Vertical Clock Regulators for SXI

John Doty, Noqsi Aerospace ltd
April 6, 2010

A *Mathematica* notebook for investigating speed and stability of the SXI vertical clock regulators.

- **Introduction**

CCD vertical clocks have high capacitances and often require large clock swings. This requires high drive current. The duty cycle is low, so the voltage regulators for the clock drivers need to behave well over a large dynamic range. If energy conservation is important, the idle current must be kept to a minimum. Loads are transient, and the driver voltages crosstalk with the video output through the substrate, so rapid recovery from transients is important.

This notebook contains an analysis stability and transient recovery characteristics of the vertical clock regulators for the ASTRO-H SXI instrument http://astro-h.isas.jaxa.jp/si/03_e.html.

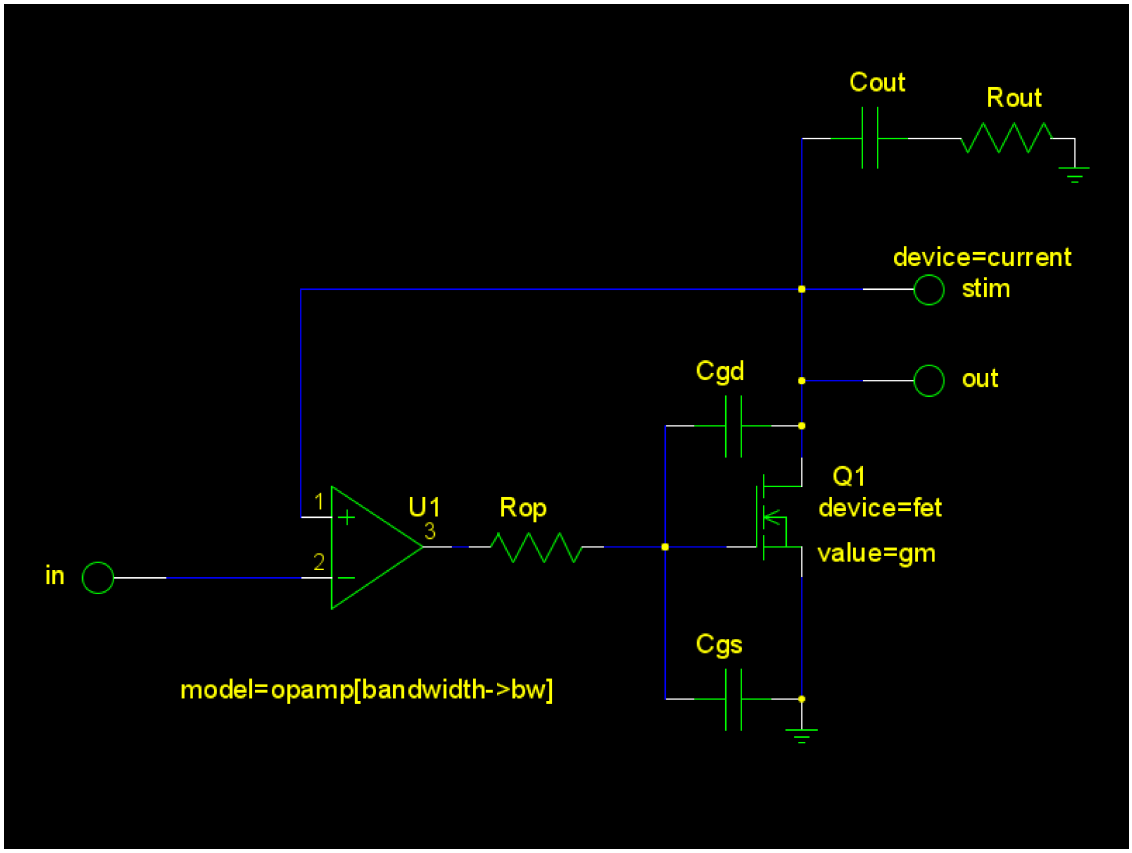
- **Calculations**

In[1]:=
```
here = NotebookDirectory[];
```

First, the negative regulator. U1 is an LT1078. Q1 is a composite of an LM195 current/temperature limiter and an IRLR014 MOSFET. Rop, Cgd, and Cgs are parasitics. Cout and Rout are discrete components. The detailed schematic may be seen in the ParallelReg subsection of the main Driver Board design document. The simplified model schematic is:

In[2]:=
```
Import[here <> "parlowmath.png"]
```

Out[2]=



Load the gEDAmath package. See www.noqsi.com/images/gEDAmath.nb.pdf for more details.

In[3]:=
```
<< (here <> "gEDAmath.m")
```

This analysis needs some additional models.
Basic linear triode model.

In[4]:=
```
fet[options___][refdes_] :=
 modeleqs[refdes, i["D"] == value (v["G"] - v["S"]), i["G"] == 0, i["S"] + i["D"] == 0] /.
   {options} /. {value → gm}
```

Current source model.

In[5]:=
```
current[options___][refdes_] := {i[refdes, "1"] == value} /. {options} /. value → iin
```

Import the "netlist".

In[6]:=
```
Clear[out]
<< (here <> "parlowmath.m")
```

in->out transfer function:

In[8]:=
```
tf = Simplify[transferFunction[in, out]] /. stim → 0
```

Out[8]=
$$(2\,\text{bw}\,\pi\,(\text{gm}-\text{cgd}\,s)\,(1+\text{cout}\,\text{rout}\,s))\Big/\Big(2\,\text{bw}\,\pi\,(\text{gm}-\text{cgd}\,s)\,(1+\text{cout}\,\text{rout}\,s)+s^2\,(\text{cout}+\text{cgs}\,\text{cout}\,\text{rop}\,s+\text{cgd}\,(1+\text{cout}\,\text{rop}\,s+\text{cout}\,\text{rout}\,s+\text{cgs}\,\text{rop}\,s\,(1+\text{cout}\,\text{rout}\,s)+\text{gm}\,(\text{rop}+\text{cout}\,\text{rop}\,\text{rout}\,s)))\Big)$$

Check for gain of 1 at DC.

```
In[9]:= tf /. s → 0
```

```
Out[9]= 1
```

Output impedance:

```
In[10]:= zf = Simplify[transferFunction[-stim, out]] /. in → 0
```

$$\text{Out[10]}= \left(s\,(1+\text{cgd rop}\,s+\text{cgs rop}\,s)\,(1+\text{cout rout}\,s)\right)\Big/ \left(2\,\text{bw}\,\pi\,(\text{gm}-\text{cgd}\,s)\,(1+\text{cout rout}\,s)+s^2\,(\text{cout}+\text{cgs cout rop}\,s+\text{cgd}\,(1+\text{cout rop}+\text{cout rout}+\text{cgs rop}\,s\,(1+\text{cout rout}\,s)+\text{gm}\,(\text{rop}+\text{cout rop rout}\,s)))\right)$$

Most of the analysis revolves about the denominator here, representing the "poles" or "natural frequencies" of the circuit. It is, however, useful to note that the numerator has a "zero" at **`-1/(cout rout)`**. We'll see below that there's a pole near there, so there's "pole-zero cancellation" going on here. There's also a zero at the origin, so the output impedance approaches zero at DC:

```
In[11]:= zf /. s → 0
```

```
Out[11]= 0
```

First, go for a simple symbolic analysis, ignoring parasitics.

```
In[12]:= sympar = {rop → 0, cgs → 0, cgd → 0}
```

```
Out[12]= {rop → 0, cgs → 0, cgd → 0}
```

```
In[13]:= zf /. sympar
```

$$\text{Out[13]}= \frac{s\,(1+\text{cout rout}\,s)}{\text{cout}\,s^2+2\,\text{bw gm}\,\pi\,(1+\text{cout rout}\,s)}$$

```
In[14]:= symf = Solve[Denominator[zf /. sympar] == 0, s]
```

$$\text{Out[14]}= \left\{\left\{s\to\frac{-\text{bw cout gm}\,\pi\,\text{rout}-\sqrt{\pi}\,\sqrt{-2\,\text{bw cout gm}+\text{bw}^2\,\text{cout}^2\,\text{gm}^2\,\pi\,\text{rout}^2}}{\text{cout}}\right\},\left\{s\to\frac{-\text{bw cout gm}\,\pi\,\text{rout}+\sqrt{\pi}\,\sqrt{-2\,\text{bw cout gm}+\text{bw}^2\,\text{cout}^2\,\text{gm}^2\,\pi\,\text{rout}^2}}{\text{cout}}\right\}\right\}$$

As expected, two "natural frequencies". Of interest is the critical damping condition where they are equal: in a circuit like this, that's often a good place to be for speed and stability.

```
In[15]:= symcrit = Solve[{(s /. symf[[1]]) == (s /. symf[[2]])}, rout]
```

$$\text{Out[15]}= \left\{\left\{\text{rout}\to-\frac{\sqrt{\frac{2}{\pi}}}{\sqrt{\text{bw}}\,\sqrt{\text{cout}}\,\sqrt{\text{gm}}}\right\},\left\{\text{rout}\to\frac{\sqrt{\frac{2}{\pi}}}{\sqrt{\text{bw}}\,\sqrt{\text{cout}}\,\sqrt{\text{gm}}}\right\}\right\}$$

Now move into the numeric domain.

Units are MHz, nF, kΩ, μs, mS.

bw->0.2 MHz	Opamp gain-bandwidthproduct
rop->1.0 kΩ	Opamp open loop output resistance
cgs->0.265 nF	FET gate-source capacitance
cout->22000	Use 22 μF

```
In[16]:= devpar = {bw → 0.2, rop → 1.0, cgs -> 0.265, cout → 22 000}

Out[16]= {bw → 0.2, rop → 1., cgs → 0.265, cout → 22 000}
```

Measured transconductance of the composite FET is 2000 mS at 1 A. For the high current case, assume 0.1 A. The LM195 is active, reducing the effective Cgd.

```
In[17]:= hicurr = {gm → 2000 Sqrt[0.1], cgd → 0.02}

Out[17]= {gm → 632.456, cgd → 0.02}
```

For low current, 1 mA, the LM195 is inactive, so raise Cgd to 50 pF.

```
In[18]:= locurr = {gm → 2000 Sqrt[0.001], cgd → 0.05}

Out[18]= {gm → 63.2456, cgd → 0.05}

In[19]:= symcrit[[2]] /. devpar /. hicurr
         symcrit[[2]] /. devpar /. locurr

Out[19]= {rout → 0.000478298}

Out[20]= {rout → 0.00151251}
```

Given lists of parameters, extract "natural frequencies".

```
In[21]:= natfreqs[p___] := s /. Solve[Denominator[zf /. devpar /. Flatten[{p}]] == 0, s]
```

Stability index: range is -1 to 1. 1 is the most stable. Negative numbers are unstable. As a rough rule of thumb, I like to see numbers >0.5 here.

```
In[22]:= stability[z_] := -Re[z] / Abs[z]
         SetAttributes[stability, Listable]
```

Start with Rout near the critical point identified above.

```
In[24]:= natfreqs[hicurr, rout → 0.0015]

Out[24]= {-38 236.2, -1.63014 - 1.07845 i, -1.63014 + 1.07845 i, -0.0311075}
```

How do we interpret this? The first solution is far outside our bandwidth, and may be ignored. It's only telling us that we cannot in practice separate the effects of Cgs and Cgd. The last is approximately **-1/(cout rout)**, representing recharge of Cout at approximately constant voltage. This suggests that the output voltage will recover quickly to a transient, so the recovery of Cout's charge is simply controlled by Rout. Another way to say this is that the zero at **-1/(cout rout)**, noted above, cancels this pole. We will therefore largely ignore it.

```
In[25]:= stability[%]

Out[25]= {1., 0.834006, 0.834006, 1.}
```

Stability is good. Now try the low current case.

In[26]:=
```
natfreqs[locurr, rout → 0.0015]
```

Out[26]=
```
{-16 088.4, -3.00469, -0.0648215, -0.0580068}
```

Slower (look at the third natural frequency), but stable. It's to be expected that the circuit slows down as it settles into the low current case.

We can take the third natural frequency as an indicator of the speed. The interesting case for speed is the high current case, because the current will only be low if the circuit is already near equilibrium.

In[27]:=
```
speed[r_] := -Re[natfreqs[hicurr, rout → r]][[3]]
```

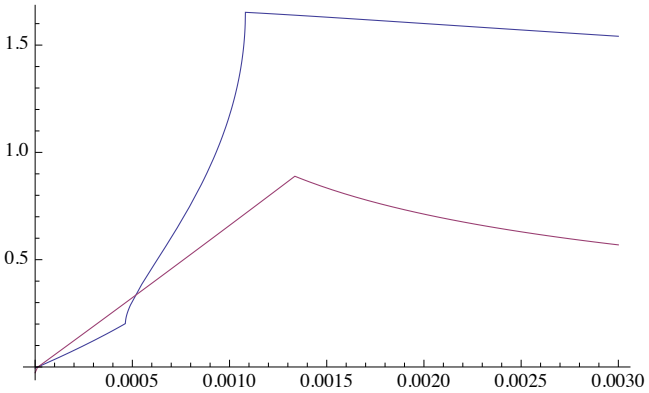
For stability, we're interested in every natural frequency in both high and low current cases. Any instability will be trouble.

In[28]:=
```
stab[r_] :=
 Min @@ Flatten[{stability[natfreqs[hicurr, rout → r]], stability[natfreqs[locurr, rout → r]]}]
```

Investigate speed and stability as a function of Rout.

In[29]:=
```
Plot[{speed[r], stab[r]}, {r, 0, 0.003}]
```

Out[29]=



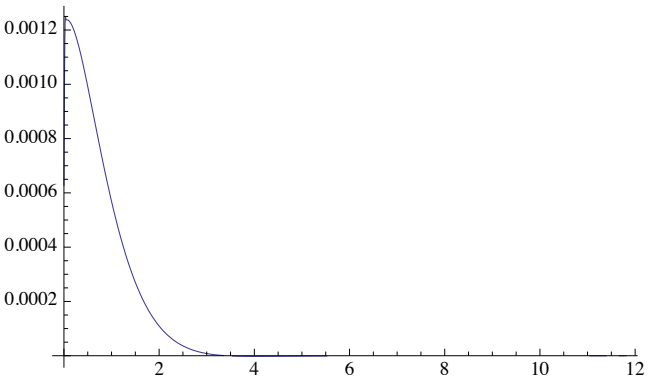
The model with parasitics is telling us pretty much the same story as the simpler analysis. The optimum Rout is near ~1.3Ω.

Plot the response to a change in load using the stepResponse function from the gEDAmath package. Its numerical inverse Laplace transform adds an unknown constant of integration, so take a sample from late in the curve to be the baseline.

In[30]:=
```
pstep[tf_, d_] := With[{sr = stepResponse[tf, 2 d, d / 1000]},
   Plot[Evaluate[sr - (sr /. t → d)], {t, 0, d}, PlotRange → All]]
```

In[31]:=
```
pstep[zf /. devpar /. hicurr /. rout → 0.0013, 16]
```

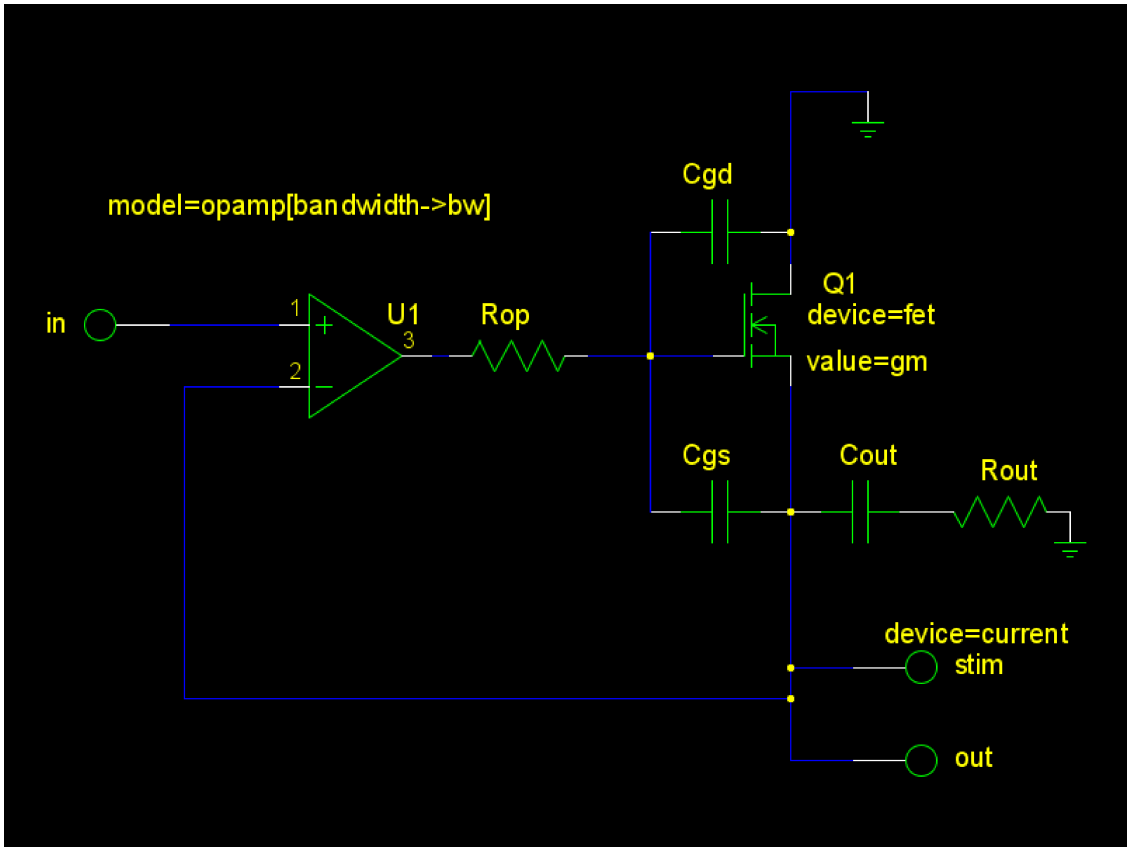
Out[31]=



Give the positive regulator the same treatment.

In[32]:=
```
Import[here <> "parhimath.png"]
```

Out[32]=



Import the "netlist".

In[33]:=
```
Clear[out]
<< (here <> "parhimath.m")
```

In[35]:=
```
tf = Simplify[transferFunction[in, out]] /. stim → 0
```

Out[35]=
```
(2 bw π (gm + cgs s) (1 + cout rout s)) /
 (2 bw π (gm + cgs s) (1 + cout rout s) + s (gm (1 + cgd rop s) (1 + cout rout s) +
       s (cout + cgd cout rop s + cgs (1 + cout (rop + rout) s + cgd rop s (1 + cout rout s)))))
```

Check for gain of 1 at DC.

In[36]:=
```
tf /. s → 0
```

Out[36]=
```
1
```

Output impedance:

In[37]:=
```
zf = Simplify[transferFunction[-stim, out]] /. in → 0
```

Out[37]=
```
(s (1 + cgd rop s + cgs rop s) (1 + cout rout s)) /
 (2 bw π (gm + cgs s) (1 + cout rout s) + s (gm (1 + cgd rop s) (1 + cout rout s) +
       s (cout + cgd cout rop s + cgs (1 + cout (rop + rout) s + cgd rop s (1 + cout rout s)))))
```

Output impedance approaches zero at DC:

In[38]:=
```
zf /. s → 0
```

Out[38]=
```
0
```

First, go for a simple symbolic analysis, ignoring parasitics.

In[39]:=
```
sympar = {rop → 0, cgs → 0, cgd → 0}
```

Out[39]=
```
{rop → 0, cgs → 0, cgd → 0}
```

In[40]:=
```
zf /. sympar
```

Out[40]=
$$\frac{s\,(1+\text{cout rout } s)}{2\,\text{bw gm}\,\pi\,(1+\text{cout rout } s)+s\,(\text{cout } s+\text{gm}\,(1+\text{cout rout } s))}$$

Once again, zeros at the origin and **-1/(cout rout)**. The same pole-zero cancellation seen above will occur here.

In[41]:=
```
symf = Solve[Denominator[zf /. sympar] == 0, s]
```

Out[41]=
$$\left\{\left\{s \to \left(-\text{gm}-2\,\text{bw cout gm}\,\pi\,\text{rout}-\sqrt{-8\,\text{bw gm}\,\pi\,(\text{cout}+\text{cout gm rout})+(\text{gm}+2\,\text{bw cout gm}\,\pi\,\text{rout})^2}\right)\Big/\left(2\,(\text{cout}+\text{cout gm rout})\right)\right\},\right.$$
$$\left.\left\{s \to \left(-\text{gm}-2\,\text{bw cout gm}\,\pi\,\text{rout}+\sqrt{-8\,\text{bw gm}\,\pi\,(\text{cout}+\text{cout gm rout})+(\text{gm}+2\,\text{bw cout gm}\,\pi\,\text{rout})^2}\right)\Big/\left(2\,(\text{cout}+\text{cout gm rout})\right)\right\}\right\}$$

In[42]:=
```
symcrit = Solve[(s /. symf[[1]]) == (s /. symf[[2]]), rout]
```

Out[42]=
$$\left\{\left\{\text{rout} \to \frac{\sqrt{\text{gm}}-2\sqrt{\text{bw}}\,\sqrt{\text{cout}}\,\sqrt{2\,\pi}}{2\,\text{bw cout}\,\sqrt{\text{gm}}\,\pi}\right\},\left\{\text{rout} \to \frac{\sqrt{\text{gm}}+2\sqrt{\text{bw}}\,\sqrt{\text{cout}}\,\sqrt{2\,\pi}}{2\,\text{bw cout}\,\sqrt{\text{gm}}\,\pi}\right\}\right\}$$

In[43]:=
```
symcrit[[2]] /. devpar /. hicurr
symcrit[[2]] /. devpar /. locurr
```

Out[43]=
```
{rout → 0.00051447}
```

Out[44]=
```
{rout → 0.00154868}
```

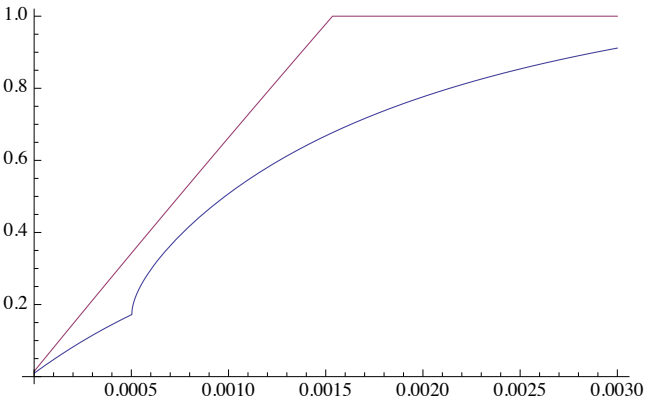
Very similar numbers to the negative regulator.

In[45]:=
```
natfreqs[hicurr, rout → 0.0015]
```

Out[45]=
```
{-38279.3, -5.70759, -0.668138, -0.0311291}
```

In[46]:=
```
Plot[{speed[r], stab[r]}, {r, 0, 0.003}]
```

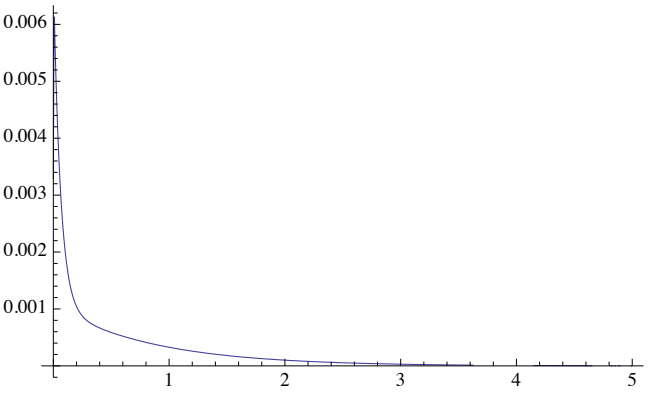
Out[46]=



This makes it appear that it's speedier to have high Rout, but it's a little misleading. With Rout of 10Ω, a load change makes a big voltage transient because Cout can't contribute much current:

In[47]:=
```
hir = pstep[zf /. devpar /. hicurr /. rout → 0.01, 5]
```

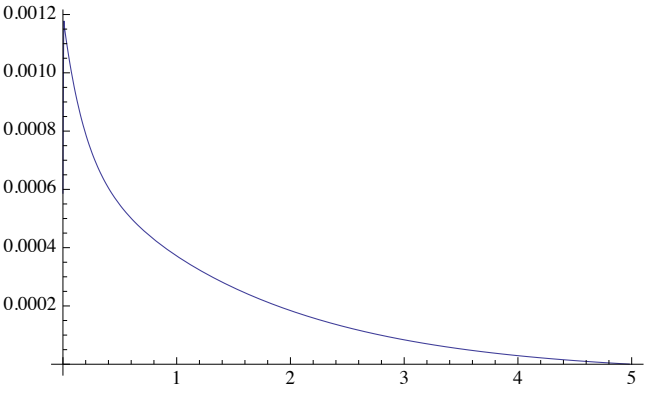
Out[47]=



With Rout of 1.3Ω, the transient is much smaller:

In[48]:=
```
lor = pstep[zf /. devpar /. hicurr /. rout → 0.0013, 5]
```

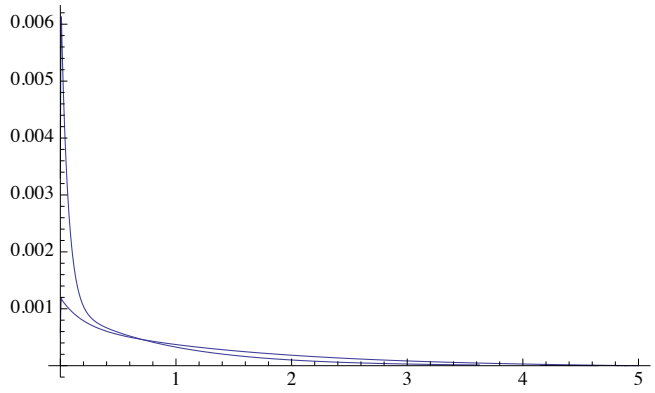
Out[48]=



Comparing the two, the settling tails are almost the same, so the lower resistance is better.

In[49]:=
```
Show[hir, lor]
```

Out[49]=



- **Conclusions**

The SXI vertical clock rate is 32 kHz, so output settling in a few microseconds represents excellent performance. The fact that the lowest frequency pole is approximately cancelled by a zero indicates that the circuit itself takes longer to settle than its output does. This means that the circuit is moderating the current surges from the power rails on clock transitions, a good thing for it to do. The components need little idle current, so these circuits are reasonably efficient.

I think this notebook demonstrates the power of a mixture of symbolic and numerical linear circuit analysis. The symbolic analysis yields insights difficult to perceive in a purely numerical environment like SPICE, and also leads to very fast and comprehensible linear numerical models.