Signals Systems and Transforms
EEET-332
Lab 4

For each section using MATLAB:

Create a new *.mlx file. Save it. Remember, no spaces in the file name!

A quiz will be given at the beginning (1^{st} 10 minutes) of the lab covering the content of the prelab. One quiz will be dropped. NO make-up quizzes will be given.

Prelab:

1) Determine Z_{in} for the circuit below using R2 = 4 ohms and C2 = 1F. The input impedance Z_{in} is the total impedance seen by a signal source when connected between the input terminals, in this case, vin2 and ground (GND).

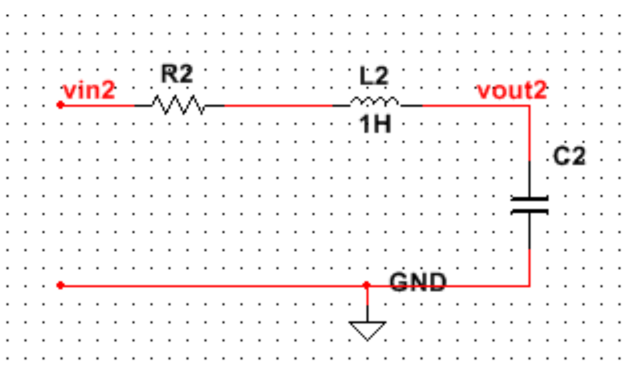
   Recall:

   $Z_R = R$

   $Z_C = \frac{1}{sC}$

   $Z_L = sL$

2) Determine the transfer function (vout2/vin2) for the circuit below using R2 = 4 ohms and C2 = 1 F.



3) Read this section on array multiplication.

Most of the multiplication ***we will perform in this lab*** requires a dot in front of the star (.*). This is also true for division and exponents. Leaving the dot out (that is, just using *) sometimes causes the "Error using *, Inner matrix dimensions must agree."

MATLAB help (below) describes the .* as an array multiply or element-by-element multiply.

Signals Systems and Transforms
EEET-332
Lab 4

```
>> help .*
 .*  Array multiply. X.*Y denotes element-by-element multiplication. X and Y must have compatible sizes. In the simplest cases, they can be the same size or one can be a scalar. Two inputs have compatible sizes if, for every dimension, the dimension sizes of the inputs are either the same or one of them is 1.
```

For example, if x = [1 2 3] and y = [4 5 6], then

```
>>  w  =  x .* y
w =
   4    10     18
```

Similarly, typing x .^ 2 squares each element of x.

```
>>  w  =  x .^ 2
w =
   1     4     9
```

4) Read this section on matrix multiplication.

MATLAB help (below) calls the * a matrix multiply.

```
>> help *
 *   Matrix multiply.    X*Y is the matrix product of X and Y.  Any scalar (a 1-by-1 matrix) may multiply anything.  Otherwise, the number of columns of X must equal the number of rows of Y.
```

For example:

```
>> a=[1 1 0 0]

a =

     1     1     0     0

>> b=[1; 2; 3; 4]

b =

     1
     2
     3
     4
>> c=a*b
```

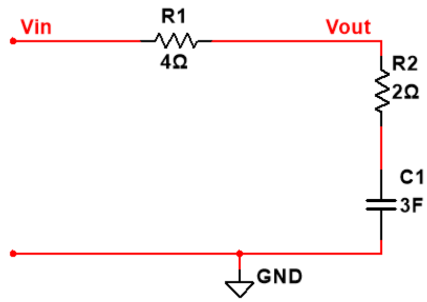
```
c =
     3
```

5) Determine the results below, given:
   w = 10, x = [1 2 3 4], y = [3 4 5 6] and z = [3; 4; 5; 6]. Verification in MATLAB is recommended.

   a) F1 = x.*y
   b) F2 = x*y
   c) F3 = x*z
   d) F4 = w*x.*y
   e) F5 = w*x*z

Signals Systems and Transforms
EEET-332
Lab 4

Section 1: Skill Building (Algebra and Symbolic Toolbox)



$$\frac{V_{out}}{V_{in}} = \frac{6s+1}{18s+1}$$

1) Create the section1.mlx file and enter the MATLAB text and code below to find the transfer function $H = \frac{V_{out}}{V_{in}}$.

`lab4.mlx`

text

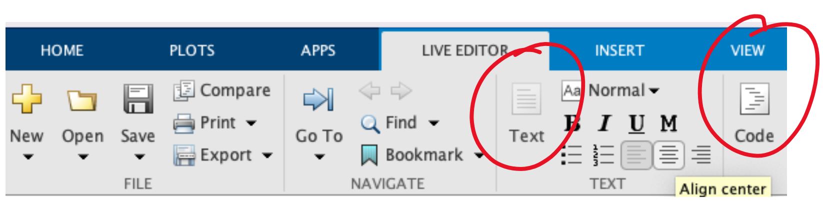
code

Lab 4

Section 1 H = Vout/Vin

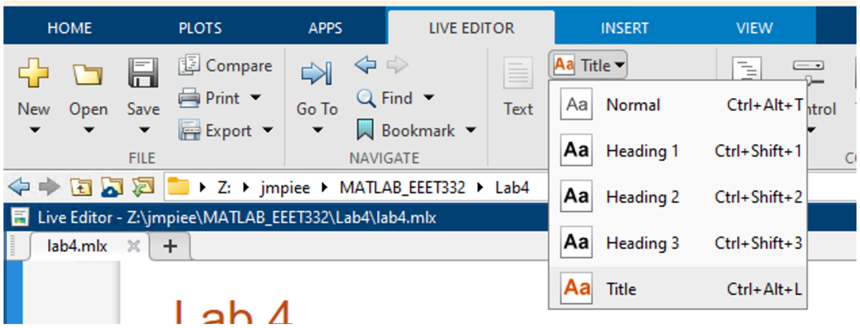
```
syms s R1 R2 C
Zc = 1/(s*C)
Zs = simplifyFraction( R2 + Zc)
H = simplifyFraction(Zs / (R1+Zs))
R1 = 4; R2 = 2; C = 3;
H = subs(H)
```

You can enter text or code by selecting the appropriate button from the Live Editor ribbon below.

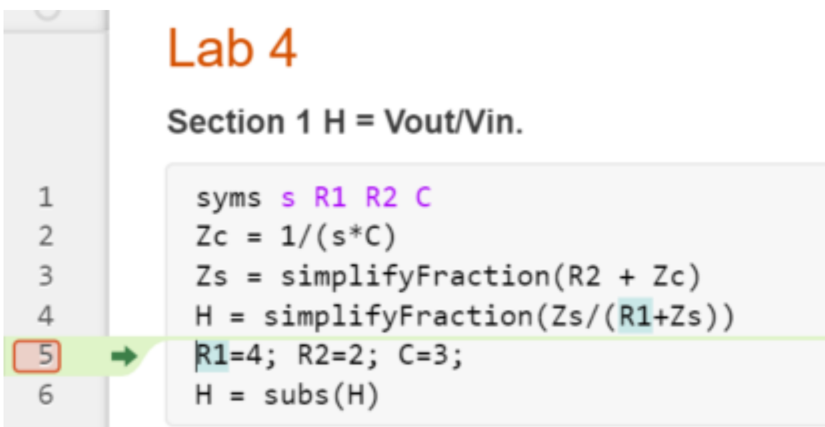


a) Section titles and headings like “Lab 4” and “Section 1” make the code more readable. As shown in the figure below, to create a title like “Lab 4” select the appropriate option.

Signals Systems and Transforms
EEET-332
Lab 4



b) Setting breakpoints allows you to stop the program at any line. Clicking the line number 5 below will cause it to highlight and create a breakpoint on that line.



2) Add the breakpoint as shown above and run the script. MATLAB will stop at the breakpoint and put a small green arrow by R.

Determine (by hand) the transfer function $H = \frac{V_{out}}{V_{in}}$ for the circuit. Refer to the MATLAB results as you work out the problem. Submit your handwritten equations and screenshots showing the MATLAB scripts and results in your report.

Why do it by hand when MATLAB can do it for you? ***Because you MUST double check your work.*** *Entering incorrect equations in MATLAB is a common mistake.*

Signals Systems and Transforms
EEET-332
Lab 4

Section 2: Continuous-time transfer functions

In MATLAB, a polynomial is represented by a row vector of its coefficients, for example, the polynomial $s^3 + 4s + 10$ is specified as [1 0 4 10].

1) Our transfer function from Section 1 can be converted to row vectors using the numden() and sympoly() functions. Add the text and code below to your script.

Section 2 Impulse and Step

```
% Convert symbolic equation to numerator and denominator row vectors
[symNum,symDen] = numden(H)
num = sym2poly(symNum)
den = sym2poly(symDen)
```

2) The tf() command is used to create Creating Continuous-Time Models that are used by bode(), impulse(), step() and many other Control System Toolbox functions. Add the code below to your script.

```
% Create a continuous-time model of the transfer function
HTF =tf(num,den)
```

3) There are multiple ways to verify stability.
   a) Impulse response. Does it go to zero over time, oscillate or grow? A stable system goes to zero, a marginally stable system oscillates and an unstable system increases in amplitude over time. Add the impulse function to your script.

```
% Check for stability using impulse
impulse(HTF)
```

   b) Poles (roots of the denominator). Use the roots(), pole() and pzmap() commands to find the roots of the denominator.
      - If all poles are in the LHP (left hand plane) the system is stable.
      - If one set of complex poles is on the y-axis and all others poles are on the LHP the system is marginally stable (oscillates).
      - If any root is in the RHP (right hand plane) the system unstable.

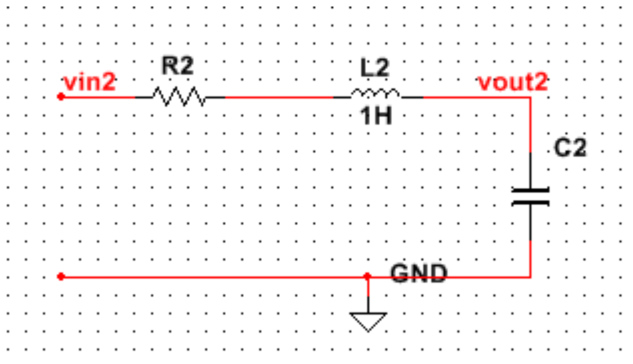
```
% Check for stability using poles
roots(den)
pole(HTF)
pzmap(HTF)
```

Submit screenshots showing the MATLAB scripts and plots in your report.

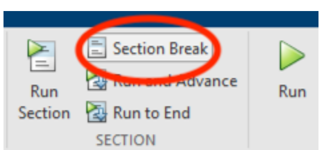
Signals Systems and Transforms
EEET-332
Lab 4

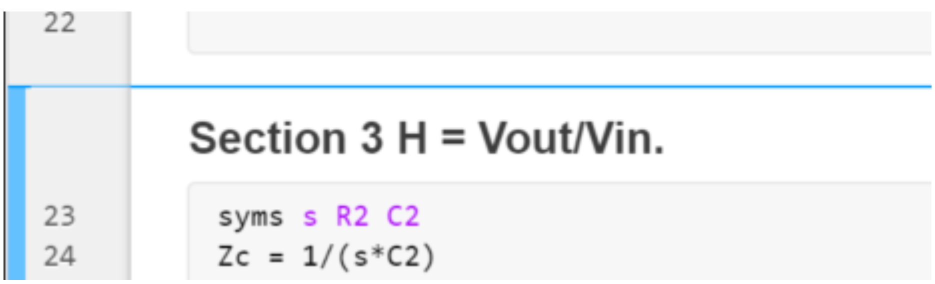
Section 3:

1) Determine (by hand) the transfer function $H = \frac{V_{out2}}{V_{in2}}$ for the circuit below. Your equation will include R2 and C2 because their values are not given. Show all steps.

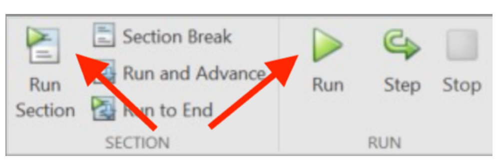


2) Now, you will use MATLAB to calculate the *H* transfer function for the circuit above.
   a) MATLAB allows you to have many sections in your script and to run just one at a time. This is an excellent place for a section break because we no longer want to run the previous sections. Click on the Section Break button to create a section break.





   b) Then you can choose either Run Section or Run in the Live Editor ribbon.



Signals Systems and Transforms
EEET-332
Lab 4

Show your handwritten calculations from Section 3-1 and MATLAB results from Section 3-2 for a sign-off.

Section 4:

Section 4-1

Create a section break with the heading ‘Section 4’.

1) Using the circuit from Section 3 pick ANY values for R2 and C2 and use the subs() command to substitute them into H.

```
R2=3; C2=4;
H = subs(H)
```

2) If you use the same values, MATLAB should calculate.

```
H =
```

$$\frac{1}{4\,s^2 + 12\,s + 1}$$

3) Using the commands from the previous section, convert the symbolic transfer function into a Continuous-Time Model.
4) Demonstrate stability by plotting the impulse function, printing the poles, and a pzmap. Get a sign-off.

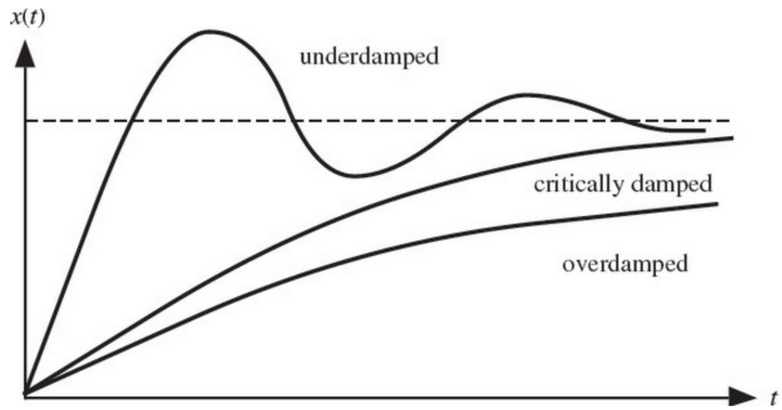
Section 4-2. Step response

1) Create a step response.
   a) To create a step response, use the function step() just like you used impulse(). An impulse is a burst of energy like a clap while a step is an input of 1 that stays on.
   b) Determine if the values you picked for R2 and C2 made your system underdamped.

A second-order system (quadratic in the denominator) can have one of three different step responses: underdamped, critically damped, or overdamped. If you imagine an elevator ride, the <u>underdamped</u> elevator would shake up and down as you came to your desired floor and the <u>overdamped</u> would get there much more smoothly but would take forever. The critically damped arrival would arrive smoothly and quickly.

Signals Systems and Transforms
EEET-332
Lab 4

Sample Step Responses



The transfer function for a second order system can be written as,

$$G(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

A system is underdamped when ζ (zeta) is less than 1, critically damped when ζ (zeta) is equal to 1, and overdamped when ζ (zeta) is greater than 1.

2) Determine ζ and ω_n for the R2 and C2 values you picked. See the example below when R3=3 and C2=4 showing overdamped.

$$\frac{1}{4s^2 + 12s + 1} = \frac{1/4}{s^2 + 3s + 1/4} = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

$${\omega_n}^2 = 1/4$$

$$\omega_n = 1/2$$

$$2\zeta\omega_n = 2\zeta\, 1/2 = 3$$

$$\zeta = 3$$

3) If your random choice of R2 and C2 made a critically or overdamped response (like the example), change R2 and C2 to ANY new values that will create an underdamped response. If your random choice of R2 and C2 made an underdamped response, change R2 and C2 to ANY new values that will create either a critically or overdamped response.
4) Is the new system stable?
5) Show the plots for the underdamped and overdamped (or critically damped) step responses for a sign-off.

Signals Systems and Transforms
EEET-332
Lab 4

Report:

Create your own cover page.

Submit your cover page, the requested solutions and screenshots from sections 1 and 2, and this sign-off sheet.

Sign-offs

Name ______________________

Section 3: Transfer function by hand and MATLAB

	/ /
Signature	Date

Section 4-1: Impulse, poles and pzmap

	/ /
Signature	Date

Section 4-2: Step Response (underdamped and critical/overdamped)

	/ /
Signature	Date