Frequency Difference Divider

T0900138-v1
March 27, 2009
Daniel Sigg

Theory of Operations

A frequency difference divider takes an input signal divides it down and adds it to a fixed frequency low noise reference. If the input signal experiences a high phase noise, the phase noise in the output signal is reduced by the divider ratio---assuming all other noise sources are negligible.

Specifications

TBD.

Setup

```
Needs["BarCharts`"]; Needs["Histograms`"]; Needs["PieCharts`"];
Needs["ErrorBarPlots`"]; Needs["PlotLegends`"];
Needs["Controls`LinearControl`"]

$TextStyle = {FontFamily → "Helvetica", FontSize → 13};

plotopt = PlotStyle → {{Thickness [0.007], RGBColor [1, 0, 0]},
              {Thickness [0.007], RGBColor [0, 0, 1]},
              {Thickness [0.007], RGBColor [0.1, 0.7, 0.2]},
              {Thickness [0.007], RGBColor [0.5, 0.5, 0.2]}};
```

Common Definitions

```
FromdBc[x_] := 10^(-x/20)

NoiseLevelBottom[f_] := FromdBc[175]
```

Model

This model takes VCO (Synergy MCF91119-10) and sends it through a frequency difference divider. The VCO runs at 1.05 GHz and has a range of ±125 MHz. The VCO signal is amplified first to 13dBm, then dvided by 8 to give a frequency of 131 MHz ± 16 MHz. The target frequency is 80 MHz, so we can drive a standard acousto-optical modulator. The divided VCO signal is then sent to the frequency difference divider. The frequency difference divider will divide the signal by a further 16, amplifier it and send it to the RF input of a mixer. The LO input of the mixer is driven by the amplified output of a OCXO which runs at 71 MHz. The IF output of the mixer is sent through a filter to reject the lower image and then amplified to yield 13 dBm outputs at a nominal 80 MHz ± 1 MHz. To lower the noise even further this output signal can be sent to a second frequency difference divider. This time the divider ratio is 10. This yields a nominal 80 MHz ± 0.1 MHz.

The result plot shows a comparison of the phase noise of the raw VCO, the VCO scaled to 80 MHz by direct division by 13, the output of the first stage frequency difference divider and the output of the second stage frequency difference divider. As expected the scaled VCO phase noise is about 20 dB better than the raw VCO phase noise. The first stage output is another 20 dB lower and the output of the second stage reproduces the phase noise of the OCXO at lower frequencies. At higher frequencies beyond 20 kHz the noise level is about 12 dB higher due to the dividers, amplifiers and mixers.

Frequency Range

f0 : OCXO frequency
freq: Synergy MCF91119-10 frequency range
vcofreq[n]: VCO output frequency after n-th stage

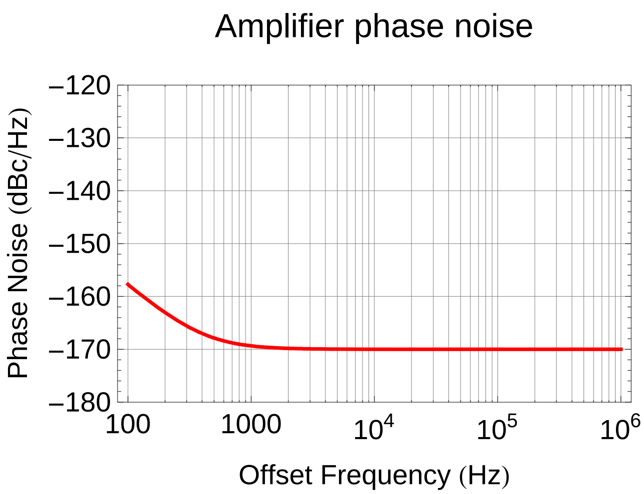
```
f0 = 71*^6;
freq = {930*^6, 1050*^6, 1240*^6};
vcofreq[1] := freq/(8 × 16.) + f0
vcofreq[n_] := vcofreq[n - 1]/10. + f0
freq 1*^-6
vcofreq[1] 1*^-6
vcofreq[2] 1*^-6
vcofreq[3] 1*^-6
vcofreq[100] 1*^-6
```

```
{930, 1050, 1240}

{78.2656, 79.2031, 80.6875}

{78.8266, 78.9203, 79.0688}

{78.8827, 78.892, 78.9069}

{78.8889, 78.8889, 78.8889}
```

Generic Amplifier Noise

```
AMPNoiseLow[f_] := FromdBc[118 + 20 Log[10, f]]
AMPNoiseLevel[f_] := FromdBc[170]
AMPNoise[f_] := Sqrt[AMPNoiseLow[f]^2 + AMPNoiseLevel[f]^2]
```

```
LogLinearPlot[dB[AMPNoise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle,
 PlotRange → {-180, -120}, PlotLabel → "Amplifier phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



VCO (MCF91119-10) at 1.05 GHz

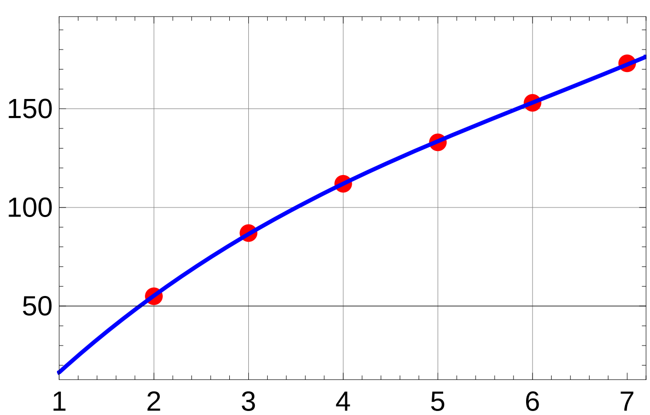
```
VCOdata = {{2, 55}, {3, 87}, {4, 112}, {5, 133}, {6, 153}, {7, 173}, {8, 193}}
```

```
{{2, 55}, {3, 87}, {4, 112}, {5, 133}, {6, 153}, {7, 173}, {8, 193}}
```

```
VCOfit[x_] = Fit[VCOdata, {1, x, x^2, x^3}, x]
```

$-31.7857 + 53.5278\, x - 5.61905\, x^2 + 0.305556\, x^3$

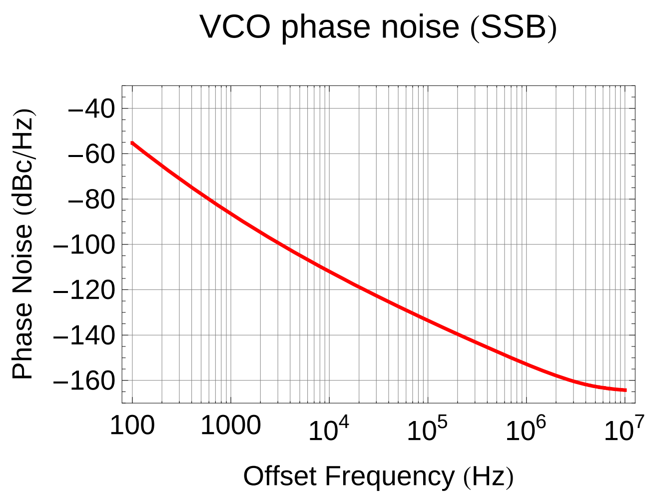
```
Show[ListPlot[VCOdata, PlotStyle → {Red, PointSize[0.03]}, Frame → True,
  GridLines → Automatic, PlotRange → {{1, 7.2}, All}, BaseStyle → $TextStyle], Plot[VCOfit[x],
  {x, 1, 7.2}, PlotRange → {{1, 7.2}, All}, PlotStyle → {Thickness[0.007], RGBColor[0, 0, 1]}]]
```



```
VCONoiseLow[f_] := FromdBc[VCOfit[Log[10, f]]]
VCONoiseLevel[f_] := FromdBc[165]
```

$$\text{VCONoise}[f\_] := \sqrt{\text{VCONoiseLow}[f]^2 + \text{VCONoiseLevel}[f]^2}$$

```
LogLinearPlot[dB[VCONoise[f]], {f, 100, 1*^7}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle, PlotRange → {-170, -30},
 PlotLabel → "VCO phase noise (SSB)",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



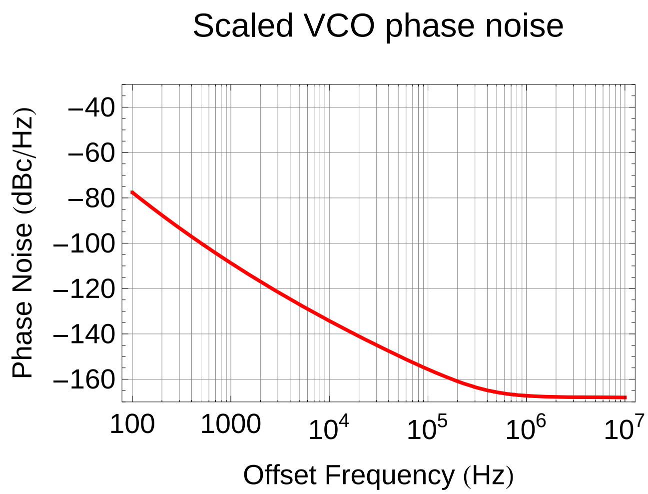
VCO scaled to 80 MHz

$$\text{VCOScaledNoiseLow}[f\_] := \frac{\sqrt{\text{VCONoiseLow}[f]^2 + \text{AMPNoiseLow}[f]^2}}{13}$$

$$\text{VCOScaledNoiseLevel}[f\_] := \sqrt{\frac{\text{AMPNoiseLevel}[f]^2 + \text{VCONoiseLevel}[f]^2}{13} + 4\,\text{NoiseLevelBottom}[f]^2}$$

$$\text{VCOScaledNoise}[f\_] := \sqrt{\text{VCOScaledNoiseLow}[f]^2 + \text{VCOScaledNoiseLevel}[f]^2}$$

```
LogLinearPlot[dB[VCOScaledNoise[f]], {f, 100, 1*^7}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle, PlotRange → {-170, -30},
 PlotLabel → "Scaled VCO phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



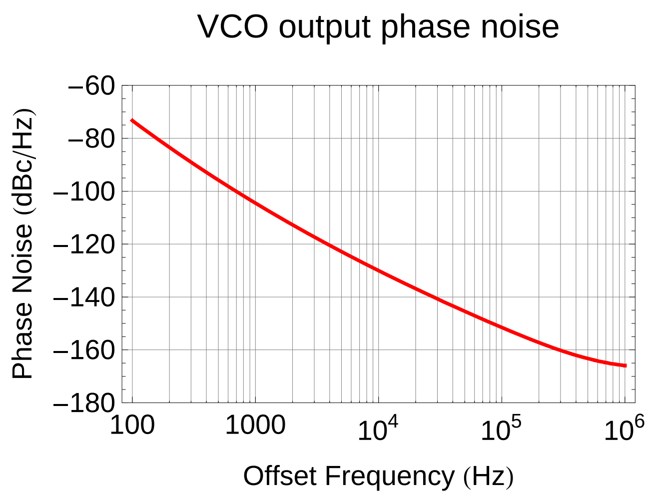
VCO Output: 11dB amplification & Divide-by-8

$$\text{VCO8NoiseLow[f\_] := }\frac{\sqrt{\text{VCONoiseLow[f]}^2 + \text{AMPNoiseLow[f]}^2}}{8}$$

$$\text{VCO8NoiseLevel[f\_] := }\sqrt{\frac{\text{AMPNoiseLevel[f]}^2 + \text{VCONoiseLevel[f]}^2}{8} + 4\,\text{NoiseLevelBottom[f]}^2}$$

$$\text{VCO8Noise[f\_] := }\sqrt{\text{VCO8NoiseLow[f]}^2 + \text{VCO8NoiseLevel[f]}^2}$$

```
LogLinearPlot[dB[VCO8Noise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle,
 PlotRange → {-180, -60}, PlotLabel → "VCO output phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



Mixer LO: Divide-by-16 & 7dB amplification

```
Clear[LONoiseLow, LONoiseLevel, LONoise];
```

$$\text{LONoiseLow[f\_, input\_, ratio\_]} := \sqrt{\left(\frac{\text{input[f]}}{\text{ratio}}\right)^2 + \text{AMPNoiseLow[f]}^2}$$

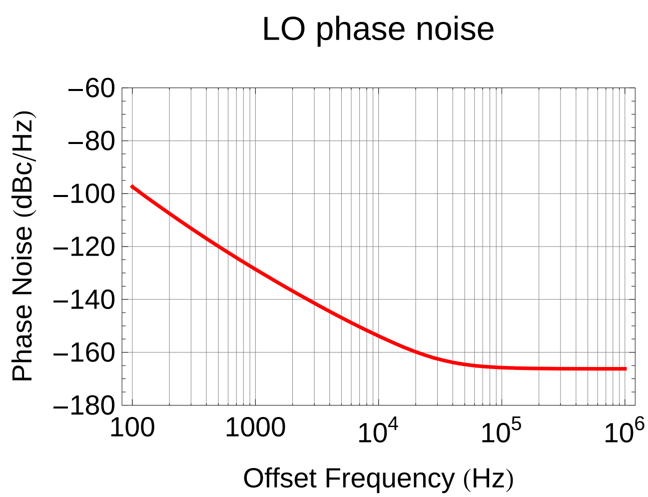
$$\text{LONoiseLevel[f\_, input\_, ratio\_]} := \sqrt{\frac{\text{input[f]}^2}{\text{ratio}} + 4\,\text{NoiseLevelBottom[f]}^2 + \text{AMPNoiseLevel[f]}^2}$$

$$\text{LONoise[f\_, in1\_, in2\_, r\_]} := \sqrt{\text{LONoiseLow[f, in1, r]}^2 + \text{LONoiseLevel[f, in2, r]}^2}$$

```
LONoiseLow[f_] := LONoiseLow[f, VCO8NoiseLow, 16]
LONoiseLevel[f_] := LONoiseLevel[f, VCO8NoiseLevel, 16]
```

$$\text{LONoise[f\_]} := \sqrt{\text{LONoiseLow[f]}^2 + \text{LONoiseLevel[f]}^2}$$

```
LogLinearPlot[dB[LONoise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red},
 BaseStyle → $TextStyle, PlotRange → {-180, -60}, PlotLabel → "LO phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



Mixer RF: Wenzel OCXO at 71 MHz & 10dB amplification

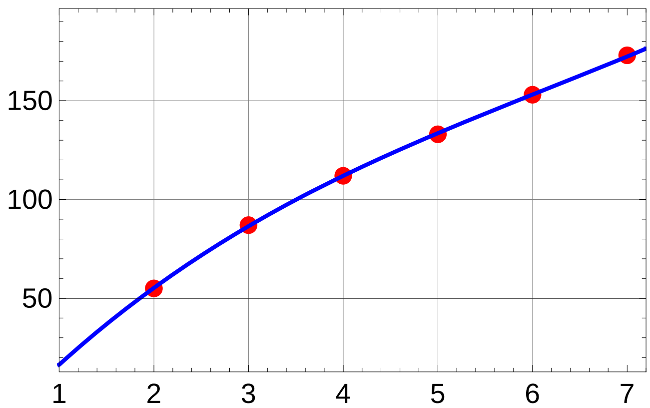
```
OCXOdata = {{1, 90}, {2, 110}, {3, 140}, {4, 160}, {5, 180}, {6, 200}, {7, 220}}
```

```
{{1, 90}, {2, 110}, {3, 140}, {4, 160}, {5, 180}, {6, 200}, {7, 220}}
```

$$\text{OCXOfit[x\_]} = \text{Fit}\left[\text{OCXOdata}, \{1, x, x^2, x^3\}, x\right]$$

$$62.8571 + 26.5476\,x - 0.595238\,x^2 - 6.34391 \times 10^{-15}\,x^3$$

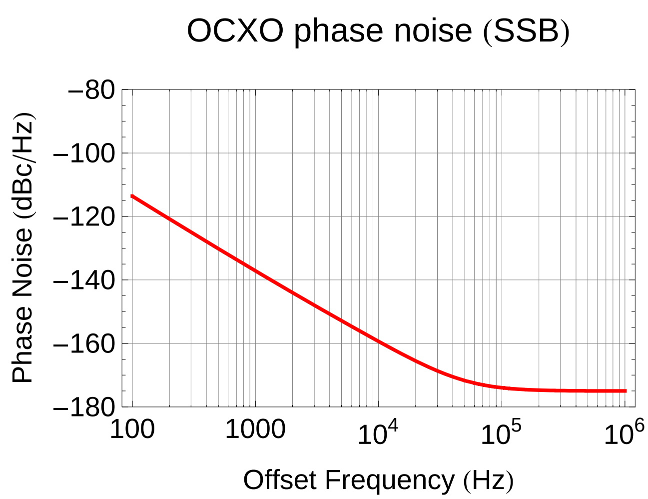
```
Show[ListPlot[VCOdata, PlotStyle → {Red, PointSize[0.03]}, Frame → True,
  GridLines → Automatic, PlotRange → {{1, 7.2}, All}, BaseStyle → $TextStyle], Plot[VCOfit[x],
  {x, 1, 7.2}, PlotRange → {{1, 7.2}, All}, PlotStyle → {Thickness[0.007], RGBColor[0, 0, 1]}]]
```




```
OCXONoiseLow[f_] := FromdBc[OCXOfit[Log[10, f]]]
OCXONoiseLevel[f_] := FromdBc[175]
```

OCXONoise[f_] := $\sqrt{\text{OCXONoiseLow[f]}^2 + \text{OCXONoiseLevel[f]}^2}$

```
LogLinearPlot[dB[OCXONoise[f]], {f, 100, 1*^6}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle, PlotRange → {-180, -80},
 PlotLabel → "OCXO phase noise (SSB)",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```

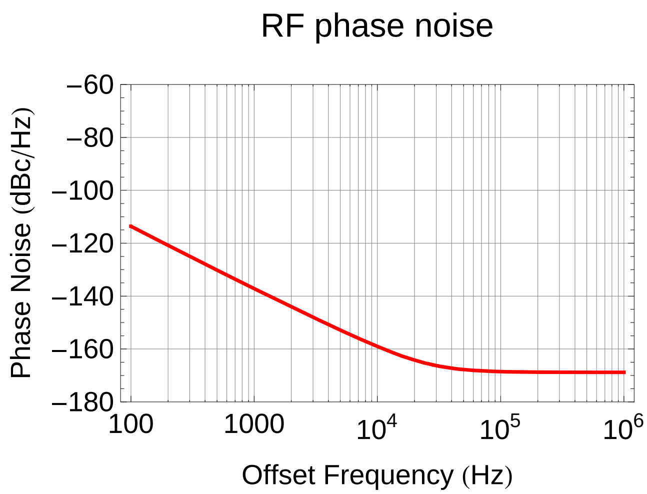


RFNoiseLow[f_] := $\sqrt{\text{OCXONoiseLow[f]}^2 + \text{AMPNoiseLow[f]}^2}$

RFNoiseLevel[f_] := $\sqrt{\text{OCXONoiseLevel[f]}^2 + \text{AMPNoiseLevel[f]}^2}$

RFNoise[f_] := $\sqrt{\text{RFNoiseLow[f]}^2 + \text{RFNoiseLevel[f]}^2}$

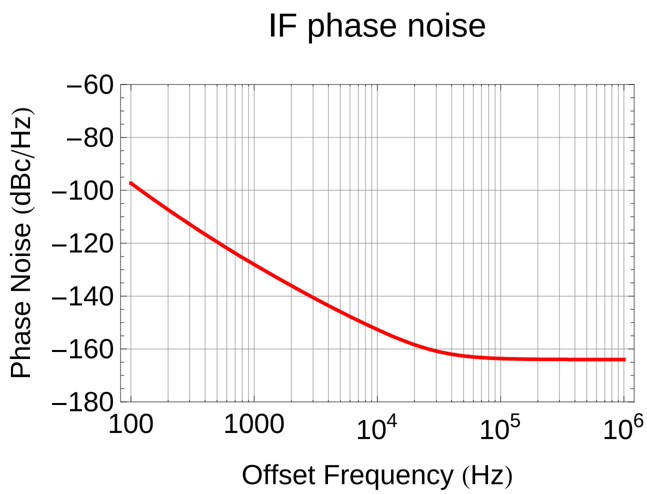
```
LogLinearPlot[dB[RFNoise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red},
 BaseStyle → $TextStyle, PlotRange → {-180, -60}, PlotLabel → "RF phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



Mixer IF: Sum of LO and RF

```
IFNoiseLow[f_, in_, r_] := Sqrt[LONoiseLow[f, in, r]^2 + RFNoiseLow[f]^2]
IFNoiseLevel[f_, in_, r_] :=
 Sqrt[LONoiseLevel[f, in, r]^2 + RFNoiseLevel[f]^2 + NoiseLevelBottom[f]^2]
IFNoise[f_, in1_, in2_, r_] := Sqrt[IFNoiseLow[f, in1, r]^2 + IFNoiseLevel[f, in2, r]^2]

IFNoiseLow[f_] := Sqrt[LONoiseLow[f]^2 + RFNoiseLow[f]^2]
IFNoiseLevel[f_] := Sqrt[LONoiseLevel[f]^2 + RFNoiseLevel[f]^2 + NoiseLevelBottom[f]^2]
IFNoise[f_] := Sqrt[IFNoiseLow[f]^2 + IFNoiseLevel[f]^2]
```

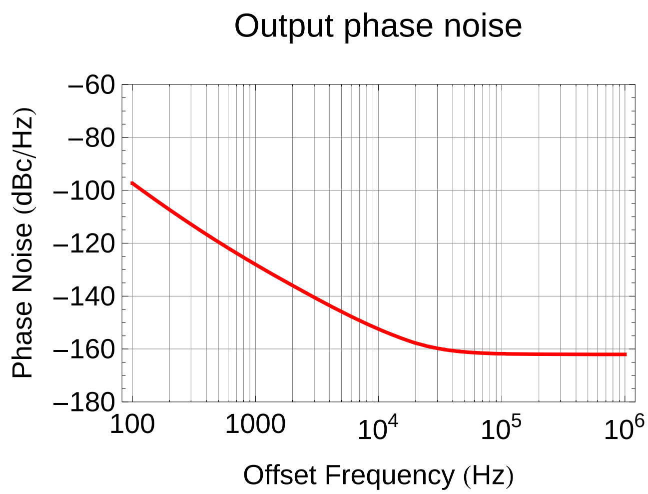
```
LogLinearPlot[dB[IFNoise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red},
 BaseStyle → $TextStyle, PlotRange → {-180, -60}, PlotLabel → "IF phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



Output: Mixer IF & filter & 12 dB attenuation & 2-way splitter

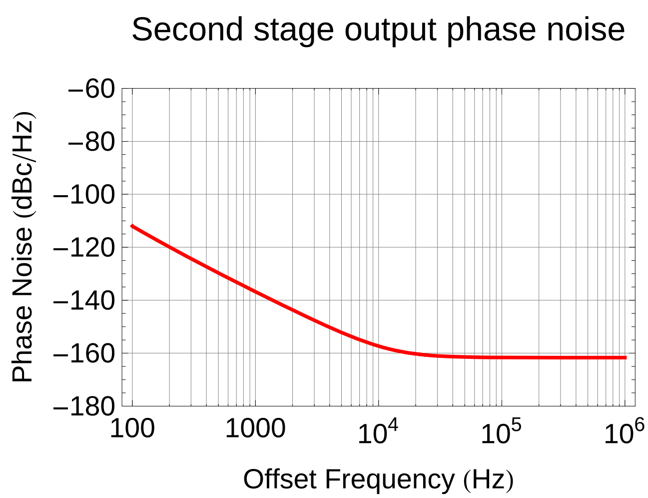
```
OutputNoiseLow[f_, in_, r_] := Sqrt[IFNoiseLow[f, in, r]^2 + AMPNoiseLow[f]^2]
OutputNoiseLevel[f_, in_, r_] :=
 Sqrt[IFNoiseLevel[f, in, r]^2 + AMPNoiseLevel[f]^2 + 4 NoiseLevelBottom[f]^2]
OutputNoise[f_, in1_, in2_, r_] := Sqrt[OutputNoiseLow[f, in1, r]^2 + OutputNoiseLevel[f, in2, r]^2]

OutputNoiseLow[f_] := Sqrt[IFNoiseLow[f]^2 + AMPNoiseLow[f]^2]
OutputNoiseLevel[f_] := Sqrt[IFNoiseLevel[f]^2 + AMPNoiseLevel[f]^2 + 4 NoiseLevelBottom[f]^2]
OutputNoise[f_] := Sqrt[OutputNoiseLow[f]^2 + OutputNoiseLevel[f]^2]
```

```
LogLinearPlot[dB[OutputNoise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red},
 BaseStyle → $TextStyle, PlotRange → {-180, -60}, PlotLabel → "Output phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```



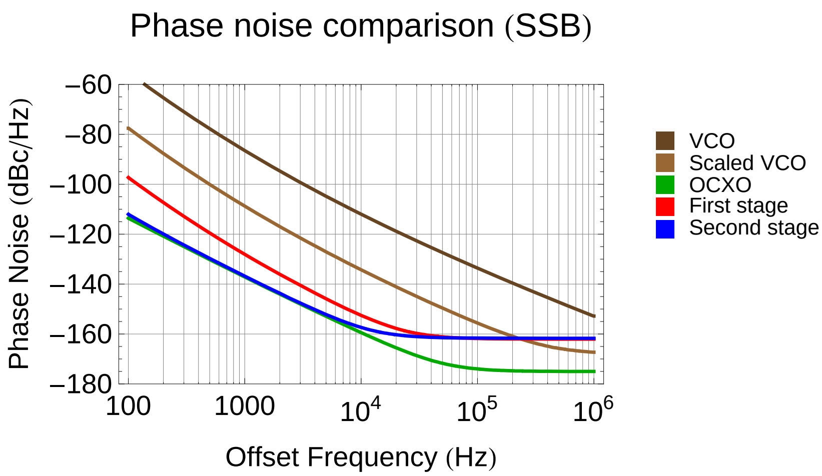
Output after second frequency difference divider

```
Output2NoiseLow[f_] := √(OutputNoiseLow[f, OutputNoiseLow, 10]^2)
Output2NoiseLevel[f_] := √(OutputNoiseLevel[f, OutputNoiseLevel, 10]^2)
Output2Noise[f_] := √(Output2NoiseLow[f]^2 + Output2NoiseLevel[f]^2)
LogLinearPlot[dB[Output2Noise[f]], {f, 100, 1*^6}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle,
 PlotRange → {-180, -60}, PlotLabel → "Second stage output phase noise",
 FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}]
```

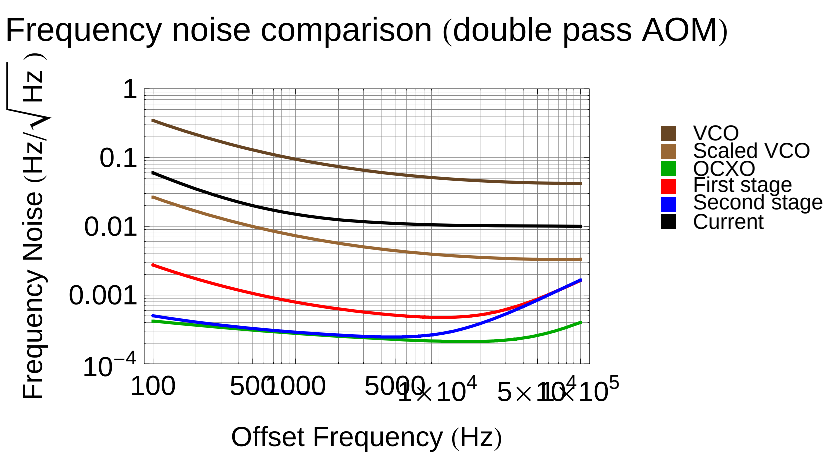


Comparison

```
ShowLegend[LogLinearPlot[{dB[VCONoise[f]], dB[VCOScaledNoise[f]], dB[OCXONoise[f]],
    dB[OutputNoise[f]], dB[Output2Noise[f]]}, {f, 100, 1*^6}, Frame → True,
   GridLines → Automatic, PlotStyle → {{Thickness [0.007], Darker[Brown]},
     {Thickness [0.007], Brown}, {Thickness [0.007], Darker[Green]},
     {Thickness [0.007], Red}, {Thickness [0.007], Blue}}, BaseStyle → $TextStyle,
   PlotRange → {-180, -60}, PlotLabel → "Phase noise comparison (SSB)",
   FrameLabel → {"Offset Frequency (Hz)", "Phase Noise (dBc/Hz)"}, ImageSize → Scaled[0.85]],
  {{{Graphics[{Darker[Brown], Rectangle[{1, 1}]}], "VCO"},
    {Graphics[{Brown, Rectangle[{1, 1}]}], "Scaled VCO"},
    {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "OCXO"},
    {Graphics[{Red, Rectangle[{1, 1}]}], "First stage"},
    {Graphics[{Blue, Rectangle[{1, 1}]}], "Second stage"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.7, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



```
ShowLegend[LogLogPlot[{2 f VCONoise[f], 2 f VCOScaledNoise[f],
    2 f OCXONoise[f], 2 f OutputNoise[f], 2 f Output2Noise[f], 50*^-3 100/f + 10*^-3},
   {f, 100, 1*^5}, Frame → True, GridLines → Automatic,
   PlotStyle → {{Thickness[0.007], Darker[Brown]}, {Thickness[0.007], Brown},
     {Thickness[0.007], Darker[Green]}, {Thickness[0.007], Red},
     {Thickness[0.007], Blue}, {Thickness[0.007], Black}}, BaseStyle → $TextStyle,
   PlotRange → {1*^-4, 1}, PlotLabel → "Frequency noise comparison (double pass AOM)",
   FrameLabel → {"Offset Frequency (Hz)", "Frequency Noise (Hz/√Hz)"},
   ImageSize → Scaled[0.85]], {{{Graphics[{Darker[Brown], Rectangle[{1, 1}]}], "VCO"},
    {Graphics[{Brown, Rectangle[{1, 1}]}], "Scaled VCO"},
    {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "OCXO"},
    {Graphics[{Red, Rectangle[{1, 1}]}], "First stage"},
    {Graphics[{Blue, Rectangle[{1, 1}]}], "Second stage"},
    {Graphics[{Black, Rectangle[{1, 1}]}], "Current"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.8, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Table

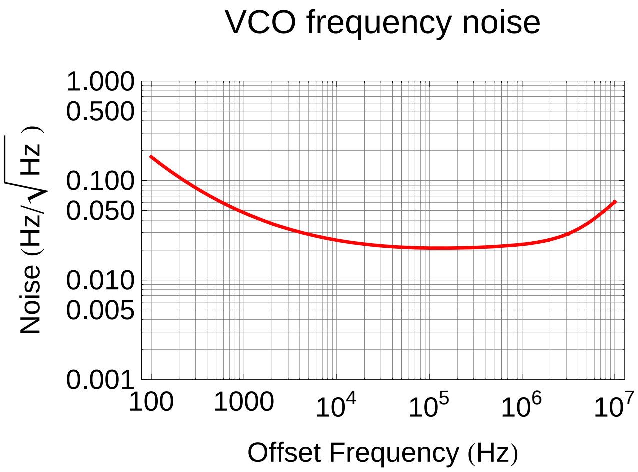
```
Join[{{"Frequency", "1st stage", "2nd stage"}},
  {#, dB[OutputNoise[#]], dB[Output2Noise[#]]} & /@ {10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	1st stage	2nd stage
10	-58.5687	-78.1761
100	-97.2791	-112.03
1000	-128.047	-136.789
10 000	-152.506	-157.34
100 000	-161.787	-161.625

Input Referred Noise

VCO Frequency Noise

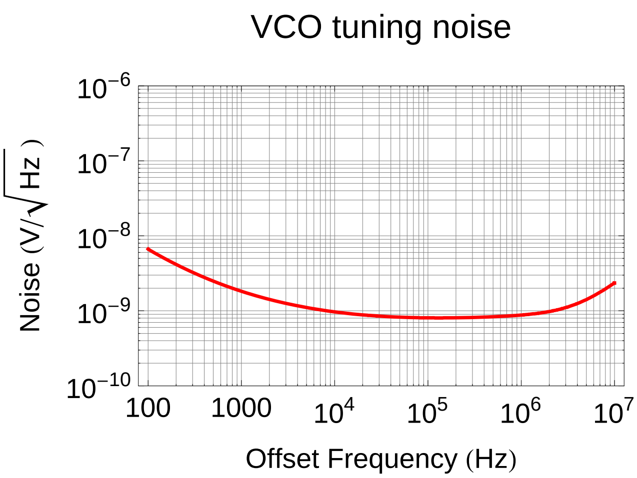
```
LogLogPlot[f VCONoise[f], {f, 100, 1*^7}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle, PlotRange → {0.001, 1},
 PlotLabel → "VCO frequency noise",
 FrameLabel → {"Offset Frequency (Hz)", "Noise (Hz/√Hz)"}]
```



Input Referred Voltage Noise

```
tuneSense = 26*^6; (* Hz/V *)
```

```
LogLogPlot[f VCONoise[f] / tuneSense, {f, 100, 1*^7}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle, PlotRange → {1*^-10, 1*^-6},
 PlotLabel → "VCO tuning noise", FrameLabel → {"Offset Frequency (Hz)", "Noise (V/√Hz)"}]
```



Pre-Amplification Noise

Parallel and Serial Impedance

```
par[r1_, r2_] := 1/(1/r1 + 1/r2)
ser[r1_, r2_] := r1 + r2
```

Transfer Function of an OpAmp

This function computes the transfer function of an idealized OpAmp circuit
g: +1 for non-inverting configuration or -1 for inverting configuration, 0 for differential configuration
z2: Impedance in feedback path [Ohm]
z1: Impedance of input path (inverting) or impedance to ground (non-inverting) [Ohm]

```
OpAmp[g_, z1_, z2_] :=
 Which[g > 0, 1 + z2/z1, g < 0, z2/z1, True, z2/z1]
```

Noise of an OpAmp

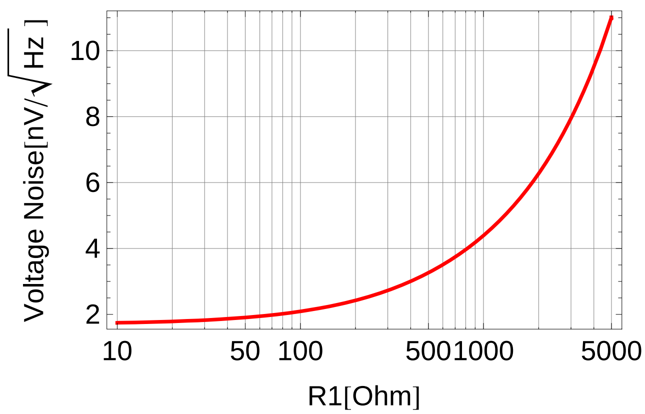
This function computes the equivalent input noise of an OpAmp circuit
g: +1 for non-inverting configuration or -1 for inverting configuration, 0 for differential configuration
z1: Impedance of input path (inverting) or impedance to ground (non-inverting) [Ohm]
z2: Impedance over feedback path [Ohm]
en: voltage noise [Volt]
in: current noise [Ampere]

```
FourKT = 1.62*^-20;(* V^2/Hz/Ohm; room temperature 20C*)
OpAmpNoise[g_, z1_, z2_, en_, in_] :=
 Which[g > 0, If[z1 == Infinity, Sqrt[en^2 + FourKT Abs[z2] + (in Abs[z2])^2],
    Sqrt[en^2 + FourKT Abs[par[z1, z2]] + (in Abs[par[z1, z2]])^2]],
  g < 0, Sqrt[(Abs[1 + z1/z2]^2 en^2 + Abs[z1]^2 (in^2 + Abs[FourKT/z1] + Abs[FourKT/z2]))],
  True, Sqrt[(Abs[1 + z1/z2]^2 en^2 + 2 Abs[z1]^2 (in^2 + Abs[FourKT/z1] + Abs[FourKT/z2]))]]
```

Examples (AD829)

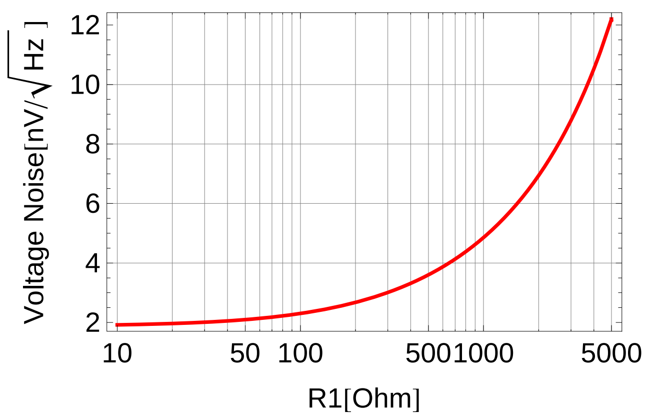
Non-Inverting configuration: input noise w/ gain of 10 as function of r1

```
LogLinearPlot[1*^9 OpAmpNoise[+1, r, 9 r, 1.7*^-9, 1.5*^-12], {r, 10, 5000},
 FrameLabel → {"R1[Ohm]", "Voltage Noise[nV/√Hz ]"}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle]
```



Inverting configuration: input noise w/ gain of 10 as function of r1

```
LogLinearPlot[1*^9 OpAmpNoise[-1, r, 10 r, 1.7*^-9, 1.5*^-12], {r, 10, 5000},
 FrameLabel → {"R1[Ohm]", "Voltage Noise[nV/√Hz ]"}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle]
```



Series Product of OpAmps

Computes the transfer function of several OpAmps circuits in series.

```
OpAmpProduct[t_, m_] := Product[t[i], {i, m}]
```

Computes the equivalent input noise of several OpAmps circuits in series.

```
NoiseSum[prev_, {t_, n_}] := √(prev^2 + n^2) Abs[t]
OpAmpNoiseProduct[t_, n_, m_] := Fold[NoiseSum, 0, Table[{t[i], n[i]}, {i, m}]] / Abs[OpAmpProduct[t, m]]
```

Spectrum Math

Propagate noise spectrum

```
SpecProp[prev_, t_] := {#[[1]], Abs[t /. s → 2. π #[[1]]] #[[2]]} & /@ prev
SpecProp[noise_, t_, m_] := FoldList[SpecProp, noise, Table[t[i], {i, m}]]
```

RMS of spectrum

```
Clear[SpecRMS];
SpecRMS[l_List ? (MatrixQ[#, NumberQ] &)] := Block[{i, sqr = 0},
  For[i = 1, i < Length[l], ++i,
   sqr += (l[[i + 1, 1]] - l[[i, 1]]) ((l[[i, 2]] + l[[i + 1, 2]])/2)^2];
  Sqrt[sqr]]
```

Integrated RMS spectrum

```
Clear[RMSSpec];
RMSSpec[l_List ? (MatrixQ[#, NumberQ] &), dir_: (-1)] := Block[{i, sqr = 0, r = N[l]},
  If[dir ≥ 0,
   For[i = 2, i ≤ Length[l], ++i,
    r[[i, 2]] = Sqrt[r[[i - 1, 2]]^2 + r[[i, 2]]^2 (r[[i, 1]] - r[[i - 1, 1]])]],
   For[i = Length[l] - 2, i >= 1, --i,
    r[[i, 2]] = Sqrt[r[[i + 1, 2]]^2 + r[[i, 2]]^2 (r[[i + 1, 1]] - r[[i, 1]])]]];
  r]
```

Instrumentation Amplifier

OpAmp 1

```
n = -1;
z1[n] = ∞;
z2[n] = 100;
opamp[n] = OpAmp[1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[1, z1[n], z2[n], 1.7*^-9, 1.5*^-12];
```

OpAmp 2

```
n = -2;
z1[n] = ∞;
z2[n] = 100;
opamp[n] = OpAmp[1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[1, z1[n], z2[n], 1.7*^-9, 1.5*^-12];
```

Diff OpAmp

```
n = 1;
z1[n] = 4990.;
z2[n] = par[2490., 1/(s 10*^-12)];
opamp[n] = OpAmp[0, z1[n], z2[n]];
opampnoise[n] =
  √(OpAmpNoise[0, z1[n], z2[n], 1.7*^-9, 1.5*^-12]^2 + opampnoise[-1]^2 + opampnoise[-2]^2);
```

Generic Gain Stage

```
n = 2;
z1[n] = 4990.;
z2[n] = 4990.;
z3[n] = 49900.; (* impedance on plus input *)
z4[n] = 1/(s 1*^-6);
opamp[n] = OpAmp[-1, z1[n], z2[n]];
radjnoise = 500*^-9;
adjnoise2 = in^2 Abs[par[z3[n], z4[n]]]^2 +
    FourKT Abs[par[z3[n], z4[n]]] + Abs[z4[n]/(z4[n] + z3[n])]^2 radjnoise^2 /. in → 1.5*^-12;
opampnoise[n] = √(OpAmpNoise[-1, z1[n], z2[n], 1.7*^-9, 1.5*^-12]^2 +
     Abs[1 + z1[n]/z2[n]]^2 adjnoise2);
```

Summing Stage

```
n = 3;
z1[n] = 4990.;
z2[n] = par[3570., 1/(s 10*^-12)];
opamp[n] = OpAmp[-1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[-1, z1[n], z2[n], 1.7*^-9, 1.5*^-12];
```

Driver Stage

```
n = 4;
z1[n] = ∞;
z2[n] = 100;
drivermul = 2;
opamp[n] = OpAmp[1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[1, z1[n], z2[n], 1.7*^-9, 1.5*^-12] / √drivermul;
```

Pole/Zero Stage

```
n = 5;
z1[n] = 2490./2;
z2[n] = ser[49.9, 1/(s 80*^-6)];
opamp[n] = z2[n]/(z2[n] + z1[n]);
opampnoise[n] = Sqrt[FourKT par[z1[n], z2[n]]];
```

Total

```
stages = 5;
opamp[0] = Simplify[OpAmpProduct[opamp, stages]];
opampnoise[0] = Simplify[OpAmpNoiseProduct[opamp, opampnoise, stages]];
```

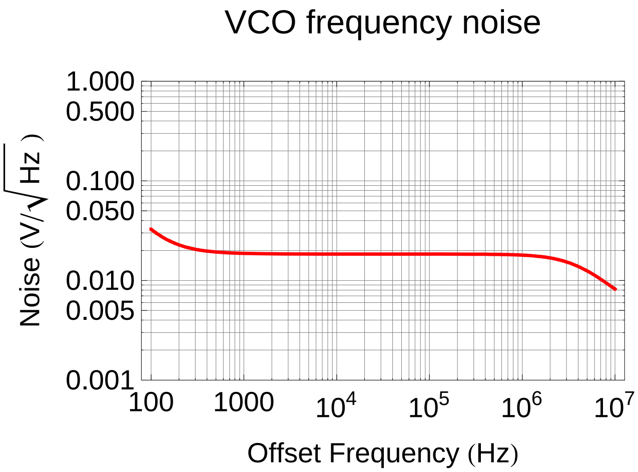
```
Join[{{"Frequency", "Magnitude (dB)", "Phase (°)"}, {}},
  {#, dB[opamp[0]] /. s → 2 π ⅈ #, Arg[opamp[0]] / Degree /. s → 2 π ⅈ #} & /@
   {0, 0.1, 1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	Magnitude (dB)	Phase (°)
0	−8.94667	0
0.1	−8.965	−3.58035
1	−10.478	−31.6228
10	−25.0531	−67.1851
100	−36.5898	−20.8584
1000	−37.2225	−2.21688
10 000	−37.2293	−0.437785
100 000	−37.2326	−2.20327

```
Join[{{"Frequency", "Input Noise", "Output Noise"}, {"Hz", "nV/√Hz", "nV/√Hz"}, {}},
  {#, 1*^9 Abs[opampnoise[0] /. s → 2 π ⅈ #], 1*^9 Abs[opamp[0] opampnoise[0] /. s → 2 π ⅈ #]} & /@
   {1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

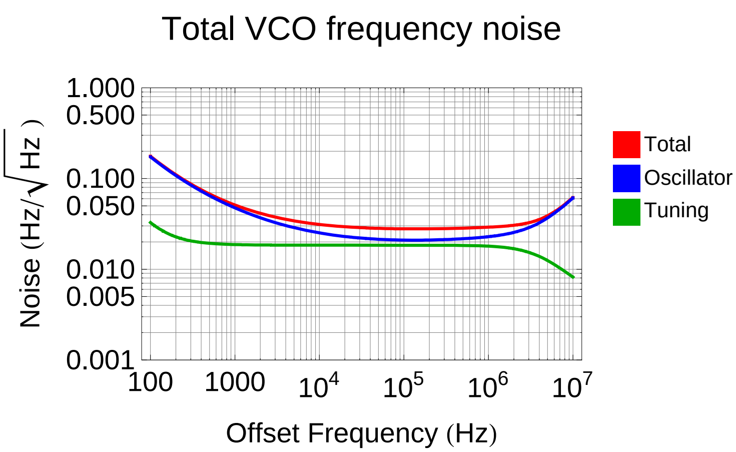
Frequency	Input Noise	Output Noise
Hz	nV/$\sqrt{\text{Hz}}$	nV/$\sqrt{\text{Hz}}$
1	1937.45	579.871
10	621.072	34.7125
100	85.0555	1.25955
1000	52.2574	0.719488
10 000	51.4997	0.708496
100 000	51.4659	0.707764

```
LogLogPlot[tuneSense Abs[opamp[0] opampnoise[0] /. s → 2 π ⅈ f],
 {f, 100, 1*^7}, Frame → True, GridLines → Automatic,
 PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle, PlotRange → {1*^-3, 1},
 PlotLabel → "VCO frequency noise",
 FrameLabel → {"Offset Frequency (Hz)", "Noise (V/√Hz )"}]
```



Combined VCO Frequency Noise

```
ShowLegend[LogLogPlot[
   {√((f VCONoise[f])^2 + (tuneSense Abs[opamp[0] opampnoise[0] /. s → 2 π ⅈ f])^2), f VCONoise[f],
    tuneSense Abs[opamp[0] opampnoise[0] /. s → 2 π ⅈ f]}, {f, 100, 1*^7}, Frame → True,
   GridLines → Automatic, PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue},
     {Thickness [0.007], Darker[Green]}}, BaseStyle → $TextStyle, PlotRange → {1*^-3, 1},
   PlotLabel → "Total VCO frequency noise", FrameLabel →
    {"Offset Frequency (Hz)", "Noise (Hz/√Hz )"}, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "Total"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "Oscillator"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "Tuning"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.7, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Readback

```
n = 6;
z1[n] = ∞;
z2[n] = 100;
opamp[n] = OpAmp[1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[1, z1[n], z2[n], 1.7*^-9, 1.5*^-12];
```

```
n = 7;
z1[n] = par[4990., 200 + 1/(s 20*^-6)];
z2[n] = 3570.;
opamp[n] = OpAmp[-1, z1[n], z2[n]];
opampnoise[n] = OpAmpNoise[-1, z1[n], z2[n], 1.7*^-9, 1.5*^-12];
```

```
(opamp[7] /. s → 2 π ⅈ #) & /@ {1*^-6, 1*^6} // Abs
```

```
{0.715431, 18.5654}
```

```
stages2 = 7;
opamp[r] = Simplify[OpAmpProduct[opamp, stages2]];
opampnoise[r] = Simplify[OpAmpNoiseProduct[opamp, opampnoise, stages2]];
```

- **Output stage only**

```
Join[{{"Frequency", "Magnitude (dB)", "Phase (°)"}, {}},
  {#, dB[opamp[r]/opamp[0]] /. s → 2 π ⅈ #, Arg[opamp[r]/opamp[0]] / Degree /. s → 2 π ⅈ #} & /@
   {0, 0.1, 1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	Magnitude (dB)	Phase (°)
0	-2.90865	0
0.1	-2.89024	3.58752
1	-1.37215	31.6725
10	13.2138	67.175
100	24.7369	20.8185
1000	25.3672	2.19067
10 000	25.374	0.219186
100 000	25.3741	0.0219188

```
Join[{{"Frequency", "Input Noise", "Output Noise"}, {"Hz", "nV/√Hz", "nV/√Hz"}, {}},
  {#, 1*^9 √(Abs[opampnoise[6]]^2 + Abs[opampnoise[7]]^2) /. s → 2 π ⅈ #, 1*^9
      Abs[opamp[6] opamp[7] √(Abs[opampnoise[6]]^2 + Abs[opampnoise[7]]^2) /. s → 2 π ⅈ #]} & /@
   {1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	Input Noise	Output Noise
Hz	$nV/\sqrt{Hz}$	$nV/\sqrt{Hz}$
1	14.2665	12.1817
10	4.98099	22.8037
100	3.37466	58.22
1000	3.33335	61.8363
10 000	3.33292	61.8767
100 000	3.33292	61.8771

■ Input to monitor overall

```
Join[{{"Frequency", "Magnitude (dB)", "Phase (°)"}, {}},
  {#, dB[opamp[r]] /. s → 2 π ⅈ #, Arg[opamp[r]] / Degree /. s → 2 π ⅈ #} & /@
   {0, 0.1, 1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	Magnitude (dB)	Phase (°)
0	−11.8553	0
0.1	−11.8552	0.00716617
1	−11.8501	0.0496656
10	−11.8393	−0.0100769
100	−11.8529	−0.039849
1000	−11.8553	−0.026201
10 000	−11.8553	−0.218599
100 000	−11.8585	−2.18136

```
Join[{{"Frequency", "Input Noise", "Output Noise"}, {"Hz", "nV/√Hz", "nV/√Hz"}, {}},
  {#, 1*^9 Abs[opampnoise[r] /. s → 2 π ⅈ #], 1*^9 Abs[opamp[r] opampnoise[r] /. s → 2 π ⅈ #]} & /@
   {1, 10, 100, 1000, 1*^4, 1*^5}] // TableForm
```

Frequency	Input Noise	Output Noise
Hz	$nV/\sqrt{Hz}$	$nV/\sqrt{Hz}$
1	1938.04	495.285
10	627.434	160.547
100	243.242	62.1431
1000	247.682	63.2603
10 000	247.679	63.2593
100 000	247.762	63.2569

Others