# Experiment 4: DC Motor Voltage–to–Speed Transfer Function Estimation by Step–Response and Frequency–Response

## Introduction

This lab introduces new methods for estimating the transfer function of a plant (in our case, the DC motor). Recall that in Experiment 2 the transfer function of the motor was obtained by measuring the various physical parameters of the motor and applying them to the known mathematical model. The methods to be used in this lab, called step– and frequency– response methods, are indirect and can be used even if a good mathematical model of the plant is unavailable.

## Lab 4: Pre-Lab

1. For a first order system, how can the transfer function be estimated from the step response, i.e. how are the DC-gain and time constant found?
2. Consider  as the input to a linear system with Transfer Function G(s). Write down the expression for the output of the system.
3. Write down expressions for the 2 sine waves depicted in Figure 1. The plot is showing magnitude vs. time with time being 0 - 2π s. The larger amplitude signal is the input and the smaller amplitude signal is the output.
4. Sketch the Bode diagram, both magnitude and phase, for the system 
5. Using the Bode plot from the previous question (#4), determine the DC gain and time constant for the system. Mark them clearly on your graph.

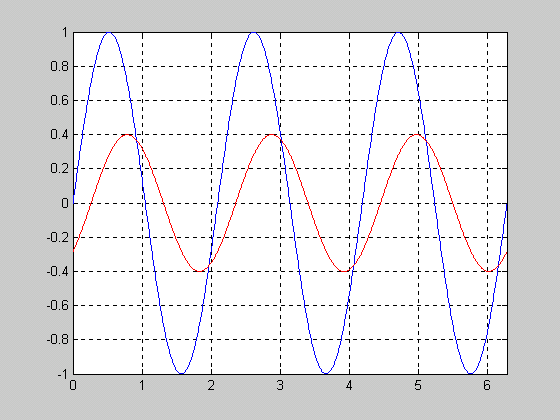


Figure 1

## Simulation of Motor Transfer Functions in Simulink.

You will build a Simulink simulation to compare to the actual output data from the motor. First simulate the transfer function you found in Lab #2. In order to compare this transfer function to the transfer function to be identified in this lab you will have to add the gain of the H-Bridge. The transfer function that will be identified in this lab is

Note: this transfer function does not take into account the static (coulombic) friction acting on the motor’s rotor. We will identify this coulombic friction as a part of this lab but not do too much with it in the analysis portion of this lab. Make sure to record this value though, as it will be important in lab 4’s simulations.

The Raspberry Pi controls the speed of the motor through a technique called pulse-width modulation (PWM). With this technique the voltage to the motor is varied proportionally to the duty cycle of the square wave input to a circuit called an H-bridge. The duty cycle of a square wave is defined as the percentage of the period that the signal is high. Therefore at a 50 percent duty cycle the square wave is high for the half of the period and the motor is supplied with approximately half of the total input voltage. The gain in the figure below and equation above correspond to this conversion from duty cycle to voltage. The transfer functions you created in Lab #2 did not include this conversion.

First start by simulating the motor transfer function found in Lab #2. Use Figure 2 and the steps below to build a Simulink simulation of the motor.

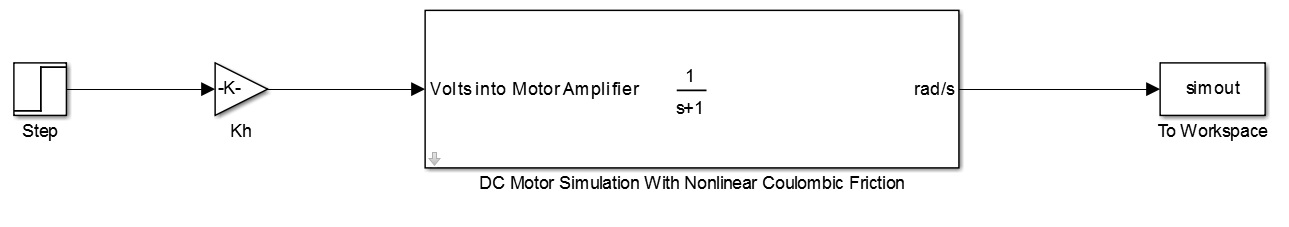


Figure 2 - Completed Lab 3 Simulink Model

Creating a SIMULINK simulation

* Start MATLAB
* Open the model GE320\_lab3starter from N:\labs\ge320\exp3. You will notice that it also includes a “special” motor block that simulates the motor’s transfer function along with the coulombic friction acting on the motor. Throughout the lab this Simulink file will be referred to as the *motor simulation model*.
* Using a unique name, save this Simulink file to the c:\matlab\ge320\netid directory. This is the same folder you have been using all semester.
* Set your MATLAB working directory to the same folder.
* Set the simulation time to 5 sec in the box on the toolbar close to the play button.
* To create the model in Figure 2 we’ll use some simulation boxes.
* From the Simulink Library Browser window select “Commonly Used Blocks”
* Drag a “Gain” block to the Model created in step 3.
* Go back to the Simulink Library Browser window and look for “Sinks”
* Drag a “simout - To Workspace” block to the Model (This box will output your measurements as variables in MATLAB: simout = speed and tout = time)
* Go back to the Simulink Library Browser window and look for “Sources”
* Drag a “Step” block source to the model
* Now connect the boxes so they look like Figure 2 in your lab. To connect them use the small triangles on the edges of each block. Left click on them and drag the connection to the next block.
* Double click on each box and change the values so they match Figure 2 and these items below:
  + In the DC Motor block initially set the motor gain and time constant, Km and τm, to the values you identified in lab 2 and set coulombic friction to 0. Later in this lab you will change these values to ones identified in this lab.
  + Make sure the “Step” block source is set as following:
    - Step Time = 1 (the source is turned on at 1 second)
    - Initial Value = 25 (That is the initial value you input when running the motor in the next step)
    - Final Value = 95.
    - Sample time = 0
  + Make sure your “simout” block has “Save Format” and set it to “Array”.
* After setting up every block, hit the play symbol (“Start simulation”) on the bar, or go to Simulation->Start.
* To graph your response type plot(tout,simout) in the MATLAB command prompt.
* Show your work to your TA.
* Print and label this step response and your simulation block diagram to be included in the post lab.

## Transfer Function Estimation by Step Response

The transfer function of an unknown plant can be obtained by analyzing its step response. As seen in Lab 2, the motor’s Voltage to Angular Velocity transfer function can be approximated by a first order transfer function,

.

The goal of this section will be to estimate the values for Km and τm by analyzing the response of the motor to a step input voltage. Figure 2 above is the block diagram of the setup that will be needed for this experiment. A step function will be input to the motor, and the signal from the tachometer will be observed. Remember that the tachometer measures the angular velocity of the motor. You will also use this step response data to estimate the coulombic friction acting on the motor.

**Testing Procedure:**

* Plug in both AC plugs to the power strip on your bench. One is the Raspberry Pi’s power supply and the other is for the motor. Make sure the red switch on the motor power supply is set to 6V. Wait for the Raspberry Pi to finish booting.
* Copy the Lab 3.slx model from n**:/labs/ge320/kit/exp3/** to the c:\matlab\ge320\netid directory that you used in the last step.
* In the MATLAB command window, run the following command
  + mypi=raspi(‘ipaddress’,’pi’, ’raspberry’), where ipaddress refers to the actual IP address on the label affixed to the base of the kit.
* Open the Lab 3.slx model.
* Set the simulation time to 10 seconds on the tool bar.
* There are two step blocks in this model. First, set the values of the block that is the input to the PWM S-Function. The values should be set as follows:
  + Starting value = 25
  + Final value = 50
  + Step time = 2 seconds.
  + Sample time = 0
* The second step (input into the GPIO pin 18) should be set as follows:
  + Starting value = 1
  + Final value = 0
  + Step time = 9.5 seconds
  + Sample time = 0
  + Note: The purpose of this block is to disable the motor before the simulation finishes so it stops running.
* Make sure the To Workspace block has a variable name of dataout and is set to a format of “Structure with time.”
* Double click on each S-Function Builder and click the build button in the upper right corner.
* Press the play button in Simulink to compile and run the model on the Raspberry Pi.
* The data will be saved in an array in the workspace similar to the previous exercise. To plot the data, type plot(dataout.time,dataout.signals.values) in the command window. This will plot the input and tachometer output in the same figure.
* Repeat the exercise 4 more times always starting with the 25% duty cycle and incrementing your final values as indicated on the data table. NOTE: with your last run from 25% to 95% make sure to also perform the last bullet item where you will compare your simulation to the actual data and estimate the coulombic friction of the motor.
* For your last input from 25% to 95% perform these additional steps to compare your response to simulation and estimate coulombic friction.
  + Double click on the “special” motor simulation block in the motor simulation model and enter your values for Km and τm that match the data you measure this week and initially enter 0 for the coulombic friction.
  + Run the motor simulation and compare your simulated step response with your actual data collected in the Lab 3 model. You should notice some similarities and some differences. If you ignore the starting and ending values of both responses you should find that the rise time and settling time match along with the voltage distance stepped. This should make sense because the gain identification you performed above only took into account the ratio of the difference between the ending and starting output and input values.
  + The differences in the starting and ending values are due to your simulation not taking into account the coulombic friction acting on the motor. To identify this coulombic friction, change the friction value in the motor simulation block until the starting and ending values of the simulation match closely with actual step response data. Start with a value of 0.25 and adjust it up by 0.02 or so until you find the correct value. Record this value in your data sheet and make sure to bring this data sheet to lab 4 as you will need this value and your Km and τm to run the lab 4 simulations.

## Transfer Function Estimation by Frequency Response

A very powerful tool for identifying systems is the *frequency response method.* The key idea behind frequency response techniques is to input sinusoids into the linear system and study the response. A well-known result of linear theory establishes that the steady-state response of a linear system to a sinusoid is a sinusoid, with the same frequency as the input sinusoid, but, possibly, with a different amplitude and phase. The change in amplitude is equal to  and the phase shift is, where G(s) is the transfer function of the system, evaluated at the input frequency. Figure 3 shows a graphical representation of this idea.

G(s)

sin ( ωt )

|

Figure 3 Steady-State response of a linear system to a sinusoid

The goal of this section will be to input sine waves of varying frequencies and measure the ratio of amplitudes of output to input sine waves and the phase shift between the two waveforms. That will provide enough information to identify the linear system G(s) ignoring static (coulombic) friction. The input wave needs to have a DC offset of sufficient magnitude (Why?). When calculating the ratio between output and input amplitudes, you will have to eliminate the DC offsets both for the input and output signals. An easy way to do this is to compare the peak to peak voltages instead of the amplitudes. *From your prelab you should have a good idea on how to use this information to obtain the transfer function of your system. If you still have doubts ask your TA.*

**Testing Procedure:**

* In the Lab 3.slx model you used in the previous exercise, replace the step input with a sine wave input.
* Double click on the sine wave block to change the parameters. The amplitude should be 25 and the bias should be 50. You will have to repeat the experiment for frequencies 0.1Hz, 0.5Hz, 1Hz, 3Hz, 5Hz, 10Hz, 15Hz. You need to identify the interval at which a 45˚ phase – shift occurs (Why?). To find the 45˚ point more accurately pick three more frequencies within that interval and find the gain and phase shift.
* Make sure your simulation time is long enough to capture at least 3 periods of your sinusoidal input.
* Once you have set the simulation time, you will also need to adjust the step time of the step input to GPIO pin 18. It should be set to 0.5 seconds less than your simulation time.
* Press play in Simulink.
* To create the magnitude plot, find the ratio of peak to peak output speed verses peak to peak input voltage. This is call “gain.” When creating your Bode plot it is common to convert this “gain” to units of decibels (dB). A dB = 20\*log10(gain). To measure the phase shift use two adjacent peaks of the input and output waves and measure the time difference between them. The output wave is said to be “behind” the input wave. You need to convert this time measurement to degrees. To do that, you know that 1 period of the wave is 360 ˚ (period = 1/frequency).
* The data will be saved in an array in the workspace just like the previous exercise. To plot the data, type plot(dataout.time,dataout.signals.values) in the command window. This will plot the input and tachometer output in the same figure.
* Create a Bode plot with the data found. One semilog plot of magnitude (in dB) verses input frequency and one semilog plot of phase (in degrees) verses input frequency. Use the command semilogx in MATLAB to plot a log plot, i.e. semilogx(freq,gaindB).
* From these two plots find values for Km and τm. (Post Lab Question #4)

## Lab 4: Post Lab

Include the answers to the following questions in you lab report.

1. Calculate the transfer function for the motor with the data obtained from the step–response experiment.
2. Comment on the similarity/differences between the first–order transfer function estimated in the step–response experiment with the similar transfer function obtained in experiment 2. Remember to account for the gain of the H-bridge.
3. Using the first–order transfer function estimated in the step–response experiment, design an open–loop control system that will spin the motor at 100 rad/sec and 200 rad/sec. That is, you will have to identify the voltage that needs to be applied to the motor to spin the motor at the required speeds. Roughly, how long should we have to wait to achieve the required speed to within 1%?
4. Calculate the transfer function for the motor with the data obtained from the frequency–response experiment.
5. In the frequency–response experiment, why did we need a DC offset?
6. How can we infer the order of the system from the Bode plot of the frequency-response experiment?
7. Include the block diagram from your Simulink simulations (step responses). Just include two step responses that compare the transfer function found in Lab 2 to one of the transfer functions found with the step response method.

# Lab 4: Data Sheet

*STEP RESPONSE OF A DC MOTOR*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delta Vin**  **Step size Input**  **From To** | **Tachometer Avg. Starting Speed (rad/sec)** | **Tachometer Avg. Ending Speed (rad/sec)** | **Delta Speed**  **Tachometer**  **(Ending – Starting)** | **Time constant (sec)** | **Motor Gain (Km)** |
| Step of 25%25% to 50% |  |  |  |  |  |
| Step of 35%25% – 60% |  |  |  |  |  |
| Step of 50%25% – 75% |  |  |  |  |  |
| Step of 60%25% – 85% |  |  |  |  |  |
| Step of 70%25% – 95% |  |  |  |  |  |

## FREQUENCY RESPONSE OF A DC MOTOR

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Frequency (Hz)** | **Xin p-p (%)** | **out p-p (rad/sec)** | **gain** | | **Gain in dB** | | **Δt (sec)** | | **Phase shift (degrees)** |
| 0.1 |  |  |  | |  | |  | |  |
| 0.5 |  |  |  | |  | |  | |  |
| 1 |  |  |  | |  | |  | |  |
| 3 |  |  |  | |  | |  | |  |
| 5 |  |  |  | |  | |  | |  |
| 10 |  |  |  | |  | |  | |  |
| 15 |  |  |  | |  | |  | |  |
| Extra measurements for 45° phase shift | | | | | | | | | |
|  |  |  |  |  | |  | |  | |
|  |  |  |  |  | |  | |  | |
|  |  |  |  |  | |  | |  | |