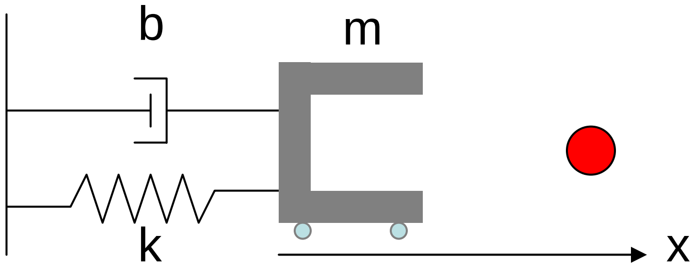
University of Washington
Computer Science and Engineering
Winter 2007

CSE 490 I: Design in Neurobotics

Problem Set 2 Due: 10:30am on 1/18/2007

You will be learning the difference between open and closed loop feedback for a robotic control. We will use MATLAB to analyze the output of this robot control.



Here is a robotic gripper that can move along the x-axis. The gripper that weighs 1 kg (*m = 1* kg) has some resistive force caused by the spring (*k = 1.5 N/m*) and dashpot (*5 N.s/m*) to fight against to get to the ball (1 meter away). The gripper has some wheels on the bottom that is driven by a motor to move left and right. The objectives are to send a desired position in x-axis and observing the behavior with two different types of controllers.

(a) Draw an intuitive (not formal) open loop box diagram that has a box for the following item: controller, motor and robot. Don't worry about the spring, dashpot, etc. They are bundled into the robot box. Label every arrow you draw (with words like "desired position").

(b) You will design an open loop controller. Use commands such as *tf* and *step* to simulate the actual output given that the desired input is 0 before t = 0, and 1 meter at t = 0 (it is a step input) and thereafter. If you are not familiar with MATLAB (or Laplace Transform), you can simply type the following to get the open loop output.

```
robot = tf(1, [1 5 1.5]);          % defines the robot to have the m, b, and k properties
controller = 0.5;                  % defines the controller gain; in this example it is 0.5
open_loop = controller * robot;    % open loop controller has the controller and the robot
step(open_loop)              % give a step input with magnitude of 1 meter and plot the output
```

All units are in metric (position is in meters).

(i) When the controller gain is 0.5 (as the default used above), what happens? What is wrong with the output? Submit the output plot along with your answers.

(ii) You are the person to calibrate the open loop controller. Change the controller gain until the output simulates the desired output. What was the gain you used? Comment on what happens as the gain is increased and decreased. How long does it take to get to 90% of the desired position? Submit the output plot along with your answers.

(c) Draw a formal closed loop box diagram that has a box for the following item: controller and robot (bundled motor). Again, label every arrow.

(d) Now expand on your MATLAB code so that you are giving a proportional control command with a given gain (start with 0.5). You can use the command *feedback* or simply type in the following in MATLAB.

```
robot = tf(1, [1 5 1.5]);
controller = 0.5;
Pcontrol = feedback(controller * robot, 1);  % proportional controller
step(Pcontrol)                               % step input for the closed loop controller
```

(i) When the controller gain is 0.5, what happens, what is wrong with the output? Submit the output plot along with your answers.

(ii) Turn the controller gain and observe what happens. Comment on what happens as the gain increases. How long does it take to get to 90% of the final position with your gain? Submit the output plot along with your answers.

The proportional controller has an inherent problem that was identified in (ii). This can be solved by other controllers you will learn in the lecture on the week of 1/16/2007.

(e) Finally we will show the benefit of the closed loop controller by adding perturbation. We will model the perturbation as follows:

```
perturbation = tf(1, [100 0]);        % perturbation modeled as a small step noise
```

You should add this perturbation to the top portion of your controller for both the open loop (code in (b)) and the closed loop (code in (d)). So “*controller*robot*” will become “*controller*robot+perturbation*.” Use the controller gain that you picked at the end of (b) and (d) and plot the output (submit both plots). Comment on the outcome of the added perturbation (including how long it takes to settle to 90% of the final value).