ELEC 341

Project P9

20 Marks

Learning Objectives

- Lead-Lag (PID) Controller Design
- Zero selection
- Master Gain
- Heuristic Tuning
- Auto-Tuning

- Matlab
  - n/a
- Simulink
  - PID Block
  - Auto-Tune

LAG control did wonders for Steady-State error, but has a low Ku. Try LEAD-LAG control.

Stable Zero

Use **margin()** to find the Phase X-Over Frequency when a P-Controller is used.
Place the zeros one decade before the Phase X-Over Frequency.
Iterate using the Lead-Lag Controller dynamics until the zeros stabilize.
Set **K** to ½ the Ultimate Gain **Ku**

- P9_z (rad/s) = Location of double-zero
- P9_D = Unity gain controller dynamics with 2 zeros at –P8_z
- P9_K = Master Gain K

Nyquist Plot

Use nyqlog() to generate a Nyquist Plot of DGH.

- Nyquist Plot

Root Locus

Use rla() to generate a Root Locus of DGH.

- Nyquist Plot

Use the Nyquist Plot to verify your zeros provide a large phase margin and zoom in on the origin of the Root Locus to verify that your zeros are where you expected them to be.

You're gripper is used to pick up chicken eggs. The gripper is actuated with a step input, but you can't move the egg until the gripper settles. You will break the shell if there is any overshoot. Your task is to maximize throughput. It makes no difference smoothly the gripper grips. It just has to be fast and not break any eggs.

Requirement:	*Perform a Step Response*
Constraint:	*No Overshoot*
Goal:	*Minimum Settle Time*

Heuristic Tuning #1

Adjust K and z to improve the response as much as possible. Try complex zeros.
Plot a step response, Nyquist Plot, and Root Locus iteratively until you can't do any better.

- P9_K1 = Tuned Master Gain
- P9_z1 = Vector of zeros (actual zero locations with negative real components)

Heuristic Tuning #2

Use P9_z1 to compute the associated Kp and Kd.
Adjust K, Kp, Ki and Kd to improve the response as much as possible.
Adjusting K effectively scales the other 3 uniformly.
Plot a step response until you can't do any better.

- P9_K2 = Tuned Master Gain
- P9_Kp2 = Tuned Proportional Gain
- P9_Ki2 = Tuned Proportional Gain
- P9_Kd2 = Tuned Proportional Gain

Heuristic Zeros

Compute the zeros that result from your Heuristically tuned gains (#2).

- P9_z2 = Vector of zeros (actual zero locations with negative real components)

It would be interesting to see how this choice of zeros affects your Nyquist plot, and it would take very little effort to check. And once you've done that, a Root Locus is just as easy.

Tuned Response

Plot the following step responses using **THE SAME TIME VECTOR**.

Use θd = 10°.

- θa with Untuned Gains (degrees) black
- θa with Heuristic #1 Gains (degrees) blue
- θa with Heuristic #2 Gains (degrees) red

If your step response doesn't get better after each Heuristic Tuning step, you don't understand the meaning of "Heuristic Tuning".

The Simulink PID Controller block has an Auto-Tuning button. Try it.

Re-implement your control system in Simulink. If you scale each of your Kp, Ki, and Kd gains by K, and enter them into the PID block, along with your derivative filter pole, you should be able to re-produce your step response exactly.

Then create a copy of your system and push the Auto-Tune button.

Auto-Tuned Response

Plot the following step responses using **THE SAME SCOPE WINDOW**.

Use θd = 10°.

- θa with Heuristic #2 Gains (degrees) black
- θa with Auto-Tuned Gains (degrees) red

So how do they compare ???

Are you smarter than Matlab ???

Deliverables

Values

1. P9_z (1 marks)
2. P9_D (1 marks)
3. P9_K (1 marks)
4. P9_K1 (1 marks)
5. P9_z1 (1 marks)
6. P9_K2 (1 marks)
7. P9_Kp2 (1 marks)
8. P9_Ki2 (1 marks)
9. P9_Kd2 (1 marks)
10. P9_z2 (1 marks)

Figures

1. Nyquist Plot (1 marks)
2. Root Locus (1 marks)
3. Tuned Response (6 marks)
4. Simulink Response (2 marks)