RF AM Stabilization

T0900198-v1
April 22, 2009
Daniel Sigg

Theory of Operations

The frequency stabilization circuit requires a low noise RF detector. The presented design uses a combination of series and shunt diode detectors in a differential arrangement to effectively quadruple the response. A parallel chain of diodes is used to both compensate for temperature drifts and to remove a bias. Under nominal RF load this detector then has zero output and directly be used in compensation network to control a variable RF attenuator ahead of the detector. A relatively high bias current is applied to the diodes to

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

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

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

Model

Parallel and Serial Impedance

$$\text{par}[\text{r1\_}, \text{r2\_}] := \frac{1}{\frac{1}{r1} + \frac{1}{r2}}$$

$$\text{par}[\text{r1\_}, \text{r2\_}, \text{r3\_}] := \frac{1}{\frac{1}{r1} + \frac{1}{r2} + \frac{1}{r3}}$$

```
ser[r1_, r2_] := r1 + r2
ser[r1_, r2_, r3_] := r1 + r2 + r3
```

Pole/zero

$$\text{pole}[\text{f\_}, \text{f0\_}] := \frac{1}{1 + i\frac{f}{f0}}$$

$$\text{zero}[\text{f\_}, \text{f0\_}] := 1 + i\frac{f}{f0}$$

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]

$$\text{OpAmp}[\text{g\_}, \text{z1\_}, \text{z2\_}] := \text{Which}\left[g > 0,\ 1 + \frac{z2}{z1},\ g < 0,\ \frac{z2}{z1},\ \text{True},\ \frac{z2}{z1}\right]$$

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]))]]
```

Flicker Noise: The variable $Flicker determines if flicker noise is added or not. It can also be explicitly specficied with the option Flicker.

```
$Flicker = True;
Clear[OpAmpNoiseFlicker];
Options[OpAmpNoiseFlicker] = {Flicker :> $Flicker};
OpAmpNoiseFlicker[f_, fknee_, opts___] :=
 If[Flicker /. {opts} /. Options[OpAmpNoiseFlicker], Sqrt[fknee/f + 1], 1]
OpAmpNoiseFlicker[f_, fknee_, floor_, opts___] := floor OpAmpNoiseFlicker[f, fknee, opts]
```

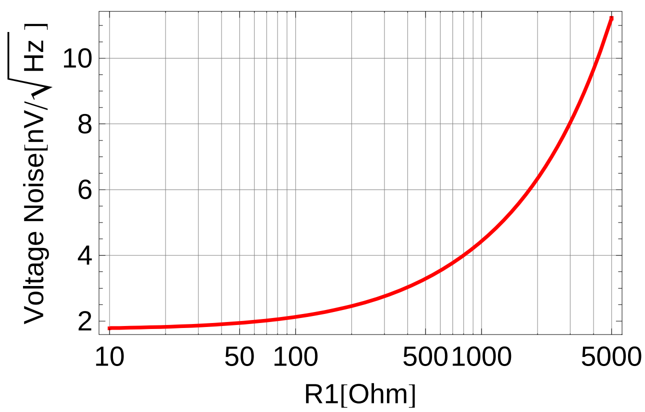
OpAmp Parameters

```
Clear[AD829, OP27]
AD829[f_] := {s → 2 π ⅈ f, en → enAD829, in → inAD829,
    enfloor → enfloorAD829, infloor → infloorAD829} //.
   {enAD829 → OpAmpNoiseFlicker[f, ekneeAD829, enfloorAD829],
    inAD829 → OpAmpNoiseFlicker[f, ikneeAD829, infloorAD829],
    ekneeAD829 → 50, ikneeAD829 → 100, (*guess*)
    enfloorAD829 → 1.7*^-9, infloorAD829 → 1.5*^-12};
OP27[f_] := {s → 2 π ⅈ f, en → enOP27, in → inOP27,
    enfloor → enfloorOP27, infloor → infloorOP27} //.
   {enOP27 → OpAmpNoiseFlicker[f, ekneeOP27, enfloorOP27],
    inOP27 → OpAmpNoiseFlicker[f, ikneeOP27, infloorOP27],
    ekneeOP27 → 2.7, ikneeOP27 → 140,
    enfloorOP27 → 3.0*^-9, infloorOP27 → 0.4*^-12};
LT1028[f_] := {s → 2 π ⅈ f, en → enLT1028, in → inLT1028,
    enfloor → enfloorLT1028, infloor → infloorLT1028} //.
   {enLT1028 → OpAmpNoiseFlicker[f, ekneeLT1028, enfloorLT1028],
    inLT1028 → OpAmpNoiseFlicker[f, ikneeLT1028, infloorLT1028],
    ekneeLT1028 → 3.5, ikneeLT1028 → 250,
    enfloorLT1028 → 0.85*^-9, infloorLT1028 → 1*^-12};
LT1128[f_] := {s → 2 π ⅈ f, en → enLT1128, in → inLT1128,
    enfloor → enfloorLT1128, infloor → infloorLT1128} //.
   {enLT1128 → OpAmpNoiseFlicker[f, ekneeLT1128, enfloorLT1128],
    inLT1128 → OpAmpNoiseFlicker[f, ikneeLT1128, infloorLT1128],
    ekneeLT1128 → 3.5, ikneeLT1128 → 250,
    enfloorLT1128 → 0.85*^-9, infloorLT1128 → 1*^-12};
AD797[f_] := {s → 2 π ⅈ f, en → enAD797, in → inAD797,
    enfloor → enfloorAD797, infloor → infloorAD797} //.
   {enAD797 → OpAmpNoiseFlicker[f, ekneeAD797, enfloorAD797],
    inAD797 → OpAmpNoiseFlicker[f, ikneeAD797, infloorAD797],
    ekneeAD797 → 50, ikneeAD797 → 100, (*guess*)
    enfloorAD797 → 0.9*^-9, infloorAD797 → 2*^-12};
LT1012[f_] := {s → 2 π ⅈ f, en → enLT1012, in → inLT1012,
    enfloor → enfloorLT1012, infloor → infloorLT1012} //.
   {enLT1012 → OpAmpNoiseFlicker[f, ekneeLT1012, enfloorLT1012],
    inLT1012 → OpAmpNoiseFlicker[f, ikneeLT1012, infloorLT1012],
    ekneeLT1012 → 2.5, ikneeLT1012 → 120, (*guess*)
    enfloorLT1012 → 14*^-9, infloorLT1012 → 6*^-15};
```

Examples (AD829)

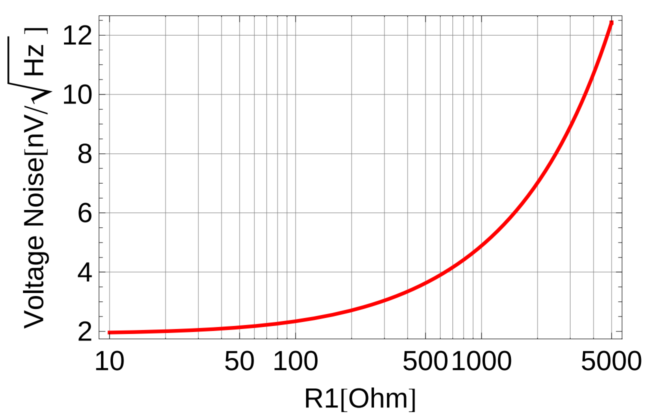
Non-Inverting configuration: input noise w/ gain of 10 as function of r1

```
LogLinearPlot[1*^9 OpAmpNoise[+1, r, 9 r, en, in] /. AD829[1000],
 {r, 10, 5000}, FrameLabel → {"R1[Ohm]", "Voltage Noise[nV/√Hz ]"}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```



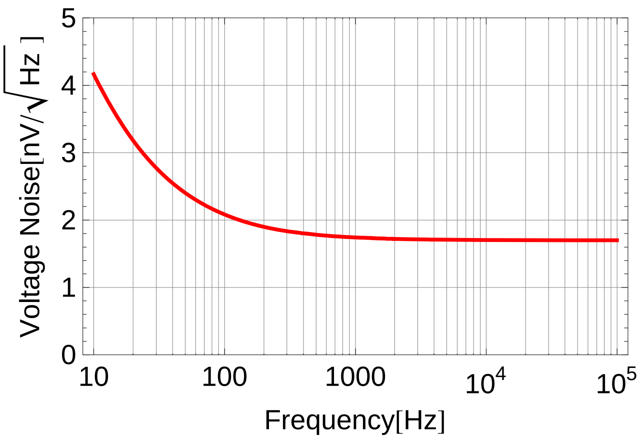
Inverting configuration: input noise w/ gain of 10 as function of r1

```
LogLinearPlot[1*^9 OpAmpNoise[-1, r, 10 r, en, in] /. AD829[1000],
 {r, 10, 5000}, FrameLabel → {"R1[Ohm]", "Voltage Noise[nV/√Hz ]"}, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```



Input voltage noise with flicker.

```
LogLinearPlot[1*^9 en /. AD829[f], {f, 10, 1*^5},
 FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"}, Frame → True, PlotRange → {0, 5},
 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];
  √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]
```

Shottky Diode Model

We are using the noise model of a HSMS-281K. See data sheet and application note 956-3.

Parameters

```
HSMS281Param =
  {BV → 25, CJ → 1.1*^-12, EG → 0.69, IBV → 1*^-5, IS → 4.8*^-9, n → 1.08, RS → 10, T → 273.15 + 25};
ShottkyParam = HSMS281Param;
```

```
8.33*^-5 n T /. HSMS281Param
```

```
0.0268228
```

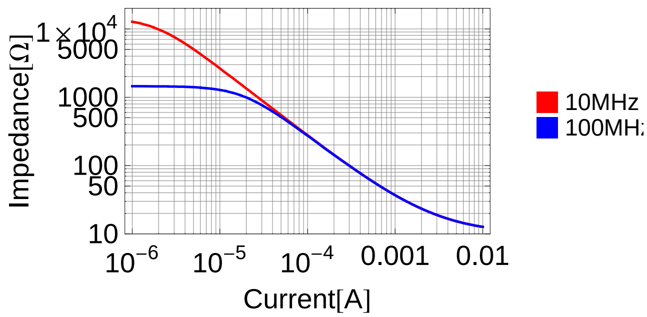
Resistance Model

```
RJ[i_] := (8.33*^-5 n T)/(IS + i) /. ShottkyParam
RV[i_] := RJ[i] + RS /. ShottkyParam
ZShottky[i_, f_] := ser[RS, par[RJ[i], 1/(2 π ⅈ f CJ)]] /. ShottkyParam
```

```
Abs[RJ[30*^-6]]
Abs[RJ[100*^-6]]
Abs[RJ[300*^-6]]
Abs[RJ[1000*^-6]]
```

```
893.949

268.215

89.4078

26.8226
```

```
ShowLegend[LogLogPlot[{Abs[ZShottky[i, 10*^6]], Abs[ZShottky[i, 100*^6]]},
   {i, 1*^-6, 1*^-2}, FrameLabel → {"Current[A]", "Impedance[Ω]"},
   PlotRange → {10, 2*^4}, Frame → True, GridLines → Automatic, PlotStyle →
    {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Black}},
   BaseStyle → $TextStyle, ImageSize → Scaled[0.65]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10MHz"},
    {Graphics[{Blue, Rectangle[{1, 1}]}], "100MHz"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 11}, LegendPosition → {0.9, -0.0},
   LegendSize → {0.4, 0.2}, LegendTextSpace → 3, LegendShadow → None}] // Quiet
```



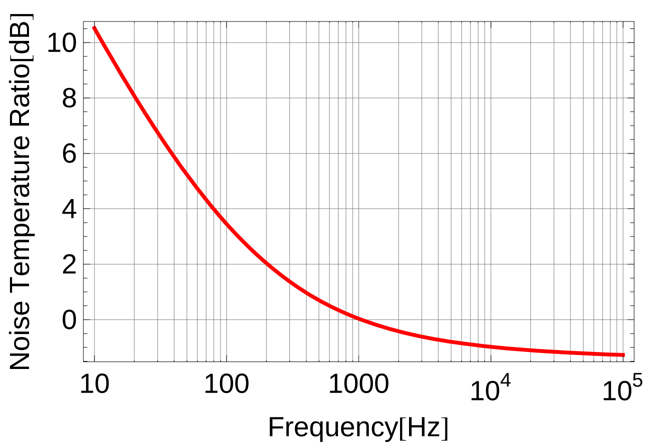
Noise Temperature Ratio

```
ShottkyNTRLevel[f_] := 10^(-1.5/20.)
```

$$\texttt{ShottkyNTRFlicker[f\_]} := \sqrt{\frac{0.8*\text{^}2}{f} + \frac{1}{3}\sqrt[10]{\frac{1*\text{^}2}{f}}} \quad \texttt{(* fudge a little bit *)}$$

$$\texttt{ShottkyNTR[f\_]} := \sqrt{\texttt{ShottkyNTRLevel[f]}^2 + \texttt{ShottkyNTRFlicker[f]}^2}$$

```
LogLinearPlot[dB[ShottkyNTR[f]], {f, 10, 100 000},
 FrameLabel → {"Frequency[Hz]", "Noise Temperature Ratio[dB]"}, PlotRange → All, Frame → True,
 GridLines → Automatic, PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```

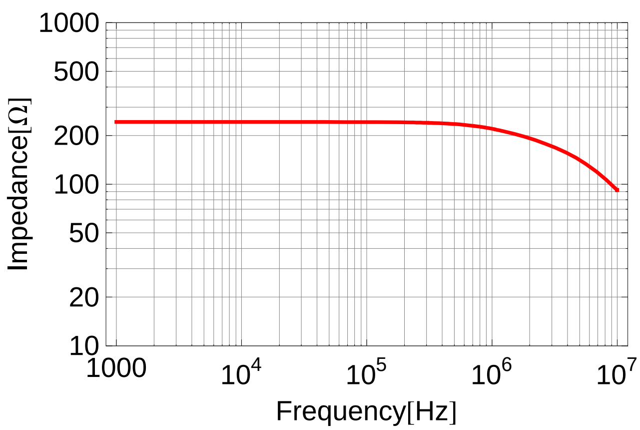


Detection chain

```
detParam = {Rbias → 100*^3, ibias → 240*^-6, Cclamp1 → 330*^-12,
   Cclamp2 → 150*^-12, Cbias → 20*^-6, Rdrv → 2000, Rterm → 50, TransformerRatio → 2};
```

$$zdet = par\left[Rbias, \frac{1}{s\,Cclamp2}, ser\left[ZShottky[ibias, f], par\left[ZShottky[ibias, f], ser\left[TransformerRatio^2\,Rterm, \frac{1}{s\,Cclamp1}\right]\right]\right]\right];$$

```
LogLogPlot[Abs[zdet] /. detParam /. s → 2 π ⅈ f,
 {f, 1*^3, 1*^7}, FrameLabel → {"Frequency[Hz]", "Impedance[Ω]"},
 PlotRange → {10, 1000}, Frame → True, GridLines → Automatic,
 PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Black}},
 BaseStyle → $TextStyle]
```

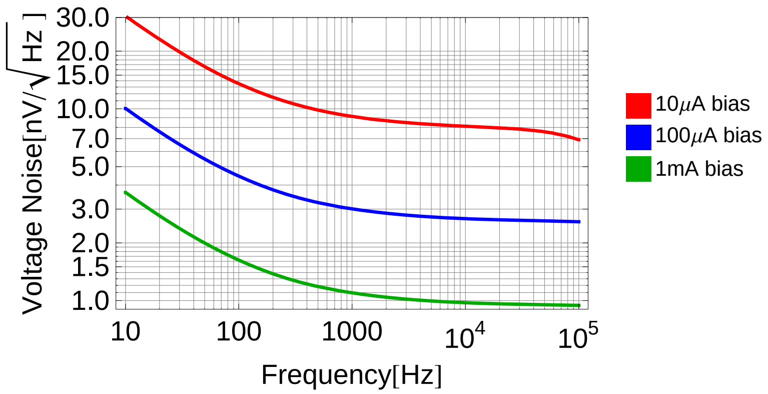


Only valid when the resistance of the bias resistor is larger than the diode junction resistance because the flicker noise is applied to the parallel arrangement as a whole.

$$diodenoise[0] = ShottkyNTR[f]\,\sqrt{FourKT\,Abs[zdet]}\ /.\ detParam;$$

$$diodenoisebias[0, ib\_] := ShottkyNTR[f]\,\sqrt{FourKT\,Abs[zdet]}\ /.\ \{ibias \to ib\}\ /.\ detParam\ /.\ s \to 2\,\pi\,ⅈ\,f;$$

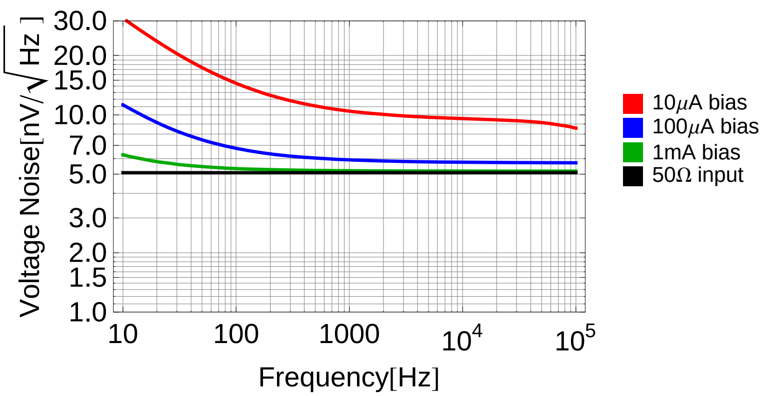
```
ShowLegend[LogLogPlot[{1*^9 Abs[diodenoisebias[0, 1*^-5]],
    1*^9 Abs[diodenoisebias[0, 1*^-4]], 1*^9 Abs[diodenoisebias[0, 1*^-3]]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   Frame → True, PlotRange → {0.9, 30}, GridLines → Automatic, PlotStyle →
    {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Darker[Green]}},
   BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.85, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Noise of the 50Ω termination resistor is added twice because we fold the two audio sidebands of the carrier on top of each other. They also have a gain of 2 × transformer ratio.

```
detnoise[0] = √(diodenoise[0]^2 + 2 (2 TransformerRatio)^2 FourKT Rterm) /. detParam;
detnoisebias[0, ib_] =
  √(diodenoisebias[0, ib]^2 + 2 (2 TransformerRatio)^2 FourKT Rterm) /. {ibias → ib} /. detParam /.
   s → 2 π ⅈ f;
```

```
ShowLegend[LogLogPlot[{1*^9 Abs[detnoisebias[0, 1*^-5]], 1*^9 Abs[detnoisebias[0, 1*^-4]],
    1*^9 Abs[detnoisebias[0, 1*^-3]], 1*^9 Sqrt[2 (2 TransformerRatio)^2 FourKT 50] /. detParam},
   {f, 10, 1*^5}, FrameLabel -> {"Frequency[Hz]", "Voltage Noise[nV/Sqrt[Hz]]"},
   Frame -> True, PlotRange -> {1, 30}, GridLines -> Automatic, PlotStyle ->
    {{Thickness[0.007], Red}, {Thickness[0.007], Blue}, {Thickness[0.007], Darker[Green]},
     {Thickness[0.007], Black}}, BaseStyle -> $TextStyle, ImageSize -> Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"},
    {Graphics[{Black, Rectangle[{1, 1}]}], "50Ω input"}},
   BaseStyle -> {FontFamily -> "Helvetica", FontSize -> 10}, LegendPosition -> {0.85, -0.0},
   LegendSize -> {0.5, 0.3}, LegendTextSpace -> 5, LegendShadow -> None}] // Quiet
```

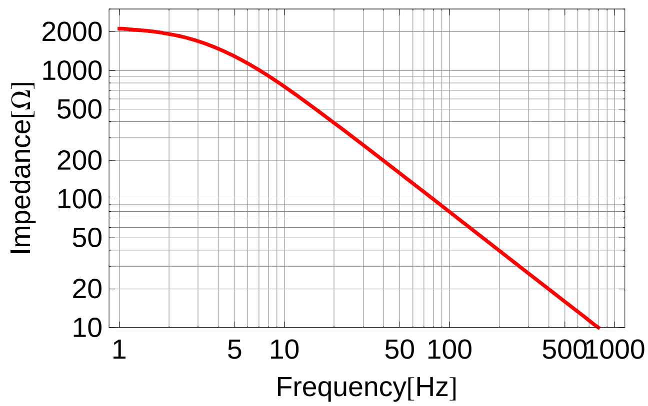


Bias chain

Resistor ans Shottky noise in bias chain:

```
zbias = par[Rbias, 1/(s Cbias), ser[ZShottky[ibias, f], ZShottky[ibias, f], Rdrv]];

NTRratio = 1 + (ShottkyNTR[f] - 1) Abs[ser[ZShottky[ibias, f], ZShottky[ibias, f]]/Rdrv];

diodenoise[1] = NTRratio Sqrt[FourKT Abs[zbias]] /. detParam /. s -> 2 π i f;
diodenoisebias[1, ib_] :=
  NTRratio Sqrt[FourKT Abs[zbias]] /. {ibias -> ib} /. detParam /. s -> 2 π i f;
```

```
LogLogPlot[Abs[zbias] /. detParam /. s → 2 π ⅈ f,
 {f, 1*^0, 1*^3}, FrameLabel → {"Frequency[Hz]", "Impedance[Ω]"},
 PlotRange → {10, 3000}, Frame → True, GridLines → Automatic,
 PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Black}},
 BaseStyle → $TextStyle]
```



Bias driver and supply noise:

Assume <100 $\frac{\text{nV}}{\sqrt{\text{Hz}}}$ noise on the BIAS drive at f > 10Hz.

Assume <200 $\frac{\text{nV}}{\sqrt{\text{Hz}}}$ noise on the 24V supply side at f > 10Hz.

```
zDrv = 10 000;
cDrv = 20*^-6;
driveNoise[0] = 100*^-9;
supplyNoise[0] = 200*^-9;
```

$$\text{splyDrv}[0] = \frac{\text{par}\left[2\,\text{ZShottky}[\text{ibias}, \text{f}] + \text{Rdrv}, \frac{1}{\text{s Cbias}}\right]}{\text{par}\left[2\,\text{ZShottky}[\text{ibias}, \text{f}] + \text{Rdrv}, \frac{1}{\text{s Cbias}}\right] + \text{Rbias}};$$

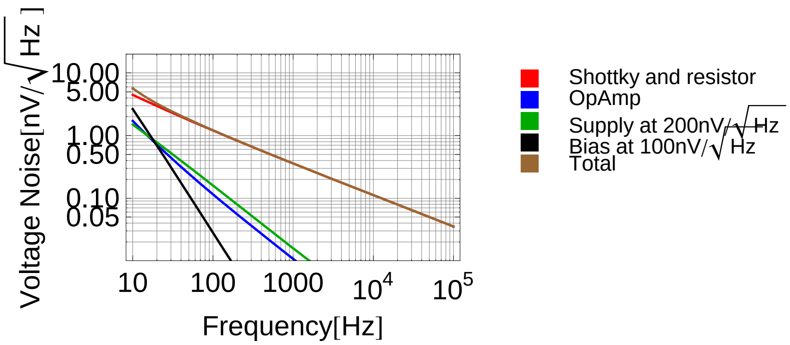
```
splyNoise[0, ib_] := Abs[splyDrv[0]] supplyNoise[0] /. {ibias → ib} /. detParam /. s → 2 π ⅈ f;
splyNoise[0] = splyNoise[0, ibias] /. detParam /. s → 2 π ⅈ f;
```

```
biasDrv[0, ib_] = par[Rbias, 1/(s Cbias)] / (par[Rbias, 1/(s Cbias)] + 2 ZShottky[ibias, f] + Rdrv) /. {ibias → ib};
biasDrv[0] = biasDrv[0, ibias];
OpAmpDrv[0, ib_] =
  biasDrv[0] OpAmp[-1, zDrv, par[zDrv, 1/(s cDrv)]] /. {ibias → ib} /. detParam /. s → 2 π ⅈ f;
OpAmpDrv[0] = OpAmpDrv[0, ibias] /. detParam /. s → 2 π ⅈ f;
OpAmpDrv[1] = OpAmp[-1, zDrv, par[zDrv, 1/(s cDrv)]] /. s → 2 π ⅈ f;
OpAmpDrv[2] = OpAmp[1, ∞, 1]; OpAmpNoiseDrv[1, ib_] =
 OpAmpNoise[-1, zDrv, par[zDrv, 1/(s cDrv)], en, in] /. {ibias → ib} /. detParam /. s → 2 π ⅈ f /.
  OP27[f];
OpAmpNoiseDrv[1] = OpAmpNoiseDrv[1, ibias] /. detParam /. s → 2 π ⅈ f;
OpAmpNoiseDrv[2, ib_] =
  √(OpAmpNoise[1, ∞, 0, en, in]^2 + in^2 Abs[par[zDrv, 1/(s cDrv)]]^2 + FourKT Abs[par[zDrv, 1/(s cDrv)]]) /.
     {ibias → ib} /. detParam /. s → 2 π ⅈ f /. OP27[f];
OpAmpNoiseDrv[2] = OpAmpNoiseDrv[2, ibias] /. detParam /. s → 2 π ⅈ f;
(* averaged output noise through bais network*)
OpAmpNoiseDrv[0, ib_] = Abs[biasDrv[0, ib]]
   √(1/2 (Abs[OpAmpDrv[1]]^2 OpAmpNoiseDrv[1, ib]^2 + Abs[OpAmpDrv[2]]^2 OpAmpNoiseDrv[2, ib]^2));
OpAmpNoiseDrv[0] = OpAmpNoiseDrv[0, ibias] /. detParam /. s → 2 π ⅈ f;

drvNoise[0, ib_] = driveNoise[0] Abs[OpAmpDrv[0, ib]];
drvNoise[0] = driveNoise[0] Abs[OpAmpDrv[0]];
```

Total noise is bias chain:

```
detnoise[1] = √(diodenoise[1]^2 + OpAmpNoiseDrv[0]^2 + splyNoise[0]^2 + drvNoise[0]^2) ;
detnoisebias[1, ib_] =
  √(diodenoisebias[1, ib]^2 + OpAmpNoiseDrv[0, ib]^2 + splyNoise[0, ib]^2 + drvNoise[0, ib]^2) /.
    {ibias → ib} /. detParam /. s → 2 π ⅈ f;
```

```
ShowLegend[LogLogPlot[{1*^9 diodenoise[1],
   1*^9 OpAmpNoiseDrv[0], 1*^9 splyNoise[0], 1*^9 drvNoise[0], 1*^9 detnoise[1]},
  {f, 10, 100 000}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
  Frame → True, PlotRange → {0.01, 20}, GridLines → Automatic, PlotStyle →
   {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Darker[Green]},
    {Thickness [0.007], Black}, {Thickness [0.007], Brown}}, BaseStyle → $TextStyle,
  ImageSize → Scaled[0.8]], {{{Graphics[{Red, Rectangle[{1, 1}]}], "Shottky and resistor"},
   {Graphics[{Blue, Rectangle[{1, 1}]}], "OpAmp"},
   {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "Supply at 200nV/√Hz "},
   {Graphics[{Black, Rectangle[{1, 1}]}], "Bias at 100nV/√Hz "},
   {Graphics[{Brown, Rectangle[{1, 1}]}], "Total"}},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.87, -0.0},
  LegendSize → {1, 0.4}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```

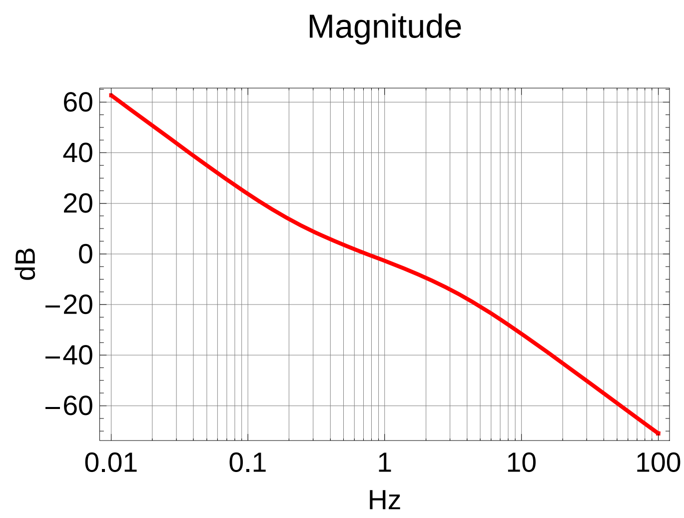


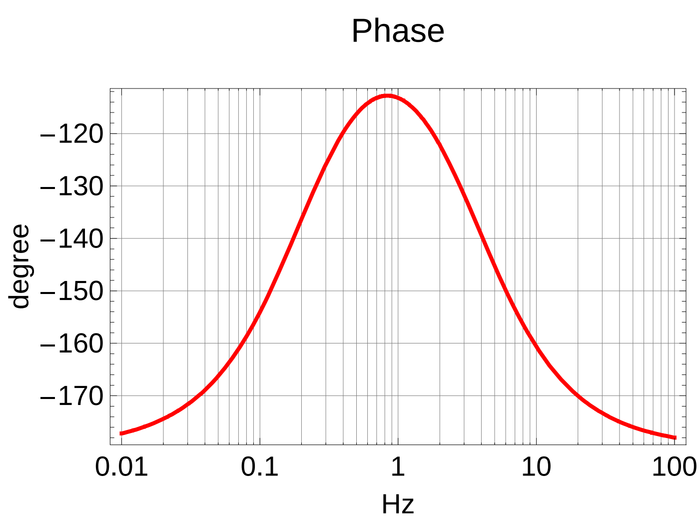
Bias chain transfer function

```
comp[f_] := (ⅈ f + 0.7)/(ⅈ f) (ⅈ f + 0.2)/(ⅈ f)
dB[OpAmpDrv[0]] /. f → 1
```

-4.62572

```
BodePlot[OpAmpDrv[0] comp[f], {f, 0.01, 100}, PlotStyle →
  {{Thickness[0.007], Red}, {Thickness[0.007], Blue}, {Thickness[0.007], Darker[Green]},
   {Thickness[0.007], Black}, {Thickness[0.007], Brown}}, BaseStyle → $TextStyle]
```



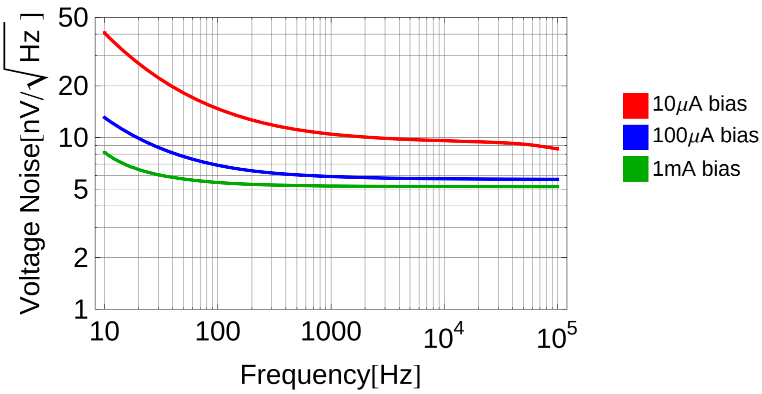


- **Sum of Detector and Bias chain**

$$detnoise[3] = \sqrt{detnoise[0]^2 + detnoise[1]^2};$$

$$detnoisebias[3, ib\_] = \sqrt{detnoisebias[0, ib]^2 + detnoisebias[1, ib]^2};$$

```
ShowLegend[LogLogPlot[{1*^9 Abs[detnoisebias[3, 1*^-5]],
    1*^9 Abs[detnoisebias[3, 1*^-4]], 1*^9 Abs[detnoisebias[3, 1*^-3]]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   Frame → True, PlotRange → {1, 50}, GridLines → Automatic, PlotStyle →
    {{Thickness[0.007], Red}, {Thickness[0.007], Blue}, {Thickness[0.007], Darker[Green]}},
   BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.85, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Instrumentation Amp

$$\texttt{z1[1]} = \texttt{par}\left[1000, \frac{1}{s\ 150*\hat{}-12}\right];$$

```
z2[1] = 110; (* half middle resistor *)
```

$$\texttt{opamp[1]} = 1 + \frac{\texttt{z1[1]}}{\texttt{z2[1]}} \text{ /. } s \to 2\pi i f; \text{ (* differential gain *)}$$

`opampnoise[1, ib_] =`

$$\sqrt{\left(2\,en^2 + (in\,\text{Abs}[zdet])^2 + (in\,\text{Abs}[zbias])^2 + 2\,(in\,\text{Abs}[\text{par}[z1[1], z2[1]]])^2 + 2\,\text{FourKT}\,\text{Abs}[\text{par}[z1[1], z2[1]]]\,\text{Abs}\left[\frac{1}{1+\frac{z1[1]}{2\,z2[1]}}\right]\right)} \text{ /. } \{ibias \to ib\} \text{ /. detParam /. } s \to 2\pi i f \text{ /. AD829}[f];$$

`opampnoise[1] = opampnoise[1, ibias] /. detParam /. s → 2 π i f;`

$$\texttt{totalnoise[1, ib\_]} = \text{Abs}[\text{opamp}[1]]\sqrt{\text{opampnoise}[1, ib]^2 + \text{detnoisebias}[3, ib]^2} \text{ /. } s \to 2\pi i f;$$

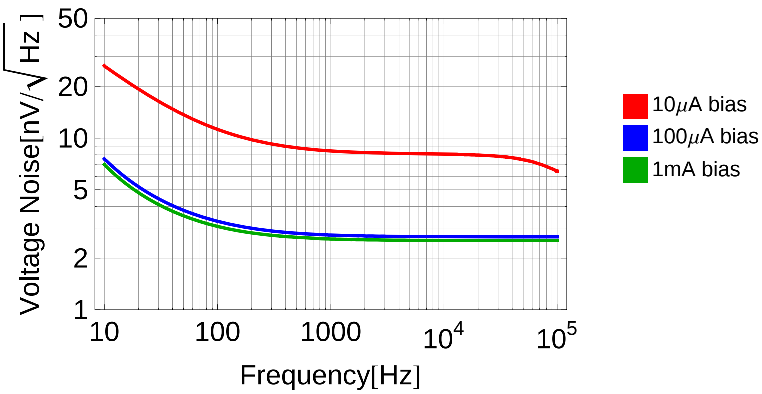
```
totalnoise[1] = totalnoise[1, ibias] /. detParam /. s → 2 π i f;
totalnoise[1, 1*^-4] /. f → 1000
Abs[opamp[1]] opampnoise[1, 1*^-4] /. f → 1000
```

6.58145×10^{-8}

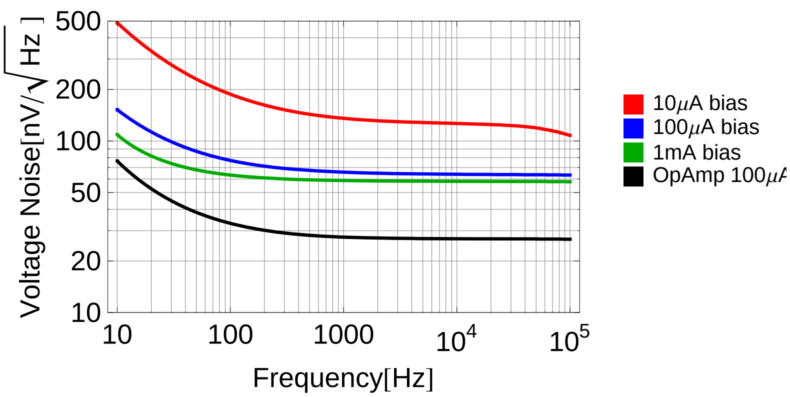
2.75513×10^{-8}

The noise of the OpAmp is dominated by the current noise through the source impedance for bias currents smaller than 100μA.

```
ShowLegend[
  LogLogPlot[{1*^9 opampnoise[1, 1*^-5], 1*^9 opampnoise[1, 1*^-4], 1*^9 opampnoise[1, 1*^-3]},
   {f, 10, 1*^5}, FrameLabel -> {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   Frame -> True, PlotRange -> {1, 50}, GridLines -> Automatic, PlotStyle ->
    {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Darker[Green]},
     {Thickness [0.007], Black}}, BaseStyle -> $TextStyle, ImageSize -> Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"}},
   BaseStyle -> {FontFamily -> "Helvetica", FontSize -> 10}, LegendPosition -> {0.85, -0.0},
   LegendSize -> {0.5, 0.3}, LegendTextSpace -> 5, LegendShadow -> None}] // Quiet
```



```
ShowLegend[LogLogPlot[{1*^9 totalnoise[1, 1*^-5], 1*^9 totalnoise[1, 1*^-4],
    1*^9 totalnoise[1, 1*^-3], 1*^9 Abs[opamp[1]] opampnoise[1, 1*^-4]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   Frame → True, PlotRange → {10, 500}, GridLines → Automatic, PlotStyle →
    {{Thickness[0.007], Red}, {Thickness[0.007], Blue}, {Thickness[0.007], Darker[Green]},
     {Thickness[0.007], Black}}, BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"},
    {Graphics[{Black, Rectangle[{1, 1}]}], "OpAmp 100μA bias"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.85, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Differential Amp

The 50Ω termination noise is correlated between the two detector chains it has to be added coherently, whereas the shottky noise does not.

$$z1[2] = \frac{3300}{2};$$

$$z2[2] = par\left[3300, \frac{1}{s\ 68*\hat{}-12}\right];$$

```
opamp[2] = OpAmp[0, z1[2], z2[2]] /. s → 2 π i f; (* gain *)
opampnoise[2] = OpAmpNoise[0, z1[2], z2[2], en, in] /. s → 2 π i f /. AD829[f];
```

$$\text{cornoise} = \text{Abs}[\text{opamp}[1]]\sqrt{(2\ \text{TransformerRatio})^2\ \text{FourKT Rterm}}\ \text{/. detParam};$$

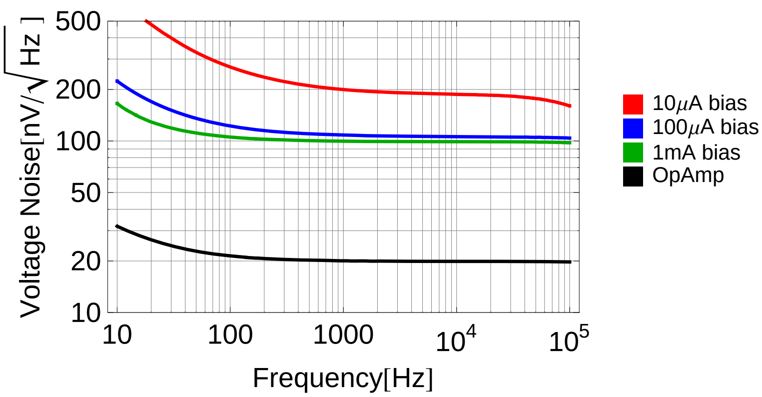
$$\text{totalnoise}[2, \text{ib\_}] = \text{Abs}[\text{opamp}[2]]\sqrt{\text{opampnoise}[2]^2 + \frac{\text{totalnoise}[1, \text{ib}]^2 + \text{cornoise}^2}{2}}\ \text{/. s} \to 2\pi i f;$$

```
totalnoise[2] = totalnoise[2, ibias] /. detParam /. s → 2 π i f;
totalnoise[2, 1*^-4] /. f → 1000
Abs[opamp[2]] opampnoise[2] /. f → 1000
```

1.08187×10^{-7}

2.00493×10^{-8}

```
ShowLegend[LogLogPlot[{1*^9 totalnoise[2, 1*^-5], 1*^9 totalnoise[2, 1*^-4],
    1*^9 totalnoise[2, 1*^-3], 1*^9 Abs[opamp[2]] opampnoise[2]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   Frame → True, PlotRange → {10, 500}, GridLines → Automatic, PlotStyle →
    {{Thickness[0.007], Red}, {Thickness[0.007], Blue}, {Thickness[0.007], Darker[Green]},
     {Thickness[0.007], Black}}, BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"},
    {Graphics[{Black, Rectangle[{1, 1}]}], "OpAmp"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.85, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```

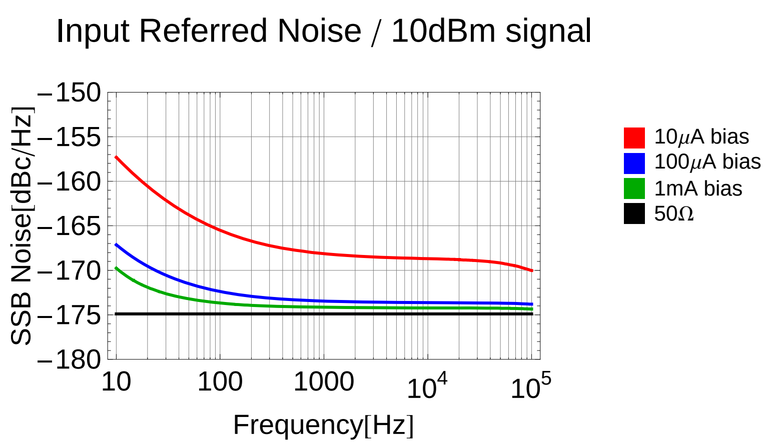


Input Referred Noise

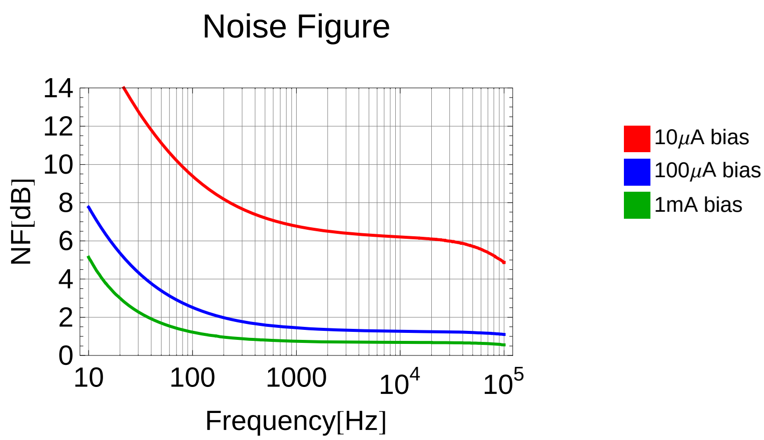
```
carrier = 1/√2 ;(* 10 dBm *)
loss = FromdB[-1.];
detectorgain = 2 Abs[opamp[1] opamp[2] /. f → 0] TransformerRatio loss /. detParam
carriernoise[ib_] := totalnoise[2, ib] / (detectorgain carrier)
FiftyOhm = √(2 FourKT 50) / carrier;
noisefigure[ib_] := carriernoise[ib] / FiftyOhm
dB[carriernoise[1*^-4] /. f → 1000]
dB[FiftyOhm]
```

```
71.9483

-173.447

-174.895
```

```
ShowLegend[LogLinearPlot[{dB[carriernoise[1*^-5]],
   dB[carriernoise[1*^-4]], dB[carriernoise[1*^-3]], dB[Sqrt[2 FourKT 50] / carrier]},
  {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "SSB Noise[dBc/Hz]"},
  PlotLabel → "Input Referred Noise / 10dBm signal", Frame → True, PlotRange → {-180, -150},
  GridLines → Automatic, PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue},
    {Thickness [0.007], Darker[Green]}, {Thickness [0.007], Black}}, BaseStyle → $TextStyle,
  ImageSize → Scaled[0.8]], {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"},
   {Graphics[{Blue, Rectangle[{1, 1}]}], "100μA bias"},
   {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"},
   {Graphics[{Black, Rectangle[{1, 1}]}], "50Ω"}},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.75, -0.0},
  LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



```
ShowLegend[
  LogLinearPlot[{dB[noisefigure[1*^-5]], dB[noisefigure[1*^-4]], dB[noisefigure[1*^-3]]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "NF[dB]"}, PlotLabel → "Noise Figure",
   Frame → True, PlotRange → {0, 14}, GridLines → Automatic, PlotStyle →
    {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Darker[Green]},
     {Thickness [0.007], Black}}, BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "10μA bias"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "100μA bias"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "1mA bias"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.75, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Bias Servo Compensation Network

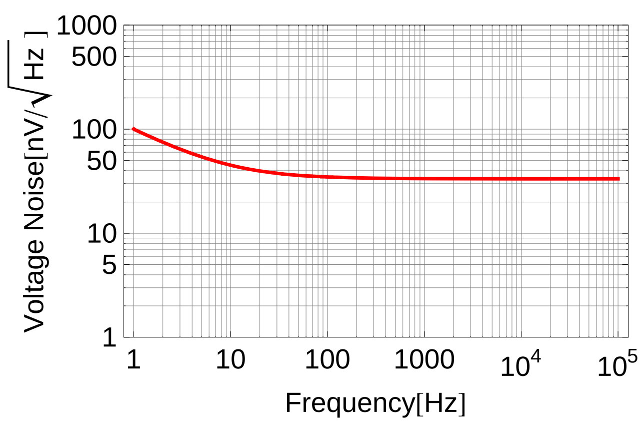
```
prm = {fknee → 0.7, R2 → 10 000, R1 → 20 000, C → 20*^-6}
op1 = OpAmp[-1, R1, 1/(s C) + R2] // Simplify
Solve[op1 == R2/R1 (s + 2 π fknee)/s, C][[1]]
1*^6 C /. % /. prm
LogLogPlot[1*^9 OpAmpNoise[-1, R1, 1/(s C) + R2, en, in] /. OP27[f] /. prm, {f, 1, 1*^5},
 FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"}, Frame → True, PlotRange → {1, 1000},
 GridLines → Automatic, PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```

$\{\text{fknee} \to 0.7,\ \text{R2} \to 10\,000,\ \text{R1} \to 20\,000,\ \text{C} \to \frac{1}{50\,000}\}$

$\frac{\text{R2} + \frac{1}{\text{C s}}}{\text{R1}}$

$\{\text{C} \to \frac{1}{2\ \text{fknee}\ \pi\ \text{R2}}\}$

22.7364




```
Abs[op1] /. s → 2 π ⅈ 1. /. prm
```

$10^{\frac{-4.63}{20}}$

```
gain = 20 % %%
```

0.638995

0.586813

7.49941

```
prm2 = {fknee → 0.2, R2 → 1/(2 π fknee C), C → 20*^-6}
prm3 = {fknee → 0.2, R1 → 300 000, R2 → 40 000, C → 20*^-6}
R2 //. prm2
op2 = OpAmp[-1, R1, 1/(s C) + R2] // Simplify
fr = FindRoot[(Abs[op2] == 1/gain //. prm2 /. s → 2 π ⅈ 1), {R1, 1000}]
LogLogPlot[1*^9 OpAmpNoise[-1, R1, 1/(s C) + R2, en, in] /. LT1012[f] //. prm2 /. fr, {f, 1, 1*^5},
 FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"}, Frame → True, PlotRange → {1, 1000},
 GridLines → Automatic, PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```

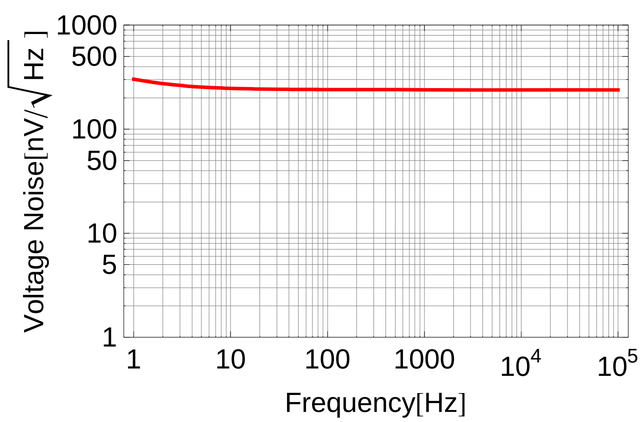
$$\left\{\text{fknee} \to 0.2,\ \text{R2} \to \frac{1}{2\,\text{C}\,\text{fknee}\,\pi},\ \text{C} \to \frac{1}{50\,000}\right\}$$

$$\left\{\text{fknee} \to 0.2,\ \text{R1} \to 300\,000,\ \text{R2} \to 40\,000,\ \text{C} \to \frac{1}{50\,000}\right\}$$

39 788.7

$$\frac{\text{R2} + \frac{1}{\text{C s}}}{\text{R1}}$$

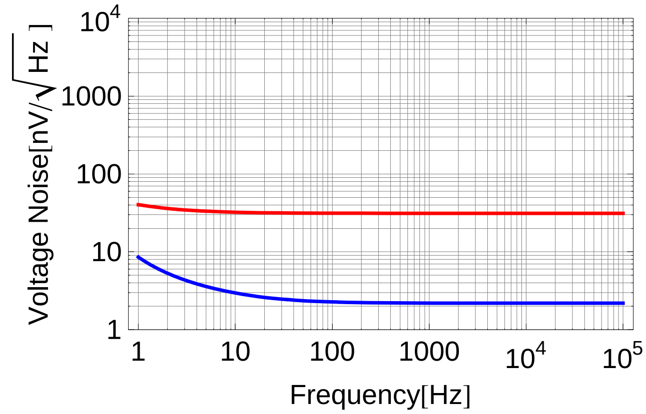
{R1 → 304 302.}



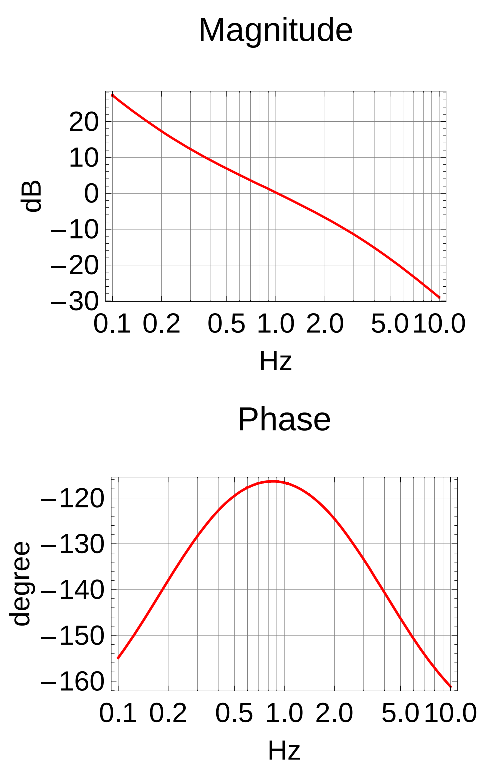

```
Abs[op1] /. s → 2 π ⅈ 1. //. prm2 /. fr
```

0.133344

```
LogLogPlot[{1*^9 Abs[op2] OpAmpNoise[-1, R1, 1/(s C) + R2, en, in] /. LT1012[f] //. prm2 /. fr,
  (Abs[op2] //. prm2 /. fr) 1*^9 Abs[op1] OpAmpNoise[-1, R1, 1/(s C) + R2, en, in] /. OP27[f] /. prm},
 {f, 1, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
 Frame → True, PlotRange → {1, 10 000}, GridLines → Automatic,
 PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue}}, BaseStyle → $TextStyle]
```




```
t1 = op1 //. prm /. s → 2 π ⅈ f;
t2 = op2 //. prm3 /. fr /. s → 2 π ⅈ f;
BodePlot[20 t1 t2 OpAmpDrv[0], {f, 0.1, 10}, PlotStyle →
  {{Thickness [0.007], Red}, {Thickness [0.007], Blue}, {Thickness [0.007], Darker[Green]},
   {Thickness [0.007], Black}, {Thickness [0.007], Brown}}, BaseStyle → $TextStyle]
```



Bias Setpoints

```
set = Table[10^(i/(20×5.)), {i, 0, 12*10 - 1}];
dac = 1 / set * (2^16 - 1) // Round // Reverse
```

```
{4231, 4330, 4431, 4534, 4640, 4748, 4858, 4971, 5087, 5206, 5327, 5451, 5578, 5708, 5841,
 5977, 6116, 6259, 6404, 6554, 6706, 6862, 7022, 7186, 7353, 7524, 7700, 7879, 8063,
 8250, 8443, 8639, 8840, 9046, 9257, 9473, 9693, 9919, 10150, 10387, 10629, 10876,
 11129, 11389, 11654, 11925, 12203, 12487, 12778, 13076, 13381, 13692, 14011, 14337,
 14671, 15013, 15363, 15721, 16087, 16462, 16845, 17237, 17639, 18050, 18470,
 18901, 19341, 19791, 20252, 20724, 21207, 21701, 22206, 22723, 23253, 23794,
 24349, 24916, 25496, 26090, 26698, 27320, 27956, 28607, 29273, 29955, 30653,
 31367, 32098, 32845, 33610, 34393, 35194, 36014, 36853, 37711, 38590, 39489,
 40409, 41350, 42313, 43299, 44307, 45339, 46395, 47476, 48582, 49713, 50871,
 52056, 53269, 54510, 55779, 57079, 58408, 59769, 61161, 62585, 64043, 65535}
```

AM Stabilization Servo Compensation Network

Voltage Controlled Attenuator

$$\text{VCApole} = \frac{1}{2\,\pi\,0.2*\hat{}-6}$$

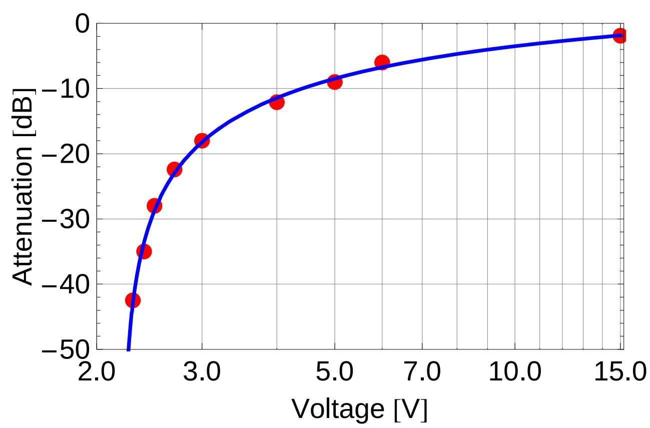
795775.

```
VCAactuator = {{15, -1.9}, {6, -6}, {5, -9}, {4, -12.1},
   {3, -18}, {2.7, -22.4}, {2.5, -28}, {2.4, -35}, {2.3, -42.5}};
```

$$\text{VCAfit}[x\_] = \text{Fit}\left[\text{VCAactuator}, \left\{1, \sqrt[2]{x-2.2}, \sqrt[4]{x-2.2}, \sqrt[6]{x-2.2}, \sqrt[8]{x-2.2}\right\}, x\right]$$

$$-355.953 + 1352.29\,(-2.2+x)^{1/8} - 1087.06\,(-2.2+x)^{1/6} + 65.0966\,(-2.2+x)^{1/4} + 9.43308\,\sqrt{-2.2+x}$$

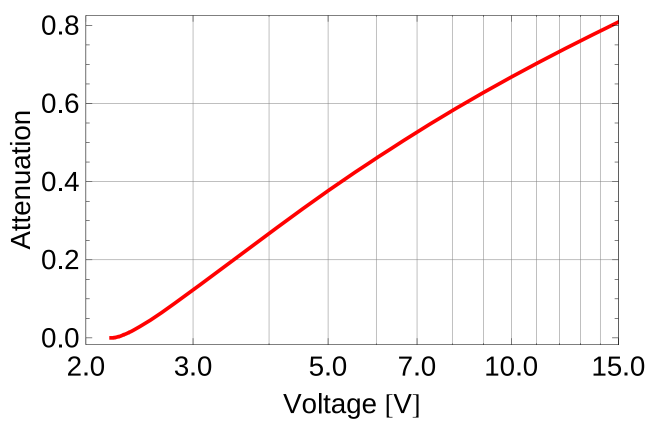
```
Show[ListLogLinearPlot[VCAactuator, PlotStyle → {Red, PointSize[0.03]},
  Frame → True, GridLines → Automatic, PlotRange → {{2, 15.2}, {-50, 0}},
  FrameLabel → {"Voltage [V]", "Attenuation [dB]"}, BaseStyle → $TextStyle],
 LogLinearPlot[VCAfit[x], {x, 1, 15}, PlotRange → {{0, 15}, All},
  PlotStyle → {Thickness[0.007], RGBColor[0, 0, 1]}]]
```



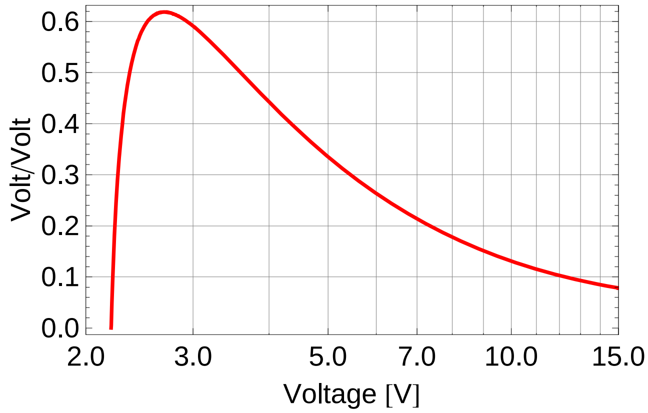
```
wpoint = -8;
wrange = 24;
wabs = 3.55; (* 24dBm *)
wvolt = x /. FindRoot[VCAfit[x] == wpoint, {x, 5}]
wvolt2 = x /. FindRoot[VCAfit[x] == wpoint - wrange, {x, 5}]
D[wabs FromdB[VCAfit[x]], x] /. x → wvolt
D[wabs FromdB[VCAfit[x]], x] /. x → wvolt2
wmax = FindMaximum[ D[wabs FromdB[VCAfit[x]], x], {x, 3}]
```

```
5.23789

2.42797

0.315633

0.558609

{0.618598, {x → 2.69043}}
```

```
LogLinearPlot[ FromdB[VCAfit[x]], {x, 2, 15}, Frame → True, GridLines → Automatic,
 PlotRange → {{2, 15}, All}, PlotStyle → {Thickness [0.007], Red},
 FrameLabel → {"Voltage [V]", "Attenuation"}, BaseStyle → $TextStyle]
```



```
LogLinearPlot[D[wabs FromdB[VCAfit[y]], y] /. y → x, {x, 2, 15}, Frame → True,
 GridLines → Automatic, PlotRange → {{2, 15}, All}, PlotStyle → {Thickness [0.007], Red},
 FrameLabel → {"Voltage [V]", "Volt/Volt"}, BaseStyle → $TextStyle]
```



Detector Transfer Function

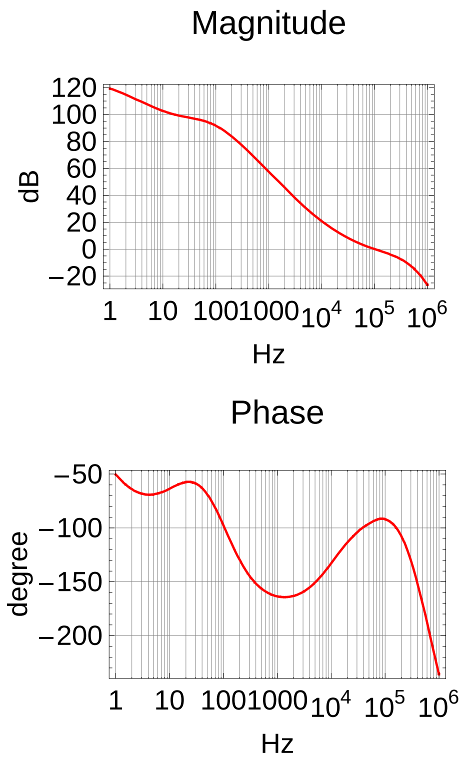
- **Data**
- **Compensation Design**

```
comp[f_] :=
 2*^5 pole[f, 100] zero[f, 1*^5] pole[f, 100] zero[f, 1*^4] pole[f, 0.756] zero[f, 15.12]
trans[f_] := comp[f] wmax[[1]] pole[f, VCApole] dettrans[f]
```

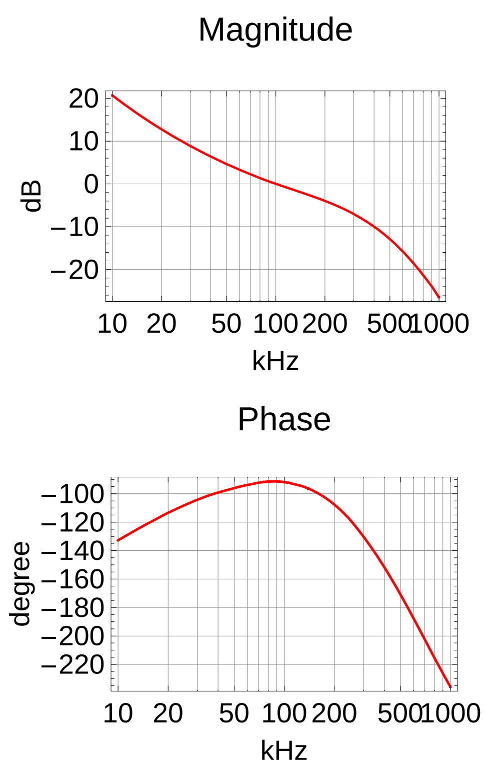
$$\text{gainadj} = \frac{1}{\text{Abs}[\text{trans}[1*\text{^}5]]}$$

```
0.294998
```

```
BodePlot[gainadj trans[f], {f, 1, 1000000}, MagnitudeRange → All,
 PhaseRange → All, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle]
```




```
BodePlot[gainadj trans[1000 f], {f, 10, 1000}, MagnitudeRange → All, PhaseRange → All,
 XAxisLabel → "kHz", PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle]
```



Compensation Network

This includes a polarity switch, an excitation summing junction, a boost gain, a resistive divider network, the compensation filter, a limiter and an output driver.

Polarity Switch

```
z1[1] = 3300;
```

$$z2[1] = par\left[3300,\ \frac{1}{s\ 27*\verb|^|-12}\right];$$

```
opamp[1] = OpAmp[-1, z1[1], z2[1]] /. s → 2 π ⅈ f;
opampnoise[1] = OpAmpNoise[-1, z1[1], z2[1], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

Excitation Summing Junction

```
z1[2] = 3300;
```

$$z2[2] = par\left[3300,\ \frac{1}{s\ 10*\verb|^|-12}\right];$$

```
opamp[2] = OpAmp[-1, z1[2], z2[2]] /. s → 2 π ⅈ f;
opampnoise[2] = OpAmpNoise[-1, z1[2], z2[2], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

Boost Gain

```
z1[3] = 1500;
```

$$z2[3] = par\left[1*\verb|^|6,\ \frac{1}{s\ 10*\verb|^|-9}\right];$$

```
opamp[3] = OpAmp[1, z1[3], z2[3]] /. s → 2 π ⅈ f;
opampnoise[3] = OpAmpNoise[1, z1[3], z2[3], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

Resistive Divider Network

```
z1[4] = 3500;
opamp[4] = gainadj;
```

$$opampnoise[4] = \frac{\sqrt{FourKT\ Abs[z1[4]] + in^2\ Abs[z1[4]]^2}}{2\ gainadj}\ /.\ s \to 2\,\pi\,ⅈ\,f\ /.\ AD829[f];$$

Compensation Filter

```
z1[5] = 82;
```

$$z2[5] = \frac{1}{s\ 10*\verb|^|-9} + 82;$$

```
opamp[5] = OpAmp[1, z1[5], z2[5]] /. s → 2 π ⅈ f;
opampnoise[5] = OpAmpNoise[1, z1[5], z2[5], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

Limiter

```
z1[6] = ∞;
z2[6] = 100;
opamp[6] = OpAmp[1, z1[6], z2[6]] /. s → 2 π ⅈ f;
opampnoise[6] = OpAmpNoise[1, z1[6], z2[6], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

Output Driver

```
z1[7] = 10 000;
z2[7] = par[10 000, 526 + 1/(s 20*^-6)];
opamp[7] = OpAmp[-1, z1[7], z2[7]] /. s → 2 π ⅈ f;
opampnoise[7] = OpAmpNoise[-1, z1[7], z2[7], en, in] /. s → 2 π ⅈ f /. AD829[f];
```

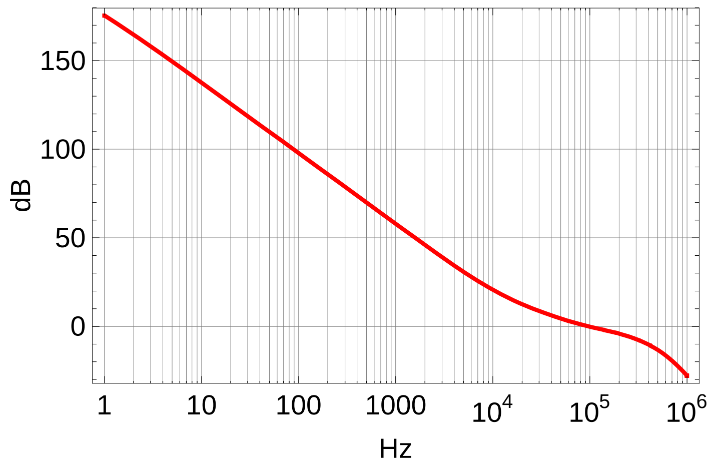
Total

```
stages = 7;
opamp[0] = Simplify[OpAmpProduct[opamp, stages]];
opampnoise[0] = Simplify[OpAmpNoiseProduct[opamp, opampnoise, stages]];

loop[0] = dettrans[f] opamp[0] wmax[[1]] pole[f, VCApole];
loopnoise[0] = Sqrt[opampnoise[0]^2 + totalnoise[2]^2];

Join[{{"Frequency", "Magnitude (dB)", "Phase (°)"}, {}},
  {#, dB[loop[0]] /. f → #, Arg[loop[0]] / Degree /. f → #} & /@
   {0.01, 0.1, 1, 10, 100, 1000, 1*^4, 1*^5, 1*^6}] // TableForm
```

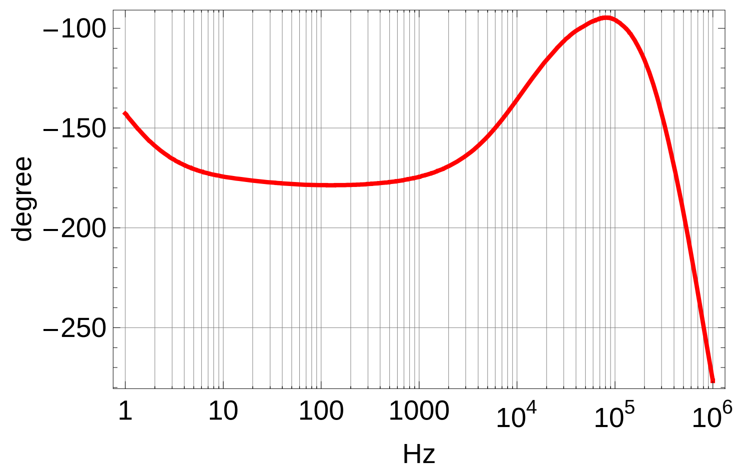
Frequency	Magnitude (dB)	Phase (°)
0.01	219.905	−90.7559
0.1	199.831	−97.5157
1	175.515	−142.718
10	137.581	−174.299
100	97.9065	−178.577
1000	57.9553	−174.435
10 000	20.7162	−135.746
100 000	−0.143135	−95.8403
1 000 000	−27.8508	83.3147

```
BodePlot[loop[0], {f, 1, 1 000 000}, MagnitudeRange → All, PhaseRange → All,
 PhaseReference → -180, PlotStyle → {Thickness[0.007], Red}, BaseStyle → $TextStyle]
```

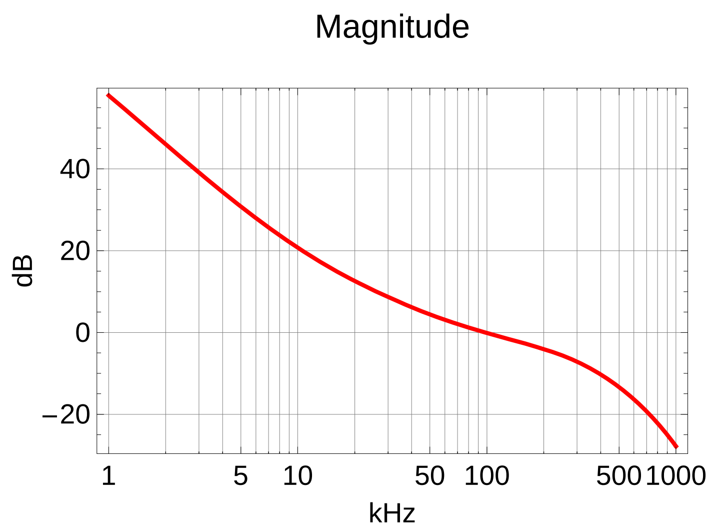
Magnitude

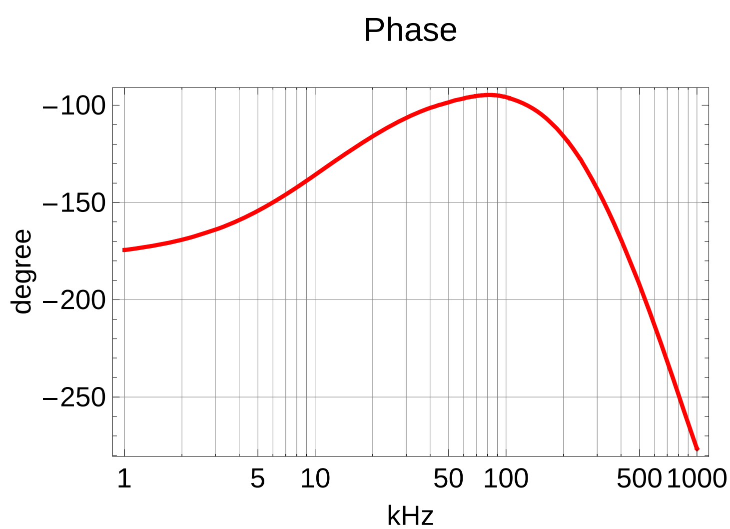


Phase

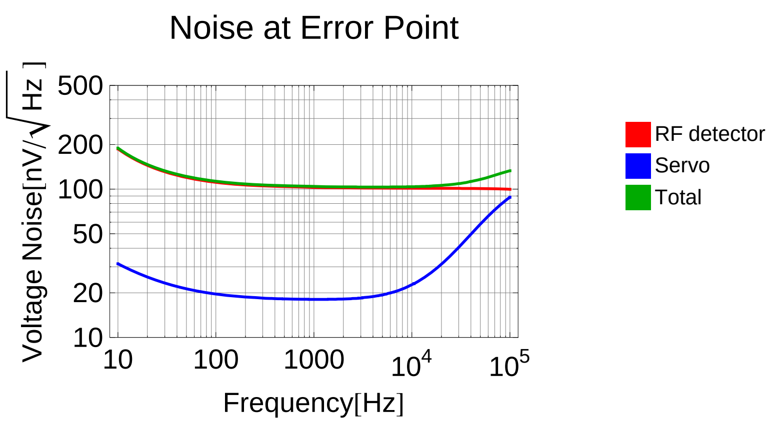


```
BodePlot[loop[0] /. f → 1000 f, {f, 1, 1000}, MagnitudeRange → All,
 PhaseRange → All, XAxisLabel → "kHz", PhaseReference → -180,
 PlotStyle → {Thickness [0.007], Red}, BaseStyle → $TextStyle]
```





```
ShowLegend[LogLogPlot[{1*^9 totalnoise[2], 1*^9 opampnoise[0], 1*^9 loopnoise[0]},
   {f, 10, 1*^5}, FrameLabel → {"Frequency[Hz]", "Voltage Noise[nV/√Hz ]"},
   PlotLabel → "Noise at Error Point", Frame → True, PlotRange → {10, 500},
   GridLines → Automatic, PlotStyle → {{Thickness [0.007], Red}, {Thickness [0.007], Blue},
     {Thickness [0.007], Darker[Green]}, {Thickness [0.007], Black}},
   BaseStyle → $TextStyle, ImageSize → Scaled[0.8]],
  {{{Graphics[{Red, Rectangle[{1, 1}]}], "RF detector"}, {Graphics[{Blue, Rectangle[{1, 1}]}],
     "Servo"}, {Graphics[{Darker[Green], Rectangle[{1, 1}]}], "Total"}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 10}, LegendPosition → {0.85, -0.0},
   LegendSize → {0.5, 0.3}, LegendTextSpace → 5, LegendShadow → None}] // Quiet
```



Attenuator Chain

```
att = Solve[{par[r1, r2 + par[r1, rt]] == rt, par[r1, rt] / (r2 + par[r1, rt]) == FromdB[-1.5]}, {r1, r2}]
```

{{r2 → 0.173554 rt, r1 → 11.61 rt}}

```
rtt = 40*^3 / 11.61 (* termination *)
att /. rt → rtt (* r2: series resistor; r1: resistor to GND *)
par[rtt, r1] /. att /. rt → rtt (* last resistor in chain *)
```

3445.31

{{r2 → 597.945, r1 → 39 999.8}}

{3172.08}

```
FromdB[7 × 1.5] (* half the gain range *)
% gainadj (* initial gain drop *)
attfirst = Solve[{par[r1, r2 + par[r1, rt]] == rt, par[r1, rt]/(r2 + par[r1, rt]) == %}, {r1, r2}]
attfirst /. rt → rtt (* First stage: r2: series; r1: to GND  *)
par[r1, rr] /. att /. rr → r1 /. % /. rt → rtt (* combined 2nd res. to GND *)
```

3.34965

0.988141

{{r2 → 0.0119306 rt, r1 → 167.641 rt}}

{{r2 → 41.1047, r1 → 577 576.}}

{{37 409.1}}

```
vc /. Solve[{v/r1 + (v - v15)/r2 + (v - vc)/r3 == 0, (vc - vcc)/r4 == (vc - v)/r3}, vc, v][[1]] // Expand
```

$$-\frac{r1\,r4\,v15}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}+\frac{r1\,r2\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}+\frac{r1\,r3\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}+\frac{r2\,r3\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}$$

$$\frac{r1\,r2\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}+\frac{r1\,r3\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}+\frac{r2\,r3\,vcc}{r1\,r2+r1\,r3+r2\,r3-r1\,r4-r2\,r4}\ //\ \mathrm{Simplify}$$

$$\frac{(r2\,r3+r1\,(r2+r3))\,vcc}{r2\,(r3-r4)+r1\,(r2+r3-r4)}$$