Feedback Control Systems (EL-3004)



Submitted by:

HAFIZ AHMAD

19L-1316

Submitted to:

Course Instructor: Dr. Omer Saleem Bhatti

19th May 2022

Department of Electrical Engineering

National University of Computer and Emerging Sciences, Lahore

# Table of Content

1. Introduction
2. Problem Analysis
3. Design Requirements
4. Feasibility Analysis
5. Possible Solutions
6. Preliminary Design
7. Design Description
8. Software Simulation
9. Experimental Results
10. Performance Analysis
11. Future Scope
12. Social and Cultural Implications
13. Conclusion
14. References

# INTRODUCTION

In this, I will describe the step-by-step procedure along with the explanation and evidence for each of the steps in the questionnaire that relate to the problem: Taking into account your roll number is 19L-2345, your forward way move capability would be

𝐺(𝑠) =

(𝑠 + 19)/(𝑠 + 1)(𝑠 + 3)(𝑠 + 1 + 6𝑗)(𝑠 + 1 − 6𝑗)

Sketch the root locus expecting it is a solidarity criticism control framework. Show every one of the computations connected with asymptotes, break focuses, jω intersections and point of flight/appearance.

Create a PID controller for your plant with a settling time of less than eight seconds and a maximum overshoot of less than ten percent. There should be no error in step response. Demonstrate your theoretically complete step-by-step design.

Create the PID compensated system's circuit.

Utilizing the transient specifications provided in part (b) and a 10% reduction in steady state error, construct a lead-lag compensator. Demonstrate your theoretically complete step-by-step design.

Plan the circuit of lead-slack repaid framework.

The steady state and transient response parameters of the two controllers should be compared and their advantages and disadvantages discussed. Additionally suggest which one ought to be taken on for

execution."

My Transfer Role:

𝐺(𝑠) =

(𝑠 + 19) / (𝑠 + 1)(𝑠 + 3)(𝑠 + 1 + 6𝑗)(𝑠 + 1 − 6𝑗)

# Problem Analysis

We must compensate our plant's transfer function in this issue. This move not set in stone by our Roll#. We need a certain steady state error and a certain transient response characteristic. There are two types of compensators at our disposal; we must design the transfer function using one of them and conduct a comparison between them. Additionally, we must design the compensator circuit. The requirements outlined in the issue are as follows:

My initial evaluation:

At first, it has two dominant poles, but the other poles are pretty close to them. The one is even slower than the poles that dominate. Additionally, the required pole of the transient response is slower than the dominant poles. However, the zero is far away. Thus this framework as of now can't be approximated as a second request framework.

The root locus of the first exchange capability will have 3 shafts going to boundlessness and one ending at a zero in the exchange capability.

# Design Requirements:

To plan a PID regulator and a lead slack regulator with the accompanying determinations:

• A settling time of fewer than eight seconds

• A percentage overshoot of fewer than ten percent

• Draw the root locus, assuming unity feedback, and demonstrate all calculations, such as:

• Asymptotes

• Break Points

• JW Crossing • Departure/Arrival Angle Additionally:

• Create the circuit for a lead-lag compensated system

• Create the circuit for a PID compensated system

• Compare the steady state and transient response parameters of the two controllers and recommend which one should be used for implementation.

# Feasibility Analysis:

# The parts utilized are really inconsequential and generally accessible.

# 

# Potential Arrangements:

# Controller PID:

# • We can have a PI compensator and a PD compensator where we can expect a zero and a post for every one of them.

# • We can put the zero expected for PI regulator, work out the point deviation after the option of new post lastly add a zero to combinedly remunerate it.

# Compensation for Lead Lag:

# • For the lead lag compensator, we can design both together or separately.

# •Preliminary Design:

|  |  |
| --- | --- |
| 1) | Root Locus Sketch |
| 2) | Fulfilling Pole Requirements Using Steady State Error and Transient Response Constraints |
| 3) | Calculating Pole Angles |
| 4) | Fulfilling the Angle Requirements by Adding a  Zero |
| 5) | Calculating the value of Gain K to form the Equation of the compensator |
| 6) | Design the Circuit of the Compensator From The  Equation |

|  |  |
| --- | --- |
| 7) | Repeat the Steps 2-6 for the Other Compensator |
| 8) | Presenting Their  Comparative Analysis |

Design Description:

# Design of a PID Controller:

# From this and by the condition given in the inquiry that OS<0.1 and Ts<8 we find that in the outrageous case z=0.6 and wn=0.8333333334

# So I planned the regulator for the outrageous case which will be material for more straightforward cases too. With the given transient response specification, the value of s is as follows: -0.5 + 0.6666667j Calculating angles from poles and zeros to this point to determine whether it is in the root locus or whether we require a compensator to provide an angle

# In order to achieve zero steady state error for step input, we must now adhere to yet another constraint, which basically adds a pole to the location (0,0). After this we work out the complete point from our posts to the given s esteem.

# So we expected to add a zero with positive point After estimations I tracked down that point. The PID controller's transfer function could be summed up like this. Now, the value of gain K was determined, which was as follows:

# 10.2

# PID repaid move capability was:

# G(s) = 0.023 (s + 19), 1.137 (s + 1.066), and s (s + 1) (s + 3) (s + 1 + 6j) (s + 1 - 6j)

# We begin by calculating the angle from the initial tf to the calculated point. Now, the calculated steady state error was 0.93 in the beginning. after 10%

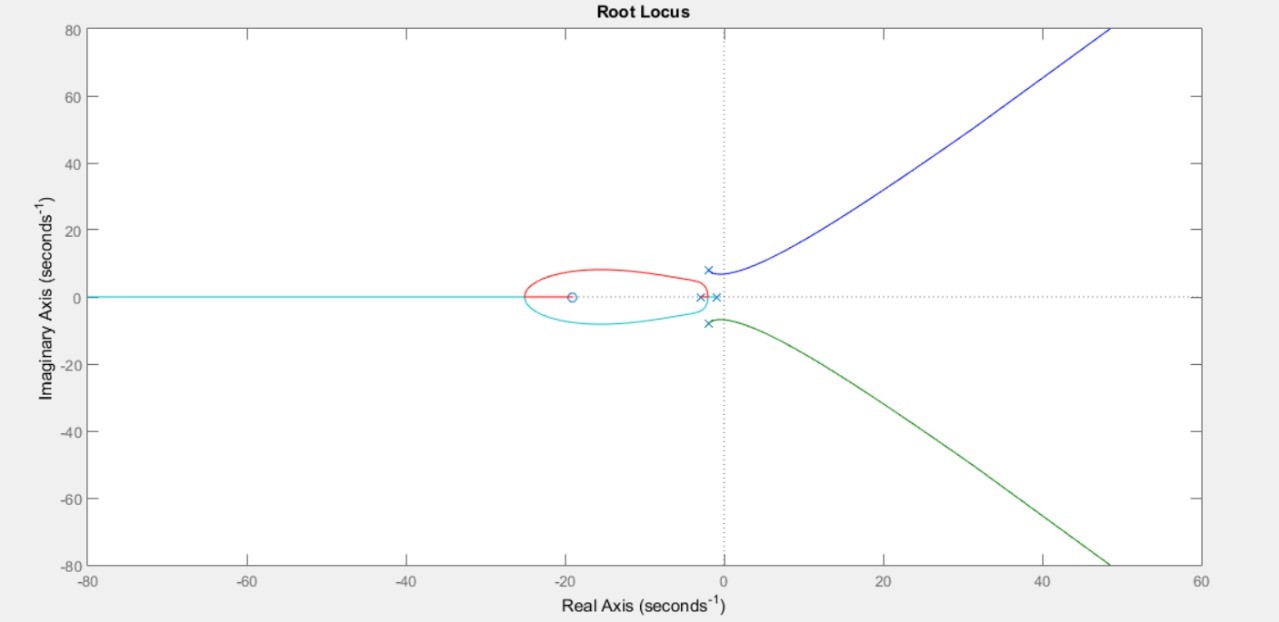
# decrease it became 0.837. Therefore, we needed a lag compensator for that.

# We discovered this by using the error to calculate the poles and zeros.

# Thus, the sum of the lead-lag compensated system's components would be:

# G(s) = 3.54(s + 19)(s + 19.163)(s + 0.00236) ((s + 20)(s + 0.001)(s + 1) (s + 3) (s + 1 + 6j) (s + 1 - 6j)) The lead lag compensator circuit was not possible because the ratio of leadcompensator zeros to lag compensator poles was not the same.

# Software Simulation:



Asymptote Angles: [60o, 180o, 300o]

Break Points: [-24.99, -2.01]

# Analyses of Performance:

# Benefits of Using a PID Controller:

# • PID controllers are relatively simple to implement • PID controllers can also be designed very precisely by tuning the proportional, integral, and derivative gains of the controller • PID controllers have zero steady state error • PID controllers are easier to stabilize • PID controllers respond faster to transients

# • PID controllers have a significant time delay; • PID controllers are not as good at dealing with disturbances; • PID controllers have a narrower range of stability; • PID controllers can amplify high frequency noise Advantages of a Lead-Lag Compensator:

# • Can accomplish framework prerequisites all the more precisely

# • The general speed of the framework increments

# • The upgraded damping upholds less overshoot alongside less ascent time and settling time

# • Further develops stage edge

# • Try not to upset the consistent state blunder of the framework

# • Boosts Kv of the framework Hindrances of a Lead-Slack Compensator:

# • Have a smaller scope of dependability

# • Can't have a consistent state blunder to be zero

# • Decreases overshoot yet at times increments undershoot, making the framework temperamental

# • Framework turns out to be more helpless to commotion

# Last Suggestion:

# A compensator is a part in a control framework that is utilized to manage the framework, changing a control framework to further develop its presentation could prompt unfortunate dependability or shakiness, so to cause the framework to act as planned, the better choice is add a compensator which would make up for the terrible showing of the first framework. Even though the PID controller is more adaptable and simpler to implement, if the gains are tuned even slightly, it may result in the system becoming unstable as a whole. As a result, a lead-lag controller is the better choice when the system needs to be extremely stable and no future adjustments or changes are intended. However, the PID controller is a better choice for you if you need something that is simpler to implement and more adaptable in the event that future modifications to the system specifications are required. Eventually, whether to utilize a PID regulator or a Lead-slack compensator is something settled on a circumstance to-circumstance premise.

# 

# Future Extension:

# We can expand this thought and we can make it a genuine item for the market on the grounds that as we have tried it is giving us exact readings consequently this is market-prepared, we simply need a few changes:

# Since our microprocessor, Tiva Launchpad, costs a lot, we could use Arduino or any other cheap processor to save money.

# To keep away from the utilization of a PC we can utilize a voice initiation sensor remotely this will liberate us from the bound of connecting of PC.

# If the use of our product requires close readings, we can use a temperature sensor.

# FANS can be utilized in place of LEDs.

# Cultural and Social Implications:

# They had no effect on society or culture.

# Conclusion:

# To finish up, first the root locus was portrayed. A PID controller with the specified parameters was developed using that root locus sketch, and a circuit was developed specifically for that PID controller. The lead-lag compensator was then designed and the circuit for it was also implemented in order to meet the same requirements while also achieving a 10% reduction in steady state error. This is not a definitive "yes" or "no" answer as to which controller should be used in the end; rather, it is an "it depends" answer. Whether to use a controller or a compensator depends on the circumstances and the features you value most, such as cost, adaptability, stability, accuracy, or robustness.

# References:

YouTube