Prelab 2: Motor System Identification

In Lab 2, we will experiment with several single-input single-output (SISO) control strategies for the Lego motors, to gain hands-on intuition for both transient and steady state behaviors. Before designing any controllers, however, we will first characterize important model elements that capture the dynamics of the plant; ideally, these should capture both linear and nonlinear effects.

First, let us hypothesize a model. Each model assumption is presented first in words and then in equation form. You should need only the equations to complete the prelab; the text is included to justify the model. During lab, you may find evidence our assumed model is not quite right, and so understanding and later refining these assumptions could be useful if you wish to improve performance in future labs.

We will assume the Power Level command to the motor, p_m, (which is a value from -100 to 100 for the Lego NXT system) can be modeled simply as a voltage (of as-yet unknown scaling). In reality, there is a PWM (pulse-width modulated) signal involved, but the inductance of the motor will (we hope) provide enough of a low-pass filtering effect that we can treat the voltage as an average value, proportional (i.e., linear) to the p_m.

$$V_m = K_p p_m$$

We also assume the electrical time constant of the motor is so fast (i.e., $\tau_m = L_m/R_m$ is very small) that the electrical impedance relating voltage and current in the motor can be approximated simply as a resistance, R_m :

$$i_m = V_m/R_m$$

And that motor torque is linearly proportional to current:

$$T_m = K_T i_m$$

Also, the back EMF (electromotive force) reduces the net voltage across the motor by an amount linearly proportional to the angular velocity of the motor. Since power is related as $V_m i_m = T_m \omega_m$, the same motor torque constant relating torque to current also relates back EMF and motor speed:

$$V_{back} = -K_T \omega_m$$

This linear loss can be lumped with any other linear damping due to the mechanism (gears, etc.) itself, collectively represented as a viscous damping term in the dynamics, b_{eff}, that is proportional to the angular velocity *of the output shaft*, ω_s :

$$T_{viscous} = -K_T \omega_m - b_{mech} \omega_m = -b_{eff} \omega_s$$

Nonlinear Coulomb friction also contributes significant losses in the system, due in large part to the sliding and rolling contact within the transmission (gears). Note that the nonlinear force due to friction is familiar to us all in day-to-day life but is quite distinct from the mechanism of linear viscous damping, mentioned in the preceding paragraph. When friction is present, such as when pushing an object across a tabletop, the object may not move at all until some minimal force threshold in force is exceeded. While in motion, friction contributes a more-or-less constant force to oppose motion, regardless of velocity. We will model the friction as providing a torque that opposes the rotation of the output shaft, such that a the shaft will not turn at all until a certain threshold torque is applied by the motor, and then that a constant magnitude force opposes motion while it is rotating.

$$T_{nonlin} = \begin{cases} -T_f & , \omega_s > 0 \\ -sign(T_m) \cdot min(T_f, |T_m|) & , \omega_s = 0 \\ T_f & , \omega_s < 0 \end{cases}$$

After subtracting this nonlinear torque from the motor torque, we would see what appears to be an offset taken from the absolute commanded Power Level, p_m , so that:

$$T_{net} = \begin{cases} K_{lego}(p_m - p_{off}) & , \omega_s > 0 \\ K_{lego} \cdot sign(p_m) \cdot max(0, |p_m| - p_{off}) & , \omega_s = 0 \\ K_{lego}(p_m + p_{off}) & , \omega_s < 0 \end{cases}$$

where K_{lego} accounts for the scaling between p_m and commanded voltage, the motor torque constant, and any gear ratio relating torque at the motor end of the transmission to torque at the output shaft. In the equation above, the basic idea is that we must overcome a minimal torque requirement to get the system to budge, and then a constant torque due to friction will oppose rotation once things are moving.

All individual torque contributions act together to accelerate the "effective inertia", I_{eff} , (i.e., reflected inertia of the motor plus load inertia) that is "felt" at the output shaft:

$$\boxed{I_{eff}\ddot{\theta}_s + b_{eff}\dot{\theta}_s = \begin{cases} K_{lego}(p_m - p_{off}) & , \omega_s > 0 \\ K_{lego} \cdot sign(p_m) \cdot max(0, |p_m| - p_{off}) & , \omega_s = 0 \\ K_{lego}(p_m + p_{off}) & , \omega_s < 0 \end{cases}} \quad (1)$$

The input here is p_m and the output is θ_s, which leaves 4 unknowns in the proposed version of the dynamic model, given in equation (1), above. Specifically, these are:

$$I_{eff} , b_{eff} , K_{lego} , \text{and } p_{off}$$

To determine all four values, we wish to perform simple, repeatable experiments that are relatively quick and easy to do in lab. As a starting point, we decide to run a series of trials where we (1) command a step input in Power Level, p_m , (2) log data from the motor encoder in MATLAB of the resulting rotation, and (3) process the data efficiently within MATLAB. The remainder of this assignment asks you develop a method for doing so.

PRELAB ASSIGNMENT:

Problem 1) Assume you command a Power Level large enough to ensure the motor can overcome friction to move (e.g., p_m=100). What information about any variable(s) can you obtain by estimating the time constant of the step response? (i.e., write an equation relating one of more unknowns to tau.)

Problem 2) Assume you repeat the experiment above for several other Power Levels.
        a) Should all data show the same time constant? Why or why not?
        b) Can you estimate p_{off} from these trials? How? How many trials should this
        estimation, theoretically, require?

Problem 3) If we know p_{off} , then we know what value to subtract from equation (1) and can now relate the steady state shaft velocity to the "adjusted" Power Level (that is, with the offset subtracted out). Write an equation relating the steady state velocity (i.e., the DC gain of the transfer function from torque input to angular velocity output) for any given trial to any of other three relevant unknown value(s).

Problem 4) Estimating the effective inertia is still a problem. However, let us assume that we can add a KNOWN inertia to the output shaft and run another step input trial. How could the new data be used with old trial data to estimate I_{eff}?

Problem 5) Use data on the next page to estimate all 4 of the unknown values.

Problem 6) Use MATLAB (Simulink, ode45, or anything else you think will work is fine!) to simulate the dynamic response of the proposed model, given the following set of parameters:
        $I_{eff} = 0.00001$ (kg-m^2), $b_{eff} = 0.0001$ (Nm/(rad/s)),
        $K_{lego} = 0.00002$ (kg-rad/s^2)/p,   and $p_{off} = 20$ (p)
Submit a print-out of your code and step responses for p_m=40 and p_m=60.

Problem 7) Now, modify your MATLAB code to verify that your estimated parameters would result in the trial data (shown on the next page) used for Problem 5. Provide plots that "look like" the data on the next page, to verify your answers in Problem 5.

Problem 8) Using the values given in Problem 6, assume the motor is used to power a simple arm that extends 10cm from the axis of rotation of the output shaft. What is the maximum mass the arm can slowly lift? (i.e., think about the torque due to gravity to be overcome…)

Problem 9) Now assume $p_{off} = 0$, so we have a linear system; otherwise, use the parameters given in Problem 6. Let us also assume we use a proportional controller, $C(s) = K_p = 5$, to control the output shaft position. What are ω_n (natural frequency) and ς (damping ratio) of the resulting closed-loop system?

Problem 10) For a robot, the total moment of inertia felt at the output shaft may depend on both arm configuration and any load carried. Repeat Problem 9 assuming I_{tot} is 4 times larger than the value of I_{eff} given in Problem 6. (That is, just replace I_{eff} with a value 4 times larger.)

Simulated Data for Problem 5 (Note: Data given do NOT represent the true Lego system!)
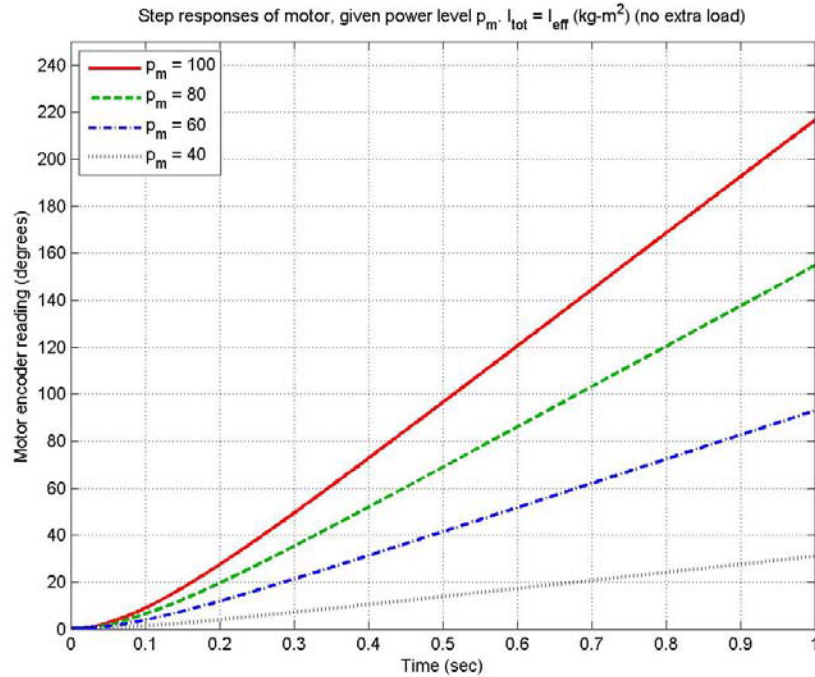

Figure 1. Trials are run with NO EXTRA LOAD on the output shaft. Commanded motor values, p_m, are 100, 80, 60, and 40. No motion is observed when $p_m=20$.
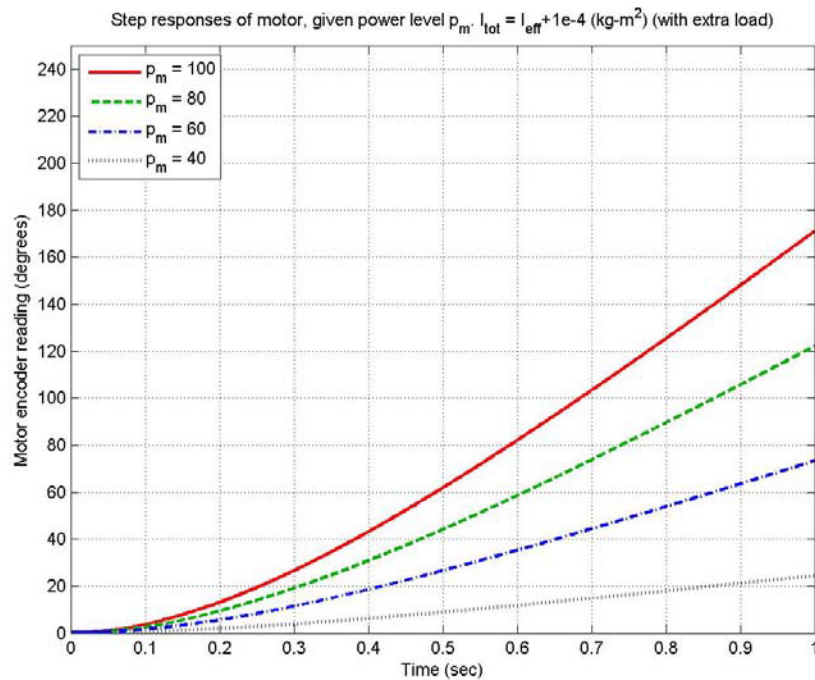

Figure 2. Trials are run WITH an extra load on the output shaft. Although I_{eff} is still unknown, the extra load has a moment of inertia estimated to be 0.0001 (kg-m^2). Commanded motor values, p_m, are 100, 80, 60, and 40. No motion is observed when $p_m=20$.