

ECE 205 Dynamical Systems  
Spring 2010 - 11

---

## Dynamical Systems



### Circuits

First Order  
Second Order  
Characteristics  
Frequency-domain  
analysis  
Filter Types

### Signals & Systems

Properties  
Transfer Functions  
Impulse Response  
Convolution  
Interconnected  
Systems  
Bode Plots  
Steady-state response

### Controls

Properties  
Poles and Zeros  
Stability  
Gain and Phase Margins

C. A. Berry

## Course Information

**Instructor:**

Carlotta Berry (berry123)

Moench Hall, D-211

(812) 877-8657

**Description:**

3R-3L-4C

F, W, S

Pre: ECE204 and MA222

Review of matrix and differential equations. Bode plots. System classification, impulse and step response, convolution. Laplace and inverse Laplace transforms, block and signal flow diagrams. Benefits of feedback. Modeling and simulating electrical, mechanical, and thermal systems. Matlab and Simulink. Integral laboratory.

**Text:** There is no text book for this course but there will be a study guide and course notes placed in the book store for purchase that will be very helpful in mastering the course concepts.

**What is expected of You**

First and foremost, professional work is the norm in this course. All of your written work and your conduct in class are to be at the level of one who is studying a profession—the profession of engineering. This means a number of things:

1. Your work is neatly done in a professional manner, using formats specified.
2. Your work is honestly done. You are encouraged to discuss course material with classmates to help each other understand and assimilate the concepts. Nevertheless, distinguish between helping someone understand concepts and providing them with specific answers. You are expected to work individually on assignments without reference to others' work.
3. Your work is done on time. As a rule of thumb, expect to put in **eight** hours per week outside of class doing homework, reading the resources, and studying.
4. You are attentive and engaged in the lecture (i.e. not sleeping, reading the newspaper, surfing the web, doing homework for other courses, disturbing others with electronics).

**Grading:**

Grades will be assigned at the end of the quarter based on the grade weights and grading scale shown in the following table:

Midterms	36%	A	90 – 100
Final Exam	24%	B+	85 – 89
Homework	10%	B	80 - 84
Labs and Memos	15%	C+	75 - 79
Quizzes	10%	C	70 - 74
Lab Practical	5%	D+	65 - 69
		D	60 - 64
		F	Below 60

**Independent of point totals:**

- You must satisfactorily *complete* all of the lab projects in order to pass the course
- You must have a *passing* weighted exam average of greater than or equal to 60% in order to pass the course.

**Homework:**

The homework is intended to help you to understand the concepts presented in the course, and to provide you with practice in problem solving. Each student should complete the homework on engineering paper using problem solving format.

- **Problem sets are due each Thursday at the beginning of class before the bell rings.**
- No late homework will be accepted without prior approval
- Many of the homework assignments require MATLAB/Simulink and/or MultiSim. If you do not complete the computer simulation portion of the assignment, you will not get full credit for the submission.
- You are expected to complete your own work. You can discuss the assignment with others and help each other but the work that you submit should be your own. If you submit the same computer simulation files, neither of you will receive credit.
- Unless specifically told otherwise, you are expected to work out homework problems by hand. If you used Maple and just copied the answer onto the paper, you will not get credit. You cannot submit in any Maple code or plot as part of the solution to a problem. However, you can use Maple to check your answers.
- In general, I will give you answers to problems, but ***I will not hand out or post solutions.*** If you have questions or cannot get the correct answer, you need to come and see me. Your homework solutions will be your primary source for reviewing material before each exam, so it is very important that you do the homework.

**Examinations:**

In this course, examinations make up **65%** of the grade and warrant careful preparation.

Examination questions will be based on the lecture material, homework, quizzes and laboratory

work. Midterm tests will be fifty minutes in duration during the regular class meeting time. Midterms and final examinations will be closed book and closed notes. However, you may be provided with a formula sheet. It is your responsibility to memorize or derive any formula missing from this document. **You will only be allowed to use calculators on the first exam.** *Anything covered from the beginning of the class up until the time of an exam is fair game for the exam.*

**Pre-labs:**

Pre-labs are intended to help you understand the purpose and theory of the weekly laboratory assignments. Each student should complete the pre-lab on engineering paper and **submit it in class the day before the lab**. The pre-lab may also involve work using MATLAB, Simulink, MultiSim or the NI myDAQ and it should be attached as well.

**Lab memos:**

The lab memo will be used to present the purpose, theory, procedure, results and conclusions of the weekly laboratory assignments. Each student must submit a typewritten lab memo.

- Each student should purchase the lab kit and check out the NI myDAQ from the instrument room before the first lab session
- Students can work together but each person is responsible for doing and understanding the lab. Each week you will use MATLAB, Simulink, MultiSim and the NI myDAQ for the lab procedure so you should bring your laptop.
- **Lab memos should be submitted in the Angel Course drop box along with any necessary code by midnight on Sunday.** After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**.
- The instructor must sign off on your lab before you submit your memo to insure that it was completed properly

**Quizzes:**

There will be weekly quizzes (**usually on Thursdays**) that involve answering short problems or multiple choice questions on lecture material and homework. The purposes for these quizzes are:

- to give me feedback on the current level of understanding of the class
- to give you feedback on your current level of understanding
- to give you practice on problems similar to the exam format
- to also take attendance

Usually before the quiz a “practice quiz” will be posted in the Angel Course folder to give you an idea of the kinds of problems you are expected to be able to solve or questions to answer.



**Attendance:**

Regardless of whether formal attendance is taken, attendance at each class is expected. As a rule of thumb you should consider yourself seriously behind if you miss more than four classes in a four credit-hour course. According to our Academic Rules and Procedures, "A student whose total absences in a course, excused or unexcused, exceed two per credit is liable to fail the course." ***Eight absences in this course are grounds for failure.*** Missing an attendance check due to lateness may be counted as an absence.

Missed exams will not be made up. ***The final exam grade will be used to replace a missing exam grade in the case of excused absences.*** Excused absence from an examination normally requires advance approval or formal documentation of an emergency. An examination that is missed for an unexcused reason will be given a grade of **zero**. Students are not excused from scheduled exams for intramural athletics or fraternity events.

**Academic accommodation:**

Those students with documented special needs may request extra time on timed tests. Students need to contact me at least 2 business days prior to each exam to make the necessary arrangements.



Class	Day	Date	Topic	Due
1-1	M	3/5	First order circuits	
1-2	T	3/6	Solving First Order Differential Equations	
1-3	R	3/8	Solving First Order Differential Equations	Hwk1, Quiz1
1-L	T/R		<i>Lab 1 – Concept Inventory, First Order Circuits (MATLAB/ Simulink/MultiSim)</i>	
2-1	M	3/12	Second Order Circuits	
2-2	T	3/13	Solving Second Order Differential Equations	
2-3	R	3/15	Solving Second Order Differential Equations	Hwk2, Quiz2
2-L	T/R		<i>Lab 2 – First/Second order circuits (Introduction to NI myDAQ)</i>	
<b>FIRST BOILERMAKER REGIONAL, Purdue (3/15/12 - 3/17/12)</b>				
3-1	M	3/19	Solving Second Order Circuits	
3-2	T	3/20	Response of Underdamped Systems	
3-3	R	3/22	<b>Midterm 1</b>	Hwk3
3-L	T/R		<i>Lab 3 – Time domain Modeling of a 1DOF Mechanical System</i>	
4-1	M	3/26	System Properties: Linearity, Time-Invariance	
4-2	T	3/27	System Properties: Causality, Memoryless	
4-3	R	3/29	System Properties: Invertible Systems, BIBO stability, LTI Systems	Hwk4, Quiz4
4-L	T/R		<i>Lab 4 – System Linearity and a Common Emitter Amplifier</i>	
<b>NSBE NATIONAL CONVENTION, Pittsburgh, PA (3/28/12 - 4/2/12)</b>				
<b>SPRING BREAK (3/31/12 – 4/8/12)</b>				
5-1	M	4/9	LTI Systems, Impulse and Step Response of a System	
5-2	T	4/10	LTI Systems, Impulse and Step Response of a System	
5-3	R	4/12	Convolution	Hwk5, Quiz5
5-L	T/R		<b>Lab 5 – Lab Practical</b>	
6-1	M	4/16	Convolution	
6-2	T	4/17	BIBO stability and Causality, Interconnected Systems	
6-3	R	4/19	<b>Midterm 2</b>	Hwk6
6-L	T/R		<i>Lab 6 – Transfer Functions and Optical Transmitter and Receiver</i>	
7-1	M	4/23	Laplace Transform of Circuit Elements	
7-2	T	4/24	Transfer Functions, Poles and Zeros	
7-3	R	4/26	Partial Fraction Expansion, Distinct Roots, Repeated Roots	Hwk7, Quiz7
7-L	T/R		<i>Lab 7 – Simulation of Feedback control systems in Matlab and Simulink</i>	
<b>FIRST CHAMPIONSHIP, ST. Louis, MO (4/25/12-4/28/12)</b>				
8-1	M	4/30	Partial Fraction Expansion, Repeated, Complex Roots	
8-2	T	5/1	Characteristic Modes, Asymptotic Stability, Initial/Final Value Theorems	
8-3	R	5/3	Block Diagrams, Feedback Systems	Hwk8, Quiz8
8-L	T/R		<i>Lab 8 – Implementation of Feedback control systems</i>	
9-1	M	5/7	Steady State Frequency Response	
9-2	T	5/8	Bode Plots	
9-3	R	5/10	<b>Midterm 3</b>	Hwk9
9-L	T/R		<i>Lab 9 – Experimental Construction of Bode Plots</i>	
10-1	M	5/14	Bode Plots	
10-2	T	5/15	Filters	
10-3	R	5/17	Filters	Hwk10, Quiz10
10-L	T/R		<i>Lab 10 – Lowpass, Highpass and Bandpass Filters</i>	

# HOMework

## Dynamical Systems



### Circuits

First Order  
Second Order  
Characteristics  
Frequency-domain  
analysis  
Filter Types

### Signals & Systems

Properties  
Transfer Functions  
Impulse Response  
Convolution  
Interconnected  
Systems  
Bode Plots  
Steady-state response

### Controls

Properties  
Poles and Zeros  
Stability  
Gain and Phase Margins

C. A. Berry

# LECTURE NOTES

## Dynamical Systems



### Circuits

First Order  
Second Order  
Characteristics  
Frequency-domain  
analysis  
Filter Types

### Signals & Systems

Properties  
Transfer Functions  
Impulse Response  
Convolution  
Interconnected  
Systems  
Bode Plots  
Steady-state response

### Controls

Properties  
Poles and Zeros  
Stability  
Gain and Phase Margins

C. A. Berry

## Lecture 1-1

## First Order Circuits

**Reading:** Chapter 1 and 2 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define first-order circuits
- Use the standard form of a first-order circuit to determine the time constant and static gain of an electrical system
- Derive the governing differential equation for first-order circuits

A **first-order circuit** is an electrical circuit that can be described by a first-order differential equation. First-order circuits include resistors and either inductors or capacitors. They are typically referred to as RL or RC circuits.

There are two key **characteristics** of a first-order circuit:

- The *time constant*,  $\tau$ , which describes how long it takes for an energy storage element (L,C) to charge or discharge or for the system to reach steady state. A system is within 2% of its final value in 4 time constants and within 1 % of its final value in 5 time constants.
- The *static gain*,  $K$ , which represents the ratio between the output and input of the system after it reaches steady-state for a step (constant) input.

The standard form for the governing differential equation is

$$\tau \frac{dy(t)}{dt} + y(t) = Kx(t),$$

where  $x(t)$  is the input and  $y(t)$  is the output. Note that there are other standard forms but this is the one we will use to derive the governing differential equation.

A first-order circuit exhibits a **natural response** which occurs when it is connected to a DC source and then suddenly disconnected and the stored energy is released to a resistive network. The **natural response** is due to stored energy in an inductor or capacitor.

A first-order circuit exhibits a **step response** which occurs when a DC source is suddenly connected to a circuit and it begins to store energy. The **step response** or **forced response** is due to an independent source.

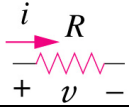
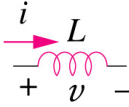
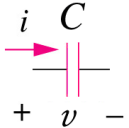
A first-order circuit with the sudden application of a DC source can also be described in terms of the **transient** and **steady-state** response. The **transient response** is the time when the inductor or capacitor is charging or discharging before 5 time constants. The **steady-state** response



exists a long time after DC switching after the inductor or capacitor is completely charged or discharged. The transient response is the temporary response that will die out with time. The steady-state response is the permanent part that will occur after a long time.

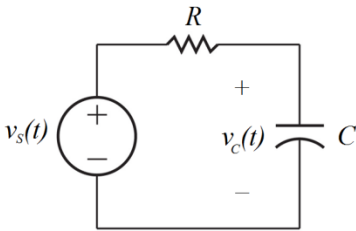
In order to derive the governing differential equation for a first-order circuit, you will use circuit analysis techniques such as KVL and KCL. You should also review the key properties of the three passive circuit elements: resistors, inductors and capacitors shown in Table 1.

**Table 1: Passive Circuit Element Properties**

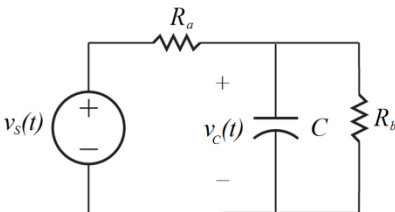
element	description	$v$	$i$	DC conditions
Resistors 	Resists the flow of current			No change
Inductors 	Produces a voltage based upon an opposition to change in electric current. Stores energy in the form of current. Current cannot change abruptly through an inductor but the voltage can be based upon whether the inductor is storing or releasing energy			
Capacitors 	Produces a current based upon a time-varying electric field. Stores energy in the form of voltage. Voltage cannot change abruptly across a capacitor but current can abruptly change based upon whether the capacitor is storing or releasing energy.			

**In-Class Activity 1:**

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain and time constant.

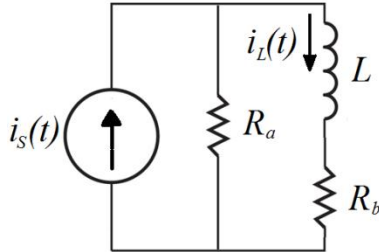
**In-Class Activity 2:**

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain and time constant.

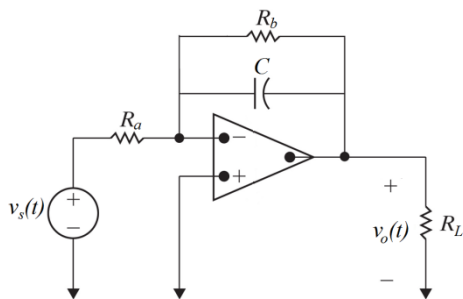


In-Class Activity 3:

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain and time constant.

In-Class Activity 4:

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain and time constant.







## Lecture 1-2

### Solving First Order Circuits

**Reading:** Chapter 1 and 2 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Determine the Thevenin Resistance, time constant and static gain of an electrical system using circuit analysis techniques
- Solve first-order differential equations using natural and forced responses

In the prior lecture, we derived the governing differential equation for RC and RL circuits. By writing this equation in standard form, it is possible to identify the static gain and time constant of the circuit.

An alternate approach to find the time constant for RC circuits is to use the formula

$$\tau = R_{TH}C$$

Note that the Thevenin resistance is the resistance seen from the terminals of the capacitor. Recall that to find the Thevenin resistance; all independent sources must be turned off. You turn off a voltage source by replacing it with a short circuit. You turn off a current source by replacing it with an open circuit.

To find the time constant for RL circuits, use the formula

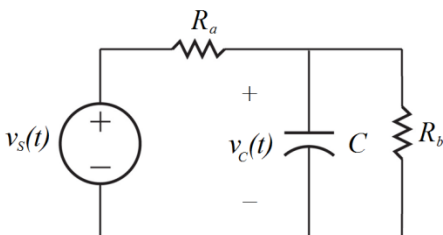
$$\tau = L/R_{TH}$$

The steady-state value when the input is a DC source,  $x(t) = Au(t)$ , is given by

$$K = \frac{\text{output}(\infty)}{\text{input}(\infty)}$$

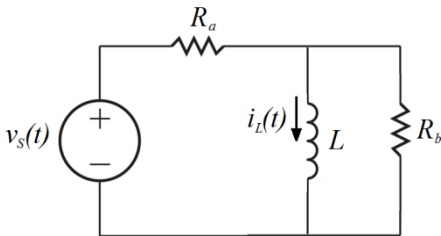
#### In-Class Activity 1:

Determine the time constant and static gain for the following electric circuit without deriving the governing differential equation.

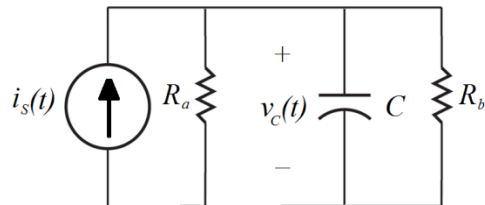


**In-Class Activity 2:**

Determine the time constant and static gain for the following electric circuit without deriving the governing differential equation.

**In-Class Activity 3:**

Determine the time constant and static gain for the following electric circuit without deriving the governing differential equation.





The solution to the first-order differential equation can be decomposed into the **natural** and **forced** response.

$$y(t) = y_n(t) \text{ and } y_f(t)$$

The natural response is due to the initial conditions such as the charged voltage on the capacitor or current through the inductor. To find the natural response, assume that the input to the system is zero and  $y_n(t) = Ce^{rt}$ .

$$\begin{aligned} \tau y_n'(t) + y_n(t) &= 0 \\ \frac{\tau y_n'(t) + y_n(t)}{\tau} &= 0 \\ &= -1 \\ r &= -1/\tau \\ \mathbf{y_n(t) = Ce^{-t/\tau}} \end{aligned}$$

The forced response due to an applied constant DC source,  $x(t) = Au(t)$  yields the following,

$$\begin{aligned} \tau y_f'(t) + y_f(t) &= Kx(t) \\ \mathbf{y_f(t) = KA} \end{aligned}$$

Thus, the complete solution is  $y(t) = \frac{\tau y_n'(t) + y_n(t) + Kx(t)}{\tau}$

To determine the constant, C assume that  $y(t_0) = Ce^{-t_0/\tau} + KA$  so  $\mathbf{C = e^{t_0/\tau}[y(t_0) - KA]}$ . Thus, the general form of the solution for a first-order circuit due to a step response is

$$\mathbf{y(t) = [y(t_0) - KA]e^{-(t-t_0)/\tau} + KA}$$

If  $t_0 = 0$  and  $t \rightarrow \infty$ , this equation becomes

$$\mathbf{y(t) = [y(0^+) - y(\infty)]e^{-t/\tau} + y(\infty)}$$

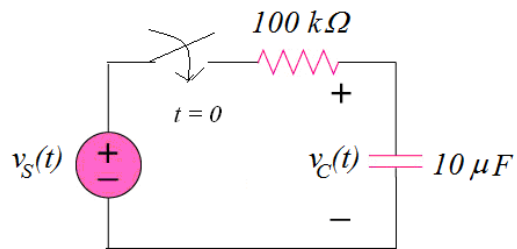
Recall that the time constant indicates how fast the system responds and if it is a step response it is demonstrated by the following table and figure assuming zero initial conditions. Note the system is within 2% of its final value after 4 time constants.

**Table 1: Step response of a first-order circuit**

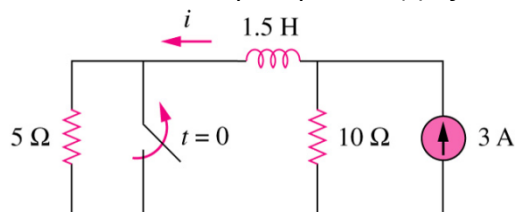
time, s	$1/\tau$	$y(t) = y(\infty)[1 - e^{-t/\tau}]$	Step response
0	0		
$\tau$	1		
$2\tau$	2		
$3\tau$	3		
$4\tau$	4		
$5\tau$	5		

**In-Class Activity 4:**

In the following circuit, assume the switch has been open for a long time and  $v_s(t) = Au(t)$ . If the switch closes at  $t = 0$ , determine the step response,  $v_C(t)$  after the switch closes.

**In-Class Activity 5:**

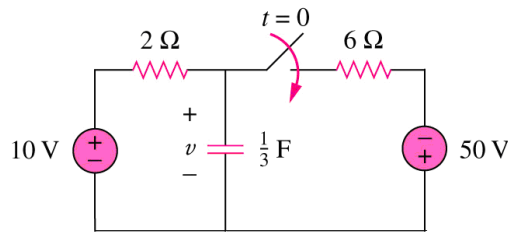
In the following circuit, assume the switch has been closed for a long time and opens at  $t = 0$ , determine the step response,  $i(t)$  after the switch moves.





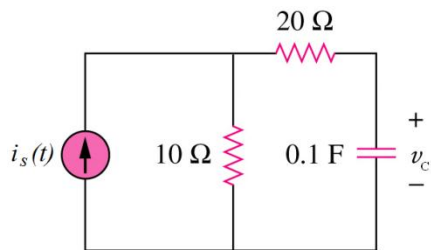
In-Class Activity 6:

For the following circuit, assume the switch has been open for a long time and closes at  $t = 0$ . Determine  $v(t)$  after the switch moves.



In-Class Activity 7:

Determine  $v_c(t)$  for the following circuit if  $i_s(t) = \begin{cases} 0 \text{ A} & t < 0 \\ 2 \text{ A} & 0 \leq t < 1\text{s} \\ -3 \text{ A} & 1 \leq t < 2\text{s} \\ 1 \text{ A} & t \geq 2\text{s} \end{cases}$



### Lecture 1-3

#### Solving First Order Circuits

**Reading:** Chapter 1 and 2 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Solve first-order circuits with arbitrary inputs using integrating factors

An alternate method for determining the solution to a first order circuit is to use **integrating factors**. Later in the quarter, we will use frequency domain and sinusoidal steady-state analysis to determine the solution to first order differential equations with arbitrary inputs.

Integrating factors are applicable to any type of input not just step inputs. Solving an ordinary differential equation with integrating factors is based upon the fact that an ordinary differential equation can be multiplied by a function to make it integrable. Let's examine a general differential equation given by

$$\frac{dy(t)}{dt} + p(t)y(t) = bx(t) \tag{1}$$

In order to find a function to solve the differential equation, multiply both sides by an arbitrary function,  $a(t)$ ,

$$a(t) \frac{dy(t)}{dt} + a(t)p(t)y(t) = \underline{\hspace{15em}} \tag{2}$$

If the product,  $a(t)p(t)$  represents the derivative of  $a(t)$  then the left side of equation (2) represents the derivative of a product and can be written as

$$\frac{d}{dt} [a(t)y(t)] = \underline{\hspace{15em}} \tag{3}$$

where

$$a'(t) = \underline{\hspace{15em}} \tag{4}$$

Solving for  $p(t)$  yields,

$$p(t) = \underline{\hspace{15em}} \tag{5}$$

Integrating both sides of equation (5) yields,

$$\int p(\lambda)d\lambda = \underline{\hspace{15em}} \tag{6}$$

Raising both sides of equation (6) to the exponential yields the *integrating factor*,

$$a(t) = e^{\int P(\lambda)d\lambda} = e^{\phi(t)}$$

Therefore, to solve an ordinary differential equation using integrating factors, multiply both sides of the equation by the integrating factor and then integrate to solve for y(t),

$$y(t) = y(t_0)e^{-(\phi(t)-\phi(t_0))} + \int_{t_0}^t e^{-(\phi(t)-\phi(\lambda))} bx(\lambda)d\lambda$$

\*\*\*\*\*

Next, we will use integrating factors to solve the standard form of the governing differential equation for the first-order circuits we have studied in prior lectures. The first step is to rewrite the standard form to look similar to equation (1),

$$\frac{dy(t)}{dt} + \frac{1}{\tau}y(t) = \frac{K}{\tau}x(t) \tag{7}$$

The integrating factor for equation (8) is

$$a(t) = e^{\int \frac{1}{\tau}dt} = e^{\phi(t)} = \underline{\hspace{10em}} \tag{8}$$

Multiply both sides of the standard equation (7) by the integrating factor

$$e^{t/\tau} \frac{dy(t)}{dt} + e^{t/\tau} \frac{1}{\tau}y(t) = \underline{\hspace{10em}} \tag{9}$$

Rewrite the left side of equation (9) as the derivative of a product

$$\underline{\hspace{10em}} = e^{t/\tau} \left[ \frac{K}{\tau}x(t) \right] \tag{10}$$

Integrate both sides of equation (10) from to t<sub>0</sub> final time t,

Finally, for y(t) which yields the general solution of the differential equation (7) for any input, x(t).

$$y(t) = y(t_0)e^{-(t-t_0)/\tau} + \int_{t_0}^t e^{-(t-\lambda)/\tau} \frac{K}{\tau}x(\lambda)d\lambda$$



*In-Class Activity 1:*

*Use integrating factors to solve for the general solution of the ordinary differential equation*

$$\dot{y}(t) + \cos(t) y(t) = x(t - 1)$$

*In-Class Activity 2:*

*Use integrating factors to solve for the general solution of the ordinary differential equation*

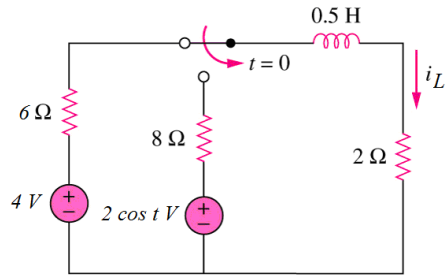
$$\frac{1}{2}\dot{y}(t) + y(t) = e^{-t}x(t)$$





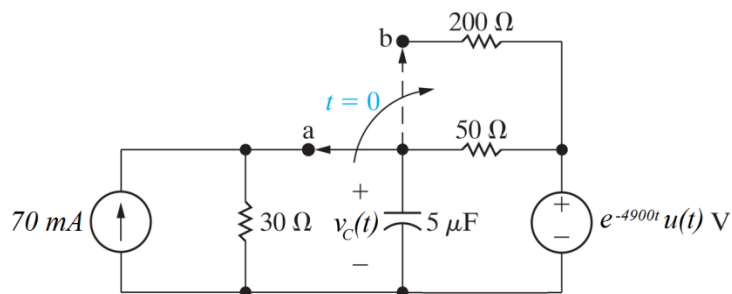
**In-Class Activity 3:**

The switch in the following circuit moves at  $t = 0$ , determine the current  $i_L(t)$  for  $t \geq 0$  s.



**In-Class Activity 4:**

The switch in the following circuit moves from a to b at  $t = 0$ . Determine the current  $v_C(t)$  for  $t \geq 0$ s.





## Lecture 2-1

### Second Order Circuits

**Reading:** Chapter 3 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define second-order circuits
- State the standard form(s) of the second order differential equation
- Derive the governing differential equation for second order circuits

A **second order circuit** is an electrical circuit that can be described by a second-order differential equation. Second-order circuits typically contain resistors, inductors, and capacitors. These types of circuits are also referred to as RLC circuits.

There are three key **characteristics** of a second-order circuit:

- The static gain,  $K$ , which represents the ratio between the output and input of the system after it reaches steady-state for a step (constant) input.
- The natural frequency,  $\omega_n$ , which indicates the oscillation frequency of the system when there is no damping
- The damping ratio,  $\zeta$ , indicates how much damping is in the system or if there is damping at all. This value is zero which indicates no damping when there are no resistors in the system.

The standard form for the governing differential equation is

$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y(t) = K\omega_n^2 x(t),$$

where  $x(t)$  is the input and  $y(t)$  is the output. Note that there are other standard forms but this is the one we will use to derive the governing differential equation.

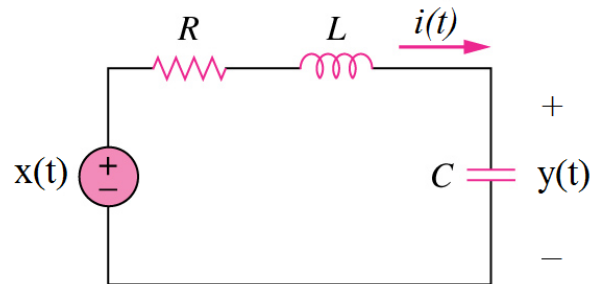
The second-order circuit also exhibits a natural and step response based upon the stored energy and application of an excitation source.

The second-order circuit also exhibits a transient and steady-state response. There are three types of responses based upon the transient and steady-state responses: **overdamped**, **underdamped** and **critically damped**.



In-Class Activity 1:

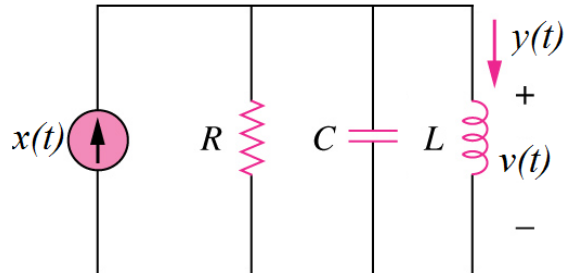
Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain, natural frequency and damping ratio.





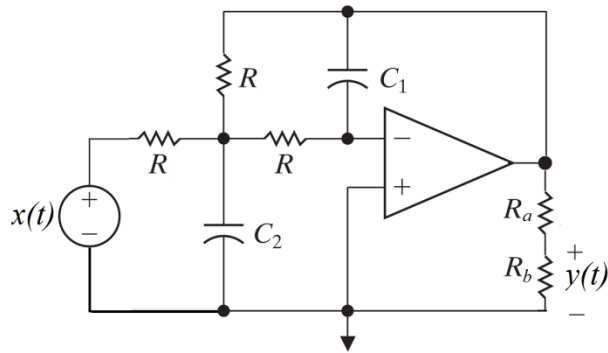
In-Class Activity 2:

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain, natural frequency and damping ratio.



In-Class Activity 3:

Derive the governing differential equation for the following circuit. Express the answer in standard form and determine the static gain, natural frequency and damping ratio.





## Lecture 2-2

## Solving Second Order Differential Equations

**Reading:** Chapter 3 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Describe the different types of responses for second-order circuits
- Solve second order differential equations of all three types

As previously discussed, it is possible to use frequency domain analysis, Laplace transforms, to solve second order differential equation with any arbitrary input,  $x(t)$ . We will study that technique later in the quarter. However, for the time domain solution we will assume a constant (step) input,  $x(t) = Au(t)$ .

The complete response of the system,  $y(t)$ , is the sum of the *natural response*,  $y_n(t)$  and *forced response*,  $y_f(t)$ . The *forced response* is due to the input assuming the initial conditions are zero and the *natural response* is due to the initial conditions or stored energy assuming the input is zero.

$$y(t) = y_n(t) + y_f(t)$$

To find the forced response:

$$\frac{d^2 y_f(t)}{dt^2} + 2\zeta\omega_n \frac{dy_f}{dt} + \omega_n^2 y_f(t) = K\omega_n^2 A$$

One particular solution is  $y_f(t) = \underline{\hspace{10em}}$

To find the natural response:

$$\frac{d^2 y_n(t)}{dt^2} + 2\zeta\omega_n \frac{dy_n}{dt} + \omega_n^2 y_n(t) = 0$$

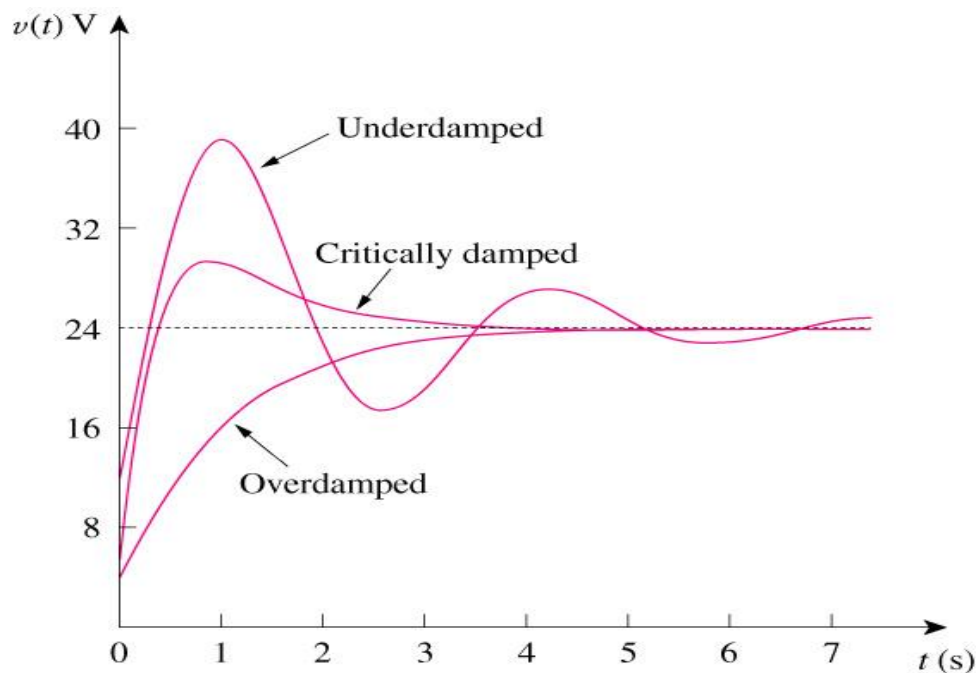
Assume the solution is of the form:  $y_n(t) = \underline{\hspace{10em}}$

$$ce^{rt}(r^2 + 2\zeta\omega_n r + \omega_n^2) = 0$$

The roots of the polynomial indicate what type of response the system will have. The qualities and characteristics are summarized in Table 1. A comparison of the three types of responses is shown in Figure 1.

**Table 1: Three types of responses of a second-order differential equation**

Type	Roots	Complete Response	Characteristics
overdamped $\zeta > 1$	2 distinct real $r_1$ and $r_2$	$y_n(t) = KA + c_1e^{r_1t} + c_2e^{r_2t}$	Slow response and long settling time
critically damped $\zeta = 1$	2 equal real $r$ and $r$	$y_n(t) = KA + c_1e^{rt} + c_2te^{rt}$	Fast response and short settling time
Underdamped $\zeta < 1$	complex conjugates $r = -\sigma \pm j\omega_d$	$y_n(t) = KA + ce^{-\sigma t} \sin(\omega_d t + \theta)$	Fastest response and long settling time
Undamped $\zeta = 0$	imaginary conjugates $r = \pm j\omega_n$	$y_n(t) = KA + c \sin(\omega_n t + \theta)$	No damping so it never settles, it oscillates forever

**Figure 1: A comparison of the three types of responses**

**In-Class Activity 1:**

Assume that the following system is initially at rest,  $y(0) = \dot{y}(0) = 0$ , and the input is a constant step,  $x(t) = A u(t)$ . What is the complete solution,  $y(t)$  for  $t \geq 0$ ?

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 4x(t)$$



**In-Class Activity 2:**

Assume that the following system is initially at rest,  $y(0) = \dot{y}(0) = 0$ , and the input is a constant step,  $x(t) = A u(t)$ . What is the complete solution,  $y(t)$  for  $t \geq 0$ ?

$$\ddot{y}(t) + 4\dot{y}(t) + 4y(t) = 4x(t)$$

**Lecture 2-3**

**Solving Second Order Differential Equations**

**Reading:** Chapter 3 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Solve second order differential equations derived from electrical circuits

**Table 1: Three types of responses of a second-order differential equation**

Type	Roots	Complete Response	Characteristics
overdamped $\zeta > 1$	2 distinct real $r_1$ and $r_2$	$y_n(t) = KA + c_1e^{r_1t} + c_2e^{r_2t}$	Slow response and long settling time
critically damped $\zeta = 1$	2 equal real $r$ and $r$	$y_n(t) = KA + c_1e^{rt} + c_2te^{rt}$	Fast response and short settling time
Underdamped $\zeta < 1$	complex conjugates $r = -\sigma \pm j\omega_d$	$y_n(t) = KA + ce^{-\sigma t} \sin(\omega_d t + \theta)$	Fastest response and long settling time
Undamped $\zeta = 0$	imaginary conjugates $r = \pm j\omega_n$	$y_n(t) = KA + c \sin(\omega_n t + \theta)$	No damping so it never settles, it oscillates forever

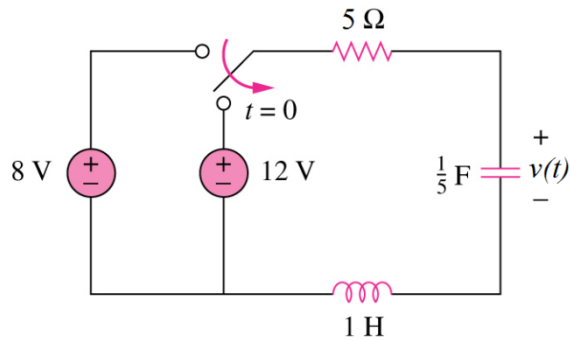
**In-Class Activity 1:**

Assume that the following system is initially at rest,  $y(0) = \dot{y}(0) = 0$ , and the input is a constant step,  $x(t) = A u(t)$ . What is the complete solution,  $y(t)$  for  $t \geq 0$ ?

$$\ddot{y}(t) + 2\dot{y}(t) + 5y(t) = 10x(t)$$

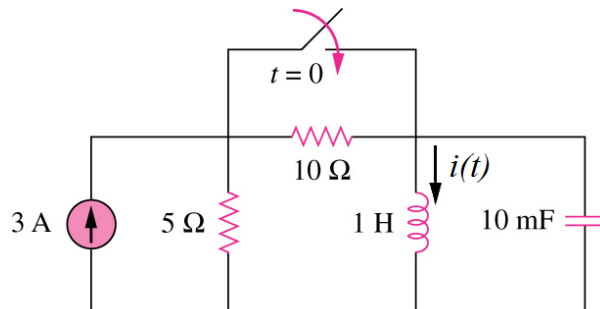
**In-Class Activity 2:**

For the following circuit, the switch has been up for a long time and moves down at  $t = 0$ , find the output voltage,  $v(t)$  for  $t \geq 0$ .



**In-Class Activity 3:**

For the following circuit, the switch has been open for a long time and closes at  $t = 0$ , find the output current,  $i(t)$  for  $t \geq 0$ .



**Lecture 3-1**

**Solving Second Order Differential Equations**

**Reading:** Chapter 3 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

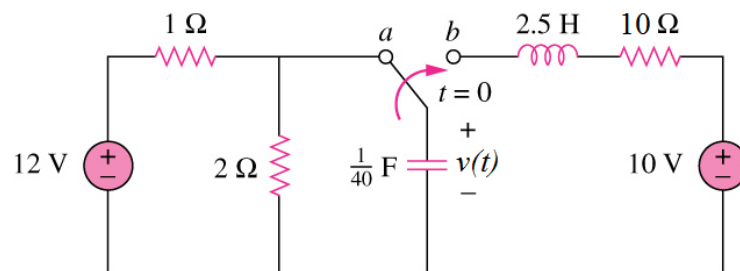
- Solve second order differential equations derived from electrical circuits

**Table 1: Three types of responses of a second-order differential equation**

Type	Roots	Complete Response	Characteristics
overdamped $\zeta > 1$	2 distinct real $r_1$ and $r_2$	$y_n(t) = KA + c_1e^{r_1t} + c_2e^{r_2t}$	Slow response and long settling time
critically damped $\zeta = 1$	2 equal real $r$ and $r$	$y_n(t) = KA + c_1e^{rt} + c_2te^{rt}$	Fast response and short settling time
Underdamped $\zeta < 1$	complex conjugates $r = -\sigma \pm j\omega_d$	$y_n(t) = KA + ce^{-\sigma t} \sin(\omega_d t + \theta)$	Fastest response and long settling time
Undamped $\zeta = 0$	imaginary conjugates $r = \pm j\omega_n$	$y_n(t) = KA + c \sin(\omega_n t + \theta)$	No damping so it never settles, it oscillates forever

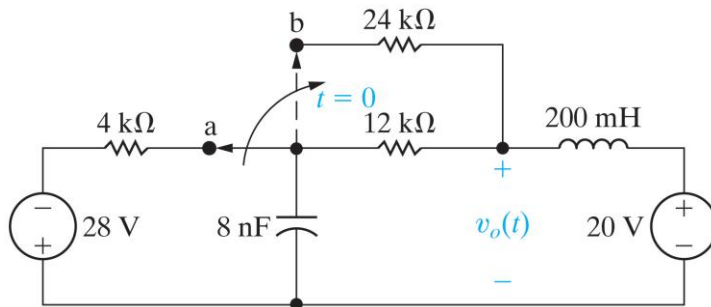
**In-Class Activity 1:**

For the following circuit, the switch has been at position *a* for a long time and moves to position *b* at  $t = 0$ , find the output voltage,  $v(t)$  for  $t \geq 0$ .



**In-Class Activity 2:**

For the following circuit, the switch has been at position *a* for a long time and moves to position *b* at  $t = 0$ , find the output voltage,  $v_o(t)$  for  $t \geq 0$ .







## Lecture 3-2

### Underdamped System Characteristics

**Reading:** Chapter 3 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Describe the characteristics of a second-order underdamped system and how they relate on the unit circle
- Describe the properties of a second-order underdamped system and determine how to find them

Recall that the second-order underdamped system has 2 complex conjugate roots of the characteristic equation,  $r^2 + 2\zeta\omega_n r + \omega_n^2$ .

Using the quadratic formula, these roots can be written as

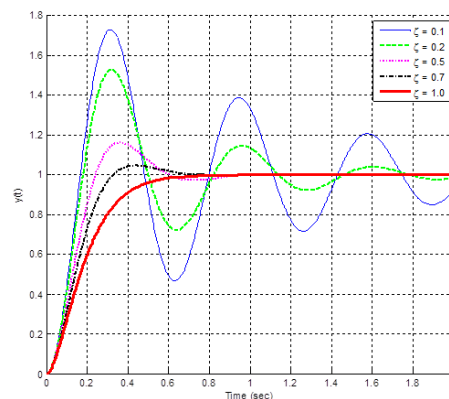
$r =$  \_\_\_\_\_

Since the expression under the quadratic can be negative, positive or zero, the damping ratio determines the type of response. Table 1 summarizes these results and Figure 1 shows the relationship.

**Table 1: Relationship between damping ration and system response**

Damping ratio, $\zeta$	Type of response
0	undamped
<1	underdamped
1	critically damped
>1	overdamped

Figure 1 presents the step response for several systems based upon the damping ratio.



**Figure 1: Step responses of an underdamped system**

An alternate method to define the roots of the characteristic equation of an underdamped system is

$r = \underline{\hspace{15em}}$ ,

where  $\sigma$  is the damping factor and  $\omega_d$  is the damping frequency. It is possible to plot the roots of the characteristic equation on a coordinate plane with a real and imaginary axis. This plot quickly illustrates the relationship between the parameters on a circle (see Figure 2).

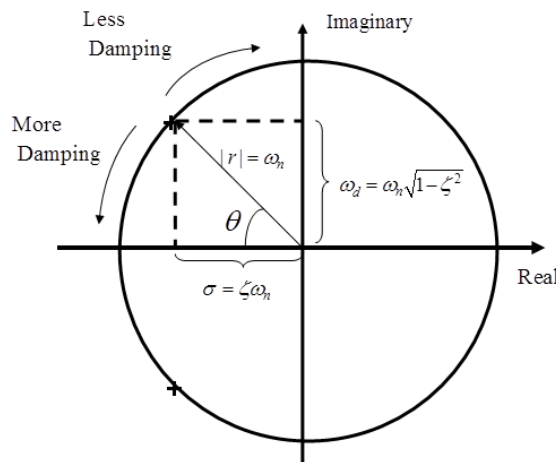


Figure 2: Second-order system characteristics on a circle

$R^2 = \sigma^2 + \omega_d^2 = \underline{\hspace{10em}}$        $\rightarrow$        $\omega_n = \underline{\hspace{10em}}$

$\cos(\theta) = \underline{\hspace{10em}}$        $\rightarrow$        $\zeta = \underline{\hspace{10em}}$

$\sin(\theta) = \underline{\hspace{10em}}$

For an underdamped system initially at rest or has no stored energy,  $y(0) = 0$  and  $\dot{y}(0) = 0$

$y(t) = KA + ce^{-\zeta\omega_n t} \sin(\omega_d t + \theta)$        $\rightarrow$        $y(0) = \underline{\hspace{10em}}$

$\dot{y}(t) = \underline{\hspace{10em}}$        $\rightarrow$        $\dot{y}(0) = \underline{\hspace{10em}}$

$\tan(\theta) = \underline{\hspace{10em}}$

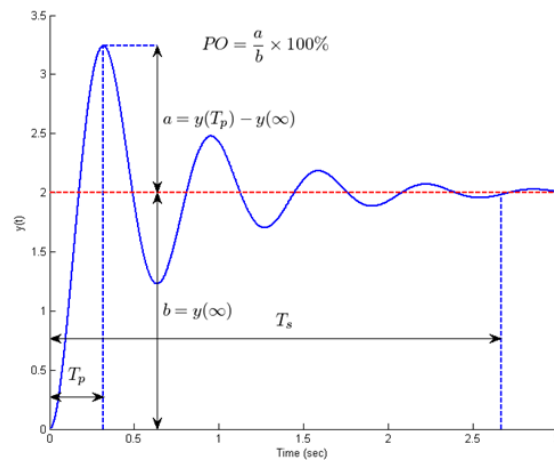
So the hypotenuse of the triangle is

$R = \sin^2(\theta) + \cos^2(\theta) = \underline{\hspace{10em}} = 1$  or creates a unit circle.

$$c = \frac{-KA}{\sqrt{1 - \zeta^2}}$$

$$y(t) = KA \left[ 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \right], \text{ where } \theta = \tan^{-1} \left[ \frac{\sqrt{1-\zeta^2}}{\zeta} \right]$$

There are three second-order underdamped system properties or parameters including **time to peak ( $T_p$ )**, **percent overshoot (PO)**, and **settling time ( $T_s$ )**.



**Figure 3: Second-order system properties**

The **time to peak,  $T_p$**  is found by taking the derivative of the complete response and setting it equal to zero. This produces the maximum value of  $y(t)$  and the time when this occurs is  $T_p$ .

$$T_p = \frac{\pi}{\omega_d}$$

The **percent overshoot, PO** is found by finding the difference between the maximum value of  $y(t)$  and the steady-state value of  $y(t)$  as a percentage of the steady-state value.

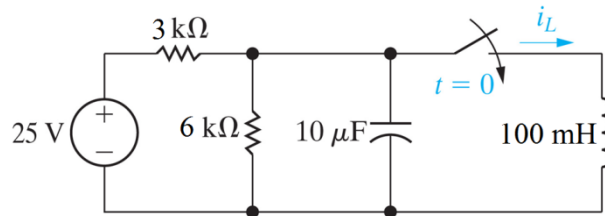
$$PO = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\%$$

The **settling time,  $T_s$**  is defined as the time it takes for the output of a system with a step input to stay within 2% of its final value, this is generally 4 time constants. Since  $\sigma = \zeta\omega_n$ , the time constant  $\tau = 1/\sigma$  is what we will use to estimate the settling time.

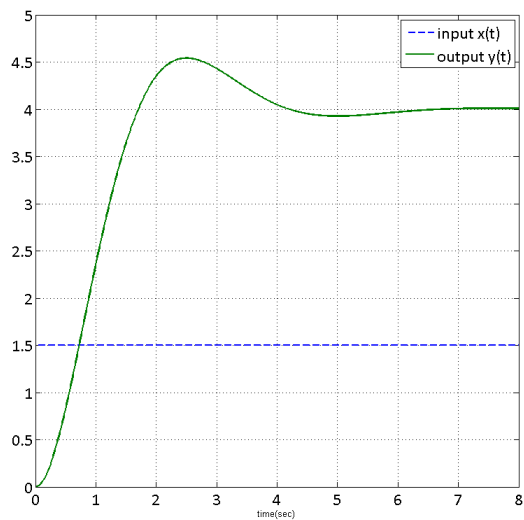
$$T_s = \frac{4}{\zeta\omega_n}$$

**In-Class Activity 1:**

Determine the time to peak, percent overshoot and settling time for the following electrical system.

**In-Class Activity 2:**

Estimate the system parameters: static gain, time to peak, percent overshoot, settling time for the response in the following figure.



**Lecture 4-1**

**System Properties: Linearity, Time-Invariance**

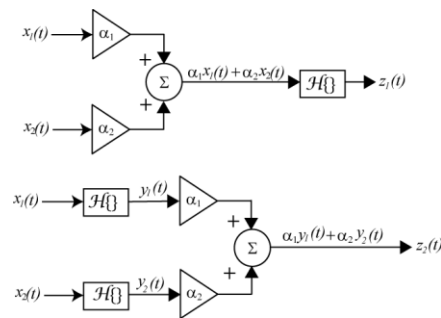
**Reading:** Chapter 4 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define the system properties: linearity, time-invariance
- Determine if a system is linear and/or time-invariant,

A system is **linear** if it is both additive and homogeneous (scaling) which means that for an input of  $\alpha_1x_1(t) + \alpha_2x_2(t) \rightarrow \alpha_1y_1(t) + \alpha_2y_2(t)$ . A quick way to determine if a system is homogeneous is to test whether the output is zero when the input is zero so there can't be any energy stored in the system or initial conditions. A system that is not linear is said to be **nonlinear**.

The signal flow graph in Figure 1 can be used to test a system for linearity.

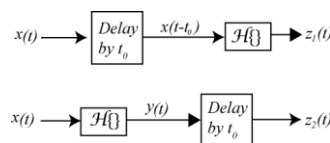


**If  $z_1 = z_2$ , then the system is linear**

**Figure 1: Signal flow graph for linearity test**

A system is **time-invariant** if for the input  $x(t)$  at any time  $t_1$ , the response to the shifted input  $x(t - t_1)$  is equal to  $y(t - t_1)$ , where  $y(t)$  is the response to  $x(t)$  with zero initial energy. A system is **time-varying** if it is not time-invariant.

The signal flow graph in Figure 2 can be used to test a system for time-invariance.

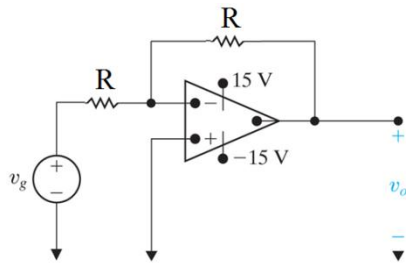


**If  $z_1 = z_2$ , then the system is time-invariant**

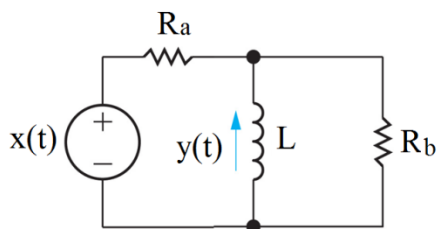
**Figure 2: Signal flow graph for time-invariance test**

In-Class Activity 1:

Determine if the following operational amplifier circuit is linear and time-invariant.

In-Class Activity 2:

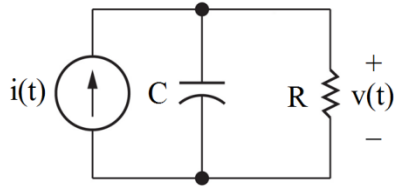
Determine if the following RL circuit is linear and time-invariant assuming zero initial conditions.





*In-Class Activity 3:*

Determine if the following RC circuit is linear and time-invariant.



*In-Class Activity 4:*

Determine if the following system  $\dot{y}(t) + 2ty(t) = x(t + 1)$  is linear and time-invariant.





## Lecture 4-2

## System Properties: Causality, Memoryless

**Reading:** Chapter 4 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

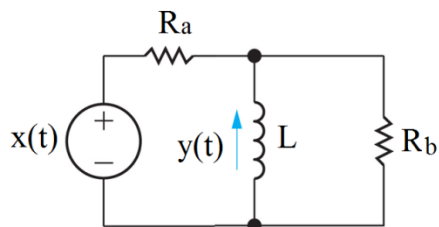
- Define the system properties: causality, memoryless
- Determine if a system is causal and/or memoryless

A system is **causal** if for any time  $t_1$ , the output response,  $y(t_1)$ , resulting from input  $x(t)$  does not depend on values of the input  $x(t)$  for  $t > t_1$ . In other words, in a **causal** system it is not possible to get an output before an input is applied to the system (assuming no initial energy). A system is **noncausal** if it is not causal. In a causal system, the output only depends on present and past values of the input.

A system is said to be **memoryless** if for any time  $t_1$ , the value of the output at time  $t_1$  depends only on the value of the input at time  $t_1$ . A causal system that is not memoryless is said to have **memory**. In a memory less system, the output only depends on present values of the input.

In-Class Activity 1:

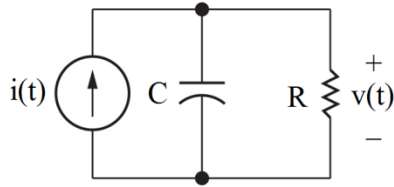
Determine if the following RL circuit is linear and time-invariant assuming zero initial conditions.





-Class Activity 2:

Determine if the following RC circuit is causal and memoryless.



*In-Class Activity 3:*

Determine if the following system  $\dot{y}(t) + 2ty(t) = x(t + 1)$  is causal and memoryless.

**In-Class Activity 4:**

For each of the following mathematic descriptions of a system, determine if it is linear, time-invariant, causal and memoryless and complete the table. You should use a formal technique to show that the system is linear.

System	Linear	Time-Invariant	Causal	Memoryless
$y(t) = x(t) + b$				
$y(t) = x^2(t)$				
$y(t) = 1/x(t)$				
$y(t) =  x(t) $				
$y(t) = \sin(x(t))$				
$y(t) = \alpha(t)x(t)$				
$y(t) = x(t/2)$				
$y(t) = x(t+1)$				
$y(t) = x(t-1)$				
$y(t) = x(1 - t/2)$				
$y(t) = e^{t+1}x(t)$				



## Lecture 4-3

## System Properties: Invertible and BIBO Stability, Linear Time-Invariant (LTI) Systems

**Reading:** Chapter 4 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define the system properties: invertibility, BIBO Stability
- Determine if a system is invertible, and/or BIBO stable

Review of system properties:

A system is **linear** if it is both additive and homogeneous (scaling) which means that for an input of  $\alpha_1 x_1(t) + \alpha_2 x_2(t) \rightarrow \alpha_1 y_1(t) + \alpha_2 y_2(t)$ . A quick way to determine if a system is homogeneous is to test whether the output is zero when the input is zero so there can't be any energy stored in the system or initial conditions. A system that is not linear is said to be **nonlinear**.

A system is **time-invariant** if for the input  $x(t)$  at any time  $t_1$ , the response to the shifted input  $x(t - t_1)$  is equal to  $y(t - t_1)$ , where  $y(t)$  is the response to  $x(t)$  with zero initial energy. A system is **time-varying** if it is not time-invariant.

A system is **causal** if for any time  $t_1$ , the output response,  $y(t_1)$ , resulting from input  $x(t)$  does not depend on values of the input  $x(t)$  for  $t > t_1$ . In other words, in a **causal** system it is not possible to get an output before an input is applied to the system (assuming no initial energy). A system is **noncausal** if it is not causal. In a causal system, the output only depends on present and past values of the input.

A system is said to be **memoryless** if for any time  $t_1$ , the value of the output at time  $t_1$  depends only on the value of the input at time  $t_1$ . A causal system that is not memoryless is said to have **memory**. In a memory less system, the output only depends on present values of the input.

There are two special cases for determining linearity and causality when the function is described in terms of a differential equation or integral. These are particularly useful when you cannot explicitly write the output as a function of the input.

For linearity, form the homogeneity and scaling equation in the differential equation and then evaluate.

For time-invariance, you can apply the rule that if there are no initial conditions and you integrate over all time, then the system is time-invariant if it is ONLY a function of the input,  $x(t)$  and output,  $y(t)$  with no other functions of time such as  $x(t + 1)$ ,  $x(2t)$ ,  $\sin(t)$ ,  $e^{2t}$ , etc. You can also use substitution on the variables of integration to determine equivalence.



*In-Class Activity 1:*

*Determine if the system described by the following differential equation is linear and time invariant.*

$$\dot{y}(t) + y(t) = x(t) + 2$$

*In-Class Activity 2:*

*Determine if the system described by the following differential equation is linear and time invariant.*

$$\dot{y}(t) + y(t)x(t) = x(t)$$



In-Class Activity 3:

Determine if the system described by the following differential equation is linear and time invariant.

$$\dot{y}(t) + \sin^2(t)y(t) = x(t)$$

A system is **invertible** if unique inputs produce unique outputs. There is a one-to-one relationship between the system input and the system output. In other words, given  $y$  you should be able to determine the exact input.

In-Class Activity 4:

Determine if  $y(t) = |x(t)|$  is invertible.

In-Class Activity 5:

Determine if  $y(t) = 1/x(t)$  is invertible.

In-Class Activity 6:

Determine if  $y(t) = \sin(x(t))$  is invertible.

In-Class Activity 7:

Determine if  $y(t) = x^2(t)$  is invertible.

In-Class Activity 8:

Determine if  $y(t) = \sqrt{x(t)}$  is invertible.



A system is **bounded-input bounded-output (BIBO) stable** if bounded inputs produce bounded outputs. Mathematically, if  $|x(t)| \leq M$  for some finite constant  $M$  then  $|y(t)| \leq N$  for some finite constant  $N$ , then the system is BIBO stable.

In-Class Activity 9:

Determine if  $y(t) = e^{x(t)}$  is BIBO stable.

In-Class Activity 10:

Determine if  $y(t) = \cos(1/x(t))$  is BIBO stable.

In-Class Activity 11:

Determine if  $y(t) = \int_0^t e^{-(t-\lambda)} x(\lambda) d\lambda$  is BIBO stable.

In-Class Activity 12:

Determine if  $y(t) = u(x(t))$  is BIBO stable and invertible.

In-Class Activity 13:

Determine if  $y(t) = \cos(2\pi t) x(t)$  is BIBO stable and invertible.

This course will focus on linear time-invariant (LTI) systems such that if the response of the system to a specific input is known then it is possible to determine the response of the system to any linear and time shifted combination of inputs.

$$\sum_i \alpha_i x_i(t - t_i) \rightarrow \sum_i \alpha_i y_i(t - t_i)$$



### Lecture 5-1

#### Impulse and Step Functions

**Reading:** Chapter 5 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Determine the output of a LTI system given the input
- Define impulse and step functions

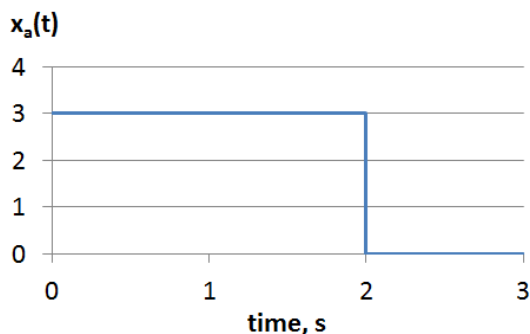
Recall that this course will focus on linear time-invariant (LTI) systems such that if the response of the system to a specific input is known then it is possible to determine the response of the system to any linear and time shifted combination of inputs.

$$\sum_i \alpha_i x_i(t - t_i) \rightarrow \sum_i \alpha_i y_i(t - t_i)$$

The primary benefit for a LTI system is that if you know the output for a given input then you can determine the output for any arbitrary input.

*In-Class Activity 1:*

If  $x(t)$  is a LTI system and  $x(t) = u(t) \rightarrow y(t) = e^{-t} u(t)$ , what is the output due to the following input,  $x_a(t)$ .



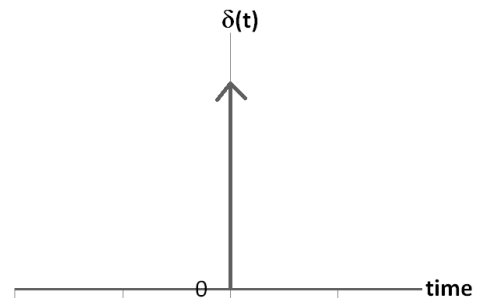
*In-Class Activity 2:*

If  $x(t)$  is a LTI system where  $x(t) = u(t) \rightarrow y(t) = e^{-t} u(t)$  and  $x(t) = \cos(t) u(t) \rightarrow y(t) = \sin(2t) u(t)$ , what is the output,  $y(t)$  due to the following input,  $x(t) = [3 - \sin(t)]u(t)$ ?

Impulse Functions

An **impulse or delta ( $\delta$ ) function** is defined as

- $\delta(t) = 0$ , for  $t \neq 0$  (zero everywhere except at zero)
- $\int_{-\mu}^{\mu} \delta(\lambda) d\lambda = 1$  (area under the curve is one)
- $\delta(0)$  is not defined



Property 1:  $\phi(t)\delta(t - t_o) = \phi(t_o)\delta(t - t_o)$

Property 2: Sifting Property

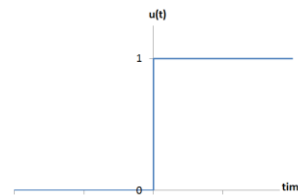
$$\int_a^b \phi(\lambda)\delta(\lambda - t_o)d\lambda = \int_a^b \phi(t_o)\delta(\lambda - t_o)d\lambda = \phi(t_o) \int_a^b \delta(\lambda - t_o)d\lambda =$$

$$\phi(t_o) \begin{cases} 1, & a < t_o < b \\ 0, & \text{else} \end{cases} = \begin{cases} \phi(t_o), & a < t_o < b \\ 0, & \text{else} \end{cases}$$

Step Functions

A **unit step or Heaviside function,  $u(\tau)$**  is defined as

$$u(\tau) = \begin{cases} 1, & \tau > 0 \\ 0, & \text{else} \end{cases}$$



$$u(t) = \int_{-\infty}^t \delta(\lambda) d\lambda$$

In-Class Activity 2:

Solve the following integral,  $\int_{-\infty}^{\infty} u(5 - \lambda)u(\lambda - 3)d\lambda$

In-Class Activity 3:

Solve the following integral,  $\int_{-\infty}^t e^{-\lambda} u(\lambda - 2) d\lambda$

In-Class Activity 4:

Solve the following integral,  $\int_{-t+2}^{\infty} e^{-\lambda} u(2 - \lambda) d\lambda$

In-Class Activity 5:

Solve the following integral,  $\int_{-\infty}^t \delta(\lambda) d\lambda$

In-Class Activity 6:

Solve the following integral,  $\int_{-t+1}^5 \delta(\lambda - 1) d\lambda$

In-Class Activity 7:

Solve the following integral,  $\int_{-6}^{t-3} \delta(\lambda + 2) d\lambda$

In-Class Activity 8:

Solve the following integral,  $\int_{-\infty}^{t-3} e^{-(t-\lambda)} \delta(\lambda - 2) d\lambda$

In-Class Activity 9:

Solve the following integral,  $\int_{-t-1}^{\infty} e^{-(t+\lambda)} \delta(\lambda + 3) d\lambda$

In-Class Activity 10:

Solve the following integral,  $\int_{-\infty}^{\infty} u(t - \lambda) u(\lambda - 1) d\lambda$

## Lecture 5-2

### Impulse and Step Response of a System

**Reading:** Chapter 5 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Determine the impulse response of an LTI system
- Determine the step response of an LTI system

The ***impulse response,  $h(t)$***  of a causal linear time-invariant continuous-time system is the output response,  $y(t)$ , when the input,  $x(t)$  is the unit impulse,  $\delta(t)$ , with no initial energy in the system.



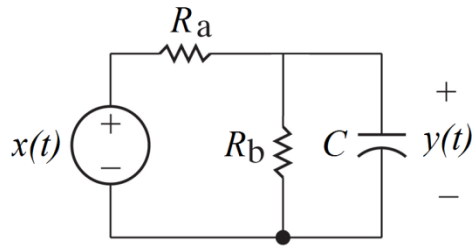
#### In-Class Activity 1:

What is the impulse response of the following first order differential equation?

$$\tau \dot{y}(t) + y(t) = Kx(t)$$

In-Class Activity 2:

What is the impulse response of the system described by the following circuit?

In-Class Activity 3:

What is the impulse response of the system described by the following mathematical model?

$$y(t) = x(t - 1) + \int_{-\infty}^t e^{-(t-\lambda)} x(\lambda + 2) d\lambda$$



The **unit step response,  $s(t)$**  of a causal linear time-invariant continuous-time system is the output response,  $y(t)$ , when the input,  $x(t)$  is the unit step,  $u(t)$ , with no initial energy in the system.



**In-Class Activity 4:**

What is the step response of the following first order differential equation?

$$\tau \dot{y}(t) + y(t) = Kx(t)$$

In the real world, it is very difficult to create an impulse response in a lab for testing a system but the unit step is easy to produce. It is not easy to measure  $h(t)$  directly because the system cannot respond fast enough to the impulse. However,

$$h(t) = \frac{d}{dt}[s(t)]$$

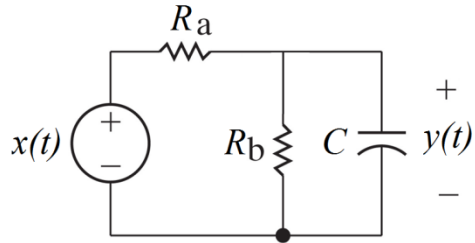
**In-Class Activity 5:**

If the step response of a system is measured to be  $s(t) = K[1 - e^{-t/\tau}] u(t)$ , what is the impulse response?



In-Class Activity 6:

Find the step response of the system with the following circuit then use the step response to determine the impulse response.







## Lecture 5-3

## Convolution Examples

**Reading:** Chapter 5 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

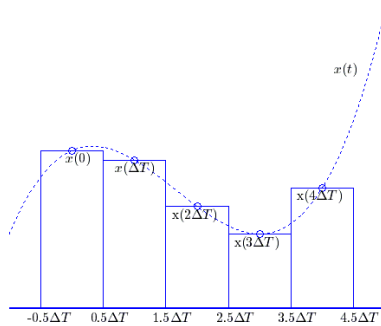
- Define the convolution integral and use it to find the output of a system
- Solve the convolution to find the output of a system

The convolution integral is used to determine the output of an LTI system given the impulse response.

This is represented by  $y(t) = x(t) * h(t)$ , where  $*$  is the convolution operator.

There are two methods for computing the convolution: analytical and graphical

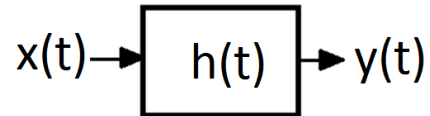
Derivation of the convolution integral





The output response of a system can be found by using the convolution integral,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)h(t - \lambda)d\lambda$$



Properties of the convolution are given in the following table.

Property	Example
Associative	
Commutative	
Distributive	
Convolution with a unit impulse	
Convolution with the shifted unit impulse	

*In-Class Activity 1 (Analytical):*

Determine the output,  $y(t)$ , of a system with  $h(t) = e^{-2t}u(t+1)$  and  $x(t) = e^{-t}u(t-2)$ .



*In-Class Activity 2 (Analytical):*

*Determine the output,  $y(t)$ , of a system with  $h(t) = e^{-t}u(t)$  and  $x(t) = u(t-1)$ .*



*In-Class Activity 3 (Graphical):*

Determine the output,  $y(t)$ , of a system with  $h(t) = e^{-t}u(t)$  and  $x(t) = u(t-1)$ .

**Lecture 6-1****Convolution Examples**

**Reading:** Chapter 5 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Solve the convolution to find the output of a system

*In-Class Activity 1:*

*Determine the output,  $y(t)$ , of a system with  $h(t) = \delta(t-1)$  and  $x(t) = \delta(t+2)$ .*

*In-Class Activity 2:*

Determine the output,  $y(t)$ , of a system with  $h(t) = e^{-(t+2)}u(t+2)$  and  $x(t) = [u(t-1)-u(t-2)] + 2[u(t-4)-u(t-6)]$ .

**In-Class Activity 3:**

Determine the output,  $y(t)$ , of a system with  $h(t) = t[u(t+1)-u(t-3)]$  and  $x(t) = [u(t-1)-u(t-2)] + [u(t-5)-u(t-7)]$ .



*In-Class Activity 4:*

Determine the output,  $y(t)$ , of a system with input  $x(t) = u(t) - 2u(t-1) + u(t-2)$  and  $h(t) = (\sin t)u(t-2)$





## Lecture 6-2

**BIBO Stability, Causality, Interconnected Systems**

**Reading:** Chapter 5 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define BIBO stability and causal systems
- Determine if a system is BIBO stable and/or causal
- Use the properties of interconnected systems to determine the output

Recall that for a **causal** system, the system output  $y(t_0)$  at some arbitrary time  $t_0$  can only depend on the system input  $x(t)$  up until (and including) time  $t_0$ . It is possible to use the convolution integral to determine if a system is causal.

For a **causal** system,  $h(t) = 0$  for  $t < 0$ .

Recall that for a **BIBO stable** system, the system output is bounded for every bounded input. It is assumed that all of the initial conditions are zero.

For a **BIBO stable** system,  $\int_{-\infty}^{\infty} h(\lambda) d\lambda$  is finite.

In-Class Activity 1:

Determine if the following systems are BIBO stable and/or causal.

i)  $h(t) = \delta(t-2)$

ii)  $h(t) = u(t+1)$

iii)  $h(t) = e^t u(t)$

iv)  $h(t) = e^{-t/2} u(t)$

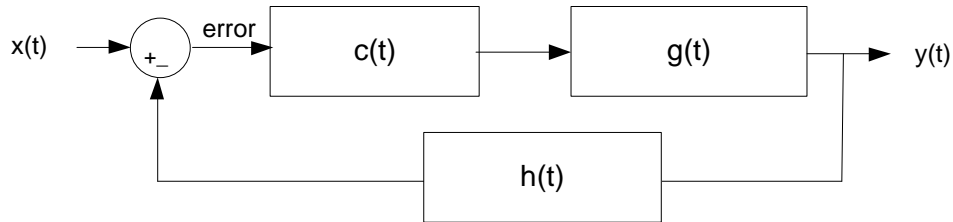
Convolution of Interconnected Systems

A block diagram can be used to simulate a differential equation. In lab, we have used this concept to find the output of a differential equation by using MATLAB and SIMULINK. See examples of the convolution of interconnected systems in the following table:

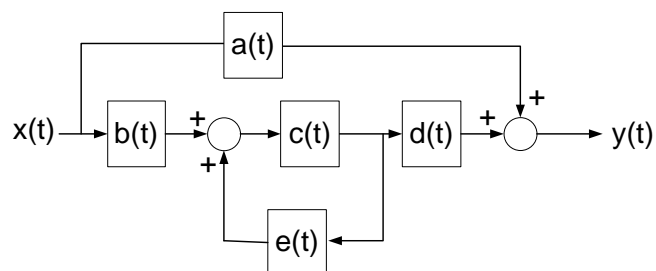
Property	Diagram	Expression
Commutative		
Associative		
Distributive		

In-Class Activity 2:

Use the properties of an interconnected system to determine the output,  $y(t)$  of the following system.

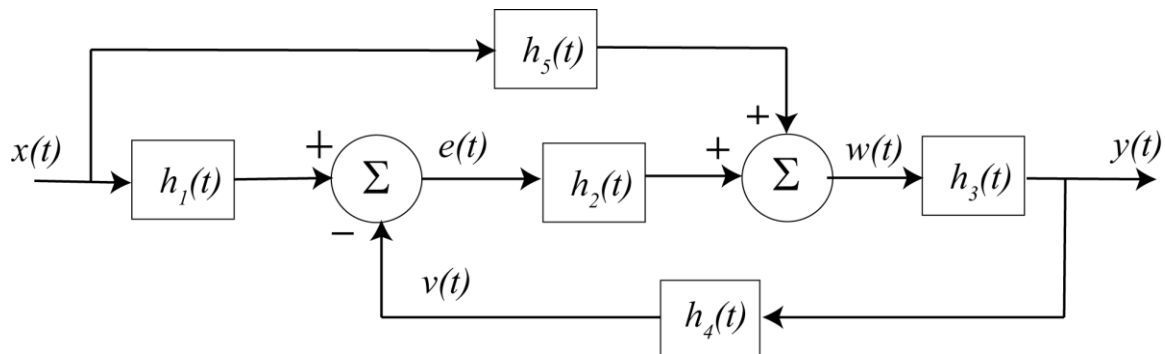
In-Class Activity 3:

Use the properties of an interconnected system to determine the output,  $y(t)$  of the following system.



In-Class Activity 4:

Use the properties of an interconnected system to determine the output,  $y(t)$  of the following system.



*In-Class Activity 3:*

*Use the poles to determine if the systems represented by the following transfer functions are stable. If it is stable, what type of response does it have?*

$$a) \quad H(s) = \frac{5}{(s^2 + 2s + 1)}$$

$$b) \quad H(s) = \frac{3s + 12}{s^2 + 5s + 6}$$

$$c) \quad H(s) = \frac{8s^2 + 24s}{2s^2 + 10s + 18}$$

### Lecture 7-1

#### Laplace Transform of Circuit Elements

**Reading:** Chapter 6 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define the one-sided Laplace Transform
- Redraw circuit elements in the frequency domain
- Solve circuit analysis problems using the Laplace Transform

As the interconnected systems become more complicated, it is difficult to use differential equations to solve for the output. The Laplace transform provides a means to solve differential equations by converting them to the complex frequency ( $s = \sigma + j\omega$ ) domain and then using algebra to determine the output.

The one-sided Laplace transform is defined as \_\_\_\_\_

It includes the time  $0^-$ , the instant before  $t = 0$  so that it can include the impulse,  $\delta(t)$ . The convention for representing the Laplace transform is

$$x(t) \xrightarrow{L} X(s) \qquad \mathcal{L}\{x(t)\} \leftrightarrow X(s)$$

Laplace transforms can be related to the phasor domain or Fourier transforms by letting  $s = j\omega$ .

*In-Class Activity 1:*

*Use the definition of the one-sided Laplace transform to find the Laplace transform of  $x(t) = \delta(t)$ .*



*In-Class Activity 2:*

*Use the definition of the one-sided Laplace transform to find the Laplace transform of*

$$x(t) = \delta(t-t_0).$$

*In-Class Activity 3:*

*Use the definition of the one-sided Laplace transform to find the Laplace transform of*

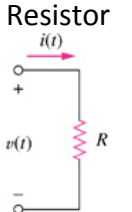
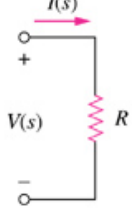
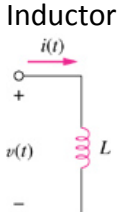
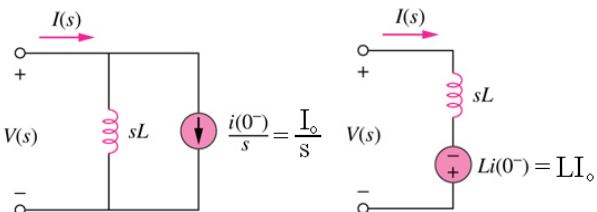
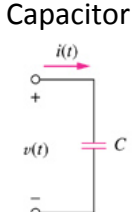
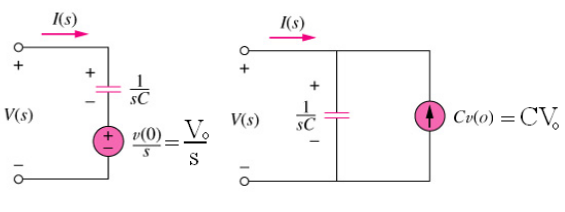
$$x(t) = u(t).$$

*In-Class Activity 4:*

*Use the definition of the one-sided Laplace transform to find the Laplace transform of*

$$x(t) = e^{-at}u(t).$$

The following table summarizes the relationship between resistors, inductors, capacitors in the complex frequency domain. These impedance relationships can be related to the phasor (sinusoidal steady-state) domain where  $s = j\omega$  after the transient has died off or  $\sigma = 0$ .

Time Domain	Voltage Current Relationships	Laplace Transform	Complex Frequency Domain (s-domain)
			
			
			

Note that the two frequency domain circuits for the inductor and capacitor initial conditions represent the Thevenin and Norton equivalent circuits with respect to the terminals for those circuits. The relationship between the voltage and current in the frequency domain is similar to Ohm's Law,

$$\underline{V} = \underline{IZ}$$

Element	Impedance ( $\Omega$ )	Admittance (S)
R		
L		
C		

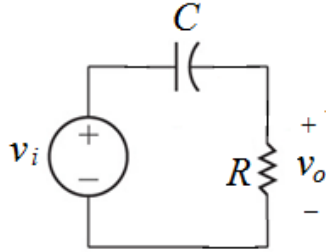
The total system response in the complex frequency domain is the sum of the **zero state response (ZSR)** and the **zero input response (ZIR)**. The ZSR is the output due to the input only, i.e. the initial conditions are zero. The ZIR is due to the initial conditions only, i.e. the input is zero.





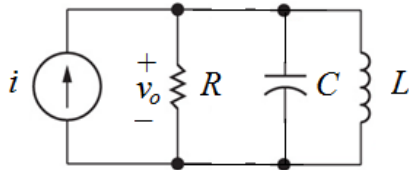
In-Class Activity 5:

Find the output  $V_o(s)$  for the following circuit and then identify the ZSR and ZIR.



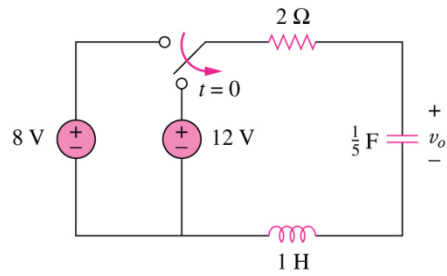
In-Class Activity 6:

Find the output  $V_o(s)$  for the following circuit and then identify the ZSR and the ZIR.



**In-Class Activity 7:**

Find the output  $V_o(s)$  for the following circuit and then identify the ZSR and ZIR.





## Lecture 7-2

## Transfer Functions, Poles and Zeros

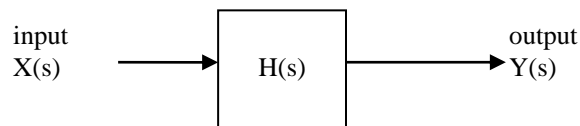
**Reading:** Chapter 6 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define the transfer function of a system
- Calculate the transfer function of a system
- Find the poles and zeros of a system and use them to determine if the system is stable

The **transfer function** of a system is defined as the s-domain ratio of the Laplace transform of the output (response) to the Laplace transform of the input (source). To find the transfer function of a system, all of the **initial conditions must be zero**.

The transfer function is  $\underline{H(s) = \frac{\text{output}}{\text{input}} = \frac{Y(s)}{X(s)}}$

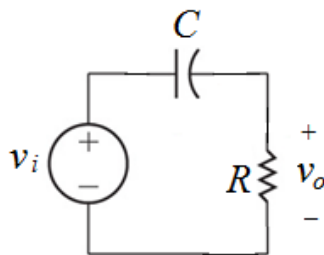


In the time domain, the system output is found from the convolution,  $\mathbf{y(t) = h(t) * x(t)}$

In the frequency domain, the system output is found from the product,  $\mathbf{Y(s) = H(s)X(s)}$

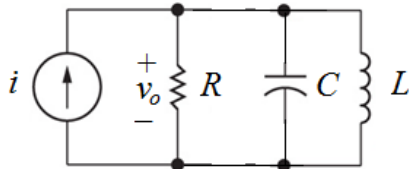
In-class Activity 1:

What is the transfer function,  $V_o(s)/V_i(s)$ , for the following circuit?

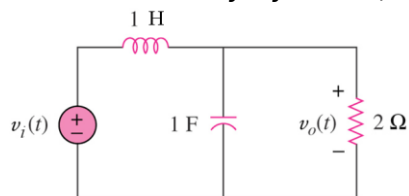


In-class Activity 2:

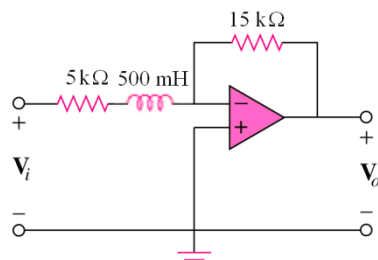
What is the transfer function,  $V_o(s)/I(s)$ , for the following circuit?

In-class Activity 3:

What is the transfer function,  $V_o(s)/V_i(s)$ , for the following circuit?

In-class Activity 4:

What is the transfer function,  $V_o(s)/V_i(s)$ , for the following circuit?



Poles, Zeros, and Pole-Zero plot on the s-plane

$H(s)$  is always a rational function of  $s$ .

$$H(s) = \frac{N(s)}{D(s)} = \frac{A(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

The **zeros** of  $H(s)$  are the roots of the numerator,  $N(s)$ .

The **poles** of  $H(s)$  are the roots of the denominator,  $D(s)$ .

If  $n \geq m$ ,  $H(s)$  is a **proper** transfer function.

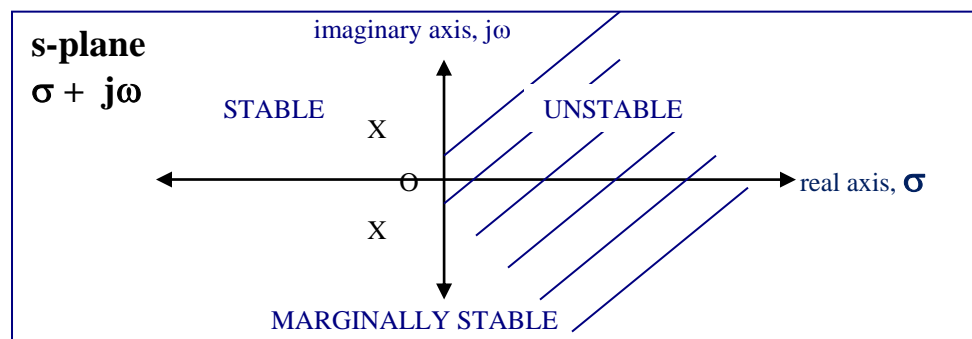
If  $n > m$ ,  $H(s)$  is a **strictly proper** transfer function.

If there are any complex poles, they always appear in conjugate pairs. The poles of  $H(s)$  must lie in the left half of the  $s$  plane if the response is to be bounded or asymptotically stable. The zeros may lie in either the right half or the left half of the  $s$  plane.

To determine system stability using the system poles,

- If all of the poles are in the **open** left-half plane, the system is **stable**
- If all of the poles are in the left-half plane (includes the imaginary axis) and no more than one at the origin, the system is **marginally stable**
- If any of the poles are in the right-half plane, the system is **unstable**

In order, to quickly identify the system response, you can plot the poles and zeros on a pole-zero plot or the  $s$ -plane. On the  $s$ -plane, the horizontal axis is the real axis and the vertical axis is the imaginary axis. The poles are represented with "X's" and the zeros are represented with "O's".



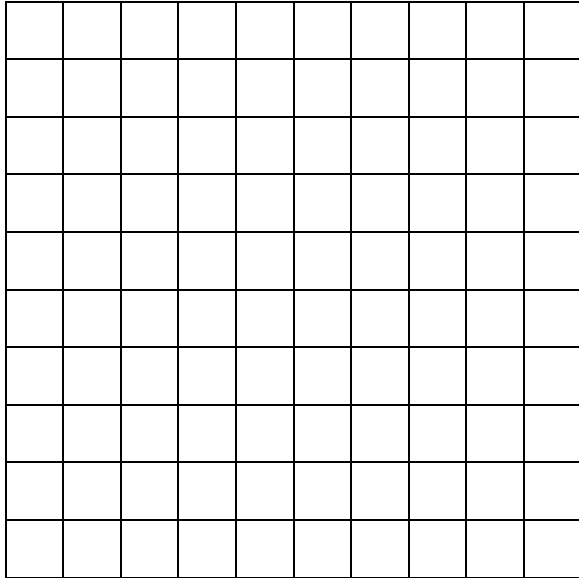
For a second-order system,

- If there are two distinct real poles, the system is **overdamped**
- If there are two complex conjugate poles, the system is **underdamped**
- If there are repeated real poles, the system is **critically damped**
- If there are two imaginary conjugate poles, the system is **undamped**

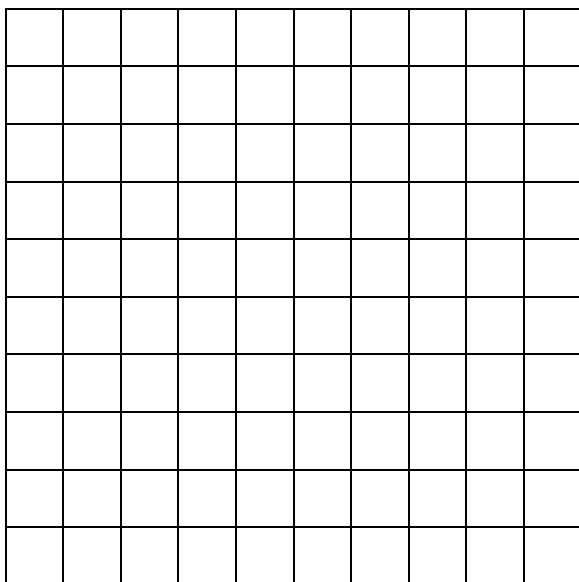
In-Class Activity 5:

Plot the poles and zeros of the system represented by the following transfer function on the pole-zero plot and determine if it is stable, unstable or marginally stable.

$$a) H(s) = \frac{s^2 + 3s + 2}{s^3 + 3s^2}$$



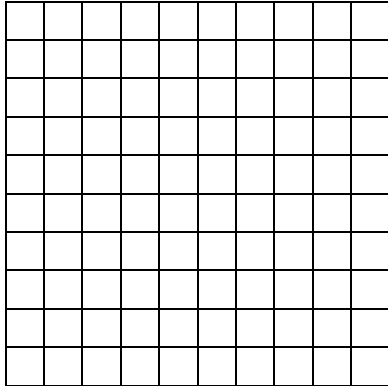
$$b) H(s) = \frac{s^2 + 2s + 2}{s^4 + 5s^3 + 17s^2 + 13s}$$



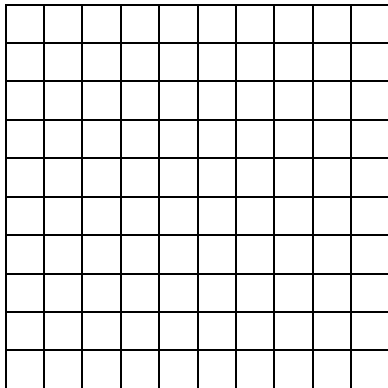
**In-Class Activity 6:**

Use the poles to determine if the systems represented by the following transfer functions are stable. If it is stable, what type of response does it have?

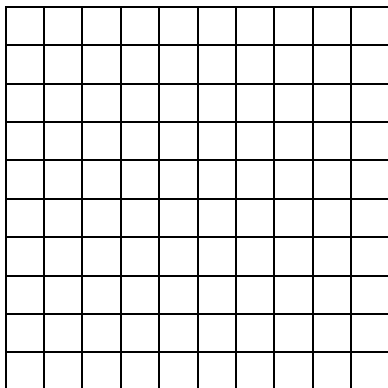
$$a) H(s) = \frac{5}{(s^2 + 2s + 1)}$$



$$b) H(s) = \frac{3s + 12}{s^2 + 5s + 6}$$



$$c) H(s) = \frac{8s^2 + 24s}{2s^2 + 10s + 18}$$





## Lecture 7-3

## Partial Fraction Expansion

**Reading:** Chapter 6 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Determine the required form to perform partial fraction expansion
- To determine the impulse response of a system using partial fraction expansion

It is possible to find the impulse response,  $h(t)$ , of a system by finding the inverse Laplace transform of the transfer function,  $H(s) = N(s)/D(s)$ . It is also possible to find the output of the system by finding the inverse Laplace transform of  $Y(s)$ .

$$Y(s) = H(s)X(s) \quad \rightarrow \quad y(t) = \mathcal{L}^{-1}\{Y(s)\}$$

In order to be able to perform partial fraction expansion,

- The rational expression must be a strictly proper ratio of polynomials
- The leading coefficient of the denominator must be one (monic)
- Temporarily remove any delay operators ( $e^{-st}$ )

**In-Class Activity 1:**

Re-write the following transfer function in the proper form to perform partial fraction expansion.

$$H(s) = \frac{s^2 + 2s + 3}{s^2 + 3s + 4}$$

**In-Class Activity 2:**

Re-write the following transfer function in the proper form to perform partial fraction expansion.

$$H(s) = \frac{s + 2}{3s^2 + 4s + 2}$$

**In-Class Activity 3:**

Re-write the following transfer function in the proper form to perform partial fraction expansion.

$$H(s) = \frac{e^{-3s}}{s^2 + 2s + 5}$$





Partial fraction expansion for distinct real roots of the denominator,  $D(s)$

In order to find the partial fraction expansion of a rational function when there are distinct real roots, you can use the cover up method.

$$H(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

$$H(s) = \frac{N(s)}{D(s)} = \frac{A_1}{s - p_1} + \frac{A_2}{s - p_2} + \dots + \frac{A_k}{s - p_k}$$

By taking the inverse Laplace transform of this expression yields,

$$h(t) = (A_1 e^{p_1 t} + A_2 e^{p_2 t} + \dots + A_k e^{p_k t}) u(t)$$

Using the cover up method, multiply both sides of  $H(s)$  by one of the terms,  $(s - p_k)$  and take the limit as  $s$  approaches  $p_k$ .

$$A_k = (s - p_k)H(s)|_{s=p_k} = \lim_{s \rightarrow p_k} (s - p_k)H(s)$$

In-Class Activity 4:

Determine the impulse response of a system with the following transfer function.

$$H(s) = \frac{2s + 3}{(s + 1)(s + 2)(s + 3)}$$

In-Class Activity 5:

Determine the impulse response of a system with the following transfer function.

$$H(s) = \frac{e^{-2s}}{s^2 + s}$$

Partial fraction expansion for repeated real roots of D(s)

There are several methods to solve for the coefficients,  $A_k$ , when there are repeated real roots of the denominator of the rational expression including using the cover up method to find the coefficient for the highest power and any terms of multiplicity one.

$$H(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s - p_1)^k}$$

$$H(s) = \frac{N(s)}{D(s)} = \frac{A_1}{s - p_1} + \frac{A_2}{(s - p_1)^2} + \dots + \frac{A_k}{(s - p_1)^k}$$

By taking the inverse Laplace transform of this expression yields,

$$h(t) = \left( A_1 e^{p_1 t} + A_2 t e^{p_1 t} + A_3 \frac{t^2}{2} e^{p_1 t} + \dots + A_k \frac{t^{k-1}}{(k-1)!} e^{p_1 t} \right) u(t)$$

To find the other coefficients, you can multiply by the term with the highest power and take the derivative one or more times. Alternately, you can multiply both sides by  $s$  and take the limit as  $s$  approaches  $\infty$  or select a value for  $s$  and solve for the coefficient.



*In-Class Activity 6:*

*Determine the impulse response of a system with the following transfer function.*

$$H(s) = \frac{1}{(s + 1)(s + 2)^2}$$



## Lecture 8-1

### Partial Fractions

**Reading:** Chapter 6 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- To determine the impulse response of a system using partial fraction expansion
- To determine the output of a system described by a transfer function nor circuit using partial fraction expansion

#### Partial fraction expansion for distinct complex conjugate roots of $D(s)$

There are several methods to solve for the coefficients,  $A_k$ , when there are complex conjugate roots of the denominator of the rational expression including writing the result in the form of the sum of a sine and cosine.

$$H(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{s^2 + as + \beta} = \frac{N(s)}{(s + a + jb)(s + a - jb)}$$

$$H(s) = \frac{N(s)}{D(s)} = \frac{A_1(s + a)}{(s + a)^2 + b^2} + \frac{A_2 b}{(s + a)^2 + b^2}$$

By taking the inverse Laplace transform of this expression yields,

$$h(t) = (A_1 e^{-at} \cos(bt) + A_2 e^{-at} \sin(bt))u(t)$$

Alternately, the cover up method can be used to find the complex coefficients and re-write the final answer as a cosine.

$$H(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s + a + jb)(s + a - jb)} = \frac{A_1}{s + a - jb} + \frac{A_1^*}{s + a + jb}$$

$$A_1 = |A_1| \angle \theta^\circ$$

By taking the inverse Laplace transform of this expression yields,

$$h(t) = (2|A_1|e^{-at} \cos(bt + \theta))u(t)$$



*In-Class Activity 1:*

*Re-write the following transfer function in the proper form to perform partial fraction expansion.*

$$H(s) = \frac{s + 2}{s^2 + 6s + 10}$$

*In-Class Activity 2:*

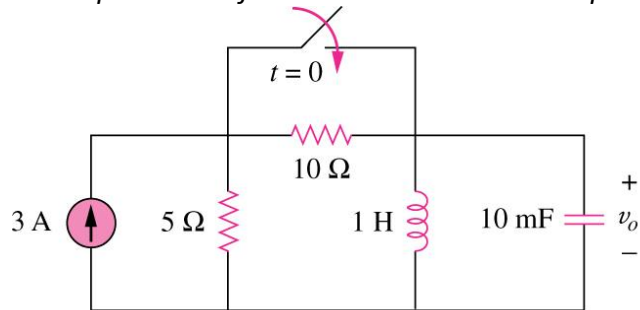
*Re-write the following transfer function in the proper form to perform partial fraction expansion.*

$$H(s) = \frac{100(s + 3)}{(s + 6)(s^2 + 6s + 25)}$$



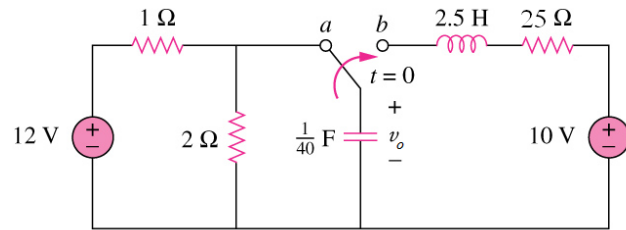
In-Class Activity 3:

Use Laplace transforms to determine the output of a system described by the following circuit.



**In-Class Activity 4:**

Use Laplace transforms to determine the output of a system described by the following circuit.





## Lecture 8-2

## Characteristic Modes, Asymptotic Stability, Initial and Final Value Theorem

**Reading:** Chapter 7 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- define the system terms: characteristic modes, dominant poles, asymptotically stable, initial and final value theorem, steady-state error, static gain
- determine if a system is asymptotically stable
- find the characteristic modes, dominant poles and system characteristics of a system with a given transfer function
- apply the initial and final value theorems to find the gain and/or steady-state error of a system with a given transfer function

The **characteristic polynomial** can be used to describe or find all of the characteristics of a system. Given the transfer function  $H(s) = N(s)/D(s)$ , denominator,  $D(s)$  is the characteristic polynomial of the system.

The **characteristic modes** of the system are found from the poles of the transfer function. They are the time-domain functions that correspond to the poles. Thus, as seen in prior lectures, the **impulse response** is the linear combination of the characteristic modes.

For example, if the transfer function of a system is

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{(s+a)(s^2+2bs+b^2+c^2)(s+d)^2}$$

The **characteristic modes** are the forms of the step response solution, in this case

$$e^{-at}, e^{-bt} \sin ct, e^{-bt} \cos ct, e^{-dt}, te^{-dt}$$

In-Class Activity 1:

Determine the characteristic polynomial, characteristic modes, and impulse response for the system described by the following transfer function.

$$H(s) = \frac{2}{s^3 + 3s^2 + 2s}$$

$$H(s) = \frac{2s}{(s-1)(s^2+2s+2)}$$





Stability is extremely important for control system design. A feedback control system cannot be designed to regulate to a set value or track a set point without stability. **BIBO stability** means that for every bounded input to a system, the output remains bounded with increasing time. Another useful definition of stability is **asymptotically stable**,

- A system is **asymptotically stable** if all of its characteristic modes go to zero as  $t \rightarrow \infty$  or if  $h(t) = 0$  as  $t \rightarrow \infty$ .
- A system is **asymptotically marginally stable** if all of its characteristic modes are bounded as  $t \rightarrow \infty$  or  $|h(t)| \leq M$  as  $t \rightarrow \infty$ .
- A system is **asymptotically unstable** if any of its characteristic modes is unbounded as  $t \rightarrow \infty$ .

### In-Class Activity 2:

Determine whether the following systems are stable, unstable or marginally stable.

a)  $h(t) = [1 + e^{-t}]u(t)$

b)  $h(t) = e^{-2t} u(t)$

c)  $h(t) = [2 + \sin(t)] u(t)$

d)  $h(t) = [1 + t + e^{-t}] u(t)$

e)  $h(t) = [1 - t^3 e^{-0.1t}] u(t)$

f)  $h(t) = [te^{-t} \cos(5t) + e^{-2t} \sin(3t)] u(t)$

The **settling time,  $T_s$**  of a system is the time it takes for the system to reach 2% of its final value, this is  $4\tau$ . The settling time can be found from the characteristics modes. To find the settling time,

- The system must be stable
- The poles of the system must be distinct
- The input to the system is a step

For example, if  $y(t) = e^{-t/\tau}$ , the settling time is equivalent to four time constants  $T_s = 4\tau$ . When there are multiple characteristic modes, the overall settling time of the system is determined by the characteristic mode with the longest settling time. The **dominant pole** provides the longest settling time. It is more difficult to define the rise time, setting time and percent overshoot for higher order systems. However, some higher order systems can be modeled as 1<sup>st</sup> or 2<sup>nd</sup> order systems if there is a **dominant pole(s)** that are significantly far enough away from some of the poles on the real part of the s-plane. A good rule of thumb is that the other poles should be at least 5 times away from the dominant poles.

**In-Class Activity 3:**

For the following system, determine the characteristic modes, dominant poles and overall settling time .

$$G(s) = \frac{1}{(s + 1)^2(s + 10)}$$

**In-Class Activity 4:**

For the following system, determine the characteristic modes, settling time and dominant pole.

$$G(s) = \frac{1}{(s + 10)(s^2 + s + 1)}$$

**In-Class Activity 5:**

For the following system, determine the characteristic modes, settling time and dominant pole.

$$G(s) = \frac{1}{(s^2 + 4s + 6)(s^2 + s + 1)}$$



The **initial and final value theorems** can be used to find the initial and final values of a system response by using the s-domain equivalent of the expression.

The initial value theorem states that  $\lim_{t \rightarrow 0^+} f(t) = \lim_{s \rightarrow \infty} sF(s)$

The final value theorem states that  $\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$

Note that these theorems are not valid if there is more than a single pole at the origin on the imaginary axis.

The **steady-state error** of a system is the difference between the input and output as  $t$  approaches  $\infty$ .

$$e_{ss} = \lim_{t \rightarrow \infty} [x(t) - y(t)] = \lim_{s \rightarrow 0} s[X(s) - Y(s)]$$

If the input is a step function,  $x(t) = Au(t)$

$$e_{ss} = \lim_{t \rightarrow \infty} [Au(t) - h(t) * Au(t)] = \lim_{s \rightarrow 0} A[1 - H(s)] = A[1 - H(0)]$$

The **static gain, K** of a system is the steady-state gain of the system when the input is a step function and all of the transients have died out. If the input is a step function,  $x(t) = A u(t) = x_{ss}(t)$ , the steady state output and gain are

$$y_{ss}(t) = \lim_{t \rightarrow \infty} [y(t)] = \lim_{s \rightarrow 0} [sY(s)] = \lim_{s \rightarrow 0} [AH(s)] = AH(0)$$

$$K = \frac{y_{ss}(t)}{x_{ss}(t)} = H(0)$$

#### In-Class Activity 6:

For the following system, determine the static gain, steady-state value and steady-state error for an input  $x(t) = 2u(t)$ .

$$H(s) = \frac{1}{(s+3)}$$

#### In-Class Activity 7:

For the following system, determine the static gain, steady-state value and steady-state error for an input  $x(t) = 3u(t)$ .

$$H(s) = \frac{s+2}{s^2 + 0.4s + 3}$$



## Lecture 8-3

## Block Diagrams, Feedback Systems

**Reading:** Chapter 7 of Course Notes

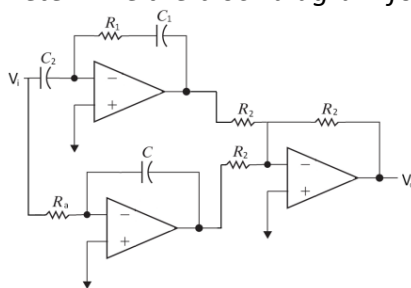
**Objective:** At the conclusion of this lecture, students should be able to

- Apply frequency domain techniques to find the transfer function of a circuit and block diagram
- Identify the components of a feedback control system
- Design a controller to improve the steady-state and transient characteristics of a system

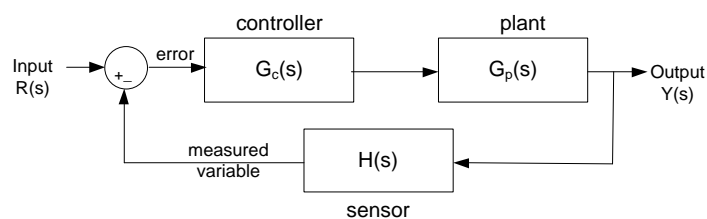
As previously stated, the benefit of Laplace transforms is that algebra can be used to find the input-output characteristics of an interconnected system or block diagram.

In-Class Activity 1:

Determine the block diagram for the system represented by the following circuit.

In-Class Activity 2:

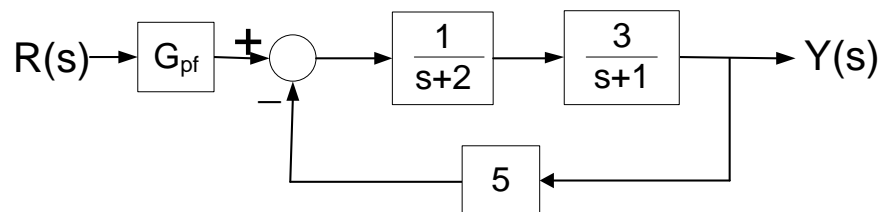
Determine the transfer function for the system represented by the following block diagram.



**Feedback control systems** are used to control a plant,  $G_p(s)$  to track a certain set point ( $R(s)$ ) or to regulate to a certain value. One common feedback control system is a thermostat where the sensor ( $H(s)$ ) is used to detect the room temperature and uses the error to provide a signal to the controller,  $G_c(s)$  which adjusts the damper or turn on a heating or air conditioning system,  $Y(s)$ . The human body temperature is an example of a self-regulating control system. There are also open loop control systems but they do not work as well because it is difficult to measure the effect of the control signal. Sometimes a prefilter,  $G_{pf}(s)$ , is used for signal conditioning in the input to a feedback control system.

*In-Class Activity 3:*

*The following block diagram describes a feedback control system. Determine the value of the prefilter to have a zero steady-state error.*





The most common types of industrial controls are

If the input to the controller is the error,  $e(t)$  and the output is the control signal,  $u(t)$ , the following table summarizes the controllers.

Controller	Time-domain	Frequency-Domain
Proportional (P)		
Integral (I)		
Proportional-Integral (PI)		
Proportional-Derivative (PD)		
PID		

The proportional and/or derivative control are used to control or change the transient response such as settling time, time to peak or percent overshoot. The integral control is used to control or change steady-state error. There are typically tradeoffs between the characteristics of each of the controllers.

*In-Class Activity 4:*

*Design a P controller,  $G_c(s) = k_p$ , to decrease the settling time and steady-state error by a factor of 2 for a system with the following plant. Assume unity feedback and a unit step input.*

$$G_p(s) = \frac{1}{s + 3}$$



*In-Class Activity 5:*

*Design a PD controller,  $G_c(s) = k_p + k_d s$ , to decrease the settling time and steady-state error by a factor of 2 for a system with the following plant. Assume unity feedback and a unit step input.*

$$G_p(s) = \frac{5}{s^2 + 2s + 2}$$



## Lecture 9-1

## Sinusoidal Steady State Frequency Response

**Reading:** Chapter 8 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define the sinusoidal steady-state response of a system
- Determine the sinusoidal steady-state response of a system given the transfer function
- Determine the sinusoidal steady-state response of a system by using the pole-zero plot

To find the steady state sinusoidal response with an input,  $\mathbf{x(t) = A \cos(\omega_0 t + \phi)}$

Use trigonometric properties to rewrite the input as  $\mathbf{x(t) = A \cos \omega_0 t \cos \phi - A \sin \omega_0 t \sin \phi}$

The Laplace transform of the input,  $\mathbf{x(s) = \frac{A \cos \phi}{s^2 + \omega_0^2} - \frac{A \omega_0 \sin \phi}{s^2 + \omega_0^2} = \frac{A(s \cos \phi - \omega_0 \sin \phi)}{s^2 + \omega_0^2}$

The output is  $\mathbf{Y(s) = H(s) X(s) = H(s) \frac{A(s \cos \phi - \omega_0 \sin \phi)}{s^2 + \omega_0^2}}$

Using partial fraction expansion,  $\mathbf{Y(s) = H(s) \text{ terms} + X(s) \text{ terms} = H(s) \text{ terms} + \frac{K_1}{s - j\omega_0} + \frac{K_1^*}{s + j\omega_0}}$

Since we have assumed that the transfer function is asymptotically stable all of the terms due to the poles of  $\mathbf{H(s)}$  will go to zero as  $\mathbf{t}$  approaches  $\infty$ . Only the  $\mathbf{X(s)}$  terms contribute to the steady-state response, the poles of the sinusoidal input,  $\mathbf{x(t)}$  are on the imaginary axis of the  $\mathbf{s}$ -plane.

$$\mathbf{K_1 = \left. \frac{H(s)A(s \cos \phi - \omega_0 \sin \phi)}{s + j\omega_0} \right|_{s = j\omega_0} = \frac{1}{2} H(j\omega_0) A e^{j\phi} \text{ where } \mathbf{H(j\omega_0) = |H(j\omega_0)| \angle \theta(\omega_0) = K_1^*}$$

The steady-state solution for  $\mathbf{y(t)}$  or the steady-state sinusoidal response is

$$\mathbf{Y_{ss}(j\omega) = (A \angle \phi) (|H(j\omega_0)| \angle \theta(\omega_0)) = A |H(j\omega_0)| e^{j(\phi + \theta(\omega))}$$

$$\mathbf{y_{ss}(t) = A |H(j\omega_0)| \cos[\omega_0 t + \phi + \theta(\omega_0)]}$$

The amplitude of the solution is equal to the amplitude of the source,  $\mathbf{A}$ , times the magnitude of the transfer function,  $\mathbf{|H(j\omega)|}$  at the frequency of the source,  $\mathbf{\omega_0}$ .

The phase angle of the response is equal to the phase angle of the source,  $\mathbf{\phi}$ , plus the phase angle of the transfer function,  $\mathbf{\theta(\omega)}$  at the frequency of the source,  $\mathbf{\omega_0}$ .



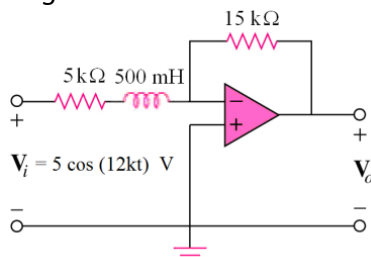
**In-Class Activity 1:**

Determine the sinusoidal steady-state response for the system with input  $x(t) = 3 \sin(2t+30^\circ)$  represented by the transfer function,

$$H(s) = \frac{3s}{(s+1)(s+2)}$$

**In-Class Activity 2:**

Determine the sinusoidal steady-state response for the system described by the following circuit diagram.





So far we have examined an analytical technique for determining the sinusoidal steady-state response, it is also possible to use the pole-zero diagram. If the strictly proper transfer function,  $H(s)$  is written by substituting  $s = j\omega_0$  we get

$$H(s) = \frac{K(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p) \dots (s - p_n)}$$

$$H(j\omega_0) = \frac{K(j\omega_0 - z_1)(j\omega_0 - z_2) \dots (j\omega_0 - z_m)}{(j\omega_0 - p_1)(j\omega_0 - p) \dots (j\omega_0 - p_n)}$$

Next, we need to represent each of these terms in polar form, rather than in rectangular form. To standardize this, let's write

$$j\omega_0 - z_i = \beta_i e^{j\phi_i}, \beta_i = \sqrt{(\operatorname{Re}(z_i))^2 + (\omega_0 - \operatorname{Im}(z_i))^2}, \phi_i = \tan^{-1} \left( \frac{\omega_0 - \operatorname{Im}(z_i)}{-\operatorname{Re}(z_i)} \right)$$

$$j\omega_0 - p_i = \alpha_i e^{j\theta_i}, \alpha_i = \sqrt{(\operatorname{Re}(p_i))^2 + (\omega_0 - \operatorname{Im}(p_i))^2}, \theta_i = \tan^{-1} \left( \frac{\omega_0 - \operatorname{Im}(p_i)}{-\operatorname{Re}(p_i)} \right)$$

$\beta_i$  = distance from the zero,  $z_i$ , to the point,  $j\omega_0$

$\alpha_i$  = distance from the pole,  $p_i$ , to the point,  $j\omega_0$

$\phi_i$  = angle (measured from the positive real axis) between zero  $z_i$  and  $j\omega_0$

$\theta_i$  = angle (measured from the positive real axis) between pole  $p_i$  and  $j\omega_0$

This yields the following representation of the sinusoidal steady-state,

$$H(j\omega_0) = \frac{K \beta_1 e^{j\phi_1} \beta_2 e^{j\phi_2} \dots \beta_m e^{j\phi_m}}{\alpha_1 e^{j\theta_1} \alpha_2 e^{j\theta_2} \dots \alpha_n e^{j\theta_n}} = \frac{K \beta_1 \beta_2 \dots \beta_m}{\alpha_1 \alpha_2 \dots \alpha_n} e^{j(\phi_1 + \phi_2 + \dots + \phi_m - \theta_1 - \theta_2 - \dots - \theta_n)}$$

This means

$$|H(j\omega_0)| = \frac{K \beta_1 \beta_2 \dots \beta_m}{\alpha_1 \alpha_2 \dots \alpha_n}$$

$$\angle H(j\omega_0) = \phi_1 + \phi_2 + \dots + \phi_m - \theta_1 - \theta_2 - \dots - \theta_n$$

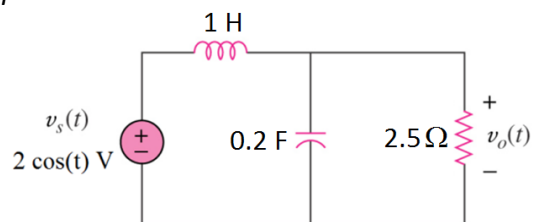
**In-Class Activity 3:**

For the system with the following transfer function, determine the frequency response,  $H(j4)$  by using the pole-zero plot.

$$H(s) = \frac{s + 2}{(s + 1)(s + 3)}$$

**In-Class Activity 4:**

Determine the frequency response for the circuit with the following input using the pole-zero plot.





## Lecture 9-2

### Bode Diagrams

**Reading:** Chapter 8 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define decibels and a Bode diagram of a system
- Sketch the straight-line approximation Bode diagram of a transfer function with constant terms, integrators and differentiators

The **bel** used to define power gain and is defined as  $\log_{10}(p_{out}/p_{in})$ .

The **decibel** is also used to measure power gain and is defined as  $10 \log_{10}(p_{out}/p_{in})$ .

**Decibels** can be used to define the voltage and current ratio as

$$10 \log_{10} \frac{P_{out}}{P_{in}} = 10 \log_{10} \left( \frac{V_{out}}{V_{in}} \right)^2 = 20 \log_{10} \frac{V_{out}}{V_{in}} \quad \text{and} \quad 20 \log_{10} \frac{I_{out}}{I_{in}}$$

**Bode diagrams** are graphical techniques that describe the frequency response of a circuit. The Bode diagram helps you quickly visualize how the system responds to a wide range of frequencies. There are two separate plots: amplitude and phase angle and they vary with the frequency.

The plots are on semi log graph paper with the frequency on the horizontal log scale and the amplitude and phase angle on the linear vertical scale.

There are several types of factors in a transfer function these include:

- a gain  $K$
- a zero  $(j\omega)$  at the origin
- a pole  $(j\omega)^{-1}$  at the origin
- a simple zero  $1 + j\omega/z_1$
- a simple pole  $1/(1+j\omega/p_1)$
- a quadratic or complex pole  $1/[1 + j2\zeta_2\omega/\omega_n + (j\omega/\omega_n)^2]$
- a quadratic or complex zero  $[1 + j2\zeta_1\omega/\omega_k + (j\omega/\omega_k)^2]$

This lecture will address creating a Bode diagram with the first 5 types of factors. To create the Bode diagram, plot each of these factors separately and then combine them graphically.

**Standard Form**

Given the following transfer function  $H(s) = \frac{K(s + z_1)}{s(s + p_1)}$  and  $H(j\omega) = \frac{K(j\omega + z_1)}{j\omega(j\omega + p_1)}$

The first step in creating a Bode diagram is to write  $H(j\omega)$  in **standard form**. In order to write  $H(j\omega)$  in standard form, divide out the poles and zeros.

$$\text{STANDARD FORM: } H(j\omega) = \frac{Kz_1 \left(1 + j\omega/z_1\right)}{p_1(j\omega) \left(1 + j\omega/p_1\right)}$$

the amplitude is  $|H(j\omega)| = \frac{K_o \left|1 + j\omega/z_1\right|}{\omega \left|1 + j\omega/p_1\right|}$ , where  $K_o = Kz_1/p_1$

The amplitude of  $H(j\omega)$  in decibels is  **$A_{dB} = 20\log_{10}|H(j\omega)|$**

$$A_{dB} = 20\log_{10} \frac{K_o \left|1 + j\omega/z_1\right|}{\omega \left|1 + j\omega/p_1\right|} = 20\log_{10}K_o + 20\log_{10}|1 + j\omega/z_1| - 20\log_{10}\omega - 20\log_{10}|1 + j\omega/p_1|$$

the phase is  $\theta(\omega) = \alpha_1 - 90^\circ - \beta_1$ , where  $\alpha_1 = \tan^{-1}(\omega/z_1)$ ,  $\beta_1 = \tan^{-1}(\omega/p_1)$

To create the Bode diagram, plot each term of the amplitude separately and then graphically combine the plots.

**i. Constant  $K_o$** 

The plot of the constant,  $K_o$ , is a horizontal straight line at the value  **$20\log_{10}K_o$**

$H(s) = K_o$	$H(\omega)$	$H_{dB}$	$\phi$
	$K_o$	$20 \log_{10}  K_o $	$0^\circ$
	$-K_o$	$20 \log_{10}  K_o $	$\pm 180^\circ$

**ii. Zero at the origin**

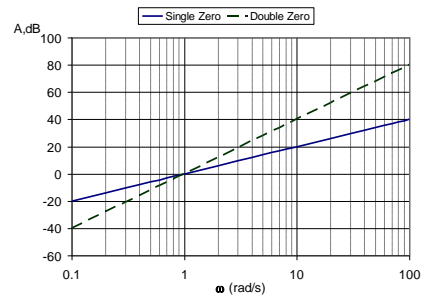
Another possible term in the numerator is a zero at the origin.

The term  $20\log_{10}|j\omega|$  is a zero at the origin. The plot of this term is a straight line with a slope of 20 dB/decade that crosses the 0dB axis at  $\omega = 1$  rad/s. The plot of the phase is a horizontal line at  $90^\circ$ .

$H(s) = s^n$	$H(\omega)$	$H_{dB}$	$\phi$
	$j\omega$	$20 \log_{10} \omega$	$90^\circ$
	$(j\omega)^n$	$20n \log_{10} \omega$	$90n^\circ$

For example, the amplitude plot of a single and double zero at the origin are shown below

$\omega, \text{rad/s}$	$20 \log_{10} \omega$	$40 \log_{10} \omega$
0.01	-40 dB	-80
0.1	-20 dB	-40
1.0	0 dB	0
10.0	20 dB	40
100.0	40 dB	80



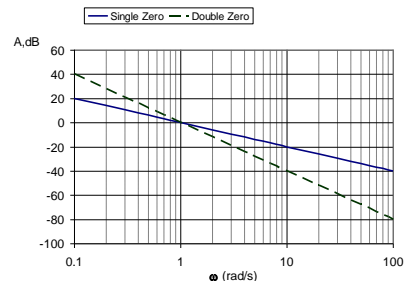
**iii. Pole at the origin**

The term  $20\log_{10}|1/(j\omega)|$  is a pole at the origin. The plot of this term is a straight line with a slope of -20 dB/decade that crosses the 0dB axis at  $\omega = 1$  rad/s. The plot of the phase is a horizontal line at  $-90^\circ$ .

$H(s) = 1/s^n$	$H(\omega)$	$H_{dB}$	$\phi$
	$(j\omega)^{-1}$	$-20 \log_{10} \omega$	$-90^\circ$
	$(j\omega)^{-n}$	$-20n \log_{10} \omega$	$-90n^\circ$

For example, the amplitude plot of a single and double pole at the origin are shown below

$\omega, \text{rad/s}$	$-20 \log_{10} \omega$	$-40 \log_{10} \omega$
0.01	40 dB	80 dB
0.1	20 dB	40 dB
1.0	0 dB	0 dB
10.0	-20 dB	-40 dB
100.0	-40 dB	-80 dB



*In-Class Activity 1:*

*Construct the Bode amplitude and phase plot for the transfer function,  $H(s) = \frac{100}{s}$*



**iv. First Order Zero**

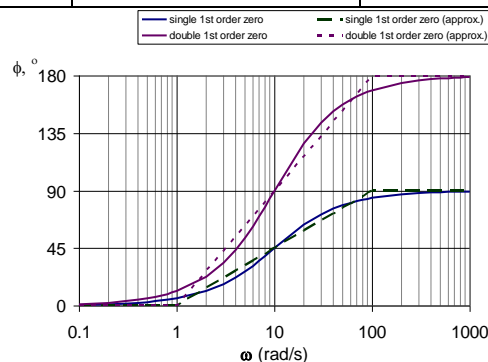
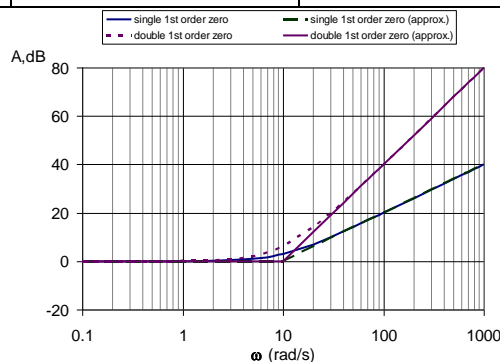
The amplitude term  $20\log_{10}|1 + j\omega/z_1|$  is a first-order zero. The plot of this term is a horizontal line approximation from  $\omega = 0$  to  $z_1$  at  $0^\circ$ . At the **corner frequency** ( $\omega = z_1$ ), the plot increases with a slope of 20dB/decade. The phase term,  $\tan^{-1}(\omega/z_1)$ , is a horizontal line approximation from  $\omega = 0$  to  $0.1z_1$  at  $0^\circ$ . At  $0.1z_1$ , the line increases with a slope of  $45^\circ/\text{decade}$  from  $0.1z_1$  to  $10z_1$ . At  $10z_1$ , the line is a horizontal approximation at  $90^\circ$ .

$H(s) = (s + z_1)$	$H(\omega)$	$H_{dB}$	$\phi$
	$1 + j\omega/z_1$	$20 \log_{10} \sqrt{1 + \left(\frac{\omega}{z_1}\right)^2}$	$\tan^{-1}\left(\frac{\omega}{z_1}\right)$
$(1 + j\omega/z_1)^N$	$20N \log_{10} \sqrt{1 + \left(\frac{\omega}{z_1}\right)^2}$	$N \cdot \tan^{-1}\left(\frac{\omega}{z_1}\right)$	

For example, amplitude and phase plot of a single and double first order zero ( $1 + j\omega/10$ )

$\omega, \text{rad/s}$	$20 \log_{10}  1 + j\omega/10 $	$20 \log_{10}  1 + j\omega/10 $ approximate, dB	$20 \log_{10}  1 + j\omega/10 $	$20 \log_{10}  1 + j\omega/10 $ approximate, dB
0.1	0.0004	0	0.0009	0
1.0	0.0432	0	0.0864	0
<b>10.0</b>	<b>3 dB</b>	<b>0</b>	<b>6 dB</b>	<b>0</b>
100.0	20 dB	20	40 dB	40
1000.0	40 dB	40	80 dB	80

$\omega, \text{rad/s}$	$\tan^{-1}(\omega/10), ^\circ$	$\tan^{-1}(\omega/10), ^\circ$ approximate	$2 \cdot \tan^{-1}(\omega/10), ^\circ$	$2 \cdot \tan^{-1}(\omega/10), ^\circ$ approximate
0.1	0.573	0	1.146	0
1.0	5.711	0	11.421	0
<b>10.0</b>	<b>45</b>	<b>45</b>	<b>90</b>	<b>90</b>
100.0	84.289	90	168.579	180
1000.0	89.427	90	178.854	180







**v. First Order Pole**

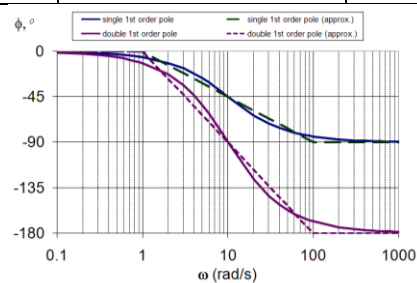
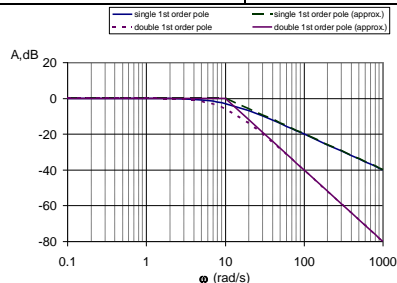
The amplitude term  $-20\log_{10}|1 + j\omega/p_1|$  is a first-order pole. The plot of this amplitude term is a horizontal line approximation from  $\omega = 0$  to  $p_1$  at 0 dB. At the **corner frequency** ( $\omega = p_1$ ), the plot decreases with a slope of -20dB/decade. The phase term,  $-\tan^{-1}(\omega/z_1)$ , is a horizontal line approximation from  $\omega = 0$  to  $0.1z_1$  at  $0^\circ$ . At  $0.1z_1$ , the line decreases with a slope of -45°/decade from  $0.1z_1$  to  $10z_1$ . At  $10z_1$ , the line is a horizontal approximation at  $-90^\circ$ .

$G(s) = 1/(s + p_1)$	$H(\omega)$	$H_{dB}$	$\phi$
	$(1+j\omega/p_1)^{-1}$	$-20 \log_{10} \sqrt{1 + \left(\frac{\omega}{p_1}\right)^2}$	$-\tan^{-1}\left(\frac{\omega}{p_1}\right)$
	$(1+j\omega/p_1)^{-N}$	$-20N \log_{10} \sqrt{1 + \left(\frac{\omega}{p_1}\right)^2}$	$-N \cdot \tan^{-1}\left(\frac{\omega}{p_1}\right)$

For example, amplitude and phase plot of a single and double first order pole  $(1 + j\omega/10)^{-1}$

$\omega, \text{ rad/s}$	$-20 \log_{10}  1 + j\omega/10 $	$-20 \log_{10}  1 + j\omega/10 $ approximate, dB	$-20 \log_{10}  1 + j\omega/10 $	$-20 \log_{10}  1 + j\omega/10 $ approximate, dB
0.1	-0.0004	0	-0.0009	0
1.0	-0.0432	0	-0.0864	0
<b>10.0</b>	<b>-3 dB</b>	<b>0</b>	<b>-6 dB</b>	<b>0</b>
100.0	-20 dB	-20	-40 dB	-40
1000.0	-40 dB	-40	-80 dB	-80

$\omega, \text{ rad/s}$	$-\tan^{-1}(\omega/10), ^\circ$	$-\tan^{-1}(\omega/10), ^\circ$ approximate	$-2 \cdot \tan^{-1}(\omega/10), ^\circ$	$-2 \cdot \tan^{-1}(\omega/10), ^\circ$ approximate
0.1	-0.573	0	-1.146	0
1.0	-5.711	0	-11.421	0
<b>10.0</b>	<b>-45</b>	<b>-45</b>	<b>-90</b>	<b>-90</b>
100.0	-84.289	-90	-168.579	-180
1000.0	-89.427	-90	-178.854	-180



*In-Class Activity 2:*

*Construct the Bode amplitude and phase plot for the transfer function,*

$$H(s) = \frac{1000}{(s + 10)^2 (s + 100)}$$

*In-Class Activity 3:*

*Construct the Bode amplitude and phase plot for the transfer function,  $H(s) = \frac{s(s + 20)}{(s + 2)(s + 200)}$*





















**Lecture 10-1**

**Bode Diagrams**

**Reading:** Chapter 8 of Course Notes

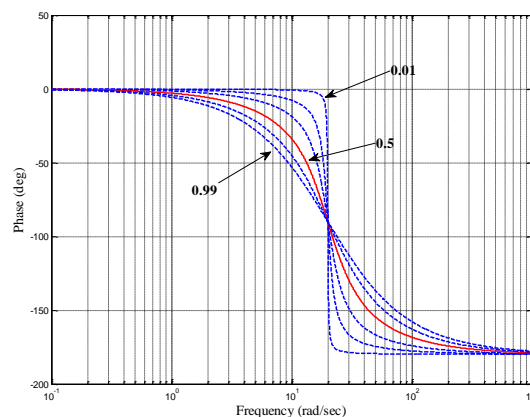
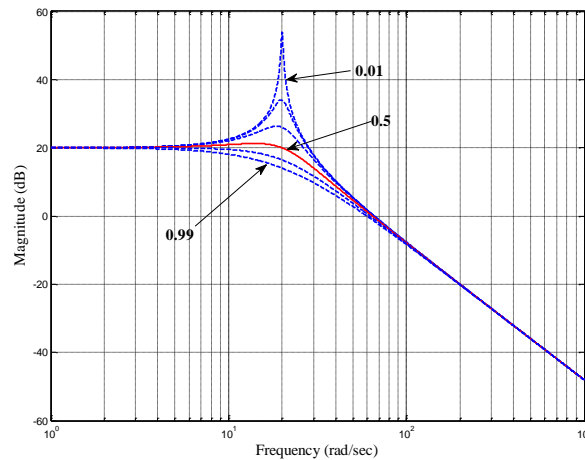
**Objective:** At the conclusion of this lecture, students should be able to

- Determine the steady-state response from a Bode diagram
- Derive the transfer function for a system given the Bode diagram

**vi. Complex conjugate poles and zeros**

The term  $20\log_{10} | \frac{1}{(1 - (\omega/\omega_n)^2)^2 + (2\zeta\omega/\omega_n)^2} |^n$  is a complex conjugate zero or pole at the origin.

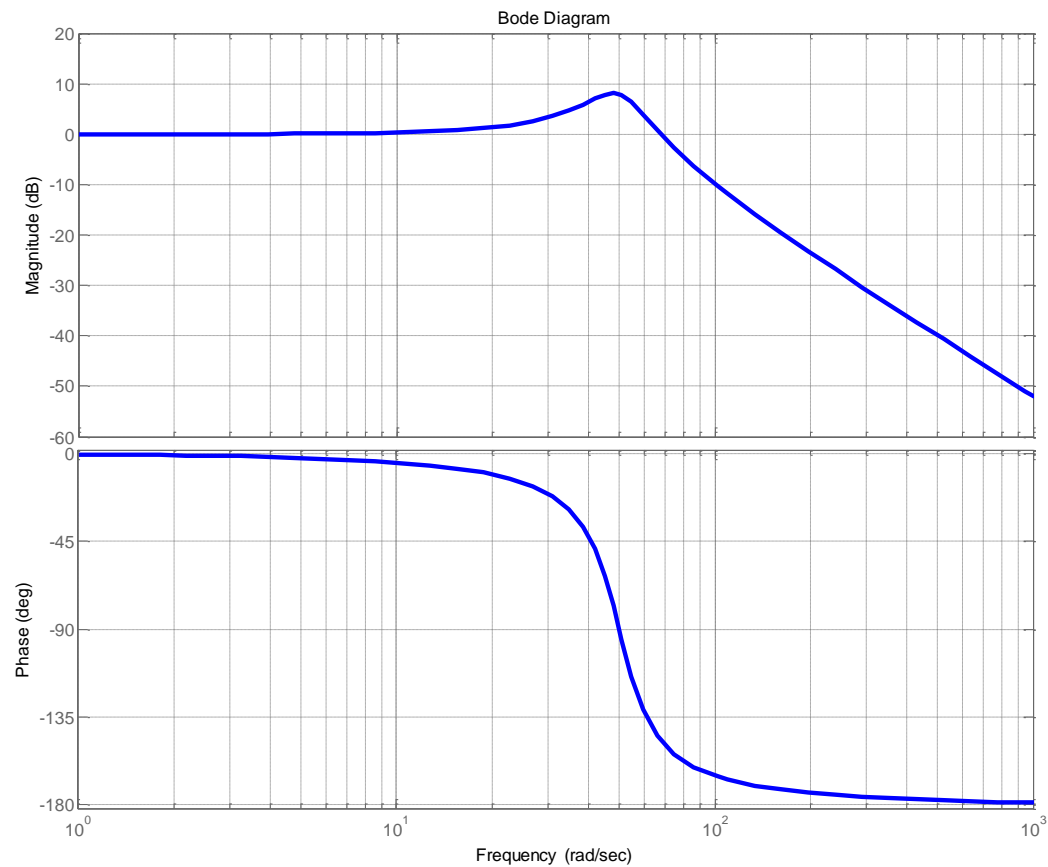
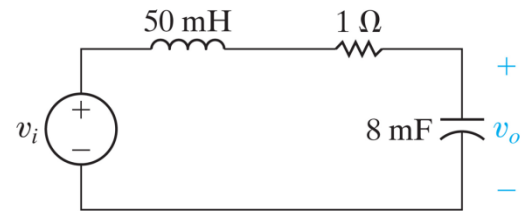
H(s)	H(ω)	Magnitude	Phase
$\frac{K}{(s^2 + 2\zeta\omega_n s + \omega_n^2)^n}$	$\frac{K_o}{\left(\frac{1}{\omega_n^2} + \frac{2\zeta}{\omega_n} s + 1\right)^n}$	$\omega \approx 0$ : 0dB $\omega = \omega_n$ : depends on $\zeta$ $\omega \approx \infty$ : slope = 40n dB/dec	$\omega = \omega_n/10$ : $0^\circ$ $\omega = \omega_n$ : $n45^\circ$ $\omega = 10\omega_n$ : $n180^\circ$



**In-Class Activity 1:**

For the system described by the following circuit, use the Bode plot to predict the amplitude of  $v_o(t)$  in the steady state for the following inputs.

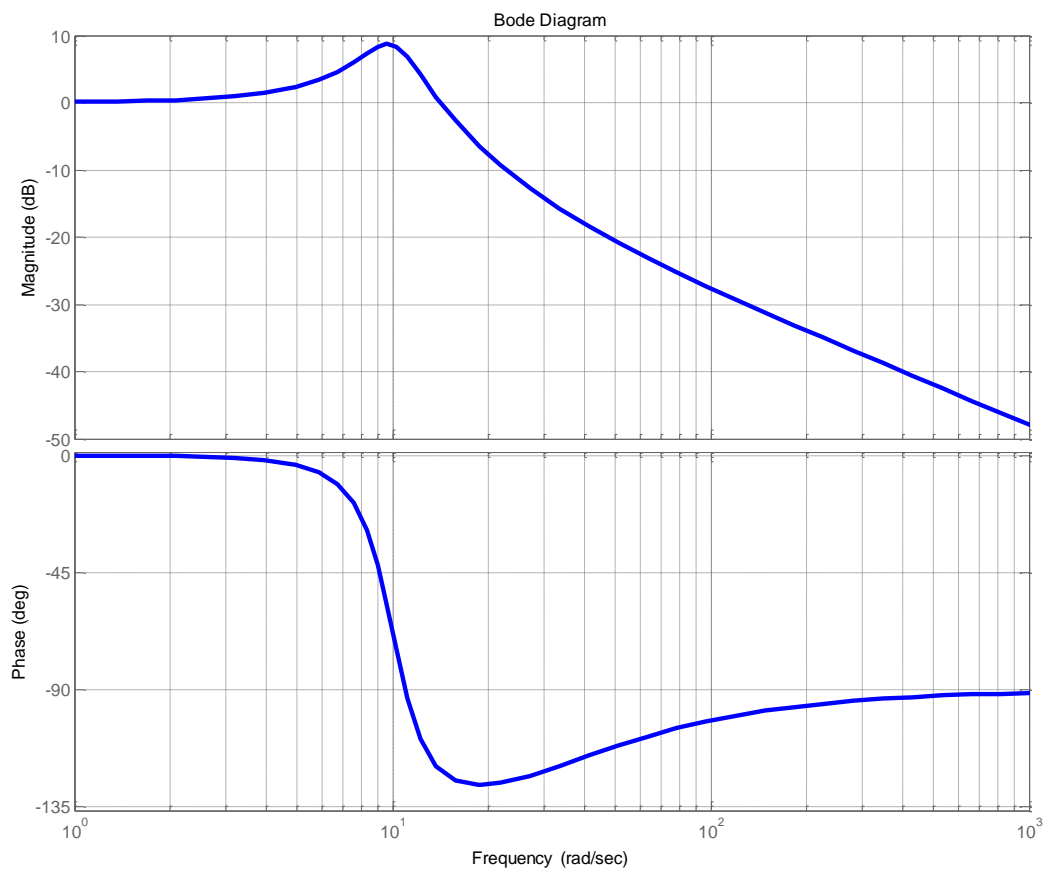
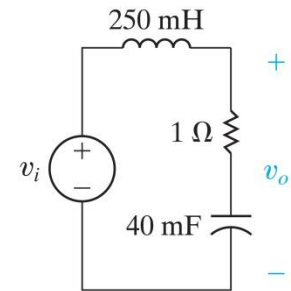
- $v_i(t) = 100 \cos(40t)$  V
- $v_i(t) = 2.5 \cos(70t - 60^\circ)$  V
- $v_i(t) = 65 \cos(300t + 15^\circ)$  V



In-Class Activity 2:

For the system described by the following circuit, use the Bode plot to predict the amplitude of  $v_o(t)$  in the steady state for the following inputs.

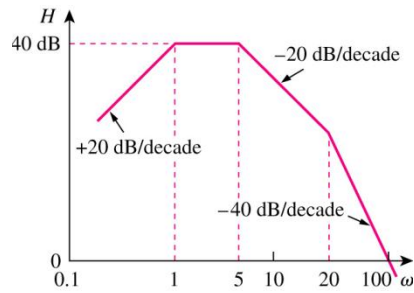
- $v_i(t) = 10 \cos(10t + 60^\circ) \text{ V}$
- $v_i(t) = 5 \cos(2t - 45^\circ) \text{ V}$
- $v_i(t) = 2 \cos(150t + 25^\circ) \text{ V}$



**In-Class Activity 3:**

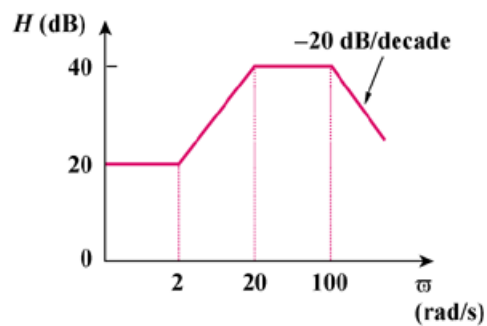
For the following Bode amplitude plot, compute the transfer function,  $H(s)$  in the form

$$\frac{K(s + z_1)(s + z_2) \dots (s + z_n)}{(s + p_1)(s + p_2) \dots (s + p_n)}$$

**In-Class Activity 4:**

For the following Bode amplitude plot, compute the transfer function,  $H(s)$  in the form

$$\frac{K(s + z_1)(s + z_2) \dots (s + z_n)}{(s + p_1)(s + p_2) \dots (s + p_n)}$$





## Lecture 10-2

### Passive Filters

**Reading:** Chapter 8 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define a filter and name the four most common types
- Identify the characteristics of a filter including the gain, bandwidth, center frequency, cutoff frequency and quality factor

The **frequency response** of a circuit is the analysis of the result of varying the source frequency on voltages and currents in the circuit. In the prior lectures, this was examined by creating Bode diagrams. **Frequency-selective circuits** are circuits that pass only certain signals in a desired-range of frequencies to the output of the system. (i.e. radios, telephone systems, car stereos, televisions). Frequency-selective circuits are also called **filters**. Filters weaken or attenuate any input signals outside of particular frequency band.

The action of a filter on an input signal results in an output signal.



The signals passed from the input to the output fall within a band of frequencies called the **passband**. Frequencies not in the circuit's pass band are in the **stopband**. The **cutoff frequency** separates the passband from the stopband. The **cutoff frequency,  $\omega_c$**  is typically defined by engineers as the frequency for which the transfer function magnitude is decreased by **30%** or the factor  $1/\sqrt{2}$  or **-3 dB**. (i.e.  $|H(j\omega_c)| = 0.707 H_{max}$ ).

The system **bandwidth, BW** is defined as the frequency range for which the power of the output is within one half of the peak output power. The frequency where the maximum average power is reduced by half is referred to as the **half-power frequency** or **cutoff frequency**.

$$P_{out}(\omega) = \frac{1}{2} P_{out}^{max}$$

$$10 \log_{10} [P_{out}(\omega)] = 10 \log_{10} \left[ \frac{1}{2} P_{out}^{max} \right] = 10 \log_{10} \left[ \frac{1}{2} \right] + 10 \log_{10} [P_{out}^{max}]$$

$$10 \log_{10} \left[ \frac{1}{2} \right] \approx -3 \text{ dB}$$

$$|H(j\omega)| \geq \frac{1}{\sqrt{2}} |H^{max}(j\omega)|$$

$\omega_{low} \leq \omega \leq \omega_{max}$



All of the filters in this lecture will be **passive filters** that depend only on the passive elements: resistors, capacitors, and inductors. All passive filters have a gain between 0 and 1.

The four major types of filters are:

- **low pass filters** which pass low frequencies and stop high frequencies (i.e. woofer)
- **high pass filters** which pass high frequencies and stop low frequencies (i.e. tweeter)
- **band pass filters** which pass frequencies within a frequency band and block or attenuate frequencies outside the band (i.e. radio)
- **band reject (or notch) filters** which pass frequencies outside a frequency band and block or attenuates frequencies within the band. (i.e. unwanted noise)

Type of Filter	Low - pass	High - pass	Bandpass	Bandreject
$H(0)$	1	0	0	1
$H(\infty)$	0	1	0	1
$H(\omega_c)$ or $H(\omega_0)$	$1/\sqrt{2}$	$1/\sqrt{2}$	1	0
Frequency response  Ideal magnitude plot				
Example Circuit				
$H(s)$	$\frac{\omega_c}{s + \omega_c} = \frac{1/RC}{s + 1/RC}$	$\frac{s}{s + 1/RC} = \frac{s}{s + 1/RC}$	$\frac{(R/L)s}{s^2 + (R/L)s + (1/LC)}$	$\frac{s^2 + (1/LC)}{s^2 + (R/L)s + (1/LC)}$
$\omega_c$	$\frac{1}{RC} = \frac{1}{\tau}$	$\frac{1}{RC} = \frac{1}{\tau}$		
$ H(j\omega) $	$\frac{1/RC}{\sqrt{\omega^2 + (1/RC)^2}}$	$\frac{\omega}{\sqrt{\omega^2 + (1/RC)^2}}$	$\frac{(R/L)\omega}{\sqrt{(1/(LC) - \omega^2)^2 + ((R/L)\omega)^2}}$	$\frac{1/(LC) - \omega^2}{\sqrt{(1/(LC) - \omega^2)^2 + ((R/L)\omega)^2}}$
$\theta(j\omega)$	$-\tan^{-1}(\omega RC)$	$90^\circ - \tan^{-1}(\omega RC)$		
$\omega_0^2$			$1/(LC)$	$1/(LC)$
$\beta$			$R/L$	$R/L$

The **center frequency,  $\omega_0$**  of a bandpass filter is defined as the frequency where the magnitude is **maximum**. The **center frequency,  $\omega_0$**  of a bandreject or notch filter is defined as the frequency where the magnitude is **minimum**. The **center frequency** or **resonant frequency,  $\omega_0$** , is the frequency for which a circuit's transfer function is purely real. This was





referred to as the **natural frequency** when analyzing second-order circuits. The center frequency is the geometric center of the passband  $\omega_o \equiv \sqrt{\omega_{c1}\omega_{c2}}$

The **bandwidth,  $\beta$**  is the width of the passband for a band pass filter. ( $\beta = \omega_{c2} - \omega_{c1}$ ).

The **Quality Factor,  $Q$**  of a bandpass filter is defined as the ratio of the center frequency,  $\omega_o$ , to the bandwidth of the filter. This gives the measure of the width of the filter and high  $Q$  filters ( $> 1/2$ ) are referred to as narrowband filters.

$$Q = \frac{\omega_o}{B}$$

In-class Activity 1:

Given the following transfer function,  $H(j\omega) = \frac{160}{200 + 2j\omega}$

- What type of filter is it?
- What is the cutoff frequency?
- What is the maximum gain?
- What is the bandwidth?

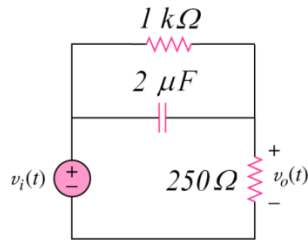
In-class Activity 2:

Given the following transfer function,  $H(s) = \frac{10s}{5s+20}$

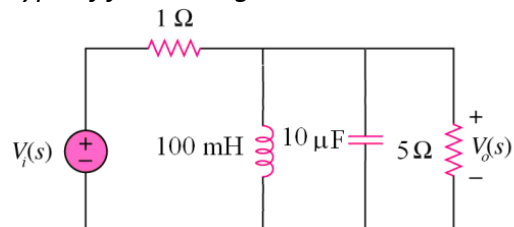
- What type of filter is it?
- What is the cutoff frequency?
- What is the maximum gain?
- What is the bandwidth?

**In-class Activity 3:**

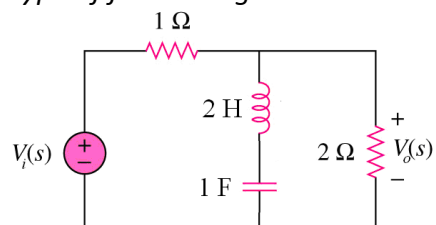
For the system described by the following circuit, use analytical and qualitative techniques to determine the type of filter and characteristics (transfer function, gain, cutoff frequency, bandwidth).

**In-class Activity 4:**

For the system described by the following circuit, use qualitative techniques to determine the type of filter and gain.

**In-class Activity 5:**

For the system described by the following circuit, use qualitative techniques to determine the type of filter and gain.





### Lecture 10-3

#### Active Filters

**Reading:** Chapter 8 of Course Notes

**Objective:** At the conclusion of this lecture, students should be able to

- Define an active filter and describe two advantages
- Identify the characteristics of a filter including the gain, bandwidth, center frequency, cutoff frequency and quality factor
- Compare and contrast active and passive filters.

#### Active Filter Circuits

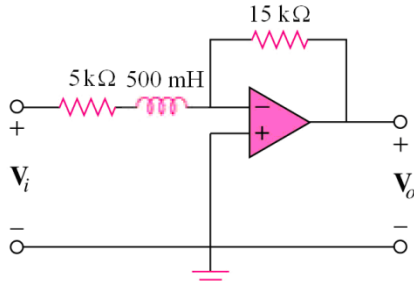
All the filters described so far have been passive; they only include resistors, inductors, and capacitors. **Active filters** also include an operational amplifier. The advantages to using operational amplifiers are that they may be less expensive than inductors and the maximum magnitude can exceed **one**. Additionally, the addition of a resistive load will not alter the passband magnitude or the cutoff frequency of an active filter. However, the disadvantage is they will require additional power for the op amp.

Low Pass Filter	High Pass Filter
$H(s) = -K \frac{\omega_c}{s + \omega_c}$	$H(s) = -K \frac{s}{s + \omega_c}$
$K = R_2/R_1$	$K = R_2/R_1$
$\omega_c = 1/(R_2C)$	$\omega_c = 1/(R_1C)$
integrating amplifier in the time domain	differentiating amplifier in the time domain

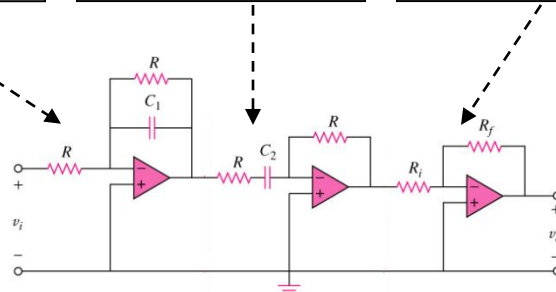
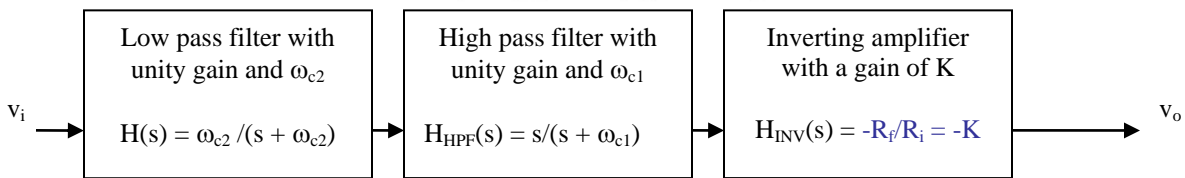
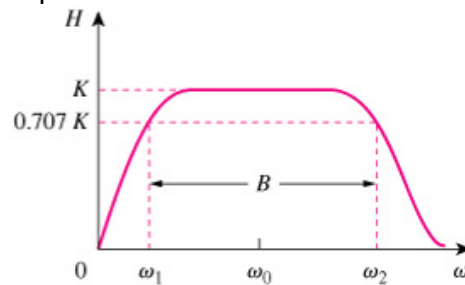


***In-class Activity 1:***

For the system described by the following circuit, use analytical and qualitative techniques to determine the type of filter and characteristics (transfer function, gain, bandwidth, cutoff frequency).



It is possible to create a **broadband** bandpass filter ( $\omega_{c2} \geq 2\omega_{c1}$ ) by cascading a low-pass filter, high-pass filter, and inverting amplifier.



$$H(s) = [H_{LPF}(s)]( H_{HPF}(s))( H_{INV}(s))$$

$$H(s) = -K \frac{\omega_{c2}s}{s^2 + (\omega_{c1} + \omega_{c2})s + \omega_{c1}\omega_{c2}}$$

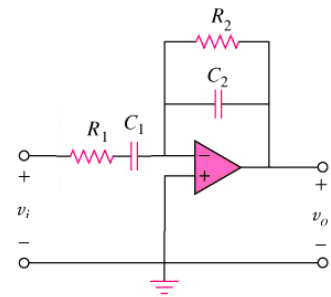
Alternate (Simplified) form of a Bandpass Filter

A passive high-pass filter cascaded with an active high pass filter

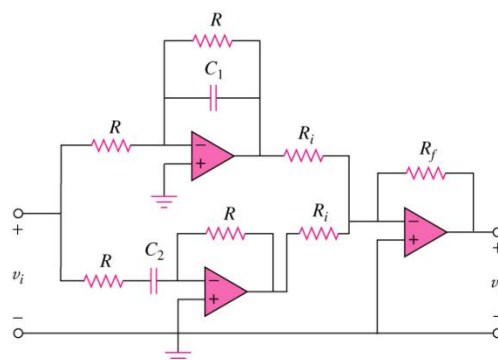
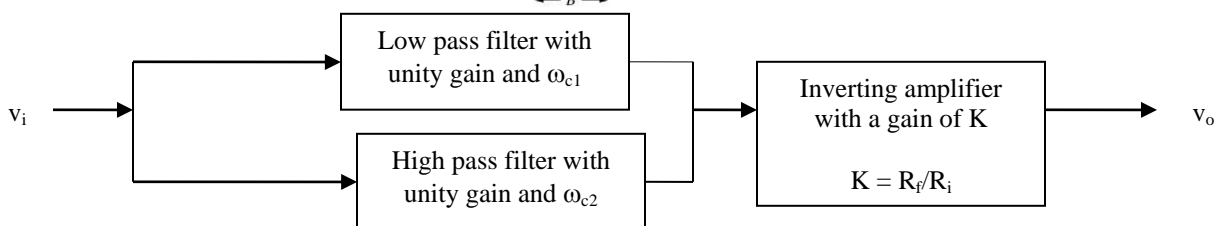
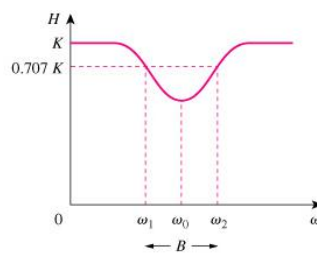
$$K = R_2/R_1$$

$$\omega_{c1} = 1/(R_1C_1) \text{ (cutoff of HPF)}$$

$$\omega_{c2} = 1/(R_2C_2) \text{ (cutoff of LPF)}$$



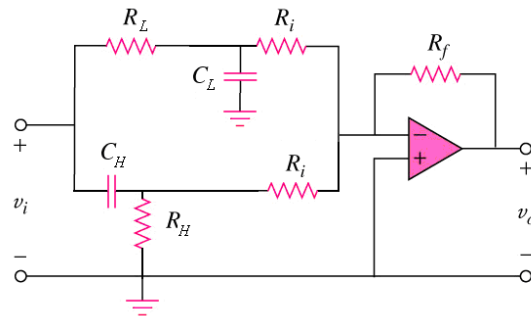
It is possible to create a **broadband** bandreject filter ( $\omega_{c2} \geq 2\omega_{c1}$ ) by putting a low-pass and high-pass filter in parallel and putting them in series with an inverting amplifier.



$$H(s) = K \frac{s^2 + 2\omega_{c1}s + \omega_{c1}\omega_{c2}}{(s + \omega_{c1})(s + \omega_{c2})}$$



Alternate (Simplified) form of a Bandreject Filter



$$(R_H \gg R_i)$$

$$H(s) = K \frac{s^2 + 2\omega_{c1}s + \omega_{c1}\omega_{c2}}{(s + \omega_{c1})(s + \omega_{c2})}$$

In-Class Activity 3:

Design a bandpass filter to have a gain of 20 dB, resonant frequency of 100 rad/s and  $Q = 0.25$ .

In-Class Activity 4:

Design a bandreject filter to have a gain of 10 dB and cut off frequencies at 4 rad/s and 16 rad/s.

ECE 205 Dynamical Systems  
Spring 2010 - 11

---

# LABS

## Dynamical Systems



### Circuits

First Order  
Second Order  
Characteristics  
Frequency-domain  
analysis  
Filter Types

### Signals & Systems

Properties  
Transfer Functions  
Impulse Response  
Convolution  
Interconnected  
Systems  
Bode Plots  
Steady-state response

### Controls

Properties  
Poles and Zeros  
Stability  
Gain and Phase Margins

C. A. Berry



## Lab 1

### Introduction to First Order Circuits using MultiSim, MATLAB and Simulink

**Objective:** In this lab, you will be introduced to some of the computer simulation tools that you will use for your homework, prelab and lab assignments. You will learn to model a first-order system and display measured characteristics.

**Equipment:** Laptop with MATLAB, SIMULINK and MultiSim installed

**Pre-lab:** Read this entire lab theory and procedure thoroughly and complete the following tasks before your laboratory session.

- Install MATLAB on your laptop from the DFS Network Drive
- Purchase the ECE205 lab kit and check out the NI myDAQ from the parts room.
- Install the NI myDAQ software suite onto your laptop. The software is available on the diskette in the box. You should use all of the default settings.
- Review the following link and build your first MultiSim circuit and circuit using the MyDAQ. You must demonstrate the MultiSim circuit working correctly at the beginning of lab to receive full credit for the prelab.  
[http://www.youtube.com/ntspress#p/a/u/0/MZiZ\\_C-ngkY](http://www.youtube.com/ntspress#p/a/u/0/MZiZ_C-ngkY)
- Please see your instructor if you have any questions.

### Theory:

A *first order circuit* contains a resistor and one energy storage element either an inductor or capacitor. It is referred to as a first order circuit because it can be described by a first-order differential equation. There are many first order systems, not just electrical circuits, and they can all be described by first-order differential equations. The standard form for a first order system is

$$\tau \frac{dy}{dt} + y(t) = Kx(t),$$

where  $x(t)$  is the input,  $y(t)$  is the output,  $\tau$  is the time constant, and  $K$  is the static gain. The time constant indicates how long it takes the storage element to store or release energy. The static gain represents the ratio between the output and input after the energy has stopped



changing or under steady-state conditions. This lab procedure will focus on transient analysis as the capacitor or inductor is charging or discharging.

In this lab, you will use MultiSim first to examine the step and natural response of first-order circuits. The *step response* is the response of the circuit after a voltage or current source is suddenly applied and the inductor or capacitor charges to a final value or stores energy. The *natural response* is the response of the circuit to the sudden removal of a voltage or current source and the inductor or capacitor releases energy or discharges.

Next, you will use Simulink to model a first-order system and then use MATLAB to run the Simulink model and create plots of the system response. You will be required to submit some screenshots and plots as part of your memo submission and these are stated in the procedure. Note that you may not finish this lab during the lab session and you are allowed to help each other and get help from me, but what you submit **must be your own**.

**Procedure:**

Part I: MultiSim simulation of a first- order circuit

*Part A: Build the RC circuit in MultiSim*

1. Create a lab 1 course folder on your laptop to store all of your simulation files. Start MultiSim 11.0 and click **Evaluate** unless you have received the license from your instructor and then click **Activate** and follow the prompts to enter the license.
2. Once the design window opens, click **Place-> Component**
3. Under group, change the pull down to *Select all groups*
4. Type *ground* in the **Component** box and place the ground in the window by clicking **OK**
5. The place component window will then open up again and you should type *resistor* in the Component box, select **RESISTOR\_RATED** and click **OK** to place it in the window.
6. The place component window will then open up again and you should type *capacitor* in the Component box, select **CAPACITOR\_RATED** and click **OK** to place it in the window.



7. The place component window will then open up again and you should type *DC\_POWER* in the Component box, select **DC\_POWER** and click **OK** to place the DC voltage source in the window.
8. The place component window will then open up again and you should type *SPDT* in the Component box, select **SPDT** and click **OK** to place the single pole double throw switch in the window. **Close** the Component Window.
9. Double-click the capacitor and change the value to **1  $\mu$ F**. Double-click the resistor and change the value to **100 k $\Omega$** . This yields a time constant for the system of  $\tau = RC = (100 \text{ k})(1 \mu) = 100 \text{ ms}$  and the gain is  $K = 1$ . In order to change the gain, it would be necessary to include more resistors.
10. Double click the DC voltage source and change the value to **5 V**. Next use the mouse to draw lines and connect the components to match Figure 1. Note that you can rotate the components by selecting them and pressing **CTRL-R**.
11. Click place text and put a title in the workspace such as your name, the date, and circuit title. Include a screenshot of this circuit in your lab memo submission.

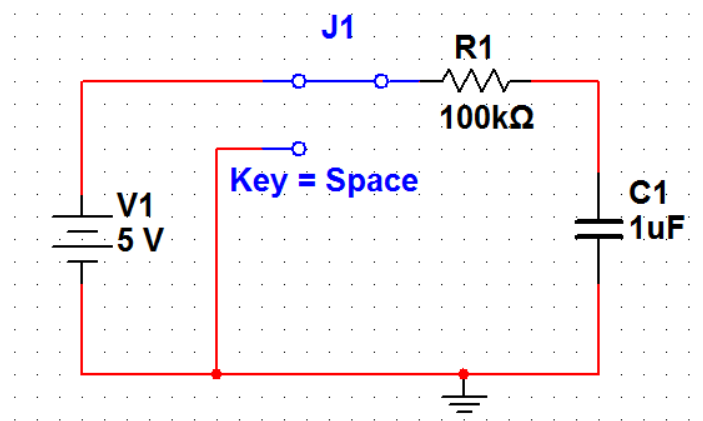


Figure 1: RC Circuit MultiSim Schematic

### Part B: Interactive Simulation

1. Click the Agilent oscilloscope in the right tool bar as shown by the arrow in Figure 2 and place it on the schematic window.



- Double-click on the oscilloscope in order to see an image of how the actual instrument looks.

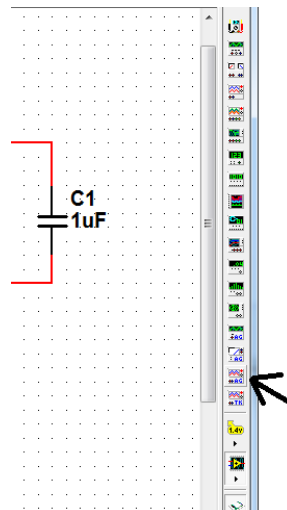


Figure 2: Agilent Oscilloscope

- Push the power button to turn on the oscilloscope. Push the Analog, Channel 1 button to turn on Channel 1. Click the knob above the Channel 1 button until the window shows **2V/div** on channel 1.
- Push the Analog, Channel 2 button to turn on Channel 2. Click the knob above the Channel 2 button until the window shows **2V/div** on channel 2.
- Click the Horizontal knob to **50 ms/div** for both channels. Connect Channel 1 to the DC voltage source and Channel 2 to measure the voltage across the capacitor.
- When you are finished your diagram should look like Figure 3.

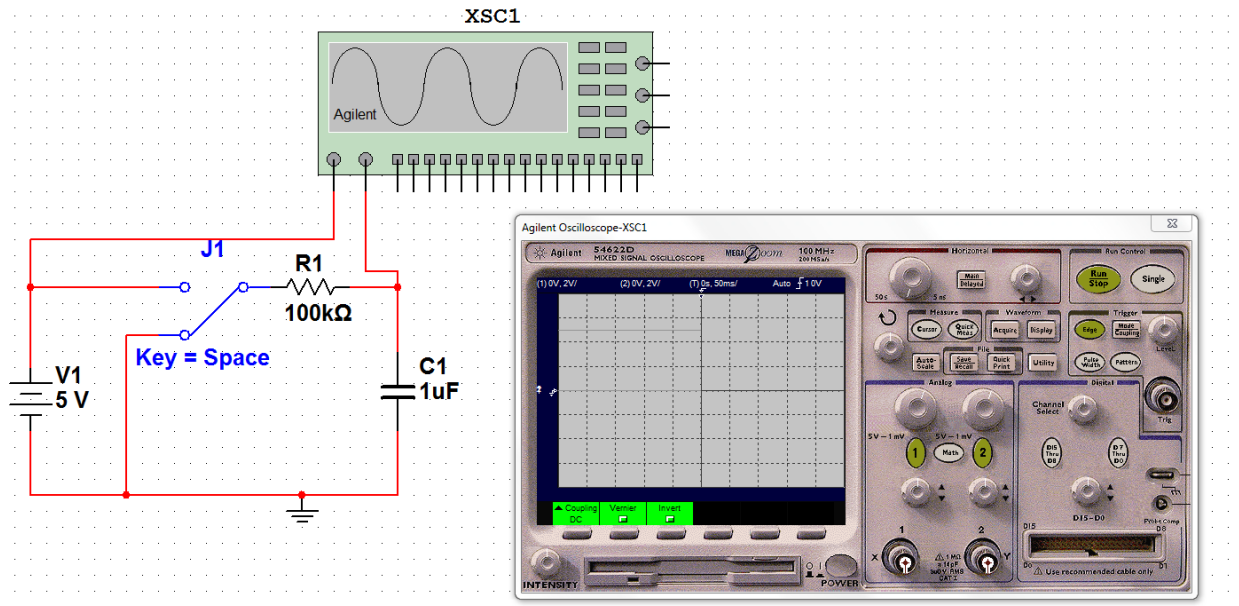


Figure 3: Interactive Simulation Setup

- Click the green **play (run)** button or the **light switch** in the upper right hand corner to turn on the MultiSim interactive simulation (see Figure 4).

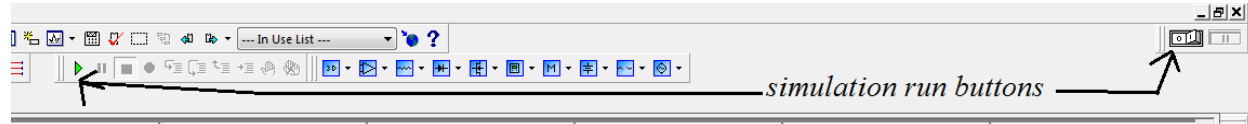


Figure 4: Simulation Run Buttons

- Now push the spacebar on the keyboard multiple times and examine the first order response of the capacitor charging and discharging up to 5V. This exhibits the step and natural response of the RC circuit and it is an exponential function.
- Now you will examine the first order response using a function generator as the input. Delete the 5V source and the switch from the circuit. Go to the toolbar on the right side of the window and add the Agilent function generator. The Agilent function generator is 2 buttons above the Agilent oscilloscope.
- Open up the image, press the **Power** button and press the **Ampl** button and use the knob to set it to **5 Vpp**. Press the **square wave** and press the **Freq** and use the knob to set it to **1 Hz**.

11. Connect the output of the function generator to the input of the circuit to replace the voltage source that was deleted. Channel 1 of the oscilloscope should measure the input also and channel 2 should still measure the voltage across the capacitor. The schematic should now look like Figure 5.

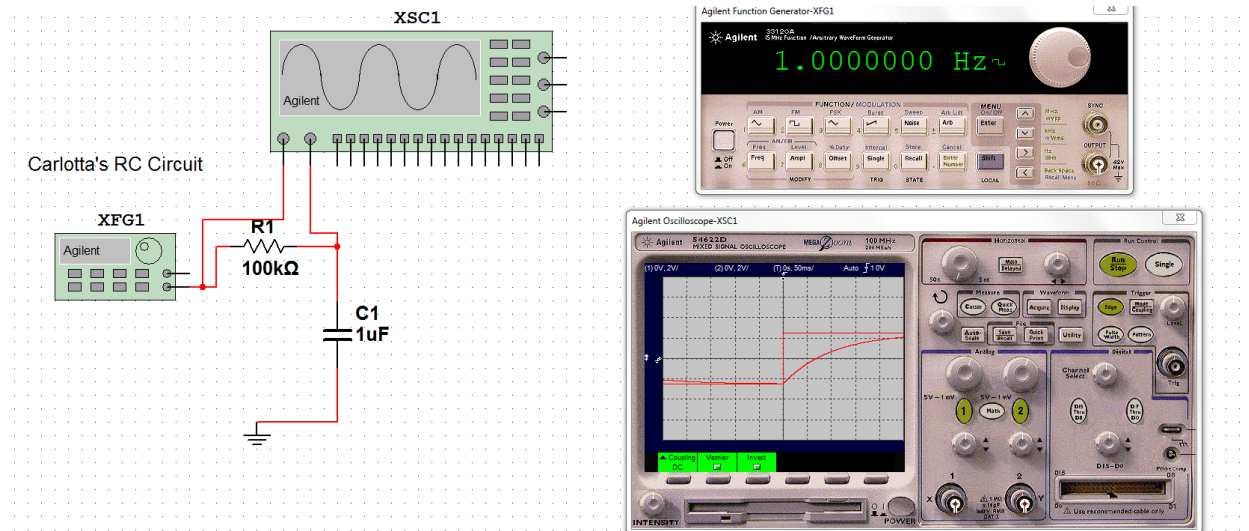


Figure 5: Function Generator and Oscilloscope Setup

12. Turn on the MultiSim simulation by flipping the switch in the upper right hand corner or press the green button. Once again the oscilloscope should show the capacitor charging and discharging.
13. Include a screenshot of the oscilloscope in your lab memo submission with an appropriate number and caption.

### Part C: Transient Analysis

1. In this section, you will setup a transient analysis to examine the first-order response of the system. A transient analysis is the plot of the circuit response as a function of time.
2. Delete the function generator and oscilloscope from the circuit schematic. Change the resistor to  $1\text{ k}\Omega$  and the capacitor to  $0.1\ \mu\text{F}$ . This makes the time constant for the system,  $\tau = RC = (1\text{ k})(0.1\ \mu) = 100\ \mu\text{s}$  and the gain is still  $K = 1$ .
3. Place the component, **PIECEWISE\_LINEAR\_VOLTAGE** on the schematic and connect it to the RC circuit as the input source.

- Double-click the piecewise linear voltage source and note that you can either enter data points from a file or enter them on a table. For this analysis, you will use the table. Change the input to match Table 1.

Table 1:  $x(t)$  piecewise linear function

Time	Voltage
0	0
0	2
0.0001	2
0.0001	-3
0.00025	-3
0.00025	4

- Close the piecewise linear voltage source data table. Double click on the wire above the capacitor and name the node **Vcap** and **check show net name**. Your circuit should now look like Figure 6.

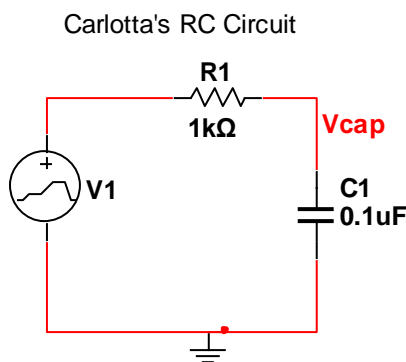


Figure 6: RC circuit with PWL source

- To set up the transient analysis, click **Simulate->Analyses->Transient Analysis** and the set up window will open up. Leave the start time as zero and set the final time to **0.0008** seconds
- Click the output tab, select **V(vcap)** and click **Add**. Then, click **V(1)** and click **Add**. Lastly, click **Simulate** and the transient analysis plot should open up. Click on the black and white square under the Cursor menu heading and the background of the plot should



change to white and be similar to Figure 7. Please include this plot in your lab memo submission.

- You are finished with this part and can close MultiSim. Finally, attach your final Simulink file (**.ms11**) to your submission.

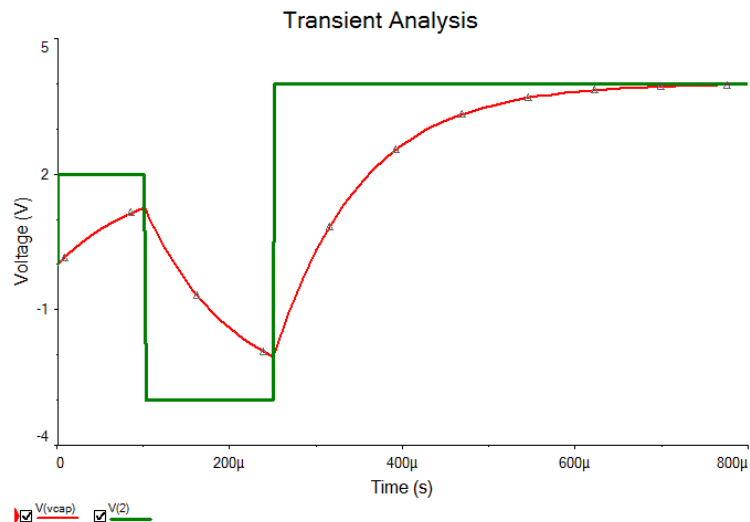


Figure 7: RC Circuit Transient Analysis

For more practice with MultiSim review the following links:

<http://cnx.org/content/col10369/latest/>

[http://www.youtube.com/ntspress#p/a/u/0/MZiZ\\_C-ngkY](http://www.youtube.com/ntspress#p/a/u/0/MZiZ_C-ngkY)

<http://inst.eecs.berkeley.edu/~ee100/su06/handouts/EE100-MultiSim-Tutorial.pdf>

<http://digital.ni.com/manuals.nsf/websearch/193FBEEDC810E62C86257260006545AE>

<http://digital.ni.com/manuals.nsf/websearch/DB7B3DC28D422B3E862572600067215C>

#### Part II: Build the Simulink Model using XY graph

- In this part you will simulate first order differential equations by using Simulink and MATLAB. The first step is to solve the standard form of the first order differential equation for the highest derivative power of the output,  $y(t)$  as shown here,

$$\frac{dy}{dt} = \frac{1}{\tau} [Kx(t) - y(t)]$$

- Since the integral of the above equation yields  $y(t)$ , this will be used to create the simulation model by using a scaling (gain), summing, and integration blocks.



3. Start MATLAB and when it finishes initializing it will show **Ready** near the **Start** button. Change the directory in the drop down to the folder created in Part I.
4. Type **simulink** at the command prompt and hit **enter**. The Simulink Library Browser in Figure 8 will open up.

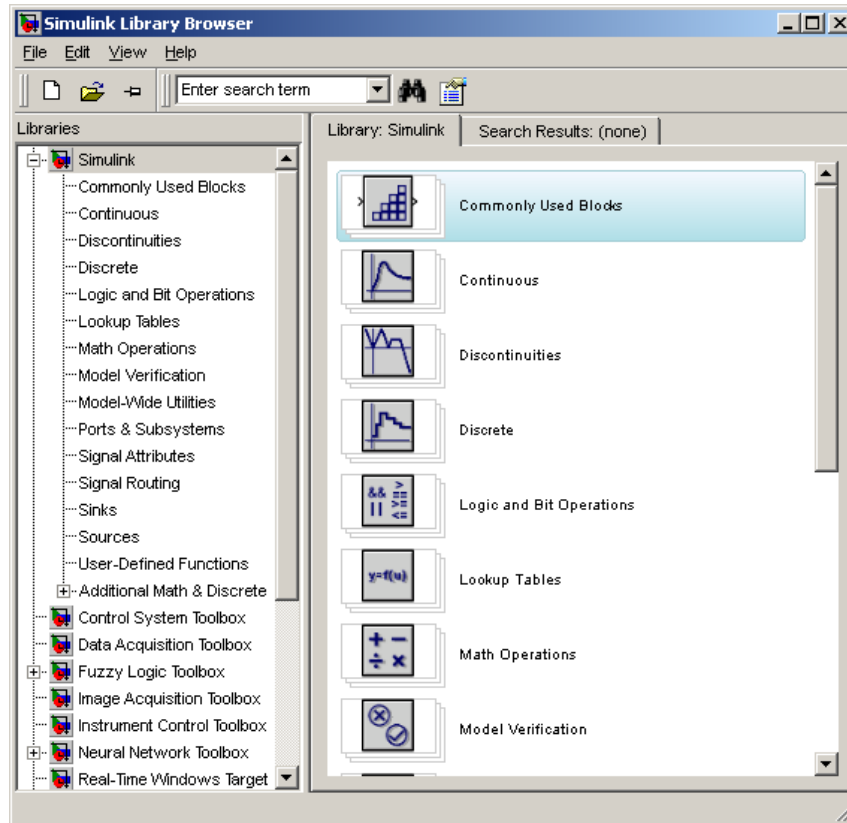


Figure 8: Simulink Library Browser

5. This window gives you access to all of the blocks you will use to make the first-order model.
6. Click **File->New->Model** and this will open the blank window where you will create the SIMULINK model. Click **File->Save As** and name the model **Lab1.mdl**.
7. Assume that  $\tau = 0.02$  seconds,  $K = 0.5$ , and the input is a step with amplitude,  $A = 0.5$  and let's build the model.
8. In the **Simulink Library Browser**, scroll down until you see a block named **Sources** as shown in the figure on the next page, and then double click on it. This is where we will find the available system inputs.
9. Drag **Clock** over to your Simulink model file, *Lab1.mdl*, and then scroll down to find **Step** and drag this over to your model file.





10. Next we are going to need a sink in order to save or plot the results. In the left panel, click on **Sinks** and then drag the **XY graph** over to Lab1.mdl. Make sure to save the SIMULINK model often so that you not lose your work.
11. We are going to need an integrator, so click on **Commonly Used Blocks** in the left panel, and drag an **Integrator** (1/s) over to your model file, drag a **Sum** over to your model file, and then drag a **Gain** over to your model file.
12. Rearrange the blocks in the general order you will use them, as shown in Figure 9. At this point you probably realize you will need another **Gain** block. You can go back to the Library Browser, or just right click on the Gain block you have and choose copy and then paste.

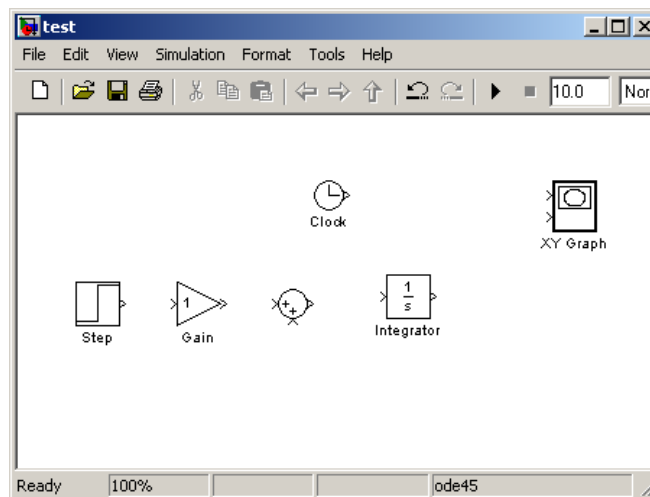


Figure 9: Simulink Model Blocks

13. Now we are ready to connect the blocks. This takes some practice, but you generally click on the source or destination block and drag the line. Connect the blocks so your model now looks like the Figure 10. To get the bottom line into the summer, make a line going down from the sum block, let go, then make a separate line from your first line to the line after the integrator.

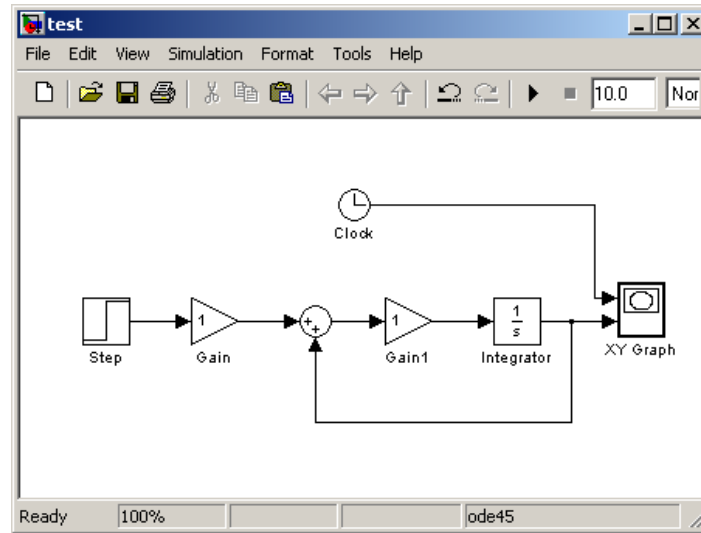


Figure 10: Simulink First Order System Model

14. Double click on the **Step** block, and set the **Step Time** to **0** and the **Final Value** (the amplitude of the step) to **0.5** (the value of A). Then click **OK**.
15. Double click on the **Gain** block next to the Step block, and set the gain equal to **0.5** (the value of K), and then click **OK**.
16. Double Click on the **Sum** block. We will change the parameters, as shown in Figure 11. Let's choose a rectangular sum block (this works better if there are more inputs), and change the bottom sign to -. Also, we will put in extra space by adding some space, denoted by |. Finally, click **OK**.

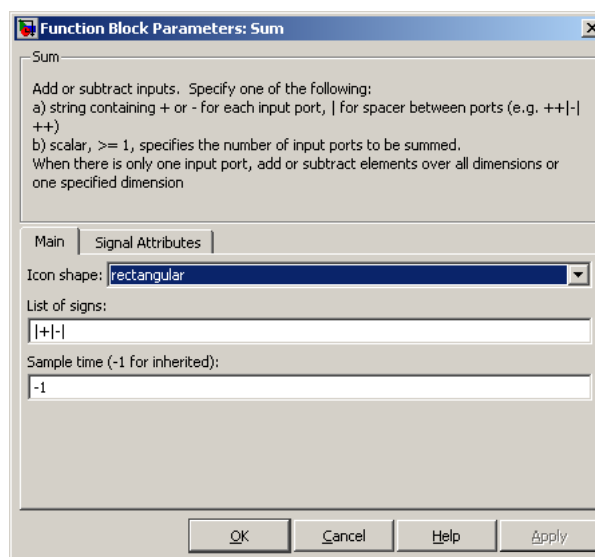


Figure 11: Function Block Parameters for the Summer

17. The **Gain** block represents  $1/\tau$  so double click on it and enter **50** and then click **OK**.
18. We want to view the output on the **XY Graph** so double click on the **XY Graph** and enter  $x\text{-min} = 0$ ,  $x\text{-max} = 0.3$ ,  $y\text{-min} = 0$ ,  $y\text{-max} = 0.3$ , then click OK.
19. Since, the output will only plot from 0 to 0.3 seconds, change the final time of the simulation to 0.3 seconds as shown in Figure 12.

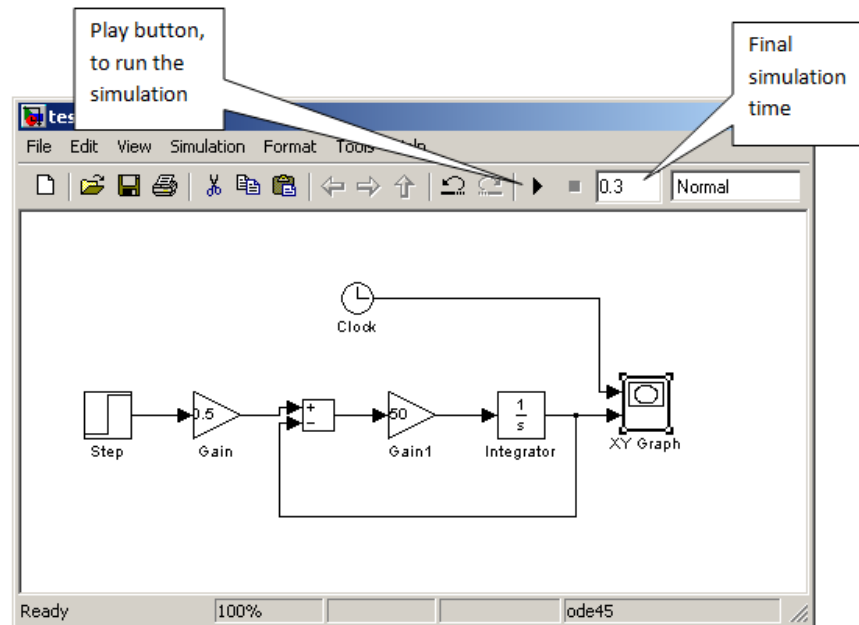


Figure 12: Finally Simulation Model

20. Finally, run the simulation by pressing the play button shown in Figure 12 and the step response graph will be shown in Figure 13.

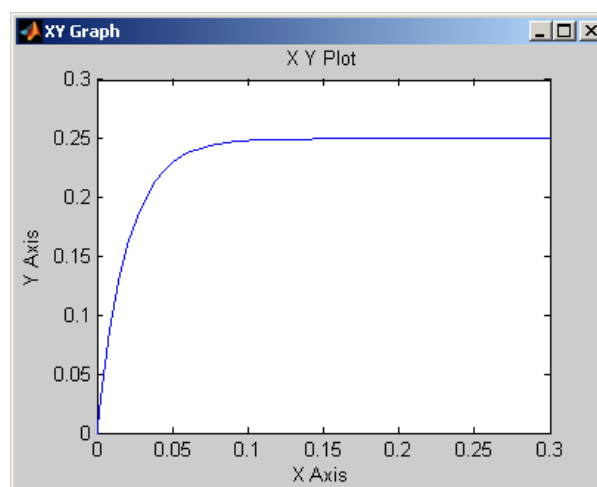


Figure 13: First-order Step Response

Part III: Use MATLAB to plot a step response

- 1) While the **XY Graph** is nice for getting a basic idea of what is going on, you have little control over how the graph is presented. You also may need to know a lot about the signal you want plotted. You can use MATLAB to create a graph with more capabilities.
- 2) Go back to the **Simulink Library Browser**, click on **Sinks**, and then drag two **simout (To Workspace)** blocks over to your simulating file (*Lab1.mdl*).
- 3) Right click on **XY graph** to delete it and then click on data paths to delete them also.
- 4) Connect the simulation output (y) and the **Clock** to the two different **simout** blocks, as shown in Figure 14.

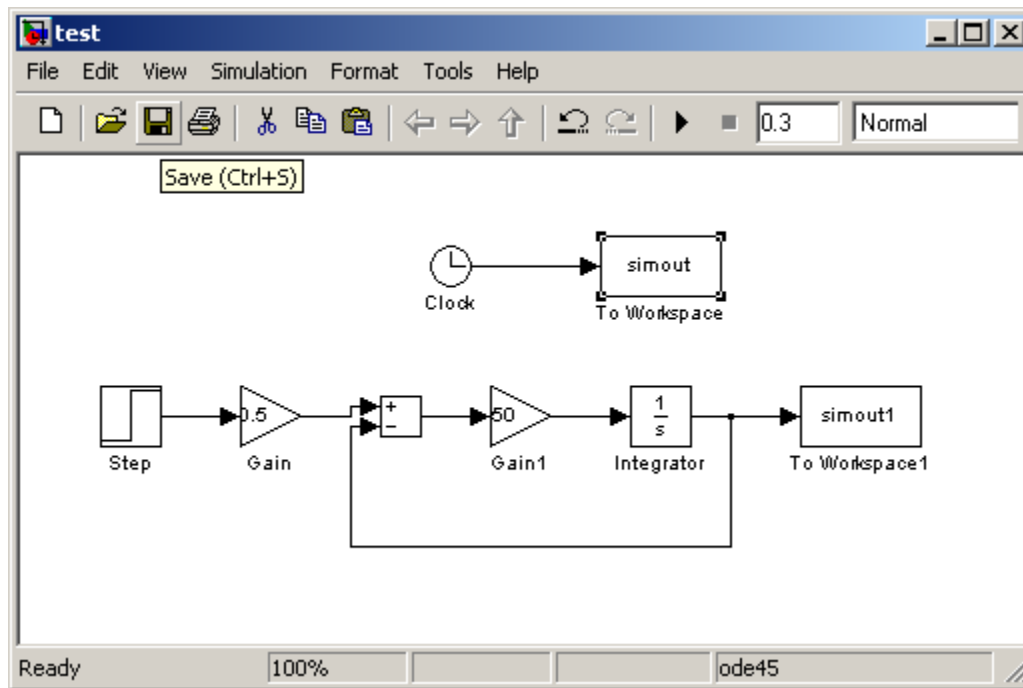


Figure 14: First-order Model with simout

- 5) Double click on the **simout** block connected to the **clock**. Set the variable name to **time** and the Save format to **Array**.
- 6) Double click on the **simout** block connected the **integrator**. Set the variable name to **y** and the Save format to Array. Your simulation file should now look like Figure 15.

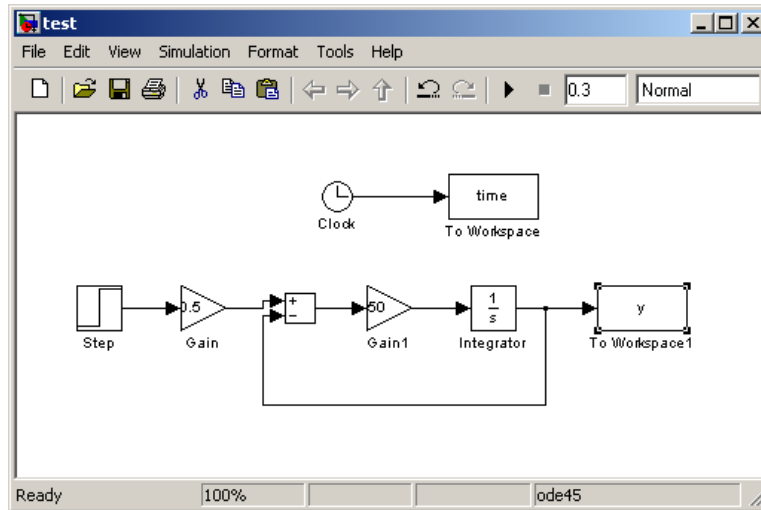


Figure 15: Final First-order Model

- 7) Run the simulation by pushing the play button. It may not appear that much has happened, (you may get some warning messages, but ignore them for now) but the simulation has saved the data to your workspace.
- 8) Type **who** in the MATLAB command window, and you should see that both the **time** and **y** (and maybe some other variables) are now in the workspace.
- 9) To plot the data, type the following lines at the MATLAB command prompt:

```
>> plot(time,y, 'Linewidth', 3);grid;
>> xlabel('time (sec)');ylabel('y(t)');title('First Order
System');;
```

A figure similar to the one shown in Figure 16 should open up.

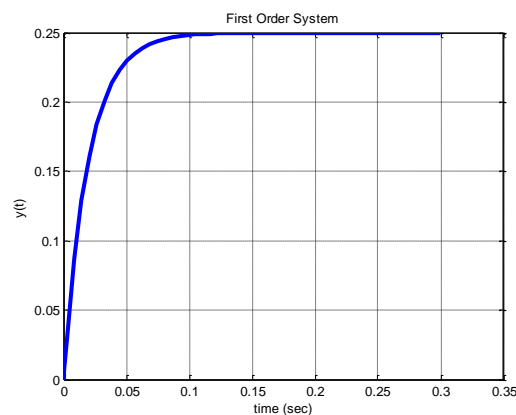


Figure 16: First-Order Step Response



### Part IV: Creating a MATLAB script file (m-file) to run simulations

1. Just as in general programming it is a bad habit to “hard code” a variable, the same is true with Simulink. For example, we would like to be able to change the parameters in our simulation, run the simulation, and plot the results in a very convenient way. This is what we will accomplish in this level.
2. In the **MATLAB** command window, select **File**→**New**→**Script** **Ctrl+N**. As an alternative, you can click on the leftmost icon just under File (it looks like a page with a red dot in the upper left corner). Yet another new window will open up!
3. In this new window, select **File**→**Save As**, and then name it **Lab1\_driver.m**.
4. In this new file, type the following lines:

```

%%%%%%%%%
%% Lab1_driver.m
%% Geordi LaForge
%% February 22, 2222
%% this program runs the file Lab1.mdl to model first-order systems
%%
%% clear all variables
clear variables;
%% close all figure windows
close all;
%% simulate the model
sim('Lab1.mdl');

```

5. Any line in MATLAB that begins with a % is a comment line. Note that it is a good practice to always include a header at the top of the m-file with the name of the file, your name, the date it was created and a short description. In addition, it is a good practice to comment your code so that it is easy to debug and easy for a reader to follow. This standard will be expected on all future m-file submissions.
6. The last line, **sim('Lab1.mdl');** actually runs the Simulink file.
7. Save this file and then run it, as shown in Figure 17.

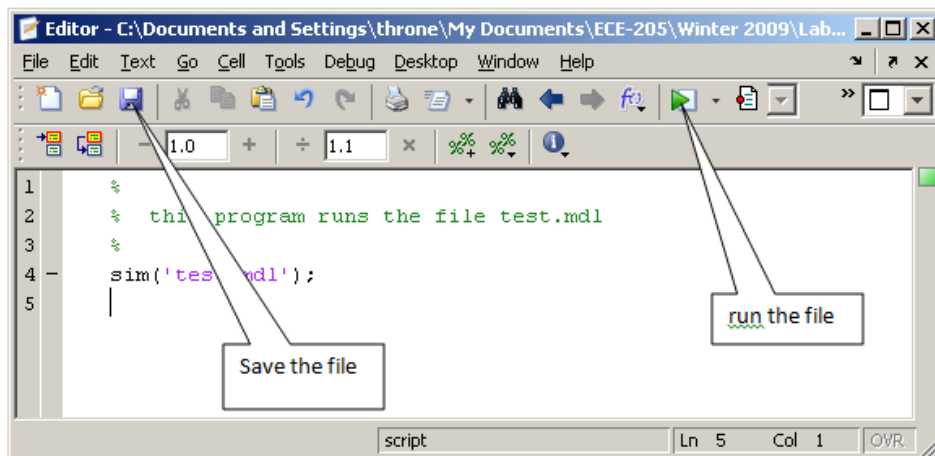


Figure 17: Save and Run an m file

8. To see that this is working, in the MATLAB workspace type **clear**, then **who**. There should be no variables in the workspace.
9. Then run m-file **Lab1\_driver.m** and type **who** in the workspace, and you should see variables have been generated.
10. We would also like to automatically generate a plot every time we run the MATLAB program, so enter the following lines into your MATLAB program (after the **sim** command), so that the data is in the workspace.

```

%% plot the data
plot(time,y,'-', 'Linewidth',3);grid;
xlabel('time(sec)');ylabel('y(t)');
title('Lab 1 - First Order System');

```

11. You should **run (play)** the program again, and it should generate a graph (figure) like before. You may want to remove any figures before you run it to be sure it generated a new figure.
12. It may also be useful for use to plot both the output,  $y(t)$  and the steady state value. To do this, first modify your Simulink file **Lab1.mdl** to look like Figure 18. Make sure to double-click on the Kx block and make the save format **Array**. Note that you also may need to select an entire group of elements in Simulink to shift your elements around.

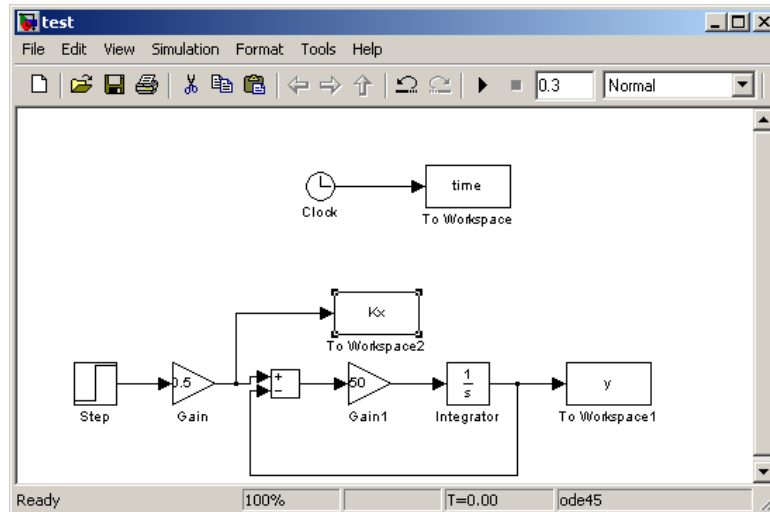


Figure 18: First-order Simulink model with 3 simout variables

13. Modify the plotting commands in your MATLAB file, **Lab1\_driver.m** as follows:

```
plot(time,y,'-',time,Kx,'o','Linewidth',2);grid;
legend('y(t)', 'y_{ss}');
xlabel('time(sec)');ylabel('y(t)');
title('Lab 1 - First Order System');
axis([0 0.3 0 0.3]);
```

14. MATLAB defaults to finding a convenient set of axes, but sometimes you want to tell MATLAB more specifically the axes you want. If you type **help plot** in the MATLAB command window you can find a number of options for plotting. If you run your code now, you should get Figure 19. Note that you can drag the Legend around and put it where you want.



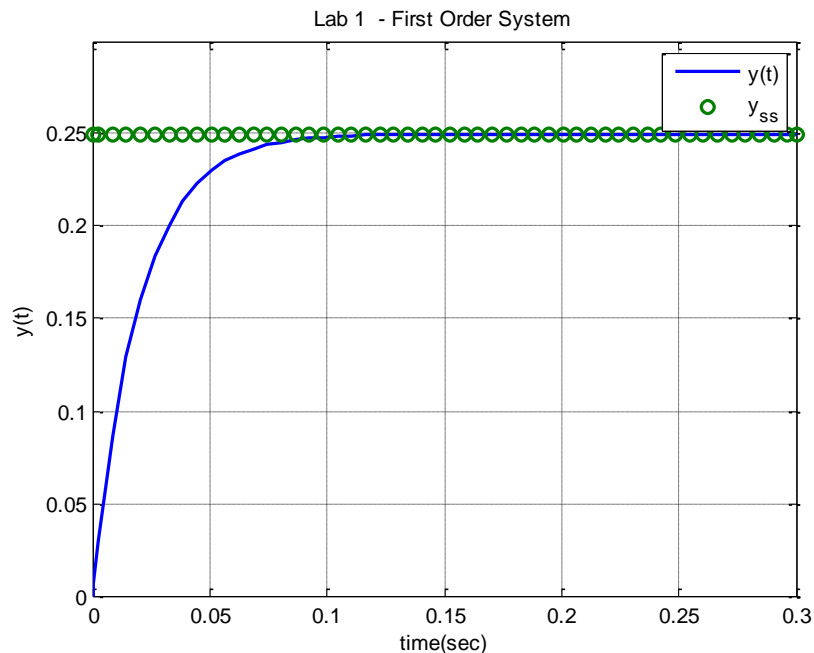


Figure 19: Step Response with input and output

15. Finally, we would like all of the parameters we have in our Simulink model file to be controlled by the MATLAB program. Modify the MATLAB program **Lab1\_driver.m** and put the following lines before the simulation command (**sim**)

```
%% set the first-order system characteristics  
  
tau = 0.02;  
  
K = 0.5;  
  
A = 0.5;  
  
Tf = 0.3;
```

16. Next, edit the Simulink file, and use these variable names instead of the values you entered previously. The final value of the **Step** should now be **A**, the gain of the first **Gain** block should be **K**, and the gain of the second **Gain** block should be **1/tau**.
17. The final time of the simulation should be changed to **Tf**. Delete any figures and run your MATLAB driver file again. You should get the same figure as before (see Figure 17). You should include this figure in your lab memo submission with your initials included in the title. To include this graph, select **Edit** and then **Copy Figure**. You can then paste the figure

in your memo. Do not include your figure using any other method! Be sure the figure is large enough and that I can see it and that it has a figure number and caption.

Part V: Generating an input in MATLAB

1. As long as we are putting all of the variables in our MATLAB file (program), we might as well figure out how we can use this to generate an input.
2. Go to the **Simulink Library Browser**, click on **Sources**, and drag a **simin (From Workspace)** block over to your Simulink model file. Delete the **Step** block and connect the **simin** as the input.
3. Double click on the **simin** block and name the data **xt**. This is where Simulink will look for its input data. Note that you will need to produce both **data values ( )** and **time values (t)** and put them into the variable **xt**. One method to do this is presented and Simulink will interpolate between the data values as needed.
4. To create the data, we will use **MATLAB's** anonymous function to generate the signal. To do this type the following lines in your **MATLAB** code before the **sim** command and after you have defined **Tf**:

```
%% generate the input
x = @(t) 1*((t>=0)&(t<0.1)) -1*(t>=0.1);
t = linspace(0,Tf,300);
xt = [t' x(t)'];
```

5. The first piece of code after the comment is the definition of  $x(t)$ . The second line of code tells MATLAB to generate an array of time values for 0 to  $T_f$ , and use 300 evenly spaced points. The third line of code puts the data in the correct form for Simulink. The apostrophe means to take the transpose, the brackets  $[]$  mean form an array, and  $x(t)$  tells MATLAB to evaluate the function  $x$  at the specified times  $t$ . (Note that you might feel that it would be easier to use MATLAB's Heaviside function here, but sometimes Simulink complains, since MATLAB defined the Heaviside function evaluated at zero to be not a number (NAN).) Comment out the axis command and run your code, you should get a graph like Figure 20.

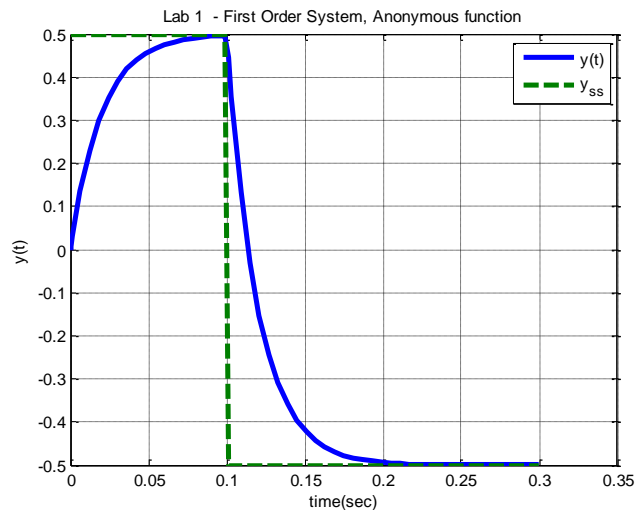


Figure 20: Step Response with an Anonymous Function

6. Now let's try something more complicated. In the notes we have an example for a first order system with parameters,  $\tau = 0.0001$  s,  $K = 0.01$ , and  $y(0) = 0.01$ . Change the time constant and static gain parameters in your MATLAB script file to match these. This is similar to what you did in Part IV of the procedure.
7. set the simulation final time to **Tf = 0.0008** seconds. Make a new variable called **y0**, and set **y0 = 0.01**;. You will no longer use the A variable so you can keep the value the same as before.
8. Click on the **integrator** in your Simulink file and set the Initial condition to **y0**.
9. The system input is

$$x(t) = \begin{cases} 0 & t < 0 \\ 2 & 0 \leq t < 0.0001 \\ -3 & 0.0001 \leq t < 0.00025 \\ 4 & t \geq 0.00025 \end{cases}$$

10. To enter the input, change the definition of the function as follows:

$$x = @(t) 0*(t<0)+2*((t>=0) & (t<0.0001)) \dots \\ -3*((t>=0.0001) & (t<0.00025))+4*(t>=0.00025);$$

11. Note that to continue a MATLAB function or statement on the next line, you end the line with three dots (...)

12. Comment out the axis command, and run the code. If you have done everything correctly, you should get the graph in Figure 21. Include this graph in your lab memo submission.

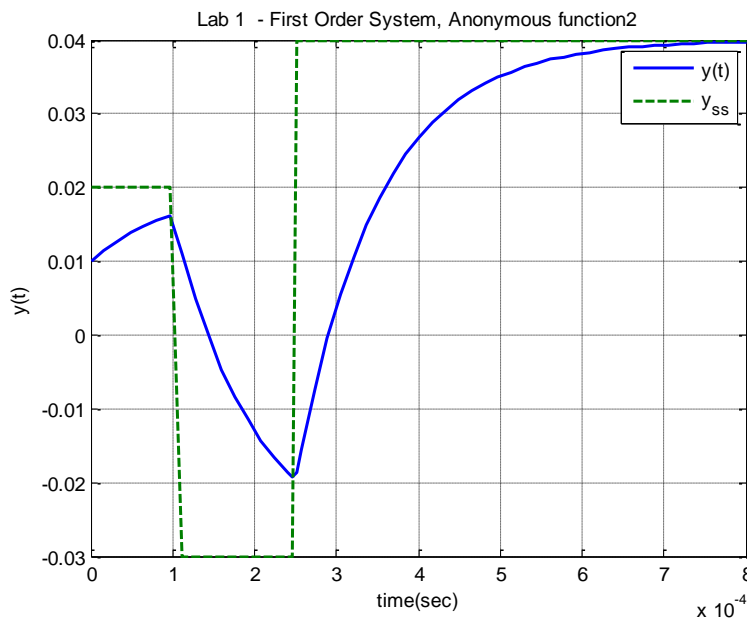


Figure 21: Step Response with an Anonymous Function

Part VI: Creating an analytical function to check your work

1. Finally, when you are doing your homework, you are going to have problems like these, and are going to need a method to check your answers.
2. Anonymous functions and MATLAB will be used to check the answer if the solution is

$$y(t) = \begin{cases} 0 & t < 0 \\ -0.01e^{-t/0.0001} + 0.02 & 0 \leq t < 0.0001 \\ 0.04632e^{-(t-0.0001)/0.0001} - 0.03 & 0.0001 \leq t < 0.00025 \\ -0.05966e^{-(t-0.00025)/0.0001} + 0.04 & t \geq 0.00025 \end{cases}$$

3. In MATLAB, type in the following anonymous function just below your definition of your anonymous definition for x,

```
ya = @(t) 0*(t<0)+(-0.01*exp(-
t/0.0001)+0.02).*((t>=0)&(t<0.0001)) ...
+(0.04632*exp(-(t-0.0001)/0.0001)-
0.03).*((t>=0.0001)&(t<0.00025)) ...
```



```
+ (-0.05966*exp(-(t-0.00025)/0.0001)+0.04) .* (t >=0.00025);
```

4. Next change your plotting and legend commands to the following:

```
plot(time,y,'b-',time,Kx,'r--
',time,ya(time),'mo','Linewidth',s);
legend('y(t)','y_{ss}','y_{analytical}');
```

5. If everything worked correctly, you should get the plot in Figure 22, which shows the analytical solution matches the simulated solution. This graph should be included in your lab memo submission.

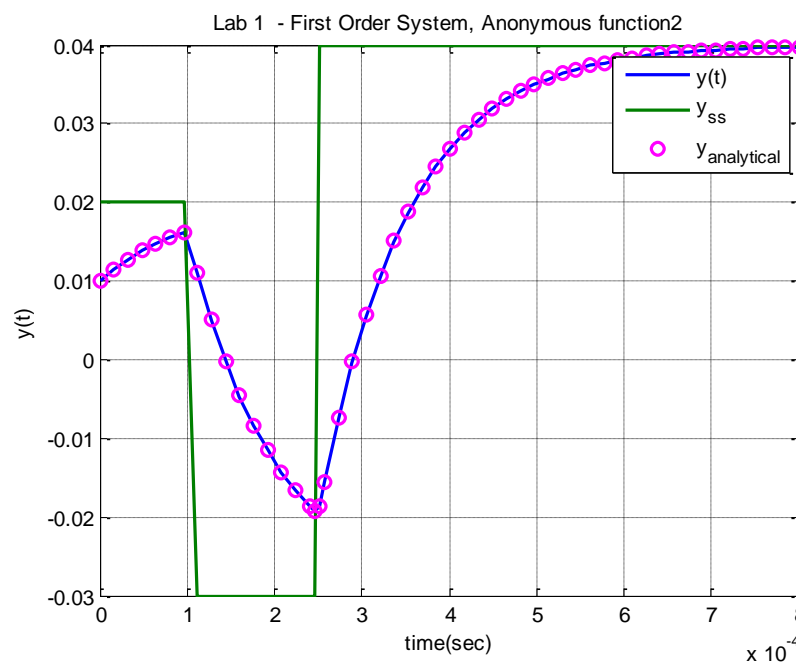


Figure 22: Analytical and Simulate Solution of First Order System

6. At this point, you have completed your lab procedure and should include all of the required figures. There should be six figures in all, 3 from the MultiSim analysis and 3 from the MATLAB/SIMULINK analysis.
7. Finally, attach your final Simulink file (**Lab1.mdl**) and the MATLAB driver file (**Lab1\_driver.m**) to your submission.

**Submission:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by 11:59 p.m. on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- Date, To, From, Subject
- Written in first person from you
- Written with minimal spelling and grammar errors
- Purpose, procedure, results and conclusions of the laboratory experiment (the procedure should be very short, a high level summary of what you did for each part). The procedure should be two paragraphs at the most.
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text
- Also, you must to include a statement in your memo that this is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results



## Lab 2

### First and Second Order Circuits

**Objective:** In this lab, you will use the NI myDAQ and MATLAB with Simulink and MultiSim to measure the response and characteristics of first order and second order circuits. In addition, you will use this analysis to identify characteristics of the first order system such as the static gain and time constant.

**Equipment:** Laptop with MATLAB and MultiSim

NI my DAQ

RLC Meter

0.01  $\mu\text{F}$  capacitor

1  $\mu\text{F}$  capacitor

1  $\text{k}\Omega$  resistor (x 2)

10  $\text{k}\Omega$  variable resistor

33 mH inductor

**Pre-lab:** Read this entire lab procedure thoroughly and complete the following tasks before your laboratory session.

Become familiar with the NI my DAQ instruments by reviewing the documentation in the box and the tutorials at the following links:

Introduction to NI myDAQ: <http://decibel.ni.com/content/docs/DOC-13041>

DMM – ohmmeter: <http://decibel.ni.com/content/docs/DOC-12880>

Analog Output and Input: <http://decibel.ni.com/content/docs/DOC-12884>

### Analytical Derivations

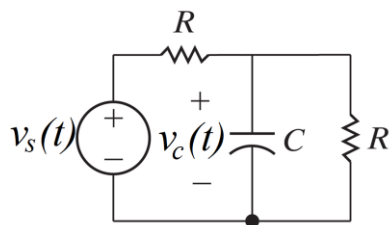


Figure 1: RC Circuit

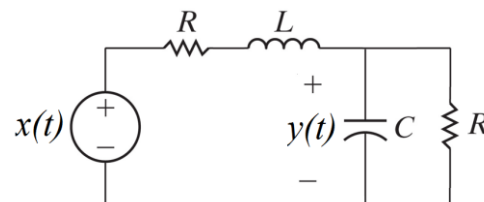


Figure 2: RLC Circuit

- Derive the governing differential equation in standard form for the circuit in Figure 1 and state the static gain and time constant.
- State the solution to the governing differential equation in Figure 1 assuming zero initial conditions, a step input with constant amplitude,  $A$ ,  $C = 1 \mu\text{F}$  and  $R = R = 1 \text{ k}\Omega$ .
- Derive the governing differential equation in standard form for the circuit in Figure 2 and state the static gain, natural frequency and damping ratio.
- If  $A = 1$ ,  $L = 33 \text{ mH}$ ,  $C = 0.01 \mu\text{F}$  and  $R = R = 1 \text{ k}\Omega$ . What is the static gain? natural frequency? and damping ratio?

**MATLAB/SIMULINK Problem**

One of the standard forms for a second order system is

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2y(t) = \omega_n^2Kx(t),$$

where  $\zeta$  is the damping ratio,  $\omega_n$  is the natural frequency, and  $K$  is the static gain. Use this form of the standard second order system in the remainder of this problem. Use a MATLAB driver to simulate a system described by this differential equation. Similar to what you did in Lab 1 for the first order system, solve for the highest power derivative (as a function of the input and lower power derivatives). Next, integrate  $\ddot{y}(t)$  to yield  $\dot{y}(t)$  and then integrate again to yield  $y(t)$ . Therefore, you will need two integrators and two feedback loops and one input into your summing block (click on the summing block and modify it to get three inputs) in SIMULINK. You may need to click on the gain block and then choose **flip block** to get the correct direction.

- We want to look at the step response, so the input to your system should be a step. The amplitude of the step should be controlled by the MATLAB program, the length of the simulation should be controlled by the MATLAB program, and the step should start at time zero.
- Your Simulink file should only contain variables (static gain, natural frequency, damping ratio, amplitude of the step, length of the simulation)
- Plot the transient output of the system and the steady state output,  $KA$ , on the same graph.





- If you use the parameters,  $\zeta = 0.1$ ,  $\omega_n = 20$ ,  $K = 2$ ,  $A = 1$ , and  $T_f = 3$  (final simulation time), you should get results like that shown below.

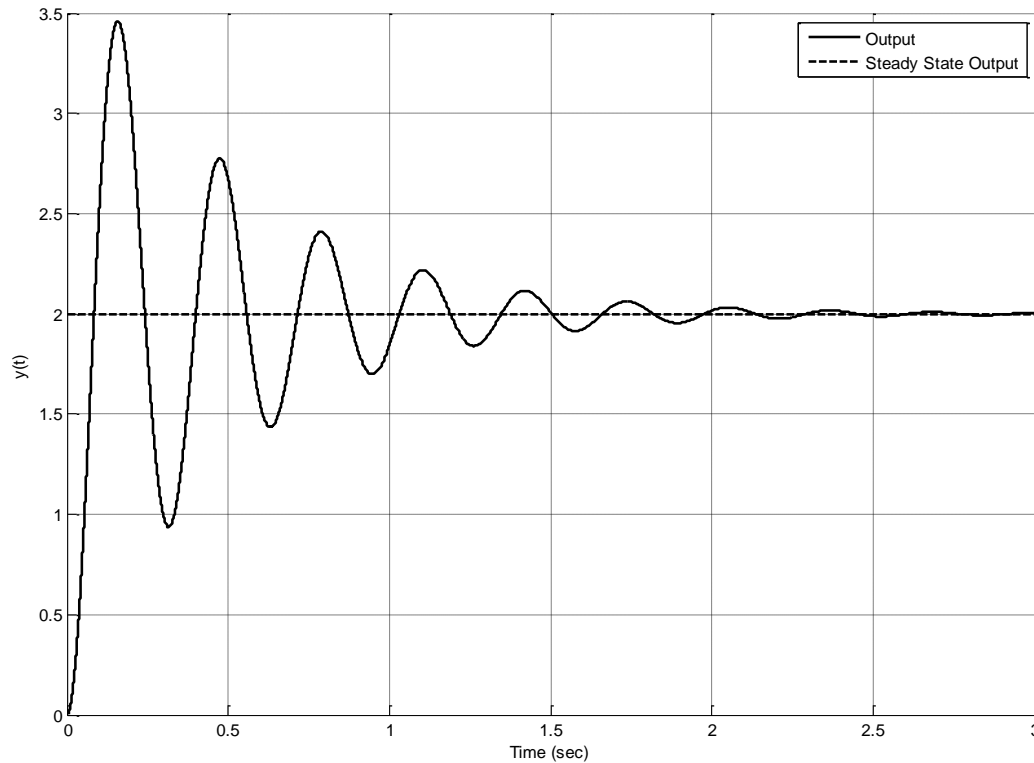


Figure 3: Second-order underdamped step response

- You should submit the following three items as part of your pre-lab submission:
  - A printout of your figure with a title and the axes labeled with units
  - A printout of your MATLAB CODE with a header comment with your name, the assignment and the date, code, and
  - A printout of your SIMULINK model

If you have any questions, please ask your instructor.

**Procedure:**

Part I – Component Measurements

Launch the NI ELVISmx Instrument Launcher from the National Instruments folder and you should see the toolbar appear on your desktop. Click on the DMM on the Instrument Launcher and use the NI myDAQ to measure the actual values of the two 1 kΩ resistors and compare them to the nominal values. Connect the red and black probes between the HI and COM terminals on the bottom of the NI myDAQ, press the Ω for resistance and press run to acquire the reading. If you get an error, make sure that you change the mode to specify range not auto. See Figure 3 for an example of the desktop.

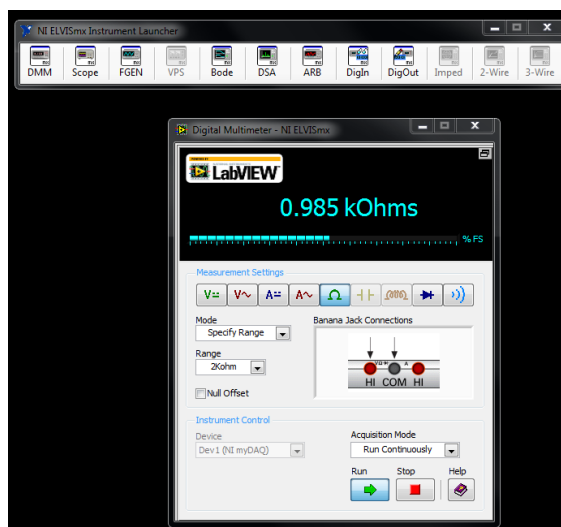


Figure 3: DMM Measurements with the NI myDAQ

You will need to use the RLC meter available in the classroom to measure the 1 μF capacitor. Compare the measured values to the nominal values of these components by including a data table with error analysis in your lab memo submission (see Table 1). All data tables included in the memo should have a number and caption and be referenced in the text. Note that percent error is calculated by using the formula,  $\% \text{ error} = \frac{\text{measured} - \text{nominal}}{\text{nominal}} \times 100$ .

	measured	nominal	% error
C, nF		1000	
R, Ω		1000	
R, Ω		1000	

Table 1: Inductor and Capacitor Actual Values

Use the measured values of the resistors and capacitor to calculate the time constants,  $\tau$ , for the circuit shown in Figures 1 ( $\tau = R_{th}C$ ). This will be the nominal value for the time constant for this circuit should be compared to the theoretical values found in the prelab. Table 2 provides an example of this comparison,  $\% \text{ error} = \frac{\text{nominal} - \text{theoretical}}{\text{theoretical}} \times 100$ . Note that since you are using 5% resistors and 20% capacitors, the time constant may have up to a 20% error.

	nominal	theoretical	% error
RC Circuit, $\tau$ , ms		0.5	

Table 2: First Order Circuit Time Constant,  $\tau$

Use the measured values of the resistors and capacitors to calculate the static gain, K, for the circuit shown in Figure 1. This value will be the nominal static gain for RC Circuit for the remaining of the measurements in the lab (see Table 3).

	nominal	theoretical	% error
RC Circuit, K		0.5	

Table 3: First Order Circuit Static Gain, K

Part II – First order circuit characteristics

In this part, you will use the NI myDAQ to estimate the time constant of the two RC circuits by measuring the 10% - 90% rise time,  $t_r$ , using the formula,  $\tau = \frac{t_r}{\ln(9)}$ .

- a) Connect red leads from to the +15V, and -15V terminals on the MyDAQ. Connect **black** leads to the two AGND terminals on the MyDAQ. Connect **white** leads to the AO 0 terminal on the MyDAQ. These are the power and function generator connections.
- b) Connect **green** leads to the AI 0- and AI 1- terminals on the MyDAQ. Connect **yellow** leads to the AI 0+ and AI 1+ terminals on the MyDAQ. These are the oscilloscope connections.
- c) Build the circuit in Figure 1 on your breadboard with **R = R = 1 k $\Omega$ , C = 1  $\mu$ F**.
- d) You will use the MyDAQ function generator to provide the source voltage so connect the leads from the AO 0 and AO AGND terminals across the input to the RC circuit.
- e) You will use the MyDAQ oscilloscope to measure the input and output of the RC circuit. Connect the leads from the AI 0+ and AI 0- to the left side of the resistor and to ground in



- parallel with the function generator connections. Connect the leads from the AI 1+ to the positive terminal on the capacitor and the AI 1- to the negative end of the capacitor.
- Click on the Scope and FGEN to launch the oscilloscope and function generator on the NI MYDAQ. Set the Function Generator to a square wave at 100 Hz and 2.00 V<sub>pp</sub> and 0.00 V offset. Confirm that signal route is AO 0 and click run to provide the source voltage to the circuit.
  - Set the oscilloscope Channel 0 settings to Source: AI 0, 500 mV/div and Timebase of 2 ms/div. Set the oscilloscope Channel 1 settings to Source AI 1, 500 mV/div and set the Trigger to Edge and the Chan 0 Source. Make sure to **enable** both channels and click run to run continuously.
  - Your function generator and scope output should look similar to the image in Figure 4 which represents the capacitor charging and discharging repeatedly. *Note that this was a different circuit so there will be some differences!!*

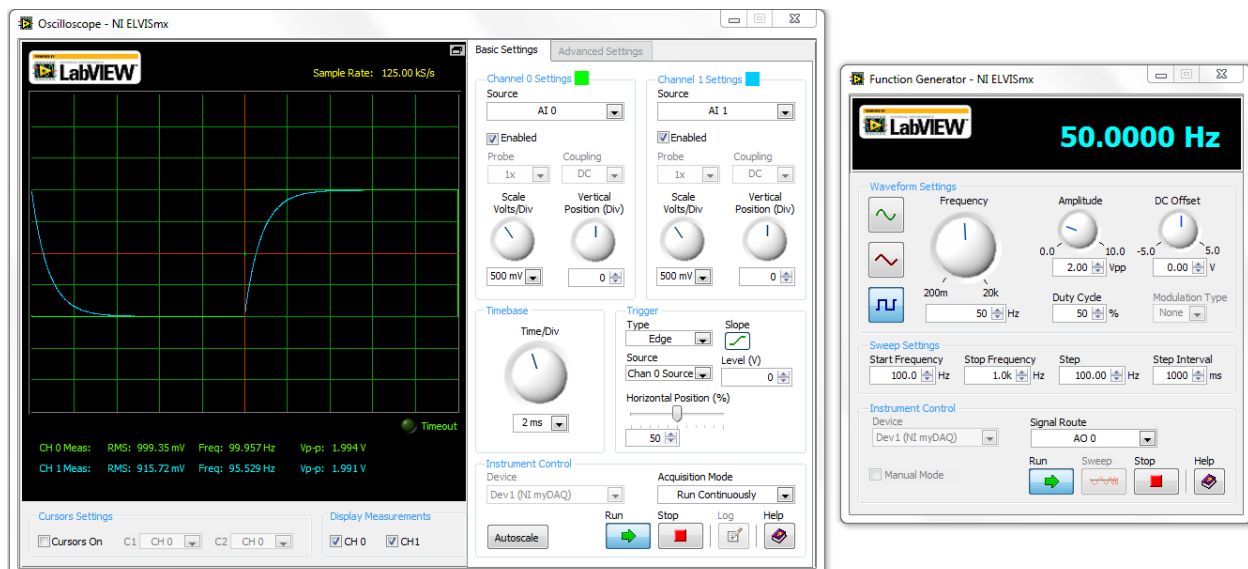


Figure 4: Scope measurements with the NI myDAQ

- Next, we will measure the rise time on the signal so click the check box to enable the cursors under the display window.
- Use the drop down menu to set C1 to CH 1 and C2 to CH1 and check both channels to display measurements.



- k) Find the yellow C1 and C2 cursors on the left edge of the window and move them toward the rising edge of the output voltage. Align C1 at the bottom of the output signal on top of the rising edge for the input square wave. Align cursor C2 at the top of the signal after it has reached steady state. Find the difference between the Channel 1 low and high value by using  $C2 - C1$ .
- l) Next, move C1 to the  $[low\ value\ of\ channel\ 1 + 0.1(C2-C1)]$  and move C2 to the  $[high\ value\ of\ channel\ 1 - 0.1(C2-C1)]$ . The time difference between these two cursors is displayed as  $dT$ . This time difference represents the time it takes for the waveform to go from 10% to 90% of the final value so  $dT$  is also the rise time,  $t_r$ . Use the rise time to calculate the time constant,  $\tau$  and compare it to the nominal value in a data table in your lab memo submission. If this value is not within 15% of the nominal value, then ask for help.
- m) In this part, you will measure the static gain of the circuit in and compare the actual results to the theoretical results found in the prelab and the nominal results found in part I.
- n) Use the peak to peak voltage for channel 2 and channel 1 after the output reaches steady-state to calculate  $K = y_{ss}/x_{ss}$ .
- o) Create a data table to compare these results and include the analysis in your memo submission.
- p) Next use the snipping tool or alt-PrtSc to capture a screen shot of the oscilloscope display for inclusion in your lab memo.

### Part III – Time constant estimation – Alternate method

In this part you will use an alternate method to estimate the time constant for the RC circuit in Figure 1. This is estimation based upon the fact that a capacitor will charge to 63.2% of its final value in one time constant,  $y(t) = KA(1 - e^{-t/\tau}) = KA(1 - e^{-1}) = 0.632KA$ .

- a. Similar to part II, enable both cursors and set C1 and C2 to CH 1. Align C1 at the bottom of the output signal at the rising edge of the step input and align C2 at the top of the output signal or maximum value of channel 1. Find the difference between C2 and C1 and calculate 63.2% of this number.

- b. Move C2 to the [low value of channel 1 + 0.632(C2-C1)], you may not be able to get the value exactly but get as close as you can. The time difference between these two cursors is  $dT$  and represents the estimation of the time constant,  $\tau$ .
- c. Compare this estimate of the time constant to the one found in part II and demonstrate the result of the analysis in a data table in the memo submission.

#### Part IV – Time constant estimation using MultiSim

In this part, you will build the circuit in Figure 1 in MultiSim and use the result to estimate the time constant.

1. Build the circuit in Figure 1 in MultiSim and use the NIELVISmx Function Generator and NIELVISmx oscilloscope to measure the input and output. These instruments are located on the toolbox down the right side of the window near the Agilent instruments.
2. Make the function generator and oscilloscope settings the same as in part II. (see Figure for the proper setup).
3. Confirm that the results are similar to Parts II and III and capture a screen shot for inclusion in your lab memo submission.

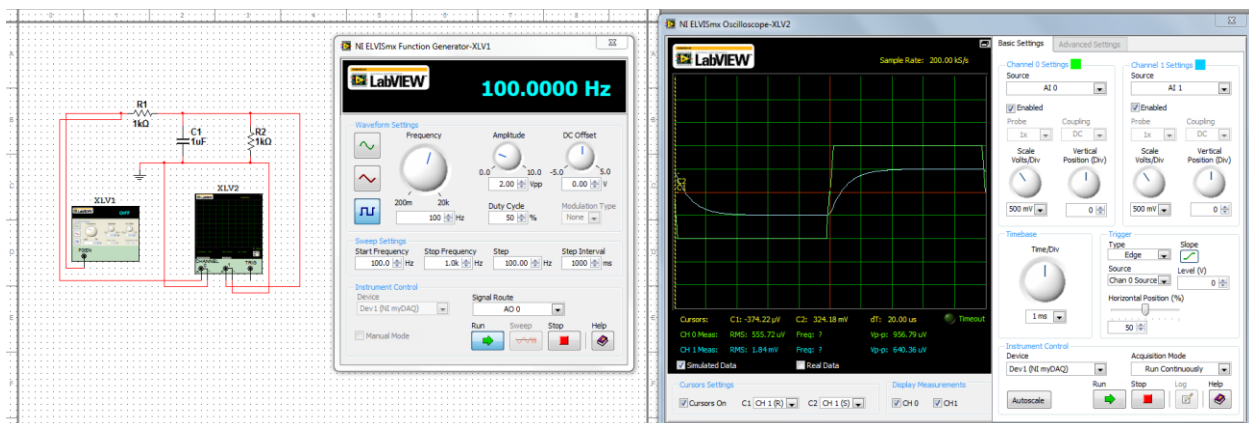


Figure 5: MultiSIM simulation of an RC circuit

#### Part V – Time constant estimation of a different RC circuit

In this part, you will measure the time constant of different RC circuits to examine how it affects the system output.



1. Replace the 1 k $\Omega$  resistor in parallel with the capacitor in the circuit you built with a 10 k $\Omega$  variable resistor (potentiometer). Make sure the connection point is between the center tap and one of the side terminals. If you place the leads across the two terminals, the potentiometer acts as a 10 k $\Omega$  resistor.
2. Adjust the variable resistor so that the time constant is significantly smaller than the original 0.5 ms.
3. Adjust the **Volts/Div** and **Time/Div** on the oscilloscope so that the waveform is as large as possible and displays one full period. You may also have to adjust the function generator frequency in order to get one full period on the oscilloscope screen.
4. Repeat the steps in part II or III of the lab procedure to estimate the time constant.
5. Include the time constant estimation and oscilloscope screen capture in your lab memo submission
6. Adjust the variable resistor so that the time constant is significantly larger than the original 0.5 ms and repeat steps 3 through 5.

#### Part VI – Time constant estimation for an unknown RC circuit

In this part, you will measure the time constant of an unknown first-order circuit. Note that there will be a question like this on your lab practical to make sure that you completely understand the procedure.

- 1) Get a numbered black box from your instructor and write the number down. Make sure to include the box number in your lab memo submission.
- 2) Place the box so that the number is right side up and the red terminals are positive voltage and the black connectors are ground.
- 3) Note that since you will devise a method to estimate the time constant, these instructions are not complete.
- 4) Connect the function generator AO and AGND to the left side of the box, the input. Connect the oscilloscope, AI 0+ AGND to the left side of the box, the input.
- 5) Connect the oscilloscope, AI 1+ AI 1- to the right side of the box, the output



- 6) Set the function generator input signal to a square wave with 2 Vpp and a frequency between 50 and 200 Hz.
- 7) Adjust the **Volts/Div** scale on the oscilloscope so that the output signal is as large as possible on the screen.
- 8) Adjust the **Time/Div** scale on the oscilloscope so you can view one full period of the output waveform.
- 9) Use the cursors to measure the time constant for the unknown first-order circuit.
- 10) You should include this estimate of the time constant and the black box number in your lab memo submission.
- 11) You should also include a screen shot of the oscilloscope output in your lab memo submission.

### Part VII – Underdamped second order system characteristics using the NI myDAQ

*In this part you will examine the characteristics of an RLC circuit to determine the static gain, natural frequency and damping ratio. Figure 6 shows some of the important characteristics of the step response for an underdamped second order system including Percent Overshoot (PO), Time to peak ( $T_p$ ), settling time ( $T_s$ ), and steady state value ( $y(\infty)$ ).*

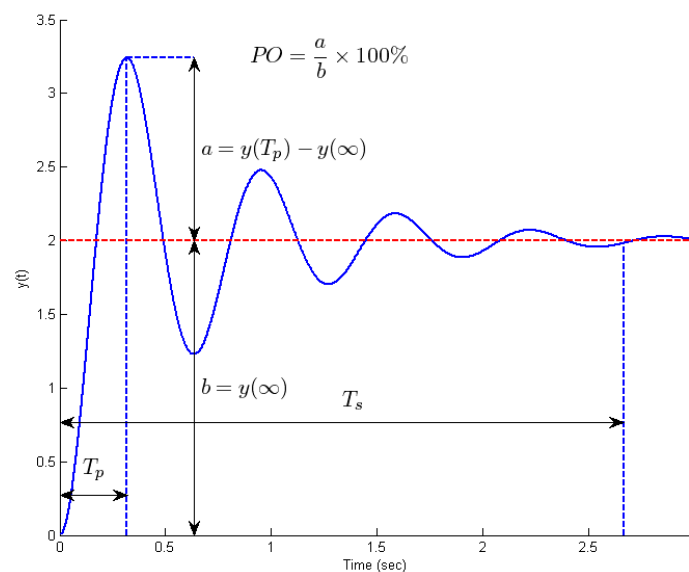


Figure 6: Second-order underdamped system characteristics



- Use the RLC meter and NI myDAQ to measure the nominal value for the 33 mH inductor.
- Build the circuit in Figure 2 **without** the resistor in parallel with the capacitor.
- Calculate the **theoretical** values of the static gain, natural frequency and damping ratio for this circuit using the nominal values for the resistor, inductor and capacitor.
- Calculate the **nominal** values of the static gain, natural frequency and damping ratio for this circuit using the measured values for the resistor, inductor and capacitor.
- Compare the nominal values to the theoretical values in a data table. This comparison should include performing an error analysis and showing the results in a data table in your memo submission. This table should have a table number and caption and be referenced in the text.
- Calculate the theoretical values for the time to peak, percent overshoot and settling time for the RLC circuit using the nominal component values.
- Calculate the nominal values for the time to peak, percent overshoot and settling time for the RLC circuit using the measured component values.
- Compare the nominal values to the theoretical values in a data table. This comparison should include performing an error analysis and showing the results in a data table in your memo submission. This table should have a table number and caption and be referenced in the text.
- The circuit characteristics with the measured component values will be the nominal values for the remainder of the measurements on this circuit.
- You should now have the circuit in Figure 2 built **without** the resistor in parallel with the capacitor. Please confirm that the component values are  $R = 1\text{ k}\Omega$  resistors,  $L = 33\text{ mH}$  and  **$C = 0.01\text{ }\mu\text{F}$**  capacitor. ***Note that the capacitor is different from Parts I - VI!***
- Set up the NI myDAQ function generator to supply a 1.5 kHz 2.00 Vpp square wave.
- Set the oscilloscope channels 0 and 1 to measure the input source and output voltage across the capacitor with 500 mV/division and a 100  $\mu\text{s}$  timebase.

- The trigger should be edge with the source as channel 0. Your system response should look similar to the scope display in Figure 7. *Note that this image is from a different circuit so they won't be exactly the same!*

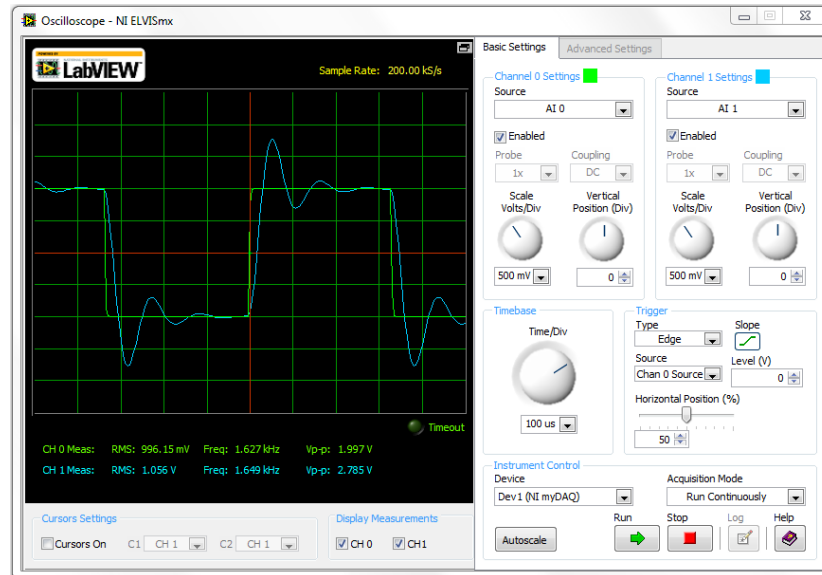


Figure 7: Underdamped system response

- Use the oscilloscope cursors to find the time to peak by putting C1 at the rising edge of the output voltage and C2 at the center of the first peak,  $dT$  represents the time to peak.
- Use the oscilloscope cursors to find the damping frequency,  $\omega_d$ , in rad/s by measuring the period,  $dT$ , between two consecutive peaks and using  $\omega_d = 2\pi/T_d$ .
- Use the oscilloscope cursors to determine the settling time when the output gets within 2% of its final value and stays there. C1 should be at the rising edge of the signal and C2 should be moved until the voltage is within 2% of the final value and at this point  $dT$  represents the settling time.
- Use the oscilloscope cursors to measure the percent overshoot using the formula,  $\frac{y_{peak} - y_{ss}}{y_{ss}} \times 100$  with C1 at the maximum peak and C2 at the final value
- Use the oscilloscope cursors to find the static gain by measuring the peak to peak value of the input channel and the peak to peak value of the output channel at steady-state.



- The damping ratio can be calculated from the measured percent overshoot by using the following formula,  $\zeta = \frac{\ln^2(OS)}{\sqrt{\ln^2(OS) + \pi^2}}$ . Note in this formula the overshoot should be expressed as the decimal equivalent of the percentage.
- The natural frequency can be calculated from the measured damping frequency and damping ratio by using the following formula,  $\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}}$ .
- You should summarize all of the characteristics for this circuit in a data table and compare them to the nominal values with an error analysis in the memo submission.

### Part VIII – System Types

- Put the 10 k $\Omega$  variable resistor (potentiometer) in parallel with the capacitor. Make sure the connection point is between the center tap and one of the side terminals. If you place the leads across the two terminals, the potentiometer acts as a 10 k $\Omega$  resistor.
- Use the screw driver to turn the potentiometer all the way clockwise to the 9 and observe the output. What type of response does this represent? You should capture this scope display for inclusion in your lab memo.
- Use the screw driver to turn the potentiometer counterclockwise to the 2 and observe the output. What type of response does this represent? You should capture this scope display for inclusion in your lab memo.
- Use the screwdriver to adjust the potentiometer until the output voltage has the fastest settling time with no overshoot. What type of response does this represent? You should capture this scope display for inclusion in your lab memo.
- Recall that the system parameters are  $\omega_n = \frac{1}{\sqrt{LC}}$  and  $\zeta = \frac{R_{th}\sqrt{C}}{2\sqrt{L}}$  so there is a direct relationship between resistance and the damping ratio but not the natural frequency.
- Remember all figures should have a number and caption and be referenced in the text of the lab memo.

**Submission:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by 11:59 p.m. on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- Date, To, From, Subject
- Written in first person from you
- Written with minimal spelling and grammar errors
- Purpose, procedure, results and conclusions of the laboratory experiment (the procedure should be very short, a high level summary of what you did for each part). The procedure should be two paragraphs at the most.
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text
- Also, you must to include a statement in your memo that this is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results



### Lab 3

## Time Domain Modeling of One Degree of Freedom Systems

### Overview

*In this lab you will model two one degree of freedom (second order) systems using time-domain analysis. The goal is develop some intuition into how the parameters in a second order system affect the output by varying parameters in a model to match the step response of two second order systems. You will also use the log-decrement method to estimate these parameters. If your station has both a rectilinear (model 210) and torsional (model 205) system you should make one model of each type of system. Otherwise make two different systems (and corresponding models) using either the model 210 or model 205.*

### Prelab

Read this entire lab procedure and background theory thoroughly. Then answer the following question on engineering paper and submit it in class the day before the lab session.

- Draw a free body diagram of the forces on the mass for the system shown in Figure 1.
- Show that the equations of motion can be written as:

$$m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + (k_1 + k_2)x_1(t) = F(t)$$

or

$$\frac{1}{\omega_n^2} \ddot{x}_1(t) + \frac{2\zeta}{\omega_n} \dot{x}_1(t) + x_1(t) = KF(t)$$

- Express the damping ratio,  $\zeta$ , natural frequency,  $\omega_n$ , and static gain,  $K$ , in terms of  $m_1$ ,  $k_1$ ,  $k_2$ , and  $c_1$ .
- For the system in Figure 2, express the damping ratio,  $\zeta$ , natural frequency,  $\omega_n$ , and static gain,  $K$ , in terms of  $J$ ,  $c$  and  $k$ .

Background Theory

A one degree of freedom rectilinear mass-spring-damper system can be modeled as shown in Figure 1.

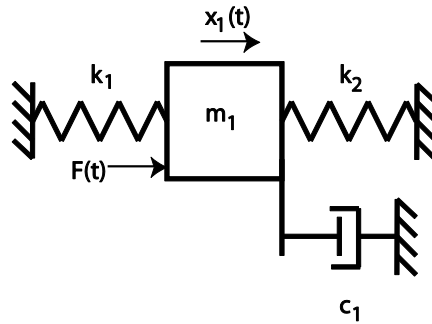


Figure 1: 1 DOF Mechanical Rectilinear System

By drawing a free body diagram and balancing forces, we get the equation of motion:

$$m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + (k_1 + k_2)x_1(t) = F(t)$$

A one degree of freedom rotational mass-spring-damper system can be modeled as

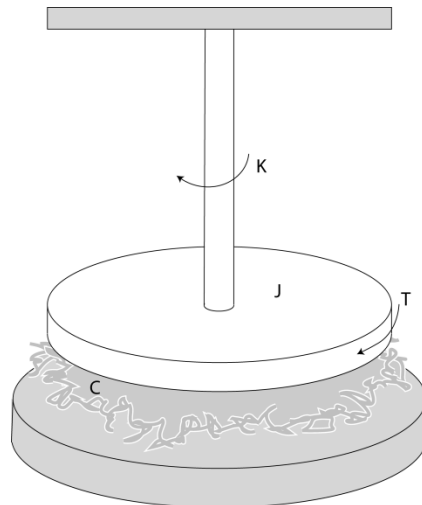


Figure 2: 1 DOF Mechanical Torsional System

By drawing a free body diagram and balancing torques, we get the equation of motion

$$J\ddot{\theta}(t) + c\dot{\theta}(t) + k\theta(t) = T(t)$$

Despite the fact that the systems appear quite different, the transfer functions for both of the one degree of freedom systems can be put into the standard form for modeling second order circuits:

$$\ddot{y}(t) + 2\zeta\omega_n \dot{y}(t) + \omega_n^2 y(t) = K\omega_n^2 x(t)$$



### Lab 3

## Time Domain Modeling of One Degree of Freedom Systems

### Overview

*In this lab you will model two one degree of freedom (second order) systems using time-domain analysis. The goal is develop some intuition into how the parameters in a second order system affect the output by varying parameters in a model to match the step response of two second order systems. You will also use the log-decrement method to estimate these parameters. You will complete this procedure on two different rectilinear (model 210) systems.*

### Prelab

Read this entire lab procedure and background theory thoroughly. Then answer the following question on engineering paper and submit it in class the day before the lab session.

- Draw a free body diagram of the forces on the mass for the system shown in Figure 1.
- Show that the equations of motion can be written as:

$$m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + (k_1 + k_2)x_1(t) = F(t)$$

or

$$\frac{1}{\omega_n^2} \ddot{x}_1(t) + \frac{2\zeta}{\omega_n} \dot{x}_1(t) + x_1(t) = KF(t)$$

- Express the damping ratio,  $\zeta$ , natural frequency,  $\omega_n$ , and static gain,  $K$ , in terms of  $m_1$ ,  $k_1$ ,  $k_2$ , and  $c_1$ .
- For the system in Figure 2, express the damping ratio,  $\zeta$ , natural frequency,  $\omega_n$ , and static gain,  $K$ , in terms of  $J$ ,  $c$  and  $k$ .

Background Theory

A one degree of freedom rectilinear mass-spring-damper system can be modeled as shown in Figure 1.

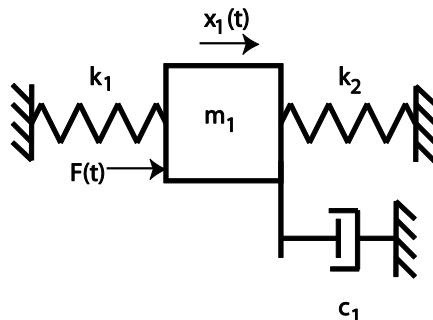


Figure 1: 1 DOF Mechanical Rectilinear System

By drawing a free body diagram and balancing forces, we get the equation of motion:

$$m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + (k_1 + k_2)x_1(t) = F(t)$$

A one degree of freedom rotational mass-spring-damper system can be modeled as

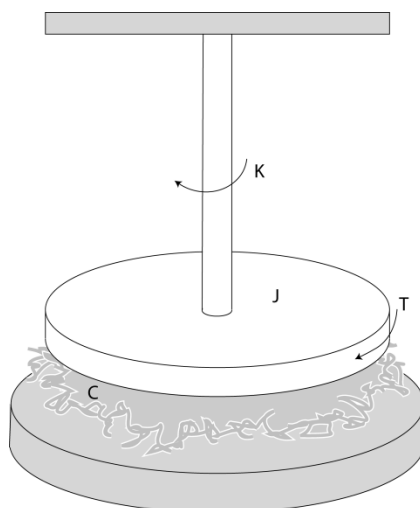


Figure 2: 1 DOF Mechanical Torsional System

By drawing a free body diagram and balancing torques, we get the equation of motion

$$J\ddot{\theta}(t) + c\dot{\theta}(t) + k\theta(t) = T(t)$$

Despite the fact that the systems appear quite different, the transfer functions for both of the one degree of freedom systems can be put into the standard form for modeling second order circuits:

$$\ddot{y}(t) + 2\zeta\omega_n \dot{y}(t) + \omega_n^2 y(t) = K\omega_n^2 x(t)$$



The system parameters are the static gain,  $K$ , the natural frequency,  $\omega_n$  and the damping ratio,  $\zeta$ . These parameters must be determined in order to create the system model.

### Procedure

#### **PART I - Set up the files**

- Create a folder for ECE 205 on the desktop of the computer and create a Lab 3 folder in this folder.
- Download and extract the **Lab3Files.rar** from the Angel course website Lab folder into the folder created on the computer desktop.
- Start MATLAB and change the default folder to the folder where you extracted the Lab 4 files.

#### **PART II - Set up the communication**

You will need to go through the following steps for **two different** configurations. This should be two rectilinear systems with different masses and/or springs with or without the damper. In this part, you will get the system ready to run and start communications between Simulink, the miniPMAC card and the ECP system.

1. Press the power button to turn on the ECP system. It is on the top bench next to the computer.
2. To inform the ECP system that we will use Simulink and the real-time windows target, click **Start -> Programs -> ECP**
3. Click on **Utility-> Download Controller Personality File**
4. Select **C: -> Program Files -> ECP Systems -> cn** (it may default to this)
5. Finally select **m210\_rtw\_3.pmc** for the Model 210 (or **m205\_rtw\_3.pmc** for the Model 205) and click on **open**. Wait for the ECP system to load the personality file, then close the window completely (do not just minimize it).
6. Now you need to reset the system. This should be done each time before you run the system. From MATLAB, type **simulink** at the command prompt to open the Simulink file, **ECPDSPReset.mdl** (see Figure 1).

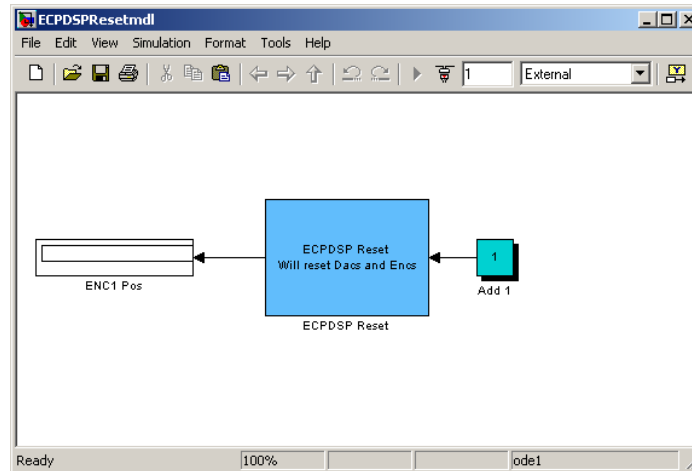


Figure 1: ECPDSPReset.mdl

- Verify that the **Base I/O Address** is correct for your work station by double clicking on the blue ECPDSP Reset box, and checking the parameters in the window that opens (see Figure 2). Since all of the computers in the room are Dell computers, the **Base I/O Address** should be **'0xD800'**. Also change the **Sample Time (Sec)** to **0.005**. Click Apply→OK and make sure that you save the file if you had to change the address.

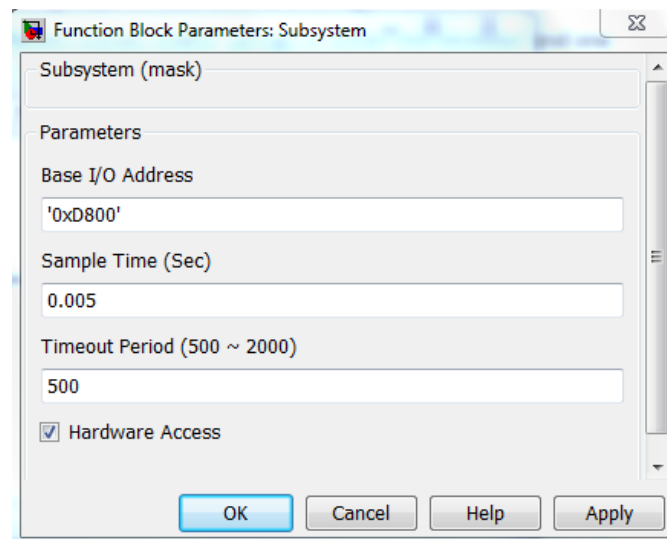


Figure 2: ECPDSP Reset Function Block Parameters

- Compile** the **ECPDSPReset.mdl** file by clicking on the icon shown in Figure 3. Make sure to wait for the file to finish compiling when the messages in the MATLAB command window stop. Ignore any errors and continue.

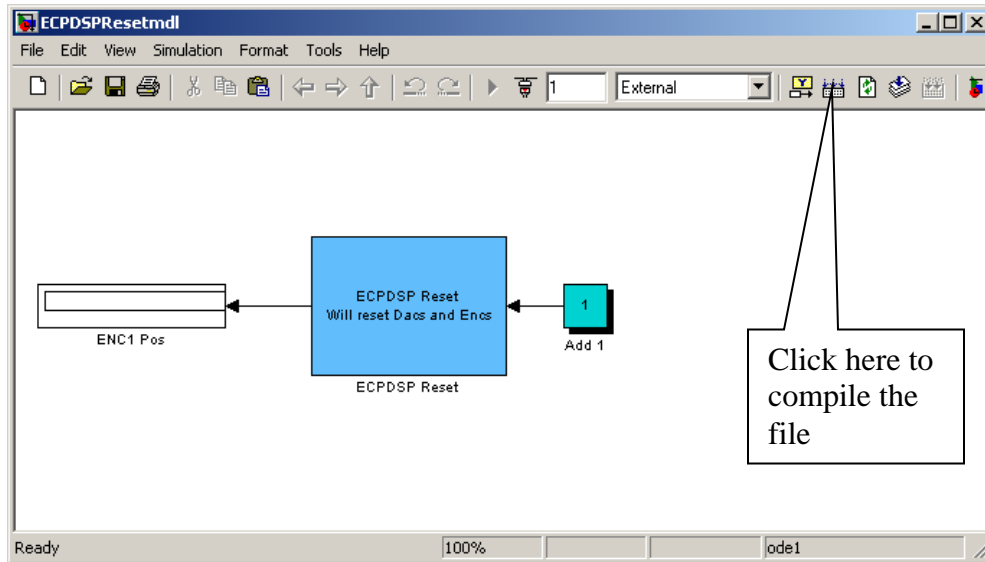


Figure 3: Compiling ECPDSPReset.mdl

9. **Connect** to the system as shown in Figure 4 and run the system by clicking **play** as shown in Figure 5.

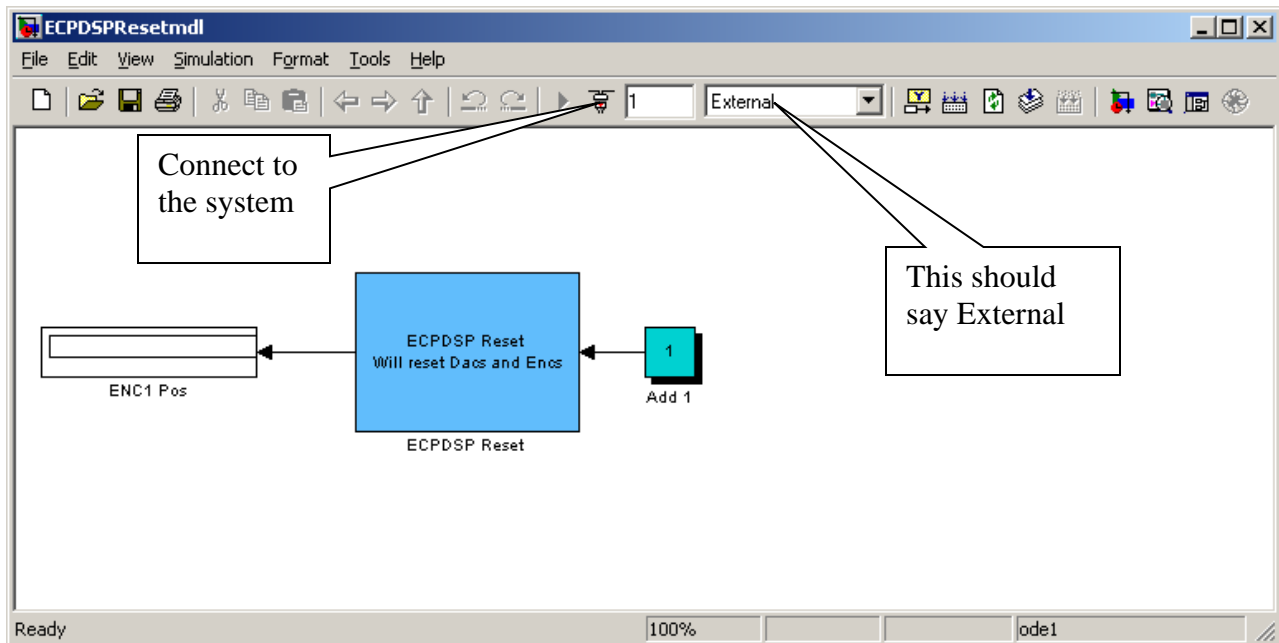


Figure 4: Connecting to the ECP system

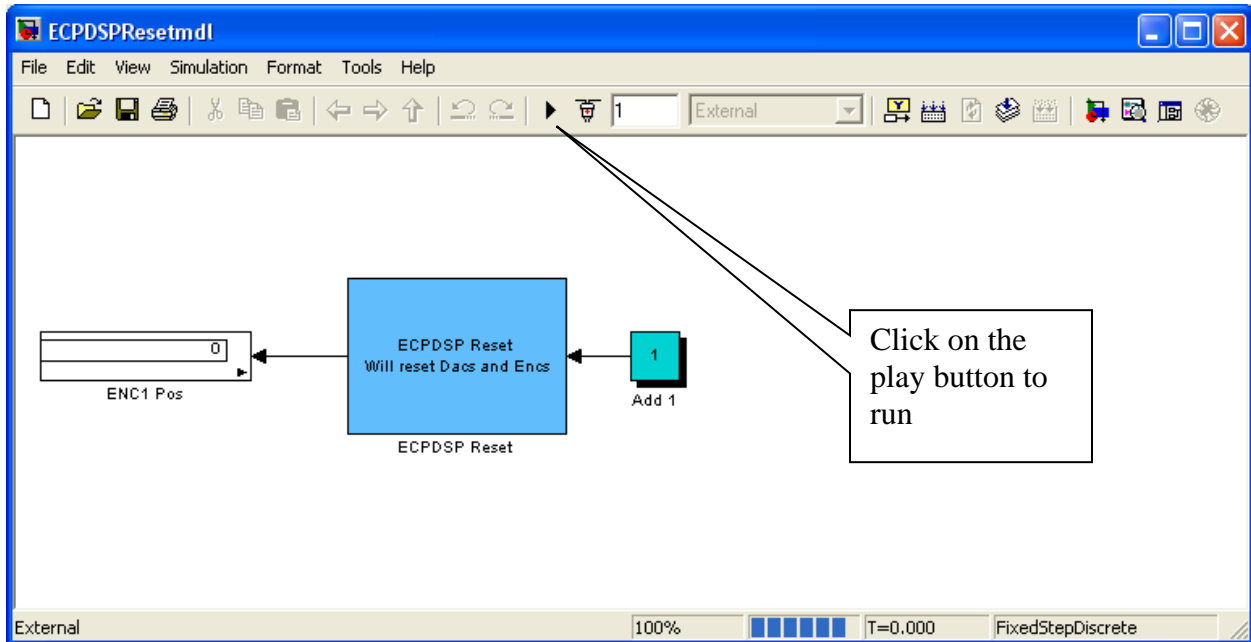


Figure 5: Running the ECP system

10. If you all of this has been done correctly, the MATLAB command window will indicate that you have connected and disconnected from the system (loaded, unloaded). This indicates that the counters have been reset and the system has been zeroed.

### PART III - Second order rectilinear system (ECP Model 210)

1. In this part you will set up the mechanical system that you will create the model for. You will need to go through the following steps for **two different** configurations. This will be two rectilinear systems with different masses and/or springs, with or without the damper.
2. Use the Allen wrenches at the bench to lock all of the carts except for the first one closest to the motor. In addition, you need to have at least one spring connected to the cart and at least one mass on the cart.
3. Open up the Simulink file, **Model210\_Openloop.mdl**, it should look like Figure 6. The yellow block in the middle is what actually connects to the ECP system. Note that the output of this system is labeled  $x_1$ , the cart position. **Do not change this!**

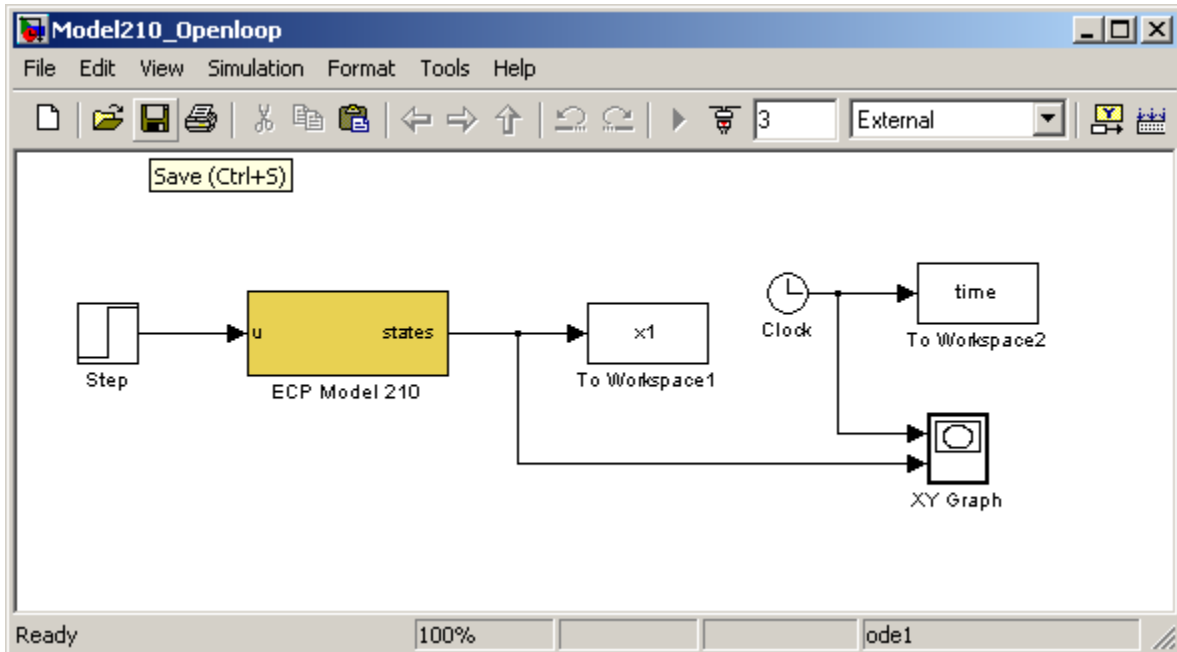


Figure 6: Model210\_Openloop.mdl

- You should also verify that the **Base I/O Address** for this system is set correctly for the Dell computers. You can do this by double clicking on the yellow block and the window in Figure 7 will open up. Double click on the blue ECPDSP Driver block to confirm that the Base I/O Address is correct. Click Apply->OK and then close the windows that were opened and save the **Model210\_Openloop.mdl** file.

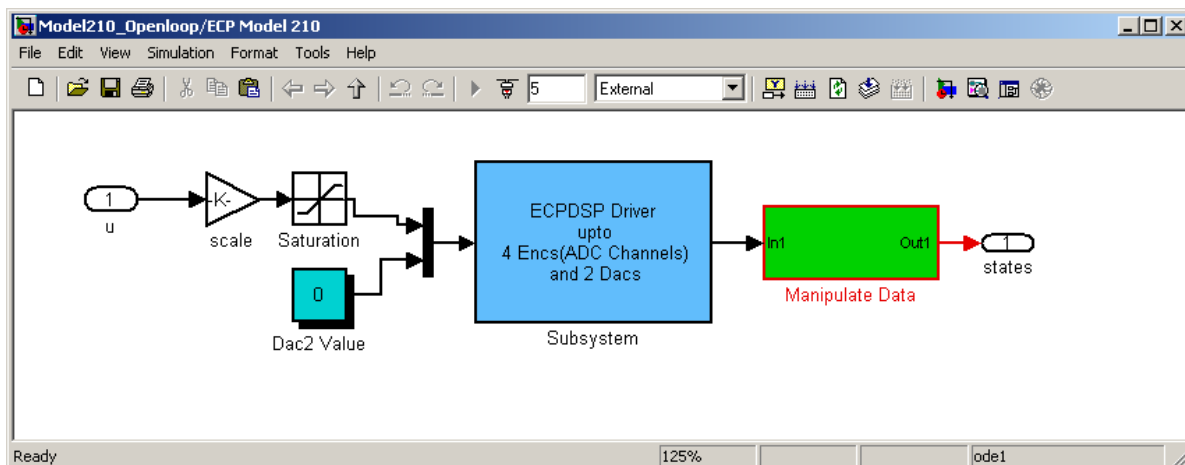


Figure 7: Model210\_Openloop/ECP model 210



5. Change the input to the system by double clicking on the **Step** block in Figure 6. To start, enter a small step amplitude or input value such as **0.05**. This value is small because some of the systems have a static gain of more than 40 and the cart should not move more than a couple of centimeters. ***You may have to increase the step input to get your particular system to have a steady state value of at least 1 cm.*** Click **OK** and then save the file.
6. **Compile** the file just as you did in PART II and wait for the messages to finish in the MATLAB window. Next, **connect** to the system, and then **run** it and the cart should move. Ignore any errors that open up and continue. There are three (usual) outcomes here:
  - a) If the cart does not move, repeat PART II. You don't need to change the Base I/O Address or recompile, but you do need to reload the controller personality file and reset the system.
  - b) If the cart moved, but the XY Graph does not look smooth, the communications are still not correctly set up, repeat part II. You don't need to change the Base I/O Address or recompile, but you do need to reload the controller personality file and reset the system.
  - c) If the cart moved and the XY Graph looks reasonable, be sure the system recorded data for three seconds. If it did not record for three seconds, first look at the output level. If the maximum value of the output is between 1 and 2 cm (or larger), try resetting (ECPDSPReset) and rerunning the system a few times. If the maximum value of the output is less than about 1.0 cm, then increase the value of the amplitude in the **Step**, reset the system (ECPDSPReset) and rerun the system (Model210\_Openloop).

Before you go on, be sure your system has run for 2-3 seconds and recorded enough good data.

7. Modify the file **second\_order\_driver.m** to plot the data from the ECP system and the results from the simulation on the same graph. Be sure to use different line types and a legend. There are two different time arrays (one from the simulation and one from the ECP system), and two different outputs. You may need to use **axis** command to limit the time axis if your ECP system did not run long enough. Also, be sure the y-axis indicates the output is displacement (in cm).



8. Modify the values of the static gain, then the damping ratio, and then the natural frequency to try and get the model to match the system as closely as possible during the first part of the step (it is likely to get worse as time progresses.) You will have to iterate a bit on this. Once you have a reasonably good fit, copy the MATLAB figure into a word file (use **Edit**, then **Copy Figure**).
9. Before you go on, rename the variables from the ECP system so that you can use them again later, to do this type the following in the MATLAB command window,
 

```
time_save = time;
x1_save = x1;
```
10. Before you go on, Include step response graph in your lab memo submission with a figure number and caption referenced in the text.

#### **PART IV - The log decrement method**

In this part, you will use the log decrement method to estimate the natural frequency,  $\omega_n$  and damping ratio,  $\zeta$ , of a second order system. Note that this method does not determine the static gain.

- 1) Since the log-decrement algorithm is so useful, it has been automated and you can implement it by using the following steps:
  - Reset the system using **ECPDSPresetmdl.mdl**.
  - Modify **Model210\_Openloop.mdl** so the input has zero amplitude.
  - Compile **Model210\_Openloop.mdl** if necessary.
  - Connect **Model210\_Openloop.mdl** to the ECP system. (The mode should be **External**.)
  - Displace the first mass, and hold it.
  - Start **(play) Model210\_Openloop.mdl** and let the mass go.
  - After the cart finishes moving, run the m-file, **log\_dec.m**. This file should be in the same directory as **Model210\_Openloop.mdl** and **log\_dec.fig**. This routine assumes the position of the first cart is labeled x1 and the time is labeled time. (These are the defaults in **Model210\_Openloop.mdl**)



- The program **log\_dec** produces the GUI shown in Figure 8

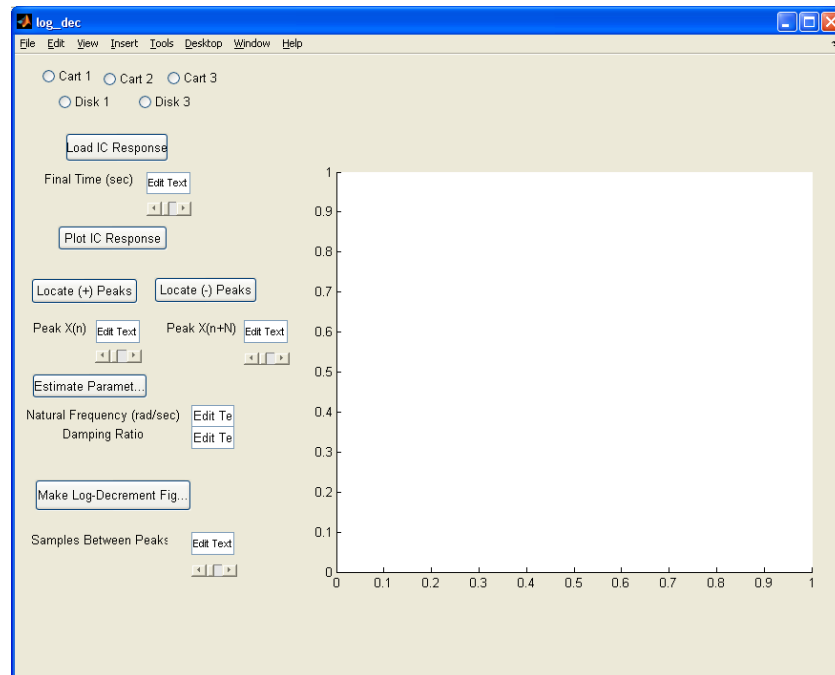


Figure 8: Model210\_Openloop/ECP model 210

You need to

- Select **Cart 1**
- Select **Load IC (initial condition) Response** (the variables time and x1 or theta1 will be loaded from the workspace). At this point some initial estimates will be made.
- Set/modify the **Final Time**
- Select **Plot IC Response** to plot the initial condition response
- Choose to identify the positive peaks (**Locate + Peaks**) or negative peaks (**Locate - Peaks**). If the peaks are not numbered consecutively, you need to decrease the **Samples Between Peaks** and try again until all peaks have been identified.
- Choose the initial peak (**Peak x(n)**) and final peak (**Peak x(n+N)**) to use in the log-decrement analysis. These should be fairly close to the beginning of the initial condition response. Don't try and use more than a few peaks.
- Select **Estimate Parameters** to get the initial estimates of  $\zeta$  and  $\omega_n$



- Select **Make Log-Decrement Figure** to get a plot and summary of the results. You need to include this figure in your memo.
- 2) Use the values of the natural frequency and damping ratio determined using the log decrement method in your simulation. Before you run your simulation you will need to recover the saved variables by typing in the MATLAB command window by typing the following:
- ```
time = time_save;
x1 = x1_save;
```
- 3) Include the log decrement graph in your lab memo submission with a figure number and caption referenced in the text. This means you should now have 3 graphs to submit for the system that you built.

**PART V - Model a different second order system**

*In this part, you will repeat Parts III and IV of the above procedure on a different second order system.*

Your memo should contain 3 graphs for each of the systems built, six total. Each graph should include a figure number, caption, descriptive title, and axes labeled with units and referenced in the text. The body of the memo should be a very brief procedure no more than one paragraph and data tables comparing the values used trying to match the step response and using the log decrement method and what you may think caused any differences.

**Submission:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- Date, To, From, Subject
- Written in first person from you
- Written with minimal spelling and grammar errors



- Purpose, procedure, results and conclusions of the laboratory experiment (the procedure should be very short, a high level summary of what you did for each part). The procedure should be two paragraphs at the most.
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text
- Also, you must to include a statement in your memo that this is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results

## ECE-205 Lab 4

### System Linearity

#### Overview

For many engineering systems, the system can be modeled as linear only over a specified region of operation. For example, the operational amplifier only operates when its linear region when the output is between the positive and negative supplies otherwise it is in saturation. In this lab you will construct the simple common emitter BJT amplifier shown in Figure 1. Then you will determine the range of input signals for which the circuit can be modeled as linear. Note that this is not a well-designed amplifier circuit, but it is easy enough for us to build and examine to learn about system linearity.

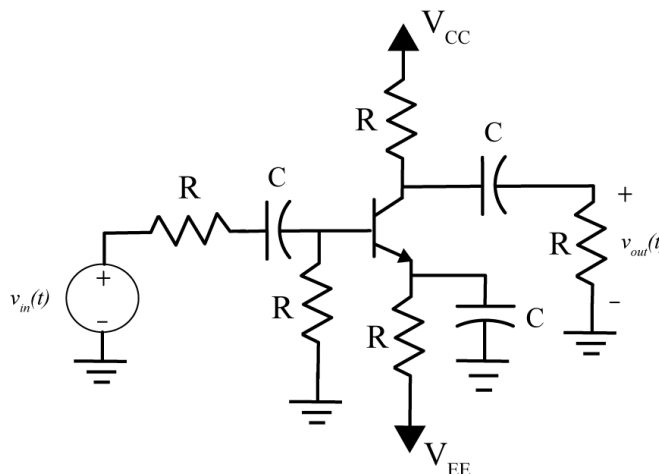


Figure 1: Simple common emitter amplifier.

#### Prelab

Read this entire lab procedure and background theory thoroughly. Then complete the following simulation using MultiSIM and submit it in class the day before the lab session.

1. Place the following components in the MultiSim drawing: BJT\_NPN\_VIRTUAL, CAPACITOR\_POL\_RATED, RESISTOR\_RATED, GROUND
2. You will need 4 - 1 kΩ resistors, 3 - 1 μF, 2 - voltage sources (15 V and -15 V)
3. Use the components to create the common emitter amplifier circuit shown in Figure 1.

## ***ECE-205 Lab 6***

### ***Transfer Functions and an Optical Transmitter and Receiver***

**Overview:**

*The purpose of this lab is to examine the transfer functions of systems and circuits in MultiSim, MATLAB and Simulink. Then, you will build an optical transmitter and receiver circuit to examine the input/output characteristics.*

**Equipment:**

IR LED Emitter

IR LED phototransistor receiver

51  $\Omega$  resistor

10 k $\Omega$  resistor

1 k $\Omega$  resistor

**Prelab:**

Read the background theory and entire lab procedure thoroughly. Then complete the following analytical calculations. Submit the prelab in class the day before the lab session.

1. Show that a first-order system represented by the following differential equation has the given transfer function,  $H(s)$ . *(Note: you will need to look up the Laplace transform of a derivative to do this. Remember to assume zero initial conditions.)*

$$\tau \dot{y}(t) + y(t) = Kx(t)$$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{K}{\tau s + 1}$$

2. Show that a second-order system represented by the following differential equation has the given transfer function,  $H(s)$ .

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2y(t) = K\omega_n^2x(t)$$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

3. Use the following components to build the circuit in Figure 1 in MultiSim (GROUND, DC\_POWER, RESISTOR\_RATED). Next, run the transfer function analysis in Multisim by clicking `Simulate` → `Analyses` → `Transfer Function...`. Remember you have to double click on the wires to name them  $V_{in}$  and  $V_{out}$ . The input should be the DC voltage source,  $V1$ , and the output is the voltage across  $R2$ ,  $V_{out}$ . What were the results? What do you think these three values represent? You should submit the system output, **not the Multisim file**, in your prelab submission.

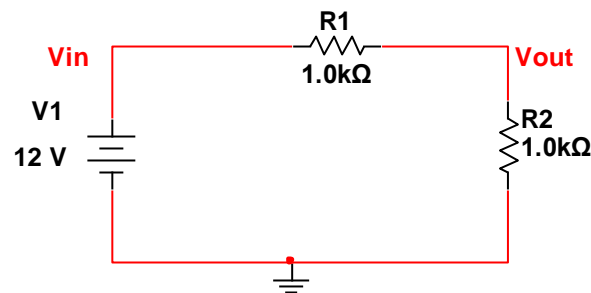
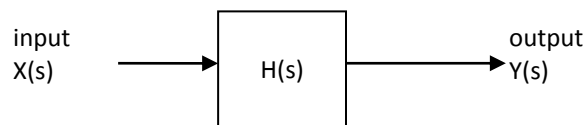


Figure 1: MultiSim Circuit for Transfer Function Analysis

### Theory:

In all of the prior labs, systems were represented in terms of the differential equation. However, it is also useful to examine the frequency characteristics of a system. The transfer function is used to represent a system based upon the frequency domain. The **transfer function** of a system is defined as the  $s$ -domain ratio of the Laplace transform of the output (response) to the Laplace transform of the input (source). To find the transfer function of a system, all of the **initial conditions must be zero**. In order to determine the transfer function of a system, apply the following steps:

- 1) Assume that the system has zero initial conditions.
- 2) If the input is  $x(t)$  and the output is  $y(t)$ , apply the Laplace transform,  $\mathcal{L}\{x(t)\} = X(s)$  and  $\mathcal{L}\{y(t)\} = Y(s)$
- 3) The transfer function is the ratio of the output to the input,  **$H(s) = Y(s)/X(s)$**



One benefit of using Laplace transforms to analyze systems is that you can use algebra to solve for system parameters instead of differential equations. Note that this is similar to phasor analysis except the system may not be at sinusoidal steady-state and the complex frequency is  $s = \sigma + j\omega$  instead of  $\omega$ .



In the time domain, the system output is found from the convolution,  $\mathbf{y}(t) = \mathbf{h}(t) * \mathbf{x}(t)$

In the frequency domain, the system output is found from the product,  $\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{X}(s)$

### **Procedure:**

#### **PART I: Transfer Functions in MATLAB**

In MATLAB a polynomial such as  $p(s) = 3s^2 + 2s + 1$  is represented by the following syntax,

```
p = [3 2 1];
```

Note that the semi-colon suppresses printing the output to the command window. The array indicates that the highest power of  $s$  is on the left and the lowest power,  $s^0$  is on the right. You should always right the array with respect to the coefficients for descending powers of  $s$ . There must be a number for each power of  $s$  so if one is missing then use a zero.

For example, to enter the polynomial  $p(s) = 2s^4 + 3s$  into MATLAB, you would type,

```
p = [2 0 0 3 0];
```

Typically transfer functions are the ratio of two polynomials and they can be created by using the **tf** command.

For example to create the transfer function,  $H(s) = \frac{s+1}{3s^2+2s+1}$ , enter the following code into

MATLAB

```
num = [1 1];%numerator polynomial of the transfer function
den = [3 2 1];%denominator polynomial of the transfer function
H = tf(num,den);%construct the transfer function
```

This could also be reduced to one command by writing the following:

```
H = tf([1 1],[3 2 1]);%system transfer function
```

If you do not include the semicolon at the end of the commands, MATLAB will print the transfer function to the workspace. Please do this to confirm that the transfer function has been entered correctly.

#### **PART II: System Response in MATLAB**

In this section, you will use MATLAB to determine the response of a system represented by a transfer function. The input may be a step or an arbitrary input.

1. Create a new MATLAB m-file (script) file [Ctrl+N] and place it in a lab 6 folder.



2. Insert a header comment by using the % that includes the filename, your name, date created and a brief description of what the program does, see the following example:

```
%{
Lab6.m
Uhura Jones
12/20/55
Transfer function analysis
%}
```

3. Add the following command line to clear all variables

```
clear variables;
```

4. Create the time constant and static gain variables and give them the following values,

```
tau = 0.001;
K = 2.0;
```

5. Use the steps from PART I to create the plant transfer function,  $G_p = \frac{K}{\tau s + 1}$
6. Set the final time variable to `Tf = 10*tau;` (be sure to use at least one capitol letter so it is not confused with the transfer function command `tf`).
7. Use the `linspace` command to create a time vector `t` from `0` to `Tf` with 1000 sample points. If you don't recall how to do this type, `help linspace` in the MATLAB command window.
8. Use the following command to create a step input,  $x(t) = Au(t)$  with an amplitude of 0.1.

```
A = 0.1;
x = A*ones(1,length(t)); %step function with amplitude, A
```

9. Next, use the `lsim` command to simulate the system output.

```
y = lsim(Gp,x,t);
```

10. Finally, create a plot of the output  $y(t)$  as a function of time, it should be similar to Figure 2. If you don't recall how to do this, please type `help plot` in the MATLAB command window. Make sure that your plot has a title with your initials and the axis labeled.

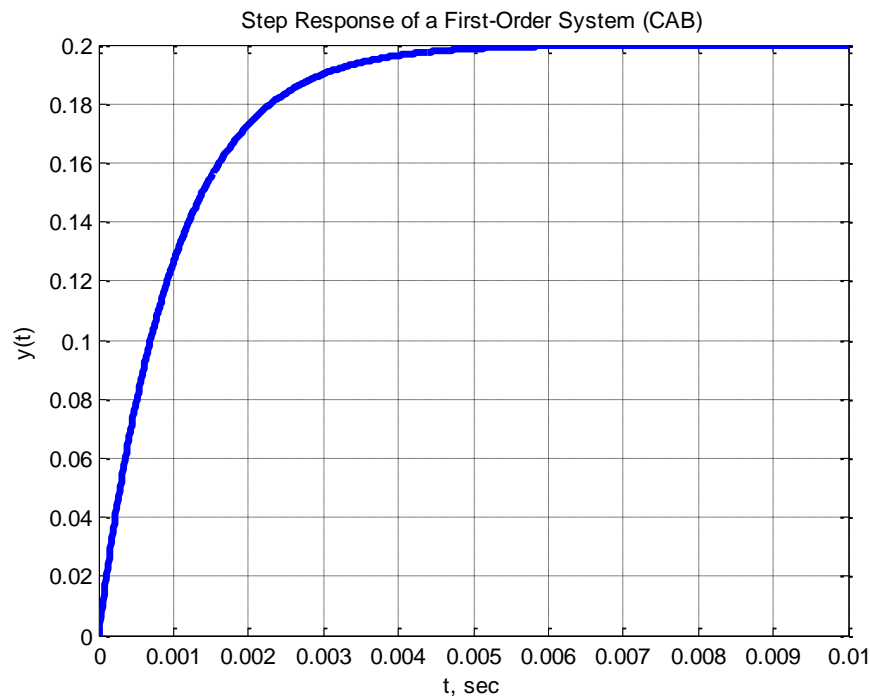


Figure 2: Step Response of a First-Order System

PART III: System Response in Simulink

1. In order to use Simulink to determine the response of a system given the transfer function it is necessary to put the following time and input values from MATLAB in the script file,

```
ut = [ t' x'];
```

2. In order to extract the numerator and denominator of a transfer function from a given transfer function, use the following

```
[num_Gp,den_Gp] = tfdata(Gp,'v');
```

3. Start `simulink` from the MATLAB command window and create a new model file [Ctrl+N] and save the model file as **openLoop.mdl**.
4. Construct the model file shown in Figure 3. You will need to look in the **Sources Library** for the **from workspace** block and the **clock** block
5. Go to the **Sink Library** for the **to workspace** blocks. Make sure to insert two of them and double-click on them and make the save format as **array**.
6. Go to the **Continuous Library** for the **transfer function** block. Click on the transfer function block to enter the numerator as **num\_Gp** and the denominator as **den\_Gp**.



7. Change the final time of the simulation to **Tf** as shown in Figure 3.

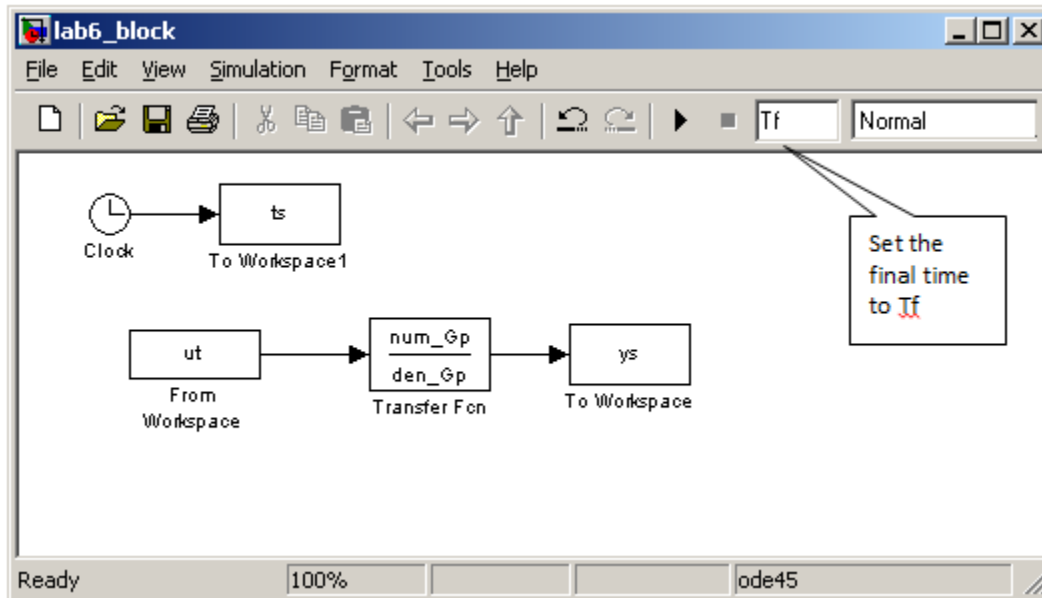


Figure 3: Open Loop Simulink model file

8. Run the simulation from the m-file using the following command,
- ```
sim('openLoop');
```
9. Modify the m-file (script) file to plot both the results from both MATLAB ( $t$ ,  $y$ ) and SIMULINK ( $t_s$ ,  $y_s$ ) on the same graph. Make sure that you include a descriptive title with your initials, label the x and y axes, include a legend and use two different line types. Your final graph should look similar to Figure 4. You need to include your graph in your memo, but not your code. The m-file should be included in your lab submission as a separate file.
10. Now create a new m-file (script) to plot the MATLAB and Simulink results for a second order system. You should start the new code by copying and pasting everything from the file created for the first order system and then modify it.
11. Create new variables for the natural frequency,  $\omega_n$ , and damping ratio,  $\zeta$ . The system has a natural frequency of 2000 rad/sec, a damping ratio of 0.2, and a static gain of 1.5. The step input amplitude is still  $A = 0.1$ . If you don't remember how to create the second order transfer function, review the prelab and PART I. Simulate the second order system and plot the results. If you have done everything correctly, you should get a graph like that shown in

Figure 5. You need to include your graph in your memo, but not your code. The m-file should be included in your lab submission as a separate file.

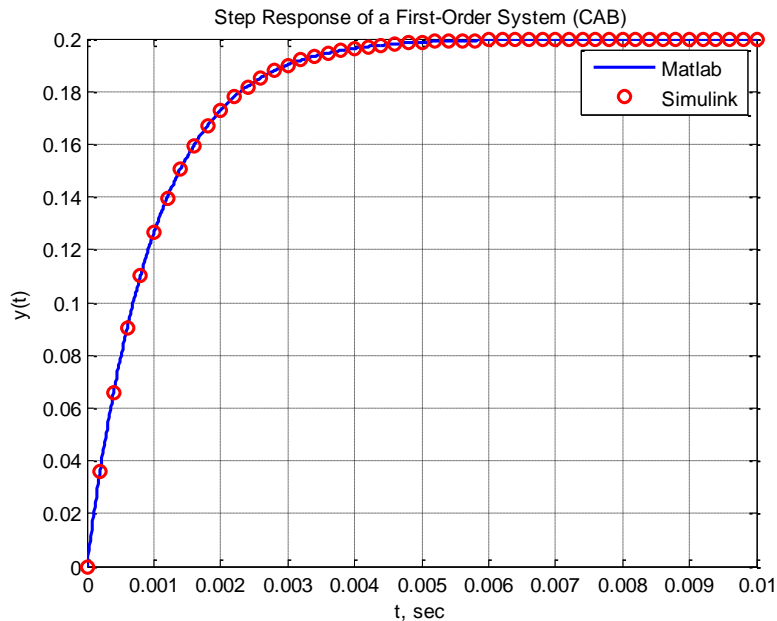


Figure 4: Matlab and Simulink first-order system results

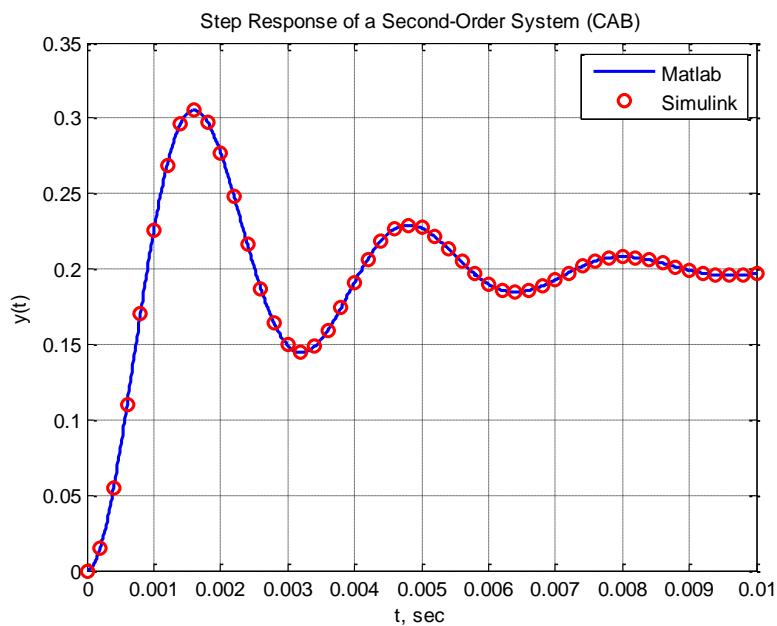


Figure 5: MATLAB and Simulink second order system results

PART IV: Building an Optical Transmitter and Receiver

In this part of the lab, you will build a simple optical transmitter and receiver that uses infrared light. This system is similar to how a TV remote control works. For the most part this should be fun, but we will also get practice wiring and using a TL072 chip. The transmitter and receiver are shown schematically in Figure 6. Do not start to build this circuit yet!

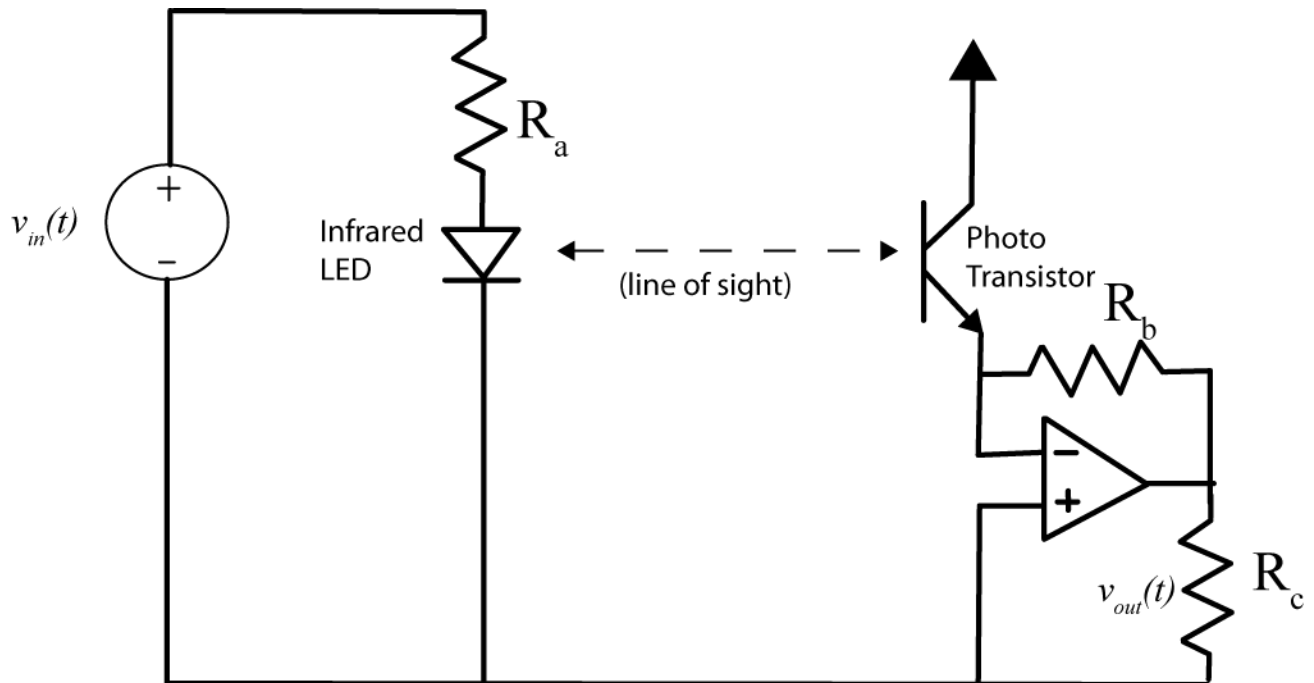


Figure 6: General schematic of an optical transmitter and receiver.

The transmitter on the left side of the schematic consists of an infrared LED. You will not be able to see when this is on, though you may be able to see it if you have a camera on your cell phone and look through that. The right side of the diagram shows the receiver, which consists primarily of a photo transistor and an amplifier.

1. You will create power and ground busses on your breadboard before you start building the circuit. Lay out your breadboard horizontal with the red stripe at the top and the blue stripe at the bottom. Use a wire to connect the two blue lines together which represent the common ground buss. Use a wire to connect the two blue lines for the ground. The top red line will serve as the positive (+Vcc) voltage rail, and the bottom red line will serve as the negative (-Vcc) voltage rail. Figure 6 provides a more detailed schematic of the optical transmitter/receiver circuit and you should refer to it as you build your system.

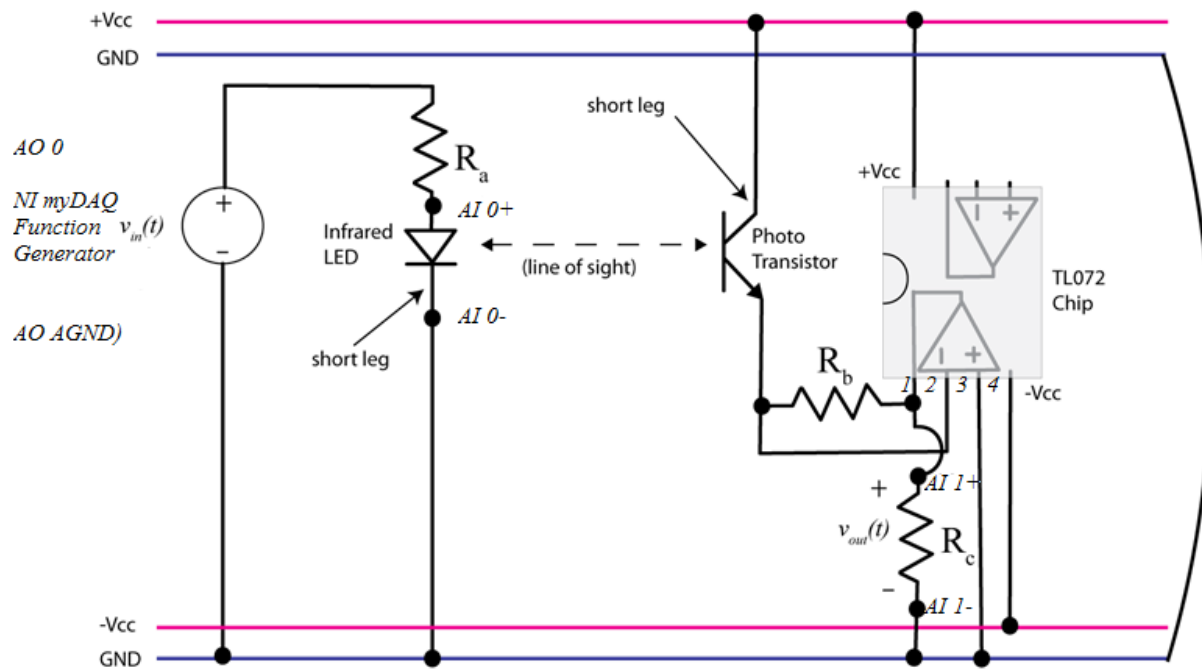


Figure 6: More detailed schematic of the optical transmitter and receiver

2. First, we will build the transmitter on the left of Figure 6. Start building as far as possible to the left side of the breadboard. Get the infrared LED (the clear LED) out of your kit and put it on the breadboard with the short leg on the ground buss. The short leg or the flat part of the rim of the cap on the LED represents ground.
3. Put the resistor,  $R_a = 51 \Omega$ , in series with the infrared LED and connect the MyDAQ function generator (**AO 0, AO AGND**) as the input between the resistor and ground. Make sure that **AO AGND** is connected to the ground buss.
4. Finally, connect Channel 0 (**AI 0+ AI 0-**) of the oscilloscope to measure the voltage across the infrared LED. Make sure that **AI 0-** is connected to the ground buss. The transmitter portion of the circuit is complete.
5. Next we will build the receiver, which should be built as far as possible to the right on the breadboard. Note that we are using a TL072 chip which includes two op amps on one chip. Be sure that the dot on the chip is to the left (the semicircle on the drawing). The pin out for the TL072 dual op amp chip is shown in Figure 7.

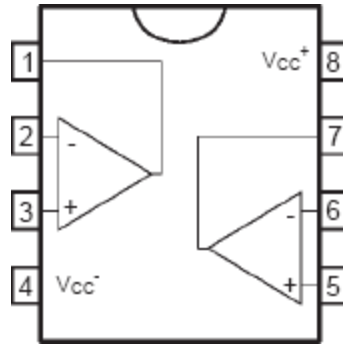


Figure 7: TL072 Dual Op Amp Pin Out

6. We will use the op amp on pins 1, 2, and 3 so connect pin 3 to **ground**. Connect pin 8 to the **+15V** or the red ground buss across the top of the breadboard. Connect pin 4 to the **-15 V** or the red buss across the bottom of the breadboard.
7. Connect the output of the op amp (**pin 1**) to  $R_c = 1 \text{ k}\Omega$  and then connect  $R_c$  to ground.
8. Also connect  $R_b = 10 \text{ k}\Omega$  to the output of the op amp and it goes to the photo transistor receiver. The photo transistor receiver is black and looks similar to the infrared LED. The short leg of the photo transistor or the flat side of rim of the cap should be connected to the **+15 V** buss. Lastly, connect a wire from **pin 2** to the top of the phototransistor. Thus, the input to the op amp is the photo transistor and the op amp serves as a voltage follower and the measured signal of the photo transistor receiver is measured across resistor,  $R_c$ .
9. Finally, the output of the receiver circuit is the voltage across the output resistor,  $R_c$ , so connect Channel 1 of the oscilloscope (**AI 1+**, **AI 1-**) across the resistor. Make sure that **AI 1-** is connected the ground buss.
10. The transmitter is at the end or very tip of the LED, and the receiver is at the end or very tip of the phototransistor, so you need to bend these at a 90 degree angle so they are pointing toward each other. You will probably have to tweak the positions once you start sending the signal to adequately measure the receiving signal.
11. Now connect the (**+15V**, **-15V**) power supply to your circuit. Make sure that you put **+15V** on the top red buss and **-15V** on the bottom red buss and **AGND** on the blue ground buss.
12. Connect the NI MyDAQ to your laptop and open up the function generator and oscilloscope instruments.



13. The infrared LED requires a 5V signal so set the function generator to a **5 kHz square wave** with an offset of **2.5 volts** and a peak to peak value of **5 volts**.
14. Set the oscilloscope to measure **500 mV/Div** for both channels. Set the source on channel 0 to **AI 0** and the source on Channel 1 to **AI 1**. The time/div should be set to **100  $\mu$ s**. Check enabled on both of the channels.
15. Press run on the function generator and the oscilloscope and your signals should look similar to Figure 8. If they do not then try adjusting the transmitter and receiver so they are pointing more directly at each other.
16. These signals indicate that the LED is transmitting when the voltage across it is high, and the phototransistor is receiving when the voltage across the output/load resistor is low. Set the trigger to **Edge** and the source is **Chan 0** and increase the level to get a stable output. Capture a screen shot of the oscilloscope image only for inclusion in your lab submission.
17. You have now finished with lab 6 and should read the submission requirements before submitting your lab.

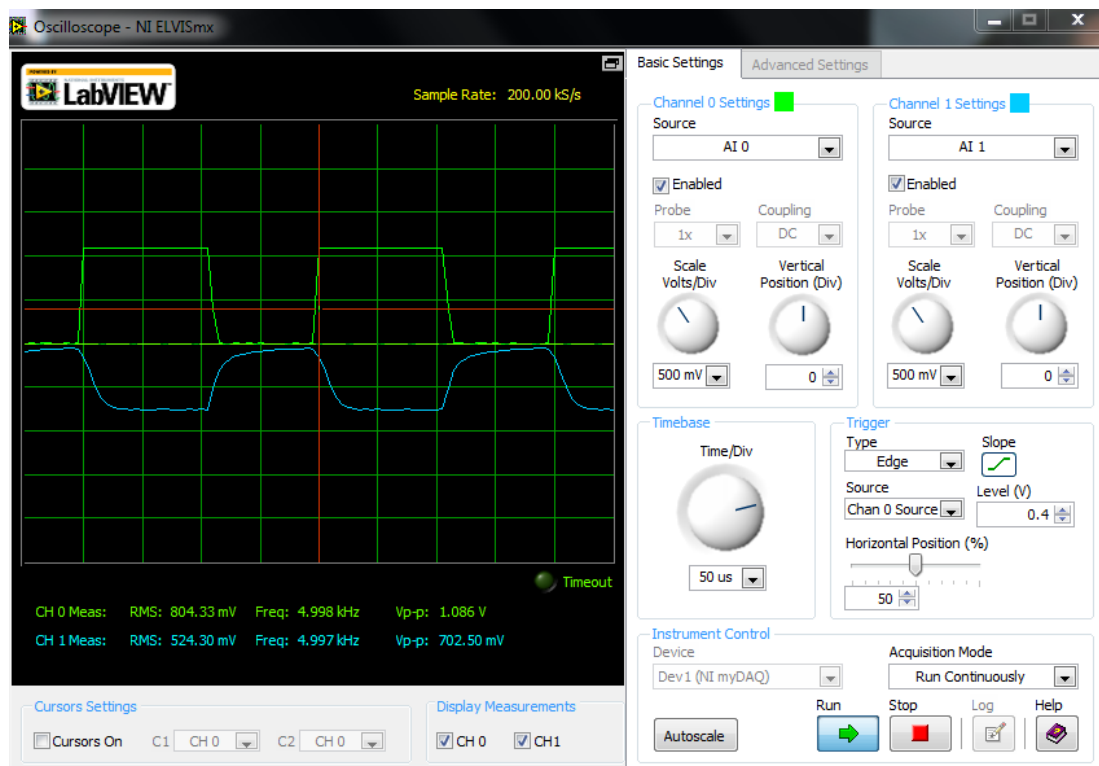


Figure 8: Transmitter and Receiver Signals

**Submission Requirements:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- A header at the beginning with *Date, To, From, Subject*
- Written in first person from you
- Written with minimal spelling and grammar errors
- *Purpose, procedure, results and conclusions* of the laboratory experiment
  - The very first sentence is the purpose and must explain why you are doing the experiment
  - The *procedure* should be very short, a high level summary of what you did for each part. The *procedure* should be two paragraphs at the most.
  - The *results* should include the tables and figures and results of the data collected and your observations. The results cannot be pages of figures and tables only, it **must** include text that references the figures and explains what they are to the reader.
  - The *conclusions* must be the end and it should be a summary of what you did, what you learned, what you observed and recommendations
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text. **DO NOT SUBMIT SEPARATE FILES**, images must be pasted into the Word document at the appropriate place with respect to the text. **DO NOT** just insert a bunch of figures and tables at the end of the document.
- Also, you must include a statement at the end of your memo that states that this submission is your own work.



- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results. This discussion may be a part of the results and/or conclusions sections of the memo



4. Connect the NI ELVISmx Function Generator to the input of the circuit. Connect Channel 0 of the NI ELVISmx Oscilloscope to the input to the circuit. Connect Channel 1 of the NI ELVISmx Oscilloscope to the output of the circuit. Set the function generator to a 3 Vpp 100 Hz sine wave. Your result should look similar to Figure A1 when you are finished.

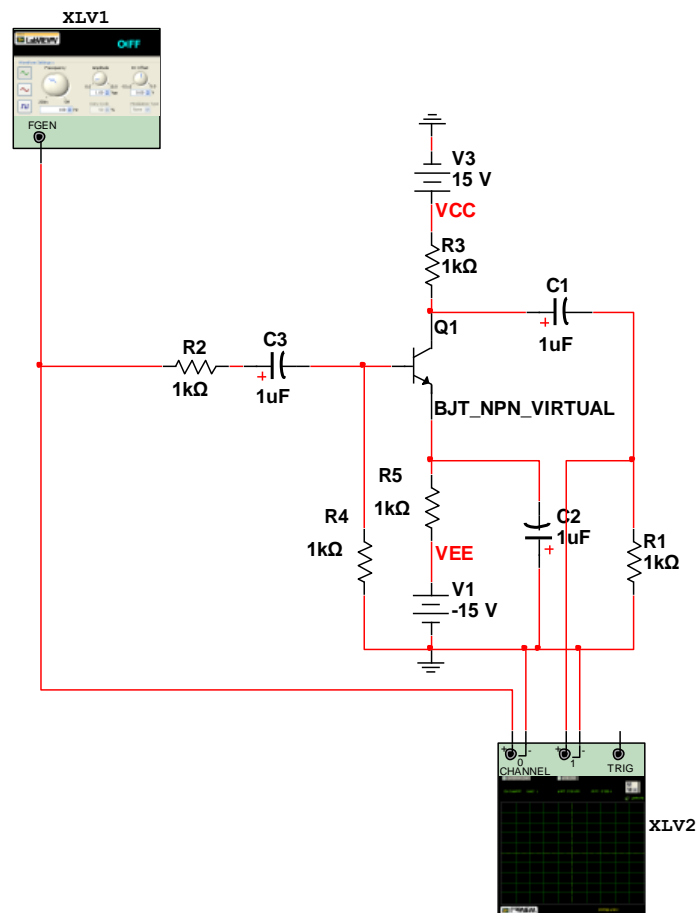


Figure A1: MultiSim Common Emitter Amplifier Circuit

5. Since in this lab you will examine the nonlinear characteristics of the common emitter amplifier, you will examine the input and output characteristics at two different input voltages. Make the waveforms as large as possible in the oscilloscope and capture the screen for inclusion in your prelab submission (see Figure A2). Using the maximum peak positive values of each waveform, what is the apparent gain (**output/input**) of the circuit?

6. Next, increase the input amplitude on the function generator until the output appears distorted. This represents the nonlinear characteristic of the amplifier circuit. Using the peak positive values of each waveform, what is the apparent gain? Is it the same as before?
7. Submit the MultiSim schematic and the screenshots before and after distortion for your prelab submission. Make sure the schematic has your name on it as a text box to indicate that it is your work.

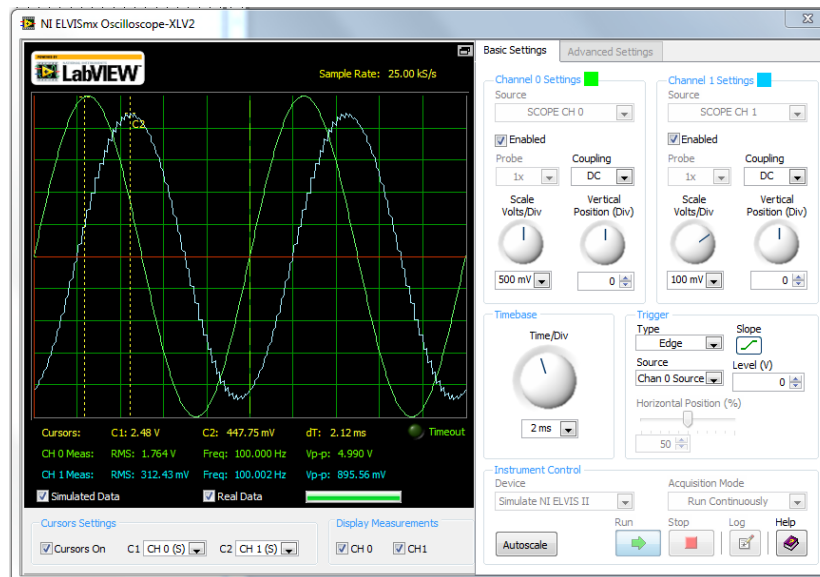


Figure A2: Common Emitter Amplifier Output

### Equipment

- 100  $\mu$ F electrolytic capacitors (x 3)
- 1 k $\Omega$  resistors (x 5)
- 2N-2222A-836 NPN BJT

### Procedure

#### **PART I: Build the common emitter circuit**

In this part you will build the circuit in two stages. DO NOT try to build the circuit all at once.

The electrolytic capacitors must be connected with the correct polarity, so be careful to read all of the instructions.

- 1) Set up the power rails on the breadboard. Use a wire from your kit to tie both of the blue buss (common) rails together and connect them to ground on the NI MyDAQ (AGND) next to the -15V terminal. Connect the red buss across the top to the +15V terminal on the NI MyDAQ. Connect the bottom red buss on the breadboard to the -15V terminal on the NI MyDAQ.
- 2) The NPN BJT is represented by the symbol shown in Figure 2, and it has three terminals. If you hold the BJT with the flat part to your right, then the collector is on top, the base is in the middle, and the emitter is on the bottom.

**INTERNAL SCHEMATIC DIAGRAM**

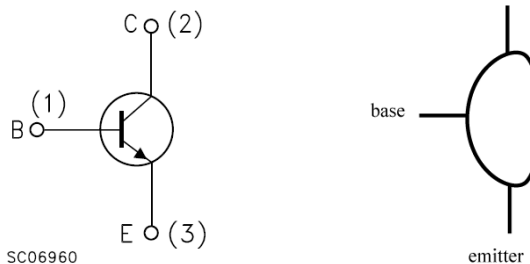


Figure 2: NPN BJT Internal schematic (left) and physical device (right)

- 3) Next, build the circuit subsystem shown in Figure 3. Note that the collector is connected to the +15 V source,  $V_{CC}$ , through a  $1\text{ k}\Omega$  resistor. The emitter is connected to the -15 V source,  $V_{EE}$ , through a  $1\text{ k}\Omega$  resistor. The base is connected through a resistor to ground. The collector is also connected through an electrolytic capacitor (note the negative sign!) and resistor to ground. There is a minus sign on the capacitor can and the shorter leg represents the negative terminal.

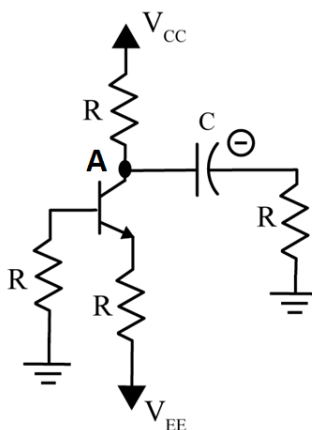


Figure 3: Biasing the transistor circuit.

- 4) To verify that the circuit is biased correctly, start the NI MyDAQ Digital Multimeter and set it to measure DC Voltage. Measure between node A and ground as shown on the circuit in Figure 3. If you have done everything correctly, this measurement should be approximately **3V**. **Do not move on until this is correct!**
- 5) Now finish building the circuit as shown in Figure 4. Verify that the polarity is correct on all of the capacitors. In addition, confirm that the remaining items are all connected to ground and not any of the voltage sources.

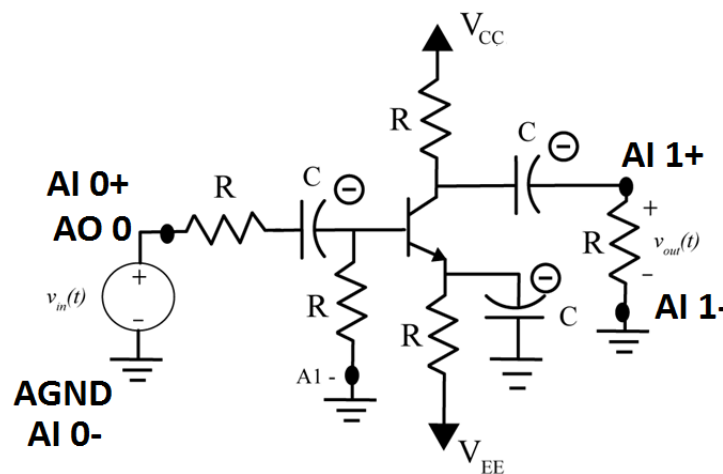


Figure 4: Common Emitter Amplifier

- 6) Set the function generator to 100 Hz, 1 Vpp sine wave and connect it to the input of the circuit,  $v_{in}(t)$ . Also connect channel 0 of the oscilloscope to the input of the circuit. Connect channel 1 of the oscilloscope to the output,  $v_{out}(t)$ . Set both channels of the oscilloscope to 500 mV/div and the timebase scale to 2 ms/div.
- 7) Use the cursors to measure the *peak positive amplitude* of the input and output signals (**do not** measure peak-to-peak!). The ratio of the output magnitude to the input magnitude is the gain and it should be between 2 and 4. If it is not, check your circuit and then ask for help. Now we are ready to make circuit measurements.



## PART II: Measuring Input, Output and Gain for the Amplifier

1. This circuit can only be modeled as a linear system for a limited range of input signals. In this part, you will measure the input and output amplitudes as the input amplitude changes. Assume that the input signal is a sine wave with a frequency of 100 Hz for all of the measurements. Make sure that the function generator has a 0 VDC offset before you start taking the measurements. Set the peak to peak voltage on the function generator to 200 mV.
2. Use the cursors on the oscilloscope to measure the input and output amplitude and the gain of the common emitter amplifier. Be sure to measure only the **maximum peak positive amplitudes** of the output signal, **do not** measure peak-to-peak amplitudes or RMS. Adjust the Volts/Div scale in order to make the waveform as large as possible on the screen so that your measurements are most accurate.
3. Measure the input amplitude, output amplitude and gain of the amplifier as the peak to peak voltage on the function generator is increased from 200 mV to 4.6 V in increments of 200 mV. *Note that as the input increases, the output will begin to look more distorted and less like a sinusoid due to saturation and nonlinearities.* However, continue to only measure the **maximum peak positive voltage** for the input and output using the cursors. Since you will need to plot this data in MATLAB in the next section, you should record the measured amplitudes in Excel and use a formula to calculate the gain as you increase the input.
4. You should also include **two screen captures** of the oscilloscope screen in your lab memo submission: one with the sinusoidal input and output and one that shows the nonlinearities in the output signal. Do not copy the entire window just the oscilloscope screen. Make sure the figures have a figure number and caption and are referenced in the text of the memo.

## PART III: Plotting the Amplifier Data

- a. Start MATLAB and create a new m-file (File->New->Script) [CTRL-N]. Give the file a name such as "Uhura\_Lab4.m".
- b. Use the % to add a comment at the top of the file with the filename, your name, a brief description of the program and the date.



- c. Create an array of the input amplitude data and an array of the output amplitude data in the m-file. You can do this by typing, **Vin = [**, then copy the column vector from Excel and then type, **];**. You should repeat this for the variable, **Vout**.
- d. Plot the output versus the input amplitudes by typing **plot(Vin, Vout, 'o')**. Make sure the graph has a grid, descriptive title and axis labels with units. Use the commands **grid**, **title()**, **xlabel()**, **ylabel()**, in order to do this. You should type **help** and the function name in the MATLAB command window if you are not sure how to do this. Your plot should be similar to Figure 5.

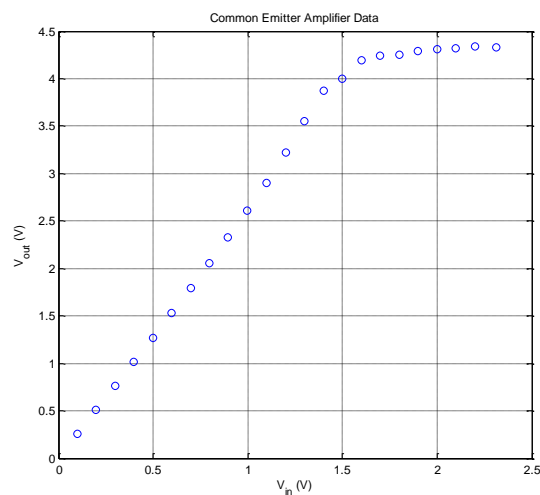


Figure 5: Emitter measured input/output data.

- e. Next, you will fit a least squares straight line approximation between the points that show a linear input/output relationship. Use the MATLAB command, **polyfit** and insert the following code into the m-file. Note that this code assumes that your input and output data arrays are named **Vin** and **Vout**. In addition, this code is written assuming that you will create a least squares line that fits the first **N data** points which has been selected to be 10:

```
N = 10; % select an N value close to the number of linear data points
p = polyfit(Vin(1:N),Vout(1:N),1);
m = p(1); b = p(2);
est_output = m*Vin+b;
plot(Vin,Vout,'o',Vin,est_output);grid;
```



- f. Your plot will look similar to Figure 6 and you should iteratively adjust the parameter **N** to include as many points necessary to generate a reasonably straight line. Use the MATLAB Edit -> **Copy Figure** command to make a copy of this plot for inclusion in your lab memo. . This figure should have a figure number and caption referenced in the text.

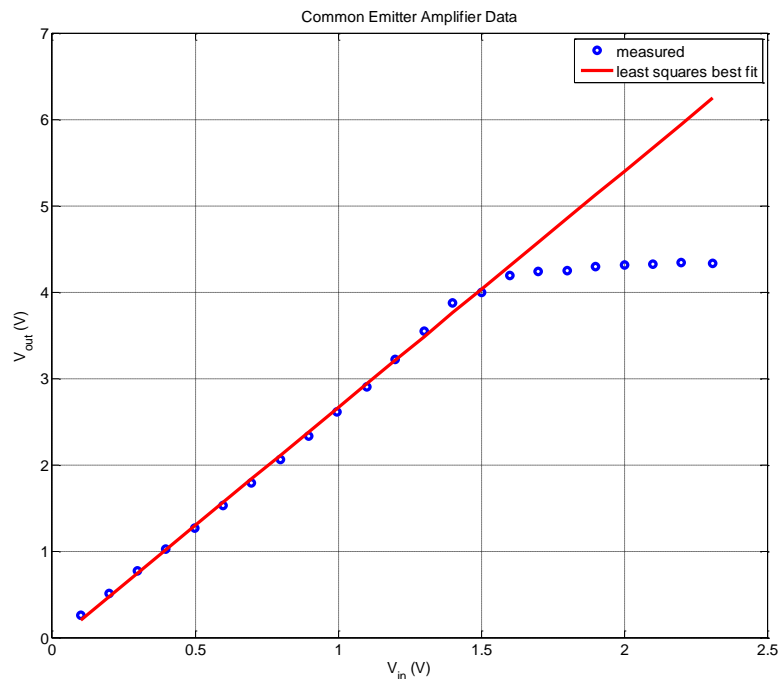


Figure 6: Measured data and least squares fit

- g. Copy your m-file and include it in the Appendix of your lab memo.

*Your memo should contain 3 graphs: one MATLAB graph and two screen captures. Your memo should also include your MATLAB m-file.*

#### Submission:

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- Date, To, From, Subject



- Written in first person from you
- Written with minimal spelling and grammar errors
- Purpose, procedure, results and conclusions of the laboratory experiment (the procedure should be very short, a high level summary of what you did for each part). The procedure should be two paragraphs at the most.
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text
- Also, you must to include a statement in your memo that this is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results



## ECE-205 Lab 7

### Simulation of Feedback Control Systems

**Overview:**

*In this lab you will be introduced to the idea of feedback control. We will simulate first and second order systems in a feedback loop with a controller. Next week we will build some of these systems so that you can see how we can relate the math and the block diagrams to a real system.*

**Equipment:**

MATLAB/Simulink

**Prelab:**

Read the background theory and entire lab procedure thoroughly. Then complete the following analytical calculations. Submit the prelab in class the day before the lab session.

1. Show that the feedback control system represented by the diagram in Figure 1 has the given transfer function  $G_o(s) = Y(s)/R(s)$ .

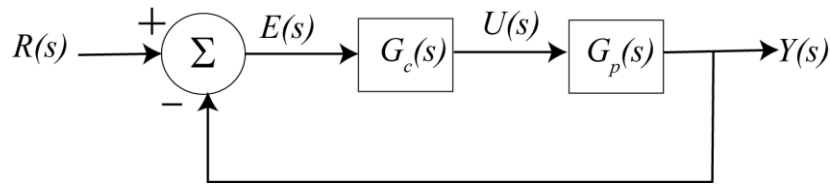


Figure 1: Basic closed loop system with unity feedback

$$G_o(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

2. For the feedback control system represented by the diagram in Figure 2, determine the transfer function,  $G_o(s) = Y(s)/R(s)$ . What do you think  $H(s)$  represents?

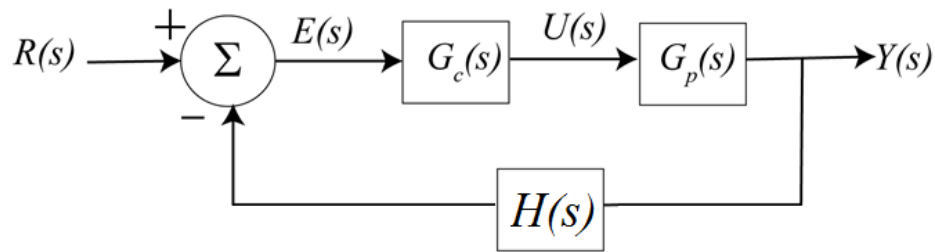


Figure 2: Feedback control system

**Theory:**

A **control system** is any device that you want to regulate or track at a certain **setpoint**. The human body is a control system that regulates our body temperature to 98.6 °F unless there is some type of disturbance such as trauma or illness. A car's cruise control is an example of a control system that tracks the speedometer set point to keep the car at a certain speed despite wind, rain or pot holes. When you drive a car, you act as a control system as you use your eyes to determine if you are driving straight and staying on the road and use the steering wheel to adjust the car. In your home, when you adjust the thermostat to a certain room temperature and the air conditioning or heater adjusts the blower or damper to meet that set point this is yet another control system.

When a system is set to a certain value (**setpoint**) and there is no measurement of the output to determine whether it actually achieved the set point, this is called **open loop control**. If you drive with your eyes closed that would be an example of open loop control. Open loop control is not a very good way to design a system unless the system is extremely reliable, predictable with no chance of failures. A **closed loop** control system or **feedback control** is when you use some type of sensor (such as your eyes, thermometer, encoder, speedometer) to determine how far the system is away from the set point (**error**) and use the **controller** to make the adjustments. The controller may be an electronic device, human, biological functions, etc. In these examples, the system being controlled is the **plant** (car speed, car position, body temperature, room temperature). A standard unity feedback control system with input,  $R(s)$ , output,  $Y(s)$ , controller,  $G_c(s)$  and plant,  $G_p(s)$ , and error,  $E(s)$ , is shown in Figure 1.



Sometimes we have a system (plant) and the response needs to be modified to meet certain design characteristics. For example, the settling time for a step input may be too long, percent overshoot may be too large or the output never reaches the value of the set point. While it is sometimes possible to directly modify the plant to achieve the design characteristics, such as changing the values of circuit elements to change the time constants, there are many times that the original system cannot be changed. This is the main focus of control systems analysis and design.

Assume that the plant,  $G_p(s)$ , for the systems we want to control can be modeled as the following first and second order system transfer functions.

$$G_p(s) = \frac{K}{\tau s + 1}, G_p(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

In order to change the behavior of the plant, we are going to do two things, both of which are depicted in Figure 1. The first of these is to provide some feedback so we can compare the input and output of the system. This difference is called the **error signal** and it is just the difference between the input and the output. In the diagram the error signal is depicted as  $E(s)$ , and it is the difference between the **reference signal**,  $R(s)$  and the output signal,  $Y(s)$ ,  $E(s) = R(s) - Y(s)$ . In most control systems we want the output to equal the reference signal. The second thing we will do is to include a new system called a **controller**, that we depict by the transfer function,  $G_c(s)$ , that operates on the error signal and produces a new signal, the **control effort**,  $U(s)$ , which is the input into the plant. Note that the control effort is **not** the transfer function of the unit step.

In this lab we will look at three common controllers, a proportional ( $P$ ) controller, and integral ( $I$ ) controller, and a proportional plus integral (PI) controller. A **proportional** controller is used to increase the gain of the system and reduce settling time but it may also increase oscillations or make the system go unstable. An **integral** controller is also called a reset controller because it can reduce the error between the system output and the set point to zero but it may also increase settling time. A **proportional-integral** controller attempts to use the characteristics of both the  $P$  and  $I$  controllers to adjust the system output.



For a **proportional** (P) controller, the signal to the plant,  $u(t)$ , is directly proportional to the error signal,  $e(t)$  thus,  $u(t) = k_p e(t)$ . In the Laplace domain, the control signal is represented as  $U(s) = k_p E(s)$ , and the transfer function for the proportional controller is then

$$G_c(s) = \frac{U(s)}{E(s)} = k_p.$$

For an **integral** (I) controller, the signal to the plant is proportional to the integral of the error signal thus,  $u(t) = k_i \int_0^t e(\lambda) d\lambda$ . In the Laplace domain this equation is  $U(s) = \frac{k_i}{s} E(s)$ , and

the transfer function for the integral controller becomes 
$$G_c(s) = \frac{U(s)}{E(s)} = \frac{k_i}{s}$$

For a **proportional plus integral** (PI) controller, the input to the plant is proportional to both the error and the integral of the error (with two scaling factors) thus

$$u(t) = k_p e(t) + k_i \int_0^t e(\lambda) d\lambda. \text{ In the Laplace domain this is}$$

$$U(s) = k_p E(s) + \frac{k_i}{s} E(s) = \frac{k_p s + k_i}{s} E(s) = \frac{k_p (s + k_i / k_p)}{s} E(s)$$

We often just write the transfer function for the PI controller as 
$$G_c(s) = \frac{U(s)}{E(s)} = \frac{k(s+z)}{s}$$

In this lab you will use all three controller types to adjust system characteristics to meet the design specifications.

### **Procedure:**

#### **PART I: Simulating Feedback Systems in MATLAB**

In this part we will show you how to simulate closed loop systems using both MATLAB and Simulink.

1. Copy the first MATLAB script files (m-file) and the Simulink model file (mdl-file) that you created for Lab 6 into a Lab 7 folder.



2. Rename the script file to something such as *Lab7(YourInitials).m*. Rename the model file to *closedLoop.mdl*.
3. Use `%{` and `%}` to create a header comment with the filename, your name, date created and a brief description of the file.
4. Use the `%` symbol to comment out the `sim('openloop')` command, since we will not be using Simulink for a while.
5. Confirm that the `clear variables;` command is at the beginning of the script file.
6. Confirm that the variables are still set to `tau = 0.001;` and `K = 2.0;`
7. Enter the first order transfer function,  $G_p = \frac{K}{\tau s + 1}$  into the script file. It may already be there from last week but if not and you don't remember how to create a transfer function, please refer to Lab 6.
8. Set the final time variable `Tf = 0.01` (be sure to use at least one capital letter so it is not confused with the transfer function command `tf`).
9. Modify the `linspace` command to create a time vector `t` from `0` to `Tf` with 1000 sample points.
10. The system will have a step input with amplitude `0.1` and the following commands will generate the input. Now add the input, `r`, for the closed loop system. **Note that the input for the open loop system is `x` and it should also be in the script!**

```
A = 0.1;
```

```
r=A*ones(1,length(t));
```

11. To create the proportional controller, define the gain, `Kp = 0.2;` and then enter the transfer function into MATLAB:

```
Gc = tf(Kp,1);
```

12. Next, construct the closed loop transfer function by entering the following command into the script. Note that the `minreal` command removes pole/zero cancellations.

```
Go = minreal(Gc*Gp/(1+Gc*Gp))
```

13. To simulate the system, we then use the `lsim` command as follows:

```
yo = lsim(Go,r,t);
```



14. Next plot the open loop system you created last week and the closed loop system with the proportional controller by typing the following command.

```
subplot(121)

plot(t, y, 'Linewidth',3);grid;

title('Open Loop Response');

xlabel('t, sec');

ylabel('y(t)');

axis([0 Tf 0 1.1*max(y)]);

subplot(122)

plot(t, yo, 'Linewidth',3);grid;

title(['Closed Loop Response, Kp = ', num2str(Kp)]);

xlabel('t, sec');

ylabel('y_o(t)');

axis([0 Tf 0 1.1*max(yo)]);
```

15. If you now run the script, the output should look similar to Figure 2.

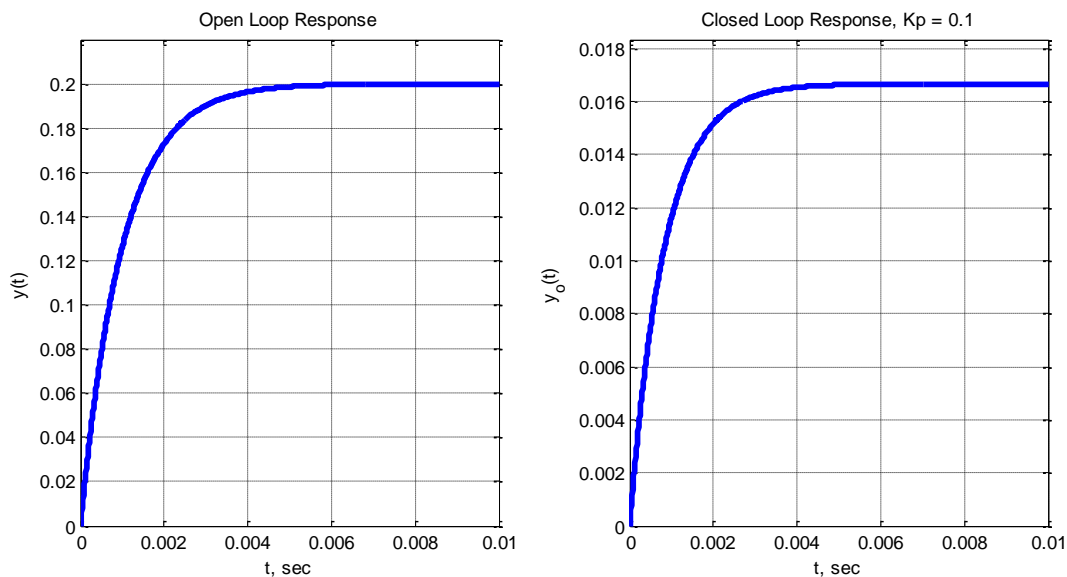


Figure 2: Open Loop and Closed Loop Response of a First Order System

16. If you compare these two figures, you should note two things:

- The settling time is much faster for the system with feedback



- The steady state values are different for the two systems, but neither one of them actually matches the reference input signal (the reference signal in this case is 0.1)

17. Do not save the figures for inclusion in your lab memo yet, go on to PART II.

PART II: Proportional Control

1. Modify your controller so that  $k_p = 1$  and generate the output again with the open loop and closed loop system. Examine the output and record the observations in your lab memo. What is the settling time? What is the output amplitude? Calculate the steady **state error = input - output**.
2. Then change the proportional controller gain to  $k_p = 5$ . Examine the output and record the observations in your lab memo. What is the settling time? What is the output amplitude? Calculate the steady state error = input - output. How do variations in the proportional controller gain affect the system output?

PART III: Integral Control

1. Now change final time to  $T_f = 0.02$ ; seconds and change the proportional controller to the integral controller,  $G_c(s) = \frac{200}{s}$ . What do you observe about the settling time and steady-state error? Record your observations in the lab memo submission.
2. Next change the integral controller to  $G_c(s) = \frac{500}{s}$ . What do you observe about the settling time and steady-state error? Record your observations in the lab memo submission.
3. You should note that the settling time is now larger, the reference input ( $A = 0.1$ ) is matched perfectly. However, as we increase the integral gain,  $K_i$ , the percent overshoot increases. The correct response for the second controller is shown in Figure 3.

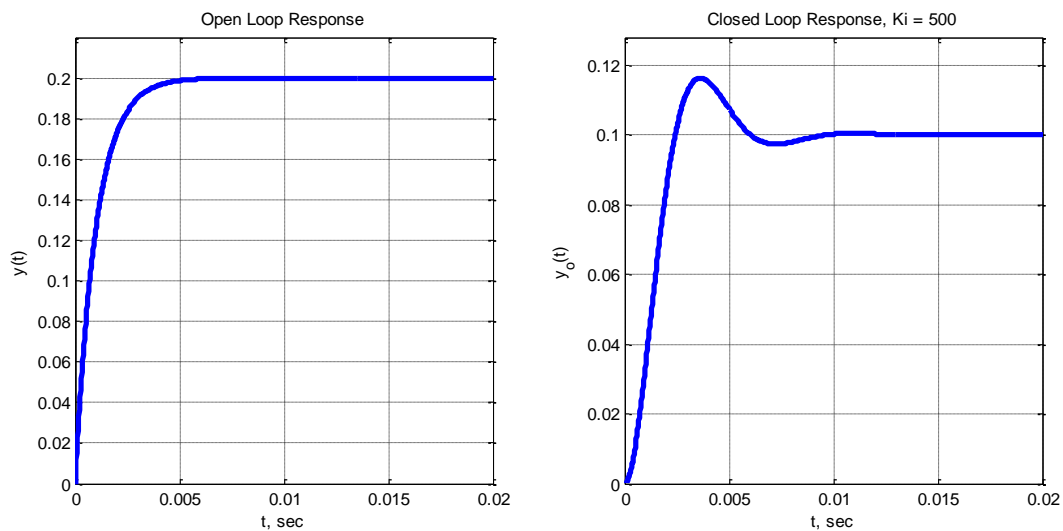


Figure 3: Integral Control

PART IV: Proportional-Integral Control

- Next change final time to  $T_f = 5e-5$ ; and simulate the system for the following proportional plus integral controller,  $G_c(s) = \frac{100(s+10)}{s}$ . What do you observe about the settling time and steady-state error? Record your observations in the lab memo submission.
- Next change the PI controller to  $G_c(s) = \frac{500(s+10)}{s}$ . What do you observe about the settling time and steady-state error? Record your observations in the lab memo submission.
- You should observe that the settling time is shorter and we also match the reference input. The correct response for the first controller is shown in Figure 4.



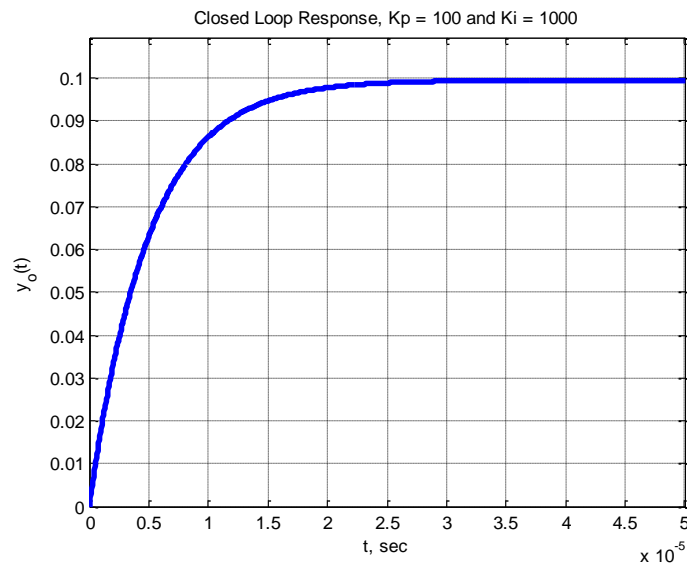


Figure 4: Proportional-Integral (PI) Control

PART V: Simulating Feedback Systems with Simulink

1. Open the Simulink model that you name **closedLoop.mdl**.
2. Modify your model so it looks like that in Figure 5.

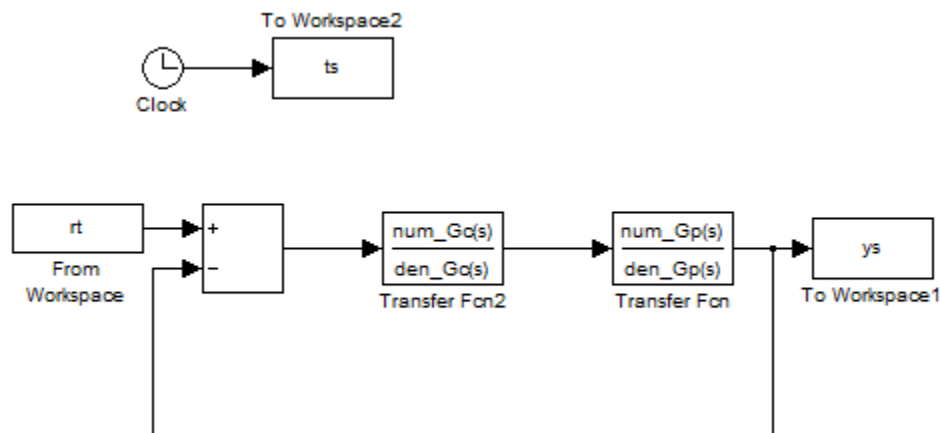


Figure 6: Closed loop Simulink model file.

3. Since we are dealing with small time scales, we will need to modify the maximum default time step in Simulink. To do this click Simulation  $\rightarrow$  Configuration Parameters. Under the **Solver** change the **Max step size** from auto to to **1e-6**. The next two figures show the basic idea. Save the model file when you are done (see Figure 7).

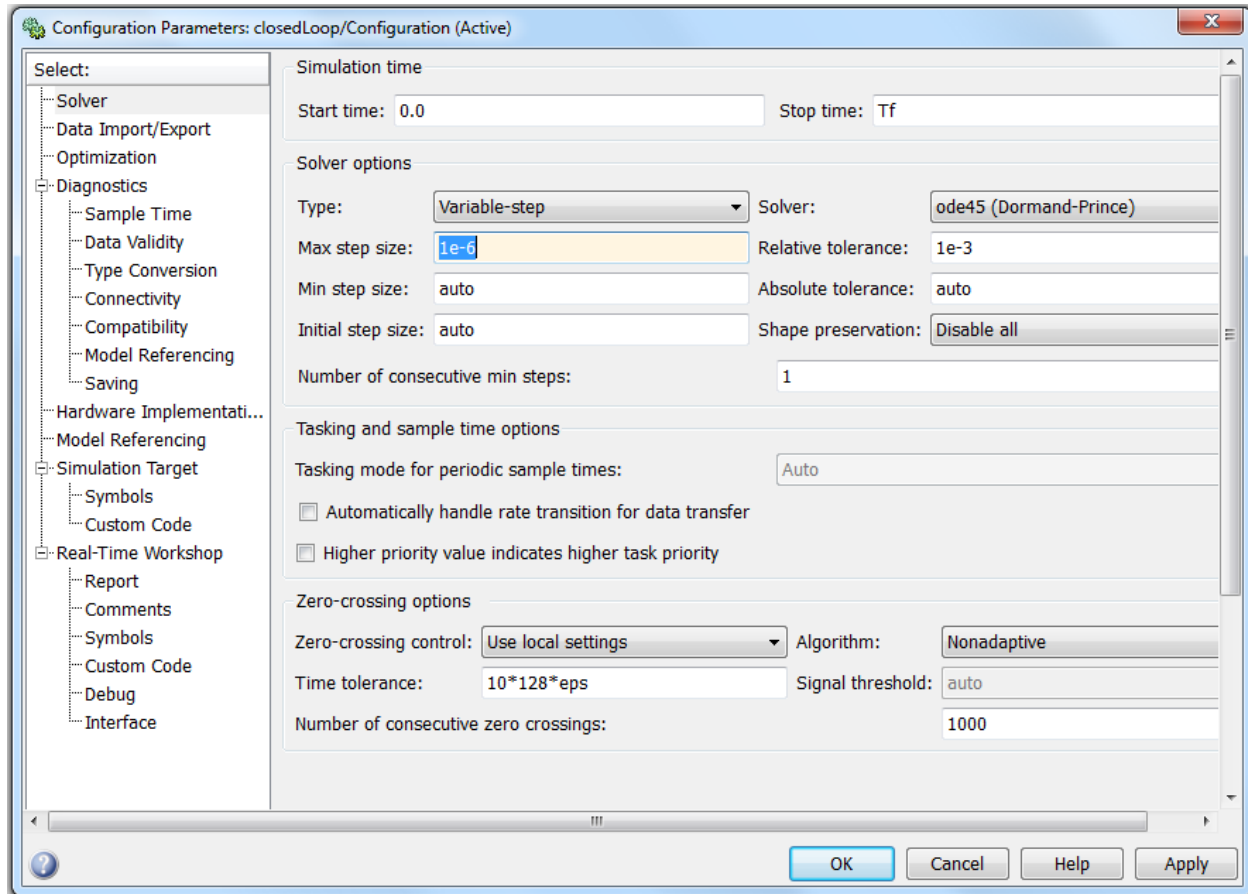


Figure 7: Simulation Configuration Parameters

- We will use the same time and input values from the MATLAB simulation so type the following command in the script file:

```
rt = [ t' r' ];
```

- Use the following command to extract the numerator and denominator of the controller:

```
[num_Gc,den_Gc] = tfdata(Gc,'v');
```

- Run the simulation from the m-file by using the following command:

```
sim('closedloop');
```

- At this point you have results from both Matlab (t, yo) and Simulink (ts, ys). Modify your m-file to plot both of these results on the same graph. The plot should have two

different line types and colors, grid, the axes labeled and a descriptive title. See Figure 8 for an example of the system output.

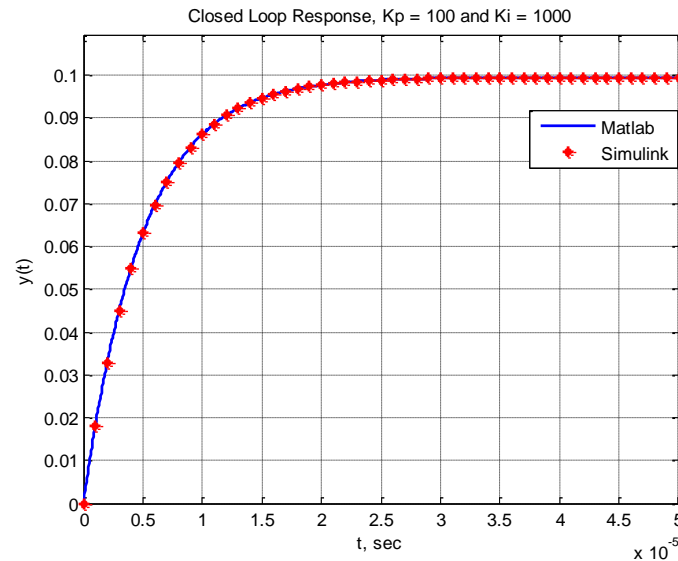


Figure 8: MATLAB and Simulink Results for closed loop control

- Now, generate the MATLAB and Simulink results for each of the controllers in PARTS II, III and IV. Make sure each plot has a descriptive title with the controller gains similar to Figure 8, axes labeled, grid, and distinct markers for MATLAB and Simulink results. You must include all six graphs in your lab memo submission.

PART VI: Second-order system closed loop control

- Modify the script (m-file) so that you plot the closed loop response for the second order system. Similar to Lab 6, the system parameters are a natural frequency of **2000** rad/sec, a damping ratio of **0.2**, and a static gain of **2.0**. Set the final time to **0.02** seconds.
- Simulate your system in both Matlab and Simulink using the integral controller with a gain of **250**. Your result should look like similar to Figure 9. Note that the steady-state output is the same as the reference input.

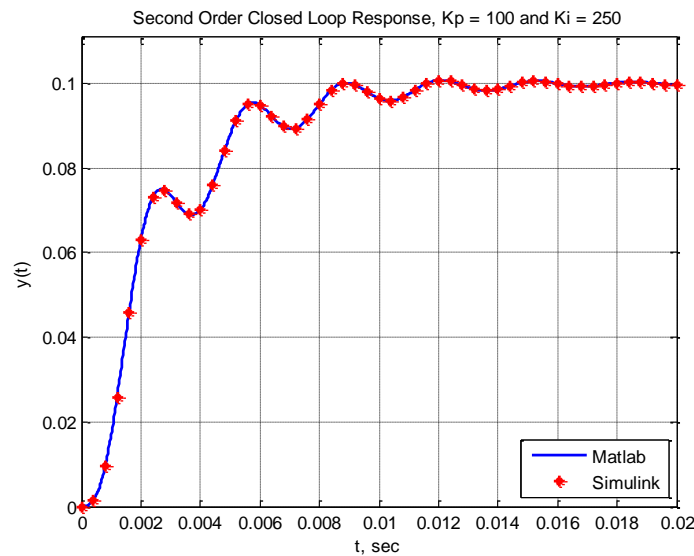


Figure 9: Second order closed loop response

3. You need to include your graph in your memo. It should have a descriptive title, axes labeled, grid on, line markers and a legend and be referenced in the text.

PART VII: Prefilters for Modifying Steady State Values

Sometimes when we use feedback control, we need to be able to adjust the steady state output value. This is very apparent in our first order system when we used a proportional controller. In that case, we were able to significantly change the settling time of the system, but the output did not equal the input. What we need is a proper scaling, and this is usually called a **prefilter**. Figure 10 shows our basic feedback loop with a prefilter. The prefilter can either be a transfer function or a constant. In this lab we will assume the prefilter is a constant, so  $G_{pf}(s) = G_{pf}$ . The purpose of the prefilter is to take the reference signal and scale it so that the input to the system matches the output of the system.

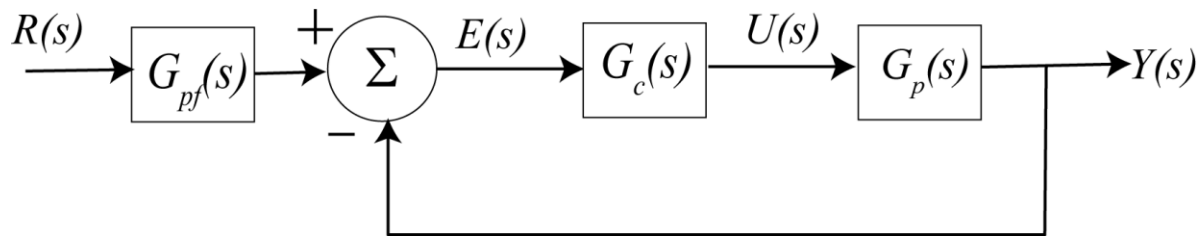


Figure 10: Basic feedback control loop with a prefilter



If the closed loop system has the transfer function,  $G_o(s) = \frac{N_o(s)}{D_o(s)}$  then we want the prefilter to

have the following value,  $G_{pf} = \frac{D_o(0)}{N_o(0)}$ . In class, you will learn why you need to use this value.

1. Modify the **closedLoop.mdl** Simulink model you created to match Figure 10 by adding a transfer function block in series with the input `rt`. The numerator is `num_Gpf` and the denominator is `den_Gpf`.
2. In order to determine the value of the pre-filter for zero steady state error, you need to evaluate a polynomial at zero. There are two ways to evaluate a polynomial at zero, but the easiest way to do this is to use the **end** parameter, as follows:

```
num_Gpf = den_Go(end);
```

```
den_Gpf = num_Go(end);
```

3. To implement the dynamic prefilter in both your Simulink model and Matlab models, use the following code to redefine `Go`:

```
Go = Go*num_Gpf/den_Gpf;
```

Insert these lines of code right after you use `tfdata` to extract the numerator and denominator of the closed loop transfer function and before you create `yo`.

4. Rerun **Parts I and II**, for the first order system with  $Kp = 0.2, 1,$  and  $5$  with a final time of  $0.005$  s. Put all three of the plots on a  $1 \times 3$  grid by using a `subplot` command. Figure 11 shows the beginning of the plots.
5. What the difference between these results and those in parts I and II?

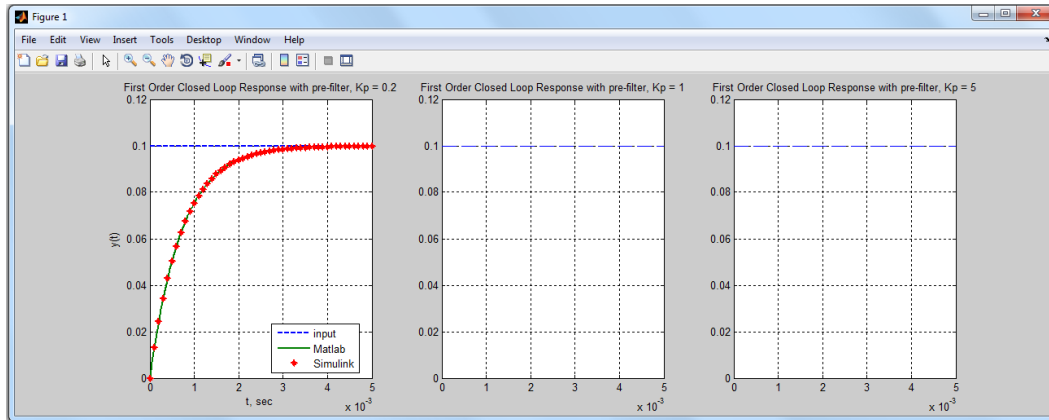


Figure 11: Closed loop control of a first order system using a prefilter

Submit this plot, the MATLAB m-file and the Simulink model file with your lab memo to the Angel Drop Box.

### Submission Requirements:

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- A header at the beginning with *Date, To, From, Subject*
- Written in first person from you
- Written with minimal spelling and grammar errors
- *Purpose, procedure, results and conclusions* of the laboratory experiment
  - The very first sentence is the purpose and must explain why you are doing the experiment
  - The *procedure* should be very short, a high level summary of what you did for each part. The *procedure* should be two paragraphs at the most.



- The *results* should include the tables and figures and results of the data collected and your observations. The results cannot be pages of figures and tables only, it **must** include text that references the figures and explains what they are to the reader.
- The *conclusions* must be the end and it should be a summary of what you did, what you learned, what you observed and recommendations
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text. **DO NOT SUBMIT SEPARATE FILES**, images must be pasted into the Word document at the appropriate place with respect to the text. **DO NOT** just insert a bunch of figures and tables at the end of the document.
- Also, you must include a statement at the end of your memo that states that this submission is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results. This discussion may be a part of the results and/or conclusions sections of the memo

## ***ECE-205 Lab 8***

### ***Implementation of Feedback Control Systems***

#### **Overview:**

In this lab you will build a simple feedback control system using both a proportional (P) and proportional plus integral (PI) controller. The systems you will control are simple first and second order systems. Similar to last week, you will also use a prefilter on the to adjust the gain on the input so that there is zero steady-state error and the output follows the input.

#### **Equipment:**

3 - 10 k $\Omega$  variable resistors (potentiometers)

4 - TL072 dual op amps

11- 1 k $\Omega$  resistors

1  $\mu$ F capacitor

0.01  $\mu$ F capacitor

33 mH inductor

#### **Prelab:**

Read the background theory and entire lab procedure thoroughly. Then complete the following analytical calculations and MultiSim simulations. Submit the prelab in class the day before the lab session.

1. Derive the governing differential equation for the circuit in Figure 3. What is the transfer function  $H(s) = U(s)/E(s)$ ?
2. Derive the governing differential equation for the circuit in Figure 4a. What is the transfer function  $H(s) = Y(s)/U(s)$ ?
3. Derive the governing differential equation for the circuit in Figure 4b. What is the transfer function  $H(s) = Y(s)/U(s)$ ?



Multisim simulations

4. In this part you will use Multisim to build a second-order system.
- Place the following parts on the Multisim page: DC\_POWER (x3), GROUND (x2), RESISTOR\_RATED, CAPACITOR\_RATED, INDUCTOR\_RATED, OPAMP\_5T\_VIRTUAL, TD\_SW1
  - Make the voltage sources **12V**, **-12 V** and **1V**. Change the capacitor to **0.01 uF** and change the inductor to **33 mH**. Change the switch to **TON = 0.005 sec** and **TOFF = .010 sec**.
  - Rearrange the components to be the second order system shown in Figure A.

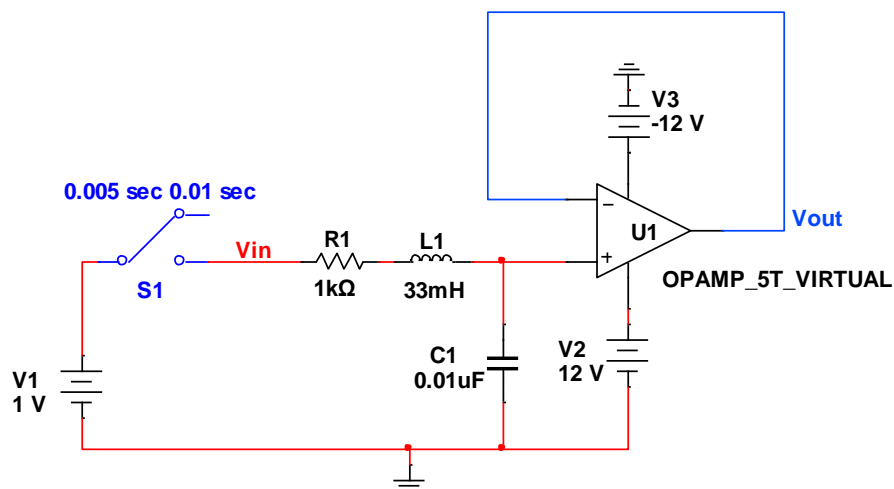


Figure A: Second Order System Plant

- Double click the wire on the output of the op amp and name it **Vout**. Double click the wire on the right side of the switch and name it **Vin**.
- To run the simulation, click Simulate->Analyses->Transient Analysis. Change the Initial conditions drop down to **Set to zero**. Set the End time (TSTOP) to **0.01 s**. Check the **Maximum time step settings (TMAX)** box. Select the radio button to set the maximum time step to **1e-005 s**.



- Select the output tab in the Transient Analysis settings and click on **V(vin)** and **V(vout)** and click **Add**. Click **Simulate**. Your results should look similar to Figure B. Note that there is a significant percent overshoot for this system and this could be an undesirable characteristic of the system.

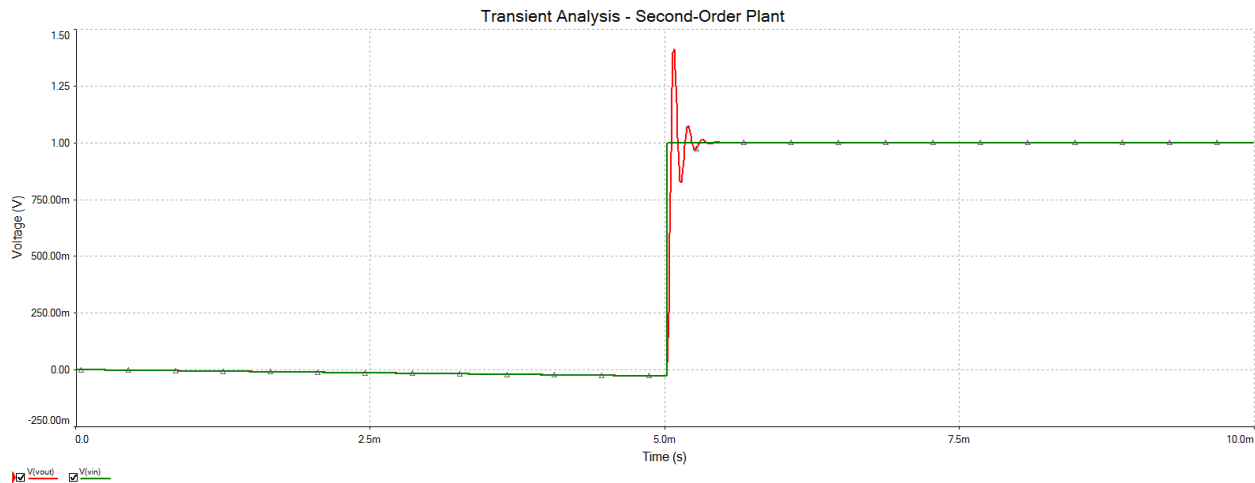


Figure B: Second order plant step response

- Now build the feedback control system to correct for the transient characteristics. You will need to add three more op amps, one for the summer and two for the controller and several more resistors and a capacitor (see Figure C). You may need to increase the Multisim sheet by clicking Options->Sheet Properties. Select the Workspace tab and change the sheet size to 22" x 17" in order to fit the entire schematic.
- Run the simulation again and note that the large percent overshoot has been corrected. In lab you will use potentiometers to adjust the gain on the controller and prefilter in order to get a desired system output.
- This completes your prelab, *you need to submit the Multisim (.ms11) file and the two graphs created (before and after feedback) as part of your prelab submission.*

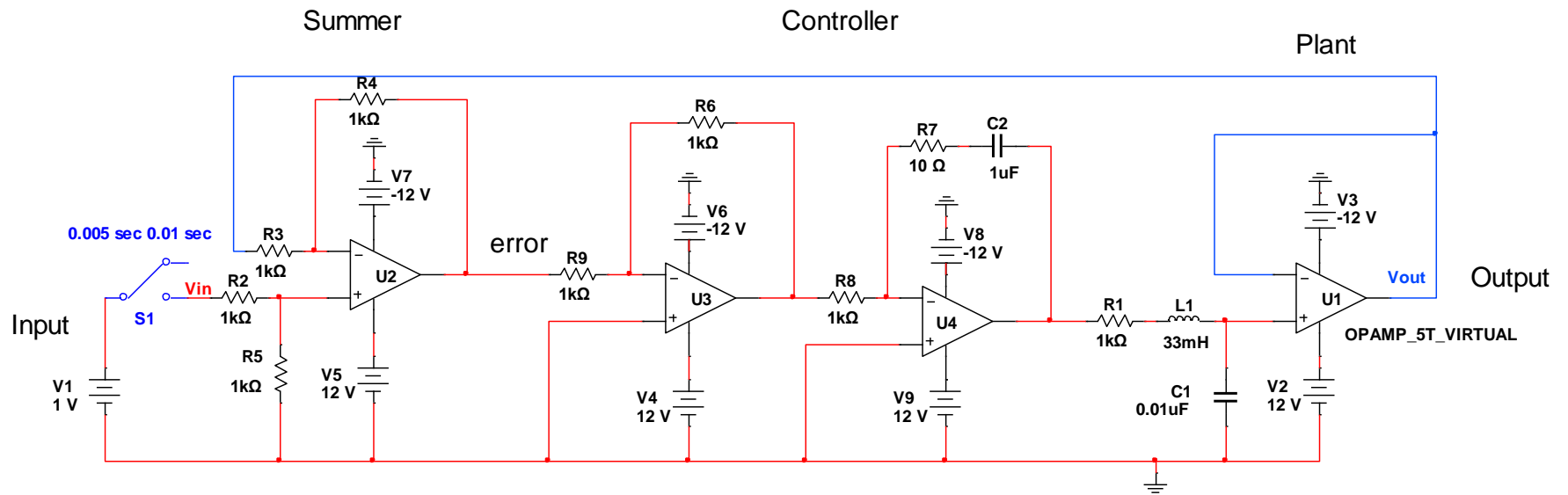


Figure C: Second order system with feedback control

**Theory:**

In Lab 7, we studied the simulation of open loop and closed loop control systems using MATLAB and Simulink. In this lab, you will actually build a closed loop (feedback) control system to adjust the characteristics of a first order and second order system. Recall that the system characteristics are *settling time, percent overshoot, steady-state error, and time to peak*.

There are four key components of the closed loop control system including the prefilter, summer, controller and plant (see Figure 1).

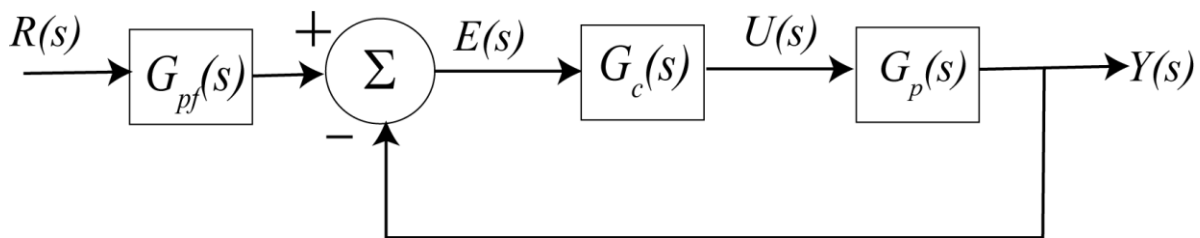


Figure 1: Basic closed loop system with unity feedback

All of these components can be built with an operational amplifier or cascaded op amps. For example, by cascading inverting amplifiers it is possible to produce a constant,  $G_{pf}$  (see Figure 2).

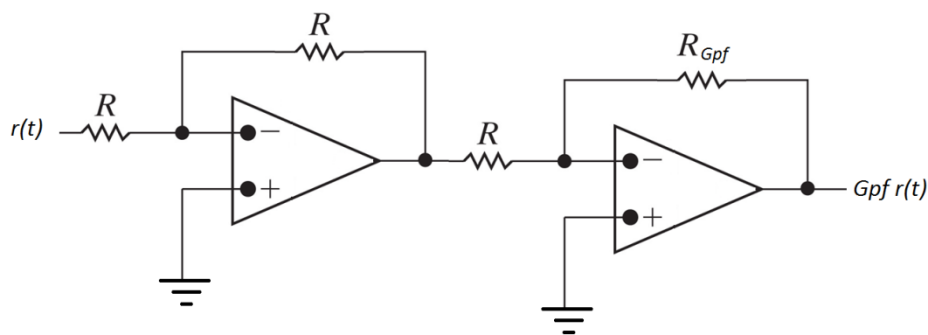


Figure 2: Prefilter ( $G_{pf}$ )

In addition, cascaded inverting amplifiers can also produce a proportional (P) controller,  $k_p$ . By adding a capacitor to the feedback path of an inverting amplifier, it is possible to create a proportional-integral (PI) controller (see Figure 3).

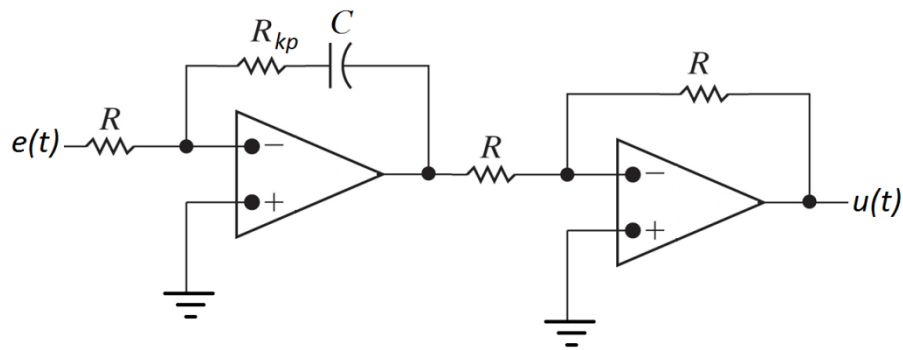


Figure 3: PI Controller ( $k_p + k_i/s$ )

Lastly, the first and second order plants can be represented by operational amplifiers as well (see Figure 4).

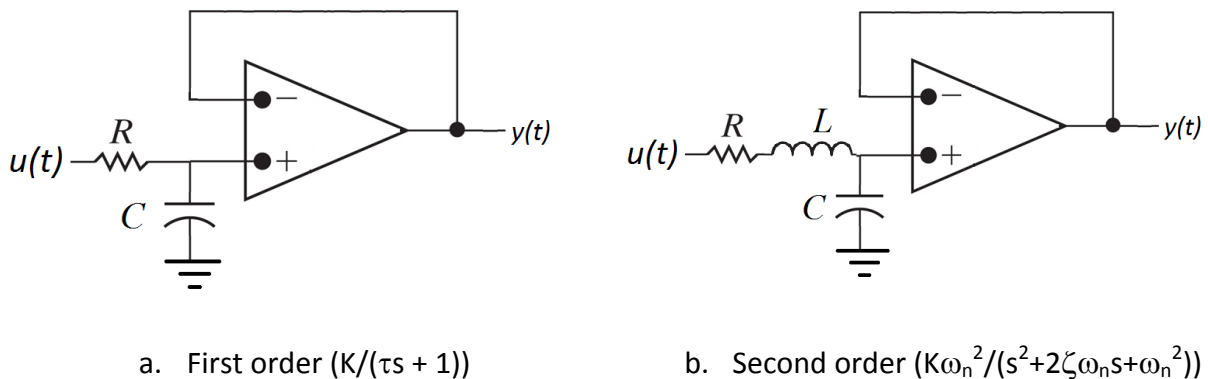


Figure 4: System Plants

In the procedure, you will actually use several op amps to build these system components. Since this is a complicated circuit, you will build the circuit in parts, test them and then connect them together to form the overall feedback control system. **Make sure you build and test the circuit in stages!**

**Procedure:**

**PART I: Proportional Control for a First Order System**

The proportional controller has five parts, shown in the different shaded regions in Figure 5. The equivalent block diagram is shown below the circuit, so that you can match elements of the block diagram with the circuit subsystems. All of the resistors are either 1 kΩ or a variable 10

k $\Omega$  resistor (potentiometer) and a 1  $\mu$ f capacitor. You will also use the TL072 integrated circuit which has 2 operational amplifiers on it. Please review Lab 6 or perform an internet search if you don't recall the pin out for the TL072. Later when you build the second order system, you will also use a 0.01  $\mu$ F capacitor and a 33 mH inductor. **We will build and debug this circuit in stages, so do not start building yet!**

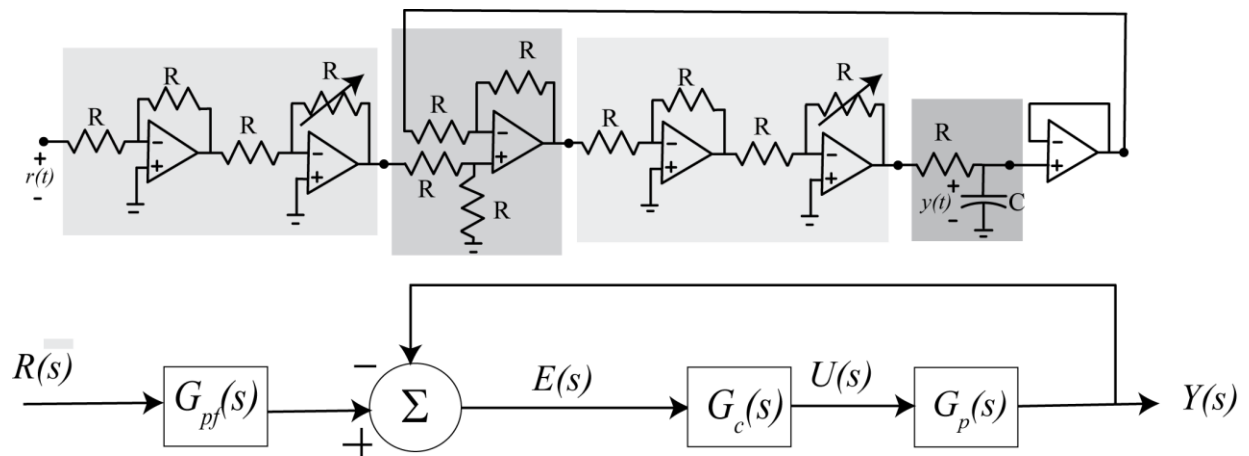


Figure 5: Proportional controller for the first order circuit.

The first shaded part is the prefilter, which changes the gain of the system so the steady state value of the output matches the steady state value of the input. The next stage is the differential amplifier (summer), or the feedback element for finding the difference between the input and output. The third stage is the proportional gain controller. The fourth stage is the plant or first-order system we want to control. The final stage is a simple isolation amplifier to avoid loading effects. Note that the output is the voltage across the capacitor.

## PART II: Build the prefilter

1. In this part you will build the prefilter shown in Figure 6. First, set up the power busses on your breadboard by turning the board horizontal such that the red buss is on top. Use a wire to jump the two blue busses together and this will be the **ground** buss. The top red buss will be for the **+Vcc** or 15V on the operational amplifier. The bottom red buss will be for the **-Vcc** or -15V on the operational amplifier. **DO NOT CONNECT THE POWER FROM THE NI MYDAQ UNTIL THE END.**

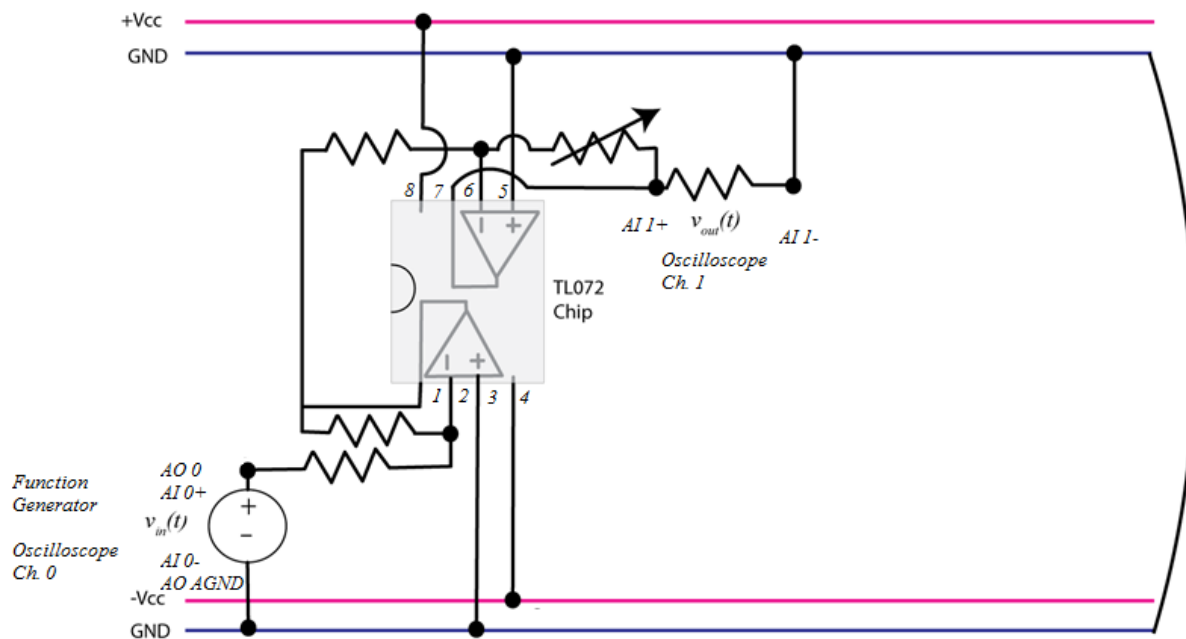


Figure 6: Prefilter (gain circuit)

2. Insert the op amp on the left side of the board with the dot pointing to the left and make sure it straddles the ditch in the middle.
3. Connect pin 3 and pin 5 to **ground** which are the positive terminals on both of the op amps. Connect pin 4 to **-Vcc** and pin 8 to **+Vcc** for the op amp power. Connect a 1 kΩ resistor between pins 1 and 2 and a 1 kΩ resistor from pin 2 to the function generator. The connection for the function generator is **AO 0** and **AGND**. Also, put Channel 0 of the oscilloscope in parallel with the function generator input (**AI 0+**, **AI 0-**).
4. Connect a 1 kΩ resistor between pins 1 and 6. Connect a 10 kΩ variable resistor between pins 6 and 7. Finally, put a 1 kΩ resistor from pin 7 to ground to measure the output. Put Channel 1 of the oscilloscope across the resistor from pin 7 to ground in order to measure the output and confirm that the prefilter is working correctly (**AI 1+**, **AI 1-**).
5. Connect the +15V (+Vcc), -15V (-Vcc) and AGND power supply terminals on the MyDAQ to the red and blue breadboard busses.

6. Set the function generator to a **50 Hz square wave** with a **2.00 V peak to peak** amplitude and a **1.00 V offset**.
7. Start the oscilloscope and enable both channels 0 and 1. Set Channel 0 and Channel to **500 mV/div** and **2 ms/div** time scale. Set the trigger to **Edge, Chan 0 Source** and increase the level to **0.1**.
8. Use the screwdriver to adjust the variable resistor until the input and output are the same or the gain is **one**. Your final oscilloscope output should look similar to Figure 7.
9. Once the circuit is working, **disconnect** the oscilloscope, function generator, power supplies and **remove** the load resistor at pin 7.

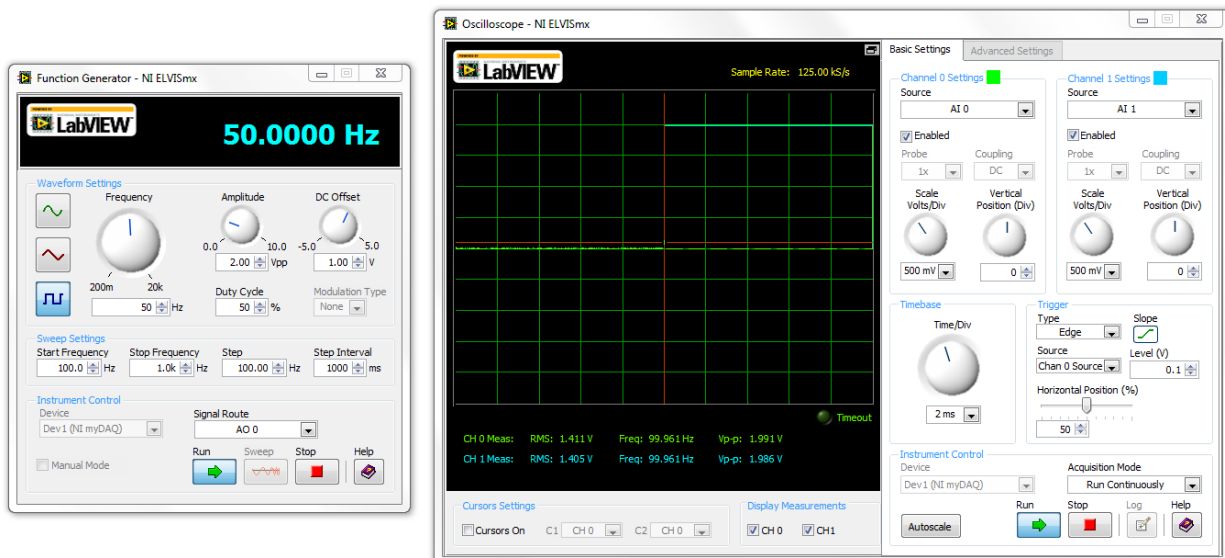


Figure 7: Prefilter oscilloscope output screen

### PART III: Build the differential amplifier

1. In this part you will build the differential amplifier in Figure 8. Place another op amp on the breadboard as close as possible to the prefilter circuit. Make sure that the dot is to the left and the op amp straddles the ditch.
2. Connect pin 4 to **-Vcc** and pin 8 to **+Vcc**.





3. Connect a  $1\text{ k}\Omega$  resistor from pin 6 to pin 7, connect a  $1\text{ k}\Omega$  resistor from pin 5 to ground and another  $1\text{ k}\Omega$  resistor from pin 5 to the function generator (**AO 0, AO AGND**). Channel 0 of the oscilloscope will also go to this same resistor (**AI 0+, AI 0-**).
4. Connect a  $1\text{ k}\Omega$  resistor from pin 6 to ground. *This resistor will eventually be the feedback for the system and be removed from the ground buss.*
5. Connect a  $1\text{ k}\Omega$  resistor from pin 7 to ground. *This resistor will be the load resistor to measure the output for debugging purposes.*
6. Connect the  $+15\text{V}$  ( $+V_{cc}$ ),  $-15\text{V}$  ( $-V_{cc}$ ) and AGND power supply terminals on the MyDAQ to the red and blue breadboard busses.
7. Set the function generator to a **50 Hz square wave** with a **2.00 V peak to peak** amplitude and a **1.00 V offset**. Start the oscilloscope and enable both channels 0 and 1. Set Channel 0 and Channel to **500 mV/div** and **2 ms/div** time scale. Set the trigger to **Edge, Chan 0 Source** and increase the level to **0.1**. If the input and output are not the same then check your wiring.

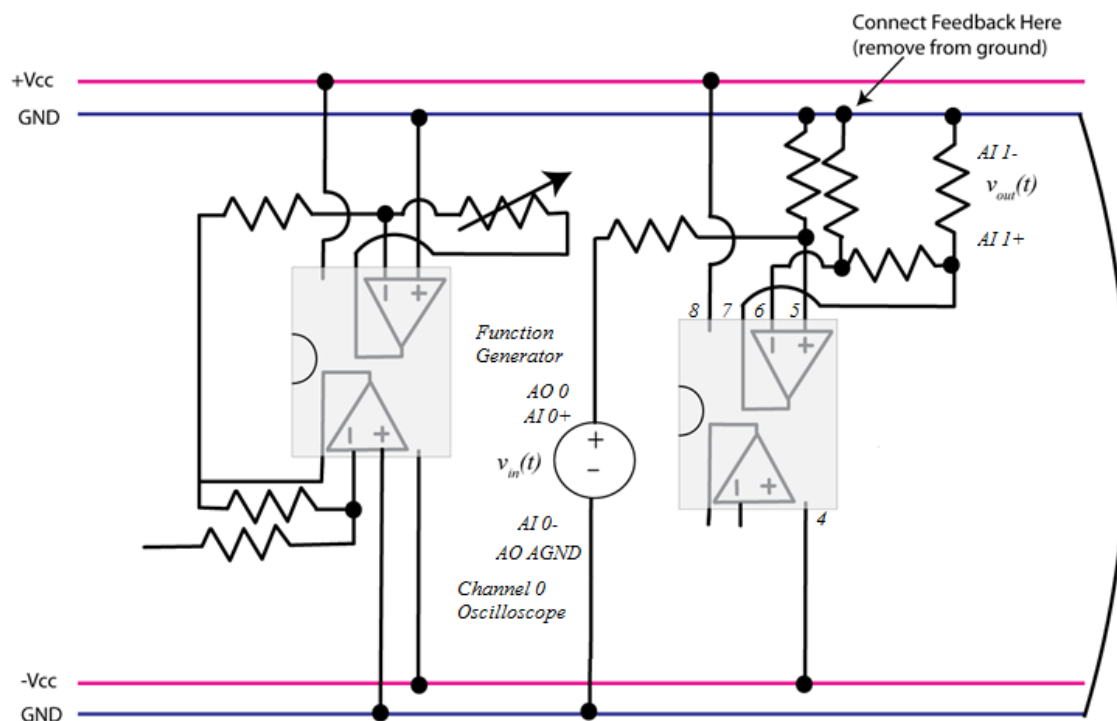


Figure 8: Differential Amplifier Schematic

**PART IV: Connect the prefilter and differential amplifier**

1. Next, connect the prefilter and differential amplifier as shown in Figure 9.
2. Move the function generator and Channel 0 of the oscilloscope back to the input of the prefilter (the resistor at pin 2 on the first op amp).
3. Connect the resistor from pin 5 of the differential amplifier which was connected to the function generator to pin 7 of the first op amp.
4. Keep the same settings for the function generator and oscilloscope and measure the input and output. The output and input of the circuit should match. If they are a little off then use the variable resistor to make them match. If you cannot get the input and output to match then do not go on until you check your circuit and correct it.
5. Finally, **remove** Channel 1 of the oscilloscope and the load resistor on the differential amplifier from pin 7 to ground.

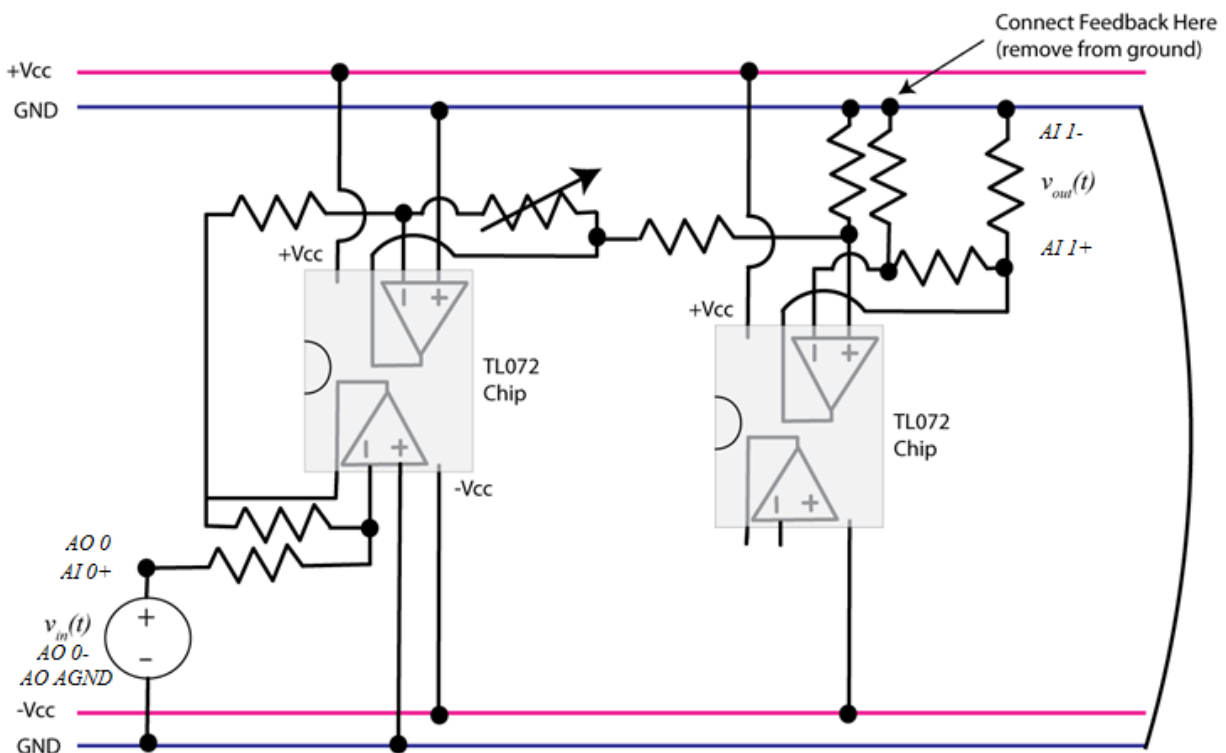


Figure 9: Prefilter and Differential Amplifier Schematic

**PART V: Add the proportional controller**

1. In this part, you will add the proportional control to the system so that it now looks similar to Figure 10. Place another op amp on the board, with the dot on the left.
2. The proportional controller consists of two cascaded inverting amplifiers. Both of these will be created on the new op amp.
3. On the new op op amp, connect pins 3 and 5 to **ground**, connect pin 4 to **-Vcc** and connect pin 8 to **+Vcc**.
4. Connect another 1 k $\Omega$  resistor between pin 7 on the differential amplifier and pin 2 on the proportional controller to cascade them together.
5. Connect a 1 k $\Omega$  resistor between pins 1 and 2 on the proportional controller. Try to use wires from your kit to keep the layout as flat as possible to help with debugging later.
6. Connect a 1 k $\Omega$  resistor from pin 1 to pin 6. Connect the 10 k $\Omega$  variable resistor from pin 7 to pin 6.
7. Lastly, for debugging purposes, place a resistor from pin 7 to ground.
8. Use the oscilloscope to measure the input and the output of the cascaded 3 stage system. Adjust the variable resistor until the output and input are the same. If you cannot get them to match, then check your circuit and do not move on until it does.
9. **Remove** the power supplies, oscilloscope, function generator and load resistor at pin 7.

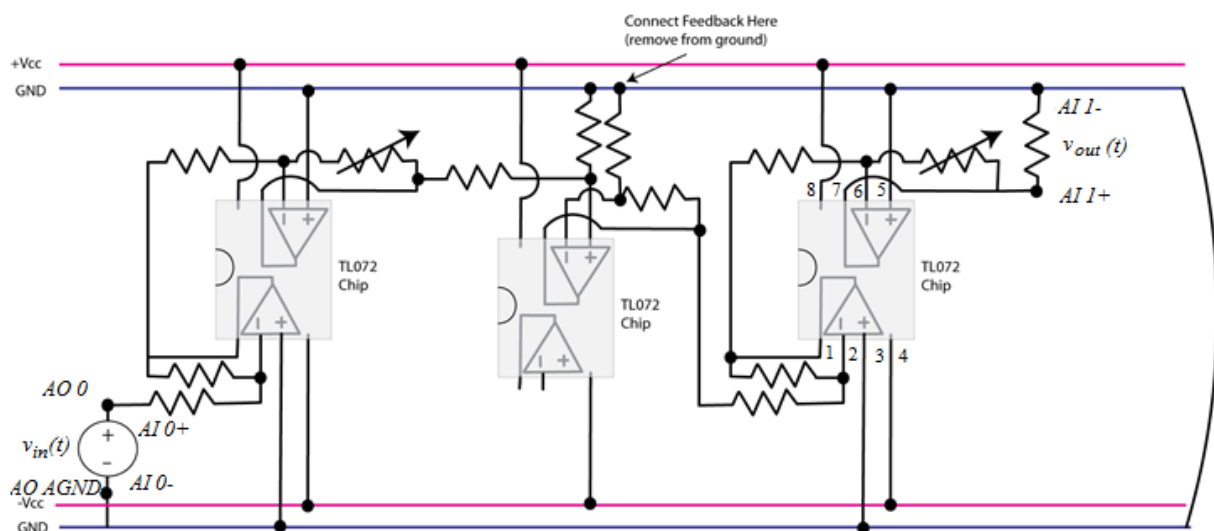


Figure 10: Proportional Control Schematic

**PART VI: Add the system plant**

1. In this part, you will build the system plant which is a first order RC circuit in series with an isolation amplifier. The purpose of the isolation amplifier is to avoid loading effects on the circuit.
2. Add a 4<sup>th</sup> op amp to the breadboard and build the circuit in Figure 11. Connect pin 4 to **-Vcc** and pin 8 to **+Vcc**.

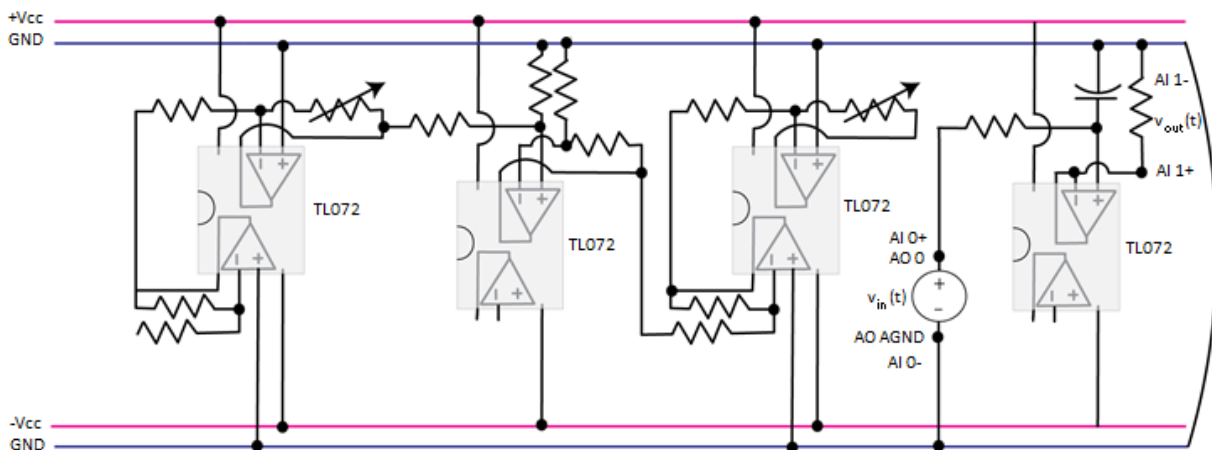


Figure 11: System Plant Schematic

3. Connect a  $1\ \mu\text{F}$  (105) capacitor from pin 5 to ground. Connect a  $1\ \text{k}\Omega$  from pin 5 to the function generator and Channel 0 of the oscilloscope.
4. Connect a small wire between pins 6 and 7.
5. Connect a  $1\ \text{k}\Omega$  from pin 7 to ground to measure the output of the system for debugging purposes. Connect Channel 1 of the oscilloscope across the load resistor at pin 7.
6. Measure the output of the plant and it should look like Figure 12. This represents the open loop response of the system plant. If you do not get a response similar to this figure, do not move on until you fix the circuit.
7. **Remove the load resistor on pin 7, function generator, oscilloscope and power supplies**

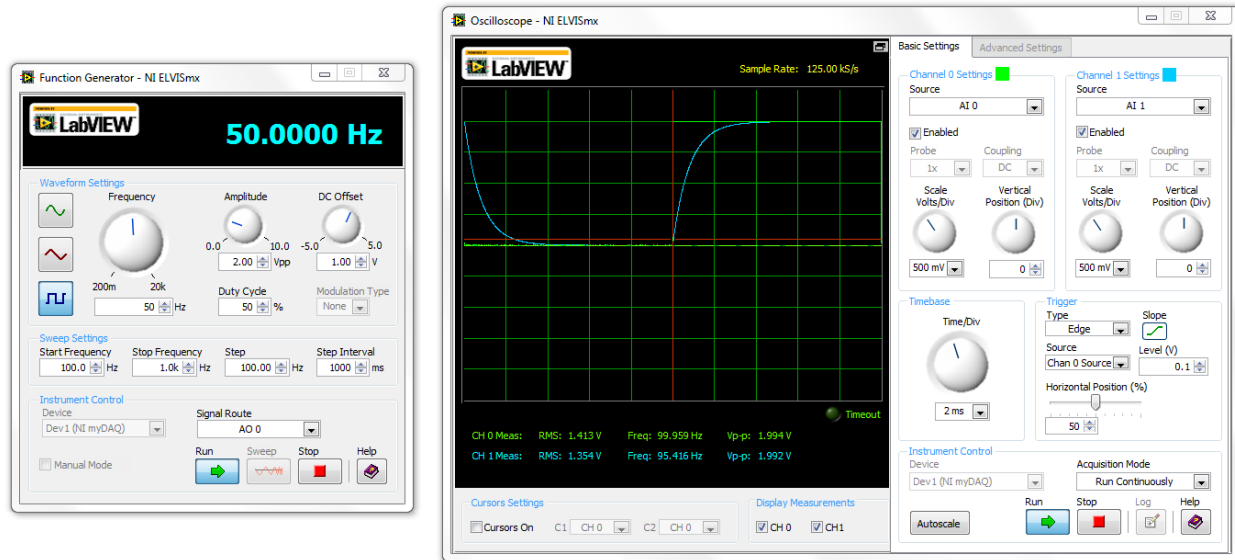


Figure 12: Open loop response of the first order plant

**PART VII: Add the feedback**

1. In this part, you will add the feedback to the system by modifying the circuit to look like Figure 13. Disconnect the 1 kΩ resistor at pin 6 on the differential amplifier from ground. Add a long wire from that resistor to pin 6 on the isolation amplifier (plant).

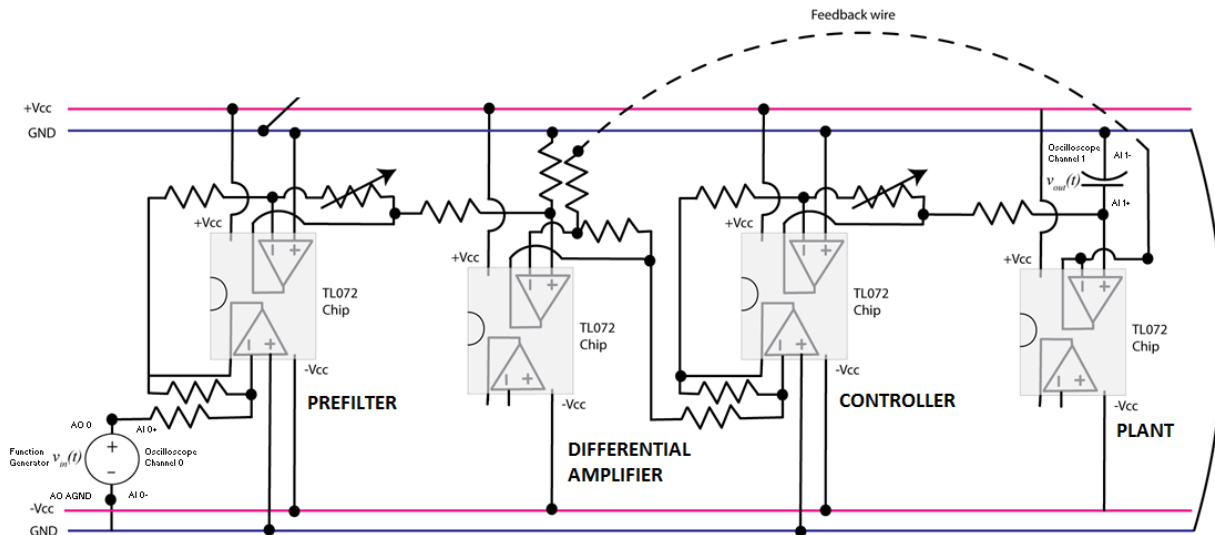


Figure 13: Closed loop system schematic

2. Move the resistor at pin 5 on the isolation amplifier (plant) to pin 7 on the controller.



3. Attach the function generator and Channel 0 of the oscilloscope to the input of the full system at the resistor on pin 2 of the prefilter.
4. Attach Channel 1 of the oscilloscope to the output of the system across the capacitor on pin 5 of the isolation amplifier (plant).
5. Measure the input and output of the system and you should get a response similar to Figure 14. Note that amplitude is about half of the amplitude for the open loop system and the settling time is about half as long.
6. Slowly change the variable resistor on the proportional controller to speed up the response. Then change the variable resistor on the prefilter so that the output is the same amplitude as the input.
7. You need to submit **two** different screen captures of the system response (oscilloscope) in your memo. They both should have a faster response (smaller settling time) and the same amplitude as the input (zero steady state error) to indicate that the controller was working.

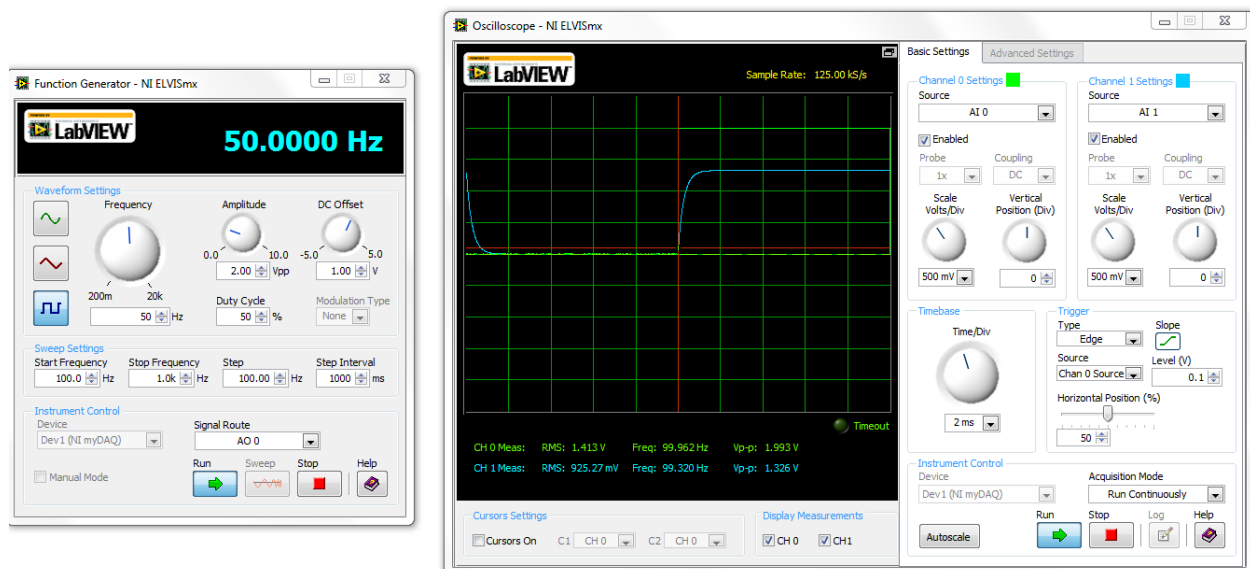


Figure 14: Closed loop response of a first order system

## PART VIII: PI Control for a first order system

1. In this part, you will change the proportional controller to a proportional plus integral controller to determine how it affects the first order plant. This requires only slight modification in the circuit to make it look like Figure 15.

- Remove the  $1\text{ k}\Omega$  feedback resistor between pins 1 and 2 on the proportional controller with a  $1\text{ }\mu\text{F}$  capacitor in series with a variable resistor (see Figure 15).

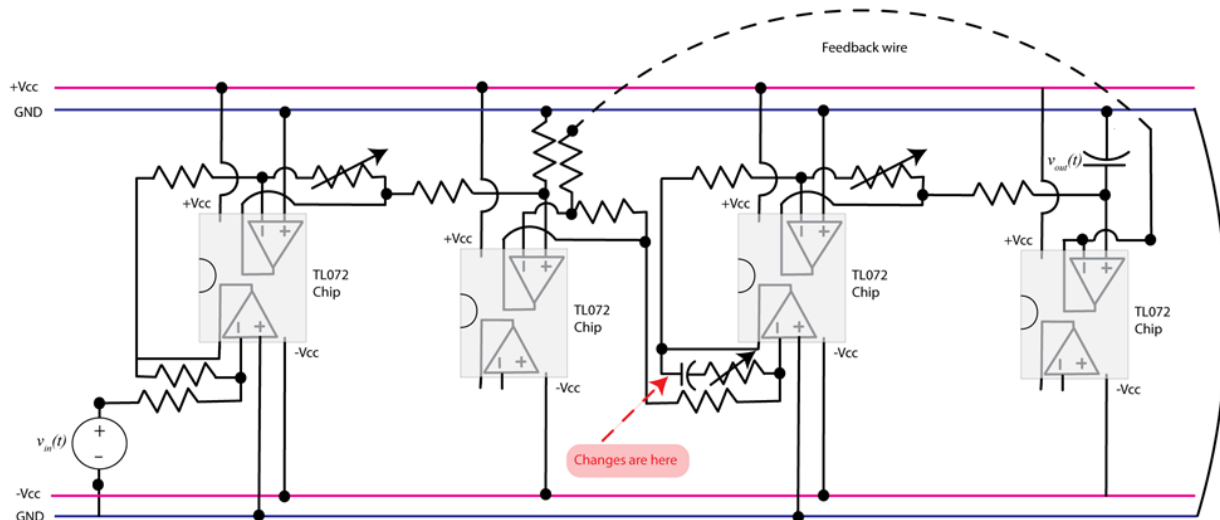


Figure 15: PI Controller Schematic

- Note that both of the variable resistors affect the proportional gain ( $k_p$ ) and the second variable resistor affects the integral gain ( $k_i$ ).
- Set the both variable resistors on the controller to the middle of the range and adjust the prefilter gain so that steady state output matches the input (zero steady-state error).
- Now go back and adjust the variable resistors on the controller. We can use these to change the gain of the proportional term and the gain of the integral term. You should note that as you change these resistors, the output still matches the input (as long as the system has time to reach steady state). This is the benefit of having an integral controller!
- Adjust your controller so you get an output similar to Figure 16 which has a **50% overshoot** and settling time of approximately **5.00 ms**. You should include a screen shot of this oscilloscope output in your lab memo.
- Next, adjust the controller so that the output has virtually **no overshoot** and a settling time of approximately **1.00 ms** similar to Figure 17. You should include a screen shot of this oscilloscope output in your lab memo.

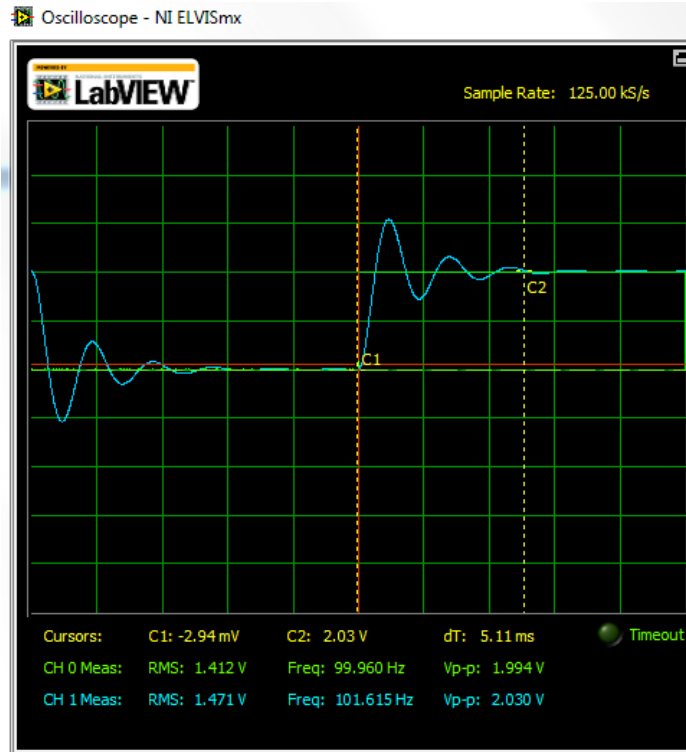


Figure 16: PI Control of a first order system with 50% overshoot

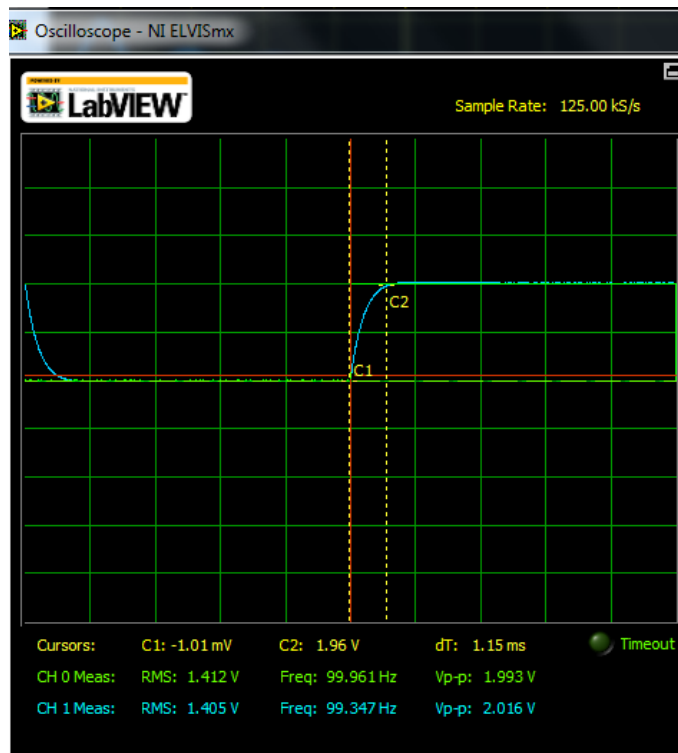


Figure 17: PI Control of a first order system with 0% overshoot



**PART IX: Second order system plant**

1. In this part, the plant will be changed to a second order system and we will use a PI controller to examine the changes in the system characteristics. This is not as straightforward as the first order system because it is typically more difficult to control a system as the order increases.
2. Remove the first order plant from the circuit. This includes the  $1\ \mu\text{F}$  capacitor and the  $1\ \text{k}\Omega$  resistor.
3. Modify the circuit so that the plant and isolation amplifier now look like Figure 18. The resistor is  $1\ \text{k}\Omega$ , the inductor is  $33\ \text{mH}$  and the capacitor is  $0.01\ \mu\text{F}$  (103).

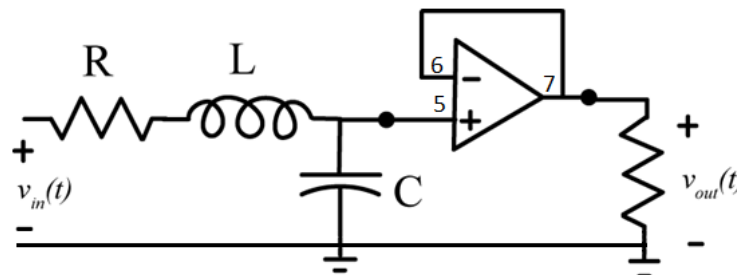


Figure 18: Second order system plant

4. To make sure the plant (system) is built correctly, disconnect the feedback wire to the differential amplifier from the plant. Also, add a  $1\ \text{k}\Omega$  load resistor at pin 7 for debugging purposes.
5. Connect the function generator and Channel 0 of the oscilloscope to the input of the plant at the  $1\ \text{k}\Omega$  resistor. Connect Channel 1 of the oscilloscope to the output of the plant across the load resistor.
6. Set the input to a **1 kHz square wave** with **2 V pp** amplitude and a **1 V peak offset**. Set both oscilloscope channels to **1V/div** and **100  $\mu\text{s}/\text{div}$** .
7. The output is the voltage across the capacitor and the graph should look similar to Figure 19. Note that this system has a settling time of approximately **200  $\mu\text{s}$**  and about a **30% overshoot**. You should include a screen shot of this oscilloscope screen in your memo.
8. **Remove** the function generator, oscilloscope and  $1\ \text{k}\Omega$  load resistor at pin 7.

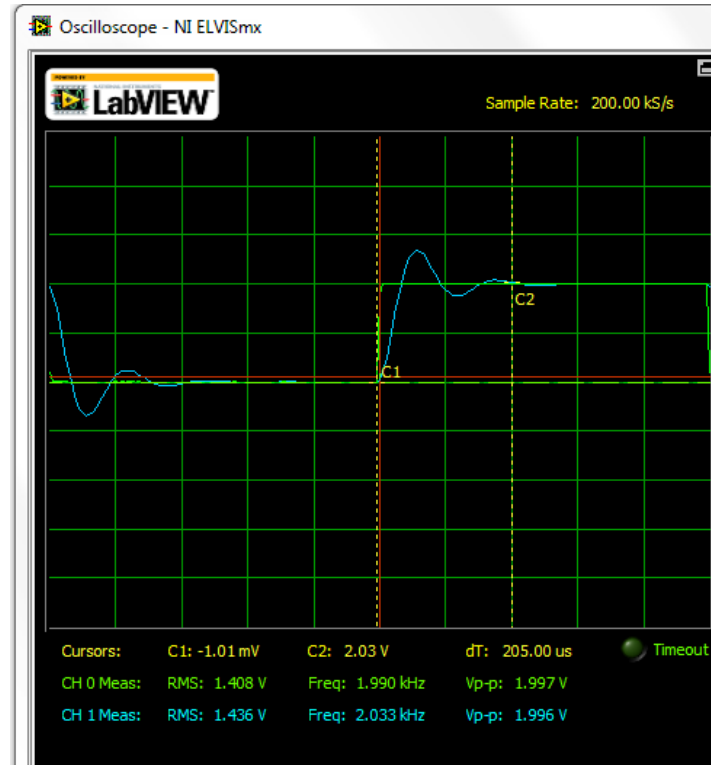


Figure 19: Open loop step response of a second order plant

#### PART X: PI Control for a second order system

1. In this part you will connect your system, or plant, to the feedback network. Connect the feedback wire back to pin 6 on the isolation amplifier (plant).
2. Connect the input resistor on the plant to pin 7 on the PI controller.
3. Move the function generator and Channel 0 of the oscilloscope back to the input of the system at the prefilter. Use Channel 1 of the oscilloscope to measure the output of the system at the voltage across the  $.01 \mu\text{F}$  capacitor.
4. Adjust the system so there is **no overshoot** and the settling time is less than approximately **500  $\mu\text{s}$**  as shown in Figure 20. You may have to adjust the time scale and frequency of the signal generator to get a good picture. Be sure your system has reached steady state (it's flat on the top!). You should include a screen shot of the system output in your memo.
5. Note that in this case, we cannot speed up the response of the system if our goal is to reduce the overshoot. You should remember that we are not changing our plants, or basic

systems, but we are changing how they behave using feedback and an active controller to change the system characteristics.

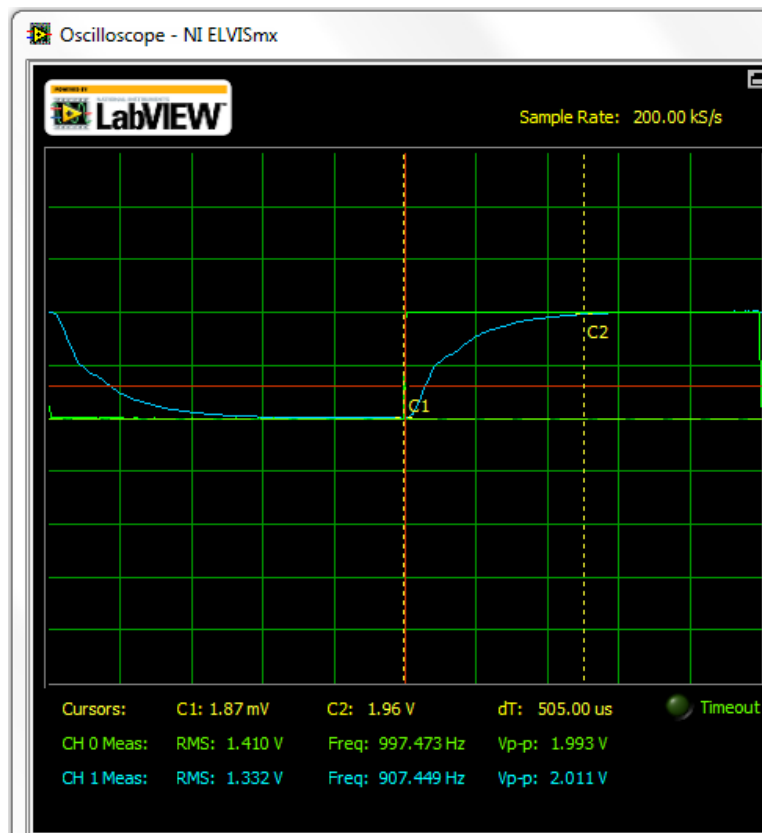


Figure 20: Closed loop step response of a second order plant

### Submission Requirements:

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- A header at the beginning with *Date, To, From, Subject*
- Written in first person from you
- Written with minimal spelling and grammar errors



- *Purpose, procedure, results and conclusions* of the laboratory experiment
  - The very first sentence is the purpose and must explain why you are doing the experiment
  - The *procedure* should be very short, a high level summary of what you did for each part. The *procedure* should be two paragraphs at the most.
  - The *results* should include the tables and figures and results of the data collected and your observations. The results cannot be pages of figures and tables only, it **must** include text that references the figures and explains what they are to the reader.
  - The *conclusions* must be the end and it should be a summary of what you did, what you learned, what you observed and recommendations
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text. **DO NOT SUBMIT SEPARATE FILES**, images must be pasted into the Word document at the appropriate place with respect to the text. **DO NOT** just insert a bunch of figures and tables at the end of the document.
- Also, you must include a statement at the end of your memo that states that this submission is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results. This discussion may be a part of the results and/or conclusions sections of the memo



## ***ECE-205 Lab 9***

### ***Bode Plots***

#### **Overview:**

In this lab you will use MATLAB to contrast the Bode plot of several systems given the transfer function. You will also measure the frequency response of two circuits and then use the NI myDAQ and Multisim to display the measured and simulated data. For many unknown (or complex) systems this is a very common method for determining the transfer function. Lastly, you will use the Bode plot to determine the steady-state response of a system given the input.

#### **Equipment:**

1 k $\Omega$  resistor  
 1  $\mu$ F capacitor  
 0.01  $\mu$ F capacitor  
 33 mH inductor  
 10 k $\Omega$  variable resistor (potentiometer)

#### **Prelab:**

In this section, you will use MATLAB to generate the Bode plot of several transfer functions. You will use the `bode` command to generate the Bode plot given a transfer function.

In order to find the Bode plot of the transfer function,  $G(s) = \frac{0.1 \left[ \frac{1}{10} s + 1 \right]}{\left[ \frac{1}{100} s + 1 \right]}$

Type the following code in MATLAB,

```
num = 0.1*[1/10 1]; %define the numerator of the transfer function
den = [1/100 1]; %define the denominator of the transfer function
G = tf(num,den); %define the transfer function
w = logspace(-2,5,1000); %define the frequency from w = 10^-2 to 10^5 rad/s
bode(G,w);grid; %create the Bode plot
```

In order to find the Bode plot of a transfer function that has repeated poles or zeros, use the `conv` command which performs a discrete-time convolution.



In order to find the Bode plot of the following transfer function,  $G(s) = \frac{\left[ \frac{1}{100}s + 1 \right]^3}{\left[ \frac{1}{10}s + 1 \right]^4}$

Type the following code in MATLAB,

```
nz = [1/100 1]; %numerator factor
num = conv(nz, conv(nz, nz)); %raised to the third power
dp = [1/10 1]; %denominator factor
den = conv(conv(dp, dp), conv(dp, dp)); %raised to the fourth power
G = tf(num, den); %create the transfer function
w = logspace(-2, 5, 1000); %define the frequency
bode(G, w); grid; %create the Bode plot
```

Alternately, you can also use the **poly** command which is based upon using the zeros and poles of the transfer function to create the numerator and denominator polynomial. Type the following code in MATLAB to demonstrate this technique,

```
num = poly([-100 -100 -100]); %raised to the third power
den = poly([-10 -10 -10 -10]); %raised to the fourth power
G = tf(num, den); %create the transfer function
w = logspace(-2, 5, 1000); %define the frequency
bode(G, w); grid; %create the Bode plot
```

Please try these examples in MATLAB before moving on.

1. Submit the MATLAB Bode plots for the following transfer functions in class the day before lab.

a.  $H(s) = \frac{100}{s^2}$

b.  $H(s) = \frac{0.1\left(\frac{1}{100}s+1\right)^2}{s^2}$

c.  $H(s) = 10\left(\frac{1}{100}s+1\right)$

d.  $H(s) = \frac{0.1\left(\frac{1}{10}s+1\right)^2}{\left(\frac{1}{1000}s+1\right)^2}$

e.  $H(s) = \frac{0.1\left(\frac{1}{10}s+1\right)^2}{\left(\frac{1}{100}s+1\right)^2\left(\frac{1}{1000}s+1\right)^5}$

f.  $H(s) = \frac{1}{\left(\frac{1}{10}s+1\right)\left(\frac{1}{100}s+1\right)}$

2. In this problem, you will examine what happens to a Bode plot as the poles and zeros of a transfer function is varied.
  - a. Download the **preLab9.m** file from the Angel course folder. This file plots the magnitude of the frequency response of the transfer function based on the location of the poles and zeros entered. This magnitude plot is not in dB and the amplitude has been scaled to have a maximum value of 1 for all of the plots.
  - b. When the file runs, the MATLAB command window will prompt you for the values of the poles and zeros. Make sure to enter the poles and zeros in matrix form, i.e. [-10 +10j -10-10j] and [].
  - c. For each of the following sets of poles and zeros create the plots and observe the results. You can enter all of the plots into Microsoft Word for you submission but do not put more than 3 figures per page to insure that they are large enough to read.
    - i. Run the code with poles at  $-10 \pm 10j$ ,  $-5 \pm 10j$ ,  $-1 \pm 10j$  and no zeros (leave the variable zeros as an empty array, do not put in a zero). Note that as the magnitude of the imaginary part of the pole gets significantly larger than the magnitude of the real part, there are well defined peaks in the frequency response. *Make sure to submit all three plots.*
    - ii. Leave the poles at  $-1 \pm 10j$  and add zeros first at  $0$ , then at  $0$  and  $\pm 5j$ , then at  $0$  and  $\pm 15j$ . Note that at the frequency where a zero of the transfer function occurs, the output of the transfer function is zero! Turn in all three plots.

**Theory:**

**Bode diagrams** are graphical techniques that describe the frequency response of a circuit. The Bode diagram helps you quickly visualize how the system responds to a wide range of frequencies. There are two separate plots: amplitude (magnitude) and phase angle and they vary with the frequency (see Figure 1). The plots are on semi log graph paper with the frequency on the horizontal log scale and the amplitude and phase angle on the linear vertical scale. Sometimes the amplitude plot is expressed in **decibels**.

$$|H(j\omega)|_{dB} = 20 \log_{10} \frac{v_{out}}{v_{in}}$$

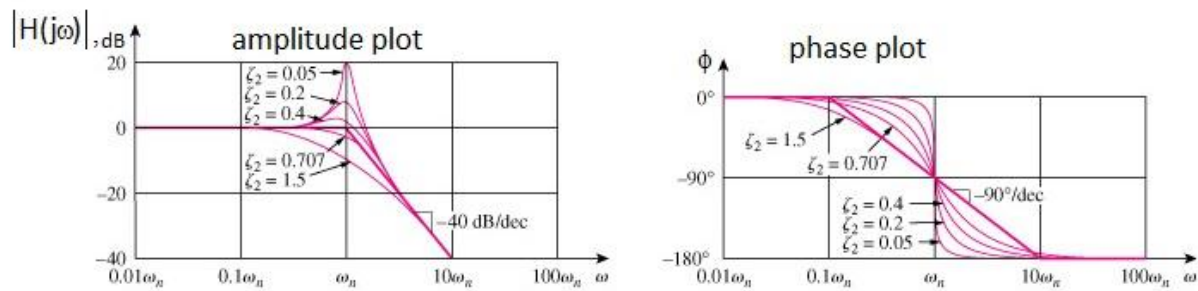


Figure 1: Bode Plots

In order to use a Bode plot find the steady-state response of a system given the input you will use the following process. For an input  $x(t) = A \cos(\omega_o t)$  we know the steady state output will be

$$y_{ss}(t) = A |H(j\omega_o)| \cos(\omega_o t + \angle H(j\omega_o))$$

Hence if we measure the input and output amplitude at each input frequency  $\omega_o$ , it should be straightforward to determine  $|H(j\omega_o)|$  at each of these frequencies. To determine the phase, we will look at the time delay between the input and output signals at each input frequency, as follows,

$$\cos(\omega_o(t - t_d)) = \cos(\omega_o t - \omega_o t_d) = \cos(\omega_o t + \angle H(j\omega_o))$$

Note that in this relationship the phase of the transfer function is measured in radians,

$$\angle H(j\omega_o) = -\omega_o t_d$$

We can then convert the phase from radians to degrees using

$$\angle H(j\omega_o) \text{ (in degrees)} = -2\pi f_o t_d \times \left[ \frac{180^\circ}{\pi} \right] = -360^\circ f_o \times t_d$$



**Procedure:****PART I: Frequency Response of a First Order System**

1. The RC circuit in Figure 2 represents a first order system. Build the circuit on the breadboard with  $R = 1 \text{ k}\Omega$  and  $C = 1 \text{ }\mu\text{F}$ . The input will be the function generator and the output will be the voltage across the capacitor.

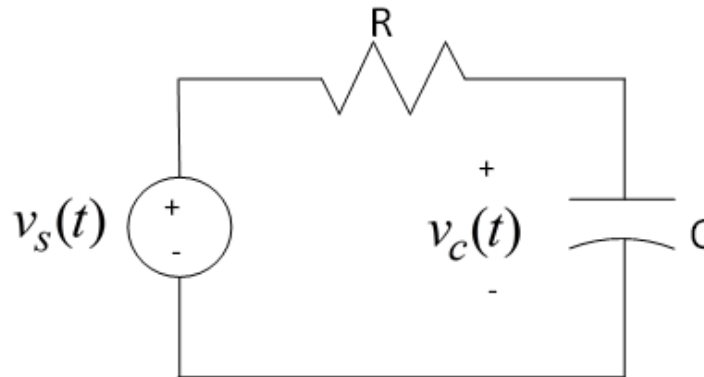


Figure 2: First order system (RC Circuit)

2. The transfer function for this circuit is

$$H(s) = \frac{K}{\tau s + 1}$$

where  $K$  is the static gain,  $\tau$  is the time constant. For this model  $K = 1$  and  $\tau = RC = 1 \text{ ms}$ .

3. This circuit is also a **low pass filter** where the **corner** or **cutoff frequency** is  $\omega = 2\pi f = 1/\tau = 1000 \text{ rad/s}$ ,  $f = 159 \text{ Hz}$ . The cutoff frequency is the frequency where the power in the input signal has been reduced to one half of its maximum value, or equivalently, the magnitude of the transfer function has been decreased to  $1/\sqrt{2}$  of its maximum value. A **low pass filter** passes low frequencies below the cutoff frequency and attenuates signals above the cut off frequency. The range of frequencies passed by the filter is the **passband**. For this first order circuit, the maximum value of the transfer function is at  $\omega = 0 \text{ rad/s}$  so the magnitude at the corner frequency is  $|H(j\omega_c)| = \frac{1}{\sqrt{2}} |H(0)|$ . Note that this point is also called the **3 dB point**.



4. Connect the NI MyDAQ to the circuit by putting the function generator (**AO 0, AO AGND**) as the source. Connect Channel 0 (**AI 0+, AI 0-**) of the oscilloscope to the input to the circuit as well. Connect Channel 1 (**AI 1+, AI 1-**) of the oscilloscope to the output of the circuit across the capacitor.
5. Connect the NI My DAQ to your laptop and start the Function Generator and Oscilloscope instruments.
6. Set the function generator to a **20 Hz sine wave** with a **2.00 Vpp** amplitude and **0.00 V** DC offset.
7. Set the oscilloscope to **5 ms** per division, enable Channels 0 and 1 and set them both to **200 mV** per division and the Trigger type to **Edge**. As you vary the frequency, you should always adjust the time base so that one or two periods of the input and output wave forms are on the screen.
8. For an input  $x(t) = A \cos(\omega_o)t$ , the steady state output will be  $y_{ss}(t) = A|H(j\omega_o)|\cos(\omega_o t + \angle H(j\omega_o))$  so if you measure the input and output amplitude at each frequency,  $\omega_o$ , then you can calculate the amplitude,  $|H(j\omega_o)|$ .
9. To calculate the phase,  $\angle H(j\omega_o)$ , determine the time delay,  $t_d$ , between the input and output signals at each frequency,  $\omega_o$ . Then use the following formula to calculate the phase angle in degrees,
 
$$\angle H(j\omega_o) = -\omega_o t_d \times (180/\pi)$$
10. You will use the following steps to create the Bode plot for the system,
  - Choose a frequency for the input signal (keep the input amplitude at 1 volt, or 2 volts peak to peak)
  - Using the cursors, measure the input amplitude (not peak to peak)
  - Using the cursors, measure the output amplitude (not peak to peak)
  - Using the cursors, measure the time delay between the input and output by putting Cursor 1 on Channel 0 and Cursor 2 on Channel 1. Put Cursor 1 on the first peak of Channel 0 and Cursor 2 on the subsequent peak of Channel 1.
11. Open up Excel and create a table with the following 4 column headings.
  - Frequency (in Hz)
  - Amplitude of the input signal (in mV)
  - Amplitude of the output signal (in mV)



- Time delay (in milliseconds)
12. For the initial Bode plot, collect and record the appropriate data in Excel for the input frequencies: 20, 60, 100, 140, 180, 220, 280, 320, 520, 720, 920, 1120 Hz. Figure 3 provides an example of the system input and output with cursor measurements.

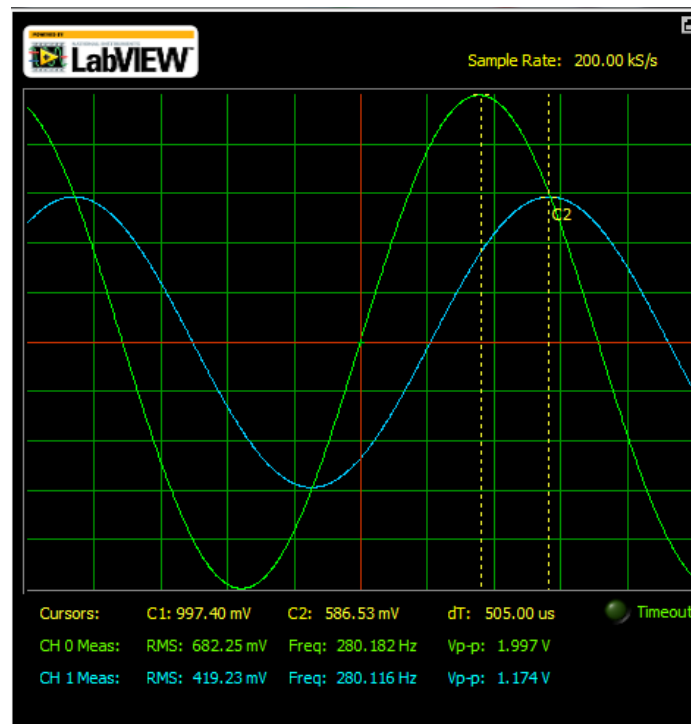


Figure 3: Sinusoidal Steady State Response of the first order system

13. Download the MATLAB script file, ***process\_data.m*** from the Angel course Lab 9 folder. This program processes the data collected from the frequency response of the system. Copy and paste the data from Excel into the **measured** variable in this program. In the Matlab window, type
- ```
data = process_data;
```
14. Download the MATLAB script file, ***model\_a.m*** from the Angel course Lab 9 folder. This program uses this data, as well as knowledge of the expected transfer function, to best fit the transfer function to the measured data and produce a couple of nice plots. There are three input arguments to this program, the array data, the estimated gain, and the

estimated time constant. How well this program works in part depends on how well you can guess these values.

15. Assume that the **gain is 1** and the **time constant is 0.001** and run the program by typing the following in the MATLAB command window:

```
model_a( data, 1.0, 0.001);
```

16. MATLAB will print a series of values on the screen and then eventually it will produce the magnitude and phase plots.
17. If the plots are not close then one of two things has happened:
- Your initial guess was way off
  - Your data, or at least some of it, was not correct
18. At this point you may want to check any data that does not look too good and rerun the programs before you go on. Remember that if you add more data points, you need to rerun ***process\_data.m*** to change your data before generating the Bode plot.
19. To add more data points to fill out your graph (spread throughout the frequency range). Your final plot should have at least 15 frequencies. Note that you can just add your new data to the end of the matrix ***measured*** in ***process\_data.m*** since it gets sorted. Figure 4 provides an example of an acceptable Bode plot for a first order system.
20. Use the final Bode plot to estimate the system the cutoff frequency. It should be close to 160 Hz.
21. ***Include your final Bode plot figure in your memo and attach the process\_data.m in the Angel drop box***

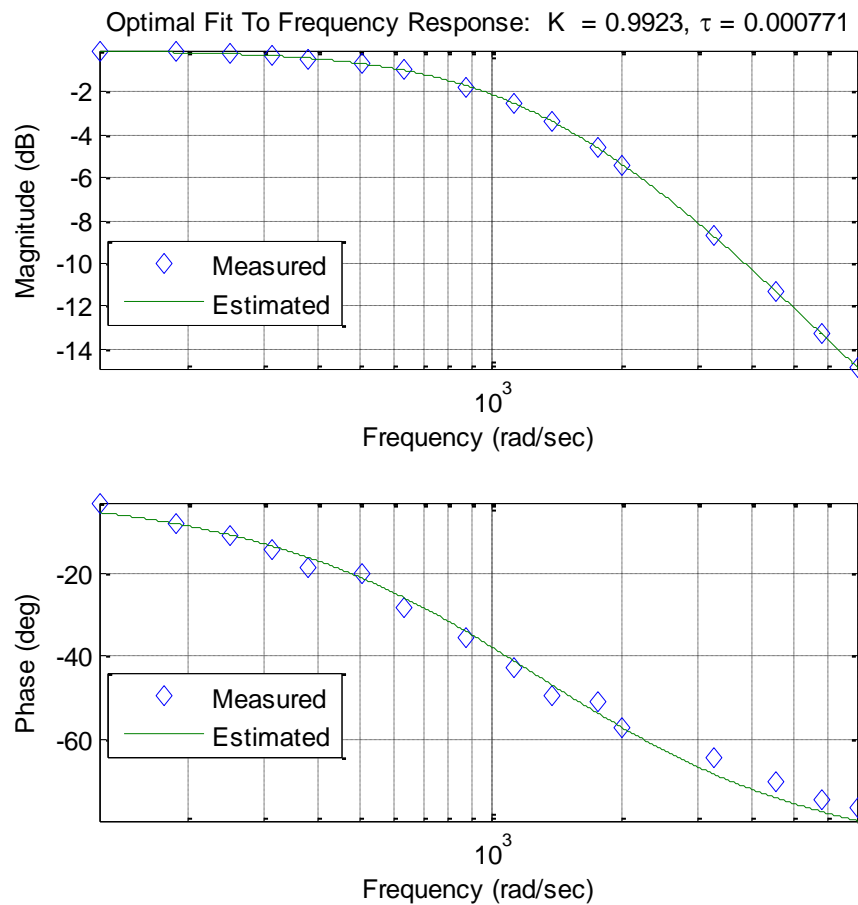


Figure 4: Bode Plot of the first order system

PART II: Frequency Response of a First Order System Again

1. Connect the NI MyDAQ to your laptop and start MultiSIM. You will actually create a design to output the simulation and measured Bode plot data on the same graph.
2. Click File → New → NI myDAQ Design. The screen will be automatically populated with the MyDAQ connection points.
3. Place the following components on the sheet: RESISTOR\_RATED, CAPACITOR\_RATED.
4. Next connect the RC circuit to the terminals to the MyDAQ terminals in MultiSIM just as you did on the breadboard so that the circuit looks like Figure 4:

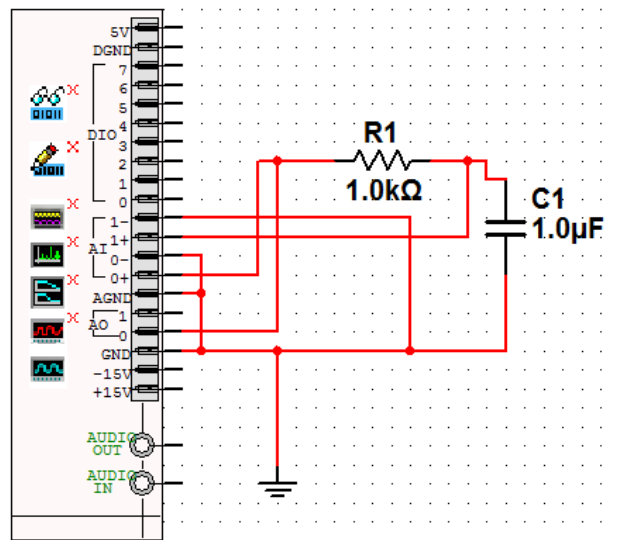


Figure 4: RC Circuit in MultiSim

5. Right click the toolbar at the top of the Multisim window and select NI ELVISmx Instruments. Once the toolbar opens, select the Bode Analyzer which is the last button.
6. Connect the Stimulus to the input the circuit and the Response to the output of the circuit so that it looks like Figure 5.

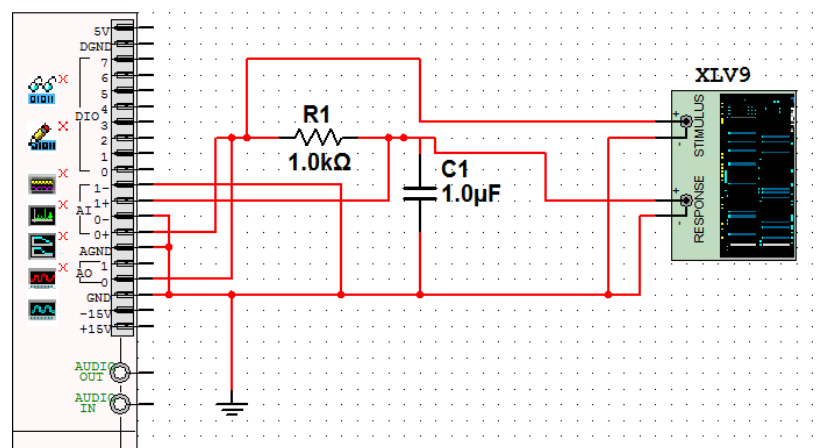


Figure 5: Bode Analyzer

7. Next double click the Bode Analyzer to open up the front panel. Also double click on the Function Generator on the MultiSim MyDAQ graphic at the bottom left.
8. On the Bode Analyzer, change the device to **Simulate NI myDAQ**, the start and stop frequencies to **10 Hz** and **100 kHz** and Steps to **10 (per decade)**.

9. You don't have to change any settings on the function generator but click **Run** on the MultiSim window and **Run** on the Bode Analyzer graphic and the Bode plot should display.
10. **Stop** the simulation and change the Device to **Dev1 (NI myDAQ)**. The actual data from the circuit will now plot on the same figure. Your plot should look similar to Figure 6.
11. *Use the Bode plot to estimate the cutoff frequency for your system.* In your memo, you should compare and contrast this result to the MATLAB result.
12. *You should include this graph output in your memo.*

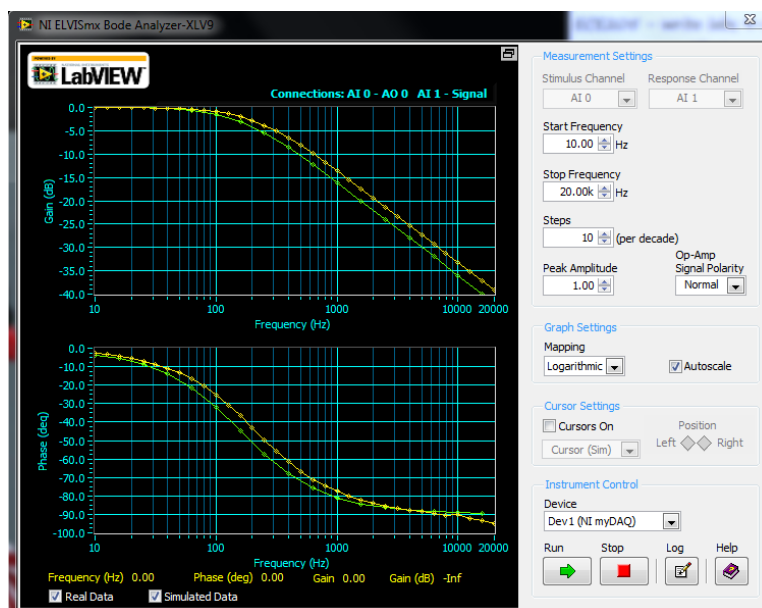


Figure 6: First order circuit Bode Plot (simulated and actual)

### PART III: Frequency Response of a Second Order System

1. In this part, you will generate the actual and simulated Bode plot of a second order system.
2. Construct the circuit in Figure 7 on your breadboard with  $L = 33 \text{ mH}$ ,  $C = 0.01 \mu\text{F}$  and R is the **10 k $\Omega$  variable resistor**.
3. Connect the function generator and Channel 0 of the oscilloscope to the input. Connect Channel 1 of the oscilloscope across the capacitor to measure the output voltage.

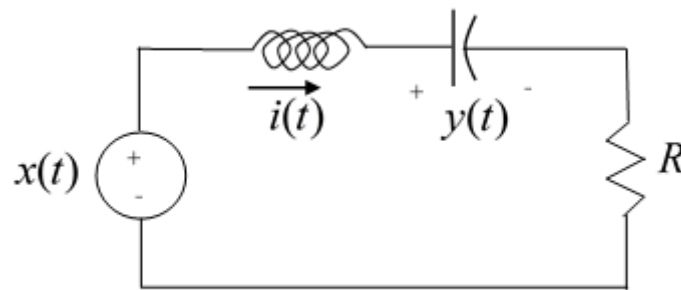


Figure 7: Second order system (RLC Circuit)

4. For this system the transfer function is  $H(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

where  $K$  is the static gain,  $\zeta$  is the damping ratio, and  $\omega_n$  is the natural frequency.

5. This circuit is a **lowpass filter** with a gain of one. However, since it is second order there is a **resonant frequency** where the amplitude of the output is maximum at a small peak in the Bode plot.
6. Start the MyDAQ function generator and oscilloscope. Set the input to **1 kHz square wave** with **2.0 V peak to peak amplitude** and a **1.0 V offset**. Set the oscilloscope to **100  $\mu$ s/div** and both channels to **1V/div**. You will need an **Edge** trigger with a level of **0.1 V**.
7. Run the function generator and oscilloscope and adjust the variable resistor so that there is a **50% overshoot** for a step response (see Figure 8). Do not change the variable resistor anymore but remove it and measure the resistance for use in the Multisim simulation. Then insert the resistor back into the circuit on the breadboard.



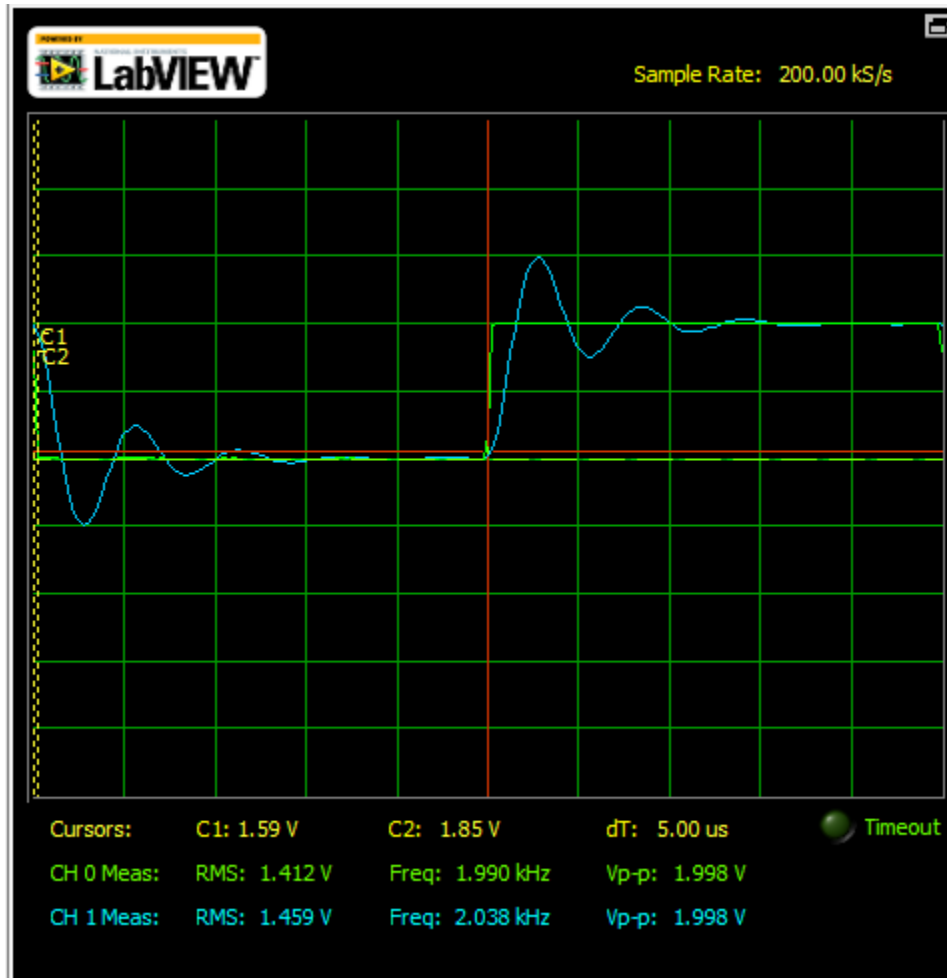


Figure 9: Second order system step response with 50% overshoot

8. Start Multisim and create a new NI MyDAQ design with the inductor, capacitor and resistor with the measured value of the potentiometer. Next, make similar connections to those for the circuit in the Multisim schematic in PART II.
9. Generate the Bode plot for the simulated and actual circuit and the results should look similar to Figure 10. The Bode plot for the actual system is correct but the magnitude and phase plot for the simulated system are actually a little bit off because the internal winding resistance of the inductor and capacitor and this has not been considered in the simulated design. In addition, there is a bug in the simulated phase plot that should be ignored.
10. Use the Bode plot to estimate the cutoff frequency, resonant frequency and maximum amplitude in absolute and decibel measurements.

11. You should include this graph output in your memo.

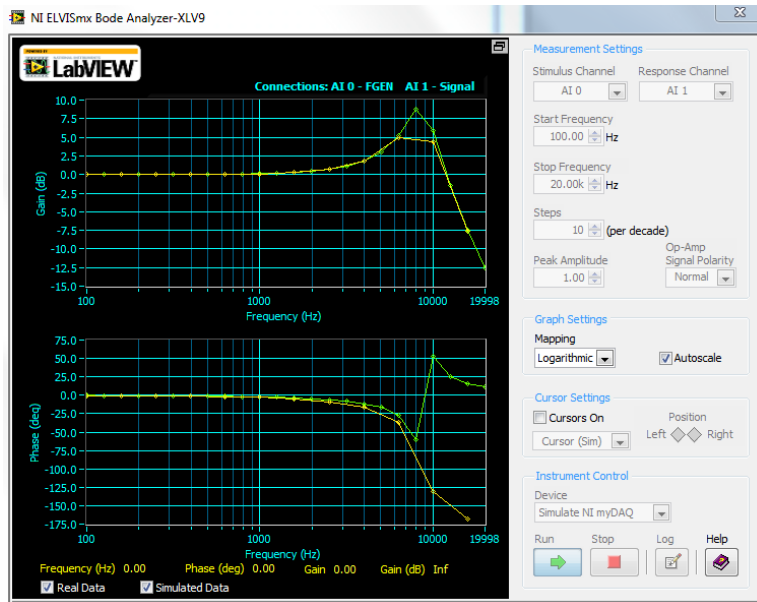


Figure 10: Second order circuit Bode Plot (simulated and actual)

#### PART IV: AC Analysis of a Second Order System in Multisim

1. In this part, you will build the series RLC circuit in Multisim where the value of the resistor is the measured value of the potentiometer. The source should be an **AC\_VOLTAGE** and keep the default values. Your final schematic should look similar to Figure 11.

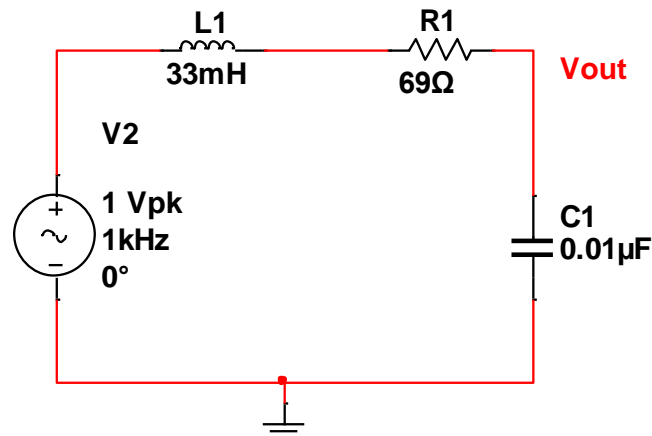


Figure 11: Second order system with AC source in Multisim



2. Click Simulate -> Analyses -> AC Analysis. Set the start frequency to **100 Hz** and the stop frequency to **20 kHz**.
3. Go to the output tab and select the voltage across the output to add to the right column. Click simulate.
4. Estimate the resonant frequency and maximum amplitude from this plot.
5. Compare and contrast these results to those found in Part III.
6. **You should include a screenshot of this figure in the lab memo submission.**

PART V: Analysis of Bode Plots

Bodes plots are useful for determining the steady state output of a system given the input.

There is nothing to turn in for this part, but you are required to know this!

For system A, whose frequency response is displayed in the Bode plot in Figure 12, verify the following steady state outputs for the given steady state inputs. *If you cannot, please ask for help!*

$$x(t) = 3\cos(10t) \rightarrow y_{ss}(t) \approx 150\cos(10t - 100^\circ)$$

$$x(t) = 5\sin(40t + 50^\circ) \rightarrow y_{ss}(t) \approx 1.6\sin(40t - 130^\circ)$$

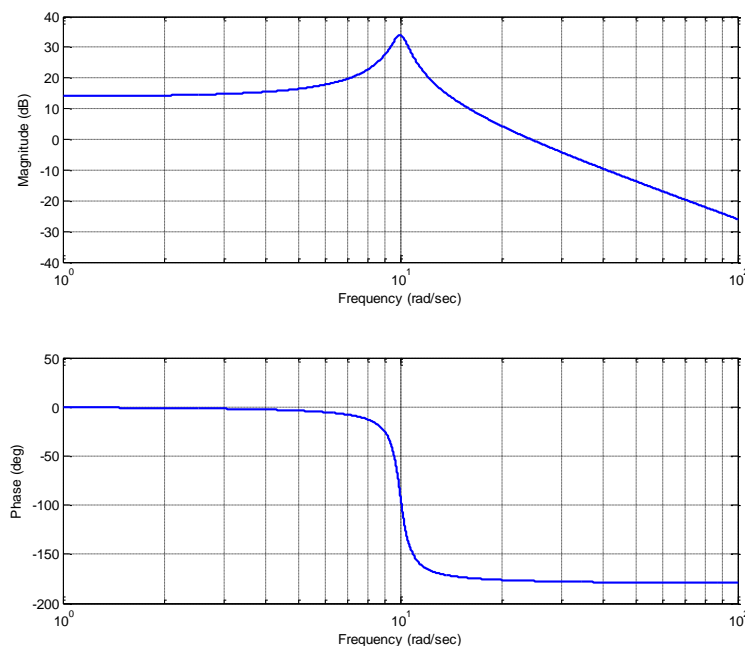


Figure 12: Frequency response of system A



For system B, whose frequency response is displayed in the Bode plot in Figure 13, verify the following steady state outputs for the given steady state inputs. *If you cannot, please ask for help!*

$$x(t) = 50 \sin(10t) \rightarrow y_{ss}(t) \approx 0.63 \sin(10t + 40^\circ)$$

$$x(t) = 100 \cos(200t + 35^\circ) \rightarrow y_{ss}(t) \approx 4 \cos(200t - 5^\circ)$$

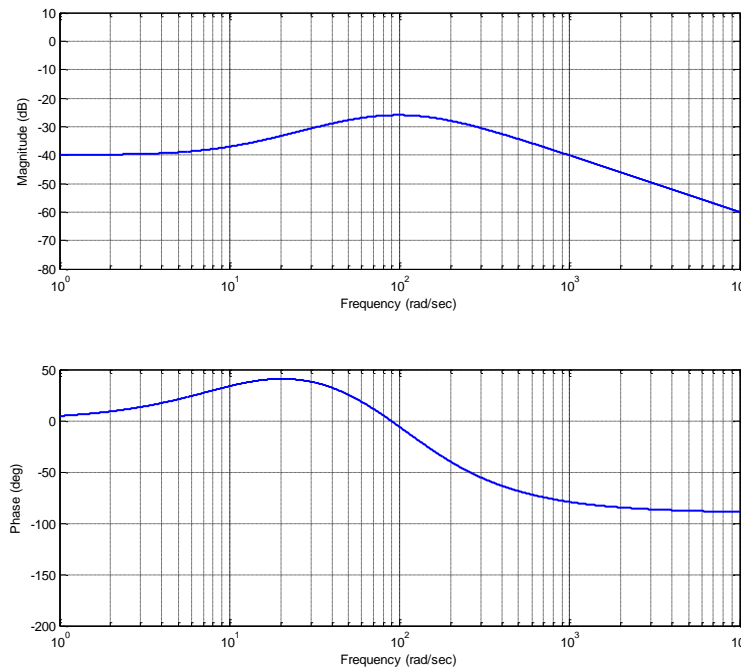


Figure 13: Frequency response of system B

### **Submission Requirements:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- A header at the beginning with *Date, To, From, Subject*
- Written in first person from you



- Written with minimal spelling and grammar errors
- *Purpose, procedure, results* and *conclusions* of the laboratory experiment
  - The very first sentence is the purpose and must explain why you are doing the experiment
  - The *procedure* should be very short, a high level summary of what you did for each part. The *procedure* should be two paragraphs at the most.
  - The *results* should include the tables and figures and results of the data collected and your observations. The results cannot be pages of figures and tables only, it **must** include text that references the figures and explains what they are to the reader.
  - The *conclusions* must be the end and it should be a summary of what you did, what you learned, what you observed and recommendations
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text. **DO NOT SUBMIT SEPARATE FILES**, images must be pasted into the Word document at the appropriate place with respect to the text. **DO NOT** just insert a bunch of figures and tables at the end of the document.
- Also, you must include a statement at the end of your memo that states that this submission is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results. This discussion may be a part of the results and/or conclusions sections of the memo

**ECE-205 Lab 10****Active Filters****Overview:**

In lab 9, you built two passive low pass filters and generated the Bode plots. A passive filter has a gain of one and is composed of only resistors, inductors and capacitors. In this lab you will build an active low pass filter, active high pass filter and an all pass filter. An active filter may have a gain greater than one and is composed of operational amplifiers in addition to the passive circuit elements. When these three components are cascaded together, they create a broadband bandpass filter. Finally, you will construct the magnitude portion of the Bode plot for each state of the filters and then put them together.

**Equipment:**

- 3 - TL072 operational amplifiers
- 2 - 1  $\mu\text{F}$  capacitors
- 4 - 1 k $\Omega$  resistors
- 3 - 10 k $\Omega$  potentiometers

**Theory:**

The bandpass filter in this lab has three parts, shown in the different shaded regions in Figure 1. The equivalent block diagram is shown below the circuit, so that you can match elements of the block diagram with the circuit subsystems. The first subsystem is the low pass filter and it has a cutoff frequency of  $1/(R_a C)$  rad/s and a gain of  $R_a/R$ . The second subsystem is the high pass filter and it has a gain of  $R/R_b$  and a cut off frequency of  $1/(R_b C)$  rad/s. The last subsystem is an all pass filter which just adjusts the gain of the system.

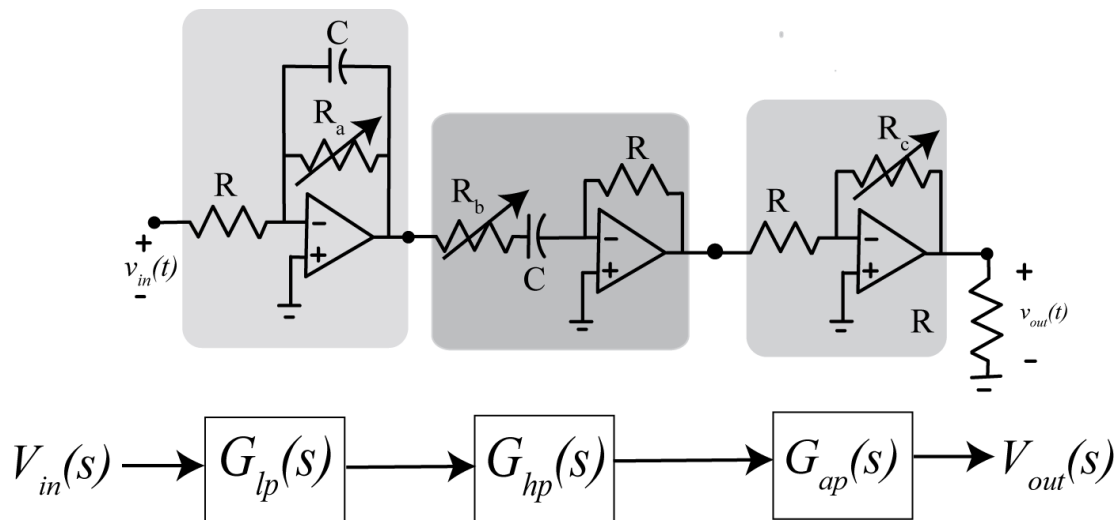


Figure 1: Broadband bandpass filter schematic and block diagram

**Prelab:**

1. Build the circuit in Figure 2 in MultiSim and run an AC analysis from 10 Hz to 100 kHz, what is the cut off frequency and gain? You should include the gridlines and reverse the plot to have a white background. Include the plot in your prelab submission. Review the end of Lab 9 if you don't recall how to run an AC analysis in MultiSim.

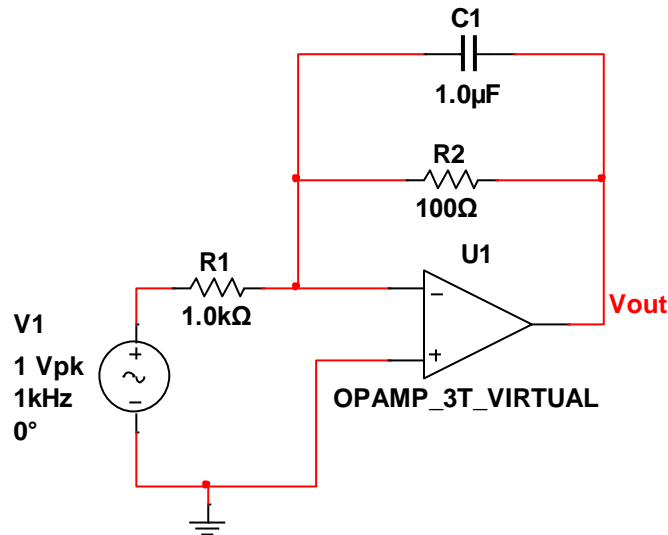


Figure 2: Active lowpass Filter

2. Build the circuit in Figure 3 in MultiSim and run an AC analysis from 10 Hz to 100 kHz, what is the cut off frequency and gain? You should include the gridlines and reverse the plot to have a white background. Include the plot in your prelab submission.

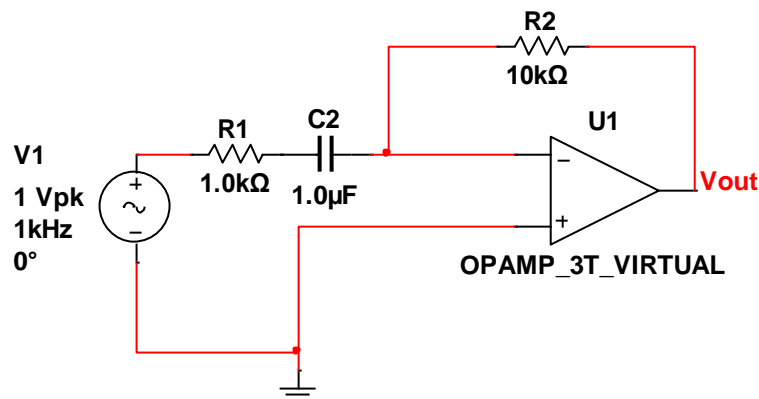


Figure 3: Active highpass Filter

3. Build the circuit in Figure 4 in MultiSim and run an AC analysis from 10 Hz to 100 kHz. What are the two cut off frequencies and the gain? You should include the gridlines and reverse the plot to have a white background. Include the plot in your prelab submission.

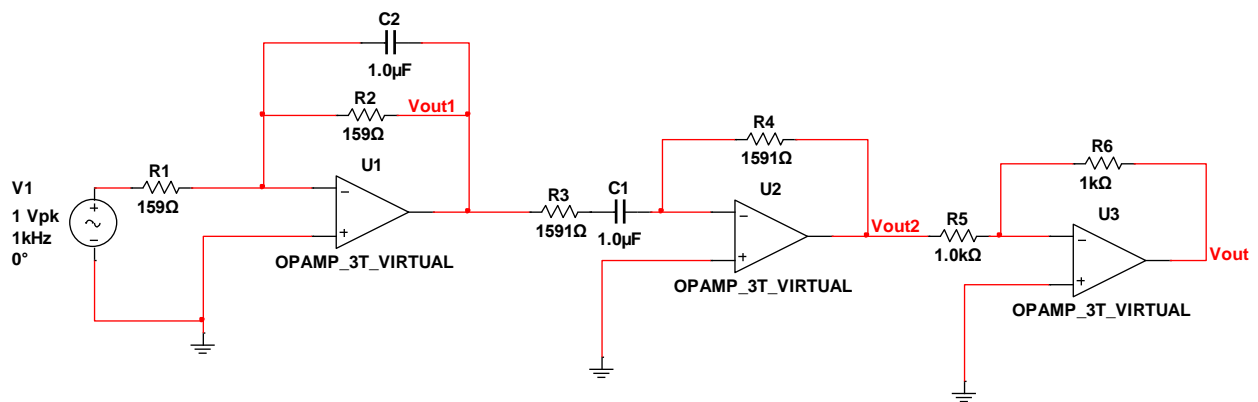


Figure 4: Broadband bandpass filter

- Adjust the component values so that the bandpass filter has a gain of 10 and a lower cut off frequency of 700 Hz and an upper cut off frequency of 10 kHz. You should include the gridlines and reverse the plot to have a white background. Include the plot in your prelab submission.
- Finally, include the final schematic for the broadband bandpass filter in your prelab submission. **In summary, you should print out 4 plots and one schematic for submission.**

### **Procedure:**

#### **PART I: Build the lowpass filter**

- Turn the breadboard so that the positive red buss is on the top.
- Connect a green wire between the two blue busses to create a common ground.
- Place the 3 TL072 op amps equally spaced down the middle of the breadboard ditch with the dot (notch) to the left.
- Connect pin 4 of all three op amps to the bottom red buss which represents **-Vcc (-15 V)**.
- Connect pin 8 of all three op amps to the top red buss which represents **+Vcc (+15V)**.
- Connect pin 3 of all three op amps to the blue buss which represents **AGND**.
- Connect the **+15 V** on the power supply to the top red buss. Connect the **-15 V** on the power supply to the bottom red buss. Connect the **AGND** on the power supply to the blue buss.
- Build the lowpass filter in Figure 5 on the first TL072 chip. All of the resistors in this lab are 1 kΩ and all of the capacitors are 1 μF.
- The input to this circuit should be the function generator (**AO 0, AO AGND**). Set the function generator to a **10 Hz sine wave** with a **2 V peak to peak** amplitude and 0 V offset.
- The input should be measured by Channel 0 of the oscilloscope (**AI 0+, AI 0-**). The output should be measured at pin 1 (**AI 1+, AI 1-**) on the op amp.



11. Set the time base on the oscilloscope to **20 ms/div**. Set Channel 0 to **500 mV/div** and set Channel 1 to **100 mV/div**. Set the trigger to **Edge** with the **Channel 0** source.

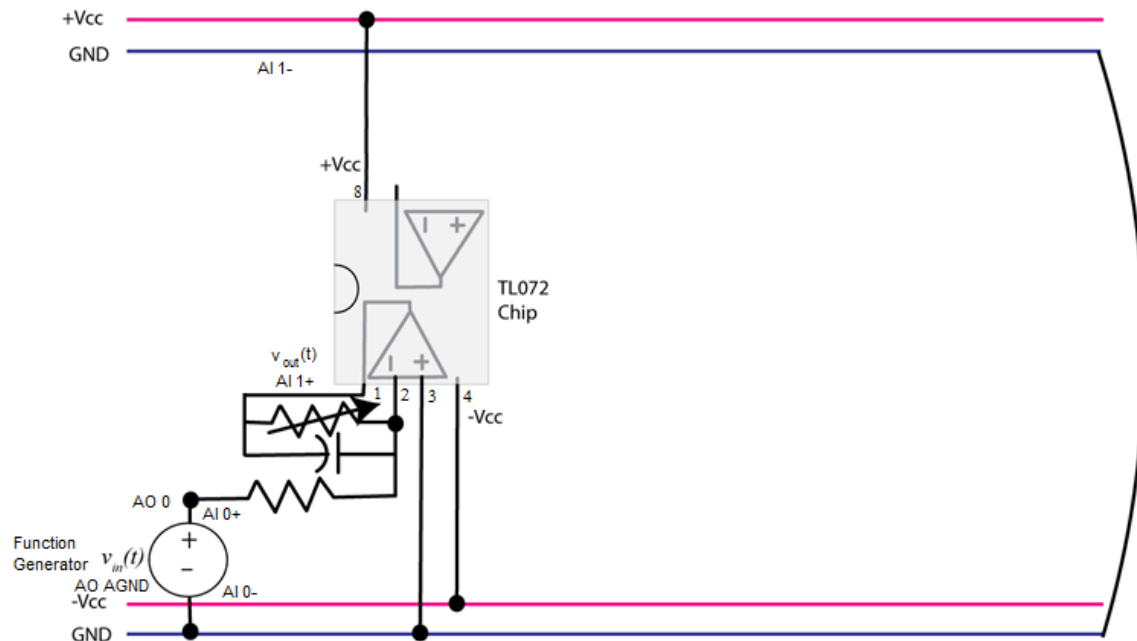


Figure 5: Active lowpass filter schematic

12. Adjust the variable resistor so the output has an amplitude of approximately **0.6 volts peak to peak** or a **gain of 0.3**.
13. The transfer function for this lowpass filter can be shown to be

$$G_{lp}(s) = \frac{-K_{low}\omega_{low}}{s + \omega_{low}}$$

$$\text{where } K_{low} = \frac{R_a}{R} \text{ and } \omega_{low} = \frac{1}{R_a C}$$

PART II: Construct the Bode amplitude plot for the lowpass filter

1. In this part you will construct the magnitude portion of the Bode plot for the lowpass filter.
2. Create a data table in Excel with the columns frequency (Hz), input (mV), output (mV).
3. For input frequencies of 100, 200, 400, 600, 700, 800, 1000, 1500, 2000, 2500, and 3000 Hz, keep the input amplitude at 1V (2 Vpp) and use the cursors to measure the amplitude of the output as the frequency is varied. Record the output amplitude on the Excel data table. Note that as your input frequency increases, you will need to decrease the Time/Div setting in order to keep only one or two periods of the waveform on the screen. Also, try and keep Channel 1 between **20 mV/div** and **100 mV/div** to keep the waveform as large as possible on the screen.
4. Download the **process\_data\_low.m** file from the Angel course Lab 10 folder. In this file, each row corresponds to a frequency. The three entries in the row are the frequency

(measure in Hz), the amplitude of the input signal (measured in mV), and the amplitude of the output signal (measured in mV).

5. Copy the data from Excel into the **measured** matrix variable.
6. In the MATLAB command window type **data = process\_data\_low;**
7. Now we will fit the data to our model using the program **model\_low.m**. The arguments to this file are the data array from part b, the estimated amplitude of the gain, and the estimated value of the cutoff frequency.
8. In the Matlab command window type **model\_low(data, 0.3, 2\*pi\*500);**
9. You should get results similar to that shown in Figure 6. Click on the Datatip icon (just below Desktop) to identify points on the graph. Once you have identified the maximum value, you may have to use a right click and create a new data tip (it usually helps if you right click when you are on a current datatip). You can fine tune your datatip locations by using the cursor keys on your keyboard. Your data points should be from the peak value to a value 3 dB lower. Include this graph in your memo. Be sure your graph has two datatip points.

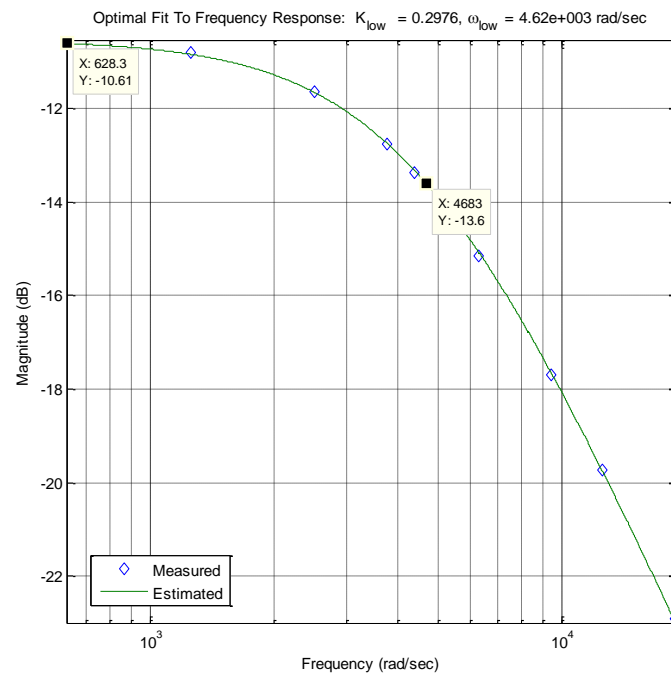


Figure 6: lowpass filter with a bandwidth of approximately 4683 rad/sec (745 Hz) and gain of -10.61 dB (0.295)

### PART III: Construct the Bode amplitude plot for the lowpass filter in Multisim

1. Create a new NI myDAQ design in Multisim and place the NI ELVISmx Bode Analyzer on the sheet and wire it as shown in Figure 7.

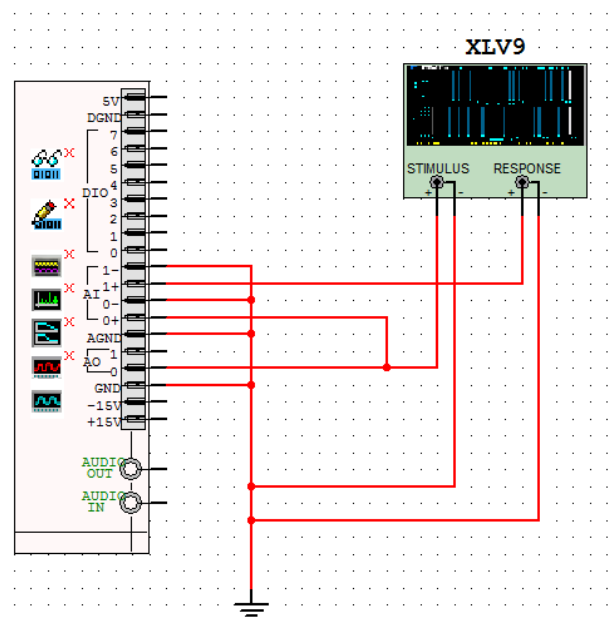


Figure 7: NI myDAQ design in Multisim

2. Open up the Bode Analyzer and change the start frequency to 100 Hz and the stop frequency to 20 kHz.
3. Change the Device to the NI myDAQ.
4. Run the simulation and the results should look similar to Figure 8.
5. Compare and contrast this result to the MATLAB result in PART II.
6. **Capture the result for inclusion in your lab memo submission.**



Figure 8: NI myDAQ design lowpass filter Bode Analyzer

PART IV: Construct the highpass filter

1. Now build the highpass filter as shown in Figure 9 on the second TL072 op amp. Do not connect the two subsystems yet.
2. Set the function generator to a **2 V peak to peak sine wave** with **0 V DC offset** at **1 kHz**.
3. Connect function generator and Channel 0 of the oscilloscope to the input of the high pass filter. Connect Channel 1 of the oscilloscope to pin 1 of the high pass filter to measure the output.
4. Set the time base of the oscilloscope to **200  $\mu\text{s}/\text{div}$**  and set both channels to **100 mV/div**. Set the trigger type to **Edge** and the **Chan 0 Source**.
5. Adjust the variable resistor that the output signal has approximately **0.8 volts peak to peak** or a **gain of 0.4**.

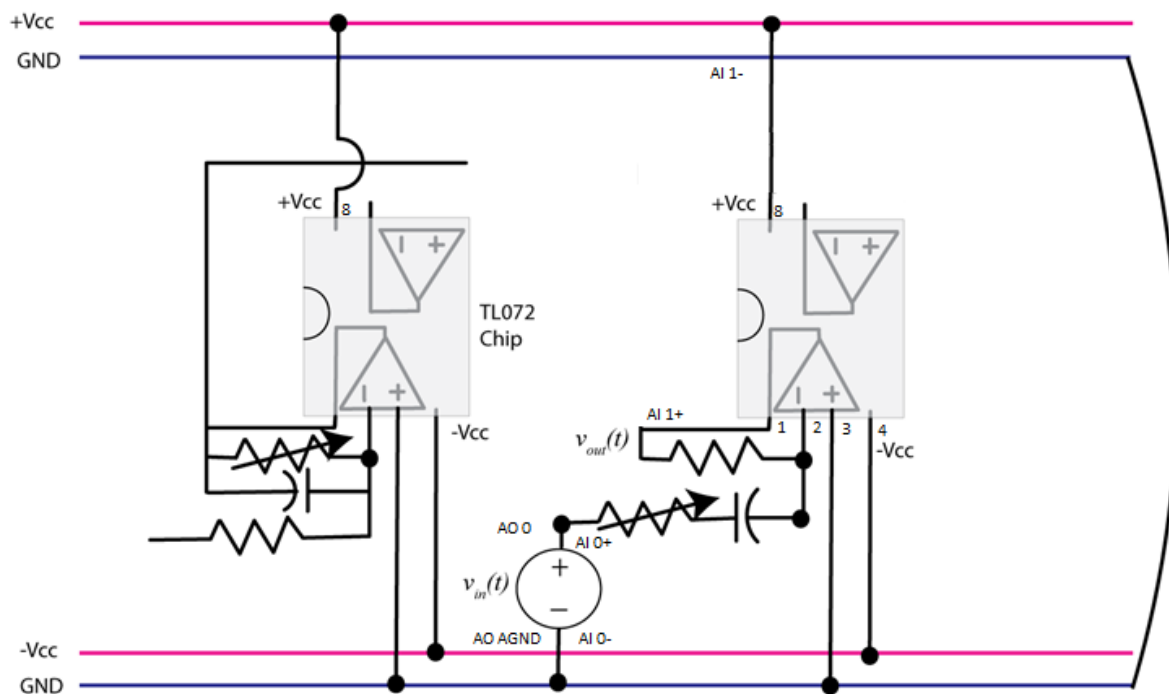


Figure 9: lowpass and highpass subsystems

6. The transfer function for this subsystem can be shown to be

$$G_{hp}(s) = \frac{-K_{high}s}{s + \omega_{high}}$$

$$\text{where } K_{high} = \frac{R}{R_b} \text{ and } \omega_{high} = \frac{1}{R_b C}$$

PART V: Construct the Bode amplitude plot for the highpass filter

1. Create a new sheet in the Excel workbook for the high pass filter and create the same columns frequency (Hz), input amplitude (mV) and output amplitude (mV).



2. For input frequencies of 10, 25, 50, 75, 100, 200, 300, 400, 500, 750, and 1000 Hz, keep the input amplitude at 2 V peak to peak and use the cursors to measure the amplitude of the output signal and record it on the Excel sheet. You may need to vary the timebase and volts/division in order to display one or two periods of the waveform as large as possible on the screen.
3. Copy the program **process\_data\_low.m** and rename the file, **process\_data\_high.m**. and enter your data into this new file for the **measured** matrix. Do not just reuse your previous file.
4. In the MATLAB command window type **data = process\_data\_high;**
5. Download the **model\_high.m** script from the Angel course Lab 10 folder. The arguments to this file are the data array, the estimated amplitude of the gain, and the estimated value of the cutoff frequency.
6. In the MATLAB command window type **model\_high(data, 0.4, 2\*pi\*200);** You should get results similar to that shown in Figure 10. Click on the Datatip icon (just below Desktop) so identify points on the graph. Once you have identified the maximum value, you may have to use a right click and create a new data tip(it usually helps if you right click when you are on a current datatip). You can fine tune your datatip locations by using the cursor keys on your keyboard. Your data points should be from the peak value to a value 3 dB lower.
7. **Include this graph in your memo. Be sure your graph has two datatip points.**

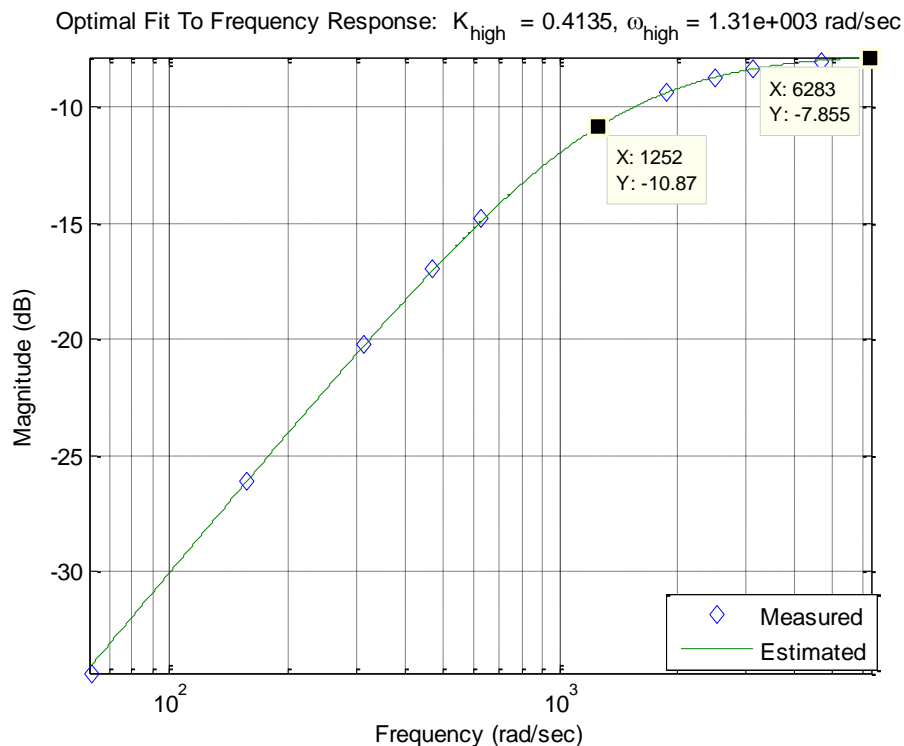


Figure 10: Highpass filter with a cutoff frequency of approximately 1.31 krad/s (208.5 Hz) and gain of -7.855 dB (0.405).

PART VI: Construct the Bode amplitude plot for the highpass filter in Multisim

1. Repeat PART III for the high pass filter to get the Bode amplitude plot using the Bode analyzer in Multisim.
2. You can actually use the same Multisim schematic that you created in Figure 7.
3. Run the simulation and compare and contrast the results with the MATLAB results.
4. **Capture the result for inclusion in your lab memo submission.**

PART VII: Construct the all pass filter (inverting amplifier)

1. Construct the gain subsystem, as shown in Figure 11.
2. Adjust the variable resistor so that for an input **sine wave** with a **2 Vpp** amplitude at **500 Hz**, the output is approximately **2V pp** or the system has a **gain of 1**.

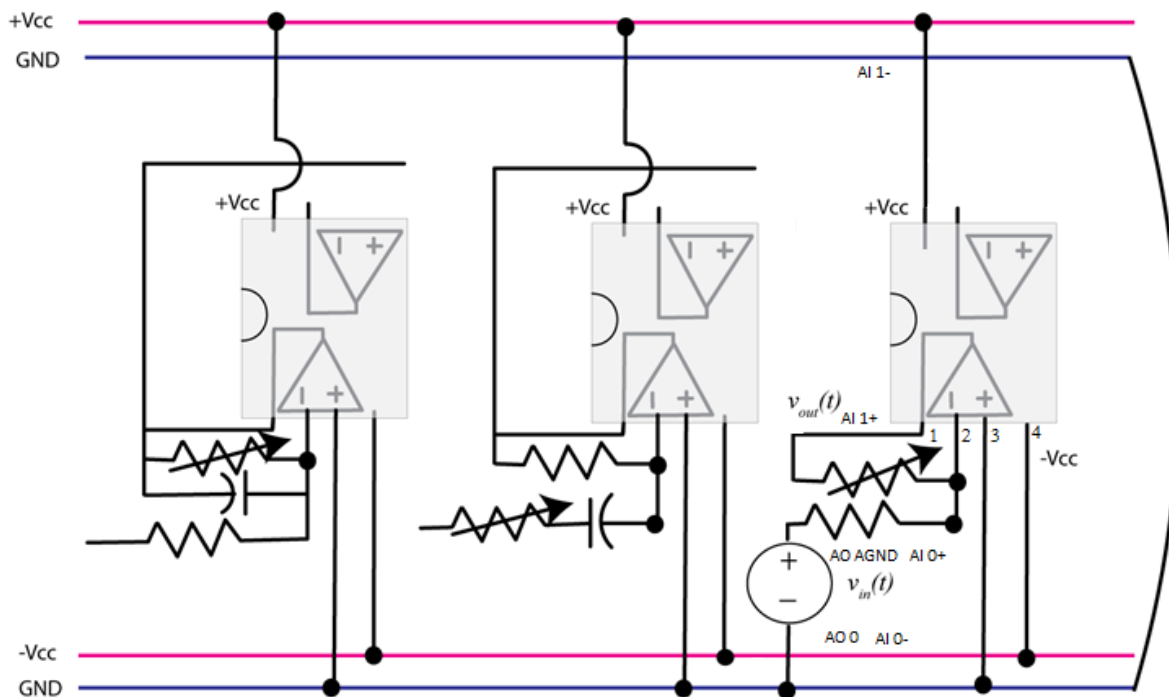


Figure 11: Lowpass, highpass, and all pass subsystems.

3. The transfer function for this subsystem can be easily shown to be

$$G_{ap}(s) = -K_{ap}$$

$$\text{where } K_{ap} = \frac{R_c}{R}$$

PART VIII: Construct the Bandpass filter

1. Now you will use 2 wires to connect all of the of the subsystems together as shown in Figure 12.

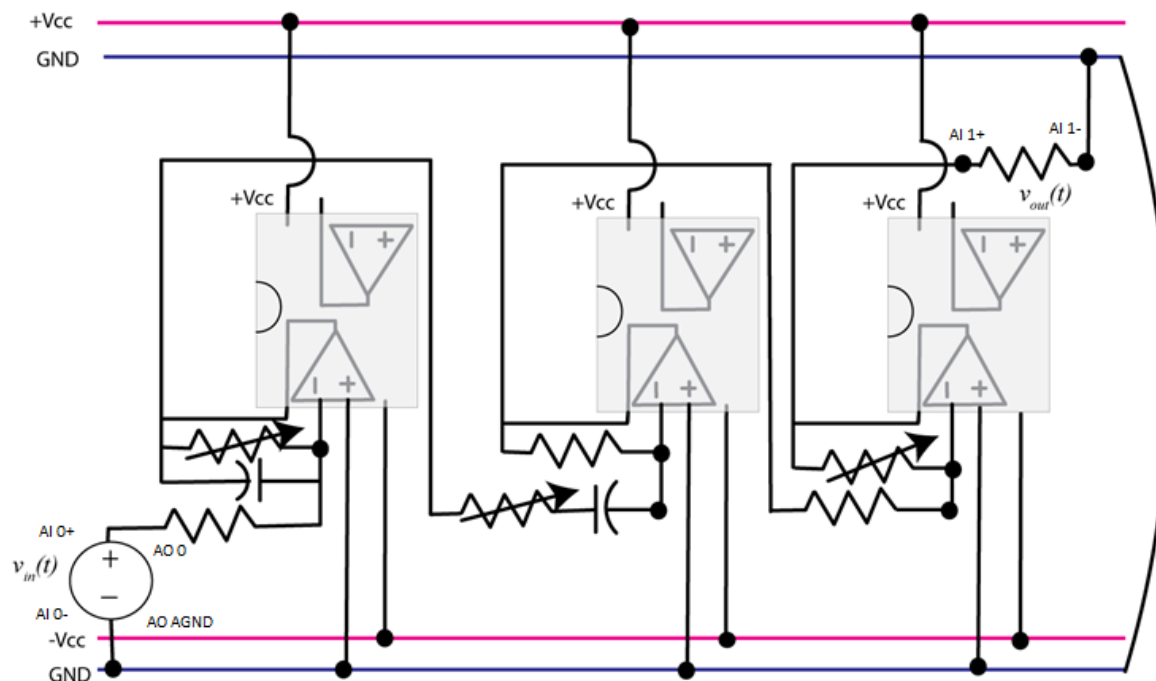


Figure 12: Broadband Active Bandpass filter

- The transfer function for this system is then the product of the transfer functions of the subsystems,

$$G_{bp}(s) = G_{lp}(s)G_{hp}(s)G_{ap}(s) = \left( -\frac{K_{low}\omega_{low}}{s + \omega_{low}} \right) \left( -\frac{K_{high}s}{s + \omega_{high}} \right) (-K_{ap})$$

We can simplify this as

$$G_{bp}(s) = -\frac{Ks}{s^2 + (\omega_{low} + \omega_{high})s + \omega_{low}\omega_{high}}$$

- Since we are only interested in the magnitude, we can ignore the negative sign (but you will see it on the output since it will be out of phase with the input).
- Finally, we will construct the magnitude portion of the Bode plot for this simple bandpass filter.
- For input frequencies of 10, 15, 25, 40, 50, 75, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500 and 2000 Hz, keep the input amplitude at **2 Vpp** and measure the amplitude of the output signal.
- Initially set Channel 0 to **200 mV/div** and Channel 1 to **10 mV/div** and the timebase to **20 ms/div**. However, you will have to change the timebase and volts/division as the frequency is varied.
- Copy the program **process\_data\_high.m** onto a new file **process\_data\_bandpass.m**, and enter your data into this new file. Do not just reuse your previous file.
- In the MATLAB command window type **data = process\_data\_bandpass;**



9. Download the **model\_bandpass.m** file from the Angel course Lab 10 folder. The arguments to this file are the data array, the estimated amplitude of the gain, and the estimated value of the high and low cutoff frequencies.
10. In the Matlab command window type

```
model_bandpass( data, 0.3*0.4, 2*pi*500, 2*pi*200);
```

11. You should get results similar to that shown in Figure 13. Note that your peak amplitude may be different depending on how much you could adjust the gain on the final stage. At this point we mostly are looking for the same general shape.
12. Click on the Datatip icon (just below Desktop) so identify points on the graph. Once you have identified the maximum value, you may have to use a right click and create a new data tip (it usually helps if you right click when you are on a current datatip). You can fine tune your datatip locations by using the cursor keys on your keyboard. Your data points should be from the peak value to a value 3 dB lower on both sides.
13. **Include this graph in your memo. Be sure your graph has three datatip points.**

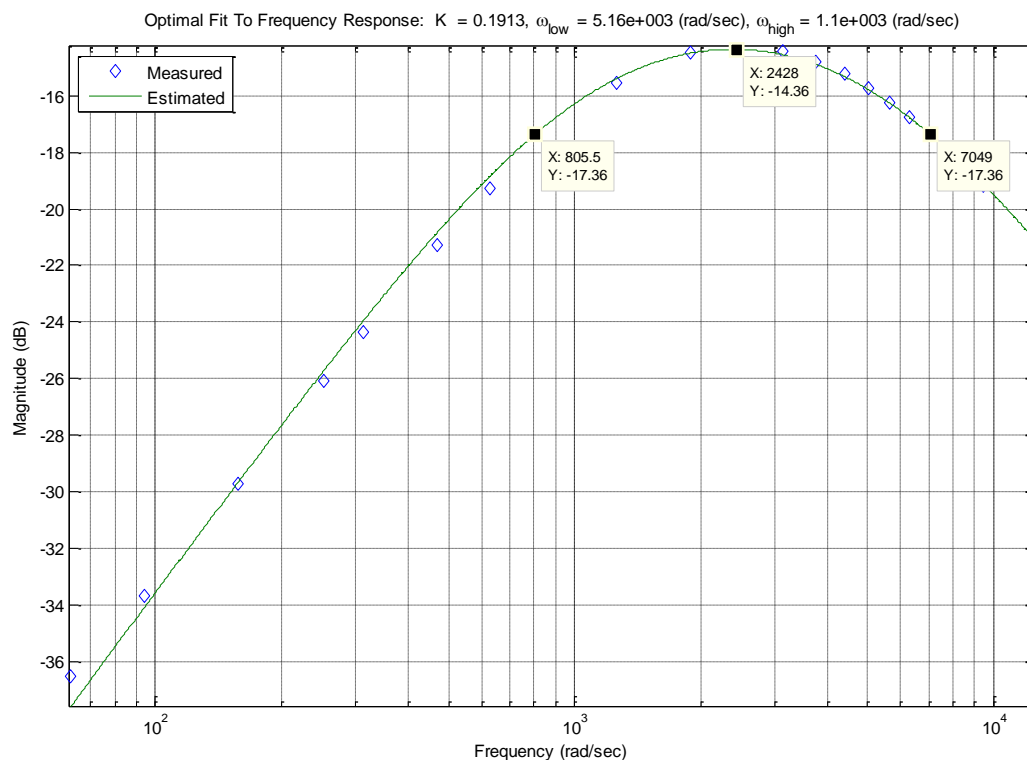


Figure 13: Bandpass filter with an approximate bandwidth of 6244 rad/sec (993 Hz) and Gain of -14.36 dB (0.191)

**PART IX: Construct the Bode amplitude plot for the bandpass filter in Multisim**

1. Repeat PART III for the bandpass filter to get the Bode amplitude plot using the Bode analyzer in Multisim.





2. You can actually use the same Multisim schematic that you created in Figure 7.
3. Run the simulation and compare and contrast the results with the MATLAB results.
4. **Capture the result for inclusion in your lab memo submission.**

### **Submission Requirements:**

The lab memo should be submitted to the instructor via the Angel Course Drop Box by midnight on **Sunday**. After midnight on Sunday, the memo is late and incurs a **20% penalty per day**. If it is not submitted by the beginning of the next lab session, the grade is a **zero**. At a minimum it should include the following:

- Typewritten, 12 point font
- A header at the beginning with *Date, To, From, Subject*
- Written in first person from you
- Written with minimal spelling and grammar errors
- *Purpose, procedure, results and conclusions* of the laboratory experiment
  - The very first sentence is the purpose and must explain why you are doing the experiment
  - The *procedure* should be very short, a high level summary of what you did for each part. The *procedure* should be two paragraphs at the most.
  - The *results* should include the tables and figures and results of the data collected and your observations. The results cannot be pages of figures and tables only, it **must** include text that references the figures and explains what they are to the reader.
  - The *conclusions* must be the end and it should be a summary of what you did, what you learned, what you observed and recommendations
- The entire memo should be concise and to the point.
- All required figures and files generated using the NI myDAQ or MATLAB or MultiSim with number and caption and they should be referenced in the text. **DO NOT SUBMIT SEPARATE FILES**, images must be pasted into the Word document at the appropriate place with respect to the text. **DO NOT** just insert a bunch of figures and tables at the end of the document.
- Also, you must include a statement at the end of your memo that states that this submission is your own work.
- The discussion should include a compare and contrast of the theoretical results or nominal results to the actual results. This discussion may be a part of the results and/or conclusions sections of the memo

## EQUATION SHEET

$$L\{\delta(t)\} = 1$$

$$L\{u(t)\} = \frac{1}{s}$$

$$L\{tu(t)\} = \frac{1}{s^2}$$

$$L\left\{\frac{t^{m-1}}{(m-1)!}u(t)\right\} = \frac{1}{s^m}$$

$$L\{e^{-at}u(t)\} = \frac{1}{s+a}$$

$$L\{te^{-at}u(t)\} = \frac{1}{(s+a)^2}$$

$$L\left\{\frac{t^{(m-1)}}{(m-1)!}e^{-at}u(t)\right\} = \frac{1}{(s+a)^m}$$

$$L\{\cos(\omega_0 t)u(t)\} = \frac{s}{s^2 + \omega_0^2}$$

$$L\{\sin(\omega_0 t)u(t)\} = \frac{\omega_0}{s^2 + \omega_0^2}$$

$$L\{e^{-\alpha t} \cos(\omega_0 t)u(t)\} = \frac{s + \alpha}{(s + \alpha)^2 + \omega_0^2}$$

$$L\{e^{-\alpha t} \sin(\omega_0 t)u(t)\} = \frac{\omega_0}{(s + \alpha)^2 + \omega_0^2}$$

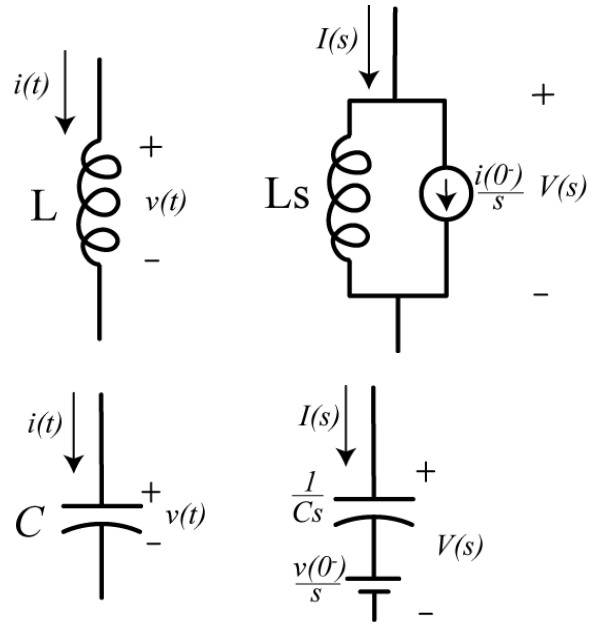
$$L\left\{\frac{dx(t)}{dt}\right\} = sX(s) - x(0^-)$$

$$L\left\{\frac{d^2x(t)}{dt^2}\right\} = s^2X(s) - sx(0^-) - \dot{x}(0^-)$$

$$L\{x(t-a)\} = e^{-as}X(s)$$

$$L\{e^{-at}x(t)\} = X(s+a)$$

$$L\left\{x\left(\frac{t}{a}\right), a > 0\right\} = aX(as)$$



### Second Order System Properties

$$\text{Percent Overshoot: } P.O. = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\%$$

$$\text{If } \beta = \frac{PO^{max}}{100} \text{ then } \zeta = \frac{-\ln(\beta)}{\pi \sqrt{1 + \left(\frac{-\ln(\beta)}{\pi}\right)^2}}$$

$$\theta = \cos^{-1}(\zeta) \text{ Time to Peak:}$$

$$T_p = \frac{\pi}{\omega_d}, \quad \omega_d = \omega_n \sqrt{1 - \zeta^2}$$

$$2\% \text{ Settling Time: } T_s = \frac{4}{\zeta\omega_n} = 4\tau$$

**Initial Value Theorem:** If  $x(t) \leftrightarrow X(s)$

$$\lim_{t \rightarrow 0^+} x(t) = \lim_{s \rightarrow \infty} sX(s)$$

**Final Value Theorem:** If  $x(t) \leftrightarrow X(s)$

$$\lim_{t \rightarrow \infty} x(t) = \lim_{s \rightarrow 0} sX(s)$$

## Bode Plot Building Blocks Summary

**Constant Terms:**  $H(s) = K$     *Note:* Magnitude and Phase are constants

**Integrators and Differentiators:**  $H(s) = s^n$

*Magnitude:* 20 n dB/decade,    *Phase:*  $n90^\circ$

*Note:* the point  $(\omega = 1, 0 \text{ dB})$  is on the Bode plot

**Simple Poles and Zeros:**  $H(s) = (\tau s + 1)^n$

|                           | <i>Magnitude</i>                               | <i>Phase</i>                                                                                                  |
|---------------------------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| $\omega \approx 0$        | 0 dB                                           | $\square H(j\omega) \approx 0^\circ$ for $\omega \leq \frac{0.1}{\tau}$ (one decade before $\frac{1}{\tau}$ ) |
| $\omega = \frac{1}{\tau}$ | $3n \text{ dB}$                                | $\square H(j\omega) \approx n45^\circ$                                                                        |
| $\omega \approx \infty$   | <i>slope</i> $\approx 20n \text{ dB / decade}$ | $\square H(j\omega) \approx n90^\circ$ for $\omega \geq \frac{10}{\tau}$ (one decade after $\frac{1}{\tau}$ ) |

**Complex Conjugate Poles and Zeros:**  $H(s) = \left( \frac{1}{\omega_n^2} s^2 + \frac{2\zeta}{\omega_n} s + 1 \right)^n$

|                         | <i>Magnitude</i>                               | <i>Phase</i>                                                                                               |
|-------------------------|------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| $\omega \approx 0$      | 0 dB                                           | $\square H(j\omega) \approx 0^\circ$ for $\omega \leq \frac{\omega_n}{10}$ (one decade before $\omega_n$ ) |
| $\omega = \omega_n$     | <i>depends on</i> $\zeta$                      | $\square H(j\omega) \approx n45^\circ$                                                                     |
| $\omega \approx \infty$ | <i>slope</i> $\approx 40n \text{ dB / decade}$ | $\square H(j\omega) \approx n180^\circ$ for $\omega \geq 10\omega_n$ (one decade after $\omega_n$ )        |

(Next page) Magnitude and phase of the frequency response for the transfer function  $G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$  for

$K = 10, \omega_n = 20,$  and  $\zeta = 0.01, 0.1, 0.25, 0.5, 0.75, 0.99$

