ME352 QUBE Motor Lab 2:
Motor Speed PI Control

Prof. Baglione, Prof. Luchtenburg
Teaching Assistants: Aidan Bowman, Jonathan Lerner, Kameron Wang
Technical staff: Michael Giglia

The Cooper Union
Fall 2022
(last updated: 9/19/22)

1 Overview

In this lab we will apply feedback control to the QUBE DC motor system with the goal of acceptable tracking and transient response. Proportional control (P) offers the advantage of fast transient response and reducing steady state error. While higher gains yield better performance, they come at the expense of increased sensitivity to noise. We will learn that integral control (I) offers the advantage of zero steady state error for a step input to this system. (This is not generally true for all systems). Unfortunately, however, as the gain of I is increased to improve rise time, overshoot will also increase. We will see that combining P and I (PI control) is better than either P or I alone by simultaneously offering the advantages of quick transient response, zero steady state error and decreased sensitivity to noise.

2 Goals

Our hands-on goals for today are to:

- Learn how to work as a designer. Use a *model* of the DC motor system to design a suitable controller which is implemented in an *experiment*.
- Design a feedback controller for reference tracking using P, I, and PI control approaches.
- Compare *measured* responses to those calculated for the *modeled* system.

3 Pre-lab Questions

Come prepared to your schedule lab session with the pre-lab questions answered. Turn in the pre-lab questions with the lab assignment.

Parameter identification

Recall from lab 1 that the first-order transfer function of the motor plant is given by:

$$P(s) = \frac{K}{\tau s + 1} \tag{1}$$

1. Write down the calculated values for K and τ you determined in lab 1, including units, using the QUBE-Servo manufacturer values of R = 8.4 Ω , L = 0.00116 H, J_m = 4.65x10^{-6} kg-m^2 (this value includes the hub attachment inertia), motor torque constant K_t = 0.042 N m/A, and motor back-emf constant K_e = 0.042 V/(rad/s). To calculate the load disc inertia, J_d, assume the disc has a mass of 0.053 kg and radius of 0.0248 m. Assume the motor damping/friction and the armature inductance are negligible.

Proportional (P), Integral (I) and Proportional-Integral (PI) control

A common block diagram for reference tracking is shown in Fig. 1. Recall that a proportional-integral controller has transfer function:

$$C(s) = k_p + \frac{k_i}{s} \tag{2}$$

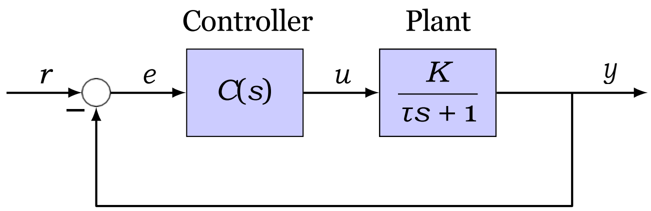

Figure 1: Block diagram for error-based feedback control (reference tracking).

In the following, we consider *proportional*, *integral* and *proportional-integral* control for the motor-tachometer[1] system. For all three types of control we'd like to design a controller such that:

(i) the rise time is less than 0.2 seconds[2]

(ii) the steady-state error is less than 1%

(iii) the overshoot is less than 20%[3]

(iv) the noise in the input signal (applied motor voltage) is small. The spread of the amplitude with respect to the mean signal should be less than 0.25 volts.

You, the control designer, need to find out if you can meet all prescribed specifications.

Proportional control

First, we consider *proportional* control, i.e. $C(s) = k_p$.

2. Use the final value theorem and reference-to-error transfer function (or determine the error from system type and error constants) to write the steady-state error to a unit step input in

[1] Actually, the QUBE motor has an encoder and a Simulink block is used to convert the position of the servo encoder to angular velocity using a low-pass derivative filter.

[2] F or this lab, use the approximation that rise time, $t_r \approx \frac{1.8}{\omega_n}$. Note this is only a rough approximation based on the behavior of second-order systems with no zeros. For first-order systems, the transient behavior can be better characterized by the time constant, which is related to the pole location.

[3] You may recall or will learn overshoot, $M_p(\xi) = e^{-\pi\xi/\sqrt{1-\xi^2}}$, which approximately yields ζ > 0.5 for M_p < 20%.

terms of the system parameters (K and/or τ) and the *proportional* control constant, k_p,

3. Design P-controllers such that the steady state error is respectively less than 30, 10 and 1%. Calculate the respective ranges of k_p values to meet each steady-state error specification.
4. Find the closed-loop pole and closed-loop time constant for the k_p value corresponding to 1% steady-state error.

Proportional-Integral control

In the *proportional-integral* controller case, $C(s) = k_p + \frac{k_i}{s}$, we can combine the favorable features of both P and I control, namely quick response, zero steady state error, and low sensitivity to noise.

5. Find the closed-loop transfer function for P-I control. Which property of the closed-loop system is influenced by k_p? Similarly, for k_i?
6. Design a PI-controller (determine k_p and k_i) for rise time $t_r \leq 0.2$ seconds, less than 1% steady state error, and overshoot less than 20%.

Note, if you did the calculations correctly, you should see the k_p gain required to meet the specifications using P-control only is much higher than in the PI-control case. In fact, the theoretical calculations should reveal that you should be able to meet specifications (i)-(iii) with I-control only. During the lab we will see if this is possible in practice.

4 Lab Assignment: Tracking the motor speed

Begin by opening MATLAB. Search in the QUBE_STUDENT folder for the "Lab 2 -Speed Control" folder. In the "matlab_files" folder double click on the lab2_speed_control.sltx template. Simulink will open, and a warning message will pop up. Click on OK and click on lab2_speed_control under Template to create model on the screen. Save the Simulink file locally. Run the lab2_params.m file in MATLAB to load the parameters and click the Monitor & Tune button in the Hardware Tab.

Model Verification

7. Indicate the best K and τ for the motor you are running for this lab. Compare your theoretical model with the actual DC motor angular velocity by comparing the model step response with the actual QUBE motor step response.

Proportional control implementation

Now let's implement *proportional* control with the k_p values calculated in the pre-lab assignment.

8. Implement the k_p values from (3). Set k_i to 0 by clicking on the integral gain block. Set k_p by clicking on the proportional gain block. Discuss how thee response changes as you increase k_p and whether you are able to meet the specifications for (i)-(iii).
   Note, you can view the response by clicking on the Scope blocks. Simulink limits the number of data points stored in the scope by default. To increase the data points, click "View", then "Configuration Properties", and on the "Logging" tab, change the "Limit data points to last" to 20,000. You can save the scopes as a figure by clicking "File" and "Copy to Clipboard".
9. Verify in an experiment whether you can meet all design specifications (i)-(iv)! The noise in the input signal (applied motor voltage) should be small. (The spread of the amplitude with respect to the mean signal should be less than 0.25 volts). Do the controllers operate as you expect? Why or why not?

Integral control implementation

Next, we consider *integral* control only, i.e., $C(s) = \frac{k_i}{s}$

10. Implement an integral controller (set k_p =0) using the rise time tr ≤ 0.2 s, less than 1% steady state error, and overshoot less than 20% specifications. The noise in the input signal (applied motor voltage) is desired to be small. (The spread of the amplitude with respect to the mean signal should be less than 0.25 volts). Can you meet all design specifications (i)-(iv)? Why or why not? Do the controllers operate as you expect based on your theoretical model?

Proportional-Integral control implementation

Now, let's implement the *proportional-integral* controller

11. Keeping integral control, start with a very low k_p gain and gradually increase k_p to see the change in response with a large k_p gain. Observe how increasing k_p affects the response. Vary both gains to see if you can meet the specifications from (i)-(iv). The noise in the input signal (applied motor voltage) should be small. (The spread of the amplitude with respect to the mean signal should be less than 0.25 volts – Look at the motor voltage scope!). How can k_p and k_i be traded off with each other to achieve the desired system behavior? List your final k_p and k_i and whether you were able to meet all the specifications.

5 Effect of Pole Location on Response

12. Use the following Matlab code to plot the root locus for the system with P-control[4]. Sketch or turn in a Matlab plot of the root locus plot. The × should show the location of the open loop pole which should be $s = -\frac{1}{\tau}$.

```
s = tf('s');
sys = K /(tau*s + 1);
rlocus(sys)
```

- *You can also use the Control System Designer Matlab application and GUI using the command* `controlSystemDesign(sys)`. *This will allow you to adjust the loop gain by dragging the closed-loop pole location (shown as a pink square).*
- *You can preview the adjusted value of the loop gain, in this case k_p, by clicking* C *in the Controller and Fixed Block Window and view the corresponding modeled Step Response. You can also see the Bode Plot which will be covered later.*
- *Try dragging the closed-loop pole to the value that yields the k_p gain you calculated in (4). You can double-click* C *and enter the k_p value that corresponds to 1% error in the Compensator Editor window if you get tired of dragging! The value of the closed-loop pole should agree with your calculations in (4).*

---

[4] You should learn the root-locus design method, in which you use the open-loop transfer function, in this case $C(s)P(s)$ since the system has unity feedback, and indicate the open-loop poles (and zeros, if any). Either with hand calculations or using Matlab, root loci branches can be plotted, which show the locations of all possible closed-loop poles as you vary one parameter in the system, usually a controller gain, between 0 and ∞.

There are good Root Locus Controller Design and other Matlab & Simulink Control Design tutorials here: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlRootLocus (or Google CTMS tutorials).

13. Add Matlab code or modify the system to plot the root locus for the system with I-control. (Remember to use the open-loop transfer function!) Sketch or turn in a Matlab plot of the root locus for the system with I-control.

14. What happens to the root loci branches (and corresponding closed-loop poles) as you increase k_i and how does this affect the closed-loop time response characteristics?

- *Try dragging the closed-loop pole to the value that yields the ki gain corresponding to the rise time tr ≤ 0.2 s, less than 1% steady state error, and overshoot less than 20% specifications.*
- *Note, you can right click on the step response, select "Design Requirements" and click "New" to add time domain specifications to visualize on the root locus and help in selecting possible gain values that, in theory, meet the specified requirements.*

6 Lab Discussion

15. Why do we need a model of the motor-tachometer system (plant)?

16. What are the advantages of a P-controller? What are its disadvantages? Similarly, for an I-controller?

17. What are the advantages of a PI-controller? Justify your answer based on your experimental results!

18. Why are the choices for the gains limited in practice?

7 Extra Credit

19. Extra Credit: Build a Simulink block diagram to track a sinusoidal signal with a frequency of 1 rad/s. How does your controller perform?