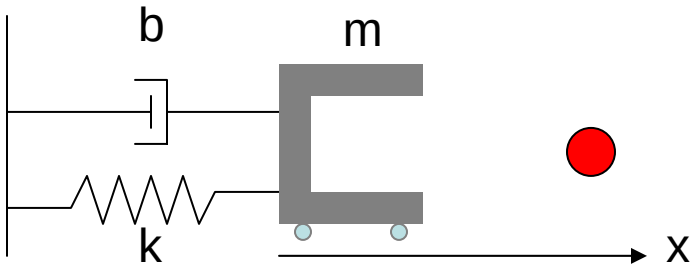
University of Washington
Computer Science and Engineering
Winter 2007

CSE 490 I: Design in Neurobotics

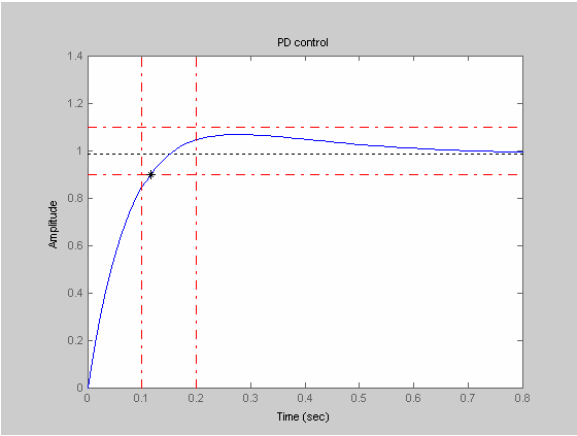
Problem Set 3 Due: 10:30am on 1/25/2007

You will use the same robot as Problem Set 2 to work with other controllers.



(Same explanation of the robot as the last problem set)
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 observe the behavior with different controllers.

1. (35pts total) Try PD controller. You can use the code that was used in class on 1/16/07 (downloadable from the class webpage). Make sure to change the robot parameters in the code from the class example to this one.
    a. Play around with the gains until the output reaches 90% of the desired value between 0.1 and 0.2 seconds. Report the gains you used and provide the output plot (with a star on the point where it crosses 90% of the desired value).

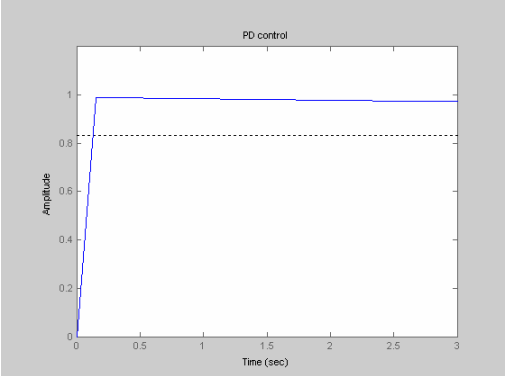


You can see the box of window between 0.1 and 0.2 seconds when the graph had to enter the value between 0.9 and 1.1. In addition, it CANNOT over or

undershoot beyond 0,9 and 1.1 after this time interval (or else it doesn't count as getting into the 90% of the desired value in 0.1 – 0.2 seconds). (10 pts for doing this right, and subtract 3 pts for overshooting after 0.2 seconds, subtract 3 pts for not getting in the box within 0.1-0.2 seconds, and 3 pts for forgetting to put the star when it crosses 0.9)

I used gains: Kp = 60, Kd = 15 (5 pts for reporting this. What they end up picking would be pretty closer to this. If they are way off, you may want to quickly test it and make sure they are not making them up.)

b. Try a reasonably low Kp and a reasonable high Kd until the behavior is undesirable. Provide the plot (and the gains you used) and comment (2-3 sentences) on the shape of the curve based on what you know of PD controller.



Gain used: Kp = 5, Kd = 150 (could be anything as long as Kp is reasonably small and Kd is reasonably big --- it all produces the same shaped curves with different decay constant.). (5pts for reporting the gains)
When Kp is small, the steady state value never reaches the desired value, and that is true in this case also. (5pts for noticing this fact) The derivative component reduces the oscillation, but it makes the convergence to the final point slower. So the larger the the Kd, the slower the convergence is. (5pts for noticing that the oscillation is reduced, and 5pts for noticing that it slows down the convergence.)

2. (25pts) Try PI controller. You can use the code that was used in class on 1/16/07 (downloadable from the class webpage). Make sure to change the robot parameters.
   a. Play around with the gains until the output reaches +/- 10% of the desired value between 0.8 and 0.9 seconds (so none of the oscillations ever go outside of +/- 10% from the desired output value). Report the gains you used and provide the output plot (with a star on the point where it crosses this boundary for the last time).

   Same guideline as 1.a. above. (15pts. Also if they couldn't reach the goal, but explain it well (like justify why they couldn't get there), then that is also good. The gains need to be fairly small)

b. Hold Kp at 100. Vary Ki from 400 to 600 (it helps to plot up to 10 seconds, and use the same time step (set t=0:0.01:10)). Plot two plots (with Ki = 400 and Ki = 600) and comment (2-3 sentences each) on the difference using what you know of PI controller.

The system is out of control and never converges to a final number. The PI controller "integrates" the error over time and uses that to control. So if that is too big with respect to the proportional gain, then it overcorrects for the error and oscillates. The oscillation simply gets bigger with bigger Ki. (10pts for a good explanation, 5pts off if they do not discuss the meaning of "integral.")

3. (40pts) Try PID controller. You can use the code that was used in class on 1/16/07 (downloadable from the class webpage). Make sure to change the robot parameters.
   a. Play around with the gains until the output looks very similar to the desired output. You probably achieve this by setting many of the gains quite high. This is fine for MATLAB, but in real life, sending a very high current to the robot is not recommended (for both the cost and safety reasons). For this reason, assume that you can only have a total of 30 for all three gain values combined. As a robotic control engineer, come up with your one optimal controller and justify it. You should report
      i. The output plot (5pts)
      ii. three gains you used (10pts, take all points off for not making them smaller than 30 total)
      iii. How you came up with it, and (10pts for good explanation (if they wrote a for loop in MATLAB to do it, that is probably the best way))
      iv. Why this is your optimal controller (2-3 sentences) (15pts for a good explanation. They should use terms including "oscillation" or "overshoot", when it reaches 90% of the final value, etc.)