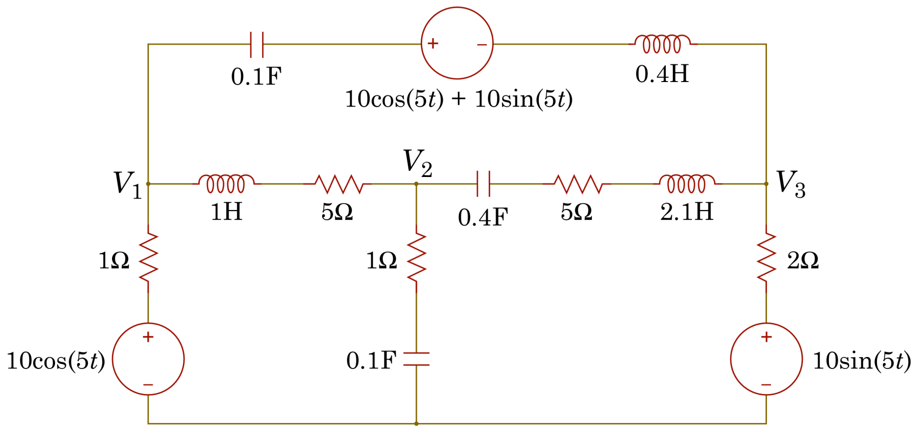
ECE 212 Spring 2017
Circuit Analysis II

MATLAB Exercises

NOTES

- Upload only PDF files.
- Combine PDFs at http://online2pdf.
- M files should have line numbers and a proper header. Print the Command Window as well (to PDF).
- Once you have a PDF to upload, go to the corresponding folder, and click "Add submission".
- Follow the directions, then click "Save changes".
- NEVER click "Submit assignment", as this means you are done, and you will not be allowed to change what you have uploaded.
- Instead, wait for feedback and the score. If you would like to change your solution at any time, click "Edit submission".
- Each exercise (M1 through M5) is worth 10 points. A perfect solution uploaded at any time earns 2 Mastery points. A perfect solution uploaded before 11:55pm on its Bonus date earns an additional 2 Bonus points.
- To maximize your first score, follow the suggestions made in the descriptions. In other words, it is not enough to get the answers correct. You must also use these applications properly and efficiently.
- Use short, meaningful names for all variables, such a V for voltage, I for current, Z for impedance, and S for complex power. Add extra symbols to distinguish them from each other, such as Ib for branch current and ZL for inductor impedance.
- When numbering elements or branches, a useful convention is "top-to-bottom, left-to-right". So, for example, in the first exercise (M1), the upper branch is #1, the two branches in the middle are #2 (on the left) and #3 (on the right), and branches #4, #5, and #6 are on the bottom (left to right).
- Each chart must have a meaningful title, which orients the reader and puts each chart in context. That is, the title should answer the question, "What makes this context different from every other context?"
- A proper check always returns a value of zero.
- Note that values not equal to zero but smaller than 10^{-13} are the remnants of floating-point errors, and may be taken to be equal to zero.
- In many cases, column arrays are best, especially when working with complex numbers, so use semicolons as delimiters. Column arrays are also useful when arrays are long, as in M4 and M5.

M1. Consider the following circuit.



(a) Transform the circuit into the "phasor" domain. (Use "amplitudes" not "effective" units.)

(b) Use nodal analysis to determine the phasors $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$.

(c) Determine the current in each branch. Show that KCL is satisfied at each node and that KVL is satisfied around each mesh.

(d) Determine the complex power supplied by each source, as well as the complex power absorbed by each impedance. Show that complex power is balanced.

Part (a) is done by hand. Extract the next page and add text boxes to submit your solution. Leave everything in rectangular form, i.e., don't use phasor form.

Parts (b) through (d) are done using MATLAB. Submit PDF files of your M file and your output window. Make sure that there are headers on both and that there are line numbers on the M file. Inefficient code will be marked wrong, so think about how to structure your code.

Note that MATLAB handles complex numbers very easily, as long as you remember not to put blank spaces in the wrong places! (You can also use commas as delimiters, if you prefer.)

Use one array to store the values of the three sources, and another array to store the values of the 11 impedances. Column arrays work well here, so use semicolons as your delimiter.

You should not do the algebra yourself; set up a matrix equation and solve it. It helps to define an array of six "branch" impedances using your array of individual impedances.

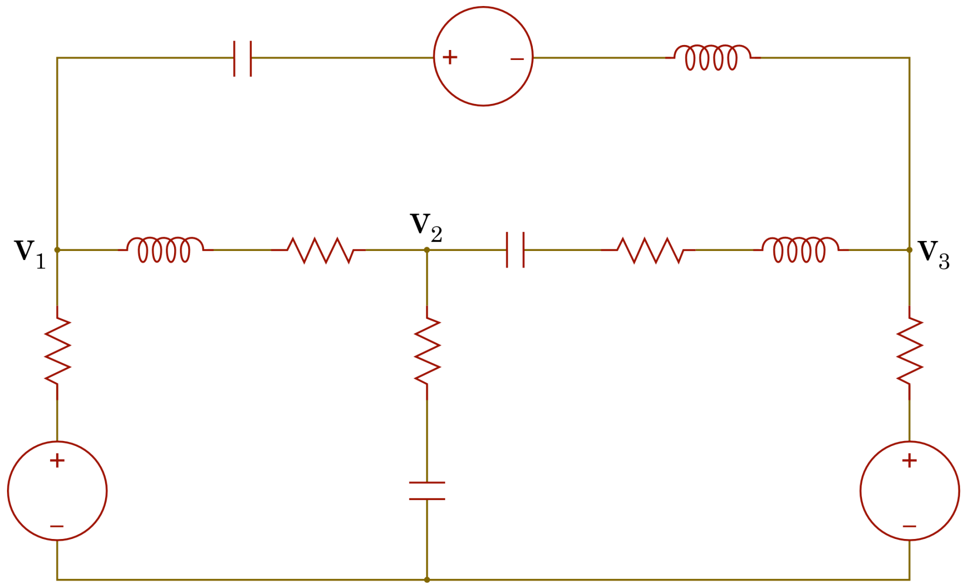
For part (b), you must solve directly for the three voltages, using nodal analysis (suitably redefined). In other words, you should not use mesh analysis to find the three mesh currents and then use these to find the node voltages. This can be tricky, so please ask if you are stuck.

Once you have an array of node voltages, use these to construct an array of (six) branch currents. Make sure the elements of your array of branch currents correspond to the elements in your array of branch impedances. (It will turn out that you will need to use KCL to determine one of the branch currents. Therefore, compute the other five branch currents first, using node voltages, then compute the last.)

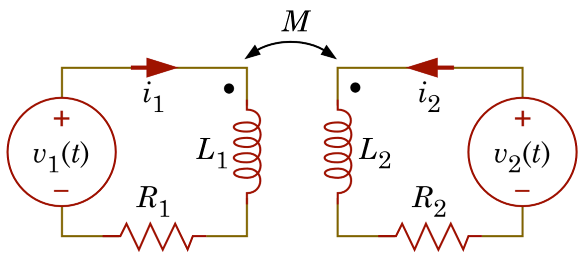
Next, construct new arrays of currents using the six branch currents, i.e., an array of three source currents and an array of 11 impedance currents. Then use dot operations to compute the complex power absorbed or delivered by each element, whichever is appropriate. The conjugate is `conj(z)`.

For the KVL and KCL checks, as well as the power balance, construct expressions that are zero. In other words, rewrite each equation so that the expected answer is zero. The `sum` function must be used for the power balance.

Answer to exercise M1, part (a)



M2. Consider the following two circuits, coupled by a mutual inductance M = 200mH.



Use: $v_1(t) = 10\sqrt{2}\cos(10t)\,\text{V}$, R_1 = 2Ω, L_1 = 100mH; $v_2(t) = -10\sqrt{2}\sin(10t)\,\text{V}$, R_2 = 1Ω, L_2 = 500mH.

(a) Determine the current phasors $\mathbf{I}_1$ and $\mathbf{I}_2$. (Use "effective" units.)

(b) Compute the voltage across each resistor and each inductor. Show that KVL is satisfied for each loop separately.

(c) Compute the complex power absorbed by each resistor and each inductor. Compute the complex power delivered by each source. Show that complex power is balanced, separately, for each loop of the network.

(d) On average, one of the circuits supplies positive power to the other. Which one supplies power to the other? Explain. How much power is supplied (on average)?

Even though this circuit yields a system of only two equations with two unknowns, you should use MATLAB to solve it, primarily because it handles arrays and complex numbers so well. In particular, you should not do the algebra yourself; instead, set up a matrix equation and solve it.

Then, continue to use MATLAB to answer the numeric parts of the other questions. Submit printouts of your M file and your output window, making sure, as always, that you have headers on both and line numbers on the M file.

For all parts, suppress the output for any line of code that does not answer a question.

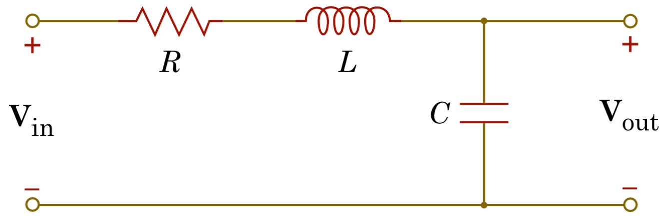
You should define variables for all given values and use references to these variables throughout your script. This makes it easier to update the script if there is a mistake.

Arrays and dot operations work well. In other words, everything comes in pairs, so you will find that it's much easier to work with an array of two sources, an array of two resistances, and an array of two inductances, then use these to compute two-element arrays for impedance, current, voltage, and complex power.

For part (b), note that a proper check always returns a value of zero, so if needed, rearrange your equation so that the answer is zero. Both KVL checks can be done with one, simple expression; the result is an array of two zeros. Similariy, for part (c), note that a proper power balance also returns a value of zero, so again, rearrange your equation as needed. Like the KVL checks, both power balances can be done with one, simple expression.

For part (d), to understand the flow of power in a transformer circuit, you might need to consider a simpler pair of coupled circuits, e.g., the one used in lecture to demonstrate power balance, which has only one source. Make sure to include units for the average power. It also helps to focus first on the inductor that absorbs positive power (on average). Where does this energy come from? Why do you think so? Where is this energy going? For instance, can this inductor store the energy? Why or why not? That is, what is the average current / average energy stored? Why?

M3. Consider the following circuit:



(a) Determine the **s**-domain transfer function for $\mathbf{V}_{\text{out}}/\mathbf{V}_{\text{in}}$ in terms of R, L, and C.

(b) Let R = 10.1Ω, L = 100mH, and C = 100mF.

1. Without using any computational aids, draw the approximate amplitude Bode diagram. Explain how you derived this diagram.
2. Without using any computational aids, draw the approximate phase Bode diagram. Explain how you derived this diagram.
3. Draw the exact magnitude Bode diagram and describe any differences from part 1.
4. Draw the exact phase Bode diagram and describe any differences from part 2.

(c) Repeat part (b) for R = 100mΩ, L = 100mH, and C = 100mF.

Parts (a), (b1), (b2), (c1), and (c2) are done by hand. You can use a word processing document, or you can do it on paper then scan it. In either case, you should submit the answers in PDF format.

For parts (1) and (2), note that approximate Bode diagrams are made with straight line segments. For full credit, you need to explain each step. If you use a word processing program, you should be able to use simple drawing tools to make your diagrams. If you use paper, a grid works well to help the grader understand what you intend. In either case, make sure the axes are labeled.

It is strongly recommended to do parts (1) and (2) before you have done parts (3) and (4), as it is easy to become distracted by the exact Bode diagrams. In one case, for instance, the approximate Bode diagram is a very poor estimate of the exact Bode diagram, but that does not change how the approximate diagram is constructed.

For part (b2), it helps to draw the individual terms first, then sum them to find the answer.

Parts (3) and (4) are done in MATLAB. Note that there are two useful functions:

```
tf([2 3],[4 5 6])
```

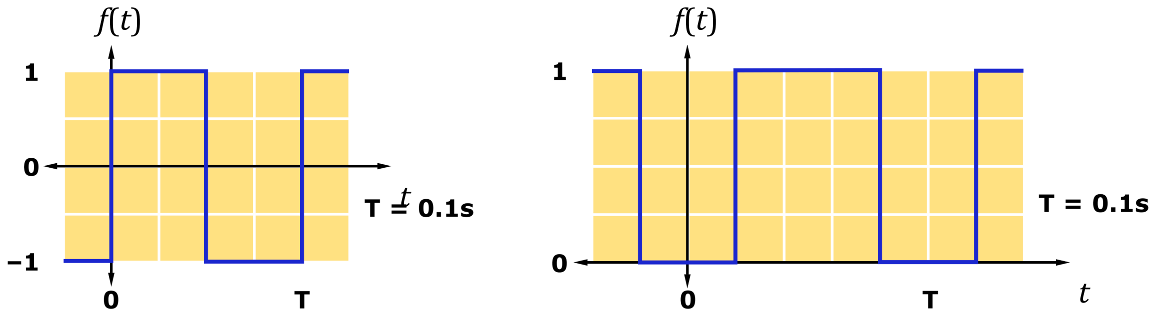
creates the transfer function $(2s+3)/(4s^2+5s+6)$, and:

```
bode(H)
```

plots the exact forms of the magnitude and phase Bode diagrams.

For part (c), you should find that the roots of the denominator are complex conjugates of each other. This means that the procedure for drawing the approximate Bode diagrams is different than when there are two real poles. In particular, there is only one break frequency. To find it, set the low frequency limit of H_{dB} equal to its high frequency limit. You also need to evaluate H_{dB} at the break frequency and place a dot on the diagram. Show your work for full credit. There is an Info Page in OWL that might help. You can also look for the lecture on "Quadratic filters". As always, please ask if you are not sure what to do.

M4. Consider the following two functions, labeled A and B:



(A) **(B)**

(a) For each function, indicate its full symmetry and which coefficients are zero as a result. Then, determine the first 10 non-zero coefficients of the Fourier series representation.

(b) For each function, use the results of part (a) to plot the terms of the truncated Fourier series, starting with the first term in the series, followed by the first plus the second, then the first, second, and third, and so on until you have the sum of the 10 terms you have calculated.

For part (a), you must do the two integrals yourself, i.e., without a computer. Use the full symmetry to predict zeros and simplify the calculations. For each chart, once you have an expression as a function of *n*, you can use MATLAB to compute the first 10 non-zero coefficients. Do <u>not</u> suppress the output for these coefficients; we need to see the actual values you are using so that you can be graded.

You <u>must</u> use a computer to do the plotting in part (b). The expressions you use to make the plots must be submitted as well, i.e., the MATLAB M file. All 10 plots should be on the same pair of axes.

Points will be deducted for inefficient code, so think about your approach. However, you do <u>not</u> need to use loops or functions to do this. Techniques learned in ECE 211 should be sufficient. For instance, it is useful to set up an array of *n* values corresponding to the non-zero coefficients. Then you can define all 10 coefficients with one very simple line of code. In other words, with one command, you can define an array of 10 coefficients. You will need to think about ways of defining the 10 functions (terms) you are going to plot for each graph and choosing an efficient one. Further, each array of coefficients should be a column array, so use semicolons to define your *n* array.

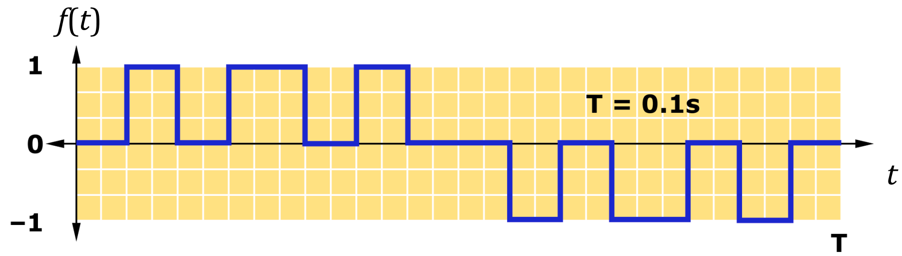
Don't forget to add proper axis labels and meaningful titles. A meaningful title tells the reader what they are looking at and what is different about this situation from other situations.

Suppress the output for the lines of code you use to set up the plots; we do <u>not</u> need to see 8,000 pieces of data. If the plots are correct, we will know it!

Note that there are two separate charts, each with 10 plots, one for function A and the other for function B. Plot only one period of each function. (If you want to add a little extra time as is done with the given functions, that will be fine, too.) Each plot should fill one page.

If you have a strong preference for using Excel, you may use it, but MATLAB is preferred. If you use Excel, you should submit one page of values and one page of formulas, as well as your plots, all as PDF files. The pages of values and formulas must have row and column headings. Note that you still need to think about efficiency and how you are going to structure your worksheet. Maximum credit is given only if the worksheet is organized well and the formulas are written to take full advantage of automatic cell indexing.

M5. Consider the following function:



(a) Indicate the full symmetry and which coefficients are zero as a result. Then, determine the first 12 non-zero coefficients in the Fourier series representation of this function.

(b) Use the results of part (a) to plot the terms of the truncated Fourier series on top of each other, starting with the first term, followed by the first plus the second, then the first, second, and third, and so on until you have the sum of the 12 terms you have calculated.

As in exercise M4, you should derive an expression for the Fourier coefficients as a function of n, then use MATLAB to compute the coefficients and do the plotting. Submit the M file, the output window, and the chart, all as PDF files.

You must determine the full symmetry of this function, which will help you to predict the zeros, as well as minimize the number of integrals you do and reduce the number of terms in your expression for the coefficients. See Prof. Leonard for advice.

As in M4, it is particularly efficient to define an array of (twelve) n values, then use it to define the non-zero coefficients. As before, use semicolons to define the n array, so that the array of coefficients is a column array.

All 12 graphs should be on one pair of axes, with proper labels and a meaningful title.

You may use Excel if you prefer. For maximum credit, you must take full advantage of the automatic cell indexing that Excel does when you copy a cell.