

**This homework is due March 28, 2016, at Noon.**

**1. Homework process and study group**

Who else did you work with on this homework? List names and student ID's. (In case of hw party, you can also just describe the group.) How did you work on this homework?

**2. Lecture Attendance**

This question is a student trust-based system for giving credit to those who attend lecture and help the course out by doing so. Lying on this (or any other part of the homework) constitutes academic dishonesty, and more importantly than any academic sanctions, lying would damage your honor and integrity. Be honest. You will carry your honor and integrity with you for the rest of your life, and they are way more important than your GPA.

Did you attend live lecture this week? (the week you were working on this homework) What was your favorite part? Was anything unclear? Answer for each of the subparts below. If you only watched on YouTube, write that for partial credit.

- (a) Monday lecture
- (b) Wednesday lecture
- (c) Friday lecture

**3. Solving Second-Order Differential Equations**

In the lecture, you were shown how to solve first-order differential equations. We will ask you to solve the following equation, which involves second-order differentiations:

$$\frac{d^2}{dt^2}y(t) - 2\frac{d}{dt}y(t) - 3y(t) = 0$$

where the initial conditions are  $y(0) = 0$ ,  $\frac{d}{dt}y(0) = 1$ .

- (a) Firstly, please write the above equations into the matrix form  $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t)$ . How do you define  $\vec{x}(t)$ ,  $A$ , and the initial conditions  $\vec{x}(0)$ ?
- (b) What are the eigenvalues and eigenvectors of  $A$ ?
- (c) Based on the above results, what is the solution to the second-order equation?

**4. Solving Differential Equations Numerically**

In this problem, we will be looking at how you can use a computer to solve differential equations. Sometimes differential equations cannot be solved analytically, and when this happens, we can use numerical algorithms to find a close approximation to the answer. In this problem, we will consider the motion over time of a particle attached to a spring. We set the mass of the particle to 1 so that we can compute the acceleration

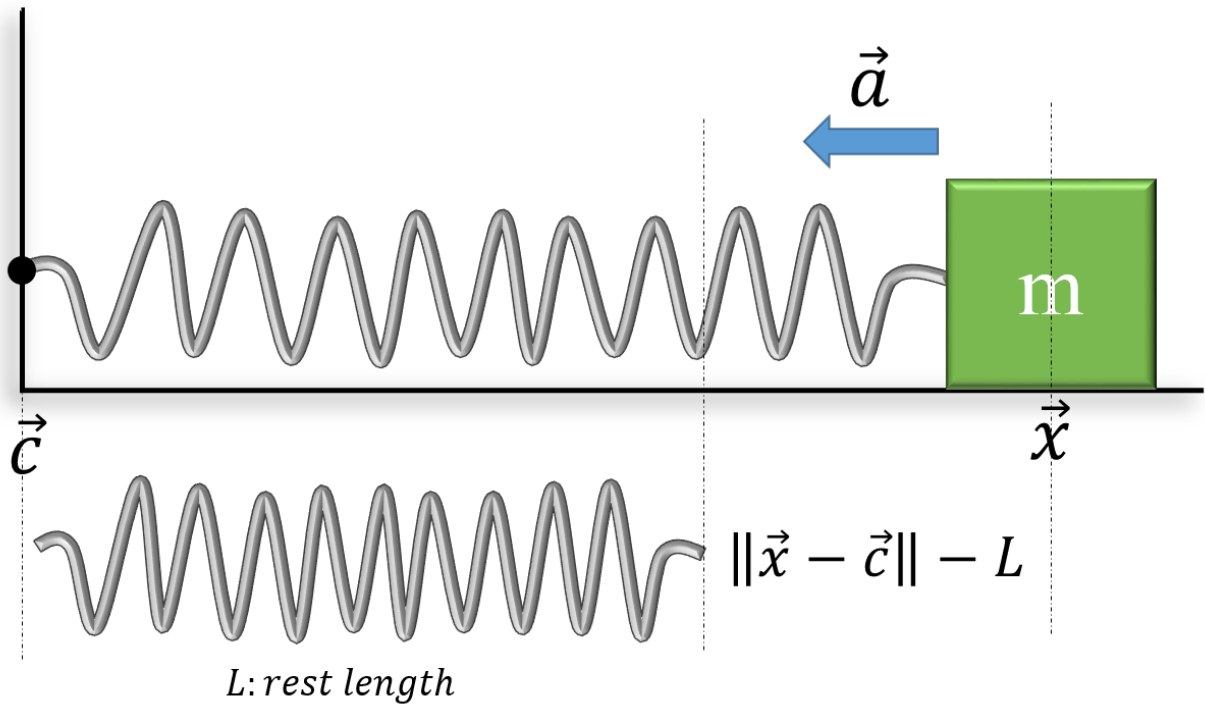


Figure 1: A spring model with one end of the spring attached to a point  $\vec{c}$  and the other end attached to some particle with mass  $m = 1$  at point  $\vec{x}$ . The rest length of the spring is  $L$  and is denoted by spring on the bottom. The spring on the top has been stretched past the rest length, and therefore, the acceleration  $\vec{a}$  is pointing towards the left.

from the force  $\vec{F}$  as  $\vec{F} = \vec{a}$ . See Figure 1 for a visualization of the spring model. We model the motion of a particle by its position  $\vec{x}(t)$ , its velocity  $\vec{v}(t)$ , and its acceleration  $\vec{a}(t)$ , and these three functions are related by  $\frac{d\vec{x}(t)}{dt} = \vec{v}(t)$  and  $\frac{d\vec{v}(t)}{dt} = \vec{a}(t)$ . If we are modeling a particle with mass 1, the acceleration is given by Hooke's Law:

$$\vec{a}(t) = - (k_s (\|\vec{x}(t) - \vec{c}\| - L)) \frac{\vec{x}(t) - \vec{c}}{\|\vec{x}(t) - \vec{c}\|} \quad (1)$$

where  $k_s$  is the spring constant,  $\vec{c}$  is the location of the other end of the spring, and  $L$  is the rest length of the spring. Note that all of these functions  $\vec{x}(t)$ ,  $\vec{v}(t)$ , and  $\vec{a}(t)$  are continuous functions that we can evaluate at any time  $t$ .

- (a) In this problem, we will use a computer to approximate the solution to the differential equation. Due to limitations of the computer, we will take our continuous function  $\vec{x}(t)$  and solve for it at discrete points in time  $\vec{x}(t_0), \vec{x}(t_1), \dots, \vec{x}(t_n)$ . Given the acceleration of a particle  $\vec{a}(t)$ , we will use the Euler method to approximate the  $\vec{x}(t)$  at these discrete points in time. However, we will still use our knowledge of the continuous functions  $\vec{v}(t)$  and  $\vec{a}(t)$  to help us make the approximation. Euler's method works with first-order differential equations, but because we have an equation for acceleration, our problem is second-order. To work around this problem, we define a new function

$$\vec{y}(t) = \begin{pmatrix} \vec{x}(t) \\ \vec{v}(t) \end{pmatrix} \quad (2)$$

Please write the derivative  $\frac{d\vec{y}(t)}{dt}$  in terms of  $\vec{x}(t)$  and  $\vec{v}(t)$ .

- (b) As you can see from the previous part, we now can write the derivative of  $\vec{y}(t)$  in terms of only parts of  $\vec{y}(t)$ . Thus we can express the derivative as  $\frac{d\vec{y}(t)}{dt} = \vec{f}(t, \vec{y}(t))$  where  $\vec{f}$  is the function you derived to compute the derivative in part (a). In this form, we can now use Euler's method to approximate the function at time  $t + h$  by

$$\vec{y}(t + h) = \vec{y}(t) + h\vec{f}(t, \vec{y}(t)) \quad (3)$$

where  $h$  is the length between each discrete point in time we are estimating  $\vec{y}(t)$ .

Let's test this method on a 1-dimensional spring example where one end of the spring is fixed and a particle is on the other end. In the supplied IPython Notebook, write the code to compute the acceleration of the particle in the function *spring\_d* by writing the code to calculate  $\frac{d\vec{y}(t)}{dt}$ .

- (c) When you run Euler's method and view the particle moving over time, you will notice that something appears wrong after some time. Describe what you see.
- (d) To fix this problem, a common method is to add a "drag" force to the particle, which applies a force opposing the velocity of the particle. Adding this term to the acceleration, we now have

$$\vec{a}(t) = - (k_s (\|\vec{x}(t) - \vec{c}\| - L)) \frac{\vec{x}(t) - \vec{c}}{\|\vec{x}(t) - \vec{c}\|} - k_d \vec{v}(t) \quad (4)$$

where  $k_d$  is the drag constant that controls how strong the drag force is. In the IPython Notebook, write the computation for this new acceleration including the drag force in the function *spring\_drag\_d*.

- (e) Although in the previous parts we have assumed that one end of the spring is fixed, the other end can also be attached to another particle. Let's suppose we have two particles *A* and *B* whose positions are given by  $\vec{x}_A(t)$  and  $\vec{x}_B(t)$ . Now, the acceleration of the two particles depends on the positions of both of them. From physics, we know that the force due to the spring on particle *B* must be equal in magnitude and opposite in direction from the force on *A*. Using this information, write an expression for the accelerations  $\vec{a}_A(t)$  and  $\vec{a}_B(t)$  by using Equation 4 as a guide. The accelerations should include just the acceleration from the spring and not the drag.
- (f) Now that we have an equation to compute the acceleration from a spring between two pairs of particles, we can compute accelerations for a system of particles by computing the pair-wise spring accelerations. In the IPython Notebook write your equation in the function *spring\_pair\_d*. Run the animation at the end of the notebook to verify that you are computing the accelerations correctly.

## 5. Simulated Inductor - Gyrator

As you have seen in other parts of the homework, you can make really interesting and useful circuits using a combination of resistors, capacitors, and inductors. Yet, the inductance needed for certain uses require very large inductors who may not be feasible to put on a small chip.

One way to work around this constraint is to simulate an inductor using a circuit model called Gyrator. Based on the model, we can simulate an inductor using only a capacitor, two resistors, and one op-amp.

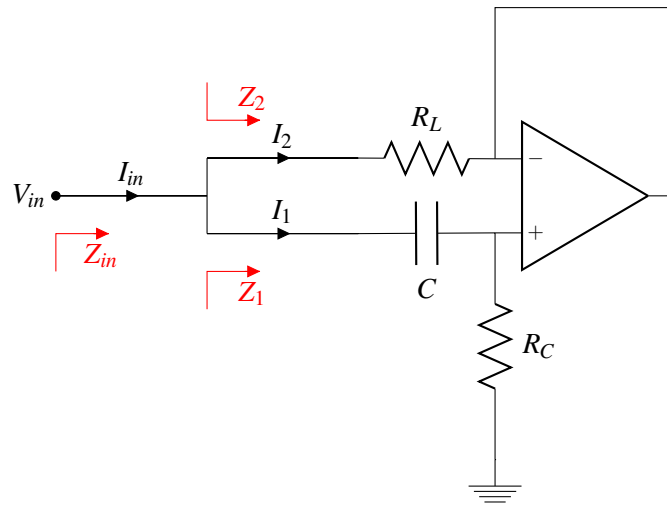


Figure 2: Gyrator

The RL Equivalent circuit is as follows:

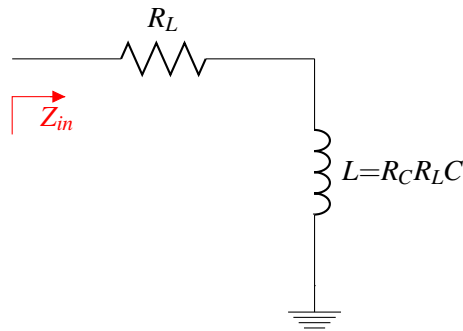


Figure 3: Gyrator - Equivalent RL circuit

The 2nd figure shows that the input impedance for both circuits is defined as

$$Z_{in} = R_L + j\omega R_C R_L C \quad (5)$$

Let us work through the steps to see that these two circuits are equivalent by showing that their impedances are very similar. In addition, we will show some cases where the Gyrator fails.

Assume that  $R_C \gg R_L$ ,  $C$  is very small, and  $V_{in}$  can only oscillate at small frequencies, unless otherwise stated.

Recall that the impedance  $Z_{in}$  into a circuit is defined as  $\frac{V_{in}}{I_{in}} = Z_{in}$ .

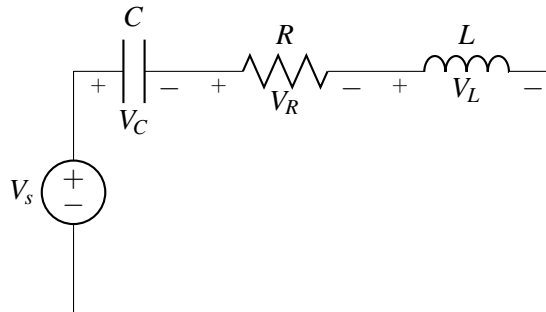
Show all your work to receive credits. All answers should be in Phasor form.

- We can determine the total input impedance by solving each branch separately. First, determine the impedance  $Z_1$ , which is the impedance of the lower branch.
- Determine the impedance  $Z_2$ . Hint: Use the properties of an op-amp you have learned.
- Now that you have found the two parallel impedance, determine  $Z_{in}$ . Remember our assumptions and the properties of parallel impedance. Your final answer should be equation 5. Hint: The solution will rely on getting a approximation which will work under our assumptions.

- (d) Now, let's check if the two circuits are equivalent under certain conditions. Let  $V_{in}$  be a DC voltage, i.e. its frequency is 0. Are the input impedances for the two circuits equivalent? If not, what is the input impedance for each circuit?
- (e) Let  $V_{in}$  be an oscillating cosine with an infinitely large frequency. Are the input impedances for the two circuits equivalent? If not, what is the input impedance for each circuit?

## 6. RLC Circuit 1

In this question, we will take a look at an electrical systems described by second order differential equations and analyze their transfer function. Consider the circuit below:



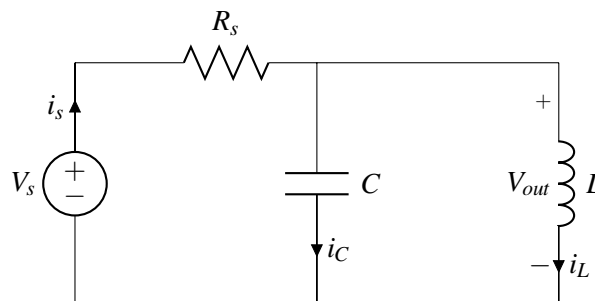
- (a) Write the KVL equation for the above circuit in time domain and convert it to the phasor domain.
- (b) Suppose the voltage source was shorted at  $t \geq 0$ , i.e.  $V_C = V_s = V$  for  $t < 0$  and  $V_s = 0$  at  $t \geq 0$ . Write down the differential equations in the form of the matrix

$$\begin{pmatrix} \frac{di}{dt} \\ \frac{dV_C}{dt} \end{pmatrix} = A \begin{pmatrix} i \\ V_C \end{pmatrix} \quad (6)$$

- (c) Find the eigenvalues of the matrix A and determine  $V_C(t)$  for  $t > 0$ . How are the transients going to behave for  $\frac{R}{2L} > \frac{1}{\sqrt{LC}}$  and  $\frac{R}{2L} < \frac{1}{\sqrt{LC}}$ .
- (d) Determine the transfer function  $\frac{\tilde{V}_L}{\tilde{V}_s}$  and find the roots of the denominator. Explain the similarity between the poles and eigenvalues of the above matrix.

## 7. RLC Circuit 2

Now consider the circuit shown below:



- Write down the KCL equation; express the total current through the source as a sum of currents in the capacitor and inductor.
- Determine the equivalent impedance of the parallel circuit in the phasor domain.
- Write down the transfer function of this circuit (take  $V_s$  as  $V_{in}$ , and the voltage across the inductor as  $V_{out}$  as indicated in the circuit.)
- Sketch the bode plot for the above transfer function. What kind of filter is this circuit?
- Suppose the voltage source is removed from the circuit (make it an open circuit) at  $t \geq 0$ , i.e.  $i_s = i_L = \frac{V_s}{R_s}$  for  $t < 0$  and  $i_s = 0$  for  $t \geq 0$ . Write down the differential equation in the form of the matrix

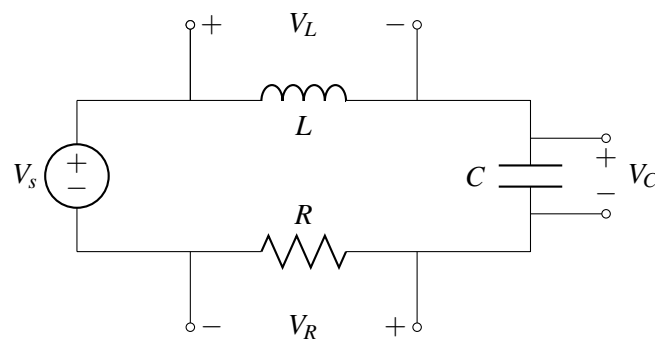
$$\begin{pmatrix} \frac{di_L}{dt} \\ \frac{dV_C}{dt} \end{pmatrix} = A \begin{pmatrix} i_L \\ V_C \end{pmatrix} \quad (7)$$

- Determine  $V_C(t)$  for  $t \geq 0$ . What is the frequency of oscillation for the above circuit?

## 8. RLC circuit as passive filters

As originally conceived by Bode in the 1930s, Bode plot is only an asymptotic approximation of the frequency response, using straight line segments. It relies on using a logarithmic scale for the input frequency  $\omega$  to express the magnitude of the transfer functions in decibels (dB).

In this question, we will go through some examples to appreciate the beauty and simplicity of Bode plots. In the iPython notebook, you will see how well the approximation of Bode plots is in different regions. In particular, we will work with the RLC circuit shown below:



In the following questions, we will be exploring how to use the above RLC circuit to construct highpass, lowpass, and bandpass filters. As the name suggests, a highpass filter will suppress the low frequency components while keeping the high frequency components of the input unblocked. Since the circuit contains only passive elements, namely resistors, capacitors, and inductors, these filters are called *passive filters*. On the other hand, if the circuit contains op amps, transistors, or other active devices, it will become *active filters*.

- Lowpass filter.** Treat  $V_s$  as the input and  $V_C$  as the output. Obtain the transfer function  $H_{LP} = \frac{V_C}{V_s}$ , and its magnitude and phase. Draw the Bode plot for the magnitude. Explain why this is a lowpass filter.
- Highpass filter.** Let  $V_L$  be the output. Obtain the transfer function  $H_{HP} = \frac{V_L}{V_s}$ , and its magnitude and phase. Draw the Bode plot for the magnitude. Explain why this is a highpass filter.
- Bandpass filter.** How can you obtain a bandpass filter based on your findings above? Write out the transfer function and its magnitude and phase.

- (d) The **resonant frequency**,  $\omega_0$ , is the input frequency (other than 0 and  $\infty$ ) that leads to the elimination of the imaginary part of the circuit impedance, i.e. the impedance is purely real. Find the resonant frequency for the RLC circuit above.
- (e) For mobile communications, the center frequency is approximately 800 MHz. In the IPython notebook, experiment with different  $L$  and  $C$  to center the bandpass filter.

You might have noticed that the advantage of Bode plot is that it makes it easier to work with transfer functions that have multiple factors, known as functional forms. We can cast  $H(\omega)$  into the **standard form**:

$$H(\omega) = A_1(\omega)A_2(\omega)\dots A_n(\omega) \quad (8)$$

where  $A_1$  to  $A_n$  assume one of the possible functional forms below:

Constant factor:  $H = K$

Zero @ origin:  $H = (j\omega)^N$

Pole @ origin:  $H = 1/(j\omega)^N$

Simple zero:  $H = (1 + j\omega/\omega_c)^N$

Simple pole:  $H = 1/(1 + j\omega/\omega_c)^N$

Quadratic zero:  $H = (1 + j2\xi\omega/\omega_c + (j\omega/\omega_c)^2)^N$

Quadratic pole:  $H = (1 + j2\xi\omega/\omega_c + (j\omega/\omega_c)^2)^{-N}$

The construction thus becomes simple addition or subtraction of the functional forms. For instance,  $H(\omega) = 10 \frac{1+j\omega/\omega_z}{1+j\omega/\omega_p}$ , where  $A_1 = 10$ ,  $A_2 = 1 + j\omega/\omega_z$ ,  $A_3 = \frac{1}{1+j\omega/\omega_p}$ .

- (f) For transfer function  $H(\omega) = M(\omega)e^{j\phi(\omega)}$ , how to represent the magnitude  $M(\omega)$  and phase  $\phi(\omega)$  with the magnitudes  $|A_i(\omega)|$  and phase  $\phi_{A_i}(\omega)$ ?
- (g) Consider the transfer function

$$H(\omega) = \frac{(j10\omega + 30)^2}{(300 - 3\omega^2 + j90\omega)} \quad (9)$$

Refer to the IPython notebook for constructing the Bode plots using functional forms.

- (h) In addition to the Bode plots, we also plotted the magnitude of the transfer functions without approximations. Please comment on the differences.

## 9. Your Own Problem

Write your own problem related to this week's material and solve it. You may still work in groups to brainstorm problems, but each student must submit a unique problem. What is the problem? How to formulate it? How to solve it? What is the solution?

### Contributors:

- Ming Jin.
- Stephen Bailey.
- Lev Taus.
- Varun Mishra.