SPICE 3: Parameters, Fourier and FFT Analyses

Chris Winstead

February 12, 2015

Preparing for the Exercises

In this session, we will simulate measuring the 741's offset voltage, open-loop gain and slew rate. Create a new directory for this session:

```
3410/
└─spice/
   ├─lab1/
   ├─lab2/
   └─lab3/
```

Obtain the Example Files

In this lab, you will download a collection of files containing the exercises: `exercise1.sp`, `exercise2.sp`, and `exercise3.sp`. Download these files and place them in these locations:

```
3410/
└── spice/
    ├── lab_parts.md
    ├── lab1/
    │   └── [files from lab 1]
    ├── lab2/
    │   └── [files from lab 2]
    └── lab3/
        ├── exercise1.sp
        ├── exercise2.sp
        └── exercise3.sp
```

Study Exercise 1

Navigate to your lab3 directory and open the exercise1.sp file. The circuit described in this file corresponds to Fig. 1 of the pre-lab for Lab 3. The circuit description should be familiar to you. Near the top of this file you should notice one new feature:

```
* Define a numerical constant for
* estimating VOFS:
.csparam Go=*** ENTER YOUR VALUE
```

The .csparam statement defines a numerical constant that can be used in the file's control block. This can be useful for automating calculations. In this case, the constant is named "Go" and should correspond to the numerical result from Exercise 1 in the pre-lab. Type your numerical result where it says "*** ENTER YOUR VALUE".

Using CSPARAMs

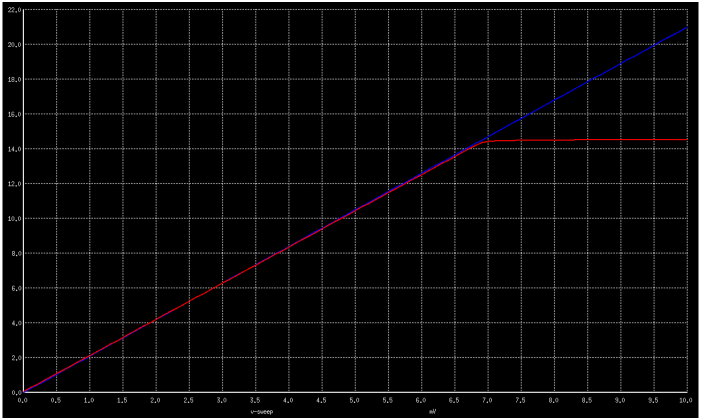
The `Go` parameter is used in the control block:

```
* Control Commands:
.control
dc Vofs 0 10m 100u

plot v(nout) $&Go*v(n1)

.endc
```

In this circuit, V_{OFS} is applied at the op amp's non-inverting input, at node n1. If your pre-lab analysis is correct, then the plot command should produce two identical traces, since we expect $V_{\mathrm{OUT}} = G_o V_{\mathrm{OFS}}$. Run the simulation and compare the results. Do you see anything unexpected?

VOFS Measurement

You should see that the observed V_{OUT} saturates when $V_{\mathrm{OFS}} > 6.8\,\mathrm{mV}$. For this reason, our particular choice of resistor values will work for measuring $|V_{\mathrm{OFS}}| < 6.8\,\mathrm{mV}$. This should be acceptable for the μA741, since the datasheet claims a maximum of 6 mV.



Study Exercise 2

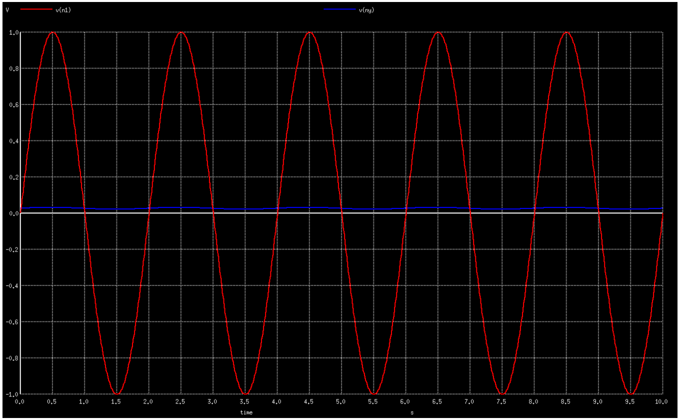
The file exercise2.sp describes the same circuit as exercise1.sp, but performs a different test to measure the open-loop gain. This time, you need to define a constant Ga with the numerical value you calculted in Exercise 2 of the pre-lab.

In this test, the input signal is a 0.5 Hz sinusoid with 1 V zero-to-peak amplitude. In the control block, meas statements are used to measure the peak-to-peak amplitudes of v_{out} and v_y. These are then used to compute the measured open-loop gain:

```
tran 10m 10s
plot v(n1) v(ny)

meas tran VY  PP v(ny) FROM=0 TO=10s
meas tran VIN PP v(n1) FROM=0 TO=10s
let A=VIN/VY*$&Ga
print A
```

Open-Loop Gain Measurements

The simulation will produce a pair of curves showing v_{in} and v_y. Although v_y is a very small signal, it should be large enough to measure with the oscilloscope.



Why Such a Low Frequency?

When you run NGSPICE on `exercise2.sp`, you should estimate a gain close to the datasheet value of 200 kV/V. Because op amps are designed to have a very low open-loop cutoff frequency, the input signal needs to be close to DC.

For example, the 741 has a specified unity-gain frequency near $f_t \approx 1\,\mathrm{MHz}$, and its DC open-loop gain is nominally $A_0 \approx 200\,\mathrm{kV/V}$, the open-loop cutoff frequency should be $f_{3\mathrm{dB}} \approx f_t/A_0 = 5\,\mathrm{Hz}$.

To see this, try changing the simulated frequency to 5 Hz. You should observe that the measured gain is reduced by about 30%, which corresponds to a 3dB drop.

Create a log.txt file and record the estimated gain for the 0.5 Hz and 5 Hz cases, and verify that they differ by about 30%.

Study Exercise 3

Next open exercise3.sp and study its contents. This circuit implements the unity-gain follower corresponding to Fig. 2 in the pre-lab. This exercise introduces several new concepts in SPICE simulation. First, we see a distinction between .param and .csparam statements:

```
.param f=30k
.csparam f=f
.csparam T=2/f
```

Here the .param statement declares a parameter f that can be used in component statements (here it is used in the Vin declaration). The .csparam statement defines constants to be used in the control block.

Transferring Data to Matlab

In this exercise, a transient simulation produces data that can be used to measure the slewing characteristics. It would be useful to analyze this data in Matlab. To do so, we use the `wrdata` command to save the desired data in a text file:

```
* Control Commands:
.control

tran 10n 1m
linearize
wrdata slewing v(nout)
```

These lines perform a transient simulation and save the v_{out} result in a file called `slewing.data`.

(Note: there are multiple ways to transfer SPICE data into Matlab. This method is very simple but may not work for very complex designs.)

Accessing Data in Matlab

Open Matlab from your project directory and use the `importdata` command:

```
d=importdata('slewing.data');
```

This command reads the file and stores the data in a matrix called `d`. This matrix should contain two columns: the first column is the time (in seconds), and the second column is v_{out} (in volts). Start by plotting the data to verify that you transferred it successfully:

```
plot(d(:,1),d(:,2))
xlabel('Time␣(s)')
ylabel('Vout␣(V)')
```

Measuring the Signal Slopes

Since we are analyzing the slew rate, we will want to know the signal's maximum slope, $\max\left(\frac{dv_{\mathrm{out}}}{dt}\right)$ (for rising voltage) and its mininum slope, $\min\left(\frac{dv_{\mathrm{out}}}{dt}\right)$ (for falling voltage). In Matlab, the slope can be estimated by using the `diff` command, which computes the difference between every pair of adjacent elements in a vector:

```
dvdt = diff(d(:,2))./diff(d(:,1));
maxslope = max(dvdt)
minslope = min(dvdt)
```

You will use this method to measure the op amp's slew rate.

Fourier Analysis

Back in the `exercise3.sp` file, the next interesting line is the fourier analysis command:

```
  linearize
  fourier $&f v(nout)
```

This command requests a detailed harmonic analysis for the specified fundamental frequency (here `f`) on the specified signal (here v_{out}). The command prints out the amplitudes of harmonic components appearing in v_{out}, and it also reports a distortion measure called THD. It must be preceded by the `linearize` command, which interpolates the simulation data onto a uniform sample grid.

```
Fourier analysis for v(nout):
  No. Harmonics: 10, THD: 0.166383 %, Gridsize: 200, Interpolation Degree: 1

Harmonic Frequency    Magnitude    Phase        Norm. Mag    Norm. Phase
-------- ---------    ---------    -----        ---------    -----------
 0        0           2.32095e-05 0             0            0
 1        30000       1.99988      -1.9567      1            0
 2        60000       0.000286781 -5.1317       0.000143399 -3.175
 3        90000       0.00328847   -100.23      0.00164434   -98.277
...
```

Total Harmonic Distortion

THD is an important measure of signal distortion. Specifically, it measures the percentage of total signal energy distributed into harmonic distortion components. If R_i is the RMS-magnitude of the i^{th} harmonic (with R_1 being the fundamental), then

$$\text{THD} = \frac{1}{R_1}\left(\sum_{i=2}^{8} R_i^2\right)^{1/2}$$

The amount of THD that can be tolerated varies depending on application. For our purposes, THD below 0.5% can be considered low-distortion, and THD above 1% can be considered high distortion.

FFT Analysis and New Plot Options

The remaining lines in the control block are:

```
fft v(nout)
plot xlog xlimit 1000 10e6 vdb(nout)
wrdata fft vdb(nout)
```

These lines request an FFT simulation on the transient results for `v(nout)`. After the FFT is computed, the result is stored in a new vector named `v(nout)` (i.e. the same name as the original vector). To view the FFT spectrum, we typically use a log-scale X axis. In NGSpice, a log-scale is requested by using the `xlog` argument in the `plot` command. We will further want to restrict the plot to a specific domain between `fmin` and `fmax`, which is achieved by the arguments `xlimit fmin fmax` appended to the `plot` command.

Lastly, the `wrdata` command is used to write the FFT data to a file named `fft.data`.

Viewing the FFT Result in Matlab

To load the FFT data into Matlab, use the `importdata` command as before, and plot it with a log-X axis using the `semilogx` command as follows:

```
d = importdata('fft.data');
semilogx(d(:,1),d(:,2))
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
```

In this case, the imported matrix `d` contains two columns. The first column is the FFT frequency in Hz, and the second column is the FFT magnitude in dB.

Study Slew Rate and Distortion

Now edit `exercise3.sp` and simulate it for three different values of `f`: 30 kHz, 40 kHz, and 50 kHz. In your `log.txt` file, record a line stating what frequency you expect to observe slew-rate distortion (from the end of Exercise 3 in the pre-lab). Then, for each of the three frequencies, perform the following tasks:

- Record the THD value in `log.txt`.
- Transfer the transient data into Matlab and calculate the maximum and minimum slope values. Record these values in `log.txt`.
- Plot the FFT result in Matlab and export the figure as a PDF file. For each figure, provide a title stating the simulated frequency.

For the 50 kHz case, save the transient figure from Matlab as a PDF file. At this frequency you should be able to see obvious slew-rate limiting, and the slope values should correspond to the datasheet slew-rate of 0.5 V/ μs. Record this as the slew-rate in the last line of `log.txt`.

What to Turn In

Turn in a ZIP file containing your `log.txt` file and the exported Matlab figures (there should be four figures).

Record the simulated open-loop gain measurement, THD values, and slew-rate in your lab book, or printout your `log.txt` file and tape it into your lab book for reference during your hardware session.