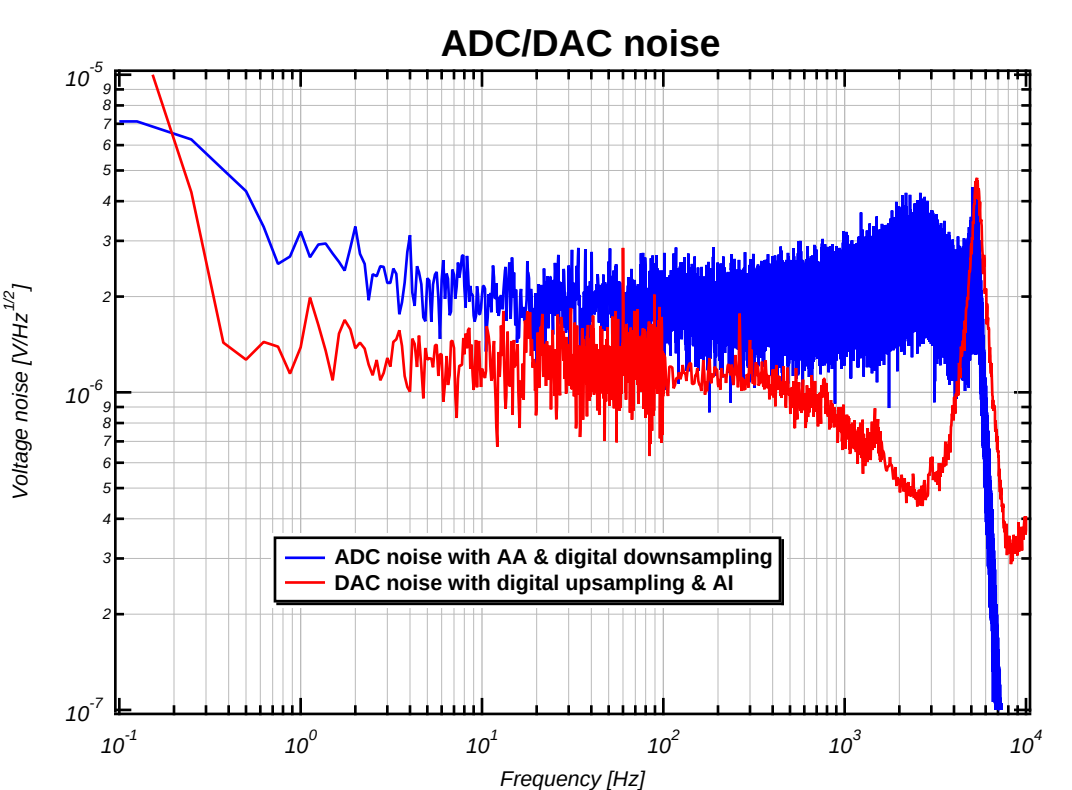
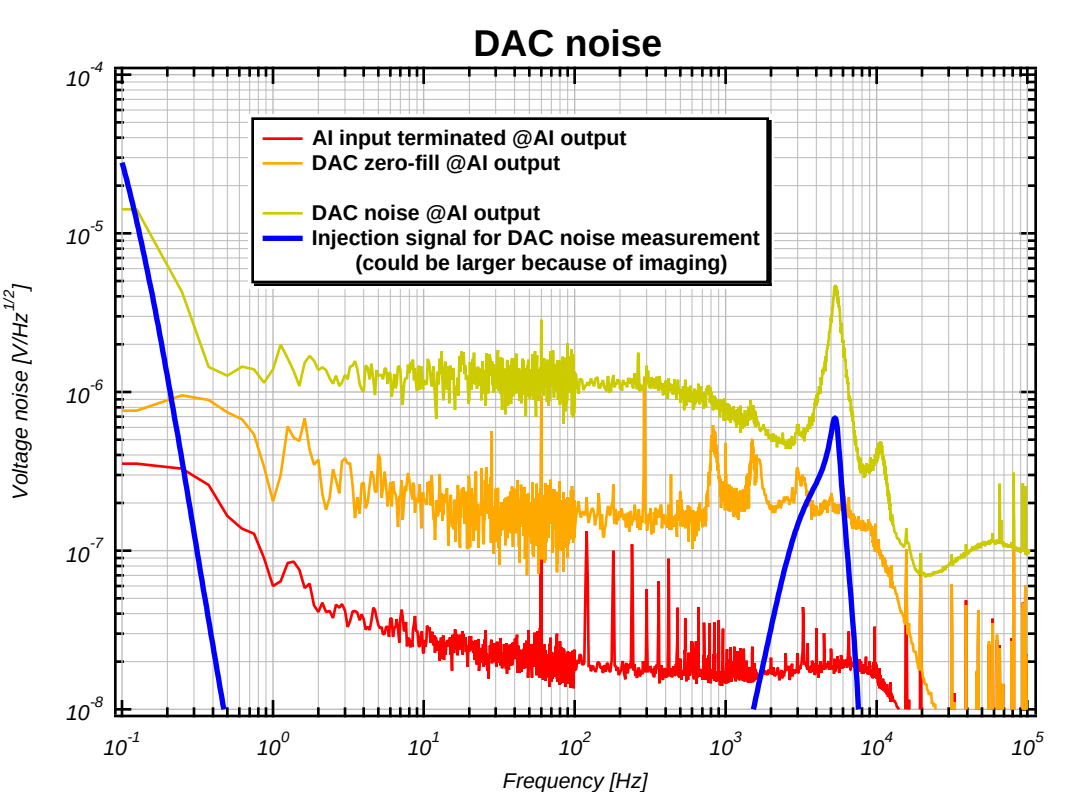
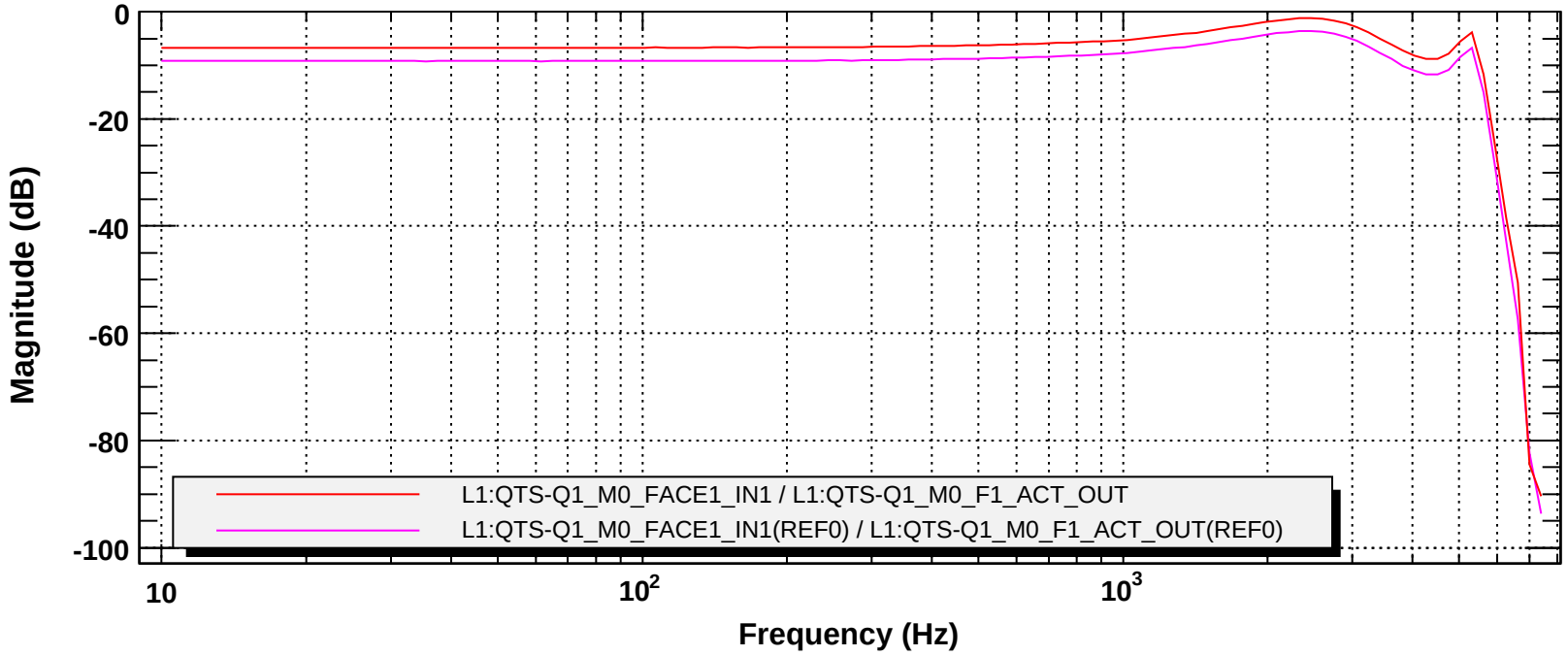
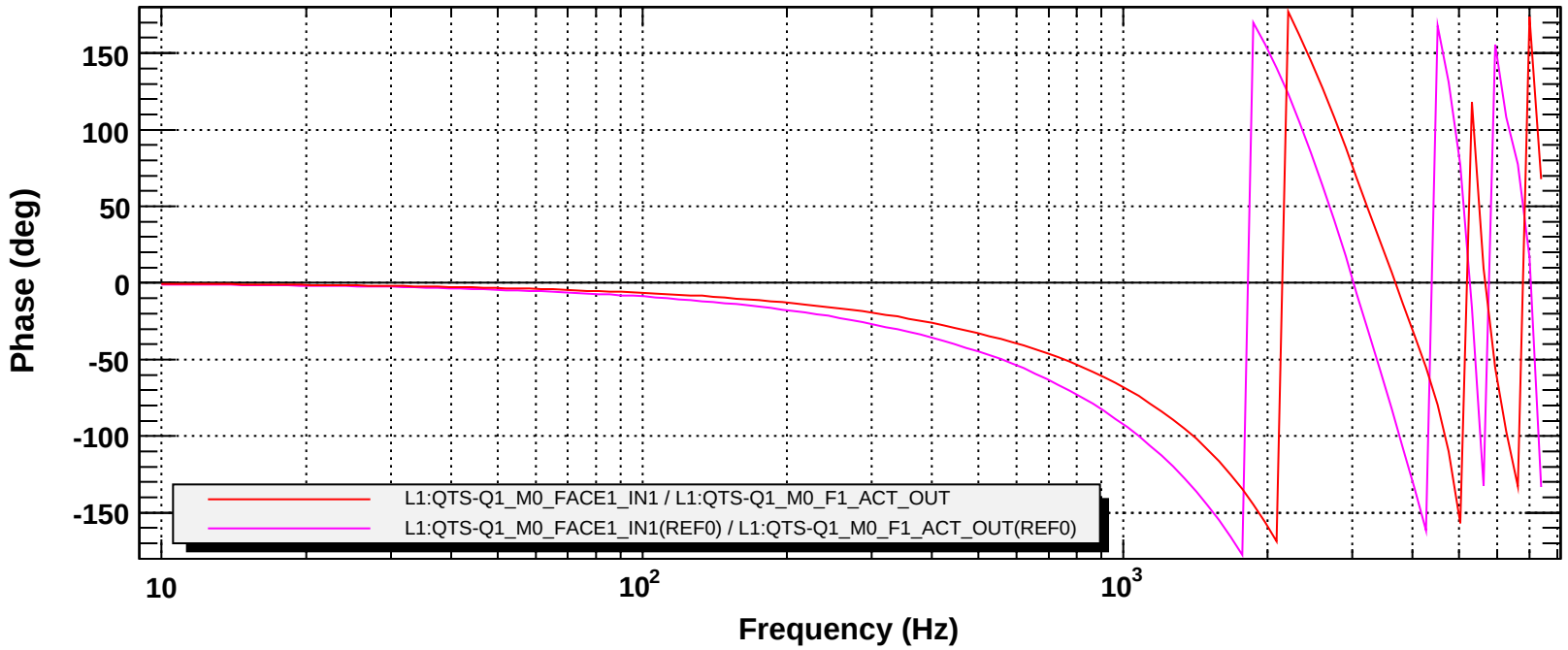

ADC/DAC noise
Voltage noise [V/Hz^1/2]
Frequency [Hz]
ADC noise with AA & digital downsampling
DAC noise with digital upsampling & AI


DAC noise
AI input terminated @AI output
DAC zero-fill @AI output
DAC noise @AI output
Injection signal for DAC noise measurement
(could be larger because of imaging)
Voltage noise [V/Hz^1/2]
Frequency [Hz]
10^-4
10^-5
10^-6
10^-7
10^-8
10^-1
10^0
10^1
10^2
10^3
10^4
10^5

Transfer function



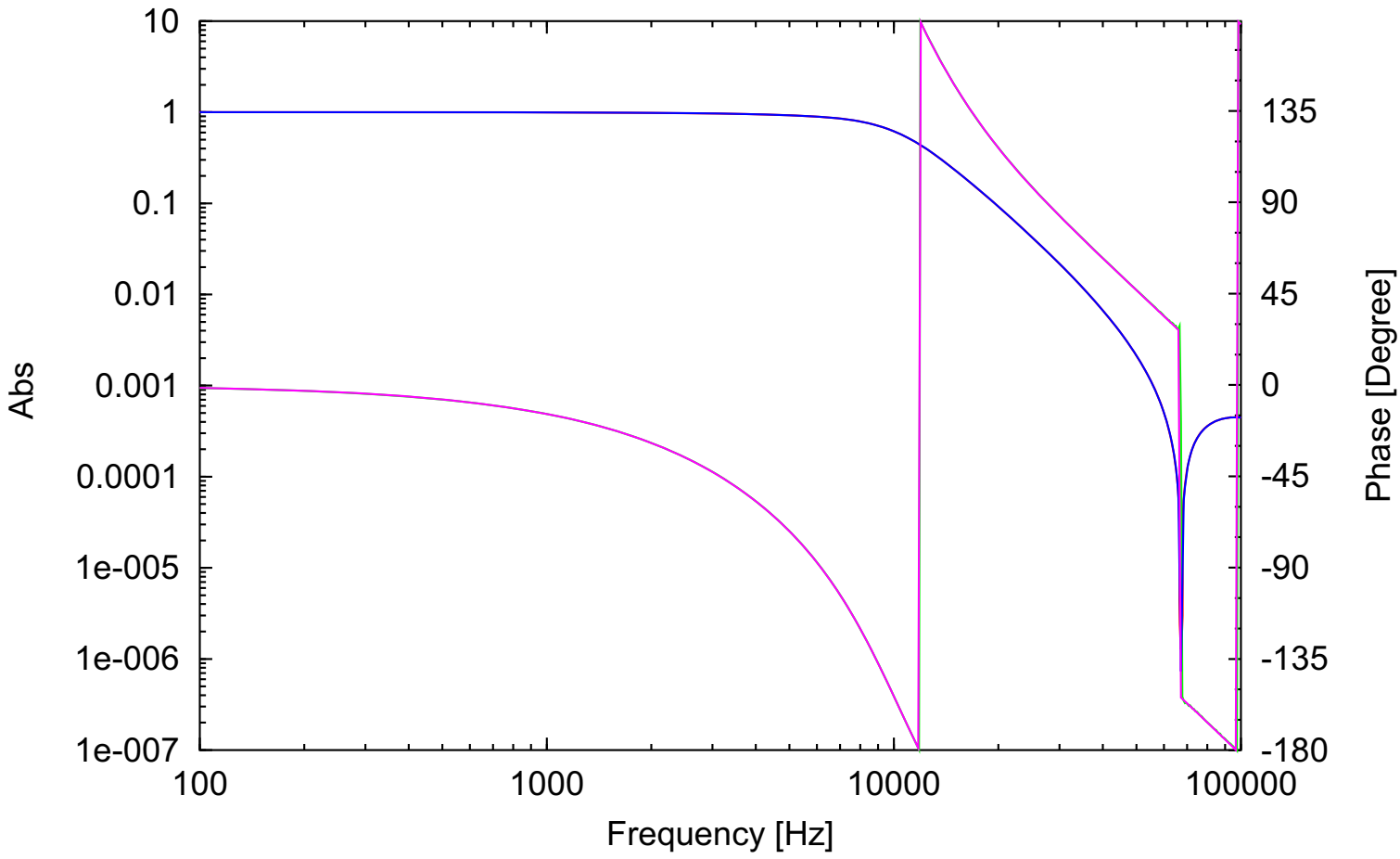
Transfer function



*T0=16/03/2009 20:35:00.020019

Avg=121

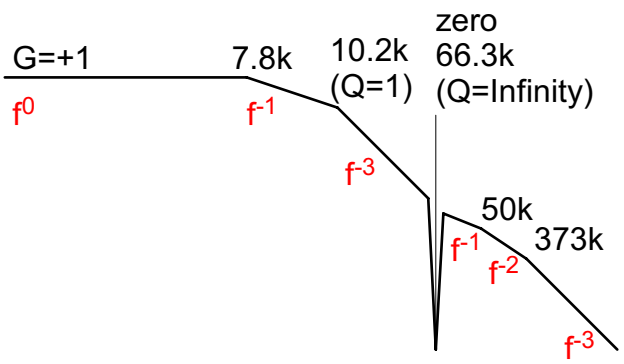
Transfer Function of AA/AI filter only



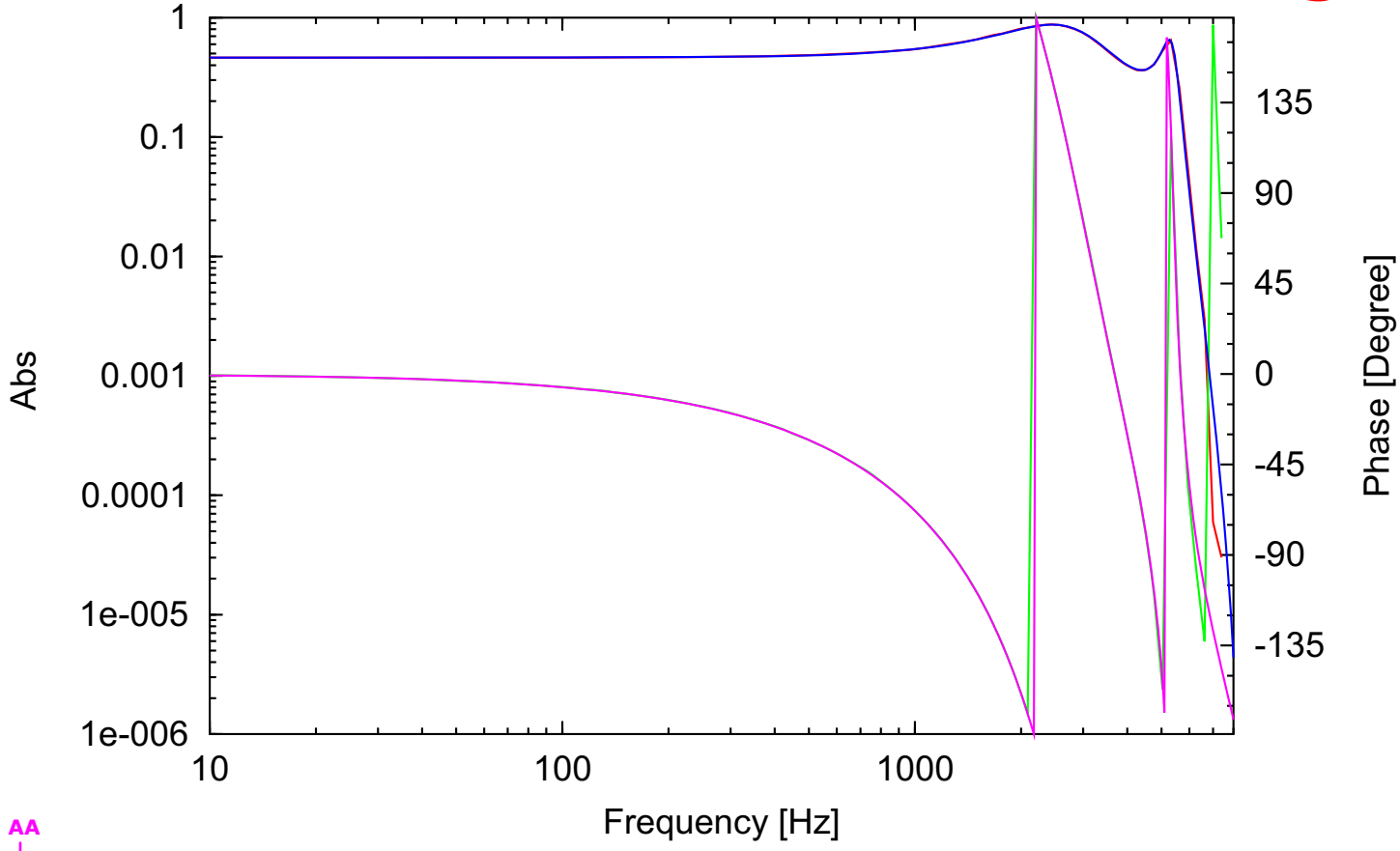
AI

```
pole 10.304784k 1.0159584  ### fitted (name = pole0)
pole 51.53319k  ### fitted (name = pole1)
pole 7.8430016k  ### fitted (name = pole2)
pole 346.96389k  ### fitted (name = pole3)
zero 67.205279k 68.668973M  ### fitted (name = zero0)
delay 660.9014n  ### fitted
factor 1.0001177  ### fitted

param pole0:f 1k 100k
param pole0:q 0.01 100
param pole1:f 1k 100k
param pole2:f 1k 100k
param pole3:f 1k 100M
param zero0:f 1k 1000k
param zero0:q 0.1 1G
param factor 1e-1 1e1
param delay 1e-20 1

fit TF_AI_090318.asc dbdeg rel
rewrite samebetter

#gnuterm cps

tfoutput abs:deg

freq log 100 100k 400 ### from data file
```

Calculated TF



Transfer Function of ADC/DAC through



Digital AA

Digital AI

```
pole 2.9126274k 1.1663357  ### fitted (name = pole0)
zero 8.7381935k 28.360464M  ### fitted (name = zero0)
pole 2.9126274k 1.1663357  ### fitted (name = pole1)
zero 8.7381935k 28.360464M  ### fitted (name = zero1)

pole 5.3800106k 8.3770775  ### fitted (name = pole2)
zero 8.7381935k 40.804087M  ### fitted (name = zero2)
pole 5.3800106k 8.3770775  ### fitted (name = pole3)
zero 8.7381935k 40.804087M  ### fitted (name = zero3)
```

20090316_delay_without_AI_AA.txt Abs
20090316_delay_without_AI_AA.txt Phase
TF Abs
TF Phase

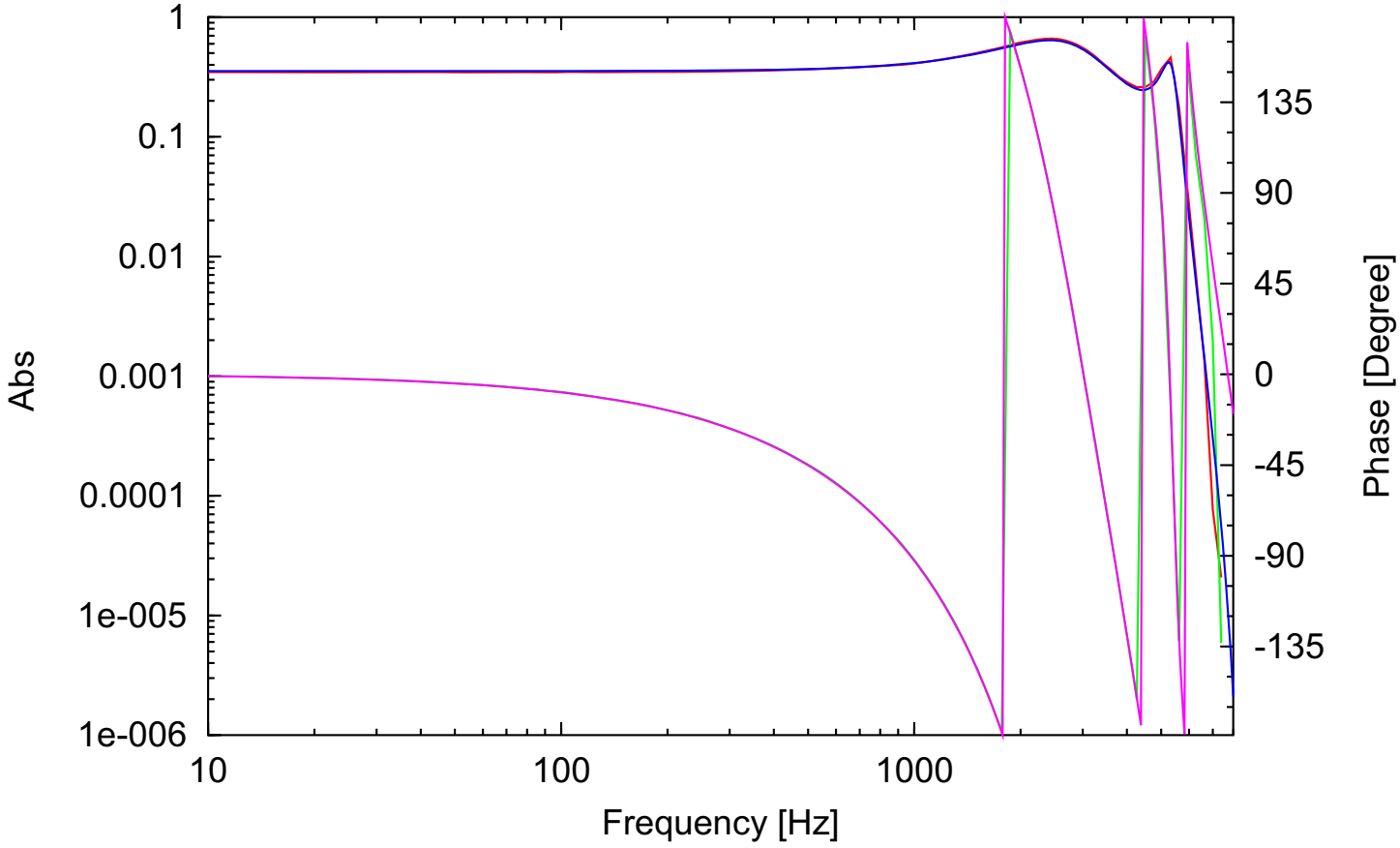
Through Delay

```
delay 79.165037u  ### fitted
factor 463.29699m  ### fitted
```

1/2 Gain
ADC +/-10V
=>digital=>
DAC +/- 5V

```
param pole0:f 1k 1G
sparam pole1:f

param pole0:q 0.01 10k
sparam pole1:q

param pole2:f 1k 1G
sparam pole3:f

param pole2:q 0.01 10k
sparam pole3:q

param zero0:f 1k 1G
sparam zero1:f

param zero0:q 0.01 1G
sparam zero1:q

param zero2:f 1k 1G
sparam zero3:f

param zero2:q 0.01 1G
sparam zero3:q

param factor .001 1000
param delay 1e-6 1

fit 20090316_delay_without_AI_AA.txt absdeg abs

rewrite samebetter
tfoutput abs:deg
freq log 10 8k 400 ### from data file
```

Transfer Function of AA->ADC->through->AI->DAC



AA

pole 10.304784k 1.0159584 ### fitted (name = pole0)
pole 51.53319k ### fitted (name = pole1)
pole 7.8430016k ### fitted (name = pole2)
pole 346.96389k ### fitted (name = pole3)
zero 67.205279k 68.668973M ### fitted (name = zero0)

AI

pole 10.304784k 1.0159584 ### fitted (name = pole0)
pole 51.53319k ### fitted (name = pole1)
pole 7.8430016k ### fitted (name = pole2)
pole 346.96389k ### fitted (name = pole3)
zero 67.205279k 68.668973M ### fitted (name = zero0)

Digital AA

pole 2.9126274k 1.1663357 ### fitted (name = pole0)
zero 8.7381935k 3.7254539M ### fitted (name = zero0)

Digital AI

pole 2.9126274k 1.1663357 ### fitted (name = pole1)
zero 8.7381935k 3.7254539M ### fitted (name = zero1)

pole 5.3800106k 8.3770775 ### fitted (name = pole2)
zero 8.7381935k 5.5411258M ### fitted (name = zero2)
pole 5.3800106k 8.3770775 ### fitted (name = pole3)
zero 8.7381935k 5.5411258M ### fitted (name = zero3)

Through Delay

delay 69.227u ### fitted
factor 354.31377m ### fitted

1/2 Gain by ADC +/-10V=>digital=>DAC +/- 5V
Additional loss???

param factor .001 1000
param delay 1e-6 1

fit 20090316_delay_with_AI_AA.txt absdeg abs

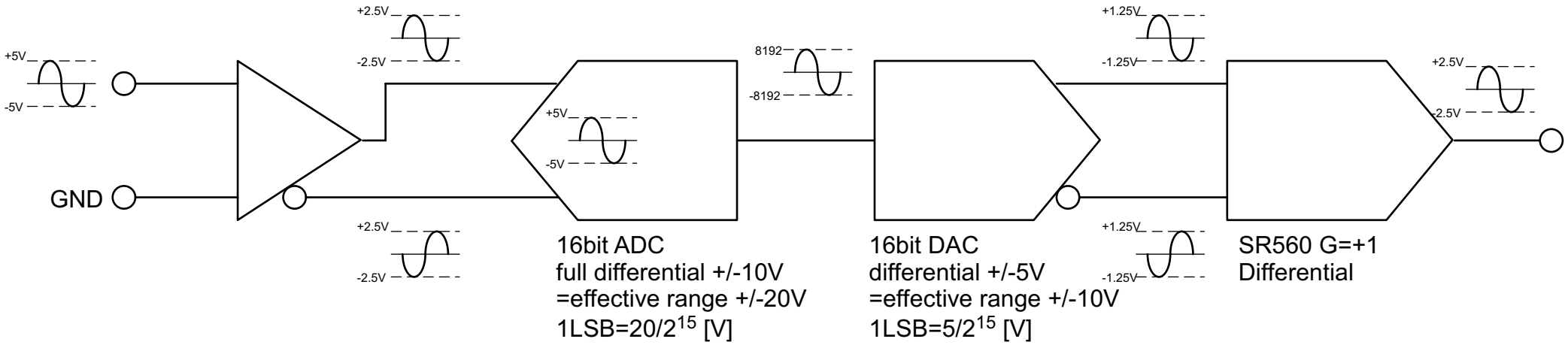
rewrite samebetter

gnuterm cps

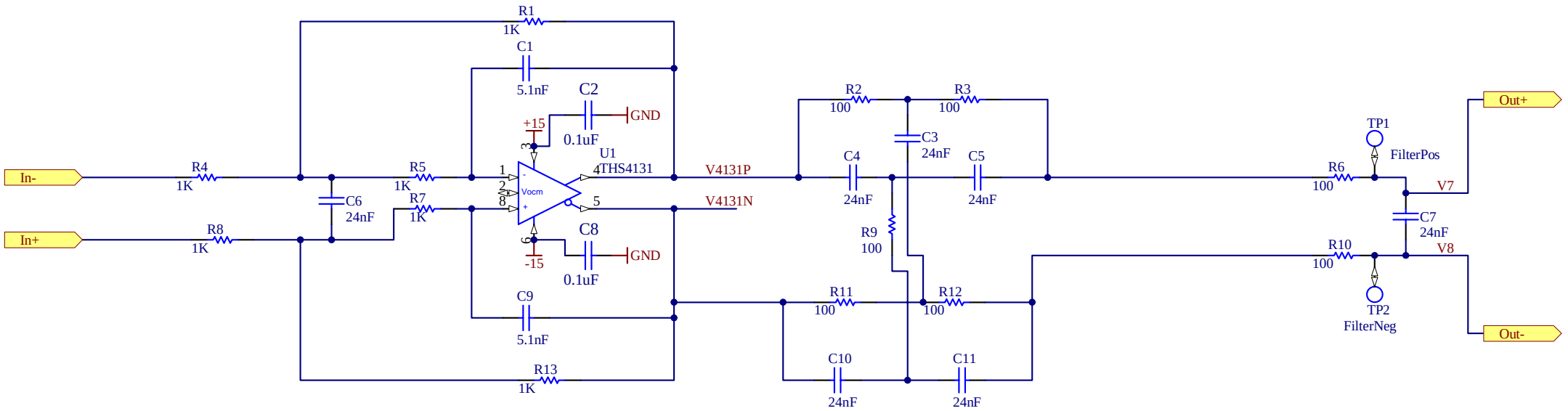
tfoutput abs:deg

freq log 10 8k 400 ### from data file

Explanation of effective gain of 1/2



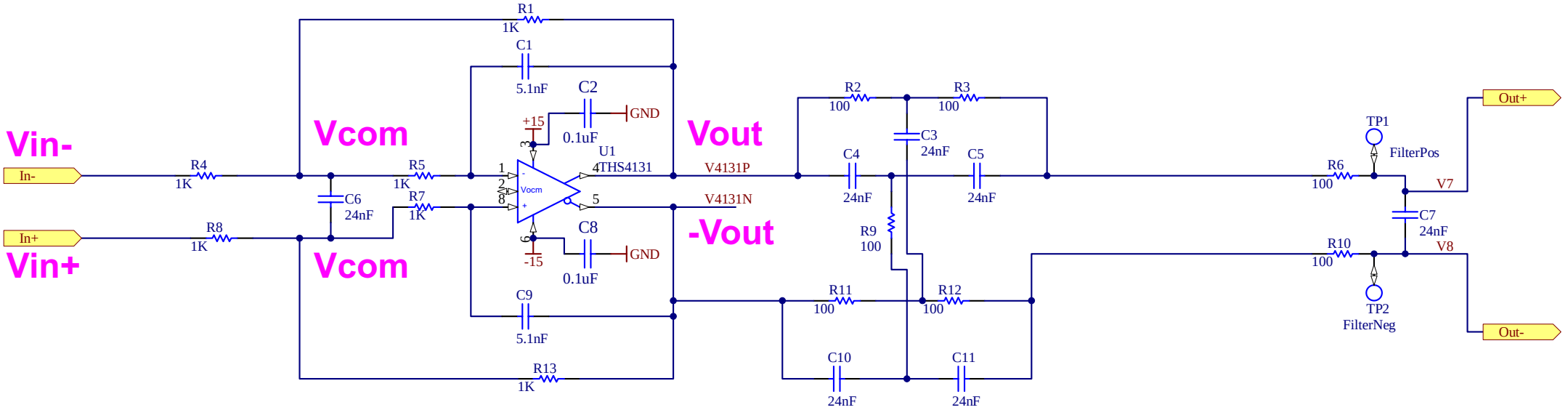
3rd order Butterworth, 10KHz, notch at 65536Hz



Title: ***AdL AA and AI Filter***			*LIGO Project* *California Institute of Technology* *Massachusetts Institute of Technology*	LIGO
Size: B	DCC Number: D070081	SCH / PCB Revision: 01	Engineer: J. Heefner	Date: 7/2/2007 Time: 9:26:29 AM
File: C:\Documents and Settings\jay\Desktop\AA_AI_Filter\AA_AI_Filter.SchDoc				Sheet 2 of 2

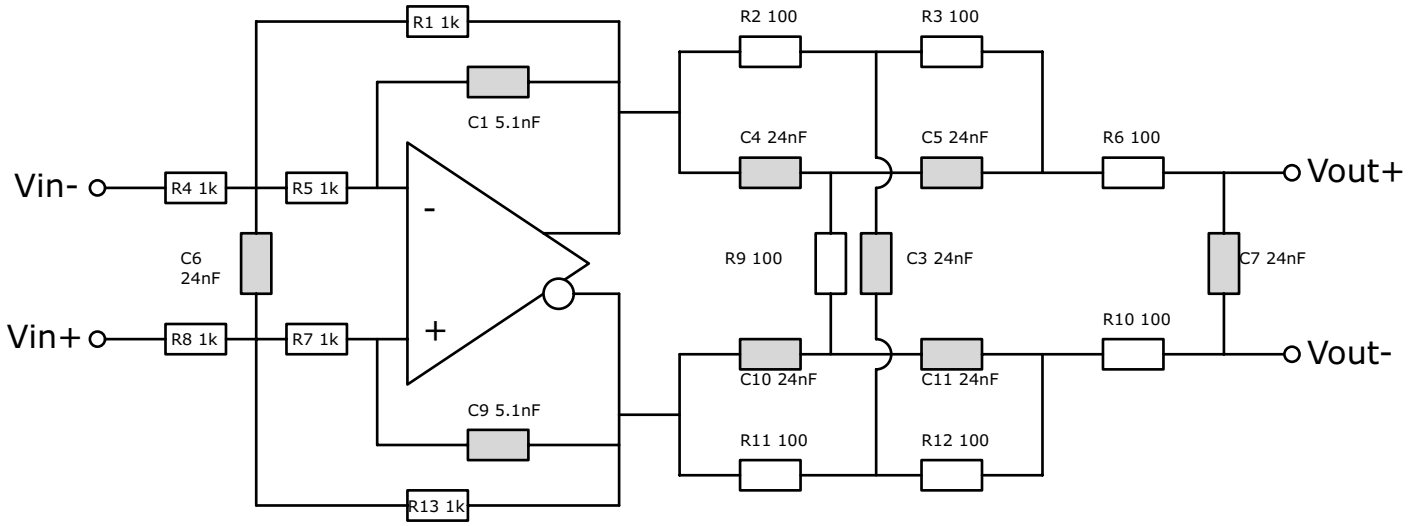
Principle of differential buffer

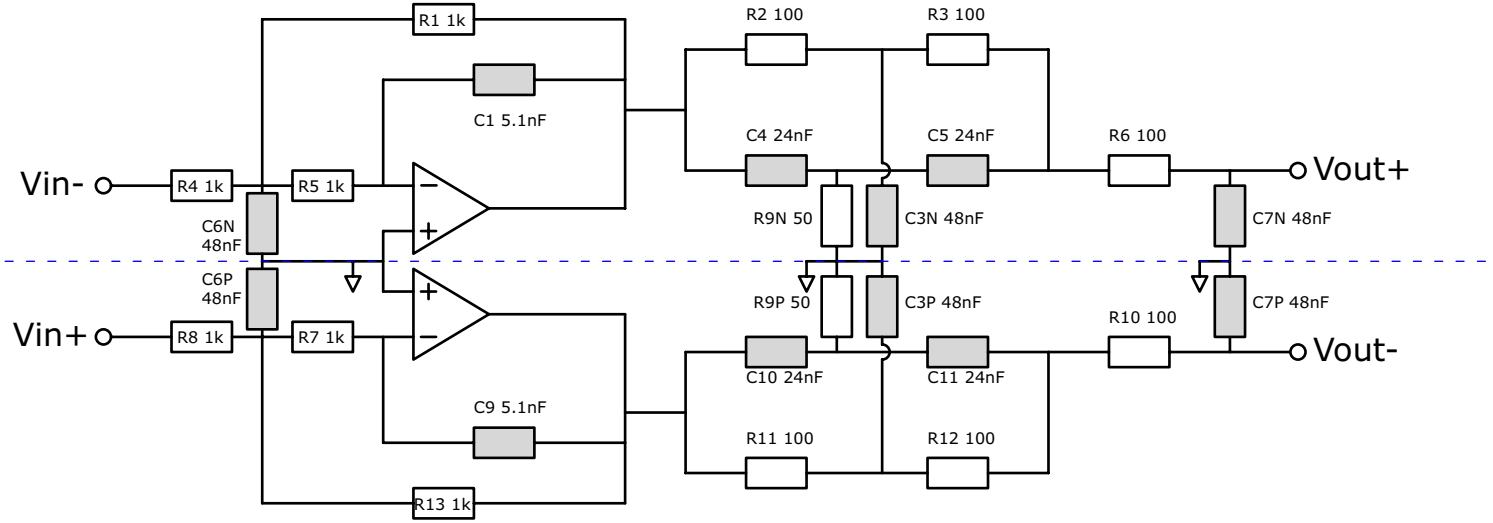
3rd order Butterworth, 10KHz, notch at 65536Hz



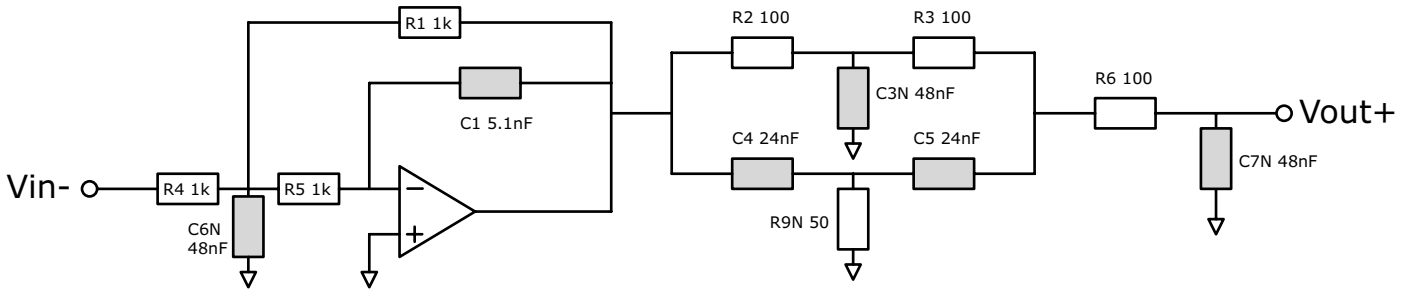
Vout-Vcom = Vcom - Vin-
-Vout-Vcom = Vcom - Vin+
=>
Vout = (Vin+ - Vin-)/2

Title: *AdL AA and AI Filter*			*LIGO Project* *California Institute of Technology* *Massachusetts Institute of Technology*	LIGO
Size: B	DCC Number: D070081	SCH / PCB Revision: 01	Engineer: J. Heefner	Date: 7/2/2007 Time: 9:26:29 AM
File: C:\Documents and Settings\jay\Desktop\AA_AI_Filter\AA_AI_Filter.SchDoc				Sheet 2 of 2

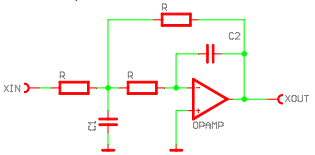




TF = ([Vout+] - [Vout-]) / ([Vin+] - [Vin-]) = - [Vout+] / [Vin-]



Multiple Feedback 2nd Order LPF



TF = -1/(C1 C2 R^2 s^2 + 3 C2 R s + 1)
ω0^2 = 1/(C1 C2 R^2)
Q = sqrt(C1/C2)/3

G=-1

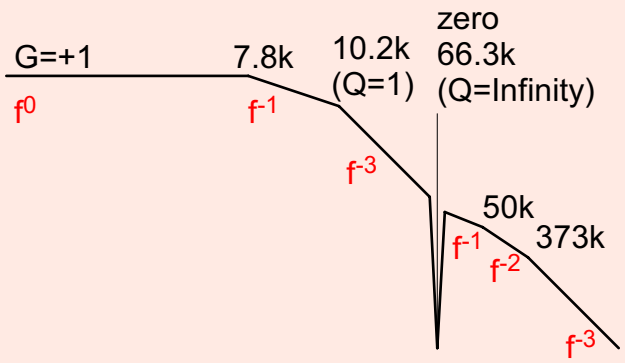
pole:
f0=10.17kHz
Q=1.023

G=1

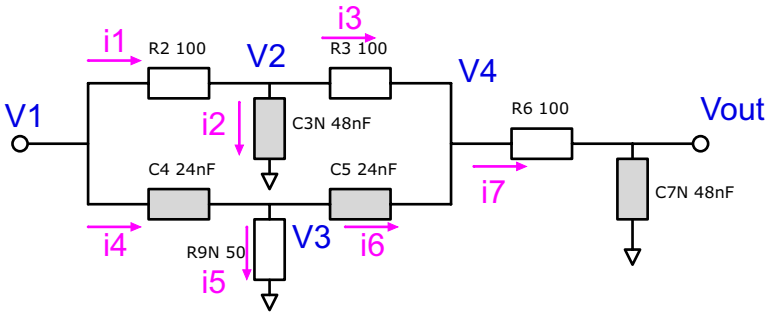
zero:
f0=66.314kHz
Q=Infinity

pole: (single x3)
f=7.8kHz
f=50kHz
f=373kHz

Total Performance of the whole circuit



c.f.
Ideal 3rd order butter worth
=>
f0=1 & f0=1 (single)
Q=1



11 unknown params:
V2,V3,V4,Vout, i1,i2,i3,i4,i5,i6,i7

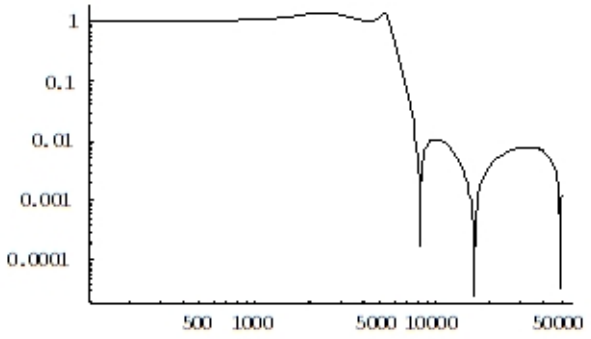
V1 - V2	=	i1 Z_{R2}
V2	=	i2 Z_{C3N}
V2 - V4	=	i3 Z_{R3}
i1	=	i2 + i3
V1 - V3	=	i4 Z_{C4}
V3	=	i5 Z_{R9N}
V3 - V4	=	i6 Z_{C5}
i4	=	i5 + i6
i7	=	i3 + i6
V4 - Vout	=	i7 Z_{R6}
Vout	=	i7 Z_{C7N}

Solve →

Zero:
f = 1/(2 pi R c) = 66.314kHz
(R=100Ω, c=24nF)

Pole:
three single poles
f=7.8kHz, f=50kHz, f=373kHz

```
In[37]:= Needs["Graphics`Graphics`"]

In[38]:= fs = 65536.;

In[39]:= zinv = Exp[-I 2 π f / fs];

In[40]:= coeff = {0.014805052402446,
          {-1.71662585474518, 0.78495484219691, -1.41346289716898, 0.99893884152400},
          {-1.68385964238855, 0.93734519457266, 0.00000127375260, 0.99819981588176}};

In[41]:= g = coeff[[1]]

Out[41]= 0.0148051

In[42]:= a1 = coeff[[2]][[1]];
         a2 = coeff[[2]][[2]];
         b1 = coeff[[2]][[3]];
         b2 = coeff[[2]][[4]];

In[46]:= da1 = a1 + 2;
         da2 = a2 - 1;
         db1 = b1 + 2;
         db2 = b2 - 1;

In[50]:= f0 = √(da1 + da2) / (2 π T) /. T → 1 / fs

Out[50]= 2726.48

In[51]:= Q = -√(da1 + da2) / da2

Out[51]= 1.21555

In[52]:= f0 = √(db1 + db2) / (2 π T) /. T → 1 / fs

Out[52]= 7980.95

In[53]:= Q = -√(db1 + db2) / db2

Out[53]= 721.065

In[54]:= H1 = (1 + b1 zinv + b2 zinv^2) / (1 + a1 zinv + a2 zinv^2);

In[55]:= a1 = coeff[[3]][[1]];
         a2 = coeff[[3]][[2]];
         b1 = coeff[[3]][[3]];
         b2 = coeff[[3]][[4]];

In[59]:= da1 = a1 + 2;
         da2 = a2 - 1;
         db1 = b1 + 2;
         db2 = b2 - 1;

In[63]:= f0 = √(da1 + da2) / (2 π T) /. T → 1 / fs

Out[63]= 5251.42

In[64]:= Q = -√(da1 + da2) / da2

Out[64]= 8.03567

In[65]:= f0 = √(db1 + db2) / (2 π T) /. T → 1 / fs

Out[65]= 14744.1

In[66]:= Q = -√(db1 + db2) / db2

Out[66]= 785.24

In[67]:= H2 = (1 + b1 zinv + b2 zinv^2) / (1 + a1 zinv + a2 zinv^2);

In[68]:= LogLogPlot[Abs[g H1 H2], {f, 10, 50000}]
```




```
Out[68]= - Graphics -
```