

Lab 8: Frequency Response

Overview

In this lab, students will investigate the frequency response of a DC motor with and without a beam attached to it. This will be done through fixed sine and swept sine inputs. In all cases, the motors will be under feedback control (P or PD). Ultimately, Bode plots will be generated for the motors with and without the beams. Fixed sine points will be overlaid onto the Bode plots from swept sine tests.

Pre-Lab

- Design a beam to attach to the motor hubs
 - your beam must include an attachment for the accelerometer near the tip
- verify that you can do P control of your motor with a sinusoidal input using the provided starter code
- adapt the starter code to do PD control of the motor
- ensure that your Arduino code is printing the correct data to the serial monitor

Procedure

PD Tuning

Before jumping into the frequency response part of the lab, convert the supplied started code from P control to PD control. Then find good values for K_p and K_d .

Part 1: No Beam

For the first part of the lab, you will investigate the frequency response of the motor system without attaching the beam. There are three parts to this investigation:

1. Generating a theoretical Bode plot based on a transfer function model from the previous lab
2. Conduct fixed sine tests and estimate the magnitude ratio and phase lag
 - plot the points corresponding to the fixed sine tests on the theoretical Bode plot
3. Conduct swept sine tests and generate an experimental Bode plot
 - overlay the theoretical and experimental Bode plots, including the fixed sine points as well

Conduct fixed sine tests at multiple frequencies in the range of 1-20 Hz. Plot the desired and actual θ values vs. time. Estimate the magnitude ratio and phase lag for each frequency that you test. Plot these points on your Bode plot. What do you observe about the desired vs actual θ signals as the input frequency changes?

Finally, conduct a swept sine test with the sinusoidal frequency increases over time. The frequency should start at zero and increase to some pre-determined maximum as time increases. You will probably want to run your swept sine tests for a relatively long period of time (5-10 seconds per test perhaps). Keep in mind that if you use a variable n to count the number of time steps a test has been running, the count will overflow at +32,768. So, there is an upper limit on how long a test can easily run that depends on your choice of dt .

Once you have swept sine data, you can use it to generate an experimental Bode plot. This is done by taking the ratio of the `fft` of the output divided by the `fft` of the input:

$$G(j\omega) = \frac{\text{fft}(\text{output})}{\text{fft}(\text{input})}$$

Part 2: With Beam

You will follow similar steps in part 2 after attaching your beam. The primary difference is that you do not have a transfer function model for the system with the beam attached, so you will use the Bode plot to help you identify the transfer function for acceleration/pwm.

You will need to connect your accelerometer to power and ground and then connect the axis you want to measure to an analog input pin. Use the function `analogRead` to measure the accelerometer signal. Make sure that acceleration is including in the data you print to the serial monitor.

It is very important that you do not break the accelerometer or your beam. In order to do this, start with small amplitudes for all fixed and swept sine tests and gradually increase the amplitude until you get good results within the safe input range for the system. You ultimately want to run all of your fixed sine tests with the same input amplitude for the θ_d sine wave. This means finding the largest safe input amplitude near the natural frequency of the beam.

The accelerometer has a range of 3.3V. The output is centered around 1.65V. This means that 0V is the maximum negative voltage, corresponding to -16g. Similarly, 3.3V corresponds to +16g, and is the maximum positive voltage. The Arduino uses 10-bit ADC, so that 5V equates to 1024

counts. That means that 676 is 3.3V on the Arduino. Take care that your accelerometer reading never exceeds 600 or so counts.

Run fixed sine and swept sine tests as you did in Part 1. The input is pwm and the output is acceleration. Generate an experimental Bode plot.

Once you have a Bode plot, use it to find a transfer function for acceleration/pwm.

Report Specifications

Part 1: No Beam

- include 4 fixed sine plots showing θ_d and θ vs. time
 - choose 4 frequencies that span low and high frequencies relative to the dynamics of the closed-loop system
 - discuss what you see in these plots and how frequency affects the fixed sine response
- include a Bode plot that overlays the theoretical Bode from the transfer function model with the swept sine experimental Bode and points from your fixed sine tests

Part 2: With Beam

- show fixed sine plots for 4 frequencies
- generate an experimental Bode plot where acceleration is the output and pwm is the input
- find a transfer function model whose theoretical Bode plot matches the experimental one as closely as possible