MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.01—Introduction to EECS I
Spring Semester, 2008

Exploration 6, Issued: Thursday, Mar. 13

Due Tuesday or Wednesday April 8 or 9

Exploration: Delays and Sound

Delays

This part of the exploration is worth 4 points.

In the design lab we found that the gains we got by minimizing the maximum pole magnitudes were generally much too large. Using those gains led to "jittery" behavior of the robot. One reason for this is that the large gains amplify small errors. However, another important problem is that the real robot system has additional delays that are not included in our model. These delays are bad because it means that the controller is basing decisions on out-dated data. Our system function analysis should be able to cope with these delays and give us gains that produce good behavior but only if we include the delays in the model.

Exploration 1: Investigate where to put additional delays into the new position controller model, the one that uses $K_1e[n] + K_2e[n-1]$. Are there additional delays that lead to gains more like the ones you found to be good in lab?

Exploration 2: How might these delays arise in the real system? Describe.

Sonars

This part of the exploration is worth 6 points.

We have been using optical sensors for our controller experiments because they are somewhat more reliable than the sonars, which tend to not get return signals unless the sonar beam is nearly perpendicular to the surface. However, with a little care, it is possible to use the sonars to follow hallways covered with bubble wrap.

Exploration 3: Write a program to use the two leftmost sonar sensors to estimate distance to the left wall and the two rightmost sensors to estimate distance to the right wall. That is, emulate the `getLR` function in the lab. Recall that the robots will return some maximum value (often 5.0, but this varies per robot) if they don't detect a return signal. You should make your program be robust to these failures, return your best estimate, even if not a good one, when one or more of the sonars does not get a return (returns its maximum reading)

Exploration 4: Extend your program so that it also calculates the angle of the robot relative to the hallway center line. That is, emulate the `getLRT` function in the lab.

Exploration 5: Debug your programs in SoaR. The `she.py` world in the SoaR distribution has some good hallways.

Exploration 6: Compare the behavior of your controllers from lab 6 using the sonar sensors to the behavior using the optical sensors. Look especially at what happens when the robot is sharply misaligned in the hallway. Make plots of the robot behavior. You should do this on the real robot. The robots are available during the TA office hours.