RADIO-FREQUENCY CHARACTERISTICS OF COMPONENTS

Lab #1

The purpose of this lab is to become familiar with the real-world high frequency performance of capacitors, inductors, and resistors. You will also become familiar with impedance measurements using an impedance analyzer (Agilent 4294A).

You should review the video demo online before coming to lab. This supplements the instructions in this handout and may help you navigate the instrument control and software.

Important note: as you go through this lab, be sure to take notes in your lab notebook about what you are doing, how you are doing things, what devices you have measured, what filenames you use, etc. The techniques we use here will be used in future labs, so the more carefully you document your work here, the easier it will be in the future.

Calibration (Fixture Compensation):

The impedance analyzer measures the characteristics of components by applying signals (e.g. an AC voltage signal) to the device and measuring the response (e.g. current) that results. However, to get from the terminals of the instrument (4 BNC connectors) to the terminals of the device to be tested, a test fixture (in our case, the model 16047E test fixture) is used. At high frequencies, fixtures like this have internal parasitics that need to be compensated for so that the measurement can show us just the performance of the device under test (instead of a combination of the device and the test fixture). In order to accurately determine the parasitics associated with the test fixture, measurements of at least two known impedance standards must be made. The firmware of the impedance analyzer assumes that the standards to be used are an open circuit and a short circuit. To perform this compensation:

1. If there is a component in the fixture, remove it and gently close the connector blocks with the connector tension knobs on the front of the fixture
2. Press the "Cal" button on the front panel, followed by the "Fixture compensation" softkey, followed by the "open" softkey (be careful; the "open" softkey is the 4th from the top; the "open on/off" softkey does something else). The instrument measures the impedance it sees at its terminals over the full range of measurement frequencies, and stores the results in internal memory.
3. Loosen the thumbscrew on the fixture to detach the shorting bar from the back of the test fixture. Adjust the connector tension knobs to open the jaws wide enough to fit the shorting bar in place, and then lightly tighten them on the bar. Only light finger pressure is needed.
4. Press the "short" softkey (5th from the top), and the instrument measures the impedance it sees at its terminals.
5. Remove the shorting bar from the test fixture terminals, and replace it in its holder under the thumbscrew.

After this procedure, the impedance analyzer automatically applies a numerical correction to all subsequent measurements, so that it displays the component impedances with the fixture parasitics removed.

Measurement Setup:

Before collecting data on the components of interest, we need to first set up the analyzer to perform the measurements over the frequency range of interest. For this lab, we will measure from 10 kHz to 10 MHz, using a logarithmic frequency sweep with 301 points. To set this up:

1. Press the "Start" button on the front panel, and enter 10 kHz by typing "10" on the keypad, and pressing the "k/m" button to indicate kHz.
2. Press the "Stop" button, and key in "10" on the keypad and press the "M/μ" button to indicate MHz.
3. Set the sweep type to logarithmic by pressing the "Sweep" button, and confirming that the "Type" softkey is set to "[LOG]". If it is not, press the "type" softkey and select "log".
4. Press the "Number of points" softkey, key in "301" on the keypad, and press "x1" to set the number of points
5. As a compromise between accuracy and measurement time, press the "Bw/avg" button followed by the "Bandwidth" softkey and select "3" (setting "1" is fast but very noisy, "5" is very accurate but quite slow; "3" is often a good compromise).

The analyzer is now set up to take data on the various components that we can analyze and model.

Component Measurement:

With the analyzer configured, we are ready to measure a selection of simple components (capacitors, inductors, and resistors) in order to see what their real-world performance characteristics are.

1. Capacitors

Capacitors have series lead inductance and resistance as well as internal resistive losses caused by the finite (i.e. non-zero) conductivity of the dielectric material. The losses in capacitors are often small, so that they can be modeled approximately as a series LC circuit. As a consequence, a capacitor is expected to have a "series resonant frequency" above which the impedance will be inductive[1]; at resonance, the impedance of an ideal series LC circuit is zero. We will be looking for this frequency, as well as some other nuances in the real-world performance of these components.

Measure the impedance vs. frequency of two capacitors with values in the 0.01 μF to 0.001 μF range:

a. Choose a capacitor, and insert it in the test fixture (use only light pressure on the knobs). The analyzer will scan the frequency and display the results. Internally, the analyzer measures the impedance of the device being tested, but it can display (by doing simple math) the results in many different ways. The display options are shown by pressing the "Meas" button. The "A" and "B" buttons in conjunction with the softkeys under the "Meas" button menu allow you to select what data each trace will display (A=yellow, B=blue). It is good practice to make sure the component has the expected general trend; this is usually most easily done by looking at the $|Z|$-θ, Y-

[1] It is common to refer to impedances with negative phase angles as capacitive, and those with positive phases as inductive. This can be rationalized by noting that the sign of the angle for $Z=1/j\omega C$ is negative, and that for $Z=j\omega L$ is positive.

θ, R-X, or G-B displays. These options show the impedance (Z) or admittance (Y) in either polar or cartesian formats.

b. For an ideal capacitor Z=1/jωC, so a plot of the susceptance should be a straight line with frequency, at least at low frequenices. This is most easily seen by choosing the G-B format ("Meas", "more 1/3", "G-B"), and looking at the susceptance, B. For a series L-C circuit (including the series inductance), a plot of reactance vs. frequency should go through zero at the resonance frequency. An easy way to see this is in the R-X display ("Meas", "R-X"), and looking at the reactance, X. This is just a quick sanity check—we will do a more careful analysis later. Do not be too concerned if you do not see a resonance, especially for small-value capacitors; it may occur at a frequency higher than the measurement range.
c. When you are satisfied that the data appears reasonable, you can transfer it from the instrument into the computer using the at4294aread.m Matlab script (you may need to download "at4294aread.m" from the course web page). Invoking this as "[x,y]=at4294aread;" will put the measurement frequencies into the Matlab vector "x" and the measured impedances (complex) into the Matlab vector "y" (you should choose more descriptive variable names). You can then plot these (or analyze them further) using the usual Matlab commands.
d. It is a good idea to save the Matlab workspace after a measurement; if a measurement fails it may be necessary to quit and restart Matlab.
e. Repeat this for the second capacitor. Be sure to use different Matlab vectors to store each component's measurement results (otherwise, it will overwrite them and you will be unhappy).

2. *Inductors*

Inductors have capacitance between turns of the coil and resistance due to the finite conductivity of the wire and loss in the magnetic core material. The loss is (hopefully) small, but can have important effects on circuit performance. Inductors can often be modeled (roughly) as a series RL circuit in parallel with a capacitor, although often additional elements are needed for accuracy. Because of the parallel L-C structure, they exhibit a "parallel resonance frequency" at which the impedance becomes very large. This frequency, sometimes also called the "self-resonant frequency" is an important parameter of high-frequency inductors.

Measure the impedance as a function of frequency for two inductors between 10 μH and 300 μH. To do this:

a. Install an inductor in the test fixture and check that the measured impedance is as expected. Since the impedance of an ideal inductor Z=jωL, the reactance should (ideally) be linear in frequency (at least at low frequencies at which the parasitics are not significant). This is most easily seen in the "R-X" mode. The onset of the parallel resonance should lead to a peak in impedance and a singularity in reactance; this can often be most easily seen by looking at |Z| (in |Z|-θ mode).
b. When you are satisfied that the measurements are correct, transfer them to Mmatlab as previously, and repeat for the second inductor.

3. *Resistors*

The general model for RF performance of resistors consists of an inductor (from the leads) in series with a parallel R-C circuit. For low resistance value resistors, the effect of the parallel capacitance is often small and the resistor can be modeled (approximately) as a series R-L circuit; for large resistances, the parallel capacitance can dominate at high frequencies, leading to a circuit model that looks (approximately) like a parallel R-C circuit.

Measure the impedance vs. frequency for a 47 Ω resistor and a 10 kΩ resistor.

a. Perform the measurement just as you did for the inductors and capacitors. For the 47 Ω resistor, the equivalent circuit suggests the impedance should resemble a series R-L circuit; confirm this (the R-X display is the most natural view for this; for an ideal series R-L circuit, R should be constant and $X=\omega L$ should be linear in frequency). For the 10 kΩ resistor, we expect something more like a parallel R-C circuit. For an ideal parallel R-C circuit, the G-B display (where $G=1/R$ and $B=\omega C$) is a natural choice.
b. As before, transfer the measured data for each component to Matlab.

If everything has gone right, you should now have separate measured impedance vectors for each of the components tested.

Component Analysis & Modeling:

We have measured the components, and have seen (at least roughly on the instrument display) how they deviate from the "ideal" behavior due to parasitics. But so far we have not been quantitative about the parasitics, and have not put this into a framework that could be used for circuit design or analysis. For that, an equivalent circuit model is needed.

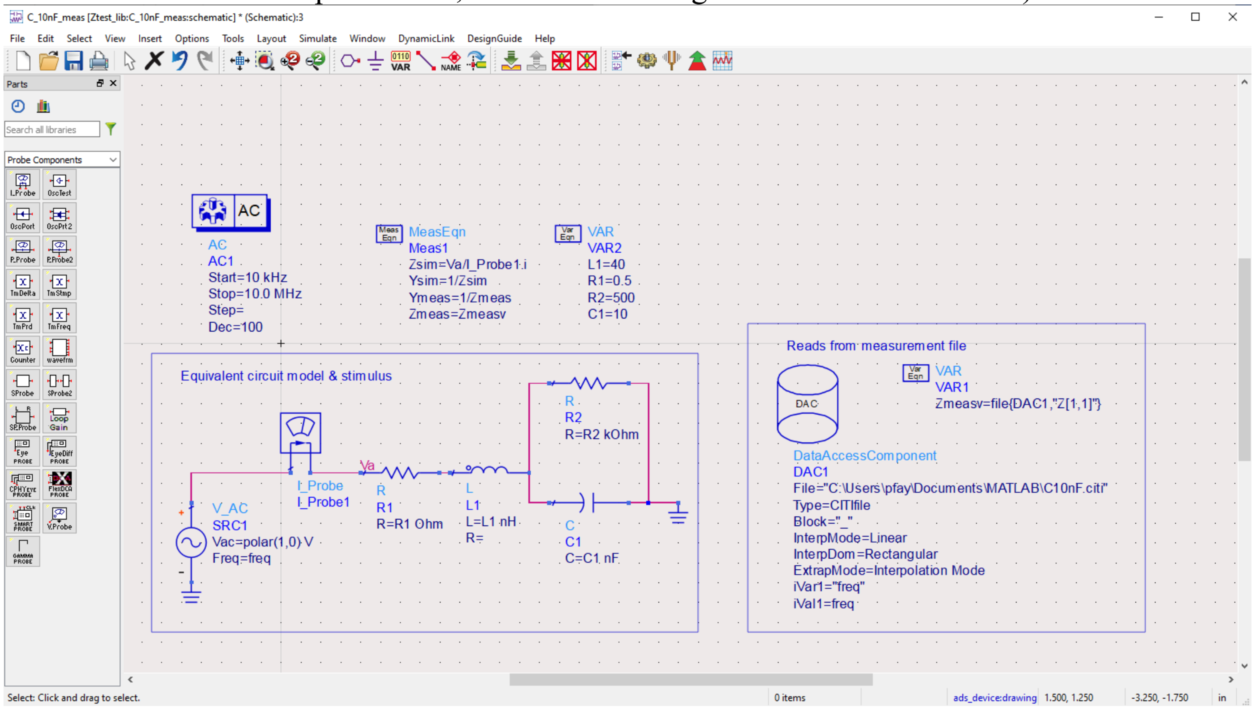
We will use the ADS software package to help develop models for each of these components. ADS cannot read matlab files directly, but a script is available to convert the Matlab vectors into an ADS-readable file. The script "citiwrite(x,y,'filename.citi')" will make a file ("filename.citi") using the vector x as the frequency variable and y as the impedance (you may need to download citiwrite.m from the course web page). Run this for each of your component measurements so that you have separate .citi files for each. We now have the measured characteristics of each of the components in a format that our circuit design CAD software can read.

To be useful for circuit design, a model for each of the components is needed. In this context, a model is just a mathematical representation of the circuit component that we (or our CAD software) can use. A convenient starting point for many devices is an equivalent circuit model, where we build up the overall model of the component from "elemental" circuit elements (e.g. our model of a real-world device consists of a network of ideal R, L, C, etc.). This is the approach we will take here. Our basic approach will be to import the measured data from our devices (real-world capacitors, inductors, and resistors from above) into ADS, and obtain an equivalent circuit model for each that (at least approximately) matches the measured behavior.

1. *Capacitors*

The first step is to get the measured data into the simulator, and to create a starting point for an equivalent circuit model. The screenshot below shows one way to do this, starting from a .citi file of the measured data. Start ADS, and when/if it asks about licensing, indicate that you want to specify a network license server and enter 1712@linux3-lic.crc.nd.edu.

a. The "DataAccessComponent" in the lower right can be found in the "Data Items" category of the "Parts" window (left side of the screen). Change the "File" entry to point to your file for one of your capacitors, and set the iVar1 and iVal1 as shown (watch the quotes carefully). You may find it easier to make these entries by double-clicking on the DAC component (this will open a window that lets you enter the parameters, rather than editing them in the main screen).



Screenshot: Capacitor measurement data and initial equivalent circuit.

b. The “Var Eqn” in the lower-right box is used to tell ADS how to pull data from the file into a variable in ADS. The “Var” element can be entered by clicking on the element from the tool bar, or from the menu with Insert/Var. Enter the formula as shown, being careful about the use of curly brackets, capitalization, and quotes (ADS is picky about syntax). We now have the measured data available in the simulations.

c. To develop a model, we need an equivalent circuit representation. The circuit on the left shows one possible option. R1 represents series resistance, L1 represents lead inductance, C1 is the capacitance, and R2 represents dielectric loss. These are ideal circuit elements, and can be found in the “Lumped element” category of the “Parts” window. When entering these, I recommend putting units on each variable as shown (you could also define the variables with units, but that tends to be more confusing).

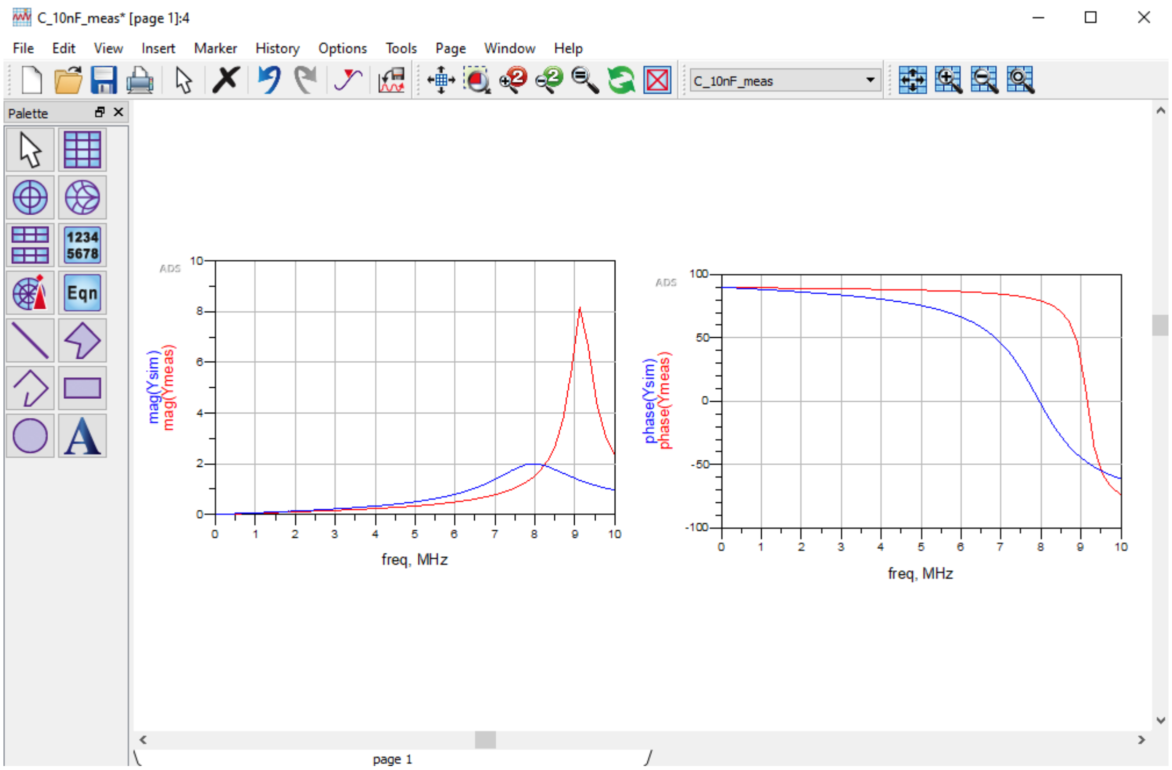
d. To assign values to each of these elements (instead of hard-coding in values) use another “Var” element as shown. To put multiple variables in a Var element, double click on the “Var” element, and type in the name for a new variable. But do not hit enter—hit “add” and this new one will be added to the list. The values shown are (at this point) just educated guesses.

e. To simulate this circuit, we need to define the frequency range of interest. Add an “AC” simulation (under the “Simulation-AC” category of the “Parts” window), and double-click on it to open a window that lets you set the start frequency to 10 kHz, the stop frequency to 10 MHz, the sweep type to “log” and the Pts./decade to 100 (this will match the 301 point sweep we used for measurements).

f. To drive the equivalent circuit, we use the ac source "V_AC" (found in the "Sources-Freq Domain" category of the "Parts" window. The Vac=polar(1,0) V expression in the V_AC source sets the voltage to 1 V, with phase angle of 0; Freq=freq sets the frequency of the source to match our simulation frequency. The I_Probe (found in the "Probe Components" category of the "Parts" window) forces the simulator to record the current in its branch. We need this to compute the impedance and admittance of the test circuit (since Z=V/I). Finally, to record the voltage at the terminal of the equivalent circuit, add a node name to the circuit. You can do this by using the menu ("Insert"/"Wire/Pin label") and clicking on a node and typing in a name (Va in my example). This makes the voltage at this node available as a variable for further calculations or analysis.
g. This is sufficient for the simulation, but it is more convenient if we have the simulator do a few preliminary calculations for us (as it is now, the simulator would compute the voltages and currents, but not the impedances or admittances). We use the "Meas Eqn" box for this. The "Meas Eqn" is in the "Simulation-AC" category of the "Parts" window. Formulas are entered in a similar way to the "Var" element. Enter the formulas shown in the screenshot. The Zsim and Ysim measurement equation expressions compute the impedance and admittance of the equivalent circuit model. The Zmeas and Ymeas expressions make a copy of the measured data in a format that is convenient for comparing to the simulated response. The simulator treats "Var" and "Meas Eqn" elements in slightly different ways internally, so we need to use both in this case.

With the framework set up, we can now simulate the equivalent circuit, and compare it to the measurements. Click the "gear" icon in the toolbar to run a simulation. This will run the simulation, and (usually) open a blank display window to show the results.

h. Since this is a capacitor (ideally $Z=1/j\omega C$, $Y=j\omega C$), the admittance will be more convenient to look at (the $1/\omega C$ dependence leads to large values compressed very close to the y-axis for impedance). Make rectangular plots (click on the "grid" icon in the palette on the left) and add Ymeas and Ysim to the plot. Since these are complex quantities, you'll need to choose magnitude or phase to make the plotted value real. The results for my example are shown below.

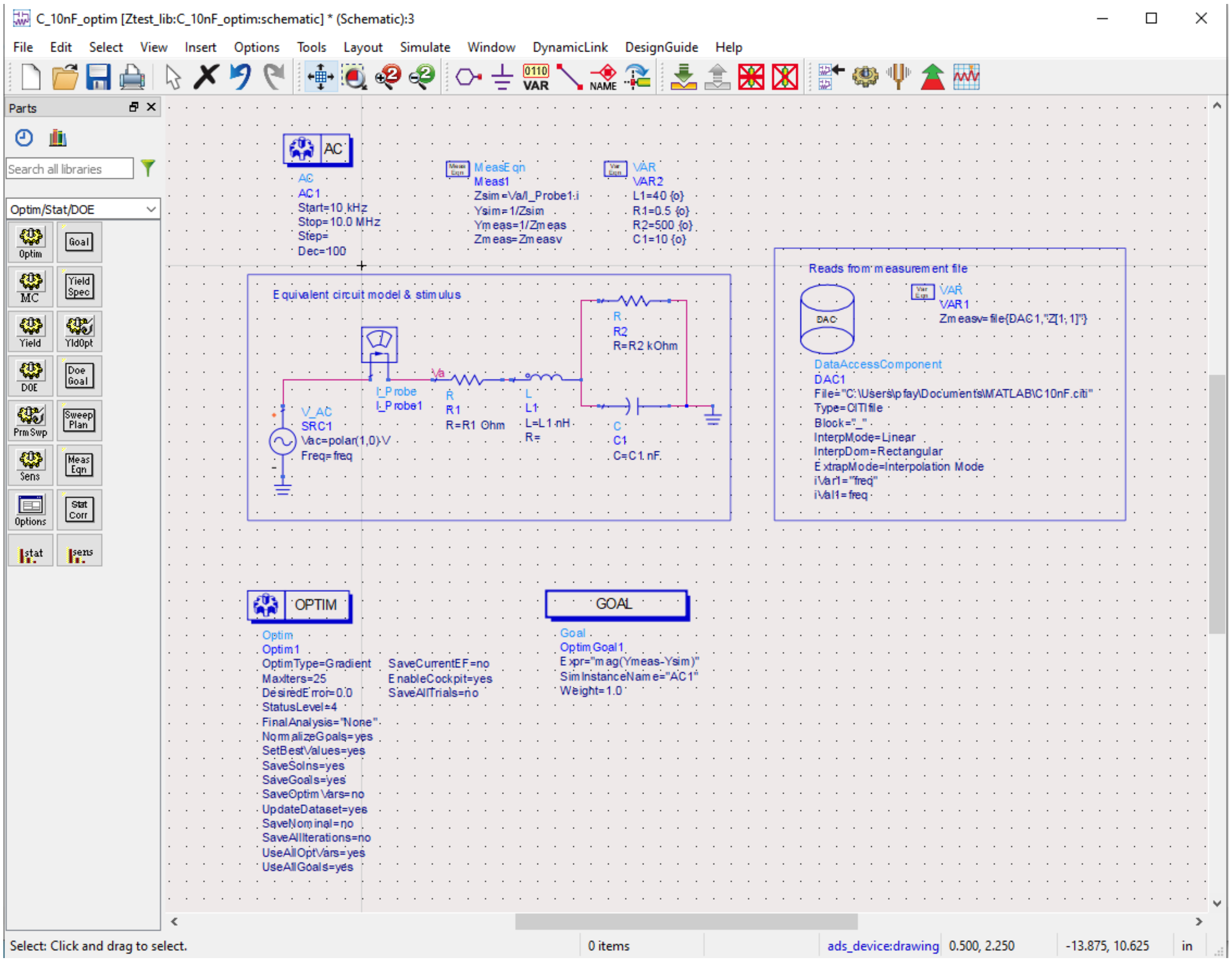


Screenshot: Initial capacitor model compared to measurement.

i. You can see that the measurements (red) and equivalent circuit response (blue) are "similar" but not a particularly close match. This is really just an indication that our initial educated guesses about the equivalent circuit parameters are close, but not quite there. If the response was completely different (no similar trends, etc.) that might suggest either our basic equivalent circuit topology was not appropriate, or our parameters were way off.

To find parameters that best match the measurements, we could make additional guesses and see the results. This can be useful to help "seed" our intuition about how the parameters affect the equivalent circuit performance. But for a case like this where the basic trends are about right but the parameter values are just off, there is a better way. ADS has a built-in optimization feature that allows tuning of parameters to match goals. The screenshot below shows how the simulation framework from above can be modified to include an optimization.

j. From the "Optim/Stat/DOE" category of the "Parts" window, add an "Optim" element. Double-click on it, and change the "Optimization type" to "Gradient." Also check to be sure "Use All Goals in Design" is enabled/checked.



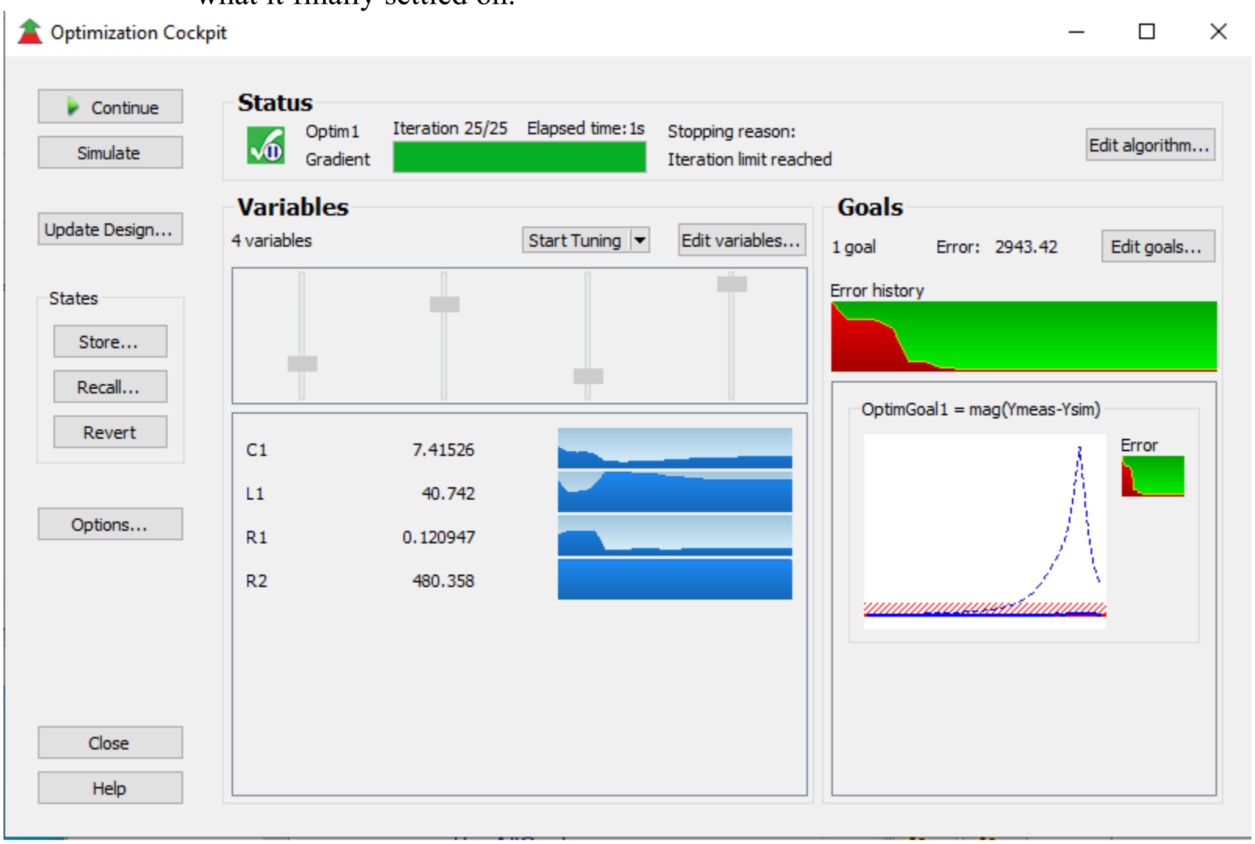
Screenshot: Capacitor with optimization enabled.

k. Also from the “Optim/Stat/DOE” category, add a “Goal” element. Double-click on the new Goal element to set it up. This item tells ADS how to assess how good (or bad) the current model is. The “expression” field is a mathematical expression that ADS will evaluate, and then use to decide how good of a solution it has. For a capacitor, a simple expression that captures what we’re likely to care most about is “mag(Ymeas-Ysim)”. This takes the difference between our measured and simulated admittances (complex), then takes the magnitude of the difference. If this is zero, we have a perfect match between measurement and simulation; if it is non-zero, the magnitude indicates how far away from a perfect fit we are.

l. Also in the “goal” element, select the analysis (probably “AC1” from the drop-down menu; this needs to match the AC element that sets the frequency range of the simulation). We also have to tell ADS what it should try to do with the results of the expression. Since our expression should go to zero for a perfect match, change the “type” to “<” (we want the expression to be less than a threshold) and set the Max to something small, perhaps 1e-3.

m. We now have to tell the simulator what parameters to vary, and by how much they are permitted to vary. To do this, double-click on the “Var” element that contains the L, R, and C parameter values. This opens a window that lets us set additional properties of the variables. For each variable (one at a time) select the variable (L1, R1, R2, C1) and click "Tune/Opt/Stat/DOE Setup." In the window that appears, click on the "optimization" tab, set the "Optimization Status" to "enabled," and enter minimum and maximum values for each parameter. There is a bit of intuition needed to select "reasonable" ranges for these variables. In this case, I selected a range from 10 to 50 nH for L1, 0 to 1 ohm for R1, 5 to 15 pF for C1, and 50 to 500 kOhm for R2. This will put an "{o}" after each variable on the main schematic view to indicate these are optimizable values.

n. Perform a simulation (using the gear icon) to confirm that the addition of the optimization elements did not change the circuit behavior. If it is OK (i.e., the equivalent circuit shows a "family resemblance" but not a sufficiently good match to the measurements), click on the "Christmas tree" icon (green and red up arrow in the toolbar) to start the optimization. This will open the (perhaps overly exciting-sounding) "optimization cockpit". The optimizer will try out different sets of parameters with the equivalent circuit, trying to drive the goal expression to zero. There is a graph (lower right) that shows (as a dashed line) the goal expression vs. frequency at the start of the optimization, and as a solid line the ending point, and also a red/green trend line for the error with each successive optimization iteration. If things have gone well, these graphs will tend towards a flat line near zero. You can also see how the optimizer varied each parameter, and what it finally settled on.



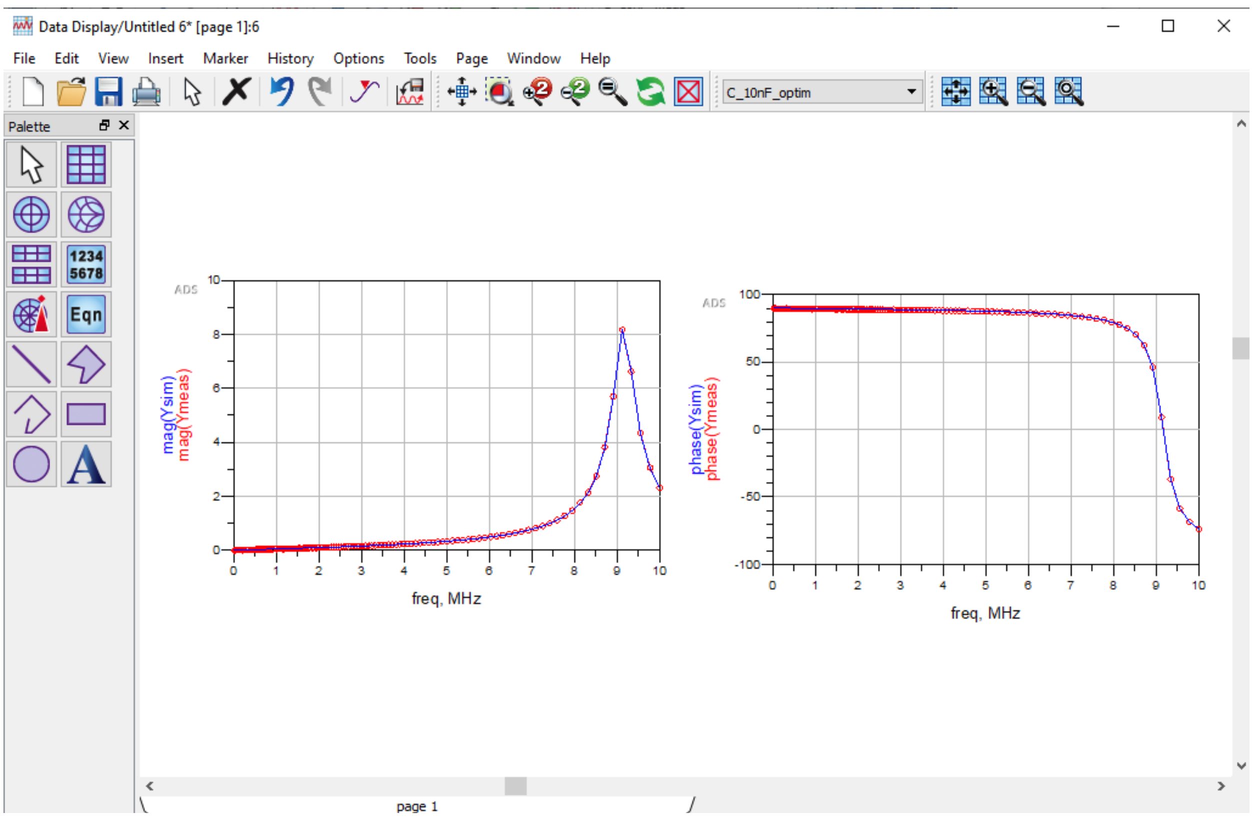
Screenshot: Optimization of capacitor equivalent circuit model.

o. If it looks like things "went OK" (i.e. the error is smaller, the parameters seem reasonable, there were not errors), click "close" in the optimization cockpit window, and choose "Update the design." This will put the optimizer's values in place of your original starting guesses in the "Var" element. If you aren't sure you like these values, click "Don't update"; it is hard to go back unless you memorized the old values.

p. To see how well the optimizer did, click the "gear" to do a simulation with the new, optimized parameters. The figure below shows how it worked for me. Note that I had to add symbols to the measured trace so that the difference between the measurement and simulation could be seen more clearly. Your model may not match quite this well, but it should be "pretty good." Make sure you save the ADS file so that you keep this equivalent circuit.

q. For the second capacitor, you can "save as" in the main schematic window, and just change the filename in the DAC (step a), parameter values (step d), and the ranges allowed for optimization (step m) to quickly arrive at optimized parameters. Be careful about which file you are modifying; it is easy to get mixed up and over-write the wrong file.

Reminder: be sure to document your work in your lab notebook.

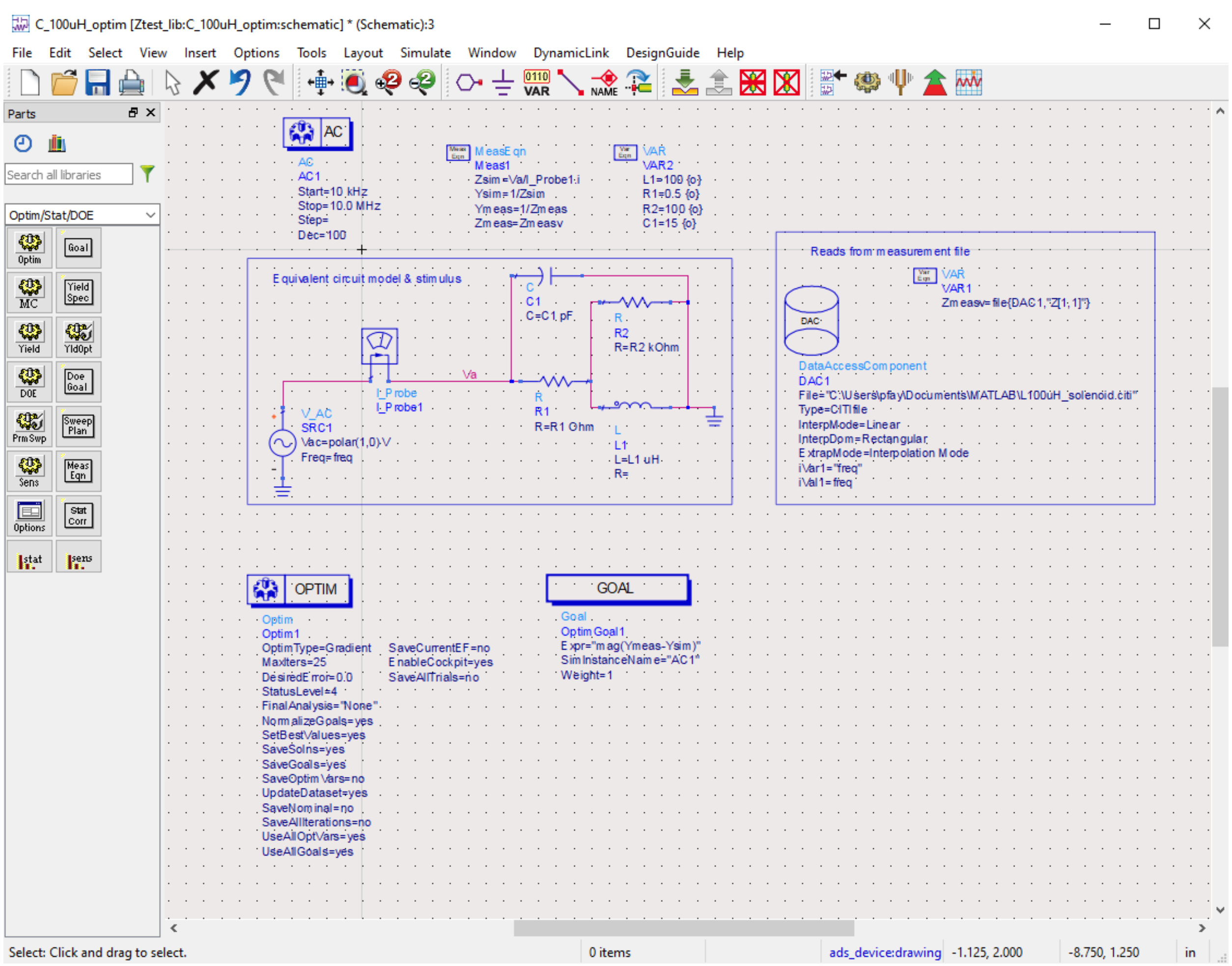


Screenshot: Capacitor model and measurement after optimization.

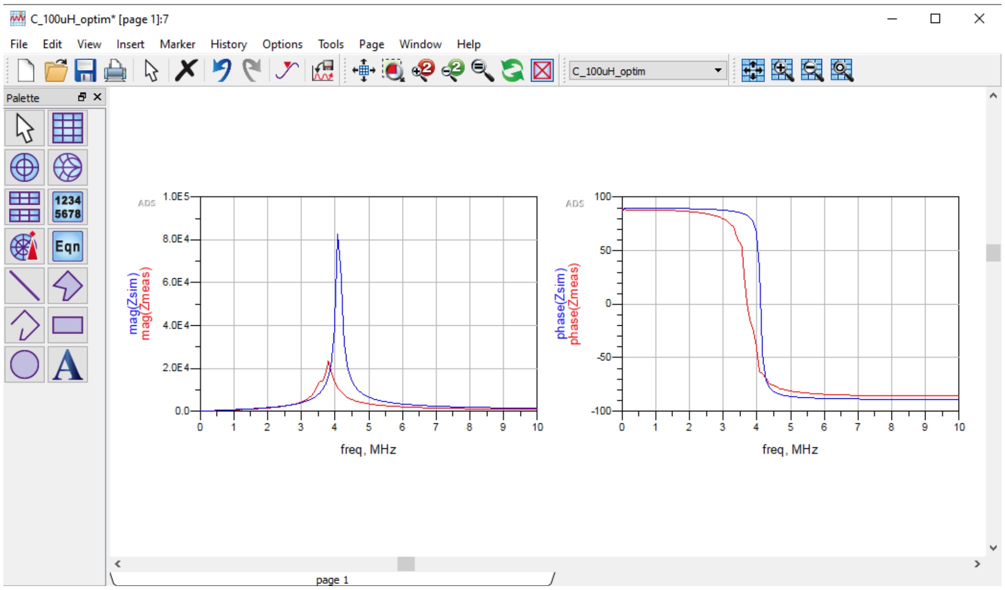
2. *Inductors*

The procedure for developing an equivalent circuit for inductors is very similar to that used above for capacitors; the main differences are the equivalent circuit topology (to capture the very different electrical behavior) and selection of optimization goal expressions.

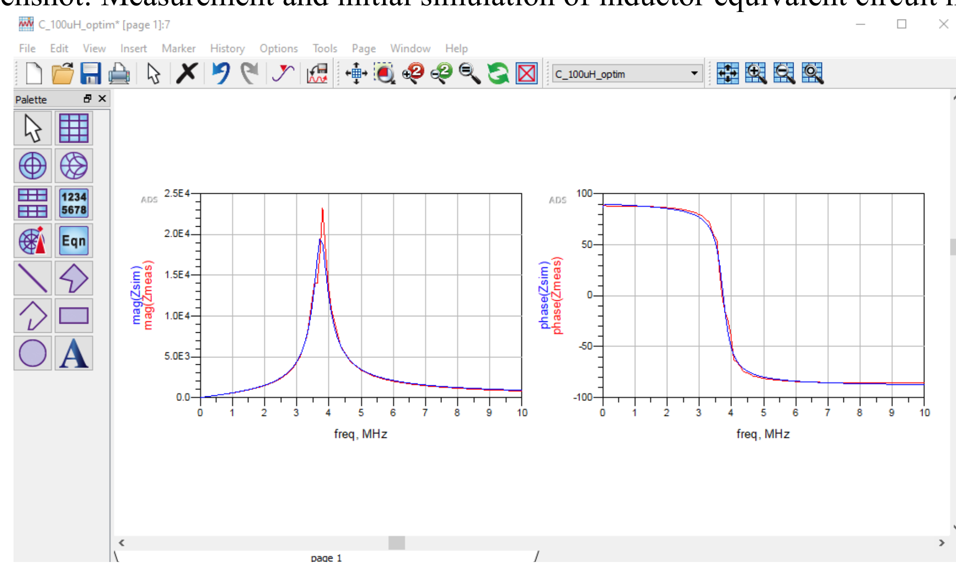
a. Using the same approach as above for capacitors, set up an optimization framework for the inductors you measured. The screenshot below shows a common topology for inductors (R1 is series resistance, L1 is the inductance, C1 captures the inter-winding capacitance, and R2 captures the loss in the magnetic core material).
b. Since an ideal inductor has an impedance of Z=jωL, it is often more convenient to work in terms of impedance than admittance. Using a goal expression of mag((Zmeas-Zsim)/Zmeas) will attempt to make the measured and simulated impedances match. In this case, a normalization to Zmeas is helpful so that the simulator does not over-emphasize the peak relative to the rest of the curve. As before, this is a "<"-type goal (we want it to go to zero).
c. Run a simulation to be sure that your initial parameter estimates are "in the ballpark." If they are too far off, the optimizer may not be able to find the correct solution. Once you are satisfied that your starting guesses are close, perform the optimization and check (with a simulation) to see how well the measurements and simulations match. An example before and after optimization is shown below.
d. Save the results, and repeat for the second inductor. Be sure to document your work.



Screenshot: Typical optimization setup for 100 μH inductor.



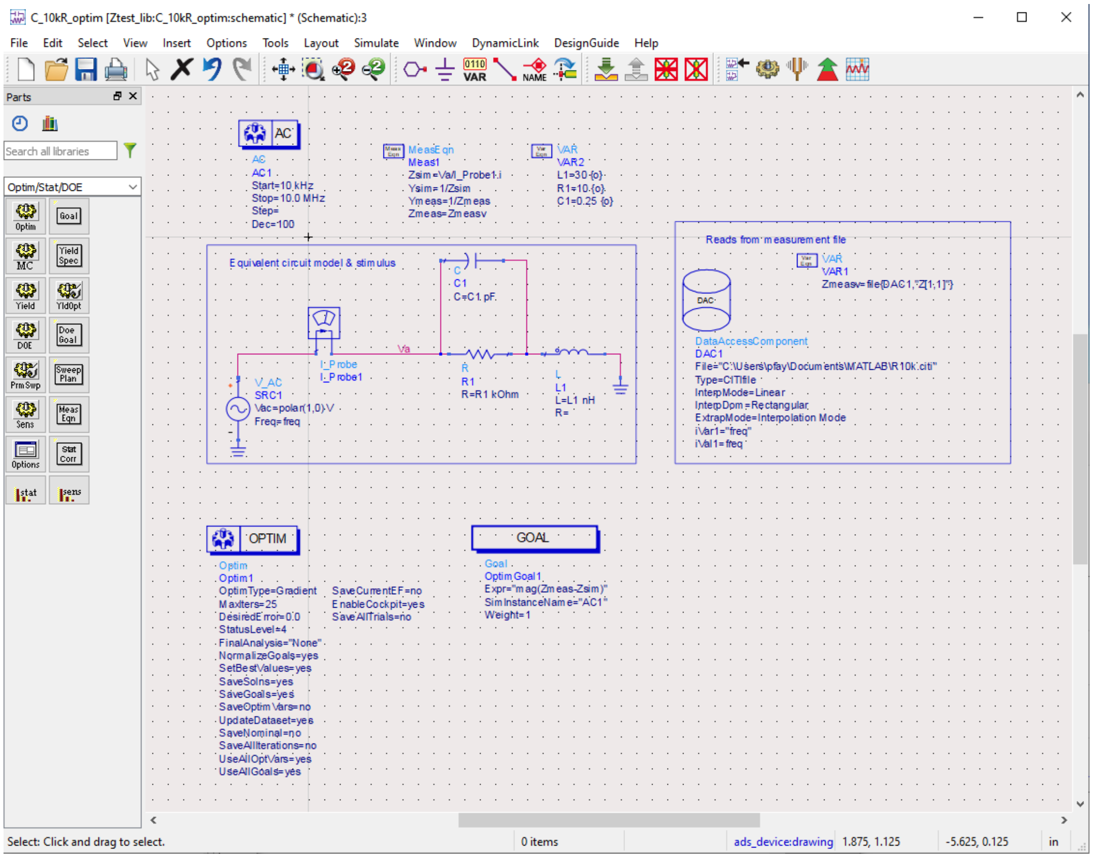
Screenshot: Measurement and initial simulation of inductor equivalent circuit model.



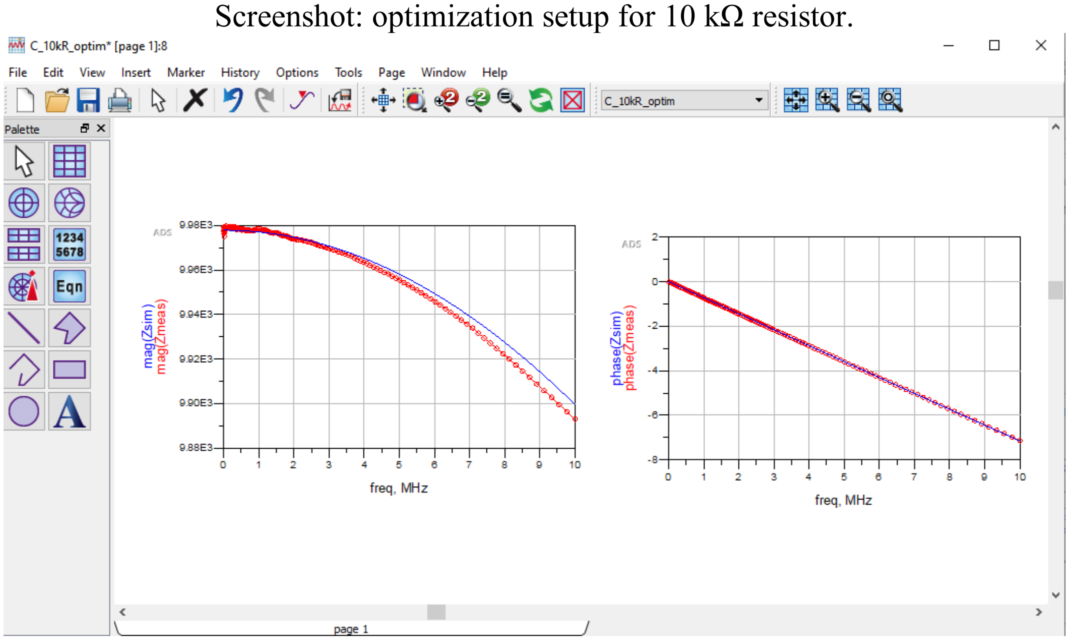
Screenshot: Measured and simulated inductor equivalent circuit model after optimization.

3. *Resistors*

The process for resistors is identical to the above examples. For resistors, a goal expression based on impedance often works ok. Using the techniques outlined above, develop models for the 47 Ω and 10 kΩ resistors you measured. A screenshot showing a model for a 10 kΩ resistor and after optimization is shown below for reference. Record your results and observations in your lab notebook.



Screenshot: optimization setup for 10 kΩ resistor.



Screenshot: Comparison of measurement (red) and model (blue) for 10 kΩ resistor. Note: the y-axis scale covers a very small range of values.

Analysis

We have developed equivalent circuit models for real-world capacitors, inductors, and resistors, and seen the significant effect the parasitic elements have on the impedance and/or admittance of the devices. Because this was a CAD-based simulation, we could use fairly complicated models to accurately represent the component behavior. For hand calculation, however, these models are unwieldy. Use ADS to compare the simplified equivalent circuits we discussed in class to the models you've developed. What aspects are captured, and what aspects do the simplified models miss? Be sure to document all of your measurements, analysis, and observations in your lab notebook.