TAKE HOME LABS
OKLAHOMA STATE UNIVERSITY

Closed Loop Frequency Response

by Carion Pelton

1 OBJECTIVE

In this experiment you will investigate the frequency response of a position control system for a DC motor. Using the linear DC motor transfer function that you developed in the *Open Loop Frequency Response* experiment, you will find the closed loop transfer function for the position control system. From the closed loop pole locations, you will sketch the theoretical frequency response of the closed loop system. Following this, you will find the closed loop frequency response experimentally, by inputting a sinusoidal reference position, for multiple frequencies, into the motor control system. You will also use Matlab to calculate and plot the theoretical frequency response. By comparing the theoretical and experimental frequency responses, you will develop insights into the capabilities and limitations of linear models of dynamic systems for predicting real-world performance. You will also study the effects of position and velocity feedback on closed loop performance.

2 SETUP

2.1 REQUIRED MATERIALS

2.1.1 HARDWARE

The hardware materials required for this experiment are the same materials used in the *Open Loop Frequency Response* experiment. A list of the materials needed for this experiment is provided below for reference.

- DC Motor with Encoder

- Arduino Mega 2560 Rev3
- Motor Shield (DFRobot L298P)
- 12V DC Power Supply (12VDC 3A Wall Adapter Power Supply)
- USB cable (Standard A to B plug)
- Female Barrel Jack (2.1mm x 5.5mm Female CCTV Power Jack Adapter)
- 8 wires
- 3D Printed Base
- 3D Printed Motor Clamp
- 2 #4-40 1/2" screws
- 2 #4-40 1/4" screws
- Small Flat screwdriver
- Small Phillips screwdriver
- Sticky Tack
- Sandpaper or Sanding Sponge (Optional)
- Pennies - Same number used in the *Open Loop Frequency Response* experiment
- Scotch tape
- Small 3-D Printed Load

2.1.2 Software

- MATLAB/Simulink 2013a or later
  - ⋆ *The steps and images related to MATLAB/Simulink for this experiment were created using MATLAB/Simulink 2013a. Therefore some steps and images may be a little different if you are not using this version. If you are in fact using a different version, make sure you know the steps for running models onto the Arduino for your version of Simulink.*
- MATLAB files

2.1.3 Prerequisite Experiments

- Open Loop Frequency Response

2.2 SOFTWARE SETUP

The necessary software files that are needed for this experiment can be downloaded from the Take Home Labs webpage. One method for downloading the files is shown in the steps below.

1. Open your internet browser and navigate to `csl.okstate.edu`
2. On the left side of the Homepage, select “Courses"
3. In the middle section of the Courses page select “System Dynamics"
4. In the middle section of the System Dynamics page find and select *Closed Loop Frequency Response*.
5. On the Closed Loop Frequency Response page select “Software/Code" in the rightmost section. A zipfile named *Experiment_CLFR* should download.
6. Right-click the file and choose “Extract All...", or any other method of extracting files on your PC.
7. Extract this folder somewhere convenient, and remember the location. This will be the folder where all of the files and plots created for this experiment are saved.

2.3 HARDWARE SETUP

Since there was no additional hardware needed for this experiment, there is no additional setup that needs to be done. The hardware should still be set up from the *Open Loop Frequency Response* experiment.

3 EXPERIMENTAL PROCEDURES

In this section you will derive the theoretical closed loop DC motor model using the open loop transfer function that was found in the *Open Loop Frequency Response* experiment. You will sketch the bode plot of the the theoretical closed loop DC motor model, for various feedback gain values. Using the same gains, you will experimentally find the closed loop frequency response of the DC motor and load. Using MATLAB, you will then calculate the bode plots of the theoretical closed loop transfer functions, and compare them with the experimental bode plots.

3.1 EXERCISE 1: SIMULATED CLOSED LOOP RESPONSE

8. The closed loop model for the DC motor is provided in Figure 3.1. Find the closed loop transfer function $\frac{Y(s)}{R(s)}$ using block diagram reduction.

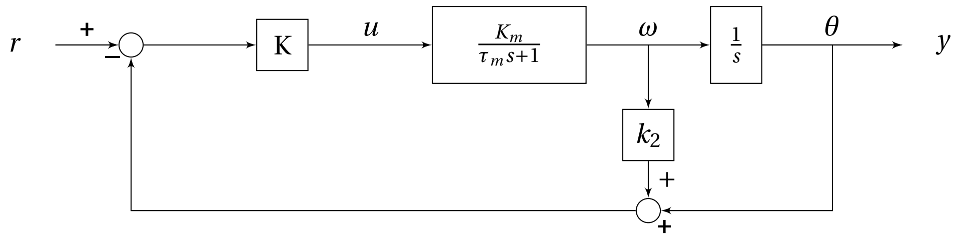


Figure 3.1: Motor Closed Loop Model

9. For $K = 2$ and $k_2 = 0$ find the closed loop poles, the damping ratio, and the natural frequency. Sketch the bode plot of the closed loop transfer function, and save it for your lab report. What type of response is this? Explain.

10. Repeat step 9 for $K = 2$ and $k_2 = 0.5$.

3.2 Exercise 2: Experimental Closed Loop Response

11. Open MATLAB and set the MATLAB "Current Folder" location to the *Experiment_CLFR* folder you extracted in the software setup section. The files *SerialPlotData.m* and *BodePlotData.m* should show in the "Current Folder" section.

12. Open your *OpenLoopFreqResponse* model used in the *Open Loop Frequency Response* experiment, and save it as *ClosedLoopFreqResponse*. Your model should now resemble the model in Figure 3.2.

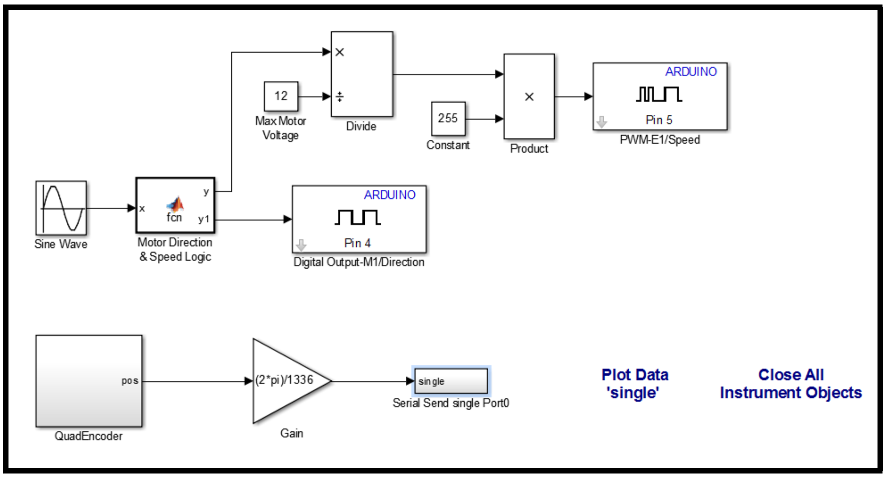


Figure 3.2: Existing Simulink Model

13. Click and drag over the motor control blocks as in Figure 3.3. Then press Ctrl+X to cut the blocks.

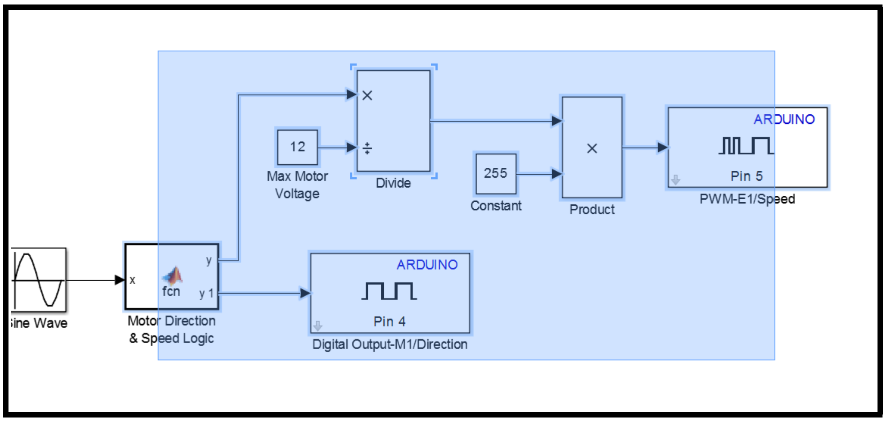


Figure 3.3: Select Motor Control Blocks

14. Add a Subsystem block to the model (Simulink → Ports and Subsystems → Subsystem).

15. Double-click the Subsystem block, and inside press Ctrl+V to paste the motor control blocks. Delete the "Out1" port and connect the "In1" port to the MATLAB function block, as in Figure 3.4.

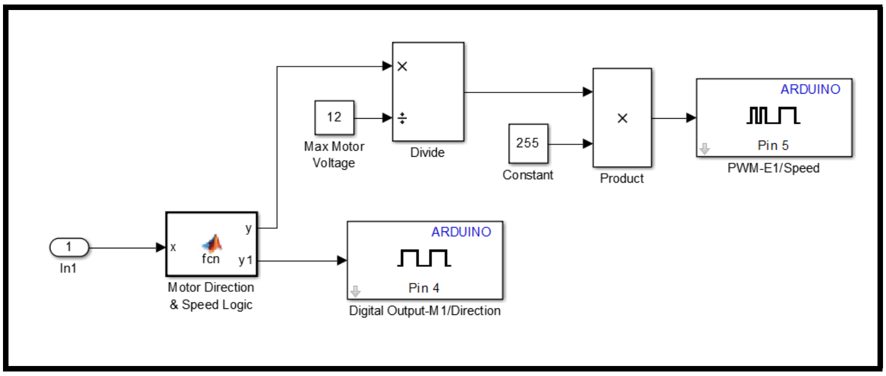


Figure 3.4: Motor Control Subsystem

16. Navigate back to the main model by pressing the "ClosedLoopFreqResponse" text, as in Figure 3.5.

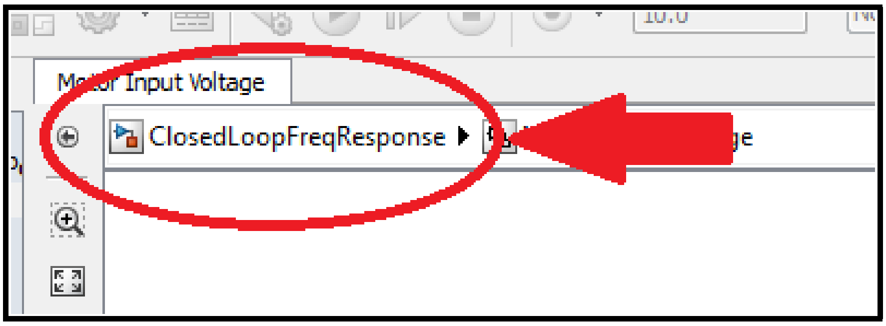


Figure 3.5: Exit the Subsystem Block

17. The model should now resemble Figure 3.6. Add a Sum block to the model (Simulink → Commonly Used Blocks → Sum).

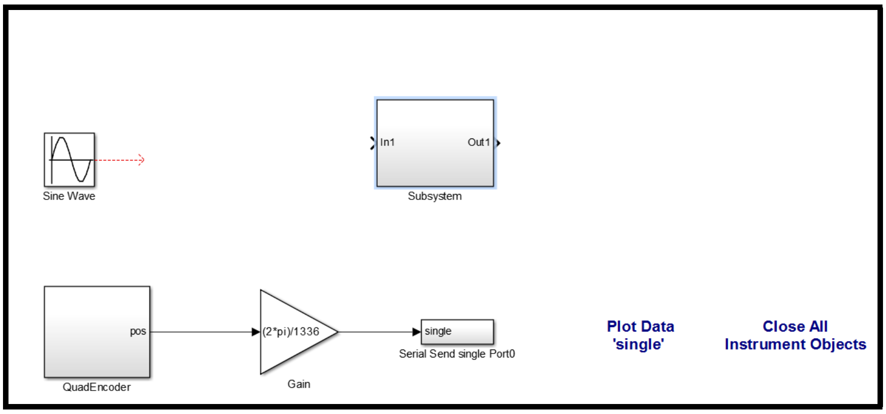


Figure 3.6: Current Model

18. Double-click the Sum block and change “List of signs:" to |+-. Now add the following blocks to the model:
    - Another Sum block
    - 3 Gain blocks: Simulink → Commonly Used Blocks → Gain
    - A Difference block: Simulink → Discrete → Difference
    - A “2 - Serial Send single Port0" block: Take Home Labs Arduino Support Package → 2 - Serial Send single Port0
19. Connect the blocks together, as in Figure 3.7.

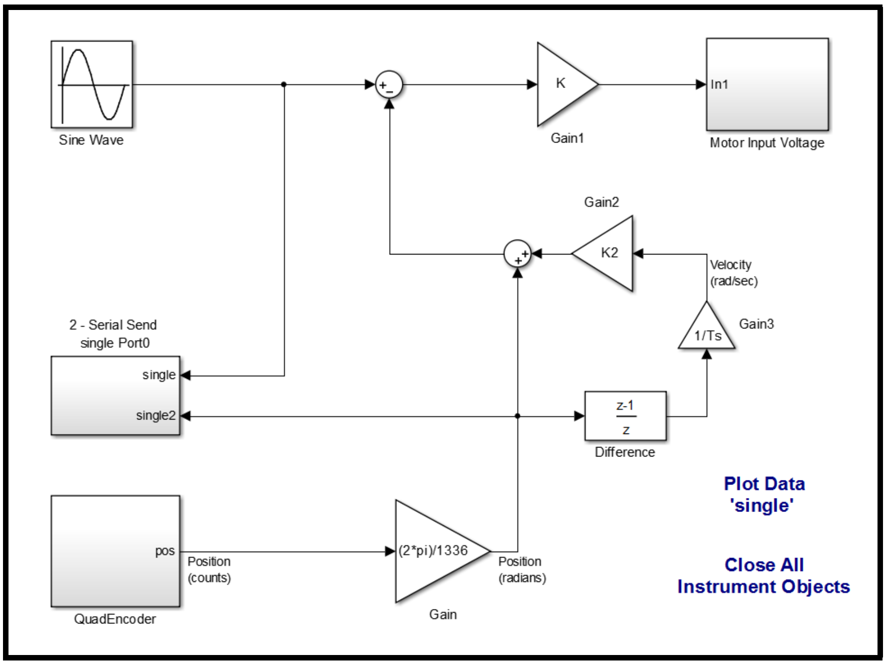


Figure 3.7: Final Model

20. Change the value of the gain block before the subsystem to *K*. Change the value of the gain blocks after the difference block to *1/Ts* and *K2*.

FIRST SET OF GAINS

21. Double-click the sine wave block and change the "Amplitude:" to *pi/2*, and the "Frequency (rad/sec):" to *2*pi*fs*.

22. Create the constant variables you need in the workspace by typing in the following expressions in the MATLAB Command Window and pressing enter after each one:
    - *Ts = 0.01;*
    - *fs = 0.05;*
    - *K = 2;*
    - *K2 = 0;*

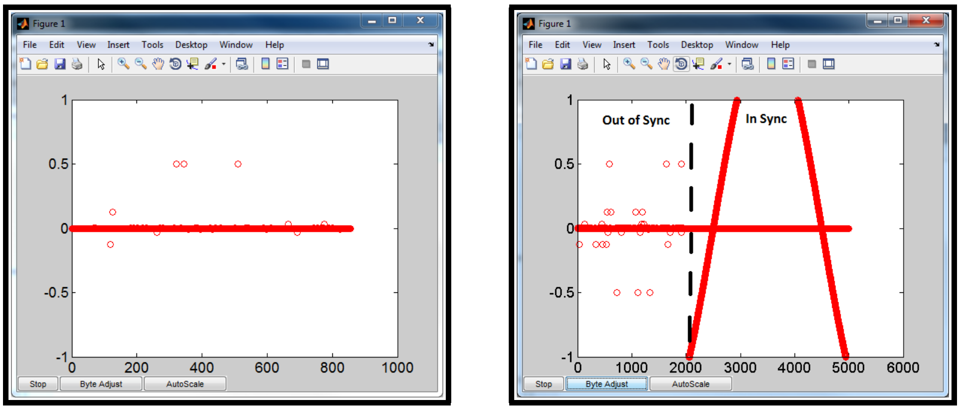
23. Connect the Arduino to the PC, and download the Simulink model to the Arduino.

24. Once the model is successfully downloaded to the Arduino, select the "Plot Data 'single'" text.

25. In the pop-up window, change the COM port to the port your Arduino is using, and set the number of samples to plot to *7500*. Press "OK" to start plotting the data. The input

sine wave and output position are both being sent over the serial port, so there will be two graphs showing in the plot window.

26. If the data is in sync, you should see a sine wave and a constant 0 being plotted. If the data looks like Figure 3.8a, it is out of sync, so press the “Byte Adjust" button. When the data becomes in sync, it will resemble the right half of Figure 3.8b. If your plot does not look like this, press “Byte Adjust" a few more times, if needed. After a few attempts, if the data is still out of sync, press “Stop" and close the plot window. Initiate the plotting process again by selecting the “Plot Data ‘single’" text and entering the number of samples to plot. Repeat this step until the data is correct. Additional techniques for troubleshooting the serial data can be found in steps 79-82 of the *Sampling and Data Acquisition* experiment.



(a) Data Out of Sync

(b) Data In Sync

Figure 3.8: Serial Plot Data

27. Once the data is in sync, plug in the power supply.

28. Press “Autoscale" once the motor starts rotating, if the values are not visible. Let the experiment run until you see at least 2 cycles of the output sine wave before pressing “Stop".

29. After pressing stop in the serial plot window, open the *SerialPlotData.m* file in the MATLAB Current Folder section, and run  the file. The file will extract the reference input and output position, and plot them vs time on the same graph.

30. Find the steady state peak to peak magnitude of the reference input sine wave, and record the value in the third column of Table 3.1. In the fourth column, record the peak to peak magnitude of the output position. In the final column, record the ratio of the output magnitude vs reference input magnitude $\frac{|y|}{|r|}$ in decibels (dB), using $20log_{10}$.

Frequency Response				
Number of Points	Frequency (Hz)	$\lvert r \rvert$ (rad)	$\lvert y \rvert$ (rad)	$\frac{\lvert y \rvert}{\lvert r \rvert}$ (dB)
7,500	0.05			
5,000	0.1			
2,000	0.25			
1000	0.5			
750	1			
500	5			

Table 3.1: Closed Loop Frequency Response

31. In the MATLAB Command Window, type *clear all; close all; clc;* and press enter.

32. Repeat steps 22-31 for each frequency provided in Table 3.1. When returning to step 22, always set *fs* equal to the frequencies listed in the table. The extra rows in the table can be used for additional frequencies. Take as many readings as you need to get an accurate frequency response.

SECOND SET OF GAINS

33. Double-click the sine wave block and change the “Amplitude:" to *3*pi/2*, and the “Frequency (rad/sec):" to *2*pi*fs*.

34. Create the constant variables you need in the workspace by typing in the following expressions in the MATLAB Command Window and pressing enter after each one:
    - *Ts = 0.01;*
    - *fs = 0.05;*
    - *K = 2;*
    - *K2 = 0.5;*

35. Connect the Arduino to the PC, and download the Simulink model to the Arduino.

36. Once the model is successfully downloaded to the Arduino, select the “Plot Data ‘single’" text.

37. In the pop-up window, change the COM port to the port your Arduino is using, and set the number of samples to plot to *7500*. Press “OK" to start plotting the data.

38. Once the data is in sync, plug in the power supply.

39. Press "Autoscale" once the motor starts rotating, if the values are not visible. Let the experiment run until you see at least 2 cycles of the output sine wave before pressing "Stop".

40. After pressing stop in the serial plot window, open the *SerialPlotData.m* file in the MATLAB Current Folder section, and run  the file.

41. Find the steady state peak to peak magnitude of the reference input sine wave, and record the value in the third column of Table 3.2. In the fourth column, record the peak to peak magnitude of the output position. In the final column, record the ratio of the output magnitude vs reference input magnitude $\frac{|y|}{|r|}$ in decibels (dB), using $20log_{10}$.

Frequency Response				
Number of Points	Frequency (Hz)	$\|r\|$ (rad)	$\|y\|$ (rad)	$\frac{\|y\|}{\|r\|}$ (dB)
7,500	0.05			
5,000	0.1			
2,000	0.25			
1000	0.5			
750	1			
500	5			

Table 3.2: Closed Loop Frequency Response (2)

42. In the MATLAB Command Window, type *clear all; close all; clc;* and press enter.

43. Repeat steps 34-42 for each frequency provided in Table 3.2. When returning to step 34, always set *fs* equal to the frequencies listed in the table. The extra rows in the table can be used for additional frequencies. Take as many readings as you need to get an accurate frequency response.

44. Open the *BodePlotData.m* from the MATLAB "Current Folder" location. Input the $\frac{|y|}{|r|}$ (dB) values from Table 3.1 inside the brackets of the *GpdB* variable. Add any additional frequencies to the variable *f*. The frequency variable should be arranged in numerical order, from smallest to largest.

45. Input the $\frac{|y|}{|r|}$ values from Table 3.2 inside the brackets of the *GpdB2* variable. Add any additional frequencies to the variable *f2*. The frequency variable should be arranged in numerical order, from smallest to largest.

46. Run the file to generate the two bode plots. Save them for you lab report, and leave them open. Discuss the relationships between your theoretical sketches of the Bode plot and your experimental results.

3.3 Exercise 3: Simulation and Experimental Comparison

In this subsection, you will use MATLAB to compute the bode plots for the motor's closed loop transfer function you found in exercise 1, using both sets of gains, and compare them to the experimental bode plots.

47. In MATLAB, using the command tf, create the closed loop motor transfer function you found and sketched in Exercise 1. Set the transfer function equal to the variable *Gcl*.

48. Create your K_m and τ_m constant variables in the workspace. Then calculate the bode plot for the gains $K = 2$ and $k_2 = 0$ on the same figure as your experimental plot, by entering the following in the MATLAB Command Window:
    - *figure(1);bodemag(Gcl,w,'--r');grid on;*
    - *legend('Experimental','Simulated','location','Northwest')*

49. Save the figure for your lab report.

50. How well do the experimental and simulated responses match? Explain any differences in bandwidth, low frequency magnitude, resonant frequency, resonant peak magnitude, etc.

51. Again, create the closed loop motor transfer function you found and sketched in Exercise 1, but this time set the transfer function equal to the variable *Gcl2*.

52. Simulate the bode plot for the gains $K = 2$ and $k_2 = 0.5$ on the same figure as your experimental plot, by entering the following in the MATLAB Command Window:
    - *figure(2);bodemag(Gcl2,w2,'--r');grid on; legend('Experimental','Simulated')*

53. Save the figure for your lab report.

54. How well do the experimental and simulated responses match? Explain any differences in bandwidth, low frequency magnitude, resonant frequency, resonant peak magnitude, etc.

4 CONCLUSION

In this experiment you derived the closed loop transfer function of a DC motor position control system, using the motor transfer function that you found in the Open Loop Frequency Response experiment. You then sketched the theoretical frequency response of the closed loop system. You also found the frequency response experimentally, for various types and levels of feedback gains. You calculated Bode plots of the closed loop model using Matlab and compared them to your theoretical sketches and the experimental bode plots. The results demonstrate the capabilities and limitations of using linear models of dynamic systems to predict their performance.