LTspice: Worst-Case Circuit Analysis with Minimal Simulations Runs

by **Gabino Alonso** and **Joseph Spencer**

When designing a circuit in LTspice, you may wish to assess the impact of component tolerances. For example, the gain error introduced by non-ideal resistors in an op amp circuit. This article illustrates a method that reduces the number of simulations needed, and as a result speeds your time to results.

Varying a Parameter

LTSpice provides several ways to vary the value of a parameter. Some of these are:

- *.step param*; A parameter sweep of a user-defined variable
- *gauss(x)*; A random number from Gaussian distribution with a sigma of x
- *flat(x)*; A random number between -x and x with uniform distribution
- *mc(x,y)*; A random number between *x*(1+y)* and *x*(1-y)* with uniform distribution.

These functions are very useful, especially if we want to look at results in terms of a distribution. But if we only want to look at worst case conditions, they may not be the quickest way to get a result. Using *gauss(x), flat(x)* and *mc(x,y)* for example, will require a simulation to run for a statistically significant number of times. From there, a distribution can be looked at and worst case values calculated in terms of standard deviations. However, for a worst case analysis we would prefer not to use a distribution approach, instead the maximum deviation from the nominal value of each component are used in the calculations.

Running Minimal Simulations

Let's say that we want to look at the worst-case impact of a *R1 = 22.5kΩ* resistor with a *1%* tolerance. In this case, we really only want to run the simulation with *R1 = 22.5kΩ * (1 - 0.01)* and *22.5kΩ * (1 + 0.01)*. A third run with an ideal *22.5kΩ* resistor would also be handy to have.

```
.step param R1 list 22.5k*(1-.01) 22.5k*(1+.01) 22.5k
```

If we were just varying one resistor value, the '.step param' method would work very well. But what if we have more? The classic difference amplifier has 4 resistors.

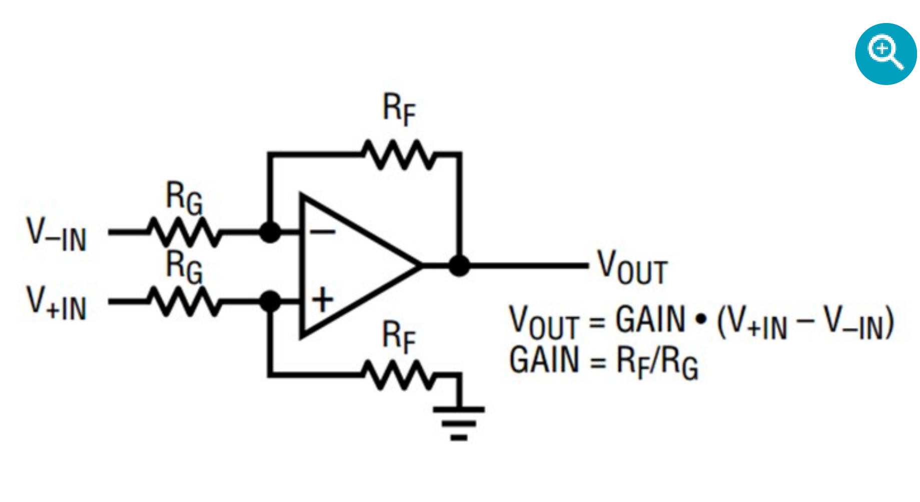


Figure 1. Difference Amplifier Configuration (From the LT1997-3 datasheet)

If a discrete difference amplifier were to be designed, each of these would have some tolerance (e.g. 1% or 5%).

For an example, let's take the front page application shown in the LT1997-3 datasheet and implement it in LTspice with a discrete LT6015 op-amp and some non-ideal resistors.

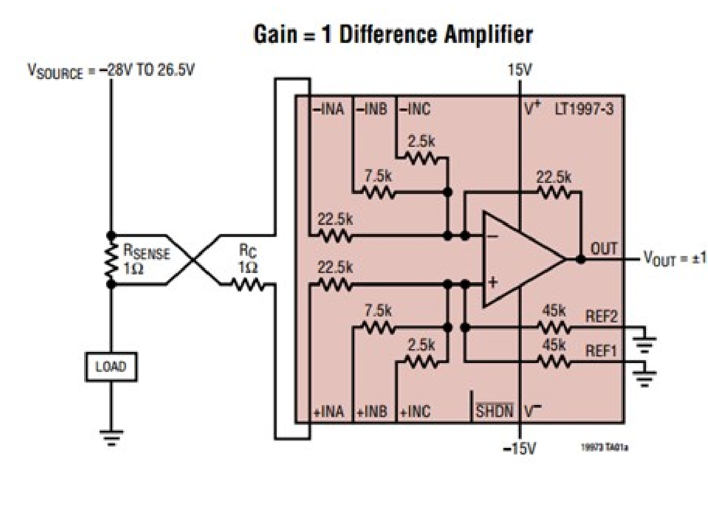


Figure 2. LT1997-3 Front Page Application Example

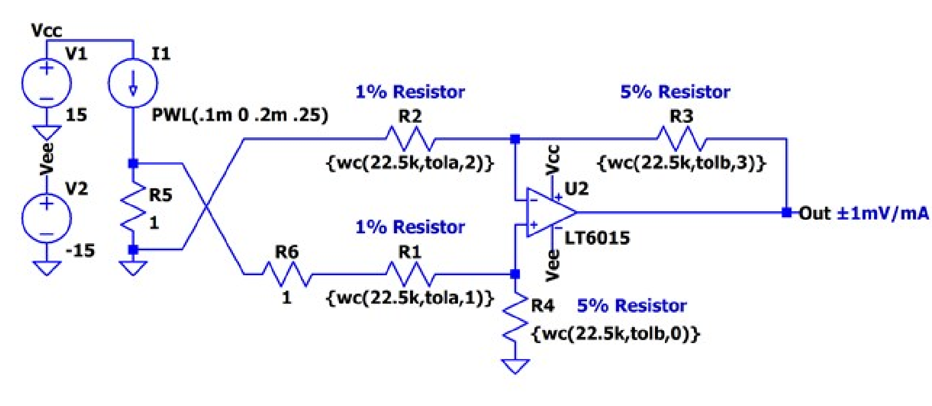


Figure 3. LT6015 Difference Amplifier with Non-Ideal Resistors

Notice the values of resistors R1, R2, R3 and R4 are replaced by a function call *wc(nominalvalue, tolerance, index)* which is defined in the simulation by a *.func* statement:

```
.func wc(nom,tol,index) if(run==numruns,nom,if(binary(run,index),nom*(1+tol),nom*(1-tol)))
```

This function in conjunction with the *binary(run, index)* function below vary the parameter for each component between its maximum value and minimum values and for the last run the nominal value.

```
.func binary(run,index) floor(run/(2**index))-2*floor(run/(2**(index+1)))
```

The *binary* function toggles each *index*'ed componment in the simulation so that all possible combination of *nom*(1+tol)* and *nom*(1-tol)* are simulated. Note that the index of components should start with *0*. The following table highlights the operation of the *binary()* function with results to each *index* and *run*, where 1 represents *nom*(1+tol)* and 0 represents *nom*(1-tol)*.

Run	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Index 0 (R4)	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Index 1 (R1)	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Index 2 (R2)	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Index 3 (R3)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

The number of runs is determined buy $2^N +1$, where N equals the number of indexed components, to cover all the max and min combinations of the device plus the nominal. In our case we need *17* simulations run and we can define this using the *.step* command and the *.param* statments:

```
.step param run 0 16 1
.param numruns=16
```

Lastly, we need to define the tola and tolb for the simulation via *.param* statments:

```
.param tola=.01
.param tolb=.05
```

You can find more information about the *.func, .step,* and *.param* statments in the help (F1) and under the *.param* section details about the *if(x,y,z)* and *floor(x)* functions.

Plotting the .Step'ed Results

If the transient analysis simulation is run, see WorstCase_LT6015.asc file, we can observe our results. For a 250mA test current, we expect the Vout net to settle to 250mV. But now with our *wc()* function, we get a spread from 235mV to 265mV.

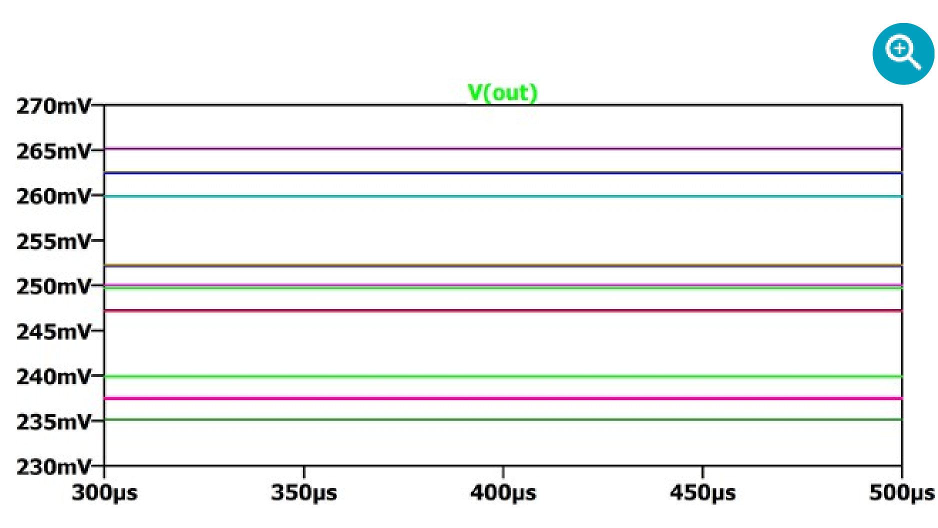


Figure 4. Difference Amplifier Transient Analysis

Plotting the .Step'ed .Meas Statement

At this point we could zoom in and look at the peak to peak spread. But let's take a lesson from another LTspice blog:

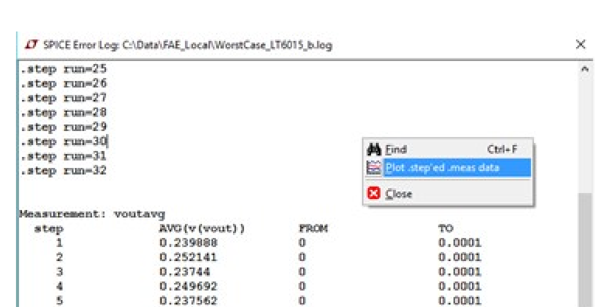
Plotting a Parameter Against Something Other Than Time (e.g. Resistance)

This blog covers how to run a simulation several times and plot a parameter against something other than time. In this case, we want to plot the V(out) vs. simulation run index. See WorstCase_LT6015_meas.asc file.

In this simulation we have add a *.meas* statment to calculated the average voltage of the output.

```
.meas VoutAvg avg V(out)
```

To plot the V(out) vs. run parameter we can view the *SPICE Error Log* (Ctrl-L), right click and select *Plot .step'ed .meas data*.



```
.step run=25
.step run=26
.step run=27
.step run=28
.step run=29
.step run=30
.step run=31
.step run=32

Measurement: voutavg
  step          AVG(v(vout))      FROM          TO
     1          0.239888          0             0.0001
     2          0.252141          0             0.0001
     3          0.23744           0             0.0001
     4          0.249692          0             0.0001
     5          0.237562          0             0.0001
     6          0.249696          0             0.0001
     7          0.235138          0             0.0001
     8          0.247271          0             0.0001
     9          0.252256          0             0.0001
    10          0.26514           0             0.0001
    11          0.249681          0             0.0001
    12          0.262565          0             0.0001
    13          0.249685          0             0.0001
    14          0.262438          0             0.0001
    15          0.247137          0             0.0001
    16          0.25989           0             0.0001
    17          0.239888          0             0.0001
```

Figure 5. Spice Error Log Interface

The plot results of our *.step*'ed *.meas* data.

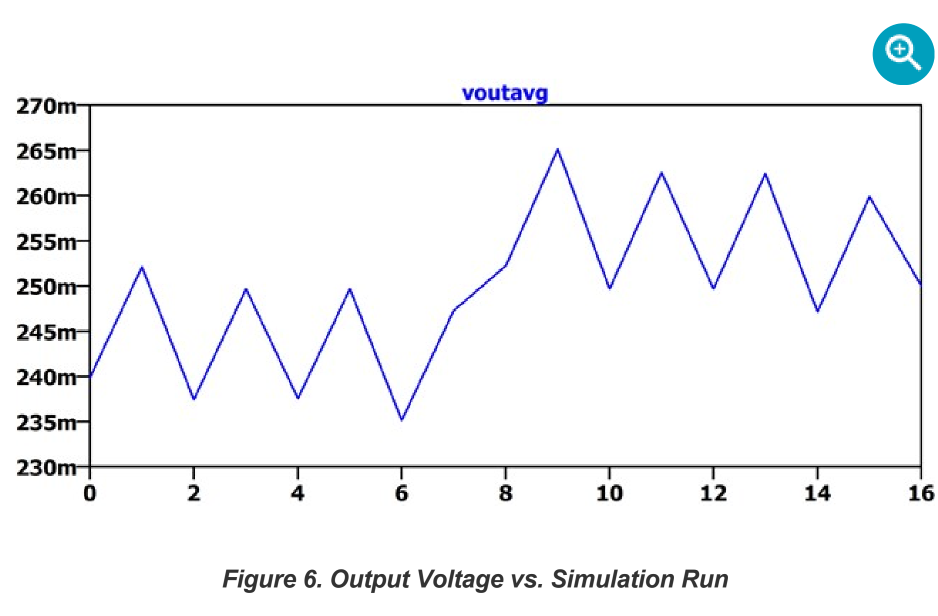


Figure 6. Output Voltage vs. Simulation Run

The trace hows us that the results vary from a maximum worse case of 265mV (run 9) to a minimum worse case of 235mV (run 6) or roughly a ±6% error. This makes some intuitive sense since we had both 1% and 5% resistors in this example. The last run (16) shows the ideal result (250mV) which is ideal resistors. Recall LTspice graphs the results from the *.meas* statment as a piece wise linear graph.

Another faster approach to this particular circuit is to use the *.op* simulation (instead of the .trans) to perform a DC operating point solution which will plot the results of our stepped.*meas* data directly.

The Value of Matched Resistors

When designing a difference amplifier not only is the appropriate op-amp needed, but equally as important are the matching of the resistors. The following references do an excellent job of explaining this topic (and associated math) in detail:

- LT5400 (Quad Matched Resistor Network) Datasheet
- Design Note DN1023
- Design Note DN502

However, you can neither achieve good Common Mode Rejection Ratio (CMRR) or Gain Error without appropriately matched resistors.

Linear Technology, now part of Analog Devices, has a number of precision amplifier products which also include matched resistors. A recently released example of this is the LT1997-3 - Precision, Wide Voltage Range Gain Selectable Amplifier. Two key specifications are:

- 91dB Minimum DC CMRR (Gain = 1)
- 0.006% (60ppm) Maximum Gain Error (Gain = 1)

These specifications are really quite excellent. According to DN1023, CMRR due only to 1% resistors (with an ideal op-amp) will limit your CMRR to 34dB. And of course, the gain error is orders of magnitude worse than what the LT1997-3 achieves.

Summary

Using the method outlined above, a simple worst-case analysis can be run at the min/max value of several parameters. In this example we looked at the impact of resistor tolerances in a classical difference amplifier, and the value of the matched resistors in the LT1997-3 are illustrated.

Authors

Gabino Alonso

Joseph Spencer

Related Products

LT5400
Quad Matched Resistor Network

LT6015
3.2MHz, 0.8V/µs Low Power, Over-The-Top Precision Op Amps

LT6016
Dual 3.2MHz, 0.8V/µs Low Power, Over-The-Top Precision Op Amp

LT6017
Quad 3.2MHz, 0.8V/µs Low Power, Over-The-Top Precision Op Amp

LT1997-3
Precision, Wide Voltage Range Gain Selectable Amplifier

Related Resources

LT6015 Difference Amplifier with Non-Ideal Resistors and .Meas Statment ASC
MAY 2017

LT6015 Difference Amplifier with Non-Ideal Resistors ASC
MAY 2017

SOCIAL

QUICK LINKS
About ADI | Alliances
Analog Dialogue | Careers
Contact us | Investor Relations
News Room | Quality & Reliability
Sales & Distribution

LANGUAGES
English
简体中文
日本語
Русский

NEWSLETTERS
Interested in the latest news and articles about ADI products, design tools, training and events? Choose from one of our 12 newsletters that match your product area of interest, delivered monthly or quarterly to your inbox.
Sign Up

Feedback