TAKE HOME LABS
OKLAHOMA STATE UNIVERSITY

Root Locus Design

by Martin Hagan
revised by Trevor Eckert

1 OBJECTIVE

The objective of this experiment is to design a feedback control system for a motor positioning system. Based on the motor model you developed in the *Open Loop Step Response* experiment, you will use the root locus diagram to determine the best closed loop pole locations when using both proportional and derivative feedback. After you have simulated the response of your feedback control systems, you will test the controller experimentally. You will then iterate your design to find the best possible response, in terms of settling time, percent overshoot and steady state error.

2 SETUP

2.1 REQUIRED MATERIALS

2.1.1 HARDWARE

- All hardware from the *Open Loop Step Response* experiment is required for this lab. (No additional hardware is required)

2.1.2 SOFTWARE

- All software from the *Closed Loop Step Response* experiment is required for this lab. (No additional software is required)

2.1.3 PREVIOUS EXPERIMENTS

- *Closed Loop Step Response*

2.2 Hardware Setup

No hardware setup is required. You should have completed the hardware setup in the *Closed Loop Step Response* experiment.

2.3 Software Setup

No software setup is required. You should have completed the software setup in the *Closed Loop Step Response* experiment.

3 Experimental Procedures

3.1 Exercise 1: Control Design (Proportional Feedback)

In this exercise you will design a proportional feedback controller for the DC motor, using the root locus diagram. The controller signal $u(t)$ (motor voltage) will be *proportional* to the difference between the reference signal $r(t)$ and the motor position $\theta(t)$ ($y(t)$).

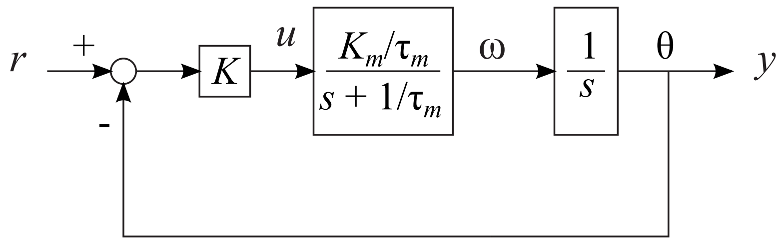


Figure 3.1: Block Diagram for Closed Loop Motor with Proportional Feedback

1. Using block diagram manipulation on the block diagram in Figure 3.1, find the transfer functions $G(s)$ and $H(s)$ for the equivalent block diagram in Figure 3.2. Plug in the values for K_m and τ_m that you found in the *Open Loop Step Response* experiment.

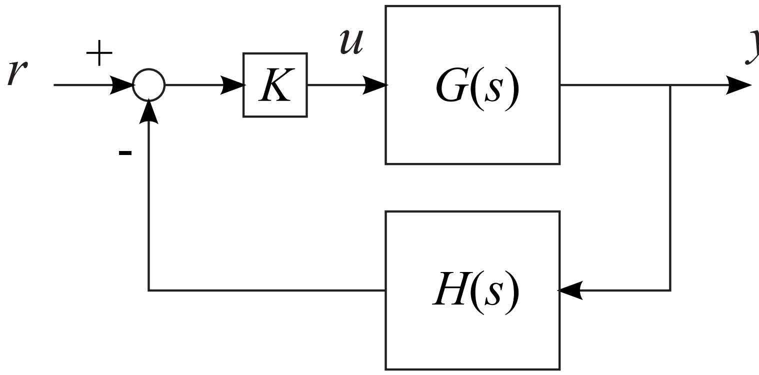


Figure 3.2: Standard Feedback Control Block Diagram

2. Find the closed loop transfer function $Y(s)/R(s)$. Find the closed loop poles as a function of K. Complete Table 3.1, computing the closed loop poles for each indicated value for K. In the table, P.O. is the percent overshoot of the step response, t_p is the time of the first peak in the step response, and t_s is the settling time (5%) of the step response. Hand plot each pair of closed loop poles in the complex plane (on the same plot). Indicate the number that corresponds to each gain next to the poles.

Table 3.1: First Set of Gains

Number	K	Closed Loop Poles	P.O.	t_p	t_s
1	0.002				
2	0.01				
3	0.1				
4	10				

3. Plot the root locus diagram for this proportional feedback system as K is varied from 0 to ∞ using the standard root locus rules. Describe how the system step response would change as the gain K is increased from a very small value to a very large value. Be as specific as you can. Make sample sketches of the step response for a very small gain and for a large gain.

4. You want to select K so that the system step response has the smallest settling time, while also maintaining less than a 5% overshoot. Where would be the best closed loop pole locations? Explain your answer carefully.

3.1.1 Checking Results with MATLAB

5. Open the **CL_Constants.m** file from the *Closed Loop Step Response* experiment.

6. Save the file as **RL_Constants.m.**

7. Press the **Run** button  at the top of the page. Navigate to the **MATLAB** command window. Under "Workspace" on the right-hand side of the page, all of the variables from **RL_Constants.m** should be listed.

8. In the command window, type *g=tf([Km/tau],[1 1/tau 0])*. This defines the motor transfer function.

9. Now you will use a MATLAB tool to simplify the design process. (See `http://www.mathworks.com/help/control/getstart/siso-design-tool.html` for a detailed description of this tool.) Type *controlSystemDesigner* in the MATLAB command window. (Depending on the version of MATLAB that you have, you may need to use the command *sisotool* instead.) You should see the windows shown in Figure 3.3 (depending on the version of MATLAB that you are using).

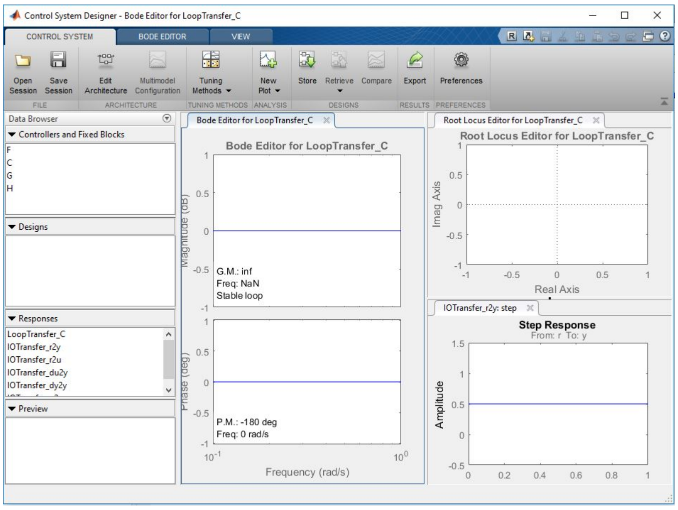


Figure 3.3: Control System Designer Window

10. Click the **x** on the *Bode Editor for LoopTransfer_C* window to get rid of the plot.
11. Drag the rest of the plots to the left to make them bigger.
12. Click the **Edit Architecture** button to add transfer functions to a block diagram. Once the window pops up it should give you a figure that looks like Figure 3.4. It shows the standard feedback control block diagram. For this experiment, there will be no pre-filter, so the F block will be left as 1 or $< 1x1zpk >$. The G block is the motor transfer function, and the H block represents the measurements, which will also be 1 for the proportional feedback system we are considering in this exercise. The C block represents the compensator, which will be the gain K for our proportional feedback system.

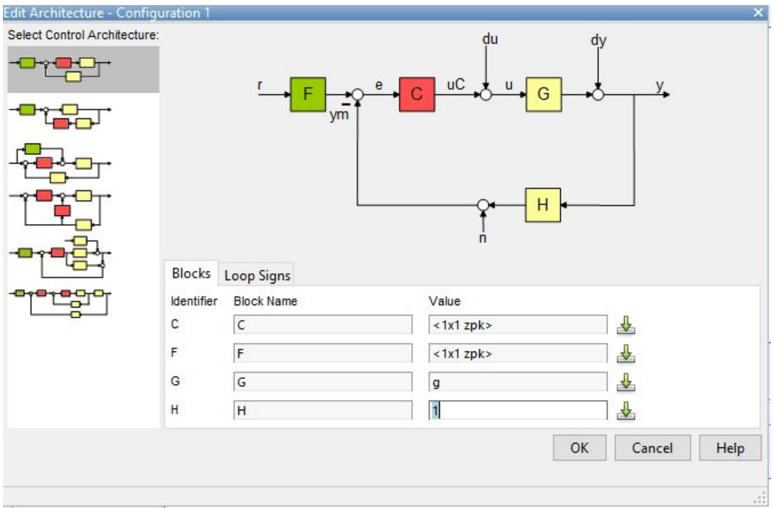


Figure 3.4: Control and Estimation Tools Manager

13. The next step is to enter the motor transfer function into the G block of the *Edit Architecture Tool Manager*. Double-click in the *Value* column of the G row, and enter *g*, as shown in Figure 3.4. Also, click in the *Value* column of the H row, and enter *1*. Then click *OK*. The root locus diagram should now be visible in one of the windows.

14. The step response that is shown will be for the default gain value of $K = 1$, since we did not change the default compensator value in the System Data window. The pole locations for this gain will be shown as small squares on the root locus plot, as shown in Figure 3.5. (Your root locus plot may look different than this figure, since you have a different motor transfer function.) You can grab the small square and move the closed loop poles. This will cause the gain K to change. (If you click on the C in the *Controllers and Fixed Blocks* subwindow at the upper left of the *Control System Designer*, the gain value will be displayed in the lower left *Preview* subwindow.) At the same time, the step response will change in the step response window. Save the root locus diagram for your lab notebook, and save the step response plot for a few different gain values. Discuss how these plots relate to the root locus and step response plots you made in Step 3.

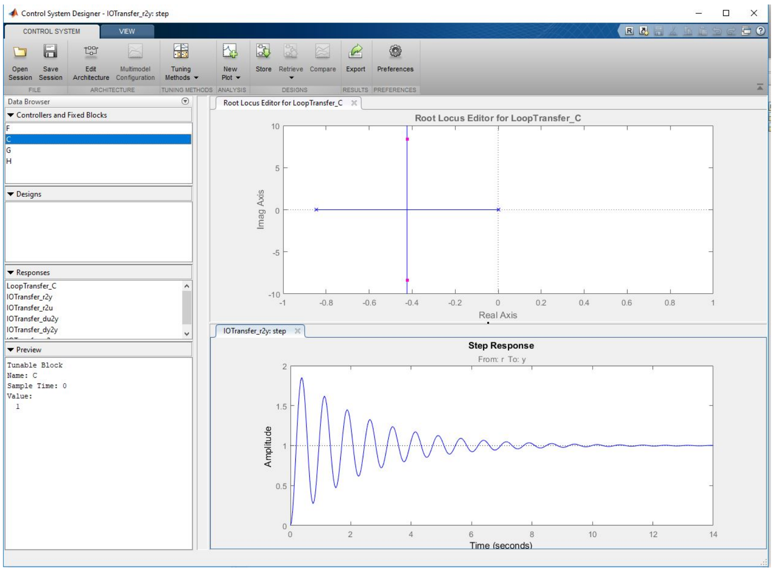


Figure 3.5: Root Locus

15. By moving the closed loop poles, and monitoring the step response, select the value of K that you believe will produce the best response in terms of smallest settling time, with minimal oscillation. Justify your choice. Save the best step response plot for your lab notebook.

16. For the K that you selected, determine the voltage that it would produce, if the error $(r - y)$ is $\pi/2$. (Remember that $u(t) = K(r(t) - y(t))$.) Is this enough voltage to move the motor? Think back to the *Simple DC Motor*, *Open Loop Step Response* and *Closed Loop Step Response* experiments. Keep this in mind, when you analyze the experimental results later in this experiment.

3.2 Exercise 2: Simulated Step Response (Proportional Feedback)

You will now simulate the closed loop step response before finding the step response experimentally.

3.2.1 Setting Up Simulink File (Simulation)

17. Open the **CL_Simulation.slx** file created in the *Closed Loop Step Response* experiment. It should appear as in Figure 3.6. Set the simulation time to 10 seconds.
18. Save the file as **RL_Simulation.slx.**

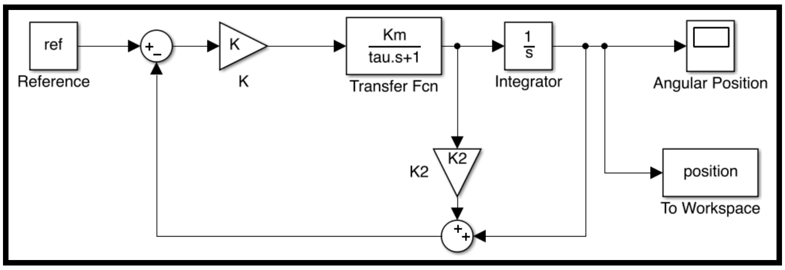
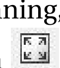


Figure 3.6: Final Simulink Simulation Model

19. Open **RL_Constants.m** and set the value of K to the value you found in Step 15. Be sure the value of $K2$ is set to zero. Then press the **Run** button  at the top of the page. Navigate to the **MATLAB** command window. Under "Workspace" on the right-hand side of the page, all of the variables from **RL_Constants.m** should be listed.
20. Open **RL_Simulation.slx.** Click the Run button  at the top of the page.
21. Once the model has finished running, double-click on the **Angular Position** scope block. Click the **Autoscale** button . Observe the plot. Does the closed loop step response appear to rise up from zero and settle to the reference value (as in your plot from Step 15)? If the plot looks to be correct (with a run time of 10 seconds) continue to the next step. Otherwise, go back to the previous section to ensure your Simulink file is correct.
22. Navigate back to the **MATLAB** command window. Under "Workspace" a variable (**position**) should now be available. Right click on **position** and click "Save As..." Navigate to the folder in which you have saved this project, type next to "File name:" **RL_position_1.mat**, and click "Save" at the bottom of the page.
23. You now have the simulation data found from Simulink for the best proportional feedback controller.

3.2.2 Simulation Plot File

24. Open the main Matlab 2017a window and click **New** at the top and then click **Script**.

25. Once the new Untitled m-file appears, Click **Save** at the top of the page. Save the file as **RL_Plot.m**.

26. Copy and paste the text in Listing 1 into the Matlab file. After adding the code click **Save** and then click **Run** .

27. Save the figure as **RL_S_1.fig** into your folder for this project. Refer to this figure for the remaining steps in this section.

28. Compare the simulation results with the plot you found in Step 15. They should be almost identical. If not, then you will need to check the gain value you found in Step 15.

Listing 1: `Code for Plotting the Closed Loop Step Response Simulated Results`

```
%Load the Simulation data and time and store into variables
RL_simResp_1 = load('RL_position_1.mat');
t = RL_simResp_1.position.Time;
RL_simResp_1 = RL_simResp_1.position.Data;

%Plot the simulation data with respect to time
figure;
plot(t,ones(size(t))*ref, 'Color', 'r');
hold on;
plot(t,RL_simResp_1, 'Color', 'k','LineWidth', 2);
title('Simulated Closed Loop Step Response');
legend('Reference','Simulated', 'Location', 'southeast');
xlabel('Time (seconds)')
ylabel('Theta (radians)')
```

3.3 Exercise 3: Experimental Step Response (Proportional Feedback)

This section will provide the setup of the Simulink file for the Arduino.

3.3.1 Setting Up Simulink File (Arduino)

29. Open the Simulink file created in the *Closed Loop Step Response* experiment named **CL_Step_Resp_Arduino.slx**. It should look like Figure 3.7

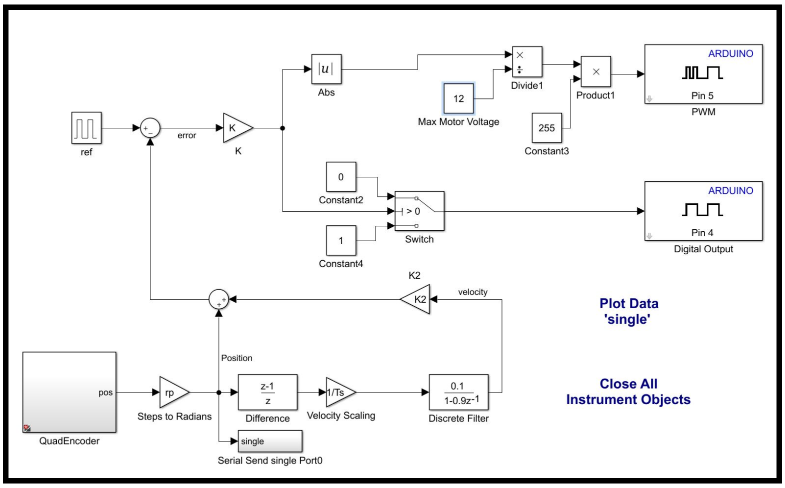


Figure 3.7: Closed Loop Simulink Model for Closed Loop Step Response Experiment

30. Delete the **Discrete Filter** block and connect the line from the previous **Velocity Scaling** to the **K2** block.

31. Click **File → Save As... → RL_Step_Resp_Arduino.xls**. The Arduino Simulink file for the experimental closed loop step response (proportional feedback) is now complete. See Figure 3.8 for the completed model.

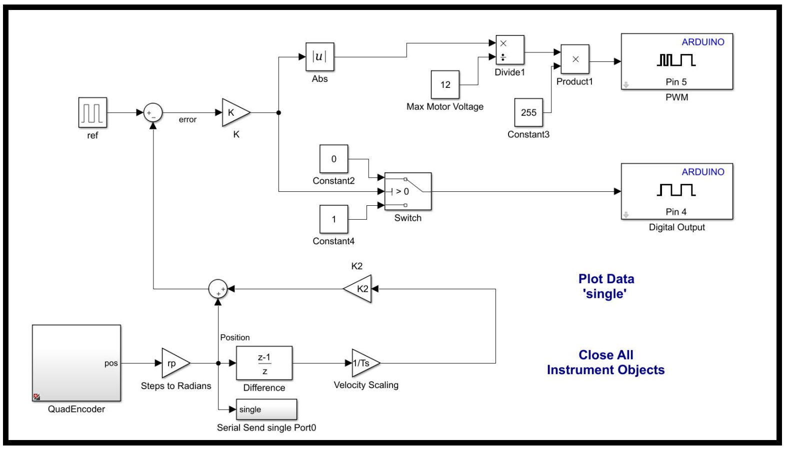


Figure 3.8: Final Closed Loop Step Response Simulink Model for Arduino

3.3.2 Collecting Experimental Data

32. Open **RL_Constants.m** and click the Run button  at the top of the page.

33. Open **RL_Step_Resp_Arduino.slx** and click the “Deploy to Hardware" button  at the top-right of the page.

34. Once the model has successfully deployed to the Arduino, double click on the text **Plot Data 'single'** inside the model window.

35. When the small window labeled "Plot Ser..." appears, enter the Arduino COM port number under "Enter COM port to collect data:." The default values for "Enter Number of Samples to plot:" is "single" and for "Enter Number of samples to plot:" is **8000**.

    Note: To find the COM port number for your Arduino, refer to the *Simple DC Motor* experiment under the section "Software Setup → Installing Arduino Mega 2560 Drivers."

36. Click Okay. Once the plot appears, plug the power cord from the power supply into the motor shield.

    CAUTION: Do not put your hands or any other parts of your body in front of the motor load trajectory.

37. The motor should attempt to turn 90 degrees and stop. (The motor may not actually turn a full 90 degrees, depending on the gain value that you used for the controller.) Then it should return to its original position and stop. This cycle should repeat itself every 20 seconds. If the plot does not reflect the movement that you see in the motor load, follow the steps you used in the *Closed Loop Step Response* experiment to obtain a reasonable plot.

38. Let the data fill the plot window as it moves to the left. Once the data has filled the screen completely *and the first pulse has moved to the left off the screen*, click the "Stop" button at the bottom of the screen. You should now have 8000 data points on the screen. **Note:** you should at least let the first pulse disappear as it usually will not be the full ten seconds.

39. Navigate back to the **MATLAB 2017a** main page. Under "Workspace" the variable **WindowDat** should now be present. Right-click on it and click "Save As." Name the file **RL_expResp_1.mat** and save it into the folder where the **RL_Step_Resp_Arduino.slx** file is saved.

40. You now have the experimental data for the closed loop step response with proportional feedback.

3.3.3 Experimental Plotting File

41. Open the **RL_Plot.m** file you created in the **Simulation Plot File** section.

42. Add the text in Listing 2 to the bottom of the **RL_Plot.m** file. After adding the code, click **Save** and then click **Run** .

Listing 2: `Code for Plotting the Closed Loop Step Response Experimental Results`

```
%Load the experimental data and store into a variable
RL_expResp_1 = load('RL_expResp_1.mat');
RL_expResp_1 = RL_expResp_1.WindowDat;

%Align the experimental data with the reference
%and compute the root mean square error.
[yplot,minrmse,~,~] = findShift2(RL_expResp_1,T*(1/Ts),D*T,ref);
T1 = Ts*(0:(length(yplot)-1));

%Plot the experimental data
hold on;
plot(T1,yplot,'b','LineWidth', 2)
ax = axis;
text(ax(2),ax(4)-0.1,['Experimental RMSE = ' num2str(minrmse)],...
'HorizontalAlignment','right','VerticalAlignment','top');
legend('Reference','Simulated','Experimental','Location', 'southeast'
    );
```

43. Save the figure as **RL_SE_1.fig** into your folder for this project. Refer to this figure for the remaining steps in this section.

44. Compare the simulation results with the experimental results you found. Estimate the settling time, percent overshoot, and frequency of oscillation of the closed loop step response from the experimental plot. Compare with the simulated plot. Also compare the steady state values from each plot, discussing similarities and explaining differences. What could cause the experimental response to differ from the simulated response? Are there nonlinear effects in the motor that could change the performance?

45. Can you adjust the gain K to improve the experimental response? The root mean square error (RMSE) between the reference and the motor angle is shown on the experimental plot. How small can you make this value by changing the gain K. Go back to the earlier steps when you selected the K value. Try different closed loop pole locations. Perhaps you need to accept a larger percent overshoot in order to achieve a smaller RMSE. Explain your design process.

3.4 Exercise 4: Control Design (Proportional plus Derivative Feedback)

In this exercise you will design a proportional plus derivative (PD) feedback controller for the DC motor, using the root locus diagram. The controller signal $u(t)$ (motor voltage) will be K times $r-(k_2\omega+k_1\theta)$. Since k_1 will be set equal to 1, K effectively multiplies $(r-\theta)-k_2\dot{\theta}=e-k_2\dot{\theta}$. The term Ke is called proportional feedback, since it produces an input that is proportional to the error. The term $Kk_2\dot{\theta}$ is the derivative feedback, and has a damping effect, like viscous friction. The block diagram of the PD controller is shown in Figure 3.9.

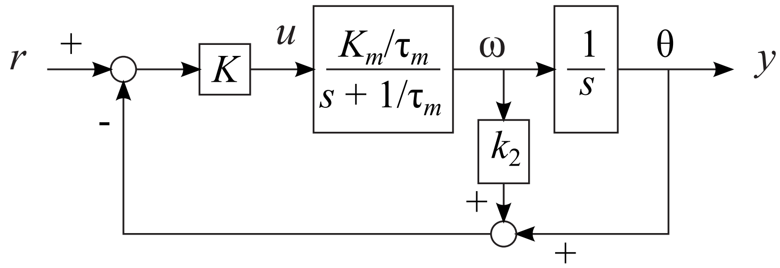


Figure 3.9: Block Diagram for Closed Loop Motor with Proportional plus Derivative Feedback

46. Using block diagram manipulation on the block diagram in Figure 3.9, find the transfer functions $G(s)$ and $H(s)$ for the equivalent block diagram in Figure 3.2. Plug in the values for K_m and τ_m that you found in the *Open Loop Step Response* experiment. Your H transfer function should be in the form $k_2(s+b)$

47. Let $k_2 = 0.2$, find the closed loop transfer function, and find the closed loop poles as a function of K. Complete Table 3.2 and hand plot the closed loop poles for each gain (on the same plot) denoting the number that corresponds to each gain next to the poles.

Table 3.2: Second Set of Gains

Number	K	Closed Loop Poles	P.O.	t_p	t_s
1	0.002				
2	0.01				
3	1				
4	2				

48. Let $k_2 = 0.2$, and plot the root locus diagram for this proportional plus derivative feedback system as K is varied from 0 to ∞. Describe how the system step response would change as the gain K is increased from a very small value to a very large value. Be as specific as you can. Make sample sketches of the step response for a very small gain and for a large gain.

49. You want to select K so that the system step response has the smallest settling time, while also maintaining less than a 5% overshoot. Where would be the best closed loop pole locations? Explain your answer carefully.

50. If you change the value of k_2, how is the root locus affected? Use sketches of the root locus for various values of k_2 to illustrate the effect. By adjusting both K and k_2, how much flexibility do you have in placing the closed loop poles? Are there theoretical

limits on the closed loop pole locations? Are there practical limits on the closed loop pole locations? Discuss these ideas in detail.

3.4.1 Checking Results with Matlab, Simulation and Experimental Results for PD Control

51. Repeat Steps 7 to 45 for the proportional plus derivative feedback system. You will need to create the H transfer function (like you created the G transfer function in Step 8) and load it into the *Control and Estimation Tools Manager* (like you did for the G transfer function in Step 13). You will also need to modify the values for K and $K2$ in the **RL_Constants.m** file. (When you are saving figures and data files, you will want to adjust the file names, and use these new file names in the **RL_Plot.m** file.)

52. After completing the simulations and experimental results for the PD controller, experiment with different values for K and $K2$. Can you reduce the steady state error, while maintaining a low overshoot and minimum settling time? Find the controller that produces the minimum RMSE. How much lower can you make the RMSE using the PD controller, when compared to the proportional controller? Explain your final tuning process and justify your final design. Discuss theoretical aspects of pole locations and their relation to overshoot and settling time, and explain practical considerations that must be taken into account to reduce steady state error when nonlinear effects must be taken into account.

53. What is the resolution of the encoder in radians, if there are 64 counts per revolution? How big is the steady state error for your experimental results? Can you make a connection between the encoder resolution, which is used to measure the motor angle, and your steady state error?

4 Table of Discussions and Questions

Before you turn in your report for this experiment, make sure that you have answered all of the questions that have been posed. It is important that your answers be expansive and that they demonstrate that you were mentally engaged in the experiment. Below is a recap of the important questions and the number of the step where each question was embedded.

steps	Discussion/Question
1	$G(s)$ and $H(s)$ transfer functions
2	Table 3.1 and hand plot of closed loop poles
3	Plot root locus
3	System response as K is increased
3	Sketches of step response for a very small gain and large gain
4	Best pole locations and selection of K
14	Step response plots and root locus for different gains
14	Comparison with Step 3.
15	Selection of K and step response plot
16	Voltage if error is $\pi/2$. Is it enough voltage?
21	Does your response appear to rise up from zero and settle to the reference value?
44	Comparison between simulation and experimental results
44	Settling time, percent overshoot, frequency of oscillation and compare with simulation
44	Similarities and differences
44	What could cause the experimental response to differ from simulation?
44	Are there any nonlinear effects in the motor?
45	Explanation of design process for making RMSE smaller
46	$G(s)$ and $H(s)$ transfer functions
47	$k_2 = 0.2$ Table 3.2 and hand plot of closed loop poles
48	Plot root locus
48	System response as K is increased
48	Sketches of step response for a very small gain and large gain
49	Best pole locations and selection of K
50	Changing k_2.
50	Flexibility of closed loop poles
50	Theoretical and practical limits
51	Make sure you answer all the of the questions (should be similar to previous in the table)
52	Discussion on process of finding gains and final gain parameter design choice and calculations.
53	Encoder resolution
53	Connections between steady state error and encoder resolution

5 CONCLUSION/STUDENT FEEDBACK

This experiment lead you through the design process for proportional and proportional plus derivative feedback controllers. The PD controller enabled more control over the placement of closed loop poles, and allowed an improved system response.