3V. That comes close to a 2.82Vppk swing across the crystal.
So given the model of the tuned circuit and the voltage across the tuned circuit, an integrated or accumulated line having a 20dB per decade slope can be drawn from where the clock frequency crosses the filtered and sampled pre-accumulated thermal noise level of the tuned circuit.

The drawn integrated or accumulated line describes the limitations of the crystal. The flat phase noise level will be present due to things outside the tuned circuit. A 1KOhm resistor has 4nV per root hertz noise. Transistors produce almost the same noise for a given impedance. It takes some supply current to get transistors to operate at low impedances. For instance, a bipolar transistor needs to draw 1mA to have the noise of 13 Ohms.

Above the 1/f point, accumulated pm noise and supply current might defined Phase Noise. Bipolar transistor have much lower 1/f noise than CMOS transistors. The 1/f point can be lowered further using some larger transistors.

---

### MacSpiceCode

```macspice
Phase_Noise_13MHz_Crystal

*Create_Signal_No_Reason*

*V_SIN# NODE_P NODE_N DC VALUE SIN(   V_DC   AC_MAG FREQ   DELAY  FDamp)
```

---
VIN VP 0 DC 0 SIN( 0 1 1 )

.control
set pensize = 2
.unlet stanDev_val
.unlet Out_percent
.unlet X13
.unlet freq
.unlet intnoise

let X13 = vector(8)
let freq = vector(8)
let intnoise = vector(8)

let offset = -87

let freq[0] = 10
let X13[0] = -100
let freq[1] = 100
let X13[1] = -127
let freq[2] = 1k
let X13[2] = -141
let freq[3] = 10k
let X13[3] = -147
let freq[4] = 100k
let X13[4] = -150
let freq[5] = 1Meg
let X13[5] = -150
let freq[6] = 10Meg
let X13[6] = -150
let freq[7] = 13Meg
let X13[7] = -150

echo 
"========================Create_Timing_Slope=============="
let index = 0
repeat 8
let intnoise[index] = -20*log(freq[index]) + offset
let index = index + 1
end

echo 
"========================Osillator_Freq_and_Period=============="
let oscFreq = 13meg
echo "OscillatorFreq_Hz = $&oscFreq"
let period = 1/oscFreq
echo "Oscillator_Period_s = $&period"

let oscVppk = 2.82
let oscVrms = 1
let osc_db = db(oscVppk)
echo "Osc_V_ppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_db"
echo "Osc_Vppk = $&oscVppk"
echo "Osc_V_rms = $&oscVrms"
echo "Osc_db = $&osc_d