

System Parameters and Frequency Response

MAE 433 Spring 2012

Lab 2

Prof. Rowley, Prof. Littman
AIs: Brandt Belson, Jonathan Tu
Technical staff: Jonathan Prévost
Princeton University

Feb. 21-24, 2012

1 Overview

In this lab we examine the dynamics of the motor-tachometer system. In the first part we briefly describe a simplified *model* for the system dynamics. The parameters of this model are estimated by making measurements. Using this model, we determine the frequency response using Matlab (we will learn how to find the frequency response analytically later in the semester).

In the second part, we obtain the frequency response of the motor-tachometer system directly by providing the motor with sinusoidal voltage inputs and observing the tachometer outputs. We will find that if the input to the system is sinusoidal, then the response (after transients decay) is also sinusoidal, but with a different amplitude and phase that depend on the forcing frequency.

Lastly, we compare the frequency response of the model with the actual system.

2 Goals

Our hands-on goals for today are to:

- Learn about the important physical parameters that characterize the motor-tachometer-flywheel system
- Become familiar with the Matlab `bode` (pronounced, “boe-dee”) command which plots the system gain (in dB) and phase (in degrees) versus the log of the frequency
- Learn that for sinusoidal forcing one must wait for all transients to decay before true sinusoidal response is observed. The time that one must wait is equal to the time that it takes to establish steady state in a step response.
- Compare theoretical predictions of the frequency to empirical measurements.

There is a brief, one-page lab report that you will hand in before leaving the lab.

3 Modeling of the Motor

The motor can be modeled with an electrical and a mechanical component, see Figure 1. For the electrical model, we have according to Kirchoff's voltage law

$$L \frac{di_a}{dt} + R i_a + e = v_a, \quad (1)$$

where R is the resistance, L the inductance, and e is the voltage generated as a result of the rotation of the motor (electromotive force, or EMF). We also have the “law of the generator” which relates the angular speed ω to the voltage induced by the motor e , where K_e is a constant:

$$e = K_e \omega. \quad (2)$$

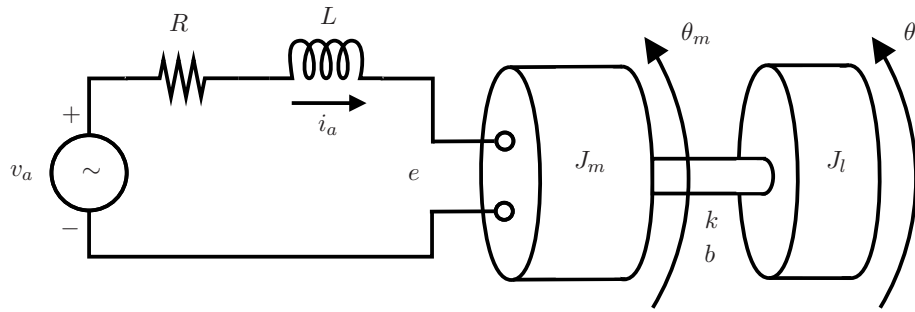


Figure 1: Schematic for the electrical and mechanical model of an electrical motor.

The mechanical equations follow from Newton's law for the rotational moment of inertia

$$J_m \ddot{\theta}_m + b(\dot{\theta}_m - \dot{\theta}_l) + k(\theta_m - \theta_l) = T, \quad (3a)$$

$$J_l \ddot{\theta}_l + b(\dot{\theta}_l - \dot{\theta}_m) + k(\theta_l - \theta_m) = 0, \quad (3b)$$

where k , b are the rotational spring and damping constant respectively, and the driving torque T is given by

$$T = K_T i_a. \quad (4)$$

This is the “law of the motor” and relates the motor torque T to the current i_a through the motor, where K_T is a constant.

Note that internal friction of the motor, stick-slip friction due to the brushes and other couplings present in the motor test stand, is not modeled. For simplicity we introduce two more assumptions:

1. Assume a rigid connection between motor and load (flywheel).
2. Neglect the influence of induction.

The combined set of equations simplifies then to

$$i_a R + K_e \omega = v_a, \quad (5a)$$

$$J \dot{\omega} = K_T i_a, \quad (5b)$$

where $J = J_m + J_l$. Recall that the moment of inertia for a cylinder about its axis of symmetry is given by $J = \frac{1}{2}mr^2$, where m is the mass and r is the radius. We will be using equations 5a and 5b throughout this lab and estimating the parameters R , K_e , J , and K_T .

4 Laboratory measurements

Estimate the rotational inertia of the motor and flywheel

As mentioned, the moment of inertia of a cylinder is

$$J = \frac{1}{2}mr^2,$$

where we will use m in kg and r in meters. We will use this formula to estimate the moment of inertia of the motor's spinning components, J_m , and of the flywheel, J_l , to find the total needed in equation 5b.

Measure J_m

1. Measure the mass of the spinning components of the motor (kg).
You will need a disassembled motor, which might be in use by another group. Feel free to skip ahead to measuring J_l and come back to this step.
2. Measure the radius of the spinning components of the motor (meters).
There are calipers available with the other tools. Remember, radius, not diameter!
3. Compute J_m .
This estimate of the moment of inertia only accounts for the motor. It is not complete because it does not include the moment of inertia of the shaft, couplings, or the encoder.
4. Verify that your J_m is in good agreement with the one specified by the manufacturer.
Refer to the motor datasheet that lists the manufacturer's specifications in the back of this notebook. The motor that we use is the Canon CKT26-T5. Note that the column heading indicating rotor inertia is incorrect – the units should be oz-in-sec²·10⁻⁴. The units of oz-in-sec² are converted to kg-m² by the multiplicative constant $7.06155 \cdot 10^{-3} \frac{\text{kg-m}^2}{\text{oz-in-s}^2}$.

Measure J_l

1. Measure the mass of the flywheel (kg).
Do not disassemble your motor setup. There are extra flywheels in the lab for you to use. You will need a flywheel of the same size as your own.
2. Measure the radius of the flywheel (meters).
There are calipers available with the other tools. Remember, radius, not diameter!
3. Compute J_l .

Compute $J = J_l + J_m$

For comparison, the value suggested for the total rotational inertia for the thin flywheel in your problem set was $J = 2.38 \cdot 10^{-5} \text{kg}\cdot\text{m}^2$.

The calculated moments of inertia and suggested values are to be included in your lab report. Be sure to check with an AI to make sure your values are correct.

Estimate the torque constant of the motor, K_T

Having found the parameter J , we will now determine the torque constant K_T . We will use Equation 5b, which is repeated here for convenience:

$$J \dot{\omega} = K_T i_a.$$

We will apply a constant current $i_a = 0.5 \text{ A}$ to the motor and record its angular speed, $\omega(t)$, as it increases linearly in time. Then we will use the slope of this line to solve for K_T , since J and i_a are known.

1. Create a Simulink model as shown below in Figure 2.

The scope will record the angular speed in units of rad/s. This model will output the angular speed, but doesn't use the Quanser board for the input applied voltage like in last week's lab.

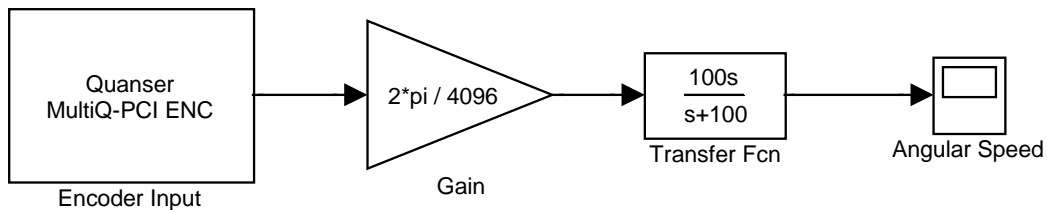


Figure 2: Block diagram for angular speed of the motor in rad/s.

2. Configure the Simulink model by setting the WinCon solver Simulation to ODE5 (Dormand-Prince), as in the previous lab.
3. Connect the motor to the current-regulated DC power supply (GW Instek), then...
 - (a) Before you switch on the power supply, check that both sides of the power supply are set to “independent” tracking, and **have your setup checked by an AI or lab technician.**
 - (b) Switch on the power supply, slide the display switch to “AMPS” and turn the current knob fully clockwise.
 - (c) Slide the display switch to “VOLTS” and set the voltage to 12 V (the motor should be spinning).
 - (d) Slide the display switch to “AMPS” again and turn off the power supply.
 - (e) Start your Simulink model and open up a Scope to plot the angular speed of the motor.
 - (f) Hold the flywheel so that the motor cannot spin, and turn on the power supply. **Do not hold the motor at stall for longer than about 15 seconds at a time or you could damage it.**

- (g) Rotate the Current Control knob until the meter reads “0.5 amps”, then release. You will see the angular speed increase linearly (constant acceleration), following Equation 5b. Stop after a few seconds.
 - (h) Save the Scope data to Matlab.
 - (i) Determine the slope of the line (after initial transients are gone) from current to angular speed.
The slope is the angular acceleration in rad/s².
4. Compute K_T from Equation 5b, using your estimated value of J .
 5. Compare your measured value with the one in the datasheet. **Check with an AI to make sure your value is reasonable.**
To convert the table value in oz-in/A to N-m/A, multiply by $7.06155 \cdot 10^{-3} \frac{N-m}{oz-in}$. Your agreement should be good.

Estimate the back EMF constant K_e and resistance R

We found the J and K_T parameters, now we need K_e and R to complete Equations 5a and 5b. These values are given to you, and require no extra work. The value of this constant for a permanent-magnet motor is numerically equal to the value of K_T (in theory), but with the units volts/(rad/sec). The value of R is listed in the data sheet in Ohms.

A brief set of equations showing your calculation of all of the parameters, as well as the comparison with suggested values, should be included in your lab report.

System Model in Matlab

Now we are ready to model the system using Matlab. The dynamic model for the angular speed can be obtained from Equation 5a and Equation 5b by eliminating the current, i_a , via substitution. The result is

$$\dot{\omega} + K_1 \omega = K_2 v_a,$$

where $K_1 = \frac{K_T K_e}{JR}$ and $K_2 = \frac{K_T}{JR}$.

To make the model complete, we also need a measurement equation. Our measurement will be the tachometer voltage, as in Lab 1. Since the model is formulated in terms of ω in units of rad/s, we need the tachometer constant to convert to voltage. Recall from lab 1: $K_{\text{tach}} = \frac{\text{Tach Voltage}}{\text{Angular speed}}$ in units of Volts/RPM.

1. Restate the equations of motion in state space form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du, \end{aligned}$$

where $x = \omega$, $u = v_a$, and $y = v_{\text{tach}}$. **These calculations must be included in your lab report.**

Make sure to double-check your expressions for A , B , C , D in terms of the variables and constants before you proceed with numerical calculation. Be careful about the units in C and y (rad/s vs. RPM).

2. Create a state-space model in Matlab by defining A , B , C , D and using the command `sys = ss(A, B, C, D)`.
3. Compute the step response by running the command `step(sys)`.
This creates a plot of the step response, and it should look like your open-loop response from last week's lab.
4. Run the command `bode(sys)` to see what the frequency response looks like.
You don't have to do anything with this yet, just observe that you can find the frequency response with this model using this built-in Matlab command.

Collecting experimental frequency response data

We will now find the same frequency response you saw in the Bode plot of the model, but we will do it experimentally by inputting several sine waves of different frequencies and observing the magnitude and phase of the output.

1. Build the Simulink model in Figure 3 with the subsystem shown in Figure 4.
There are multiple sine wave blocks. Use the one from the Sources library.

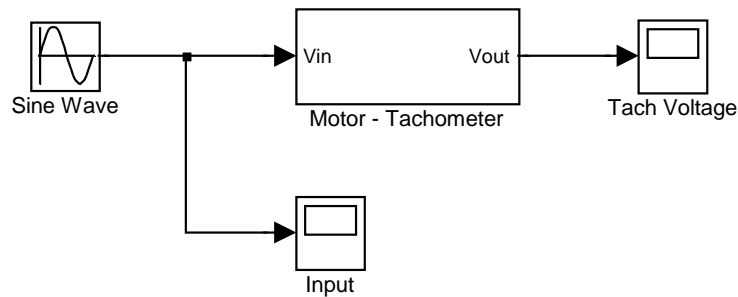


Figure 3: Block diagram for experimental frequency response of the motor-tachometer system.

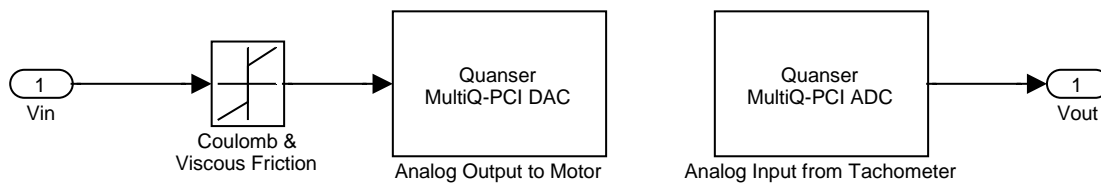


Figure 4: Motor - Tachometer subsystem of the block diagram in Figure 3.

2. Set the range of the Quanser Analog Input module to 10.
3. Set the amplitude of the sine wave to 6 V.
4. Set the Coulomb & Viscous Friction block as you did last week.
5. For each frequency in the set $\{0.5, 1, 2, 5, 10, 20, 50, 100\}$ rad/s, capture 20 seconds of data and save the results in files `resp_to_freq0p5.m`, `resp_to_freq1.m`, `resp_to_freq2.m`, etc. Notice these are “.m” files, not “.mat”. **Be sure to save both the input signal and your motor**

response.

Set the buffer to 25 seconds in the Update pull-down menu. This is most easily accomplished by plotting both signals on the same Quanser scope, which you can do by selecting both the Input and Tach Voltage variables for the Scope.

Comparison of frequency responses: model versus measurement

We are going to compare the model frequency response (computed using `bode`) to the frequency response data you just collected manually at a range of frequencies. First we will determine the magnitude and phase change for each frequency you tested with a function provided to you called `motorbodepoints.m` (this can be downloaded from Blackboard).

1. Download `motorbodepoints.m` to your working directory. Open this function in the Matlab editor to see how it works.
2. For each frequency data set, do the following.
It's not required, but it may save time to write a Matlab script which loops over this procedure.
 - (a) Run the m-file of the frequency response for a particular frequency.
We will be graphically extracting data from this plot. It may be helpful to turn on the grid by typing `grid on` in the Matlab command window.
 - (b) In the plot window, zoom in on a section of the data representative of the settled response containing 2-3 periods.
This should be after a delay of approximately 2s. Recall from the open-loop step response last week that the system's initial transients were completely gone after about 2s.
 - (c) Enter `[x, y] = ginput(3)` in Matlab. This will turn your mouse pointer into crosshairs on the plot.
This allows you to click on the plot 3 times, and the x and y values will be recorded to the x and y variables, each having three entries.
 - (d) Click on the plot three times: first where the input crosses the x-axis (from negative to positive), second where the motor response first crosses the x-axis after the input crosses (from negative to positive), and third where the input period ends, again crossing the x-axis (from negative to positive).
The input period ends on its third crossing of the x-axis. As a check, display the vectors x and y and check that the values of y are all essentially zero and x values are in ascending order.
 - (e) Before closing the plot, enter `[u, v] = ginput(1)`, and click on the apex of one of the motor response curves.
As a check, display the value of v and check that it is the amplitude of your motor response.
 - (f) Set the applied voltage to 6 by entering `vApp = 6;`
 - (g) Run `[magPoint, phasePoint] = motorbodepoints(x, v, vApp)`.
This is computing one point on the Bode magnitude and phase plots.
 - (h) Record `magPoint` and `phasePoint` so we can use them later.
This is best done by entering the points into Matlab arrays. If you're unfamiliar with how to do this, ask an AI.

3. Recreate your state space model from before and have Matlab compute the frequency response of your model with the following commands.


```
sys = ss(A,B,C,D);
[sysMag, sysPhase, w] = bode(sys);
sysMag = mag2dB(squeeze(sysMag));
sysPhase = squeeze(sysPhase);
```

You need the four values that characterize your motor system (A, B, C, D) in your Matlab workspace. Note that we again converted to dB, this time using the built-in Matlab function, `mag2dB`. Feel free to use this in the future.
4. Make a Bode plot of your experimentally measured and model-based frequency responses on. Do they match? If not, which one is “right?” **Discuss with an AI before moving on.** *Remember, a Bode plot contains a magnitude plot and a separate phase plot. For both, plot the input frequency on the x-axis using log-scale. You can do this easily using `semilogx`. The model `sysMag` and `sysPhase` lines should be solid, with measured data overlaid as discrete marks of a different color (for instance, using hollow circles). The experimental data points should match the theoretical model curves. An example plotting command could be `semilogx(w, sysMag, 'k-', freqs, magPoints, 'bx')`. Use `help plot` or `doc plot` to see many other options.*

5 Deliverables

- You need to include all calculations mentioned above:
 - The moments of inertia and comparison to the manufacturer’s values
 - The torque and back EMF constants and comparison to the manufacturer’s values
 - The state space matrices A, B, C, D (scalars in this case)

All results should be expressed in SI-units.

- Comparison of the theoretical and experimental frequency responses. Two plots are to be included: one for magnitude and one for phase.

One report per group is fine.