Lab 4: Root Locus Based Control Design

References:

Franklin, Powell and Emami-Naeini. Feedback Control of Dynamic Systems, 3rd ed. Addison-Wesley, Massachusetts: 1994.
Ogata, Katsuhiko. Modern Control Engineering, 5th ed. Prentice Hall, New Jersey: 2009.

Introduction

The objective in control system design is to manipulate a system's response via a compensator (i.e. controller). The response of a compensated system is presumably more useful than the uncompensated version.

Lab 3, *Basic Control Actions*, demonstrated the use of the PI compensator. For the hydraulic motor, a proportional gain decreased the time constant of the response although it left a non-zero steady state error. An integral gain term eliminated this steady state error. Using a PID compensator with the hydraulic piston, the derivative gain slowed down the system response, thus reducing overshoot in the system.

A desired system response is obtained by carefully selecting the PID compensator's gain parameters. Gain selection, aside from deciding the form of the compensator, is a major part of basic control system design.

There are many methods by which one may choose compensator gains. In Lab 3, students were asked to try different proportional and integral gains that were chosen semi-randomly by the instructor. The point in that exercise was to observe the effects of changing the various gain parameters.

If the point is to obtain a "good" response from a system whose mathematical representation is unknown, the Ziegler-Nichols tuning rules (Franklin p. 191, Ogata 5th ed. p. 568) are used to select the PID controller gains. When a reasonably accurate mathematical representation of the system is known, as in the case of the hydraulic motor in the Parker-Hannifin Motion Control Laboratory, there are more sophisticated and powerful methods by which to select the gains.

The root locus method is one such method that will be explored in this laboratory exercise. Frequency response methods and others will be investigated in future exercises.

In this laboratory exercise, a proportional and PI compensator will be designed for the hydraulic motor using the root locus method and MATLAB. The mathematical model for the motor system was developed in Lab 2, *Time Domain System ID – First Order System*. The designed control system will be simulated using Simulink and then implemented on the Parker Trainer Stand. A comparison of the simulation to the actual response will yield differences that must be explained in the laboratory report.

Experimental Objectives

Completion of the laboratory exercise will have required you to:

- Simulate a physical system using the model obtained from a Plant Identification.
- Design a proportional and PI compensator using root locus design methods.
- Implement the compensators on the physical hydraulic motor.
- Explain discrepancies between the recorded speed output from the physical system and that of the simulation.
- Understand qualitatively the information contained within a system's root locus plot.

Overview of Design Process

Design of a compensator using the root locus design method typically involves the following steps:

- Obtain a plant model.
- Translate design requirements into ζ and/or ω_n information. System characteristics such as time constant (for first order systems) and overshoot and rise time (for second order systems) are design requirements.
- Translate ζ and/or ω_n information into closed loop pole locations.
- Draw the root locus plot of the compensated system for a range of gains K_c.
- Select a gain from the root locus plot which corresponds to the desired pole locations and use it in the compensator.
- Iterate if the system characteristics are not met due to unmodeled dynamics.

In the Prelab assignment, you completed the first three steps. In this lab, you will complete the remaining steps in the compensator design process and then test your compensators in simulation and on the experimental system.

Root Locus Design Using MATLAB

A compensator design using the root locus method can be done completely in MATLAB with the tool "rltool".

Once the compensator gains have been designed, one should always **test the closed loop system using Simulink before hardware implementation**. This is recommended in case errors have been made in the formulation of the controller.

Experimental Procedure

Experiment 1: Root Locus Design with a Proportional Compensator Only

The open loop hydraulic motor system that utilized the BD90 valve, as identified in Lab 2 at 300 psi, should have a time constant of about 50 ms and a steady state gain of about 0.60. In this first exercise, we desire the time constant to be 25 ms. Thus, we close the loop and implement a proportional only controller because there is no requirement on the steady state error or overshoot. **In the Prelab assignment you determined the *closed-loop* pole needed such that the *closed-loop* time constant of**

the motor's response to a step input is 25 ms. Here you will use the root locus method to determine the appropriate proportional controller gain for the closed loop system.

Save all data as you work in C:\ME460_SP*xx*\AB*x*. This includes any M-files, should you choose to use them.

- First, draw a block diagram of the closed loop system with a proportional controller K_c on paper. Use symbolic names for the values of the other gains as well. Refer to Figure 1 in the prelab.

Using block diagram algebra or any other acceptable method, find the closed loop transfer function of the system.

- Specify the plant transfer function in MATLAB using the function *tf(num, den)*. The model for the hydraulic motor (standard first order system) takes the form:

$$G(s) = \frac{K}{\tau s + 1}.$$

- Run *rltool(G)* to start the Control System Designer with your transfer function G.
- You can modify the closed-loop architecture under the *Edit Architecture* tab. The red block represents the compensator of which the root locus analysis is being applied to.
- Double-click on the compensator editor (*Data Browser* > *Controllers and Fixed Blocks* > *'C'*) to add poles/zeros, and adjust gain of the controller.
- You can also adjust controller in the *Root Locus Editor for LoopTransfer_C* plot. This figure should automatically open when you start rltool.
- In this 1st exercise, the controller is a proportional controller. So there is no need to add pole/zero to the compensator.
- The root locus plot which relates pole location to specific K_c values for the given controller and plant. The blue 'x's represent the open-loop poles while the solid pink squares represent the closed-loop poles at the given K_C value. You can click and drag the pink square(s) to change the K_C value. The blue line represents all possible closed-loop pole locations when K_C takes values from 0 to ∞. A grid of damping and natural frequency is available by right clicking on the plot and select *Grid*.
- By adjusting controller gain, you can determine a gain (in the root locus plot) which satisfies the design requirement (performance of the r2y step response plot).
- Find the controller gain required to achieve a closed-loop time constant of 25ms.
- Hint: Right click on the *r2y Step Response* plot and select Characteristics > Rise Time. Then right click again on the plot and select Properties > Options and change the rise time to be from 10% - 90% to 0% - 63.2%. This will effectively give you the time constant for first-order systems.
- Easier hint: Where should you move the closed-loop pole (pink square) such that the closed-loop time constant is 25ms (this was solved in the prelab).

- Using Simulink, create a model of this controlled system and simulate the compensated step response of the hydraulic motor to a step input of 800 RPM. Set the fixed time step in the Simulation/Parameters menu at 1 ms and then plot the output and reference.
- If you need MATLAB and Simulink help, refer to Lab 1.

Next, we will implement this controller on the actual system.

- Modify the template N:\HydraulicsLab\ME460\rloc_template.mdl so that it is the proportional controller just designed. Save it in C:\ME460_SP*xx*\AB*x*. The procedure to run your real-time Simulink model will be the same as Lab 3.
- Make the physical connections such as the wiring between the MW2000 and the tachometer output, and the hose connections for the BD90 valve and the hydraulic motor. Refer to Lab 3 for this setup.
- **Have an instructor check the block diagram, wiring and hosing.**
- Start the system then start the controller. Obtain a good response for a step input of 800 RPM. Save the data in C:\ME460_SP*xx*\AB*x*.
- Stop the hydraulics.
- Plot the data and the simulated response on the same plot. Be sure to account for any time delay. Label, title and save the figure.

Experiment 2: Root Locus Design with a PI Controller

Unfortunately, the above proportional compensator was not good enough for 'the boss', who has decided that the design requirements will now be that the overshoot, M_p, be exactly 1.5% and the peak time, t_p, be approximately 0.21 seconds. In the Prelab assignment you related these design requirements to values for ζ and ω_n.

Again, we choose to use the root locus method to design the PI compensator. However, now there are two gains, K_i and K_c, which need to be chosen. Unfortunately, basic root locus methods allow us to investigate the movement of the closed loop poles with respect to only ONE gain as it is varied from zero to infinity. To circumvent this problem, we specify the ratio K_i/K_c. This changes the problem so that there is only one independent gain that needs to be varied. In *rltool* you can modify both gain and pole and zero locations.

- Draw a block diagram on paper of the closed loop system with a PI compensator. Keep everything symbolic for now.
- Using block diagram algebra, or any other acceptable method, find the closed loop transfer function of the system. The integrator has added another pole to the closed loop transfer function when compared with the proportional only system. **What else has the integrator added to the closed loop equations?**
- Manipulate the denominator until it has the form of Equation (1) in the Prelab assignment. K_c is the gain from the proportional controller, and the ratio K_i/K_c should appear somewhere.
- For the first part of this lab experiment, assume the ratio K_i/K_c to be 32.
- Create the root locus plot for the modeled system using *rltool* by following the same procedure in the first exercise.
- Activate the grid in the root locus to plot the lines of constant ζ and ω_n determined in the first step. Note that their intersection is the location of the complex closed loop poles that you calculated in the Prelab assignment. The root locus should pass through the intersection or pretty close to it. Note that only the closed loop poles which lie on the root locus are possible for the system. Quite often, the locus will not pass through the intersection of the desired ζ and ω_n, which means that the design requirements cannot be met simultaneously. In this case, the problem is 'professorial' enough that the locus should pass reasonably close to the desired complex poles.
- Title, label and save the root locus plot.

- Simulate the response of the closed loop system to a step input of 800 RPM using Simulink. Save and plot the results.
- Implement the control system on the physical system. **Remember to have an instructor check the hydraulics and the block diagram before starting the system.** Record the response to a step input of 800 RPM. Save and plot the results as before.

Lab Report

1. Include all plots created and saved during lab in your lab report.

2. Include your answers to all of the Prelab assignment questions, as well as the question under the second bullet in Experiment 2.

3. While there is too much uncertainty and variation in the system for the actual and simulated responses to correspond exactly, they should have the same qualitative 'shape'. In some cases the actual step response may exhibit different 'shape' features when compared to the simulated response. Why? Use the plots from both the proportional only and PI compensated systems in formulating an answer.

4. Theoretically, the system's response (i.e. ω_n or τ) can be made to be arbitrarily fast simply by increasing the proportional gain and perhaps the ratio K_i/K_c. In practice, this is not true. Why not?

***Last Updated: Apr 2018, R2017a*