

# |LTSpice Workshop

IEEE R4 SAC  
November 2020

## Table of Contents

Overview .....	2
Part 1: Background and Setup .....	2
What is LTSpice .....	2
Why we use LTSpice .....	2
How to obtain LTSpice software .....	2
Part 2: Basics .....	3
Creating New Schematic .....	3
Placing Components .....	3
Moving and Rotating components .....	4
Wiring Circuit .....	5
Assigning Values to Components .....	5
Simulation .....	6
Interpreting Simulation Results .....	8
Measuring Voltage .....	9
Measuring Current .....	10
Part 3: LTSpice Features .....	12
Types of Sources .....	12
DC .....	13
PULSE .....	14
SINE .....	15
PWL (Piecewise-Linear) .....	16
PWL FILE .....	17
Types of Simulation .....	17
Bias Point .....	17
DC Sweep .....	17
AC Sweep/Noise .....	17
Time Domain (Transient) .....	18
Monte Carlo Analysis .....	18

## Overview

This workshop aims to teach the basics of how to navigate LTSpice. These basic skills will allow the participant to create through this software and move into larger projects at their own pace. LTSpice is a useful tool to know whether it be for industry, school, or home projects.

## Part 1: Background and Setup

### What is LTSpice

LTSpice is a high-performance SPICE (Simulation Program with Integrated Circuit Emphasis) software that is among a family of SPICE software's. This software is enhanced to use analog components such as resistors, diodes, and transistors in simulations. The LTSpice download includes a variety of libraries that each include many different components.

### Why we use LTSpice

LTSpice is popular because it is an open source software used for high-performance simulations. This software has the same computing power as other SPICE software's that can cost thousands of dollars. LTSpice also allows you to upload components that do not come with the software initially, so you can build the specific circuit that you are working on. Software just like this is used in every industry, so being able to say you have a basic understanding of it will also help for job opportunities.

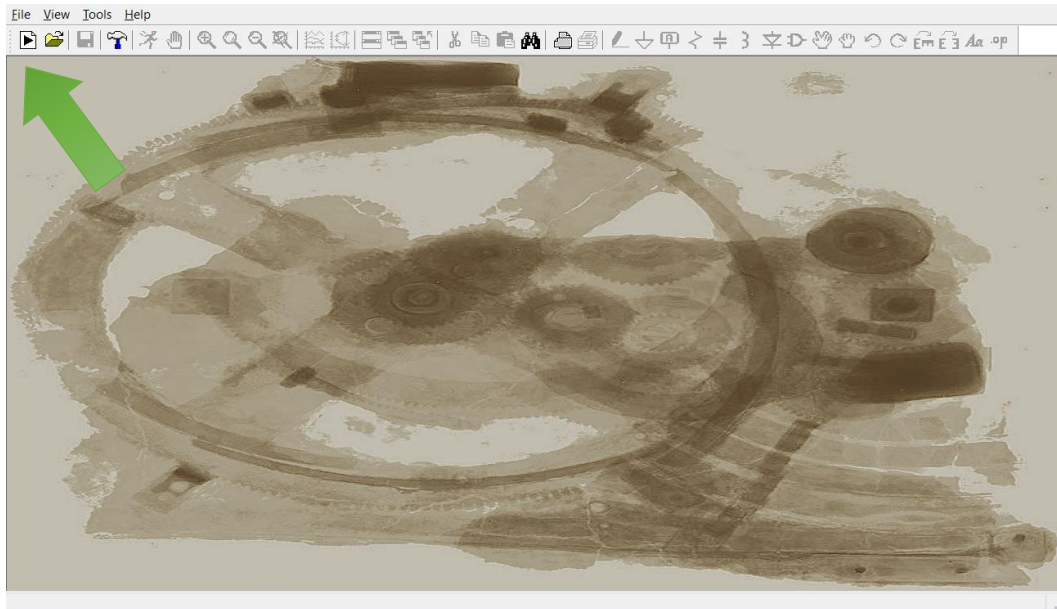
### How to obtain LTSpice software

Since LTSpice is an open source software, it can be downloaded off the internet. The download link is here: <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>

## Part 2: Basics

### Creating New Schematic

The first step is to create a new schematic. In order to do this, you can click on the little paper with a play button on it in the top left of the screen, as shown in Figure 1. You can also open a new one by going to File > New Schematic.



*Figure 1.* Opening a new schematic

### Placing Components

Once you have a new schematic open, you can then start placing components. The first way to do this is by using the Edit tab at the top. You can choose Resistors, Capacitors, Inductors and Diodes straight from the menu, but everything else will be found under Component. These can also be found on the menu bar at the top. Both options can be seen in Figure 2 below.

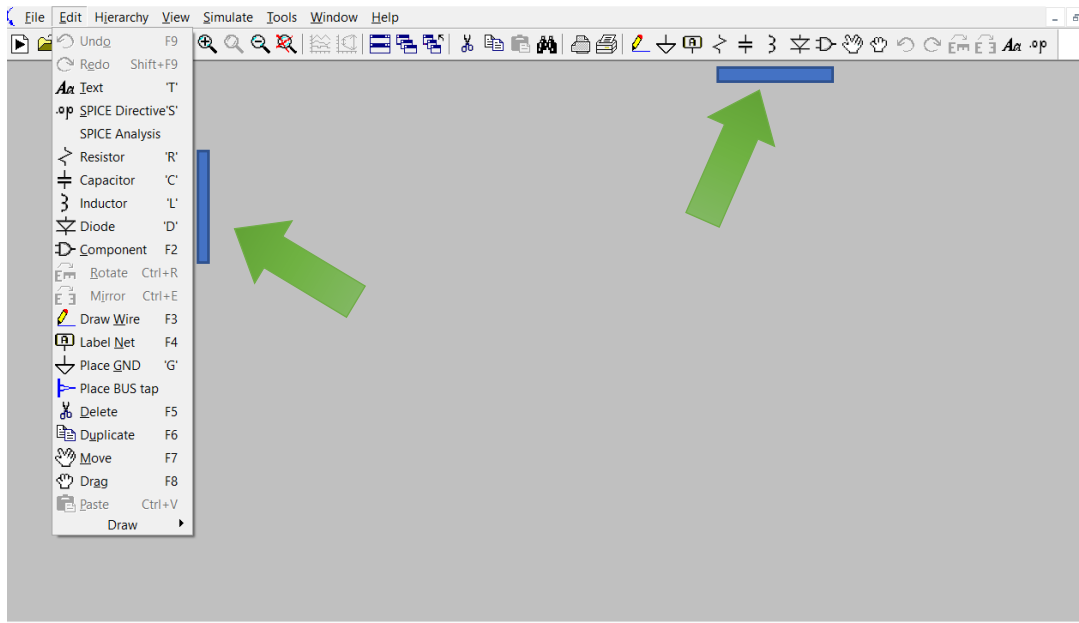


Figure 2. Selecting components

### Moving and Rotating components

After placing some components in the schematic, you may need to move them around to be able to hook up the circuit. There are two options when moving the components: move and drag. The move feature will disconnect everything and move the component to wherever you place it. Using drag will keep all wires connected when you move it. Before we hook any wires up, these will function the same. This process can be seen by looking at the green arrows in Figure 3.

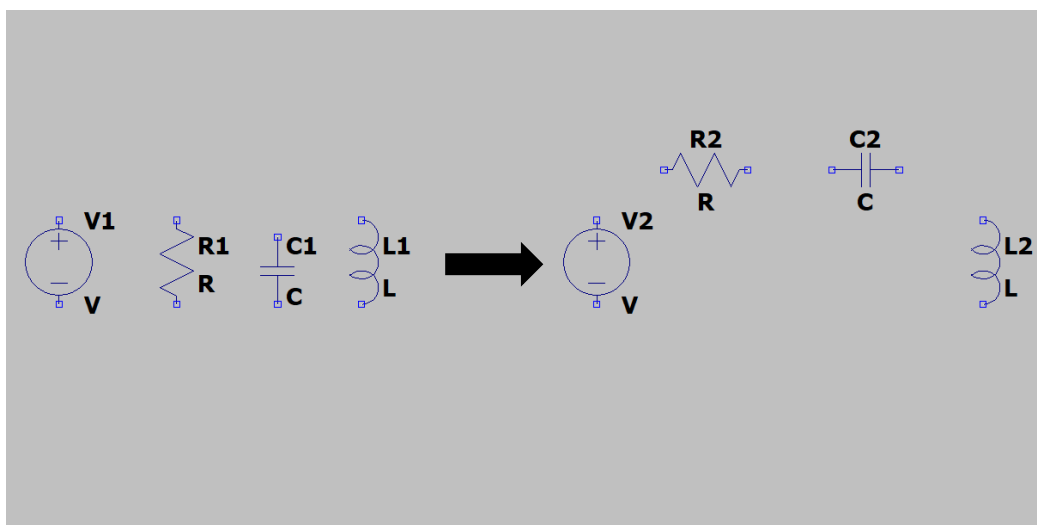


Figure 3. Rotated components

## Wiring Circuit

After the components are placed where we want them, they are ready to be wired. You can start this by clicking the pencil in the menu bar (to the left of the components) or “Draw Wire” in the Edit menu. Once you click this, a little box will pop up on each end of the components. These are the connection points of the component. Click on a connection point and drag your mouse to add wire. If you click while in this mode, it will place your wire. This is useful for when you have to switch the direction you need the wire to go. While doing this, your result should look similar to Figure 4.

Note: When creating a circuit, you always need a ground. Without one in the simulation, it will not run and will give you an error. You can find the ground either right next to the pencil in the menu bar or in the Edit tab.

Note: For this example, add an LED on the end with the inductor. This can be found in Components.

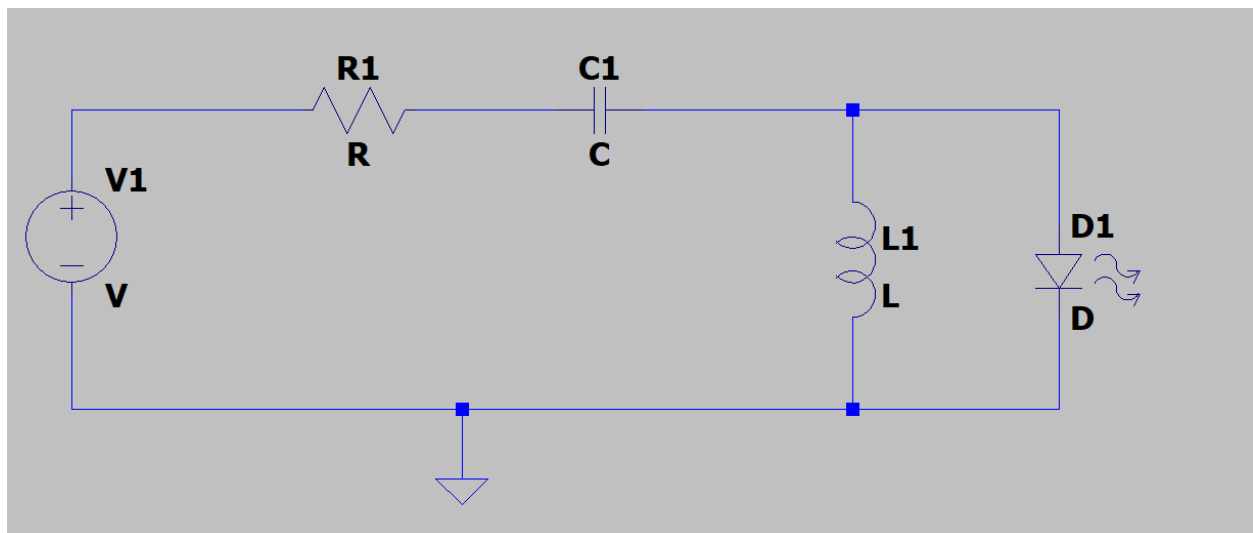


Figure 4. Circuit with wire connections

## Assigning Values to Components

Once we have the circuit setup, we need to assign values to basic components. In order to change the value of any component, hover over it and right click. This will allow you to add a value, tolerance, and power rating (You typically only need a value). Although, this only works for basic components (Resistors, Inductors and Capacitors). For all other components, you can right click and select a component for it, rather than just assign values. For example, if you right click on the LED, you get the menu shown in Figure 5 below. Then if you click on “Pick a Diode”, you can choose from the long list that pops up.

Note: If the component you want is not in the library, you can add your specific component as long as the model is open source. This will be explained later in the lab.

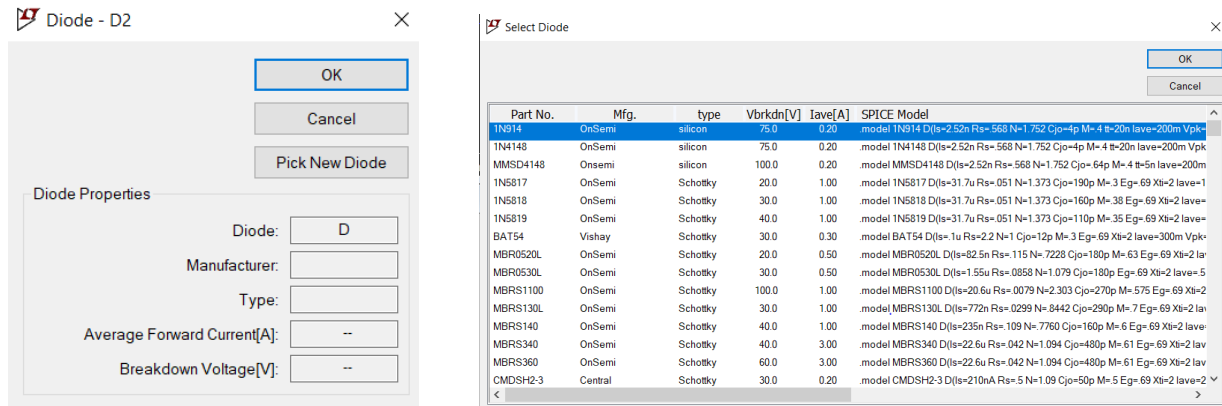


Figure 5. Component list

Let's choose:  $R = 100$ ,  $C = 1\mu\text{F}$ ,  $L = 1\text{mH}$  and a 1N914 diode. Then for the voltage source, use SINE with amplitude of 5V and frequency of 10kHz. This should look similar to Figure 6

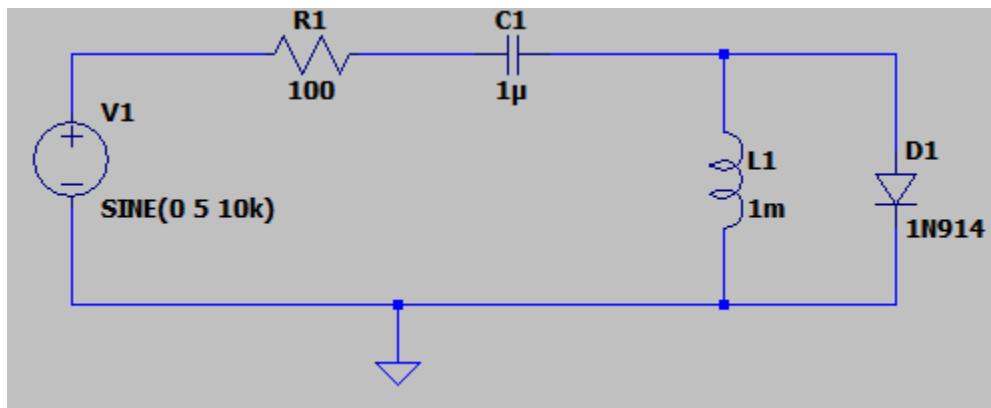


Figure 6. Circuit with values

Entered the wrong value and need to change it? Right click on the component and you can change it!

## Simulation

After we get the circuit completely setup, we are ready to run the simulation. In order to do this, go the Simulate tab and select Edit Simulation Cmd. This can be seen in Figure 7:

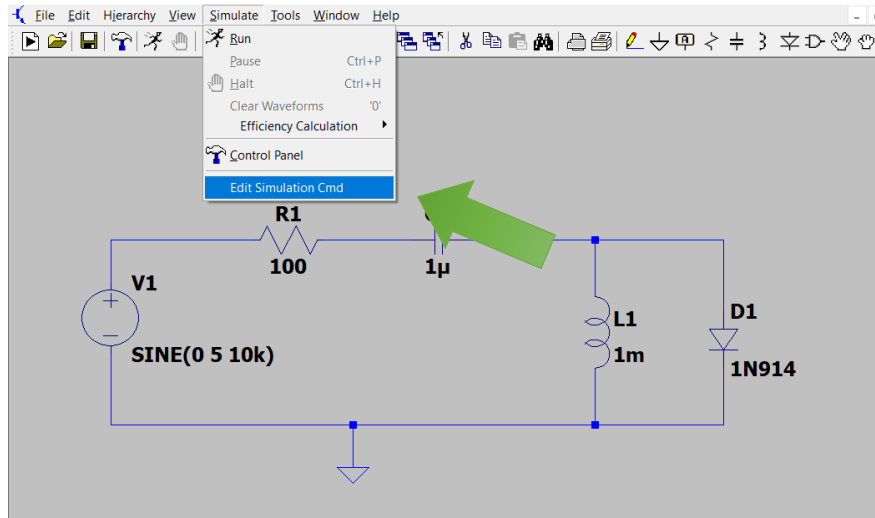


Figure 7. Finding the Edit Simulation Cmd link

Once we open that, we can select which simulation type we want. Once you select which type you want to run, you can fill in the parameters, then click Ok. This window can be seen in Figure 8.

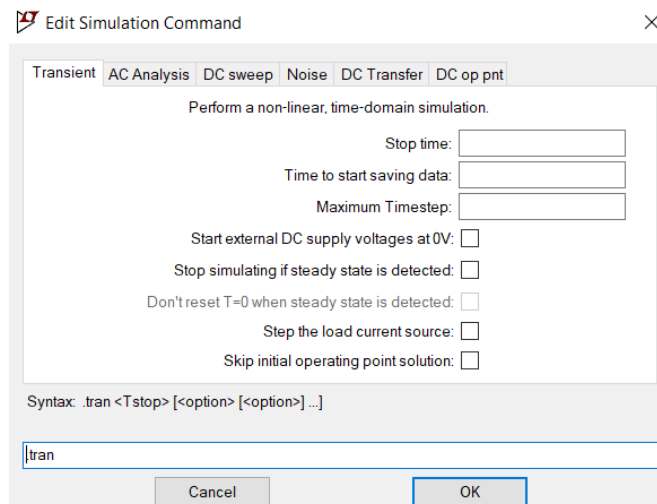


Figure 8. Setting simulation parameters

For this simulation, let's use a Transient Analysis with Stop time of 1mS.

Once this is all filled out, you can run it! This can be done in two different ways. The first is under the Simulate tab, Run. The other is the little running person on the menu bar. Both are shown in Figure 9.

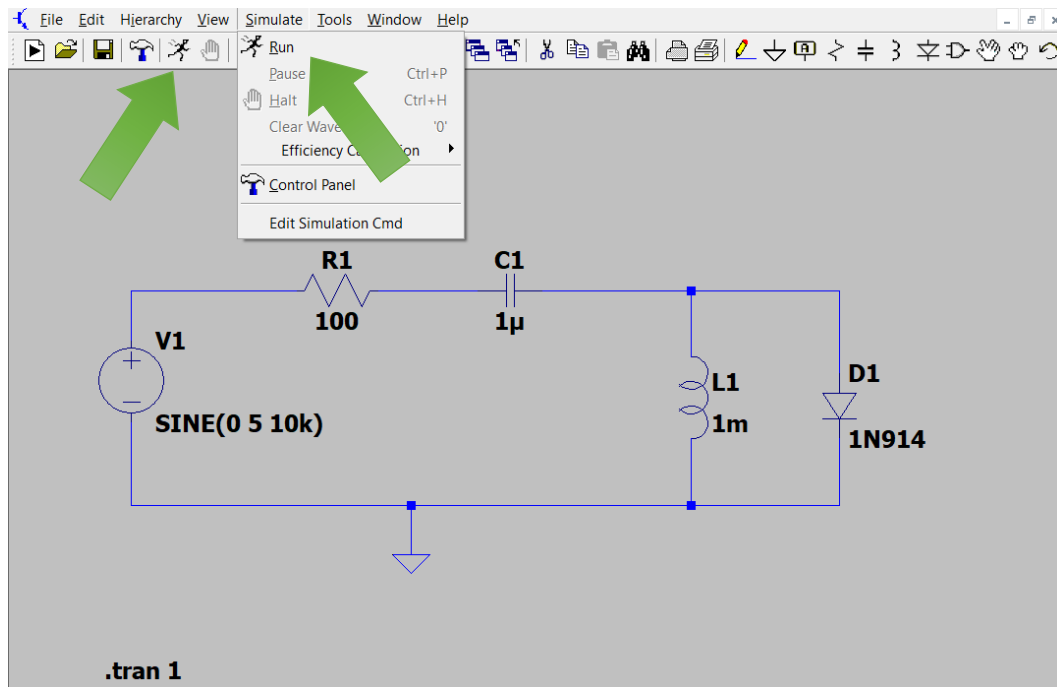


Figure 9. Run simulation options

### Interpreting Simulation Results

Once the simulation runs, a graph will automatically appear alongside your schematic. This is what we are going to use to interpret the data. Your screen should look something like the one below.

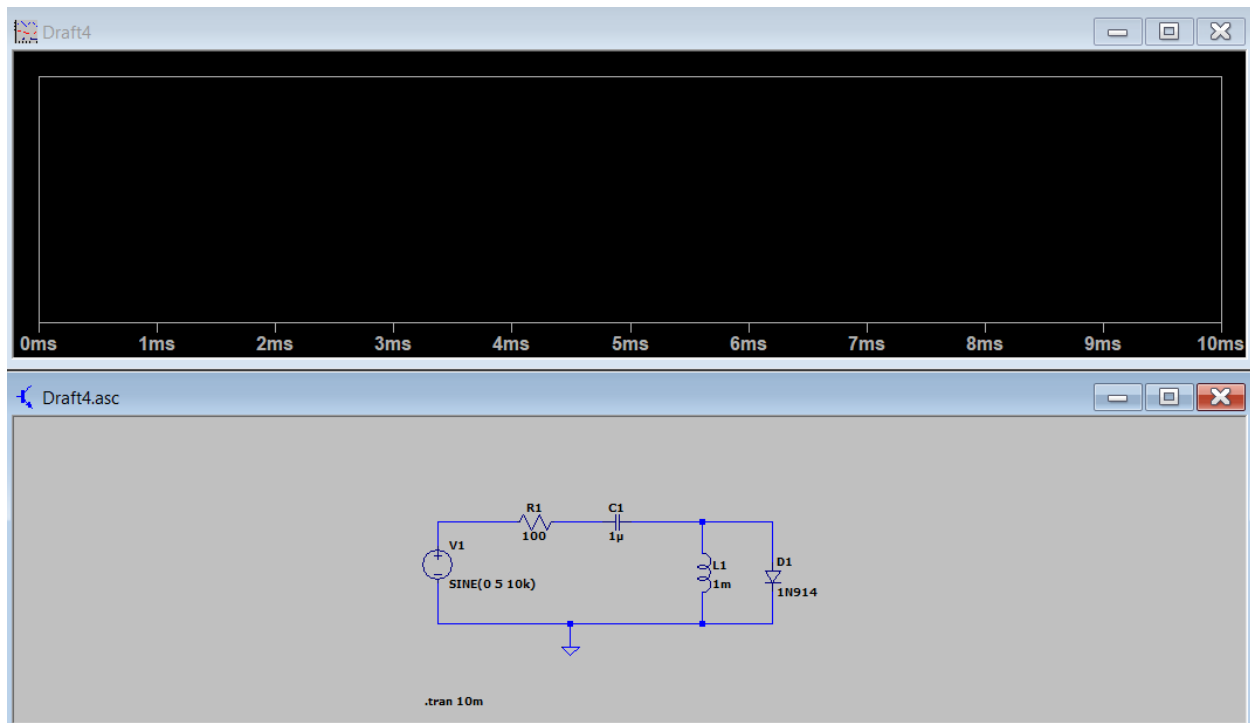


Figure 10. Simulation screen

## Measuring Voltage

After this, we are ready to plot some data. If you want to plot voltage at a node, all you have to do is click on the wire anywhere in that node. For example, if you want to see the voltage at the positive end of V1, click on the wire anywhere in the highlighted area below.

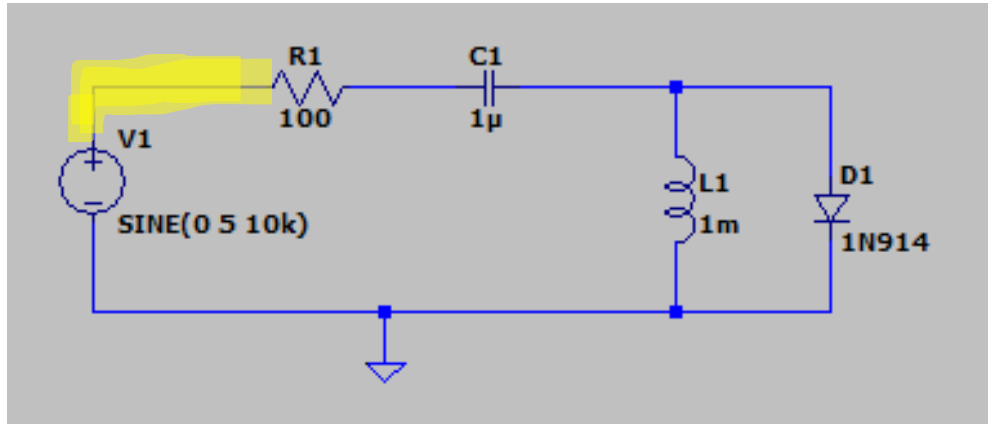
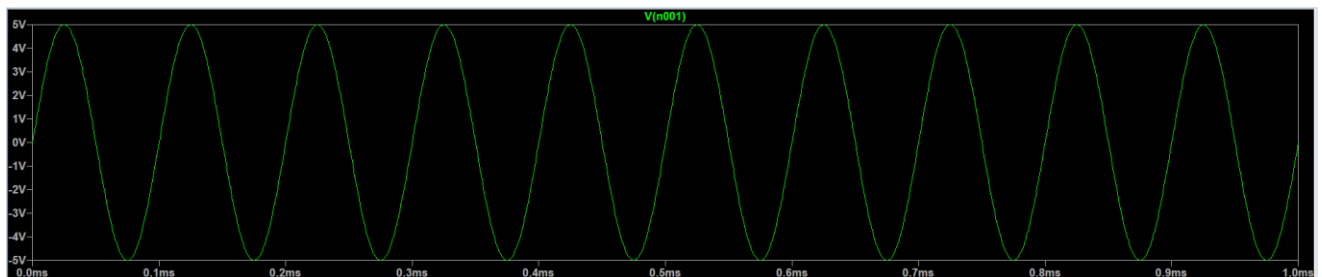


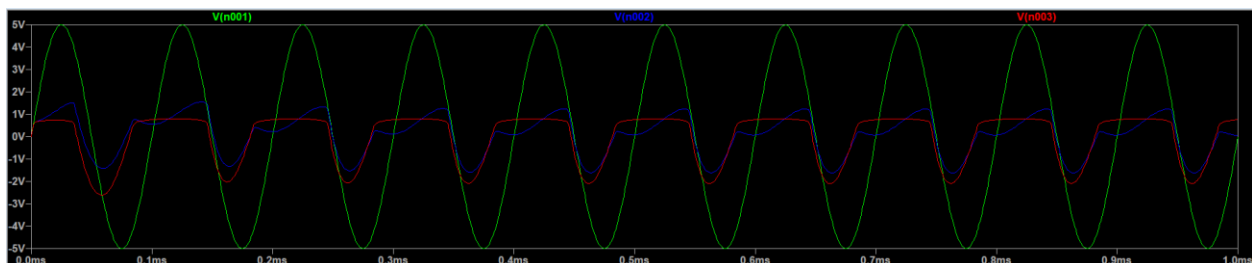
Figure 11. Node for measuring positive V1

Which should give you a result that looks like this:



Graph 1. Results from Figure 11

Then plotting all 3 nodes in this circuit should give you the result:



Graph 2. Results from 3 nodes

You can notice the top of these graphs can give you a key of which line is which. Like the green line is V(n001), which is the voltage at node 1. The blue line is V(n002), which is the voltage at node 2. Although, this can easily get confusing if you do not know which node is which. So, in order to know which node is which, we can label them. If you hover over any of the node, just like you are going to plot the voltage, right click, and select “label net”. Then this box will come up:

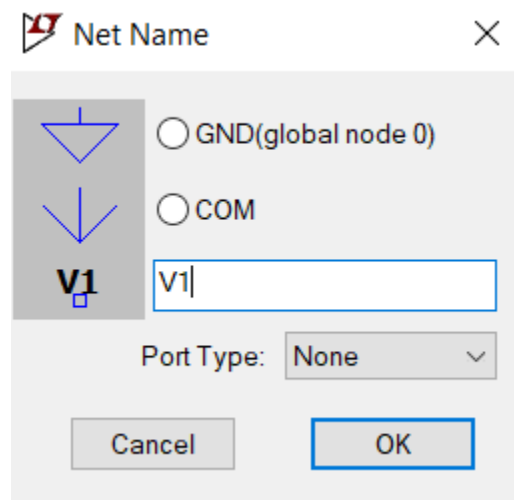


Figure 12. Window for labeling nodes

Then you can type any name for the node and click “OK”. Make sure you place the label on the desired node and repeat this for the other 2 nodes. Let’s use Vs, V1 and V2 for our 3 nodes. Then after re-running the simulation to update the graph, we can interpret the data a little easier.

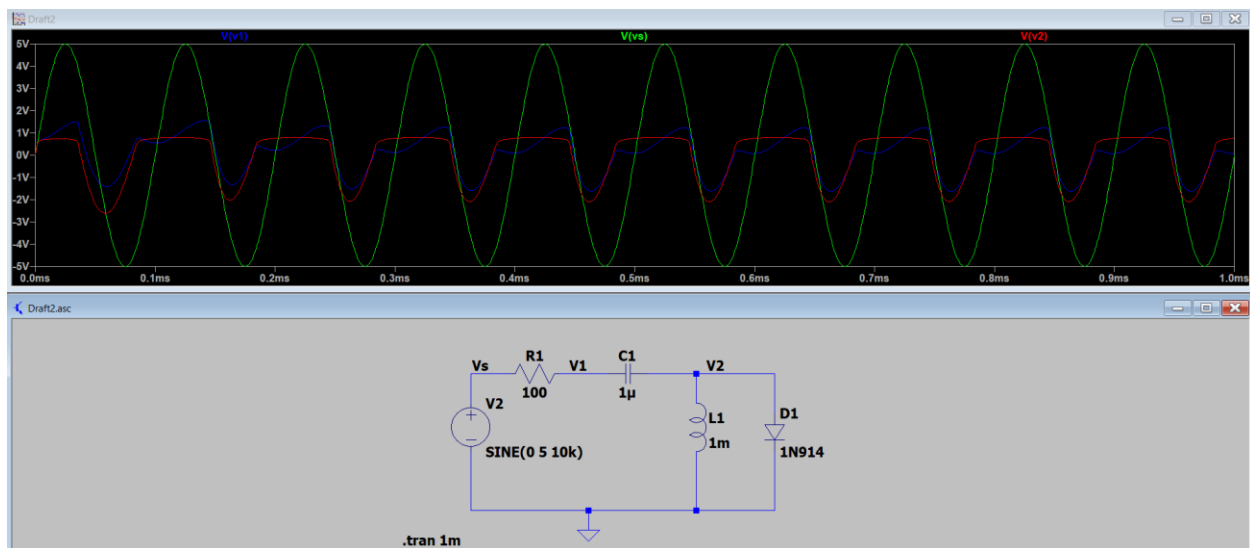
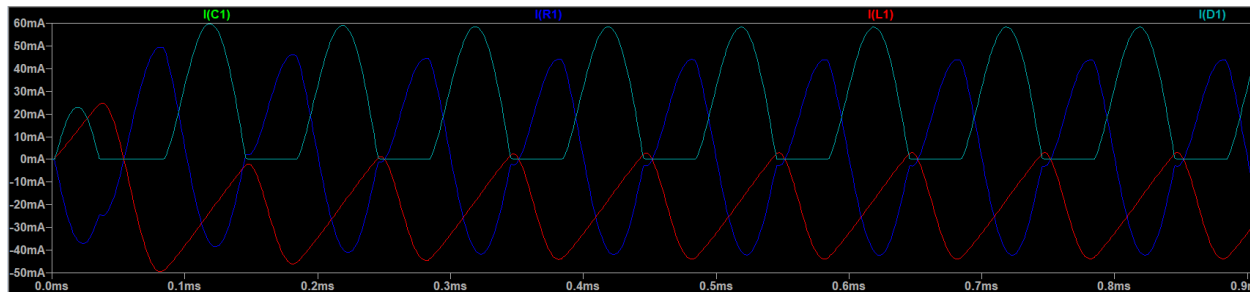


Figure 13. Labeled circuit and graph

## Measuring Current

In order to measure the current through any component, it is the same concept as voltage, except you click on the desire component, instead of the node. When you hover over the component you want to observe, a

little emblem will appear with an arrow. This will tell you the direction of current the software is going to measure. If this is wrong, it is explained after this how to change it. Plotting all the currents should look something like this:



Graph 3. All currents in the circuit

Now when we plot the current for this, we want to assume the current is flowing in a certain direction to ensure our data is accurate. Let's assume the path of current flows according to the arrows below in Figure 14.

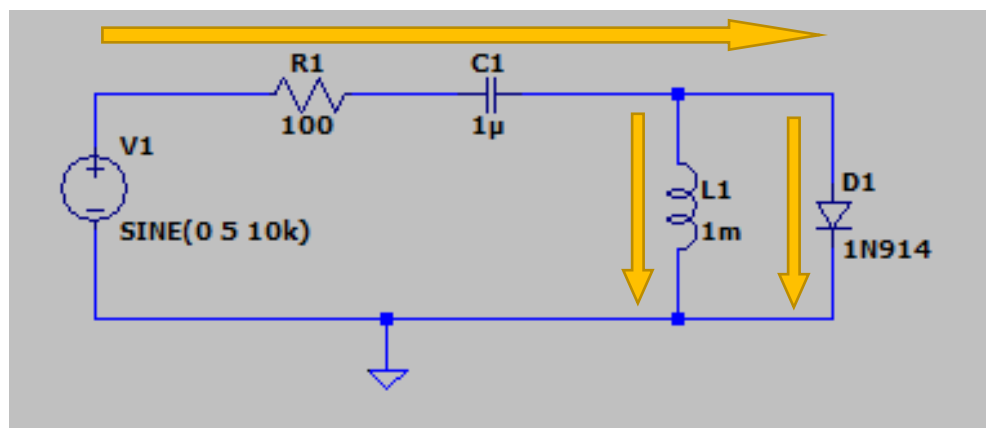


Figure 14. Assumed direction of current flow

With this assumption, if we go to measure the current through R1 or C1, it will point the arrow to the left, which is not what we want. The easiest way to fix this is by plotting the negative of this plot, since we want to know the current going the other way. To do this, plot the current normally, then on the graph where it says I(desired component), right click. This window will appear:

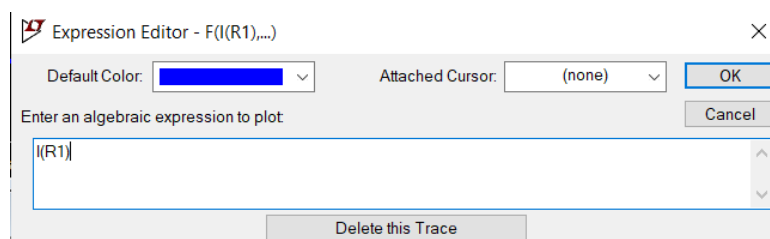
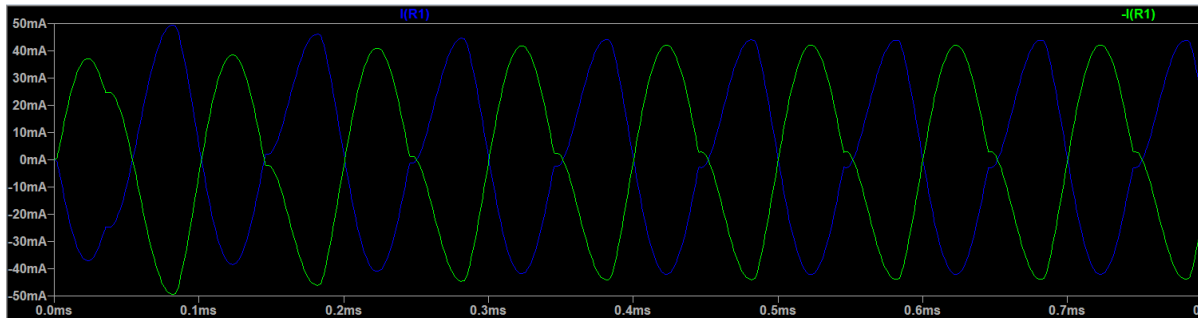


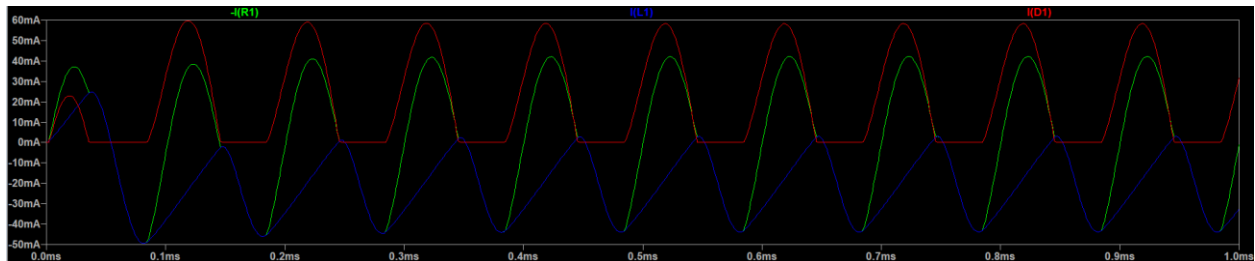
Figure 15. Plot options window

Then change I(desired component) to -I(desired component). For example, we want to change R1 in this case. So, change I(R1) to -I(R1) and your result should change to something like this:



Graph 4. Current graph for R1 (both positive and negative)

So, since the software plots R1 assuming the current is moving from left to right (blue line), this change is not based on the assumption it is moving from right to left, which is what we want. So, the final current plots for this graph are seen in Graph 5.



Graph 5. Final graph with fixed current direction

## Part 3: LTSpice Features

Parts 1-2 are the basic concepts to help you setup a schematic, part 4 is designed to give you more in-depth explanations along with some other features that LTSpice offers.

### Types of Sources

LTSpice has a variety of different sources you can use. These are the most common ones that are used along with an explanation of the parameters.

## DC

Basic direct current voltage source that simulates a simple battery and allows you to specify the voltage value. The only parameter required for this is your DC voltage value

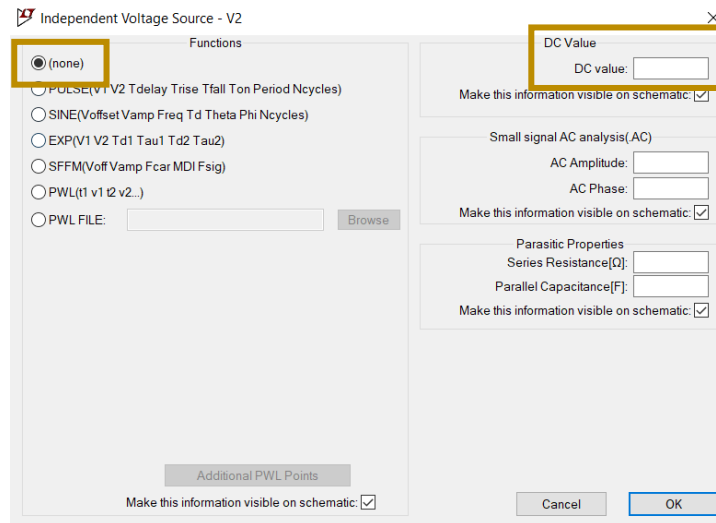


Figure 16. DC parameter window

## PULSE

Creates some sort of pulse that you need to simulate. Usually used to create a square wave to test sudden spikes in voltage.

Figure 17. Parameter window for pulse voltage

Vinitial	LOW voltage level
Von	HIGH voltage level
Tdelay*	Delay time from 0 to start
Trise	Rise time from LOW to HIGH
Tfall	Fall time from HIGH to LOW
Ton	Time signal is high
Tperiod*	Time for one cycle
Ncycles*	How many times it runs

\*Not required

Table 1. Collection boundaries and their meanings needed for pulse voltage

## SINE

Creates a basic sinusoidal wave.

Figure 18. Parameter window for sine voltage source

DC Offset	Offset voltage
Amplitude	Amplitude voltage
Frequency	Oscillation frequency
Tdelay*	Delay time from 0 to start
Theta*	Damping coefficient
Phi*	Phase difference
Ncycles *	How many times it runs

\*Not required

Table 2. Collection of boundaries and their meanings for sine voltage

PWL (Piecewise-Linear)

This is commonly used for creating your own linear functions. You can give it a set of points and it will create a function based on those that are connected linearly. This is very common in industry to test devices hold up to specific requirements. This may seem a little confusing, so the output of the source created in the outline is below.

Functions

☐ (none)

☐ PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

☐ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

☐ EXP(V1 V2 Td1 Tau1 Td2 Tau2)

☐ SFRM(Voff Vamp T Car Width Fsig)

☒ PWL(t1 v1 t2 v2...)

☐ PWL FILE:

time1[s]:

value1[V]:

time2[s]:

value2[V]:

time3[s]:

value3[V]:

time4[s]:

value4[V]:

Make this information visible on schematic: ☒

DC Value

DC value:

Make this information visible on schematic: ☒

Small signal AC analysis(AC)

AC Amplitude:

AC Phase:

Make this information visible on schematic: ☒

Parasitic Properties

Series Resistance[Ω]:

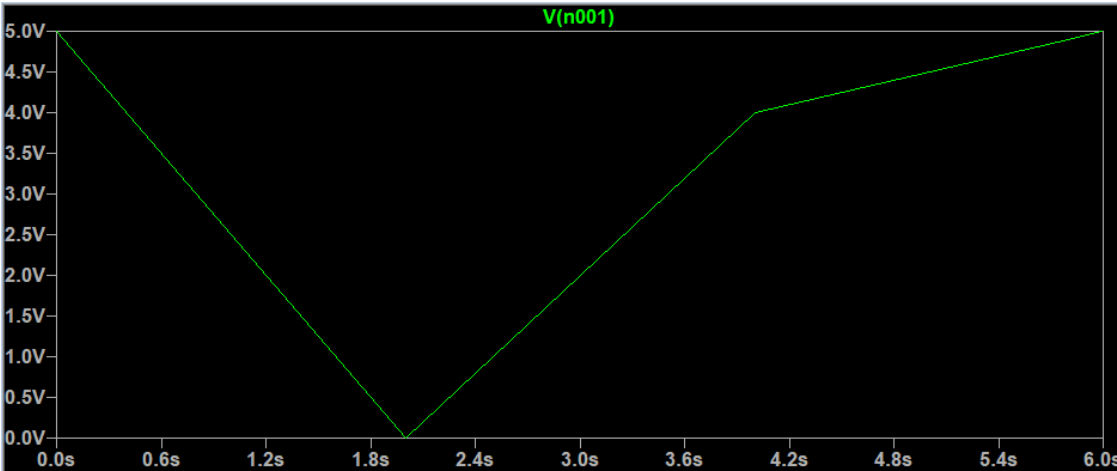
Parallel Capacitance[F]:

Make this information visible on schematic: ☒

Figure 19. Parameter window for PWL voltage with example numbers

time_	n time
value_	voltage change at n time

Table 3. Boundaries and their meanings for PWL



Graph 6. Graph of example PWL function

## PWL FILE

This source is a regular PWL, except you can upload a file with the data you want. This must be a .txt file and each line must be a new point in the data, with a space between the time and voltage. This can be used for when you are running the same tests on multiple different circuits. So instead of entering the data every time when you go to test a circuit, you can upload the .txt file with the PWL information already set (These files are usually 50+ lines). Using the example from above, we can type the text below into a .txt file, and upload it for the same result:

0 5

2 0

4 4

6 5

In order to upload this file, Click the PWL FILE and browse to find the desired file you want to use.

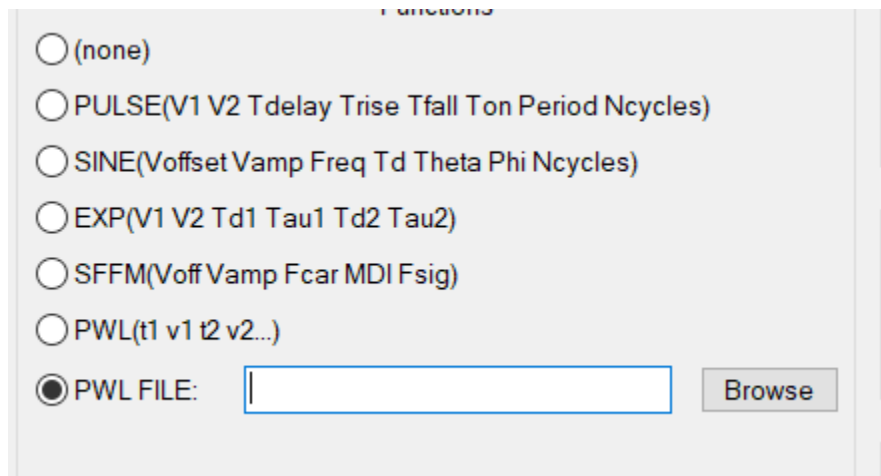


Figure 20. Uploading PWL FILE

## Types of Simulation

### Bias Point

The Bias Point analysis is the starting point for all analysis. In this mode, the simulator calculates the DC operating point of the circuit.

### DC Sweep

DC Sweep analysis varies a voltage source over a range of voltages in an assigned number of increments in a linear or logarithmic fashion.

### AC Sweep/Noise

The AC Sweep/Noise analysis varies the operating frequency in a linear or logarithmic manner. It linearizes the circuit around the DC operating point and then calculates the network variables as functions of frequency. The start and stop frequencies as well as the number of points can be assigned.

### Time Domain (Transient)

The Time Domain (Transient) analysis is probably the most popular analysis. In this mode, you can plot the various outputs as a function of time. The starting and ending times for the various plots can be input. The accuracy (smoothness) of the output plots can also be controlled by regulating the maximum (time) step size.

### Monte Carlo Analysis

When using components in the real world, they are not ideal. They all have some sort of tolerance even though this software assumes all ideal components. So, to simulate this, there is an embedded feature to do a Monte Carlo analysis. What this does, is it allows you enter the tolerance of each component, then you can run this analysis and it will choose a random value within the tolerance of your component. For example, if you have a  $100\Omega$  resistor with a 5% tolerance, it will choose a random number between 95-105 $\Omega$ . Then you can run this analysis numerous times (usually 100+ times) to ensure you get a variety of outputs, and they are all within your desired output.

To do this, we are going to go back to the example we did earlier, but without the diode in it:

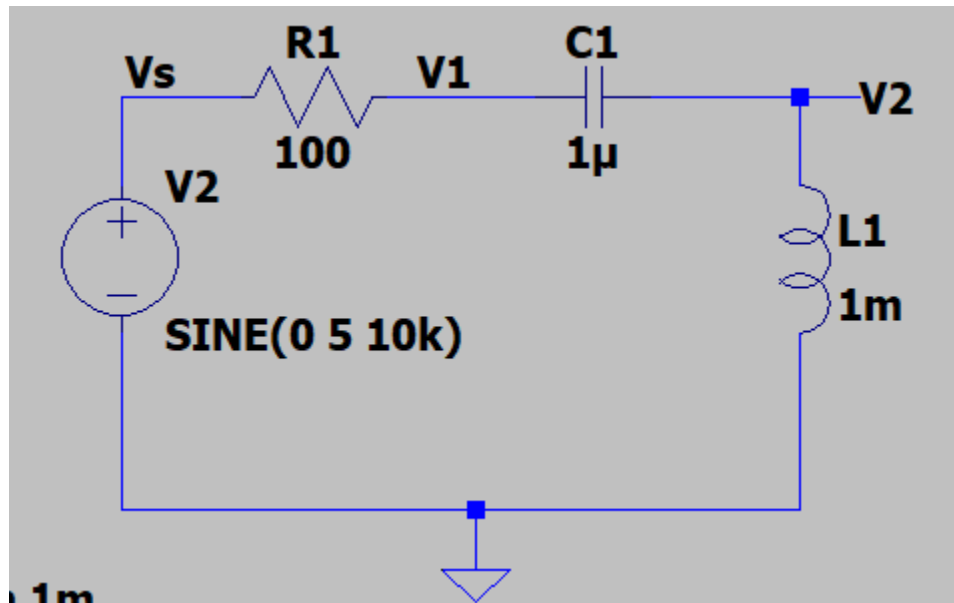


Figure 21. Circuit for Monte Carlo Analysis example

Then, to setup the analysis, we need to tell the software what components we want to vary, along with how much. To do this, we need to change the value of the components to the format of {mc(value, tolerance)}. This means our resistor will change from 100 to {mc(100,5)}, assuming 5% tolerance.

Let's use tolerance values of 1% for R1, 1% for L1 and 1% for C1 (These are pretty unrealistic, but it shows this analysis better for this circuit). Assuming our voltage source is ideal.

After setting up all the values to run the Monte Carlo, we should have something that looks like Figure 22.

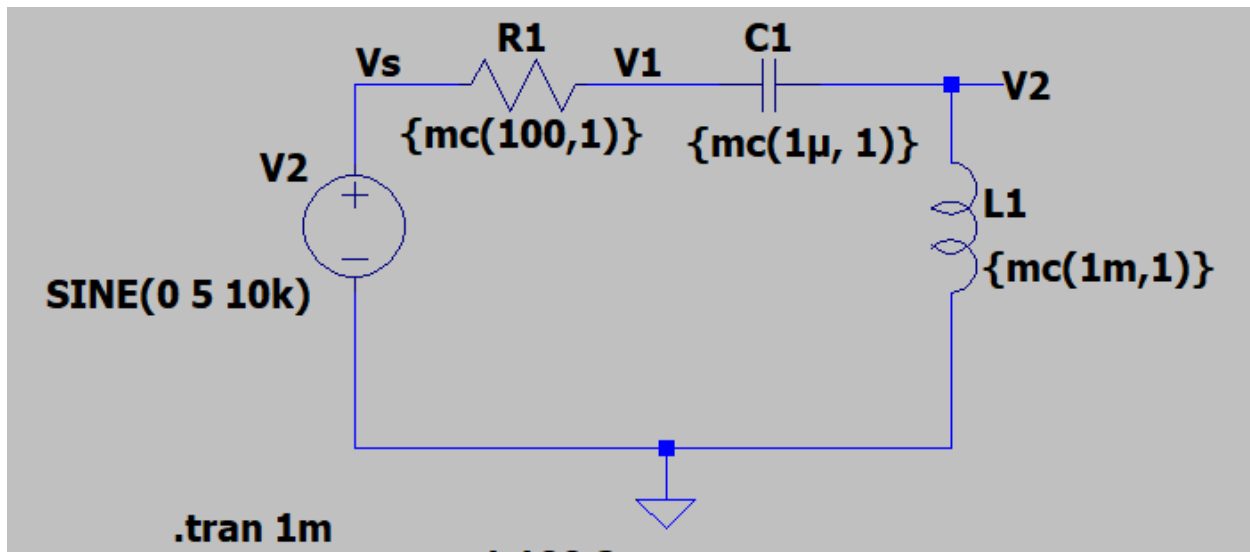


Figure 22. Circuit set up with tolerances for analysis

Now that we setup all of the components for the simulation, we need to tell the software that we want to run the simulation, and how many times. To do this, we are going to use a directive. A directive tells the software what we want it to do when we click “run”. For example, “.tran 1m” is a directive that tells the software to run a transient analysis for 1mS. So, we need to add another directive to setup the Monte Carlo analysis. You can add this by going to Edit > Spect directive. Then use directive “.step param run 1 10 1”. Next, place that directive somewhere on the schematic (I would choose near the .tran command to keep it simple and have them together). This is telling it to run from 1-10, stepping in increments of 1. That is the .step 1 10 1 part of it. Then the “param run” part is the software knowing to run the Monte Carlo analysis that we specified. The final schematic should look something like:

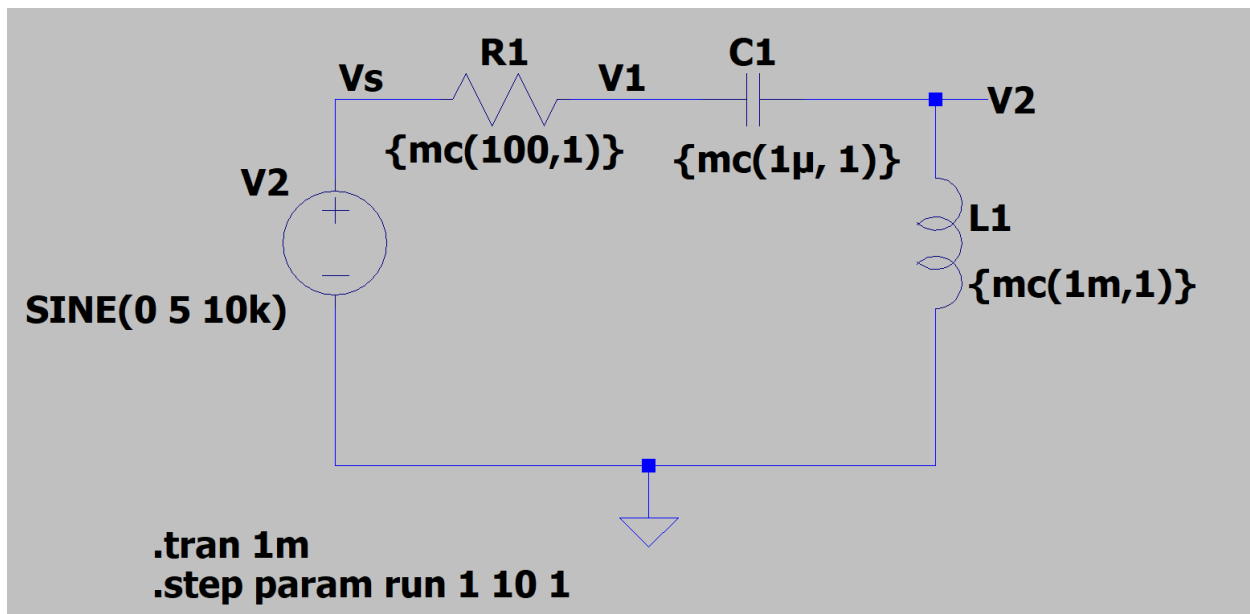
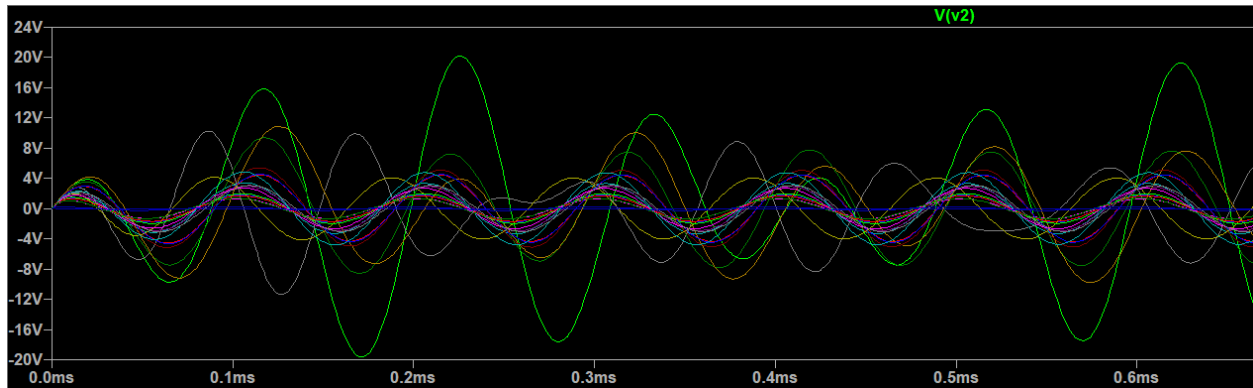


Figure 23. Final schematic setup

Then the output (V2) should look something like this:



*Graph 7. Final graph of Monte Carlo analysis*

If yours does not match this, that is fine. It randomly selects the values, so it should be slightly different.

This shows how much this signal can vary even with only a 1% tolerance in 3 components. Additionally, it shows how important this analysis is to run on a circuit you are going to be mass producing.