Lab 12
Phasor Nodal, Mesh, and Thevenin

Objectives – in this lab you will

- Perform Nodal and Mesh analysis in AC circuits using complex phasors
- Determine the Thevenin Equivalent of an AC circuit
- Use the ang.m and magPhs.m MATLAB function files
- Perform time domain analysis of AC circuits using CircuitLab

Key Prerequisites

- Chapter 10

Required Resources

- PC with internet access

This lab brings on the return of Nodal and Mesh analyis—along with complex phasors—to help us solve more advanced AC circuit problems. It shows how MATLAB or FreeMat can make these computations a lot easier to perform, and it concludes with an exploration of Thevenin equivalents in the AC domain.

Vocabulary

All key vocabulary used in this lab are listed below, with closely related words listed together:

MATLAB function file
Lagging and Leading Phase

Discussion and Procedure

In Lab 11, we worked with complex numbers the hard way. We had to convert phasor magnitude and phase quantities into rectangular form by typing a = 10*cosd(45) + j*10*sind(45), etc. And when we got our answers (also in rectangular form) we had to take the abs of the answer for the magnitude, and also the use the angle function for the angle, and translate the results from radian to degrees.

A lot of extra steps where we could easily make mistakes.

In this lab we are going to be working with even more complicated circuits, so we're going to use two user-defined functions for doing the complex number conversions, and MATLAB's matrix solving capabilities to solve complex valued systems of equations.

Part 1. Using ang.m and magPhs.m MATLAB / FreeMat

Recall from Lab 11 that we are using j as the square root of -1 in MATLAB. Therefore, complex values like 4+3j, 5-4j, etc, can be entered into MATLAB at the command line.

If we wish to enter into MATLAB the complex number 10<45, we would have to type:

```
10*cosd(45) + 10*j*sind(45)
```

However, using the function file ***ang.m***, all we have to type in MATLAB is 10*ang(45). This command computes the equivalent complex number in rectangular form, saving us the extra typing. The result is

```
--> a = 10*ang(45)
a =
   7.0711 +  7.0711i
```

Similarly, when we compute our final complex-valued result (usually a phasor), we need to convert the rectangular form result into polar form so we can do the inverse phasor transform. For example, if we wish to determin the magnitude and phase of a, we would have to type abs(a) and angle(a)*180/pi, such as this:

```
--> abs(a)
ans =
 10
--> angle(a)*180/pi
ans =
 45
```

However, using the function file ***magPhs.m***, all you have to type in MATLAB is magPhs(a). This command computes the magnitude and phase angle of a complex value and displays it using a '<' symbol to indicate "angle" (in degrees). The result is:

```
--> magPhs(a)
10.000 < 45.00
```

Note that this is not a numeric value, so it cannot be used for any further math operations.

These two files are available on the Lab 12 page in Canvas. You can also download them here: ang.m and magPhs.m. These must be copied into the folder that FreeMat is referring to by the current directory indicator in the top center of the user interface. Downloading and using these functions is shown in the lab 12 video.

Complete the exercises now under Part 1 of the Lab 12 Datasheet.

Part 2. Using MATLAB to solve complex valued systems of equations

As we solve circuits in the phasor domain, we will arrive at systems of equations where the coefficients are complex values. For example,

```
j*V1        +  (1-j)*V2   +      (2+j)*V3     = 30
3*V1        -  j*V2       +       5*V3        = 3 - j
(2 + j)*V1  +  1*V2       +       1*V3        = 2
```

We can solve these in MATLAB just as if they were real valued coefficients. In other words, in MATLAB, we would make a matrix A with all the coeffients of V1, V2, V3, and B a column vector with all the values right of the equals sign, then use A\B to solve for the unknowns.

```
            A = [ j,  1-j,  2+j; 3 -j 5; 2+j, 1 1]
            B = [30; 3-j; 2]
            V = A\B                                (or V = inv(A)*B)
V =
  -8.9558 -  2.6637i
  13.3186 + 10.2212i
   3.9292 +  4.0619i
```

Since we are usually after polar form to make it easier to interpret the answers, we usually type magPhs(u) to convert all the answers to polar form:

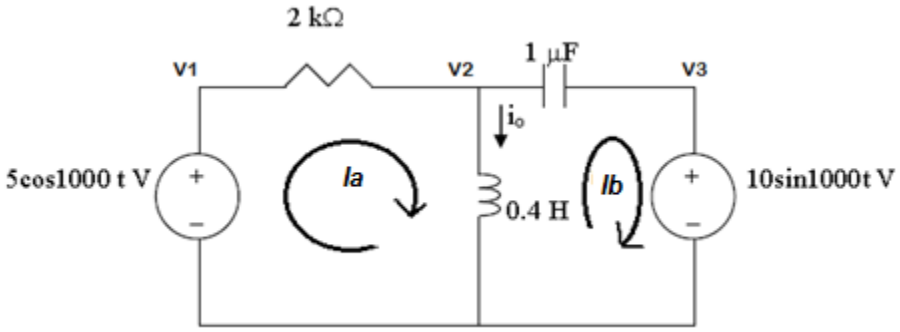
```
      --> magPhs(V)
      9.343 < -163.44              V(1)
      16.789 < 37.50               V(2)
      5.651 < 45.95                V(3)
```

Complete the exercises now under Part 2 of the Lab 12 Datasheet.

Part 3. Nodal and Mesh Analysis

We can now use these skills to solve a real AC circuit problem. Go to the datasheet and solve Problems 1 and 2 there. A partial solution to problem 1 is below.

Problem 1 Solve for io using a) Nodal and b) Mesh Analysis



Circuit Parameters:

w=1000
C = 1e-6
L = 0.4
ZC = -j/(w*C)
ZL = j*w*L

Example Nodal Equations:

V1: V1 = 5<0
V2: (V2-V1)/2000 + V2/ZL + (V2-V3)/ZC = 0
V3: V3 = 10<-90 (sin to cos phase shift = -90)

Group coefficients of V1,V2, V3 to define A and B matrices

V1	V2	V3	=	
1	0	0	=	5<0
-1/2000	(1/2000 + 1/ZL + 1/ZC)	-1/ZC	=	0
0	0	1	=	10<-90

Enter into Freemat and solve. Here is an example solution for this problem.

Then compute Io = V2/ZL and convert to polar, then inverse phasor transform back to time domain.

The answer for Io turns out to be fairly small, `0.020 <-18.43 A`, which doesn't give us a lot of precision on the magnitude. To get more accuracy for your answer you can use the command `magPhs(1000*Io)`, which will convert the units to mA, give you `19.764 < -18.43 mA`, a much more accurate representation.

Finally, to convert Io from a complex phasor back to the time domain, we use the inverse phasor transform, taking the magnitude and phase angle and inserting them into a cosine function (the real part of the complex phasor).

$i_o(t) = 19.8 \cos (1000t - 18.43^o)$

Just type this out in your datasheet. The degree symbol is a lowercase 'o' formatted as a superscript (exponent) in Word. FreeMat won't produce the time-domain function for you.

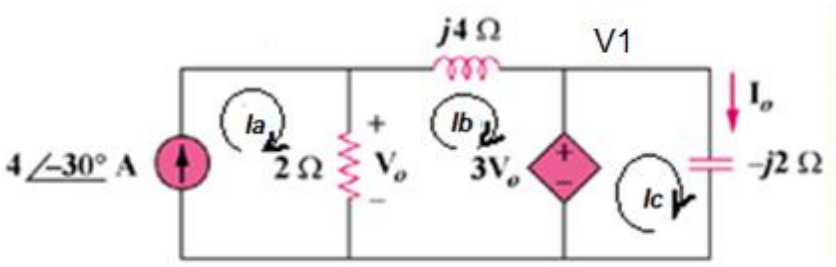
Once you have the Nodal solution working, you can save the file, then save as and give it a new name so you can modify it to work out the mesh version. The mesh version will use the same circuit parameters, but the equation and matrix development will be for a mesh analysis.

Example Mesh Equations:
Ia: -5<0 + 2000Ia + ZL(Ia – Ib) = 0
Ib:

Enter into Freemat and solve.

Then compute Io = Ia – Ib and convert to polar, should be the same as above

Problem 2 Solve for Vo and Io using a) Nodal and b) Mesh Analysis.



Nodal Analysis – If you examine the V1 node voltage you should be able to find a way to express that voltage in terms of Vo. Then you can set up one node equation with one unknown (Vo) to solve, and you won't need a matrix, just algebra to solve for Vo and from that you can derive Io.

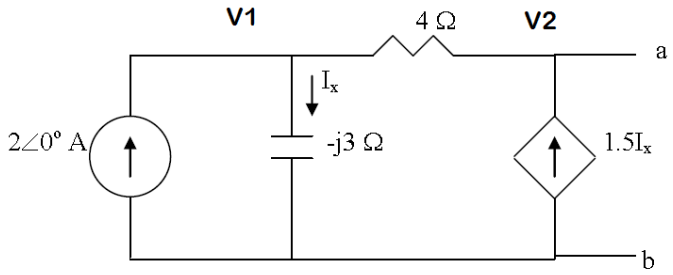
Mesh Equations

Part 4. Thevenin Equivalent – OPTIONAL for 10% Extra Credit

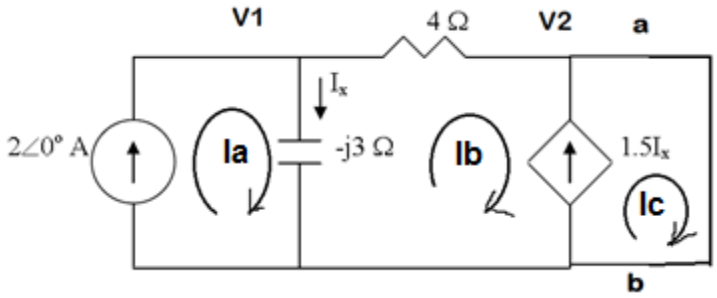
For practice, we close the lab by tackling one Thevenin Equivalent circuit problem.

Problem 3. Find the Thevenin equivalent for the following circuit.

a) use Nodal Analysis to find Vth



b) Use Mesh Analysis to find In



c) Use Vth/In to find Zth

d) Turn off independent source, connect a test voltage (or current source) equal to (typically) 1V. Use Mesh Analysis to find the current delivered by the test source (io). Find Zth by dividing 1V/io. Check against part c.