SIMON FRASER UNIVERSITY
School of Mechatronic Systems Engineering
MSE 381 – Feedback Control Course Project
Summer 2019

Project Description:

The purpose of this project is to gain an understanding of the real world applications of negative feedback control in electronic systems. All control design methods explored in lectures will be put to the test in this project in order to improve the behavior of an electronic system. Firstly, students will investigate the behavior of the open loop system which in this case is the simple LRC electronic circuit seen below in Figure 1. The students must transform this circuit into its Laplace transform, in order to analyze its behavior.

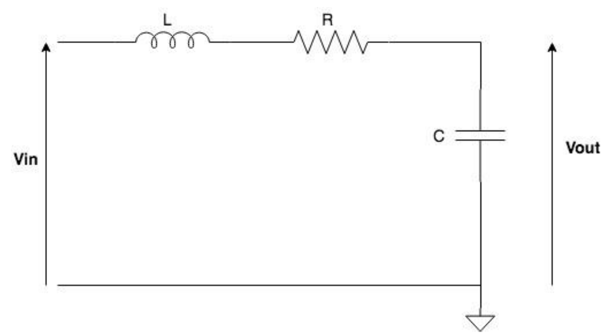

Figure 1. LRC Circuit

Students use MatLab code to analyze the open loop system, then use circuit simulator LTSpice to compare their results. In the next section, they get introduced to the circuit topologies necessary to achieve negative feedback and investigate the closed loop system without any controller.

In the controller section, the students must design a PID controller for the system in MatLab and then test out their design in the LTSpice file provided. They repeat this process for the analysis and design of the lead-lag compensator.

MSE 381 - Feedback Control Systems

Course Project

Department of Mechatronics Engineering
Simon Fraser University

1 Introduction

The purpose of this project is to gain an understanding of the real world applications of negative feedback control in electronic systems. All control design methods explored in lectures will be put to the test in this project in order to improve the behaviour of an electronic system.

2 Setup

Download the **MatLab** code and **LTSpice** files from Canvas, these will be used throughout the project. In each section the files used will be mentioned at the top. Please note that the LTSpice models work, so only component values should be edited. **All results and graphs from the following sections must be included in final report.**

3 System analysis

The first system explored is a simple LRC circuit as shown in Figure 1

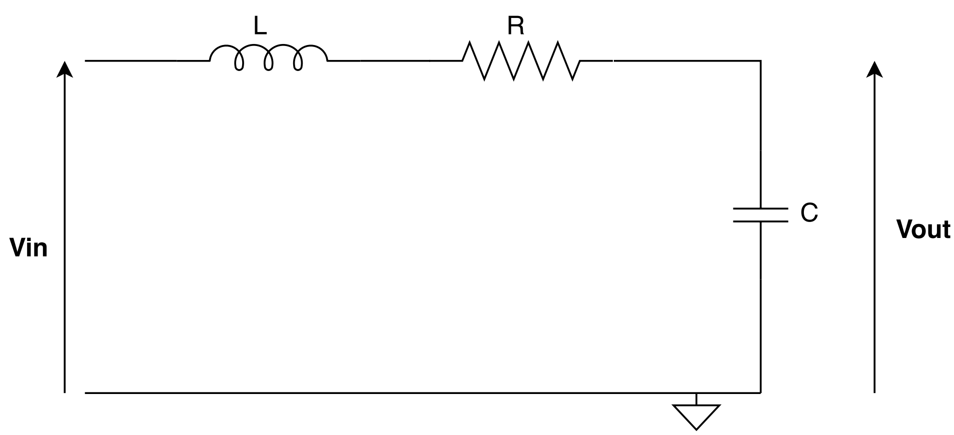

Figure 1: LRC circuit

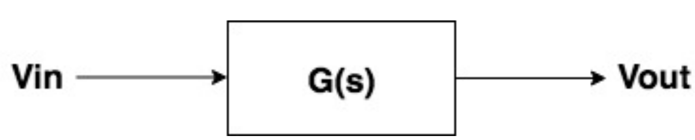

Figure 2: Transfer Function diagram

- As a first step establish what the transfer function G(s) of the system is, based on L R C, knowing that their impedances are $Z_L = Ls$, $Zc = \frac{1}{Cs}$ and $Z_R = R$.

G(s) =

3.1 Open Loop Testing

In this section you will use **openLoopTesting.m** and **LRC.m** - comment in/out the code that you need.

3.1.1 MatLab Simulation

- Model G(s) in MatLab using L = 100 mH, C = 47 uF, R = 10Ω. Code is provided, and make sure to use L R C variables rather than hard coding in case the system changes later. Check that the transfer function agrees with the one found in the previous section.
- Analyse this open loop system, using the **openLoopTesting.m** file provided. Plot the system's **step response, bode, nyquist diagrams** as well as its **root locus** and **poles**. Discuss the properties of this system. Is the system stable?
- Measure phase and gain margins from Bode diagram or by using margin command.

3.1.2 LTSpice Simulation

All LTSpice files work, only component values should be changed when specified. To run the simulation click the following icon, in the top left corner:

To see output voltage at a point in the circuit, click on the schematic at that point, and the result will be displayed in the signal window.

- Using LTspice **stepresponse** file run the step response of this circuit and compare it to the step response simulated in MatLab.
- Using the **frequencysweep** file run the frequency sweep on the system and compare the bode diagram simulated in MatLab.
- In MatLab and/or LTSpice, change the values of L R and C individually and describe the effect these changes have on the system.

As we have seen in this section, in this system we can change the values of the components to get the output step response that we desired. However in systems where we cannot do this (black box system), we have to design a negative feedback system around it in order to improve the system's performance, which we will explore in the next section.

4 Negative Feedback

Negative feedback is a process in which the output of the system is fedback to the input in order to improve the system's behaviour. Since we are implementing negative feedback on an electronic circuit we will firstly explore how this can be achieved before designing a controller. Figure 3 below, shows the block diagram for negative feedback without a controller.

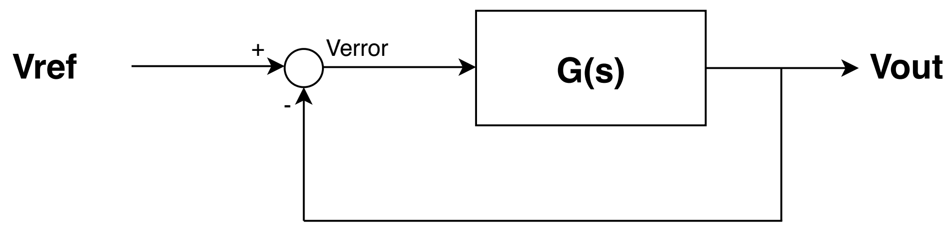

Figure 3: Negative Feedback System

4.1 Circuit Implementation

The circuit shown in Figure 4 can be used to produce an electronic feedback output.

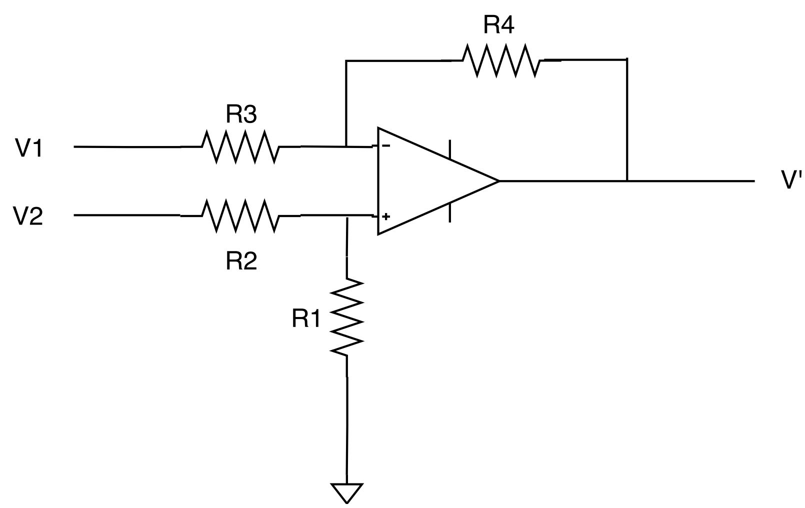

Figure 4: Negative Feedback in a circuit

- Using an ideal op amp, establish the relationship of V' to V1 and V2, if R1=R2=R3=R4. Therefore conclude on the values of V', V1 and V2 with respect to Vref, Vout and Verror shown in Figure 3.

V' =

- In MatLab only simulate the closed loop response of the system G(s), without a controller. In your report include the **step response** as well as the **closed loop poles**. Has the system improved by using feedback only, without a controller?

5 Controller Design

Putting it all together and including the controller K(s) that will be designed later, we obtain the block diagram seen in Figure 5, and the equivalent circuit implementation seen in Figure 6, where the unity gain buffer is used to buffer separate parts of the circuit to avoid impedance issues.

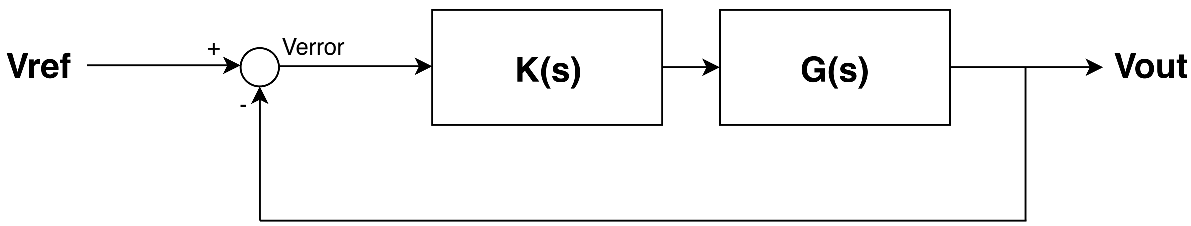

Figure 5: Negative Feedback System with controller - block diagram

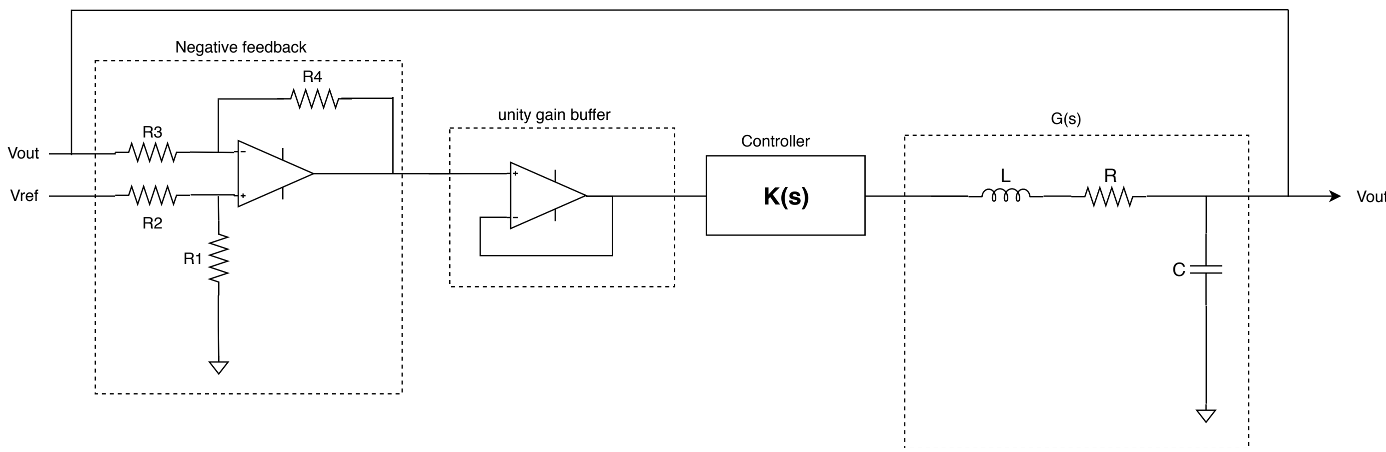

Figure 6: Negative Feedback System with controller - circuit implementation

In the next section we will design different controller K(s) and assess performance of the overall system, by using MatLab and then you will test it out your designs in LTSPice. The two types of controllers that we will look at are: PID controllers, and lead/lag controllers. The performance of these controllers should be assessed by look at the following factors:

- stability of the system
- response time
- oscillatory behaviour
- steady state offset

6 PID control

PID control, seen in Figure 7, is made up of proportional, integral and derivative components. It is a very powerful tool when it comes to feedback design and improving system performance. In this section we explore how PID is implemented in electronic circuits without the need for sensors and software.

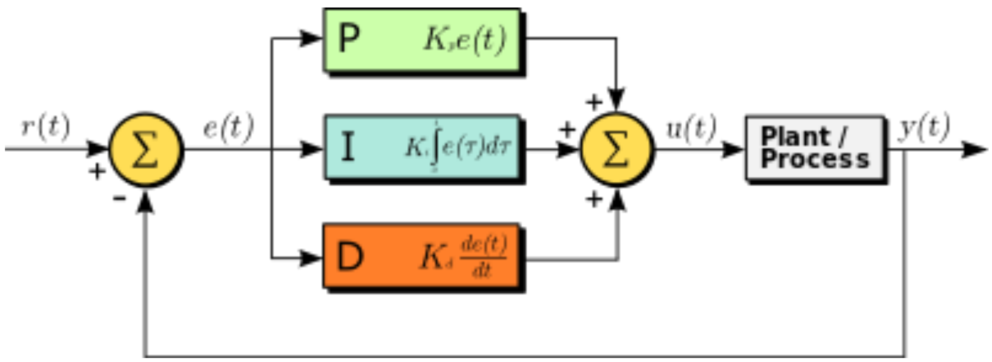

Figure 7: PID Controller [1]

6.1 Circuit Implementation

In this section we explore how PID control is implemented within a circuit.

- Find the relationship between Verror to Vp, Vd and Vi respectively for each circuit below, solving for Kp, Kd, Ki as seen in Fig 7, in terms of resistors and capacitors, in both the **time domain and laplace domain**.

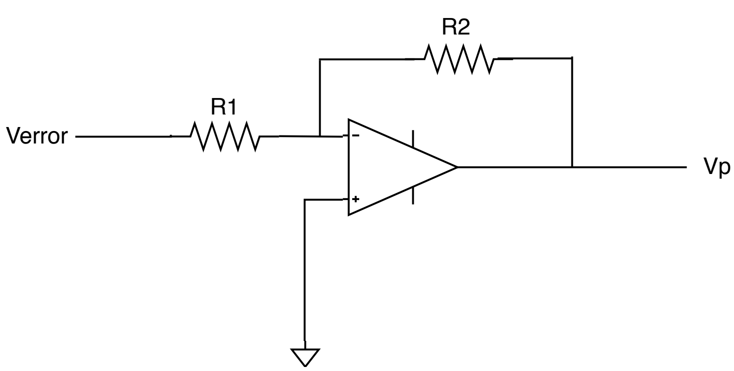

Figure 8: Proportional

$$K_p =$$

- For derivative control, find Kd assuming R3 >> R4.

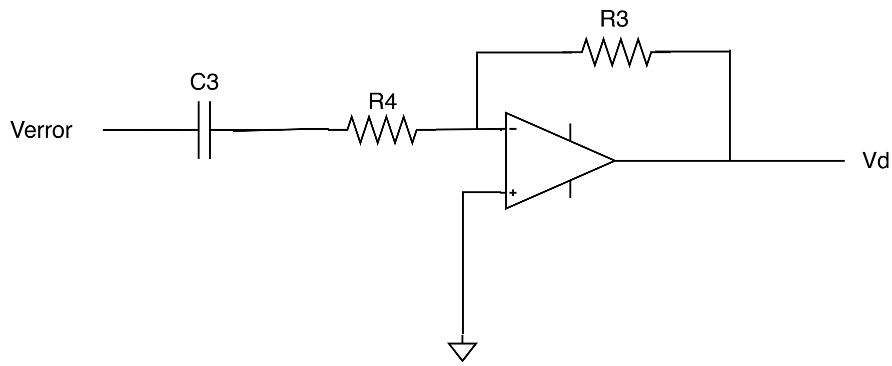

Figure 9: Derivative

$$K_d =$$

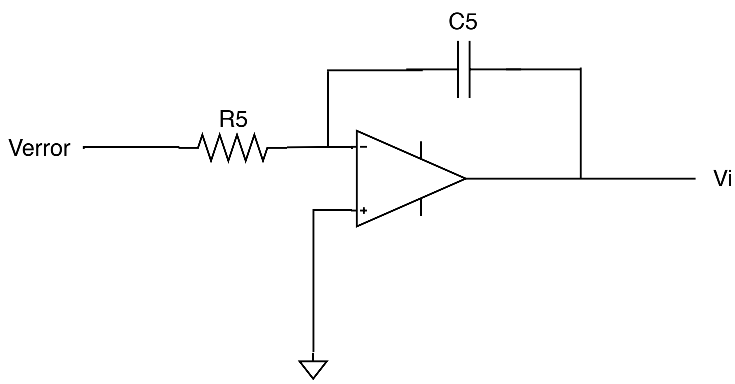

Figure 10: Integral

$$K_i =$$

- Establish the relationship between Vs and Vp, Vd and Vi, and therefore the relationship between Vs and Verror.

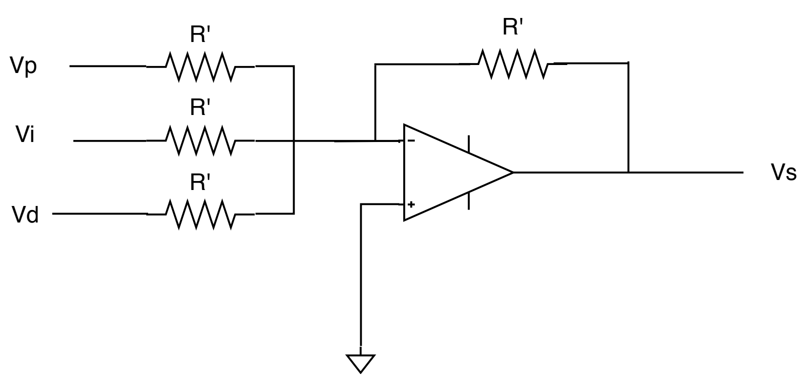

Figure 11: Summing Circuit

$$V_s =$$

Open LTSPICE file known as **PID.asc** to see the system in full (seen below in Figure 12) using the PID controller, and make sure you understand what circuit sections correspond to which feedback elements.

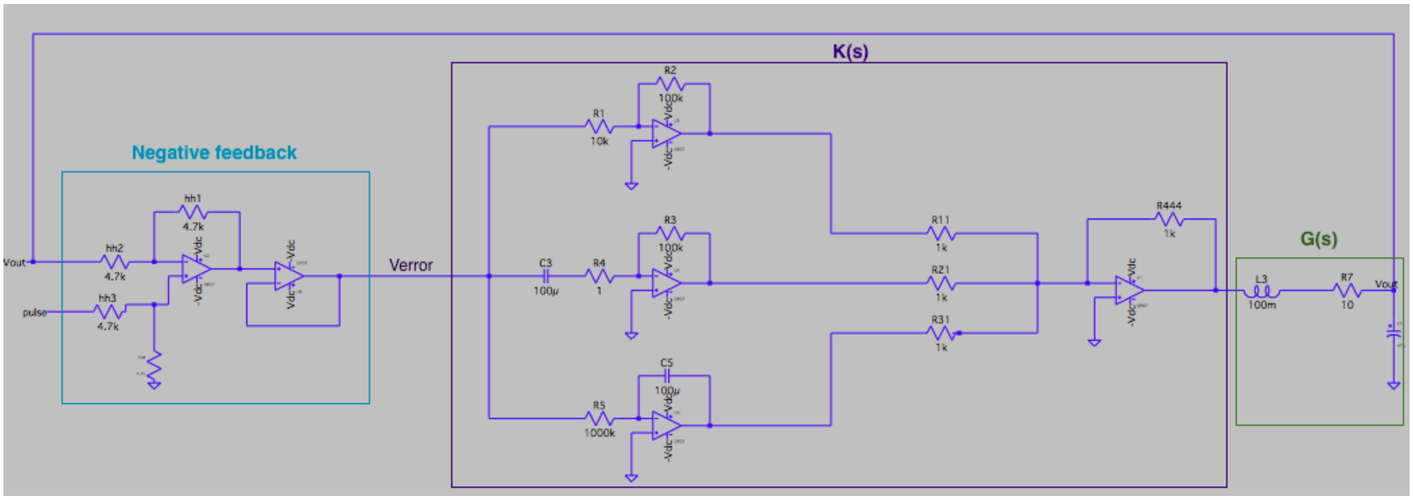

Figure 12: PID controller in LTspice

6.1.1 MatLab Simulation

Firstly, by using MatLab file **PIDControl.m** test out the effects of using PID by following these steps:

- Add proportional control - what happens? (e.g set Kp and let Ki=Kd=0)
- Add derivative control - what happens? (e.g set Kd, and let Kp = 4, Ki=0)
- Add integral control - what happens? (e.g set Ki, and let Kp = 4, Kd=0)
- Test different Kp, Kd, Ki values to achieve the best step response and system performance you can get. Obtain a graph which includes the **open loop and closed loop step step responses** as well as the associated **poles**.

6.1.2 LTSPice Simulation

- Once you are happy with the PID constants from the MatLab simulation, set up the LTSpice resistor, capacitor and inductor values to form Kp, Ki and Kd, chosen in MatLab. Compare the new step response to the open loop response (Vopen).

Note: You may have to choose different values of Kp, Ki and Kd, as you are limited to the real component values (e.g a capacitor of 1F is not a realistic value). Check typical L and C values online.

- Discuss the circuit behaviour observed in LTSPICE and compare the step responses to that of the MatLab model. What are your observations?

7 Lead Lag Compensation

Another way of designing a controller is by using lead-lag compensators. To improve performance of a system we must improve the gain and phase margins. In this section we design a controller, seen in Figure 6, through lead lag compensation design.

7.1 Lag

The circuit implementation of a lag compensator is seen in Figure 13.

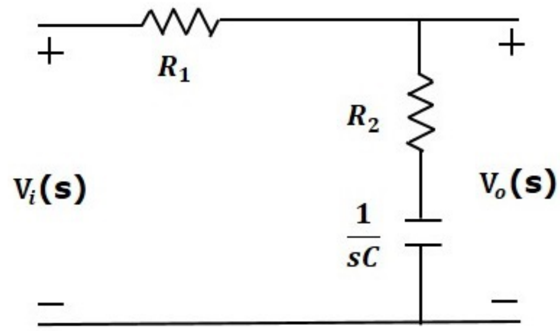

Figure 13: Lag compensator [2]

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{\alpha}\left(\frac{s+\frac{1}{\tau}}{s+\frac{1}{\alpha\tau}}\right)$$

Where $\tau = R_2C$ and $\alpha = \frac{R1+R2}{R2}$

7.1.1 MatLab simulation

- Model this is MatLab within the **leadlagcompensator.m** file. By choosing R1 = 1 kΩ, R2 = 1 kΩ and C2 = 1μ F and setting lead = 1, explain the effects a lag compensator using bode and nyquist diagrams, of G G*K and K the controller alone. (To help with analysis have all bode diagrams on the same graph).

- By choosing R1 = 2 kΩ, R2 = 1 kΩ and C2 = 1μ F, explain what happens to the closed loop system. Use bode and nyquist diagrams to explain this phenomenon.

7.1.2 LTSpice simulation

- Using **lag.asc** in LTSPICE test out the lag compensator above, describe and compare the step response to that seen in MatLab.

7.2 Lead compensator

The circuit implementation of a lead compensator is seen in Figure 14.

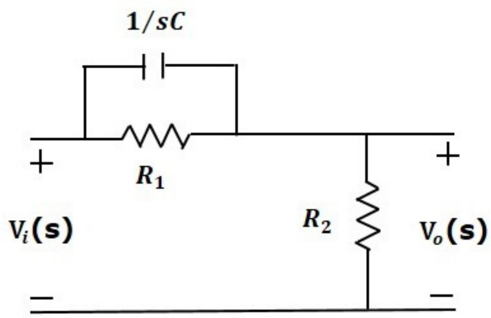

Figure 14: Lead compensator [2]

$$\frac{V_o(s)}{V_i(s)} = \beta \left(\frac{s\tau + 1}{\beta s\tau + 1} \right)$$

Where $\tau = R_1 C$ and $\beta = \frac{R2}{R1+R2}$

7.2.1 MatLab simulation

- Model this in MatLab within the leadlagcompensator.m file. By choosing R1 = 1 kΩ, R2 = 1 kΩ and C2 = 1μ F and setting lag = 1, explain the effects a lead compensator using bode and nyquist diagrams, of G G*K and K the controller alone.

7.3 Lead lag compensator

By combining them we get the following, lead-lag compensator.

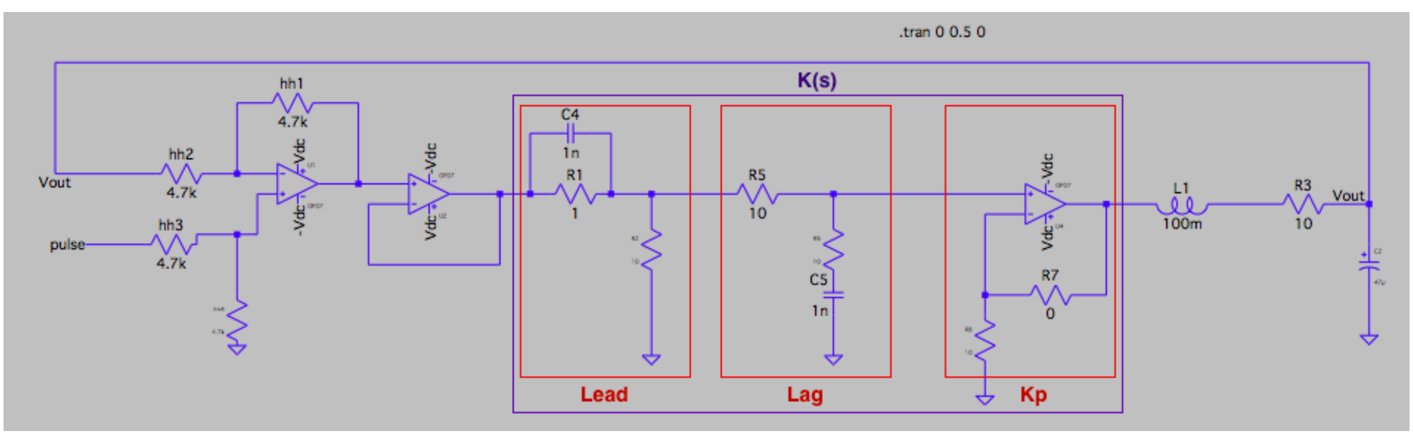

Figure 15: Lead lag compensator in LTSpice

$$K(s) = \beta \left(\frac{s\tau + 1}{\beta s\tau + 1} \right) \frac{1}{\alpha} \left(\frac{s + \frac{1}{\tau}}{s + \frac{1}{\alpha\tau}} \right) K_p$$

7.3.1 MatLab Simulation

Design a lead lag compensator in Matlab using the following step:

- Design a lead compensator in order to improve the transient response of the system (ignore the steady state error). In order to do this you must add a phase lead to improve the phase margin of the system. Find the phase margin of G(s) and apply a lead compensator at that frequency.
- Once you are happy with the transient response, add a lag compensator to increase the magnitude at low frequency and improve the steady state error.
- Finally add a proportional gain Kp to further reduce the steady state error.
- In your report compare the open loop step response to the closed loop state response. Discuss steps taken to design your lead-lag compensator, including nyquist and bode diagrams.
- Vary your system for the following three cases: (R = 100 L = 100e-3 and C = 47e-6), (R = 10 L = 10e-3 and C = 47e-6), (R = 10 L = 100e-3 and C = 47e-7), does your system produce a good step response in all cases? Can you conclude if your controller is robust?

For more information about lead-lag design refer to reference [3].

7.3.2 LTSpice Simulation

- Finally add in your values for your lead-lag compensator and compare the step response from the open loop system to the closed loop system.

8 Report

Write your report using the following format. You may include your results in an appendix if you wish. Make sure to include a good discussion about your results, the importance of feedback control and controller design, as well as a comparison between PID and lead-lag control.

- Abstract
- Introduction
- Theory
- Results
- Discussion
- Conclusion

If you have any questions, please contact TA [redacted]

9 References

[1] https://en.wikipedia.org/wiki/PID_controller
[2] https://www.tutorialspoint.com/control_systems/control_systems_compensators.htm
[3] http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_Leadlag