Modification of System Dynamics by Pole Placement
MAE 433 Spring 2012
Lab 3

Prof. Rowley, Prof. Littman
AIs: Brandt Belson, Jonathan Tu
Technical staff: Jonathan Prévost
Princeton University

Feb. 28–Mar. 2, 2012

1 Overview

In this lab we will *modify the dynamics* of the motor-tachometer system using feedback-control. Last week, we saw that the open-loop dynamics of the motor were the same as those of a a low-pass filter. This week, we will add feedback control such that the closed-loop transfer function is the same as that of a harmonic oscillator (for example, a spring-mass-damper). By varying the pole locations of the closed-loop system, we can tune the response of the system (frequency, damping, etc.). We will first test our controller designs in simulation using a model of the dynamics. Once the design criteria are met, we will implement the controllers experimentally and verify that the physical motor-tachometer system has the desired response.

2 Goals

Our hands-on goals for today are to:

- Learn how pole locations influence the dynamics (growth rate and/or frequency) of a system
- Understand how Proportional (P) and Proportional-Integral (PI) control can be used to modify pole locations
- Use a model to design controllers that will produce desired behaviors in experiment

There is a brief lab report that you will hand in before leaving the lab.

3 Modeling the motor-tachometer system

For today's lab we will need a model of the motor-tachometer system. For a DC motor, the input voltage and angular velocity are related by

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

We will identify the parameters K and τ system using a step response. Recall that step response for the above transfer function is given by

$$\omega_m(t) = K\left(1 - e^{-\frac{t}{\tau}}\right). \tag{2}$$

(We analyzed these dynamics in HW 2.)

1. Build a Simulink model for the motor-tachometer system. **DO NOT use a `Coulomb & Viscous Friction` block in your model.**
2. Run a step response, save the output, and plot it in Matlab.
   Remember to use an input voltage large enough to avoid complications from the dead band voltage.
3. Determine the value of K by looking at the steady-state response.
   Hint: Remember to account for your step height. For today's lab, just identify the steady-state value by eye, rather than taking a mean.
4. Determine τ by looking at the exponential growth/response speed.
   Hint: Consider Eq. (2). What is the analytic value of $\omega_m(\tau)$? Can you identify when this value is achieved in your experimental step response? How is this related to the value of τ? Again, for this lab identify τ by eye. Don't do any linear fitting.
5. For your values of K and τ, plot the predicted step response against the experimental one. Make sure that they match. Save this plot for your lab report. **Before proceeding, check in with an AI to discuss your results.**

4 Transfer function analysis

We will be modifying the dynamics of the motor-tachometer system using feedback. In this lab, we will use a controller that responds to the measurement y (rather than an error, as we would do for reference tracking). The block diagram for this setup is shown below, where P denotes the plant (motor-tachometer system) and C the controller.

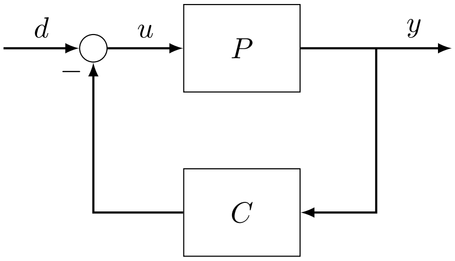

Figure 1: Block diagram for feedback control based on the measurement y.

The transfer function of the plant is given by Eq. (1), and for the controllers we will use one of the following:

- Proportional (P) control, with transfer function

$$C(s) = k_p \text{ , or} \tag{3}$$

- Proportional-Integral (PI) control, with transfer function

$$C(s) = k_p + \frac{k_i}{s}. \tag{4}$$

For all the questions below, consider both P and PI control. Be sure to record all calculations from this part of the lab as part of your lab report.

1. What is the transfer function from d to y?

2. What is the steady-state value of y (zero-frequency/DC gain) of the closed-loop system?

3. If the input d is a *disturbance*, what is the optimal DC gain? How can we choose our controller gains to achieve this?

4. **Check in with an AI to discuss your analysis.**

5 Modifying the motor-tachometer dynamics

Proportional control

1. Determine what range of gains k_p, if any, will drive the system unstable.
Hint: for what pole locations is a system unstable?

2. Determine what range of gains k_p, if any, will make the system oscillatory.
Hint: for what pole locations is a system oscillatory?

3. Design a proportional controller that will yield a fast response. Design a controller that will respond slowly. How much faster or slower do you expect these systems to be? **Do not consider de-stabilizing controllers.**
Hint: What pole locations yield a fast response? What about a slow one? To determine the exact speed of the system, compute the time constants of the controlled systems. Consider putting the closed-loop system in the same form as Eq. (1).

4. What are the steady-state values of y for the closed-loop system for the fast and slow controllers you designed above? Do these controllers achieve the goal of disturbance rejection?

5. Is there a limit on how fast a response you can achieve? What about on how well you can reject disturbances?

6. Design a controller that is at least 10 times as fast as the fast one you designed above. Is there any practical reason you would not implement so fast a controller in real life?

7. **Check in with an AI to discuss your analysis.**

8. Use Matlab to compare the step responses of the *model* with and without feedback. Confirm that the behavior you predicted is achieved. **Do not drive the physical motor-tachometer system yet!**

   (a) Perform step responses for the following systems: open-loop, closed-loop with slow controller, closed-loop with fast controller, closed-loop with 10x faster controller.
   To store the data for plotting, use the syntax `[y,t] = step(sys)`.

   (b) Plot all four stable step responses in the same figure.

   (c) Plot the unstable step response in a separate figure.
   Only consider a short time window, or else the step itself will be hard to see in the plot.

   (d) Save these plots for your lab report.

9. Make a Simulink model for controlling the actual physical motor-tachometer system.
   You can simply put a feedback loop in the model you used earlier.

10. Run the experiment for the following systems: open-loop, closed-loop with slow control, closed-loop with fast control, closed-loop with 10x fast control. **If the motor begins to whine or buzz, turn it off immediately!** For each case:

    (a) Save the input and output signals for plotting in Matlab.

    (b) Plot the experimental and simulated step responses in the same figure.

    (c) In this figure, identify (mark) the predicted steady-state value and time constant. Make sure that the experimental behavior agrees with your analysis from above.

11. **Check in with an AI to discuss your results.**

Proportional-Integral control

1. Using PI control, can we drive the system unstable? Can we make it oscillatory? Consider all four combinations of the above two criteria.
   Hint: You DO NOT have to compute any pole locations. Look at the transfer functions. How many poles does they have? How many free parameters do you get to choose? What does this mean about where you can put the poles?

2. We saw in the last lab that the motor acts like a low-pass filter. Design a controller to change its dynamics so that it behaves like a spring-mass-damper with a frequency of 1 rad/s. To do so, the following steps are suggested:

   (a) Write an analytic expression for the poles of the closed-loop transfer function with PI control. Are the poles real or complex? Which control gains affect the decay rate? Which affect the oscillatory behavior?

   (b) Fix a value of k_p. Solve for k_i to achieve the desired behavior.
   Since there are two free parameters, we need to fix one before we can solve for the other.

3. **Check in with an AI to discuss your analysis.**

4. *Simulate* the step response of the motor with PI control. Does it behave as expected? What happens if you make k_p large? What if k_p is small, or even negative? Adjust your design so that the step response shows clear oscillations at a frequency of 1 rad/s.

5. Implement your PI controller in experiment. Record the behavior for plotting in Matlab.

6. Plot the simulated and experimental step responses in the same figure. Save this plot for your lab report.

7. **Check in with an AI to discuss your results.**

8. Determine the resonant frequency of the system. Force the system at this frequency, as well as higher and lower frequencies, to demonstrate that resonance does occur.
   Hint: Bode plots.

9. **Check in with an AI to discuss your results.**

10. Design a controller such that the closed-loop system has two real, stable poles. Each pole has a corresponding decay rate. What will the decay rate of the closed-loop system be?
    Hint: The step response of this system will be a sum of exponentials, just like the impulse response (as we saw in lecture). The decay rate of each exponential will correspond to the value of a pole. Which one will dominate the response for long time horizons?

11. Verify this behavior in experiment.

12. **Check in with an AI to discuss your results.**

6 Deliverables

Your lab report should include the following:

- A plot comparing the simulated and experimental step responses (from the modeling portion of the lab). Include your values of K and τ.
- Computations from the transfer function analysis.
- Plots of the simulated step responses with P control (all four cases in one plot).
- Plots comparing the simulated and experimental step responses with P control (4 cases, each with its own plot).
- A plot comparing the simulated and experimental step responses for PI control.
- Brief (one-two sentences) answers to all of the questions posed above.