

Tunnel Robot

Team 1
Lab 3

Daniel Bigelow, Ben Brown, and Brian Vees

Tunnel Robot

Problem Description:

Write a program for your robot to go through a tunnel with a dead end, come out of it from where it started and not get stuck.

This robot will use two touch sensors for its guidance system, and is required to use a random turn when it gets stuck. This random turn should be enough to get it un-stuck again. The robot is modeled after the simple bug from the text, with some slight modifications.

Solutions:

We built the simple bug, and programmed it. The program did the following:

- The robot goes when the start button is pressed, and stops when the stop button is pressed.
- When a bumper is hit, the robot will back up a little bit, then turn away from the side that was hit. A bump counter is also incremented each time a bumper is hit.
- If the counter is incremented more than four times within five seconds, the robot initiates a random turn.
- This random turn will always be greater than the normal turn, and will turn in the opposite direction of the normal turn.
- Each time a bumper is hit, the robot beeps. A single, lower-pitched beep signifies a normal turn; 3 low-pitched beeps signify a random turn, and one high-pitched beep signifies that the counter has been reset.

Problems and Solutions:

The bumpers were not very sensitive, and needed a lot of force to be triggered. We redesigned the bumper mechanism slightly, which helped a bit, although it never became as sensitive as we would have liked.

The initial design for the front wheels prevented the robot from turning very well. We solved this by replacing the front wheels with rolling skids, which worked magnificently.

The random turn function wasn't triggering, because the amount of time it took to execute four normal turns exceeded the amount of time before the bump counter was reset. We fixed this by shortening the turn time, and increasing the time it took before the bump counter was reset.

We also had various small programming issues like this:

- *When the robot executed the random turn, the flags were still set so that a normal turn would happen as well. We fixed this by resetting the flags in the random turn function.*

Most of these programming issues just required simple debugging and tracing of the code to fix.

Conclusion:

Reactive robots can be more complex to make than we originally anticipated. Despite only using two sensors, there were quite a few issues in designing the program to make the sensors work together. Thus, we anticipate that future labs with more sensors may be even more complex.