

CSE 490 I: Design in Neurobotics

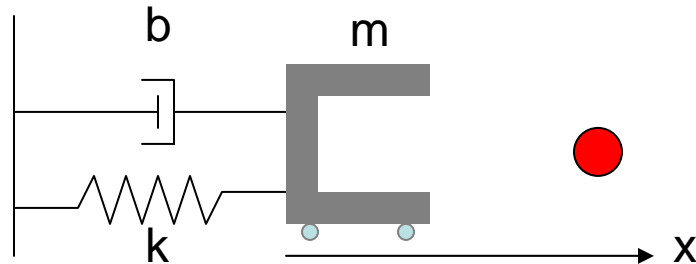
Problem Set 2

Due: 10:30am on 1/18/2007

Answers in RED

Point distribution in BLUE

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").

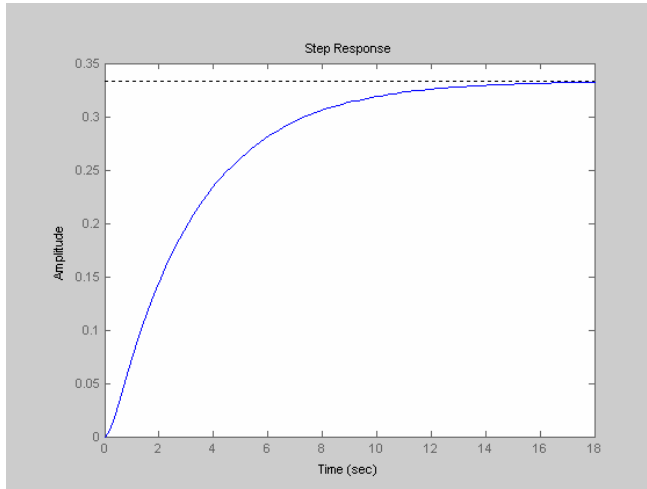
Desired position -> controller -> robot -> actual position (10pts)

- (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).

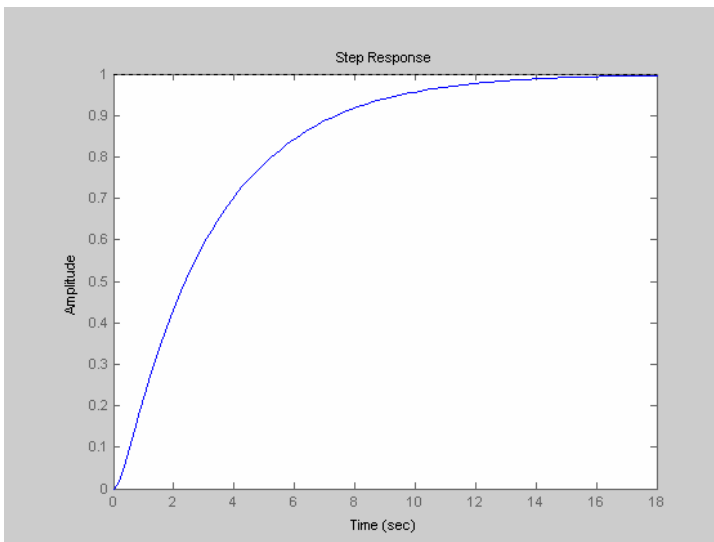
- (i) When the controller gain is 0.5 (as the default used above), what happens? What is wrong with the output? Submit the output plot along with your answers.



(5pts total) Undershoots the desired position by a lot. (5 pts for noticing the undershoot)

- (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.

(10pts total) Gain of 1.5 makes the output to match up to the desired position. (5 pts)
It takes a long time to reach the steady state value (about 8 seconds) this is not acceptable for a real system. (5 pts)



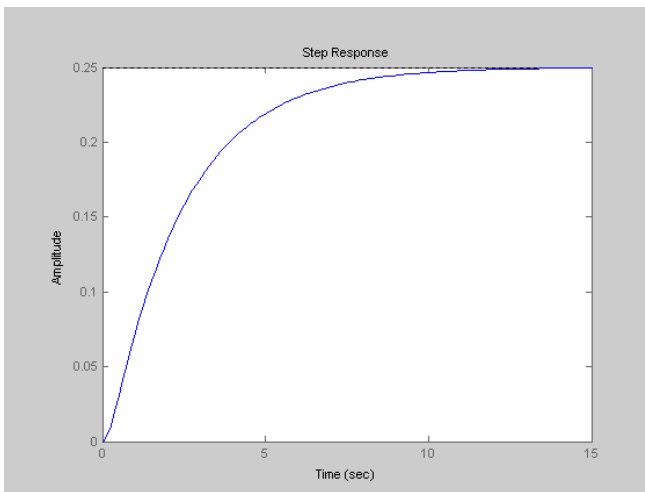
(c) Draw a formal closed loop box diagram that has a box for the following item: controller and robot (bundled motor). Again, label every arrow.

(15pts total) To the open loop controller, add the closed loop arrow and the summation sign (with + and – on the arrows). Make sure that all the arrows are labeled with reasonable names. (15pts with every label missing is 2 pts off including + and – signs)

(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.

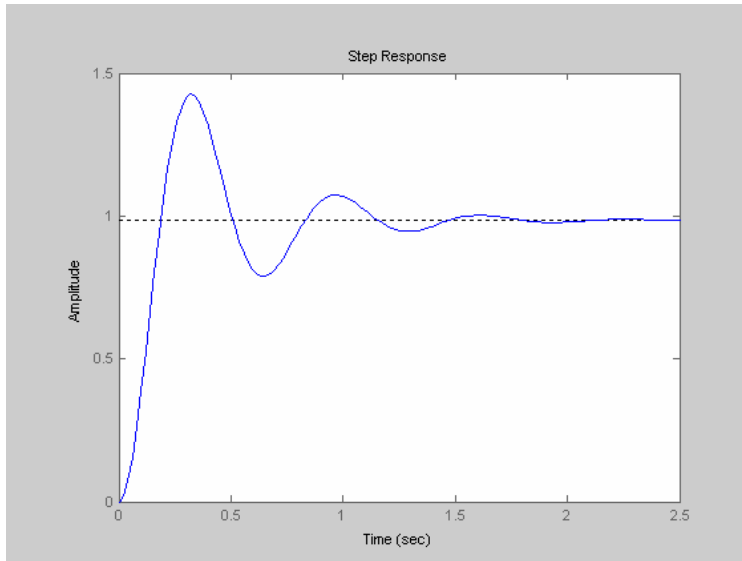
```
robot = tf(1, [1 5 1.5]);
controller = 0.5;
Pcontrol = feedback(controller * robot, 1); % proportional controller
step(Pcontrol) % step input for the closed loop controller
```

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



(10pts total) The output is still no where near the desired position. Also, it is very slow still (takes ~8 seconds to reach the steady state value). (5 pts for undershoot, 5pts for noticing the slowness)

(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.



(20pts total) As the gain became higher, (1) the steady state value got closer to the desired position, (2) it reached that value faster, and (3) oscillated more. (5 pts each for each of the three points.) For my example, it took 1.13 seconds to reach 90% of the final position. (5pts for reporting the 90% number) (Time to reach 90% should be measured when the graph oscillates within the 90% scope. -2 if it wasn't.)

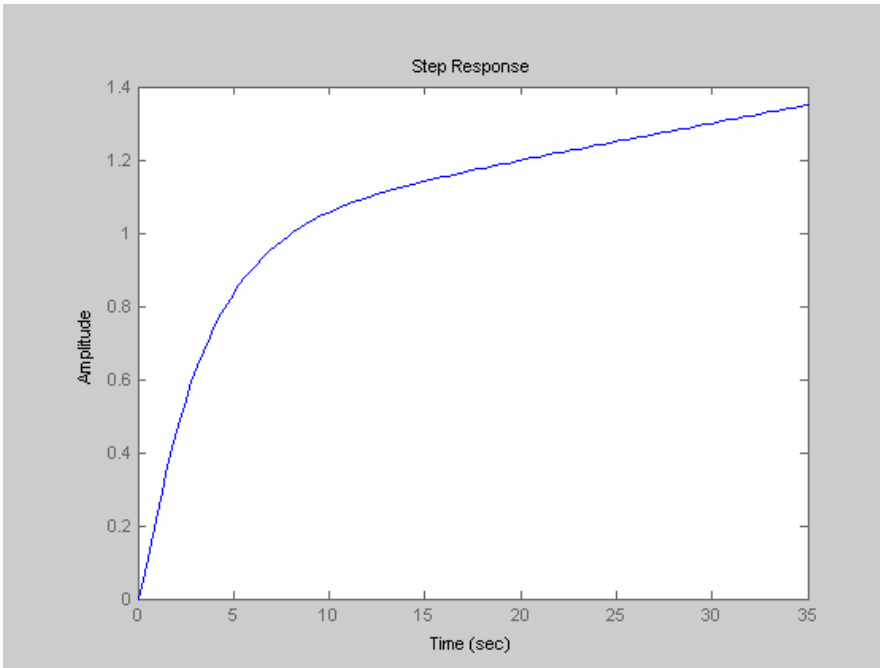
The proportional controller has an inherent problem that was identified in (ii). This can be solved by other controllers you will learn in the lecture on the week of 1/16/2007.

- (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
```

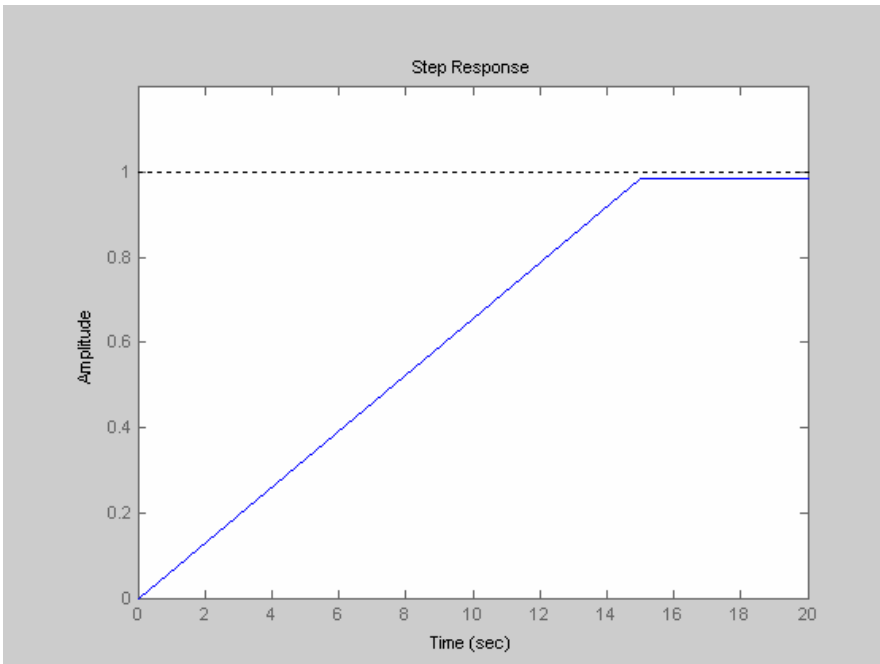
You should add this perturbation to the top portion of your controller for both the open loop (code in (b)) and the closed loop (code in (d)). So “*controller*robot*” will become “*controller*robot+perturbation.*” Use the controller gain that you picked at the end of (b) and (d) and plot the output (submit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

1. Open loop



(10 pts total) It drifted off away from the desired position. This is considered unstable and is a bad behavior. There is no way to fix this problem in open loop control. 90% was reached in 7 seconds. (5pts for the 90% time, 5pts for the drift comment)

2. P control



(20 pts total) It actually reaches the desired position now without oscillating. It takes a lot time (13 seconds) to reach the steady state, so it is not very good, but the overall shape is acceptable. The reason why suddenly the desired position can be achieved is because this specific perturbation acts as an integrator. (5 pts for the zoomed in graph,

5pts for the slow rise comment, 5 pts for noticing that it goes to the desired position, 5pts for 90% time. Do take points off for those people forgot to zoom into the plot and didn't get the answer correct. Also do take points off for those people who said that this is a great response (again probably because they didn't zoom in to notice how slow the response is.)