ELEC 341

Project P6

10 Marks

Learning Objectives

- Micro-Controller Dynamics
- Reduced Models
- Proportional Control
  - Ultimate Gain
  - SS Error

You have identified all the sub-systems comprising a robot gripper.

- *In P2, you identified a voltage amplifier from experimental data.*
- *In P4 & P5, you identified a motor which is powered by your voltage amplifier, and a gripper which is actuated by your motor.*

Now add a controller to make the gripper do what you want.

Simulink Model

Add a controller to the Simulink Model you developed in P5. Make the original model a **sub-system** with the following interfaces.

- Input (V) = Motor Voltage
- Output #1 (deg) = Actual Motor Angle
- Output #2 (deg) = Finger Angle

Include the following in your control system.

- Input (deg) = Desired Motor Angle
- Proportional Controller with **Gain = Kp** (unity gain for now)
- Voltage Amplifier (from P2)
- Ideal motor angle sensor (no dynamics, unity gain)
- Micro-Controller Dynamics with **CF = 10 x #F**

The Simulink model should be a SIMO system with 2 outputs, motor and finger angle.

Upon receiving your components, you find that the friction specified in the data-sheet of the gear is underestimated. It is a no-load value since the gear manufacturer cannot predict what you are going to attach it to.

The friction is approximately 10x larger after attaching the gripper, due to the added tooth friction which is negligible at no-load.

Loop Gain

Re-calculate the Ym you calculated in P4, using the updated gear friction.
Compute the new **Path Gains** of your control system.
Simplify your system using **minreal()** and a tolerance of **1e-2**.

- P6_G = Forward Path Gain (deg/deg) = Desired Motor Angle / Angle Error
- P6_H = Feedback Path Gain (deg/deg) = Sensed Motor Angle / Desired Angle

Refer to your Simulink system when setting up your loop gain equation. It helps to see the system all laid out. Simulink is a very handy drafting tool that can be used for "system visualization" even before you have all your information. You can always update that later.

*Make sure you do your unit conversions in the right place. The input to you control system is specified in **degrees**, so your output must match.*

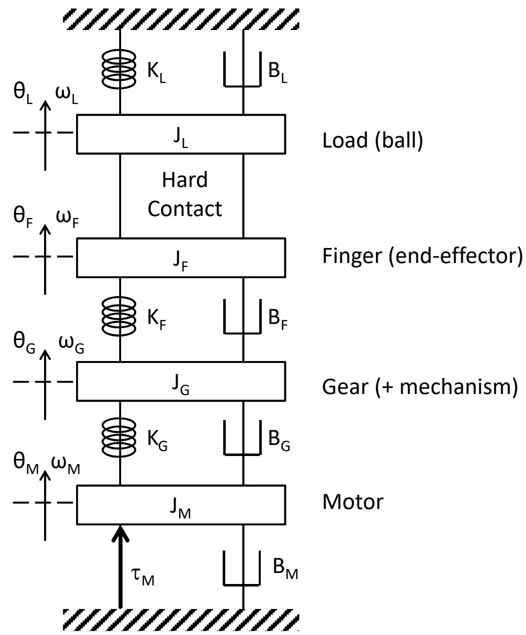
Mechanical Circuit of Gripper

- JG = #Ax75 mg-cm^2
- JF = #Bx25 mg-cm^2
- JL = 60 mg-cm^2

- BG = #Cx20 g-cm^2/s
- BF = #Dx400 mg-cm^2/s
- BL = 500 mg-cm^2/s

- KG = #Ex3000 Kg-cm^2/s^2
- KF = #Fx1800 Kg-cm^2/s^2
- KL = 2500 Kg-cm^2/s^2

- JM = From Maxon data-sheet
- BM = From Maxon data-sheet
- τM = 1 Nm (per-Unit)



Pole-Zero Plot

Use **pzmap()** to generate a Pole-Zero plot of the Loop Gain GH.
- Pole-Zero Plot

Notice that there are some poles & zeros that very nearly cancel out that have a very small real component and a very large imaginary component.

Bode Plot

Use **bode()** to generate a Bode plot of the Loop Gain GH.
- Bode Plot

When you have poles & zeros with large imaginary components, you get very sharp transitions in your Bode plot. This "noise" in your frequency response can create natural frequencies that make controller design very difficult. And they may not even represent the true behaviour of your system because you made a lot of estimates when you developed your system models.

In practice, it is wise to cancel these poles & zeros out, and design a controller based on what is left over. Once we are done, we can make adjustments to handle anomalies that show up in the ACTUAL SYSTEM.

Reduced Loop Gain

Use **minreal()** with a high enough tolerance to cancel out the "problematic" poles and zeros. Try a tolerance of 1 or 10.

Compute the reduced **Loop Gain** of your control system.

- P6_GH = Loop Gain (deg/deg) = Sensed Motor Angle / Desired Angle

This would be a great time to generate a Reduced Pole-Zero plot. How else will you know If you found the right tolerance ???

Reduced Bode Plot

Use **bode()** to generate a Bode plot of the Reduced Loop Gain GH.

- Bode Plot

Compare the Reduced Bode Plot to the original. Do you see what I meant by the phrase "noise" in the frequency response ???

Any one of those sharp spikes can cause instabilities to show up in your simulations, even though they probably won't show up in real life because of other losses (friction, heat, electrical resistance) that you neglected to consider.

Ultimate Gain

Use **margin()** to find Ku. Round it to 3 significant digits.

- P6_Ku = largest **STABLE** gain value

Don't carry around too many decimal positions. This is an in-exact science so 3 significant digits is plenty.

Step Response

Compute the **CLOSED-LOOP** Transfer Function of your **CONTROL SYSTEM**.

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

- θd = 10° (degrees)
- Kp = Ku/2 (degrees) blue solid
- Kp = Ku/5 (degrees) green solid
- Kp = Ku/10 (degrees) black solid
- Time Axis (msec)
- *Add a **legend()*** with descriptive labels.

When plotting a bunch of curves together, it helps to use the same time vector so everything is consistent. Choose an end time that allows the least stable curve to settle and use it for all curves.

If you put the same controller gain into your Simulink Model, you should see a similar response, even if you didn't reduce the equations.

Simulink Plot

Make Kp = Ku/5.
Display the following, all in the same scope window.
Set the simulation time so it matches your Step Response plot.

- Actual Motor Angle (deg)
- Finger Angle (deg)

Do you have reasonable control over finger angle which you aren't even measuring ???
Recall the magnitude of your desired angle. Is this an acceptable Steady-State error.
If not, can you increase the gain to make it reasonable ???
In other words, is this a useful gripper ???

Deliverables

Values

1. P6_G (1 marks)
2. P6_H (1 marks)
3. P6_GH (1 marks)
4. P6_Ku (1 marks)

Figures

1. Simulink Model (1 marks)
2. Pole-Zero Plot (1 marks)
3. Bode Plot (1 marks)
4. Reduced Bode Plot (1 marks)
5. Step Response (1 marks)
6. Simulink Plot (1 marks)