**Lab 1: Introduction to MATLAB**

**Objective:**

This lab will introduce students to some of the capabilities of MATLAB. Students are advised to try different things and analyze the outcomes to expand their learning beyond the basics required for this lab.

**Preliminary:**

As a computational tool, MATLAB operates in the abstract realm of mathematics so that it may be applied to many different types of problems, thus the name MATrix LABoratory or MAT-LAB. MATLAB has three user interfaces that can be used:

* The **Command Line** is similar to a calculator in which commands may be entered. All commands will display the answer unless the “echo” feature is disabled by concluding the command with a semi-colon “;”
* The **Editor** can be opened by typing edit at the command line. A sequence of commands/program/function may be entered, saved, and ran from inside in the Editor. In addition, MATLAB uses the extension of .m. If you wish to access the scripts, programs, and functions from the command line it must be saved as a .m files.
* **Simulink** is a graphical interface in which information (signals) are passed through commands (blocks) to yield the desired results.

In MATLAB if you are unsure what a command “plot” does, type

>> help plot

MATLAB uses the “%” to identify user comments.

1. Using a computer go to www.mathworks.com/products/matlab and watch the

2. Select “Videos” and watch the following tutorials.

1. Product Overview (“MATLAB Overview”)
2. Getting Started with MATLAB
3. Writing a MATLAB Program
4. Using Basic Plotting Functions

3. Select one additional tutorial to watch.

**Procedure:**

1. Write a script file that will plot from 0 to 2π with calculations done every 0.01 seconds using two different techniques:

a. Use a **for** or **while** loop: In a loop, a variable t is increased from 0 to 2π with the data being stored and plotted after v(t) has been calculated.

For Example (NOTE *the following is a sudo-code and is not complete*):

>> for x=0:0.1:2\*pi

>> y[k] = 5-4\*exp(-0.5\*x)\*cos(2\*x); % vector index must be whole number

>> end

b. Using Vector Math: In MATLAB vectors can be multiplied in two different ways. This first way is the traditional matrix multiplication or product.

For example, if

 and , then .

The second way is called the element-by-element multiplication:



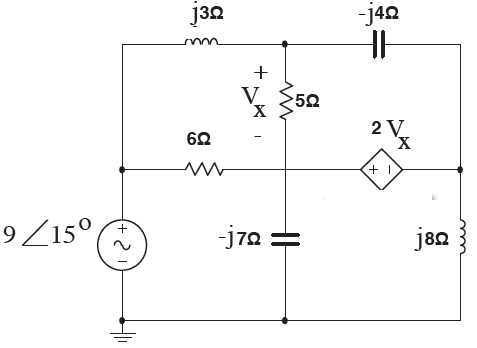
For example,

>> x=[0:0.1:2\*pi];

>> y = 5-4\*exp(-0.5\*x).\*cos(2\*x); % Note: “**.\*”** is used between exp and cos

Include copies of the both scripts and plots in your lab report.

2. Use the capability of MATLAB to perform calculations with complex numbers and matrix to solve for the node voltages of the following circuit



by:

1. Develop the underlining equations.

2. Represent the simultaneous equations in matrix form: Ax = B

>> A = [ ….];

>> B = [ … ];

3. Solve for x by multiplying the inverse of A by B, i.e. x = A-1 B

>> x = inv(A)\*B; % this time we use the matrix product.

Include copies of all scripts in your lab report.

**Conclusion:**

Summarize the key concepts presented in the tutorials for use later on in this class.

**APPENDIX**

**Help for Lab #1**:

Part 1: Plot from 0 to 2 with sampling interval of 0.01

% Part 1, Approach 1 using for-loop

k=0;

for x=0:0.1:2\*pi

k=k+1; % array index must be an integer

y(k)=5-4\*exp(-0.5\*x)\*cos(2\*x);

end

figure(1), plot(y)

% Part 1, Approach 2 using vector calculation

x=[0:0.1:2\*pi];

y=5-4\*exp(-0.5\*x) .\* cos(2\*x);

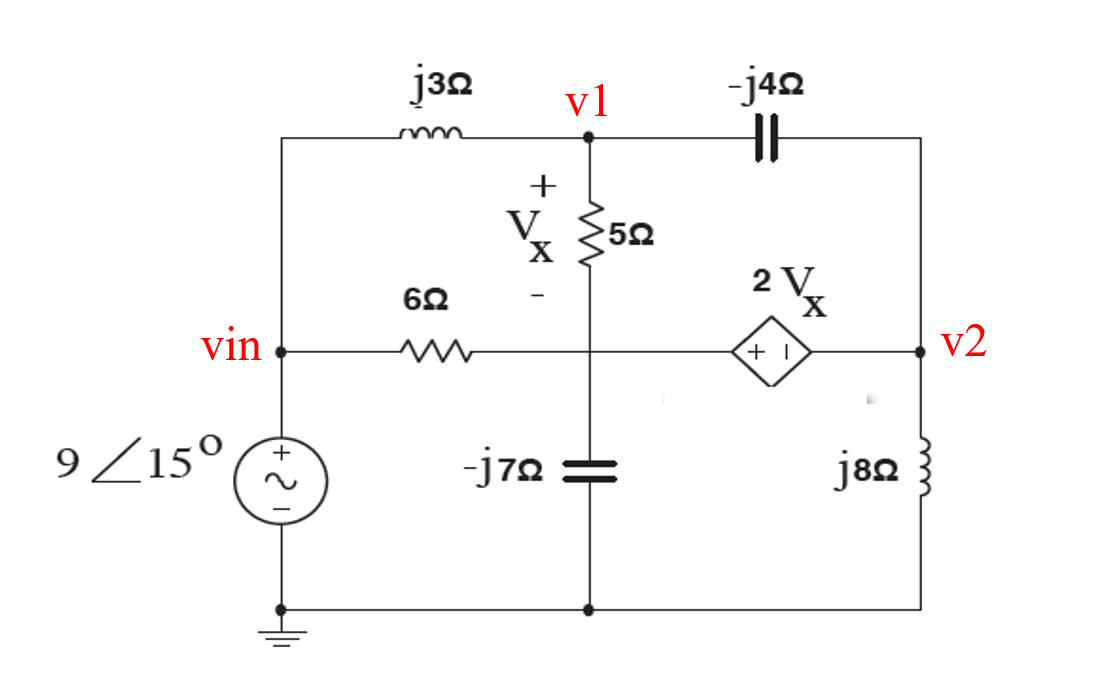
figure(2),plot(x,y)

xlabel('Time(s)')

ylabel('Voltage(V)')

title ('v(t)')

Part 2: Solve for the node voltages:



Setup equations:

v1

vin

 →

 → 

 → 

% Part 2, Setup the linear equations

[vinR, vinI] = pol2cart(15\*pi/180, 9);

vin=vinR + j \* vinI; % The voltage source in complex # form

%A=[1/(j\*3)+1/(5-j\*7)-(1/j\*4), 1/(j\*4), 0;...

% 1/(j\*4), -1/(j\*4)+1/(j\*8)+1/6, 2/6;...

% -5/(5-j\*7), 0, 1]

%B=[(1/j\*3)\*vin; 0; 0]

A=[1/(j\*3)+1/(5-j\*7)-(1/j\*4), 1/(j\*4), 0;...

1/(j\*4), -1/(j\*4)+1/(j\*8)-1/(j\*7), -2/(j\*7);...

-5/(5-j\*7), 0, 1]

B=[(1/j\*3)\*vin; 0; 0]

% Part 3, solve the equations

V = inv(A)\*B % solving A V =B

Vr =real(V) % taking the real part

Vi = imag(V) % taking the imaginary part

[ang, mag] = cart2pol(Vr,Vi) % obtain the phasor notation

angdeg = ang\*180/pi % converting angle unit to degree

% To find the angle in a different way

angdeg1 = atan2(Vi,Vr)\*180/pi %inverse tan (y,x)

**Appendix**

**NI**

----------------------------------------------------------------------------------

1SPICE: **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis