Introduction to SPICE Analysis Techniques

Background

In this experiment, a simple circuit will be investigated using SPICE simulations. The experiment has been designed so that the student will encounter situations that are common in the simulation of electronic circuits. The three major types of analyses will be used: DC, AC, and transient. This experiment is not meant as an exhaustive tutorial; students should have some previous exposure to SPICE and access to some reference material or help.

The text of this experiment gives some guidance regarding the use of **MicroSim PSPICE version 8.x.** Footnotes have been included to help with **Orcad PSPICE version 9.x.** Standard PSPICE text on menus and buttons is shown in **`this font.`** Variable names and expressions that are typed by the user or listed by PSPICE will appear in quotes. It is assumed that the PSPICE postprocessor, called PROBE, will be used to display the simulation results.

An addendum to this experiment describes changes that apply when **writing the SPICE file as text** rather than using either schematic capture program. In this case, the use of a postprocessor is still assumed.

Preliminary Calculations

DV **Step 1.** For the circuit of Figure 1, the default component values are: $R_1 = 4.7\text{k}\Omega$, $R_2 = 2.2\text{k}\Omega$, and $C = 0.02\mu\text{F}$ (your instructor may assign different values). Calculate the magnitude of the transfer function $\frac{v_2}{v_S}$ at frequencies for which the capacitor can be considered an open circuit. Also calculate the upper -3dB frequency of $\frac{v_2}{v_S}$ (see appendix A3.2).

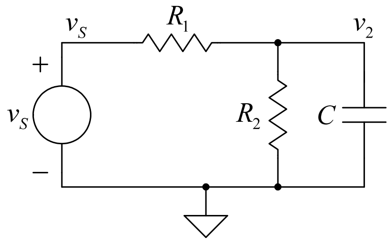


Figure 1. Drawn Schematic

Step 2. Calculate the 10% to 90% risetime of v_2 in response to a step input at v_S. See Appendix A11.1 for a relationship between bandwidth and risetime.

SPICE Simulations

Part I: Setting up the simulation

Step 3. Enter the circuit of Figure 1 into PSPICE. Figure 2 shows the corresponding PSPICE schematic capture display after the part values are edited. The part values are edited by double-clicking on the value. These values should be the same as those used in Step 1. The part values and references (R1, R2, C1) may also be moved by clicking and dragging. The voltage source used here may be located by typing the part name "vsrc" in the **`Draw/Get New Part`** dialog box[1]. The source parameter **`DC`** is its DC voltage. The source parameter **`AC`** is the magnitude of the source voltage for AC analysis (described before Step 7). Note that the parameter **`AC`** is used only for AC analysis; it is not the amplitude of the sinusoidal function of time produced for a transient analysis. For transient analyses, the source must be changed to the one called "vsin," and parameters for frequency, amplitude, offset, and phase must be entered. To edit the source

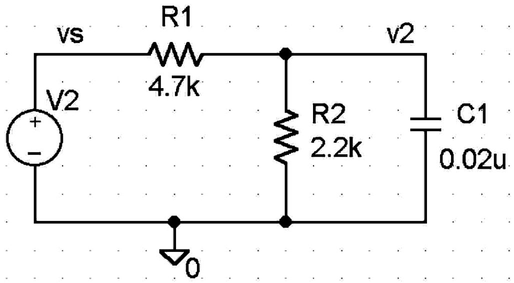


Figure 2. PSPICE Schematic

[1] Vers. 9: **`place/part`** This part may automatically display attributes such as **`ACMAG`** or **`VAMPL.`**

parameters, double-click on the source to get a dialog box. The parameters of the source can be made to appear on the schematic after clicking the **`Change Display`** button[2].
PSPICE allows the use of different ground symbols, but the schematic must have at least one ground which is designated as node 0 in the circuit netlist. All single-ended voltages are expressed with respect to this ground. To get this symbol, go to the **`Draw/Get New Part`** menu and type "AGND"[3]**.**
In Figure 1, the voltage source is labeled "v_S" because that is the name for the signal produced by that source. Hence the positive terminal node of the source is also labeled v_S. To label the corresponding node in PSPICE, double-click on a wire that is part of the node and a dialog box will appear. Assign labels to the nodes v_S and v_2, as shown in the figure[4].

Part 2: DC sweep

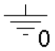
This type of analysis is used to find the DC transfer characteristic of a circuit (Appendix A3.1). The simulation will sweep the DC value of a particular source specified by its part reference (i.e., its name). Any of the PSPICE voltage (or current) sources may be swept in a DC sweep.

Step 4. Go to **`Analysis/Setup/DC Sweep`**[5]**.** You must specify the source to be swept by its reference designator (e.g., V1). Then specify a linear DC sweep of v_S from –2 to +2 volts. An increment of 10mV will provide sufficient data. Run the simulation. The default setup is that the post-processor, called Probe, will automatically start up after the simulation is finished. Probe will allow the graphical display of the simulation results, and it usually begins by displaying an empty graph with the horizontal axis being the variable that was swept in the simulation.

Step 5. Use the **`Trace/Add`** menu to select the traces to display. A dialog box will appear listing all possible traces. The length of the list can be reduced by unchecking some of the trace categories. To display traces for v_S and v_2, leave only the following categories checked: **`Analog`**, **`Voltages`**, and **`Alias Names`**. From the list, select "V(vs)" and "V(v2)" to display the voltages at those nodes. To delete a trace from the display, click on its name at the bottom left of the graph and press the delete key. The trace for v_S will appear as a straight line with a slope of 1 since it is also the horizontal axis variable. The trace for v_2 is also a straight line, but it has a lesser slope since it is a fraction of v_S. Use the cursors to measure $\frac{\Delta v_2}{\Delta v_S}$ and compare this to the calculated value of $\frac{v_2}{v_s}$ from Step 1. The cursors may be enabled from the menu **`Tools/Cursor/Display`**[6]. The cursors will be useful for obtaining numerical data from plots for the remainder of this experiment.

Step 6. This step will explore alternate ways of finding the gain (small-signal) from the simulation data. Clear the traces on the graph from the previous step. In the **`Trace/Add`** dialog box, enter the expression "v(v2)/v(vs)". Sketch or describe the resulting trace. Does it have a reasonable value? Is it continuous?

[2] Vers. 9: **`Display`**

[3] Vers. 9: go to **`Place/Ground`** menu and select the part called **`0/SOURCE`**, shown here: 

[4] Vers. 9: assign a *net alias* to each node. To do this, type capital "N", then type the alias in the dialog box, and then place the label on the schematic in contact with a wire that is part of that node. The display should look like Figure 2.

[5] Vers. 9: Create a simulation profile for the DC sweep. To do this, go to **`Pspice/New Simulation Profile`** and give the new profile a name. After clicking **`Create`**, a dialog box will appear. Enter the desired sweep parameters. Click "**`Apply`**" to finish creating the profile.

[6] Vers. 9: **`Trace/Cursor/Display`**

Now clear the previous trace and enter the trace expression "dv(v2)". This is the derivative of v_2 with respect to the horizontal axis variable. Sketch or describe the resulting trace. (See conclusion item 2.)

Part 3: AC Sweep

AC analysis is used to determine the frequency response of a circuit. Although the analysis is based on the sinusoidal steady-state response, the voltages and currents in the circuit are not expressed as sinusoidal functions of time. Roughly speaking, the analysis calculates the terminal characteristics of each element (e.g., complex impedance) at each frequency of the sweep, and finds a solution for all of the voltages and currents in the circuit. For this type of analysis, the signal source must be assigned an AC magnitude and phase. Normally only one source in the schematic is assigned an AC value, and that source then acts as the input. Typically, the AC voltage source is assigned 1V magnitude and $0°$ phase.

Step 7. In the schematic, be sure that the voltage source is assigned an AC magnitude of 1 volt. Go to **`Analysis/Setup/AC Sweep`**. Then set up a logarithmic sweep by selecting **`Decade`**. The sweep should start at 100Hz, end at 100kHz, and have about 30 points per decade[7]. Run the simulation. Again, display traces for v_S and v_2. On the graph, use the cursors to determine the magnitude of v_2 at frequencies well below the upper –3dB frequency. Note that the magnitude of v_2 is the same as the magnitude of the transfer function $\frac{v_2}{v_s}$, since the magnitude of v_s is 1 volt. (The AC magnitude of input voltage sources is usually set to 1 volt for this very reason.). Use the cursors to find the magnitude of $\frac{v_2}{v_s}$ at low frequencies (i.e., where the magnitude response graph is flat), and compare this to $\frac{\Delta v_2}{\Delta v_S}$ obtained from the DC sweep of Step 5.

Step 8. Delete all traces on the graph and obtain a new trace by typing the expression "DB(V(v2))". This will display the magnitude of v_2 in dB. Alternately, click on v(v2) in the list of traces, and by typing insert the letters "db" in the expression to obtain "vdb(v2)". From the resulting graph, obtain the midband gain in dB, and convert it to $\frac{\text{V}}{\text{V}}$ to compare it to previous results. (See conclusion item 3.)

Step 9. From the AC sweep plot of Step 8, determine the upper –3db frequency of $\frac{v_2}{v_s}$. Compare this to the –3dB frequency calculated in Step 1. (Hint: This is the frequency at which v_2 drops by 3dB from its low-frequency value.)

Step 10. Continue to display the magnitude of v_2 in dB. Add a plot to the window by going to the **`Plot`** menu. Click on **`Add Plot`**.[8] The existing plot will shrink, and a blank plot will appear above it. Click on the area of the blank plot to make sure that it is the active plot (as indicated by **`SEL>>`** to the left of the vertical axis). To this plot, add the trace "P(V(v2))" or the equivalent "Vp(v2)". This is the phase of v_2, and is equal to the phase of $\frac{v_2}{v_s}$ since the phase assigned to the AC source (by default) is zero degrees. From the plot, determine the phase of $\frac{v_2}{v_s}$ at the upper –3dB frequency.

[7] Vers. 9: Create a new simulation profile (with a different name) that performs an AC sweep. Select **`AC Sweep/Noise`** in the simulation settings dialog box. Then set up a logarithmic sweep. Now that you have more than one simulation profile, it is helpful to know how to change between them. To change to a different simulation profile, right click on that profile in the project manager window and select **`make active`**.

[8] Vers. 9: **`Add Plot to Window`**

Part 4: Transient simulation

Transient simulations are used to obtain the circuit voltages and currents as functions of time. This type of simulation is different from those previously discussed in that the sweep must start at zero $(t = 0)$, and that the results at each time step must depend on the results at the previous time step. In order to approximate a continuous-time system, the time steps must be short so that circuit voltages and currents do not change too much in any one time step. During the simulation, SPICE will automatically adjust the step size to take smaller steps when voltages or currents change faster. This *adaptive step size* saves memory and simulation time while keeping good resolution for fast waveform transitions. The user can set a maximum step size, called a step ceiling.

In PSPICE, different source types produce different signals (waveforms) for transient analysis. Each type of transient source can also be assigned values that apply to DC and AC sweeps. Note that the assigned AC value of the source is used only for AC sweeps; it is not the sine wave amplitude for transient analyses.

Step 11. Edit the parameters of the voltage source **`Vsin`** in the PSPICE schematic. Assign the following parameters to the source: a DC offset voltage of 0V, an amplitude of 1V, and set its frequency equal to the –3dB frequency obtained from previous results. Go to **`Analysis/Setup/Transient`** and set up a transient sweep with a final time corresponding to 5 times the period of the sine wave source[9]. Run the simulation and display the two traces "v(vs)" and "v(v2)". The sine wave traces displayed may not look like smooth curves, due to large time step sizes. If that is the case, return to the schematic and in the setup menu[10] enter a step ceiling (i.e., the maximum step size) so that there are at least 100 points in each cycle of the sine wave. Re-run the simulation and note the improved smoothness of the two traces. Use the cursors to obtain the peak-to-peak amplitude of each sine wave at this frequency.

► Read the peak-to-peak values near the end of the simulation, after the initial transient has decayed.

Find the peak-to-peak magnitude of v_2, the peak-to-peak magnitude of v_s, and determine $\left|\frac{v_2}{v_s}\right|$ at this frequency.

Step 12. In the plots obtained in Step 11, how is the very first cycle of the output sine wave different from subsequent cycles?

Step 13. Display the trace "v(v2)/v(vs)". Why doesn't this trace show the value of $\left|\frac{v_2}{v_s}\right|$?

Step 14. Now edit the circuit of Figure 2 by replacing the **`vsin`** voltage source with the part **`vpulse`**. Assign the source parameters as given in Table 1. Again, run a transient simulation. Obtain plots of v_S and v_2. Identify and record the "final" value to which v_2 settles after v_S transitions from 0V to 1V (but before v_S goes low again). Is this final value of v_2 consistent with the results of the DC sweep done earlier?

Property	Spice	Value
initial voltage	**V1**	0
pulsed voltage	**V2**	1
time delay	**TD**	0.25m
risetime	**TR**	1u
falltime	**TF**	1u
pulse width	**PW**	0.25m
period	**PER**	0.5m

Table 1.

Step 15. From the simulation results of the previous step, determine the 10% to 90% risetime of v_2 and compare this value with that calculated in Step 2.

A Non-Linear Circuit (DC Sweep)

Step 16. Now change the circuit to that of Figure 4. Use the same value of R_1 as before. In PSPICE, the diode D_1 should be a 1N4148,

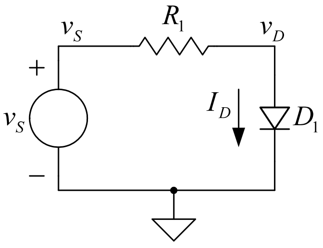


Figure 4.

[9] Vers. 9: Create a new simulation profile of the type **`Time Domain (`** to about 5 times the period of the sine wave source.

[10] Vers. 9: Edit the simulation profile.

which can be obtained by typing "D1N4148"[11] in the **`Draw/Get New Part`**[12] dialog box. Label the voltage across the diode v_D, as shown in the figure. Give the voltage source v_S a DC value of 1.5V by double-clicking on the source. Repeat the DC sweep of Step 4. After the simulation, obtain a plot showing v_S and v_D as functions of v_S. Add a plot (as in Step 10) showing I_D as a function of v_S. Using the cursors, find the value of I_D at $v_S = 1.5\text{V}$. Close the PROBE window and return to the schematic. Find the buttons that enable the display of node voltages and branch currents on the schematic (in both versions, these are buttons on the simulation toolbar labeled "**`V`**" and "**`I`**"). For $v_S = 1.5\text{V}$, note and record the values of v_D and I_D. Do the values on the schematic agree with those taken from the graph? Obtain a plot of the derivative of v_D using the same technique used in Step 6. (See conclusion item 5.)

Conclusions

Item 1. The graph of Step 5 was used to find the value of $\frac{\Delta v_2}{\Delta v_S}$. The same value can be found by dividing a single value of v_S by its corresponding value of v_2. Why is this so? What conditions must exist for this to be true?

Item 2. In general, will obtaining a trace of "V(vy)/V(vx)" show the small-signal gain $\frac{v_y}{v_x}$? For which types of analyses will this trace show the small-signal gain, and what are the limiting factors?

Item 3. Is the midband (low-frequency) gain found in Steps 8 and 9 the same as the gain obtained from the DC transfer characteristic (convert dB to $\frac{\text{V}}{\text{V}}$)? Should the gains be the same? Under what condition can a DC transfer characteristic be used to find the midband AC gain? Explain.

Item 4. Consider the graph of the AC sweep data produced in Step 7. Explain the general shape of the graph (where is it flat? where is it increasing or decreasing?) Is this a high-pass filter or low-pass filter? The simulation is based on principles of sinusoidal steady-state analysis, yet the resulting graph does not look like a sine wave. Why?

Item 5. What should be the phase response of the circuit of Figure 1 at its upper –3dB frequency? Do the simulation results from the AC and transient analyses agree?

Item 6. Does the diode-resistor circuit of Step 16 circuit have the same value of $\frac{\Delta v_D}{\Delta v_S}$ for all v_S? What are the maximum and minimum values of $\frac{\Delta v_D}{\Delta v_S}$?

Addendum for Those Writing SPICE Code

This section will provide guidance for those wishing to write their own SPICE circuit file. A few brief reminders on spice code are included here for convenience. The student should consult the text (chapter 4) or another SPICE reference book for help. Follow the directions in the previous steps, and refer to the corresponding steps below for help.

Step3. Write a circuit file for the circuit of Figure 2 in SPICE code. This file should be a text file and have the extension ".cir". The first line of the circuit file must be the circuit title. The circuit file also contains *netlist* lines, which describe the circuit elements and their interconnection, and *command* lines, that specify the simulations to be done and their parameters. Finally, a *program control* line is at the end of the circuit file.

[11] This part is in the **`Eval`** library of the student version, but may be in the **`diode`** library of other versions.

[12] Vers. 9: **`Place/Part`** dialog box.

To write the netlist, consider that the circuit has three nodes, which will be named "vs", "v2", and "0". The circuit ground must be named "0" (zero). Each element is described by a line of code. For example, the lines for resistors have the following syntax:

`r`(*ref.desig.*) (*pos. node*) (*neg. node*) (*part value*)

Specific circuit parameters have been indicated by parenthetical expressions. For example, R_1 is described by this line:

```
r1  vs  v2  4.7e3
```

The line of code describing the voltage source has the following syntax:

`v(`(*ref.desig.*) (*pos. node*) (*neg. node*) **`dc`** (*dc bias value*) **`ac`** (*ac signal amplitude*)

When writing the code, give the source a dc value of zero and an ac amplitude of 1.

Step 4. To perform a DC sweep, add a command line to the SPICE file. The line is as follows, with spaces between entries:

`.dc` (*sourcename*) (*lowerlim*) (*upperlim*) (*increment*)

Complete the command line using the simulation parameters given in the handout.

To cause PSPICE to generate a data file (usually has a .dat file extension) suitable for use with the postprocessor **`Probe`**, include the line **`.probe`** in the circuit file and the statement **`.end`** at the end of the file. For other postprocessors, consult the user's manual.

Now, the SPICE simulator should run, using the circuit file just written as the input. If using PSPICE, start the program **`Pspice`** instead of the program **`Schematics`**. To do this, go to the **`Design Manager`** window. Use the **`File/Open`** menu to find the .cir file just written, and open the file. Pspice will then run the simulation. The postprocessor will not start automatically; again, go to the **`Design Manager`** window and start **`Probe`**. For those using a unix system, use a command syntax like:

spice (*circuitfilename*).cir >! (*outputfilename*).out

Of course, the program name may vary (e.g., hspice) and the correct path to the program is needed in the line above or in the .cshrc file.

Steps 5 and 6. If the postprocessor is not **`Probe`**, refer to user's information for the particular program in use.

Step 7-10. To run the ac sweep, add a line of code with this syntax:

`.ac dec` (*pts / dec*) (*lowerlim*) (*upperlim*)

Other sweep types are available, but not described here.

Steps 11-15. For the transient sweep, add a line of code with this syntax:

`.tran` (*print interval*) (*final time*)

It should be noted that most postprocessors will use <u>all</u> data from the simulation, not just the values obtained at multiples of the print interval. For small circuits, this does not create a problem with storage space. For larger circuits and/or longer simulations, however, other commands may be used to limit which data is kept. In these cases, the print intervals may represent the only data points kept.

The netlist line for the voltage source will also need an addition. Add the following to the end of the line:

`sin`(*offset amplitude frequency*)

It should be noted that the waveform parameters are separated by spaces.

Step 16. Delete the netlist lines corresponding to deleted circuit elements. To add the diode, use a line with the following syntax:

`d`(*ref. desig.*) (*anode*) (*cathode*) (*modelname*)

Recall that the cathode is the "bar" end of the diode. There must be an additional statement in the file with the model having the same name given above. That statement would have the following syntax:

`.model` (*modelname*) **`D`**(*list of parameters*)

In order to write this statement for the 1N4148 diode, the list of parameters for that diode must be known. If you are using PSPICE, the model name is D1N4148 and the model file is part of a library. If the entire library is included in the circuit file, the program will find and use the model with the correct name. The entire library may be included in the circuit file by including the following single statement in the circuit file:

`.inc` (*path\libraryname*.**`lib`**)

If PSPICE is installed on the computer, the libraryname is typically called "nom.lib" or "diode.lib". Otherwise, find the *libraryname* for the library containing the 1N4148 diode model by searching for all files with the extension ".lib" that also contain the text "D1N4148".